Access Control In and For the Real World

Sara Sinclair
Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/dissertations

Part of the Computer Sciences Commons

Recommended Citation
Sinclair, Sara, "Access Control In and For the Real World" (2013). Dartmouth College Ph.D Dissertations. 43.
https://digitalcommons.dartmouth.edu/dissertations/43

This Thesis (Ph.D.) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Ph.D Dissertations by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
Access Control In and For the Real World

Sara Sinclair

November, 2013

Technical Report TR2013-745
Department of Computer Science
Dartmouth College
Abstract

Access control is a core component of any information-security strategy. Researchers have spent tremendous energy over the past forty years defining abstract access-control models and proving various properties about them. However, surprisingly little attention has been paid to how well these models work in real socio-technical systems (i.e., real human organizations). This dissertation describes the results of two qualitative studies (involving 52 participants from four companies, drawn from the financial, software, and healthcare sectors) and observes that the current practice of access control is dysfunctional at best. It diagnoses the broken assumptions that are at the heart of this dysfunction, and offers a new definition of the access-control problem that is grounded in the requirements and limitations of the real world.
Acknowledgments

I would like to thank all the people who helped me get to this point and supported me through my research and writing processes.

I thank my committee and my advisor for their counsel and assistance. I thank my dissertation coach, who helped me pick up the pieces, put one foot in front of the other, and reach the finish line.

I thank the executives at the partner organizations who used their leadership positions to clear the path for my work, and the professionals who took the time to share their experiences with me.

I thank my teachers, who pointed the way and provided a foundation upon which I might build. I thank my friends, who cheered me on and commiserated when I hit rough patches. Finally, I thank my family — particularly my husband and my mother — for their unwavering faith that I would see this through.

The research described in this document was supported in part by the US National Science Foundation’s Trustworthy Computing award #0910842, and conducted as part of the Trustworthy Information Systems for Healthcare project sponsored by Dartmouth College’s Institute for Security, Technology, and Society.
# Contents

1 Introduction ........................................... 1
   1.1 This research ........................................... 2
      1.1.1 Access-control realities .................................. 2
      1.1.2 The root of the problem .................................. 3
      1.1.3 Toward better solutions .................................. 4
   1.2 This document ........................................... 5

2 History and related work .......................... 8
   2.1 The beginnings of access control .................. 8
      2.1.1 Introduction to computer systems ..................... 9
      2.1.2 Philosophical groundwork in the protection boom ...... 9
      2.1.3 Mandatory access control (MAC) ...................... 13
      2.1.4 Discretionary access control (DAC) .................. 14
   2.2 Role-Based Access Control ...................... 14
      2.2.1 Role engineering and role mining .................... 15
      2.2.2 RBAC extensions ........................................ 16
   2.3 Other access-control models ................... 17
      2.3.1 Attribute-Based Access Control ..................... 17
      2.3.2 Optimistic Security ...................................... 18
      2.3.3 Break-Glass ............................................ 18
3.5.4 Bluetooth transfer ........................................ 52
3.5.5 Usability .................................................. 52
3.6 New applications and directions ............................... 54
  3.6.1 Offloading keypair generation ......................... 54
  3.6.2 Trusted path and stateful browsing ................. 54
  3.6.3 Wireless certificate issuance ......................... 55
  3.6.4 Human-to-human delegation ......................... 56
3.7 Conclusions .................................................. 56

4 Qualitative methods and corporate study design 58
  4.1 Qualitative methods ......................................... 58
    4.1.1 Objectives in qualitative research ................ 59
    4.1.2 Semi-structured interviews ....................... 59
    4.1.3 Ethnographic observation ......................... 61
    4.1.4 Qualitative data analysis ......................... 61
  4.2 Corporate study design ..................................... 64
    4.2.1 Preliminary work ................................ 64
    4.2.2 Participant organizations ....................... 66
    4.2.3 Background observations ......................... 68

5 Corporate study results: the organization against itself 71
  5.1 Users circumvent controls ................................. 72
    5.1.1 Routing work through the cloud ................ 73
    5.1.2 Password use and abuse .......................... 75
    5.1.3 Modeling the circumvention of controls ........ 80
  5.2 Nature abhors under-entitlement ......................... 83
    5.2.1 A parallel with “system high” .................. 84
    5.2.2 Entitlement errors ............................... 86
6 Corporate study results: factors external to the user

6.1 Complications: culture, compatibility, and clunkers

6.1.1 Corporate culture

6.1.2 Reputation risk

6.1.3 External systems and regulations

6.1.4 The technology ecosystem

6.1.5 Acquisitions and mergers

6.2 Organization maturity

6.2.1 Incremental and iterative philosophy

6.2.2 Expectations of infosec

6.2.3 Fostering a vision of risk management

6.2.4 Self-understanding and structure

6.3 Entitlement review and the IAM lifecycle

6.3.1 Entitlement review timeline

6.3.2 Entitlement review results

6.3.3 Infosec team observations

6.3.4 The next go-round

6.4 Conclusions

7 Clinical study design and results

7.1 Introduction

7.2 The study

7.2.1 Teaching hospital
7.3 Background observations ........................................ 128
  7.3.1 Technical topics ........................................ 129

7.4 Results: user perception and experience .................. 130
  7.4.1 Relating to the machine ................................ 131
  7.4.2 The technology ecosystem ................................ 133
  7.4.3 Of and for the patients ................................ 134

7.5 Results: compliance and circumvention .................... 136
  7.5.1 Under-entitlement ...................................... 137
  7.5.2 Deauthentication and styrofoam cups .................... 140
  7.5.3 Compliance checks ..................................... 142
  7.5.4 Compliance budgets .................................... 144

7.6 Results: the Phalanx Problem (or: bands of roving doctors) 145

7.7 Conclusions .................................................. 146

8 Old assumptions and new wisdom .......................... 148
  8.1 Old assumptions ........................................... 148
    8.1.1 The nature of policy ................................ 150
    8.1.2 Organizational structures ............................ 152
    8.1.3 Knowledge of the users ............................... 153
    8.1.4 Knowledge of the system ............................. 155

  8.2 New wisdom applied to old assumptions ................ 156
    8.2.1 Fundamental properties of humans and organizations 156
    8.2.2 Scoping the old wisdom and expanding with the new 159

  8.3 Conclusions ................................................ 161

9 Rebooting access control ...................................... 162
  9.1 Reframing the problem ..................................... 164
    9.1.1 A functional representation of access mediation 165
9.1.2 Mapping embodies organization objectives ........................................... 169
9.1.3 Context embodies organization state ..................................................... 172
9.2 Types of input “errors” ........................................................................... 175
9.2.1 Omission ............................................................................................. 176
9.2.2 Granularity .......................................................................................... 176
9.2.3 Freshness ............................................................................................. 176
9.2.4 Reliability ............................................................................................ 176
9.3 Choosing a deterrence approach ................................................................. 177
9.4 Attributes of the ex-ante approach ............................................................ 178
9.4.1 Potential for rigorous protection ............................................................. 178
9.4.2 Known quantities .................................................................................. 178
9.4.3 Costs of mapping .................................................................................. 179
9.4.4 Organizational change ......................................................................... 179
9.4.5 Cost of false negatives ......................................................................... 179
9.4.6 Highest priority: minimize over-entitlement ......................................... 180
9.5 Attributes of the ex-post approach .............................................................. 180
9.5.1 Flexible without false negatives ............................................................. 181
9.5.2 Potential for abuse ................................................................................ 181
9.5.3 Costs for set-up and review .................................................................. 181
9.5.4 Break glass ........................................................................................... 182
9.5.5 A history of clinical reviews .................................................................. 183
9.5.6 Optimistic security ............................................................................... 184
9.6 Attributes of the uno-tempore approach .................................................... 185
9.6.1 Reactive access control ....................................................................... 185
9.7 Conclusions ............................................................................................... 186

10 Organization profiling .............................................................................. 188
10.1 False positives .......................................................................................... 191
10.1.1 Probability of damage (A) 192
10.1.2 Gravity of damage (B) 192
10.1.3 Permanence of damage (C) 193

10.2 False negatives 193
10.2.1 Scope of obstruction (D) 193
10.2.2 Magnitude of obstruction (E) 194
10.2.3 Time to correction (F) 194
10.2.4 Cost of correction (G) 195
10.2.5 One-off corrections (H) 196

10.3 Organizational change 197
10.3.1 Turnover rate (I) 197
10.3.2 Function turnover (J) 197
10.3.3 Function dynamism (K) 198
10.3.4 Technical change (L) 199

10.4 User accountability 199
10.4.1 Available consequences (M) 200
10.4.2 Proportionality of consequences (N) 200
10.4.3 Consistency of consequences (O) 201
10.4.4 Awareness of consequences (P) 201

10.5 Contextual information properties 202
10.5.1 Organizational understanding (Q) 202
10.5.2 Function homogeneity (R) 204
10.5.3 Simplicity of necessary data (S) 204
10.5.4 Availability of contextual data (T) 205

10.6 Resources and budget 206
10.6.1 Resourcing priorities (U) 206
10.6.2 Progressive metrics (V) 207
10.6.3 Trust and resource flexibility (W) ........................................ 207
10.7 Example application: healthcare ........................................... 208
  10.7.1 Consequences of false positives ...................................... 209
  10.7.2 Consequences of false negatives ..................................... 211
  10.7.3 Organizational change .................................................. 212
  10.7.4 User accountability ..................................................... 213
  10.7.5 Contextual information availability .................................. 213
  10.7.6 Resources and budget .................................................. 214
  10.7.7 Implications ............................................................. 214
10.8 Comparative example: investment bank .................................... 216
  10.8.1 Consequences of false positives ...................................... 216
  10.8.2 Consequences of false negatives ..................................... 217
  10.8.3 Organizational change .................................................. 218
  10.8.4 User accountability ..................................................... 218
  10.8.5 Contextual information availability .................................. 219
  10.8.6 Resources and budget .................................................. 219
  10.8.7 Implications ............................................................. 220
10.9 Conclusions ................................................................. 221

11 Future work and conclusions ................................................. 222
  11.1 Future work ............................................................... 222
    11.1.1 Direct extensions of my research .................................. 222
    11.1.2 Ex-post evaluation ................................................... 224
    11.1.3 Logs analysis ....................................................... 226
    11.1.4 The Phalanx Problem ............................................... 227
    11.1.5 A grand challenge .................................................. 227
  11.2 Conclusions ............................................................. 228
    11.2.1 Background content ............................................... 228
Chapter 1

Introduction

One fundamental challenge in the study of secure computer systems is measuring how well a given system works in practice. Because the major success criterion is the absence of an damaging event — e.g., the absence of a data breach — it can be hard to argue that a system has performed well unless one has concrete evidence that some adverse event would have transpired in the absence of the system.

The task of evaluating the quality of a secure system becomes even more challenging when it is not just a technical system, but a socio-technical system. This term, first used by Trist et al. in describing research into the impact of mechanization on the coal-mining industry in the 1950s [92], refers to a system in which there exists an interdependence between technology and the people who use it. Unlike a pure computer system, whose performance can often be measured using clear and objective metrics (how much does it cost, how much data does it store, how fast does it compute answers), the success of a socio-technical computer system is determined by a wide diversity of human factors (are users happy interacting with the system, do they understand it, do they see it as valuable).

One important class of socio-technical systems within the field of information security is that which addresses the access-control problem. I define an access-control
system as the set of mechanisms and processes employed to maintain the confidentiality, integrity, and availability of digital resources vis à vis the users who interact with them. As I argue in later chapters, the need to protect digital resources from harm inflicted by users (either malicious or unintentional) has long been recognized in the literature, and significant work has gone into the study of access control over the years as computers have gradually come to play a larger and more critical role in our organizations and society. However, although both researchers and practitioners recognize the importance of access control, and the literature is filled with access-control models, little effort has been made to evaluate how well access-control models actually work in practice. There exist proofs that the state transitions of a given model preserve the security of the data, or that one model can reproduce the security properties of another, but almost\(^1\) no attention has been paid to the critical question: is any of this work on access control actually doing any good in the real world?

1.1 This research

My research represents an initial foray into this critically-important task of evaluating the practical success of access-control tools. Armed with insights from the real world, my work offers novel insights into the root causes of real-world problems. Finally, it offers a new approach to thinking about the access-control problem, one which will better align the work of researchers with the needs of practitioners.

1.1.1 Access-control realities

First, to understand how well access control is working (or whether it is serving the needs of real organizations at all), I have conducted two qualitative studies of large, dynamic organizations. The first is my Corporate Study (described in Chapters 4

\(^1\)See Chapter 2 for descriptions of other recent work in this space.
through 6), which examines the practice of access control through semi-structured interviews of practitioners at two investment banks and one software company. The second is my Clinical Study (described in Chapter 7), which looks at similar topics through interviews and ethnographic observations of both administrators and end users in a large teaching hospital.

Combined, these studies reflect the personal experiences and opinions of 52 people in a variety of industries, and represent one of the largest bodies of research to date on the real-world practice of access control.

As my fieldwork observations indicate, I have found that access control as it is practiced today is profoundly broken. Some of the symptoms of this dysfunction include both frequent over-entitlement, where a user has greater access than she needs, and regular under-entitlement, where users are prevented from doing their jobs because they are incorrectly denied access. The risks associated with the former are easy to imagine, but the latter is perhaps more troubling because it inevitably drives users to circumvent controls — not because these users are malicious, but because they will be (directly or indirectly) penalized by their organization if they are unable to get their jobs done. More generally, and perhaps most distressingly of all, organizations seem to have very little understanding of what types of access their users are enacting — or even what types of accesses are appropriate for a particular user.

1.1.2 The root of the problem

Informed by my fieldwork, I have also sought to understand why the current practice of access control is so mired in dysfunction. Because my observations indicate that the problems are rooted not (as many security researchers might suggest) in a lack of willingness of organizations, laziness of administrators, stupidity of users, or maliciousness of any party, I have sought explanations in our historical approach to the
access-control problem. By examining the early literature and the environments for
which early access-control systems were developed, I have come to believe that the
source of current woes lies in the basic assumptions we have made about the access
control problem. In Chapter 8 I distill my observations into a set of nine flawed
assumptions that the community has made, and offer 14 “new wisdom” maxims to
counteract them.

1.1.3 Toward better solutions

Finally, after documenting the issues with access control and diagnosing their source,
I have embarked on a path toward fixing the current dysfunction.

First (in Chapter 9) I recast the problem of access control as one of access me-
diation, in which an organization brings to bear any technologies or processes that
will help it mitigate the potentially-negative impact of users on its digital resources
(not just technologies that preemptively control users’ access). This more expansive
definition allows us to consider for the first time on a single scale the three basic
approaches to the access-mediation problem:

- *ex-ante* systems, which work by allowing or disallowing access based on policies
defined before the user initiates access

- *uno-tempore* schemes, which rely on real-time access verdicts by human users

- *ex-post* systems, which incentivize users to act responsibly by allowing an orga-
nization to hold them accountable for the accesses they have performed in the
past

After introducing these three basic approaches, I offer the first characterization of
their relative strengths and weaknesses, and suggestions on the situations for which
each is most appropriate.
I continue this guidance in Chapter 10 by providing what is to my knowledge the first taxonomy of features that can determine the success of an access-mediation approach within a particular organization, and whose consideration can assist an organization seeking to deploy or improve an access-mediation solution. I believe that a review of these factors by an organization, with consideration of itself and its digital resources, is the first step in avoiding the problems I have documented in my fieldwork.

### 1.2 This document

As summarized in Figure 1.1, the remainder of this document progresses as follows.

Chapter 2 provides a brief history of the development of access control, and describes a variety of relevant prior work, including research in secure systems, the usability of secure systems, and qualitative research from other fields.

Chapter 3 is a revised version of a text that was co-authored by my advisor and
published as “Secure dongles and PDAs”, a chapter of *Phishing and Counter-measures* published in 2006 (Jakobsson and Myers, eds.) [43], which was itself was an expanded version of a paper presented at *ACSAC* in 2005 [83]. It offers a survey of second-factor authentication mechanisms, including the PorKI system that we devised early in my graduate career. It serves as an example of the type of research I originally expected to pursue in starting discussions with the organizations that eventually became participants in the Corporate Study.

Chapter 4 describes the qualitative methods I used in both the Corporate and Clinical Studies, and introduces the design of the former. Chapters 5 and 6 convey my observations from the Corporate Study, some of which were previously described with multiple co-authors in “Information risk in financial institutions: field study and research roadmap” (published in the proceedings of *FinanceCom* in 2007) [85] and a chapter called “Preventative directions for insider threat mitigation via access control” in *Insider Attack and Cyber Security* [89], a volume for which I also served as a co-editor.

Chapter 7 was released as a technical report in April of 2012 [84], and describes both the structure and the results of the Clinical Study.

In Chapter 8 I present my diagnosis of the issues uncovered in my fieldwork by observing a number of outdated assumptions the research community has made about the access-control problem. This first section of the chapter is a revised version of “What’s wrong with access control in the real world?”, co-authored by my advisor and published in IEEE Security & Privacy in 2010 [82]; the second section is new with this document.

Informed by my fieldwork observations and driven by my diagnosis of the root cause, I offer initial solutions to the issues I document by rebooting our understanding of the access-control problem in Chapter 9 and offering a taxonomy in Chapter 10 of the factors that impact the success of a particular solution within an organization.
The work described in these chapters has not previously been published.

Finally, in Chapter 11 I conclude by describing directions for future research and by offering a few final remarks on this work.
Chapter 2

History and related work

2.1 The beginnings of access control

The phrase “access control” has its origins in the transportation literature of the first half of the twentieth century. Limited-access roads [1] (also referred to as “controlled access” highways), such as the Bronx River Parkway, started in 1907, offered an urban center “a permanent outlet for its fast-growing motor traffic, from the cramped and growing metropolis to the open country” [32].

Although early automobiles were not fast by today’s standards, they did allow drivers to achieve speeds that turned the mixed-use roadways of the day into literal death traps. By forcing cars to enter and exit via one-way ramps, controlled access highways reduce the probability of cross-traffic accidents, and increases the speed at which traffic flows. Although some drivers must have considered the new traffic pattern an inconvenience, they were willing to adapt in exchange for better overall safety and efficiency. (The public quickly demanded that the few intersections on the Bronx River Parkway that were not originally grade-separated be converted [37].)
2.1.1 Introduction to computer systems

By the early 1960s, advances in multiprogramming resulted in systems with more than one input device. Hopner et al. filed a patent in 1961 describing a “keyboard access control system,” for “limiting the number of keyboards accessible to the multiplexing system at any given time” [39].

Corbato’s 1964 description of MIT’s Project MAC touches on memory protection in early timesharing systems, and foresees that “the average commercial user has no desire to let his competitor browse in his records, the project manager doesn’t want his staff to accidentally see each other’s personnel records and salaries” [13]. The next year Dennis described segmentation, which he advocated as a protection mechanism for shared systems that could otherwise be brought down by a single user’s rogue code [17].

By 1969 Hoffman makes the distinction between access controls that are “necessary for a properly operating time-sharing system” (which “protect [...] computer memory from alteration by an errant program”) and those which “enhance data privacy” [38]. Early access control thus served the goal of both the larger community (preserving the correct operation of the system) and of individual users (keeping secret data from one another).

2.1.2 Philosophical groundwork in the protection boom

The literature of the 1970s reflects a surge of interest in computer security. Scientists (and a variety of government agencies) recognized the utility of computers, but wanted assurances that sensitive data would be protected from compromise during computation. Given the literature’s focus on this protection, I think of this period as the “protection boom”.

Lampson’s 1971 “Protection” [51] offers a particularly cogent survey of coetaneous work. In it, he defines protection as:
“a general term for all the mechanisms that control the access of a program to other things in the system, [... including ...] a supervisor/user mode, memory relocation and bounds registers, some kind of file numbers for open files, [and] access control by user to file directories.”

Early papers of this period use “protection” and “access control” interchangeably, perhaps because, as Lampson states,

“... the original motivation for putting protection mechanisms into computer systems was to keep one user’s malice or error from harming other users.”

As layers of software abstraction distanced the user from the low-level operation of the machine, the concept of protection expanded to cover the automated interaction of various computer subsystems, while access control (or “authorization”) maintained a focus on mediating users’ actions. During this period, the community came to see access control as one of several mechanisms necessary for achieving security properties (e.g., confidentiality, integrity, and availability) in a computer system.

**Design principles for protection systems**

Figure 2.1 summarizes eight principles assembled by Saltzer and Schroeder in 1975 to guide the design of protection mechanisms [75]. The authors draw many of these principles from the decade of protection literature that preceded their publication; several on the list have their roots in security knowledge that predated the computer age. While not all are directly applicable to questions of access control, they are representative of the conventional wisdom that came out of the protection boom. I argue that these principles make sense for the computers of the protection boom, but that we should reconsider their wisdom while applying some of them to modern systems.
<table>
<thead>
<tr>
<th>Design principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Economy of mechanism</strong></td>
<td>Keep the design as simple and as small as possible.</td>
</tr>
<tr>
<td><strong>2. Fail-safe defaults</strong></td>
<td>Base access decisions on permission rather than exclusion. (Default deny rather than default allow.)</td>
</tr>
<tr>
<td><strong>3. Complete mediation</strong></td>
<td>Every access to every object must be checked for authority.</td>
</tr>
<tr>
<td><strong>4. Open design</strong></td>
<td>The design should not be secret.</td>
</tr>
<tr>
<td><strong>5. Separation of privilege</strong></td>
<td>Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key. (Avoid single points of failure.)</td>
</tr>
<tr>
<td><strong>6. Least privilege</strong></td>
<td>Every program and user should operate using the least set of privileges necessary to complete the job.</td>
</tr>
<tr>
<td><strong>7. Least common mechanism</strong></td>
<td>Minimize the amount of mechanism common to more than one user and depended on by all users. (Minimize the trusted code base.)</td>
</tr>
<tr>
<td><strong>8. Psychological acceptability</strong></td>
<td>It is essential that the human interface be designed for ease of use.</td>
</tr>
</tbody>
</table>

**Figure 2.1:** Saltzer and Schroeder’s principles for the design of protection mechanisms (compiled in 1975) [75].
Before moving on to describe the major formal reference models for protection systems in the following section, I would like to quote from Saltzer and Schroeder’s description of the eighth principle, psychological acceptability. It is easy to forget (particularly when reading about access-control models) that even early designers were focused on the usability of their systems. Although the literature of the protection boom could not foresee the usability challenges we face today, they understood that the success of any secure system is profoundly dependent on its ability to meet the needs of its human users.

“It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user’s mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he will make errors.”

The early 60s saw access control primarily as a mechanism to ensure availability of the system. As the protection boom progressed, confidentiality and integrity quickly became important, particularly in computer systems within the military and intelligence communities. With time, though, the efforts to achieve these goals came to focus on the more abstract notion of “authorization”: whether or not an individual user was sanctioned by the organization to perform a particular action.

In some ways this abstraction simplified things, for it allowed system designers to have confidence that their data would be protected as long as their system restricted access to users who were authorized. On the other hand, this posed a different challenge: how does one translate the human notion of authorization into terms that a computer can understand? Formal access-control models are designed to do exactly that.
2.1.3 Mandatory access control (MAC)

Perhaps the most famous access-control model from the protection boom is Mandatory Access Control, or MAC. A multi-level security version of MAC (one which relied on users and data objects being assigned a security clearance) was definitively described in the Department of Defense Trusted Computer System Evaluation Criteria, colloquially referred to as the “Orange Book” [66].

Figure 2.2, adapted from one in Sandu and Samarati’s “Access Control: Principles and Practice” [78], depicts the basic structure of a MAC system. There is a central policy administrator who defines whether a user is allowed to access a particular digital object. When the user seeks to access the object, she must first identify herself to the system via authentication. Her access request is mediated by a reference monitor, which queries the authorization policy defined by the administrator to determine whether the access should be allowed.

Note that this vision of access control was somewhat stark: it was assumed that a user’s authorization was determined by a careful, well-governed process, so any user who truly needed access to data would get it — and inversely, users without appropriate clearance had no reason to access sensitive data. Moreover, it assumed
that there was a single source of truth — that the system’s central administrators would be able to figure out a workable policy for the whole organization.

2.1.4 Discretionary access control (DAC)

The Orange Book also defined an alternative model to MAC, one that does not assume that the access-control policy is managed by a central authority. As Jordan summarized in 1987 [44], Discretionary Access Control (or DAC) is:

“... discretionary in the sense that that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.”

The discretionary model allows for there to be more than one data owner in the system: even if the computer is owned by the organization, Alice and Bob may have their respective files to which they manage other users’ access. This approach to authorization management was more scalable, as it allows the administration to be distributed across a larger number of people, and makes it possible for the users who are most familiar with the data to define the policy for accessing it.

This in turn allowed the system to be more flexible, and adapt more quickly as the set of users who needed access to a particular resource evolved over time. However, even when policy management was decentralized, adding or removing users from the system (and updating the policy to reflect these changes) could still be time-consuming for data owners.

2.2 Role-Based Access Control

Since the introduction of MAC and DAC, the alternative (and dominant) access-control scheme has come to be Role-Based Access Control (RBAC) [23]. This approach aims to reduce the overhead necessary for managing permissions for large
numbers of users by first assigning users to one or more roles, and then assigning permissions to those roles (rather than individual users). Since its introduction, significant work has been done to prove that the basic form of RBAC can (in theory, at least) be used to achieve certain data-protection goals, such as separation of duties [81], or simulate MAC [76], DAC [77] or other models.

NIST commissioned a study of RBAC and its economic impact [65], which found that it is “arguably the most important innovation in identity and access management since discretionary and mandatory access control”. Among international companies participating in a survey conducted under the study in 2009, 41% reported some use of RBAC; the authors estimate that “just over 50% of users at organizations with more than 500 employees ... [would] have at least some of their permissions managed via roles” by the end of 2010.

2.2.1 Role engineering and role mining

One challenge of deploying an RBAC system is role engineering, which is the process by which an organization defines roles and assigns permissions to them. Approaches to address this problem are generally either top-down (i.e, starting with a high-level understanding of the organization, its users, and their access needs) or bottom-up (i.e., built from existing permissions) [23]. Some efforts focus on providing structure to top-down approaches, e.g., by providing a scenario framework to allow an organization to understand users’ access needs [63].

However, top-down engineering in a large organization is expensive, even with such frameworks; in recent years many researchers have started exploring bottom-up approaches, frequently referred to as “role mining”. Frank et al. [30] identify requirements for an automated role-mining system, including “minimality, interpretability, and generalizability”. During the Corporate Study I conducted, some organizations were exploring the purchase of role-mining tools, but it was not clear that they would
produce roles that were *meaningful* (or “interpretable”) to the humans who manage the access-control system. There exists some work that seeks to augment automated role-mining techniques in order to produce more meaningful roles [98], and that even applies algorithms to real-world permission data [12], but real-world validation of roles produced using these methods is scarce.

### 2.2.2 RBAC extensions

The literature describing extensions to the basic RBAC model is extensive. Some examples include efforts to model the introduction of obligations [69], and the delegation of privileges from user to user [95] or also from role to role [102].

Significant work has gone into introducing the notion of temporal [45] or geographic [5] constraints in RBAC policies, so that one might restrict access not only to users with a certain role, but to users in that role at specific times of day or at a particular physical location. Some attempts have been made to motivate these types of extensions by describing concrete use cases in which they might be applied [91]. However, I am not convinced that these motivating cases can really attest to the practical utility of such increasingly-complex models, because they leave so many questions unanswered. What is the administrative overhead associated with gathering this additional data, with writing these rich policies? How well can human administrators reason with the primitives provided by these models? How frequently will these policies need to be updated, and how easy is that process?

With the possible exception of models that extend RBAC to interoperate between disparate systems within an organization [46], I thus observe that the majority of extensions are presented without empirical evidence that they would actually be useful in the real world.
2.3 Other access-control models

The research community has proposed a number of non-RBAC access-control models. While many of these are couched in terms of how they might operate in an RBAC environment, they are not necessarily dependent on the existence of roles in a system, and could also work in the presence of user groups or even simple access-control lists.

2.3.1 Attribute-Based Access Control

In recent years there has been a particular interest in the use of attributes in access-control policies. XACML (an “extensible access-control markup language”) [64], exemplifies the trend by allowing policy-writers to specify arbitrary attributes of users (subjects) and resources (objects) and write rich rules limiting access based on different combinations of attributes.

Kuhn et al. [49] explore the possibility of combining RBAC with an attribute-driven approach. They note that the increased flexibility of attributes could mitigate some of the expense associated with role engineering, and reduce the number of roles necessary to capture workflows in dynamic organizations (i.e., prevent “a ‘role explosion’ to cover every possible contingency for permission sets that might be required by users”). They suggest that a role structure could be defined on slow-changing “static” user attributes (e.g., office location), and attribute-based policies be defined for more dynamic properties (i.e., the time of day). However, they do not offer commentary on the manageability of dynamic attributes, nor evidence that the organizations that are currently prone to experience “role explosion” would be able to define and update appropriate attributes over time.
2.3.2 Optimistic Security

Povey introduced the notion of optimistic security [70] addresses the complexity of defining policies in advance by simply abandoning them — or, at least simplifying them to the point that they are manageable, and relying on after-the-fact evaluation to regulate access behaviors. He declares that optimistic security makes “enforcement of the security policy retrospective” by relying on administrators “to detect unreasonable access and take steps to compensate for the action”, including “undoing illegitimate modifications”, “Taking punitive action (e.g. firing, or prosecuting individuals)”, and “Removing privileges”.¹

Although Povey’s description is formal and his practical motivation for the model brief, I find that this approach has a strong potential to be applicable in the real world. (Apparently so do Dekker [15] and others, who have re-invented varieties of the scheme for specific domains, as described in Section 2.6.) In later chapters I discuss systems with this after-the-fact flavor as performing ex-post access mediation.

2.3.3 Break-Glass

Several authors have described ways to introduce “break-glass” controls to existing models [8] and RBAC in particular [25]. The concept behind break-glass is relatively simple: if a user finds she does not have sufficient access to do her job, she can “break the glass” and temporarily escalate her privilege level. A break-glass option can be provided for all actions and users or only a subset, depending on the risks associated with such privilege escalation; it thus has the potential to strike a balance between the inflexibility of classic policy systems and the (potentially too-extreme) openness of optimistic security.

¹This last bullet seems to imply that there still exists a non-“optimistic” policy — i.e., what we consider a normal access-control policy — at work, although the formal system as described by Povey excludes this notion.
2.3.4 Real-time access-control systems

Many other systems exist to help users share data without necessarily requiring a traditional access-control policy, or that allow users to adapt a policy in real time.

One of the simplest is email, which Voida et al. [94] document to be a favored file-sharing mechanism for many users, even in the presence of more specialized alternatives. They observe that users “select which tools to use based on how well the affordances and features of those tools map to the sharing situation at hand”, and that email effectively “constrains the sharing to a specific set of users” and “provides the recipients an explicit cue that new content is available”. Furthermore, email alleviates fears around “availability of a sharing technology for all recipients” or “problems communicating through a firewall”.

The security pitfalls of email as a sharing mechanism — including the ease of re-distribution and the lack of coherent auditing — pose problems in a corporate setting. However, the flexibility of email allows it to respond to users’ needs in real time; Jane sends a message to Bob asking for the latest profits report, and Bob responds with the report attached.

Bauer et al. replace physical keys to a campus building with the Grey System [3], which leverages smartphones to provide a more structured form of on-the-fly policy management. Grey users unlock doors with their phones, and can request access to a room from another user; in this latter case, the other user receives a notification on their smartphone and choose to grant short-lived access, long-lived access, or deny the request altogether.

Similarly, Mazurek et al. [58] study dynamic policy management by creating a simulated “reactive” access-control system. They first solicit from participants a list of contacts and digital resources, and also a policy for the matrix of contacts and

\footnote{This may be why one study in the same year found that up to a third of companies were monitoring their employees email [60] — a number that has undoubtedly gone up in intervening years.}
resources. Over the course of a week they then simulate requests over email or text message from randomly-chosen contacts seeking access to randomly-chosen resources; users are given a choice to grant short-lived access, long-lived access, deny the request, or ignore it. The authors find that 12% of requests are met with an answer that conflicts with participants initially-stated policy, and that 62% of conflicts are due to the reactive decision being more permissive than the initial one. Their survey of participants at the end of the study period reveals a generally positive reception of the experience using the simulated system.

These studies reveal that reactive systems can, in the words of Mazurek et al., “supports many [users] policy creation needs, including the desire for more control and interactivity”. As I discuss in Chapter 9 (where I refer to such systems as enabling \textit{uno-tempore} access mediation), reactive schemes seem well-suited to discretionary access management, in which control is given to distributed data owners who have a high probability of personally knowing the requestor. I also echo Bauer et al. and Mazurek et al. in observing that real-time management seems best-suited to situations where the total number of requests is likely to be small.

### 2.3.5 Economic approaches to adaptive models

Some efforts to understand adaptive access-control models have taken an economic approach. Zhao and Johnson [103] use game-theoretic analysis to evaluate the utility of both penalties and rewards in incentivizing users to only escalate privilege levels when it is in the best interests of the organization. They find that while penalties can discourage undesirable escalations, they can also stifle desirable ones — but that bonus incentives tied to the performance of the organization can compensate. Moreover, they find that in such systems that this need for bonus incentives can be traded off with an improvement in the organization’s ability to accurately and reliably audit user behavior.
With “Fuzzy MLS” (an extension of the MAC-flavored Multi-Level Security model) Cheng et al. [10] attempt to treat risk as a quantifiable resource by measuring the relative gap between a subject’s security-clearance label and the label of the object they wish to read. Each user is then allowed a “line of risk credit”, which they can spend to achieve a limited amount of privilege escalation. Over time users’ access and their credit spend is evaluated (and adjusted depending on whether their accesses were justifiable). I find this work compelling, but am not convinced that one could realistically quantify risk in systems that do not share MLS’s strict hierarchy of monotonically-increasing access levels.

2.4 Usability in access control

Aside from the large variety of formal access-control models that have been proposed by the research community, the literature pertaining to the usability of access-control mechanisms is relevant to my work. In this section I address the topic in general; in later sections I examine work on user compliance and work targeted to healthcare settings in particular.

2.4.1 Authentication

The literature on the usability of authentication schemes is particularly rich. Although a plethora of password alternatives have been proposed (and I myself discuss the merits of replacing passwords with usable PKI authentication in Chapter 3), I choose to focus instead on work that analyzes current user behavior surrounding passwords.

Inglesant and Sasse [41] conduct a study of password practices among end users from two different organizations by asking them to maintain a “password diary”. They find that, despite significant advice and admonitions by administrators or “security
experts” to the contrary, password sharing is alive and well today. They document the lost productivity users experience in struggling with strict password policies, and note that the context in which a password is to be used (e.g., entered multiple times per day on a mobile device) plays a strong role in determining its burden on the user.

Singh et al. [86] conduct a qualitative study of the sharing of banking passwords and ATM PINS among a diverse group of 108 people. Their study includes individuals drawn from populations traditionally under-represented in the HCI literature, such as indigenous peoples, persons with disabilities, and subjects of lower socio-economic status. They find that sharing a bank password or PIN is often the only way for users to conduct basic banking operations. They argue that, rather than trying to prevent end users from sharing passwords, “security design should take into account social and cultural practice and enable the sharing of passwords. The design should minimize the risks, while allowing customers to personalize the degree and limits of sharing”.

2.4.2 Policy-authoring

Maxion and Reeder [57] and Reeder et al. [72] explore the usability of interfaces for managing filesystem permissions. They compare the interface included with Windows XP with novel interfaces that provide “an accurate, clear and salient external representation of the information needed to achieve the user’s primary goal” [72]. They find that these systems vastly outperform Windows XP in terms of user accuracy and time-to-completion for a variety of tasks, including viewing permissions for individual files, making simple and complex changes, and resolving policy conflicts.

Inglesant et al. seek to create a “virtuous circle” of policy authoring [40] by employing controlled (i.e., limited) natural language for crafting policies for distributed computing environments. Their interface allows users to specify a policy with natural language, a graphical interface, or directly in XML — and switch between them.
Most interestingly, the tool allows users to verify the policy crafted using the GUI or a complementary “wizard” workflow by describing the policy back to the user in controlled natural language, with mixed but generally positive results.

### 2.4.3 Access-control case studies

Whalen, Smetters and Churchill [96] present survey and interview data pertaining to user management of file-sharing permissions. They note that “people have to manage access control policies that are both social and technical”, and their study focuses on “how people manage file sharing among various groups, organizations, and tasks”.

An initial survey of 56 employees of a medium-sized (approx. 200 person) technical-research organization revealed an overwhelming majority engage in file-sharing of one type or another (98% via email, 55% via network file-sharing, and 25% via each commercial content-management systems and portable devices). 71% reported that access-permission problems interfere with their work; seven users described intense frustration, which is exacerbated when they encounter the same problems repeatedly. The authors note that “even in a highly technically competent group, with good technical support, problems arise regularly, leading to frustration and difficulty”, and that they found “little correlation between skill level and experience, although it was clear that technical competence led to improved understanding of the issues at large”.

Follow-up interviews with twelve survey participants underlined that, in a technologically diverse environment where users have multiple options for sharing data, social conventions play a strong role in how users manage others’ access to files. The authors also note that interview subjects “revealed a wide range of ‘mental models’ or belief systems around digital content protection and a concomitant range of practices”, and that they were surprised to see such variety among a small sample drawn from a single organization.

Among the design recommendations they offer, I particularly appreciate the fol-
lowing advice:

“Support, rather than replace, social controls: social conventions are a powerful, real-world tool for managing appropriate access, which can provide a simple and flexible shorthand for access policies.”

Overall, Whalen et al. observe that the majority of subjects engage in the management of file-sharing access, and that they do so repeatedly over time.

As user self-reporting of behaviors is not always accurate, Smetters and Good [87] take a “cyberethnography” approach to understanding how users manage permissions to shared files, by applying automated data-mining techniques to ten years’ worth of one medium-sized (approx. 200 person) company’s access-management logs. The organization’s access model is discretionary across the four file-sharing mechanisms studied, in that users can manage other users’ access to files. However, the authors observe that “administrative users are often the last people to modify the majority of groups, suggesting that it is they, not the regular users, that take on the more mundane task of “cleaning up” old access rights”.

In later chapters I discuss the notion of “over-entitlement”, which is when users have more access than they need to get their jobs done, often because this “cleaning up” is not done (or not done thoroughly); Smetters and Good note that their data reveal “a number of classes of such ‘ghosts in the machine’ in our sample; groups with no members, old user accounts that still belong to active groups, or groups corresponding to management functions for parts of the organization that no longer exist. In an extreme case, a user account belonging to a dead person was still active on the system”. They also note that having a limited number of software licenses for one system (DocuShare) appears to have created an incentive for administrators to remove old user accounts, whereas no such incentive existed in the classic OS-shared filesystems (Windows and Unix).
The authors hypothesized that users configure relatively simple access-control policies, but discover more complexity than expected. However, they note some of this complexity can actually be explained by the frequent occurrence of user errors: “users end up defining policies with effects other than what they might have intended, or redundant policies that could in fact be expressed in much simpler ways”.

Unlike the majority of access-control research, which is conducted in the absence of real-world validation and which seems keenly focused on adding features or otherwise enriching access-control models, the authors use their cyberethnographic observations to call for increased simplicity. In particular, they argue that limiting the policy language to only include positive grants of access, simplifying the inheritance model for access-control changes, and limiting the types of permissions that can be granted can go a long way toward making access-management by end users more usable and less prone to errors.

2.5 User compliance

One important observation from my fieldwork is that users will not only fail to comply with access-control policies that get in the way of doing their jobs, but that they will actively circumvent controls that they believe to be unreasonable. Other researchers have also studied why users fail to comply with security policies or advice.

2.5.1 Compliance budgets

Beauteument, Sasse and Wonham introduce the notion of a compliance budget [4] to help understand why users do not always comply with an organization’s security policy.

As summarized in Figure 2.3, they argue that users can internally perceive a number of costs and benefits associated with compliance. For example, a user may
choose to comply because she does not want to be responsible for a security breach in general, but may avoid compliance at the end of a tiring day because it represents an additional cognitive burden. They note that compliance can be seen as a type of “organizational altruism” when it imposes “pain but no [perceived] gain” to end users, and that individual users may have different thresholds for enduring that pain — i.e., different compliance budgets.

Beautement et al. also note a number of external factors that can influence an individual’s compliance budget. Among them, the design of the security system, the visibility and quality of the organization’s monitoring, and both the availability of sanctions and the consistency with which sanctions are imposed are within some measure of the organization’s control. Other factors, such as user awareness, the efficacy of user training, and the organization’s culture can be harder for an organization to change. (Beautement and Sasse later explored some of these themes in a solidly-constructed study of employees at a utility company [79]. Among other results, they found that there were significant differences in user behavior between the US and UK employee populations, even though both populations held similar attitudes with respect to compliance with security policies)

Most importantly, the authors argue that non-compliance among non-malicious users is an expected and predictable outcome of several observable phenomena, and not a result of user stupidity or laziness.

2.5.2 Rational rejection of guidance

Hurley [36] argues that average users are often acting rationally when they choose to ignore the guidance currently given to them by security experts. He examines three such classes of canonical guidance, pertaining to:

1. choosing a strong password

---

3One participant in my Clinical Study asked me for advice on what type of constraints his
<table>
<thead>
<tr>
<th><strong>Internal factors</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Costs</strong></td>
<td>Increased physical load</td>
</tr>
<tr>
<td></td>
<td>Increased cognitive load</td>
</tr>
<tr>
<td></td>
<td>Embarrassment</td>
</tr>
<tr>
<td></td>
<td>Missed opportunities</td>
</tr>
<tr>
<td></td>
<td>The “hassle factor”</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Avoiding consequences of a security breach</td>
</tr>
<tr>
<td></td>
<td>Protection from sanctions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>External factors</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>System design</td>
<td></td>
</tr>
<tr>
<td>Awareness, training, and education</td>
<td></td>
</tr>
<tr>
<td>The culture of the organization</td>
<td></td>
</tr>
<tr>
<td>Monitoring</td>
<td></td>
</tr>
<tr>
<td>Sanctions</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.3:** A summary of factors that go into users’ compliance budget calculations, as described by Beaument et al. [4].

2. protecting against phishing by carefully examining web URLs

3. defending against man-in-the-middle attacks by heeding certificate errors

By carefully comparing the benefit each type of guidance is intended to achieve with the actual experience of users, Hurley makes a convincing case that this guidance on the whole causes more harm than good. However, he also notes that “security advice that has compelling cost-benefit tradeoff has real chance of user adoption”, but that “the costs and benefits have to be those the user cares about, not those we think the user ought to care about”.

In that vein, Engelman, Molnar, Christin, Acquisti, Herley and Krishnamurthi [20] describe an experiment through which they examine factors that can impact users’ tolerance of computer delays. They asked subjects to “count the total number of times a certain term was repeated in a multi-page document” (a task that lends itself well to cheating) using Amazon’s Mechanical Turk service, and exposed subjects in the experimental groups to delays that made the task take longer.

organization should impose on its users’ passwords; I include the document I gave him in response as Appendix A.
They found that “subjects were significantly more likely to cheat or abandon the task when provided with non-security explanations or a vague security explanation for the delay”, but that subjects who were first primed with detailed information about the type of attack that the supposed “security delay” was designed to protect against had a similar occurrence of cheating or task abandonment as members of the control group.

This indicates that providing a user with a more concrete understanding about the nature of a security delay may make her more tolerant of it. However, it is not clear that this result would be replicable in conditions where users experience repeated or regular delays as part of their daily life.

2.6 Access control in healthcare settings

Recent years have seen significant interest in the security requirements of healthcare organizations, and the operational constraints that the healthcare setting places on secure systems.

In the context of a European trend toward digitization of health records, Ferreira et al. [24] survey publications on 1) access control in healthcare and 2) access control in general, and find that the focus in the domain-specific work closely mirrors the generalized research. In particular, they note that “there is a great interest in defining and studying access control models”, but observe that “this kind of academic modelling approach works because the vast majority of the models were not implemented in practice”, and that “proper system evaluation is needed before one can conclude that these models are either appropriate or effective”. The conclusions they make as part of this survey closely mirror those I have observed directly during my fieldwork.

Heckle [35] describes her observations studying the deployment of a SSO (single sign-on) technology in a regional hospital. As I did in my Clinical Study, she employed
ethnographic observation to capture “the interplay between formal and informal systems”; her work focuses on the process of adopting a single technology, from the initial vendor selection to organization-wide deployment.

She documents “policies that were ill defined for the work environment”, and instances of what I call “product” thinking instead of “process” thinking (e.g., hospital officials expected the SSO rollout to take six months, but experienced significant delays “as the complexities of the implementation began to unfold”). She reports that the hospital in question experienced significant benefit from conducting a pilot study of the SSO technology, but that the team deploying it was caught in a fundamental conundrum: were they “to design the SSO system to correspond to organizational policy or to fit with the actual work practices”? I find the following quote to be particularly evocative of the spirit of my own Clinical Study findings:

“In clinical areas, management must accept that staff can’t always follow policy 100 percent, and systems can’t be 100 percent secure for quality multipatient care.”

2.6.1 Auditing

Dekker and Etalle [15] describe an “Audit-based access control for electronic health records”, which is very similar to Povey’s optimistic security [70] (described in more detail in Section 2.3.2), although motivated by and crafted specifically to satisfy HIPAA requirements. In describing this work they note that medical records must be both available and confidential, but that HIPAA “does not require that every risk of an incidental use or disclosure of protected health information be eliminated”.

While they offer a cogent analysis of some of the strengths and weaknesses of an audit approach to access management in comparison to a policy-based approach (i.e., one offers flexibility while the other aims for strong prevention of misbehavior), their work is firmly grounded in the theoretical. In particular, they do not appear to
consider the burden that the additional logging tasks or formalized auditing process would have on users of the system.

Mashima and Ahamad [56] describe and prototype a “patient-centric monitoring agent” that allows users to audit the way their health data is used by notifying them when their records are shared. While the protocol is interesting, the authors do not validate their assumption that patients can be reasonably equipped to “determine whether the sharing is reasonable”. (In fact, although the human factors would seem to be critically important to the success of any scheme in this vein, their evaluation is focused solely on the performance overhead associated with the protocol they define.)

2.6.2 Mining roles and policies

Zhang et al. [101] describe the application of a role-mining algorithm (which they inexplicably re-name “role prediction”) to three months of health-record access logs from Northwestern Memorial Hospital. They attempt to measure the accuracy of the algorithm by first defining three levels of “correct” roles for each user based on 1) the user’s formal job title and 2) the opinions of “several clinicians at Northwestern”. The levels are designed to be of augmenting abstraction, such that a user with a “Unit Secretary 1” job title could be classed as the increasingly general “Unit Secretary 1” (the “specific position” role, of which there were 140) “Unit Secretary” (the “general position”, of which there were 62), “Admin” (the “conceptual position”, of which there were 5).

Training and testing the algorithm at each of the different levels of role abstraction yield 51% and 52% accuracy for the “specific position” and “general position” roles, and 82% accuracy when evaluated at the “conceptual position” level (at which a given user can be grouped into one of five roles). When allowing for some users to have roles defined at a different level of abstraction, and optimizing the algorithm for accuracy, it predicts “correct” roles for 63% of users. Given that the authors expect
this role-mining approach to be used in conjunction with break-glass controls, and that they maintain the current state-of-the-art for role engineering in organizations is to adopt a user’s job titles as her roles, perhaps this approach could prove beneficial in helping manage the number of times users need to employ the break-glass mechanism. However, without further evidence I remain skeptical of the practical benefits of this approach.

Malin et al. [53] take a different approach to the application of data-mining in support of access control. Rather than trying to directly derive an access-control policy, they mine access logs to deliver data that “enables the review of predefined policies, as well as the discovery of unknown behaviors” that can occur in the organization’s dynamic teams, which are “more complex than the pairwise relationship typically expected by access control frameworks (e.g., provider-patient)”_. Their mining of user-to-data and department-to-department relationships finds that:

“(1) [Healthcare organizations] are dynamic environments in which associations fluctuate over time, (2) departmental interactions [e.g., those where members of one department access data originating in another department] are more stable than those of [individual] EHR users, and (3) intra-departmental relations tend to be more likely than inter-departmental.”

Gunter, Liebovitz, and Malin argue that this sort of data mining can be repeatedly applied by an organization to evolve its access-control policies over time [34].

### 2.7 Commentary

The work in access control and information security for specific domains is extensive and varied. However, it is dominated by formal models and theoretical frameworks that are rarely grounded in empirically-demonstrated need, and even more rarely validated in real-world settings. The usable security literature offers insights into
some aspects of the real-world access-control problem, but is often focused on end-user experiences with graphical user interfaces. I believe that understanding the real-world practice of access control requires both a broad understanding of the challenges that users and administrators from a variety of domains experience and an effort to reconcile those challenges with the current trends in formal access-control research.
Chapter 3

PKI Devices: secure dongles and PDAs

This chapter reprises a body of work that our lab published in 2006 as “Secure Dongles and PDAs”, in *Phishing and Counter-measures: Understanding the Increasing Problem of Electronic Identity Theft* (M. Jakobsson and S.A. Myers, eds.) [43], and for which I was the lead author. I include\(^1\) this work here because it is indicative of the type of research I had planned to conduct after initial fieldwork with our financial partners: inspired by a problem at the intersection of human factors and information security (phishing and credential exposure in general), we designed a tool that leveraged usability insights to increase the overall security of the system. I still think that work crafting tools in this vein would be valuable, but the rest of this dissertation deals primarily with problems that currently obstruct the utility of development efforts in real-world organizations.

\(^1\)For the most part I include the text in its original form, but offer commentary on some of its obviously outdated statements.
3.1 Introduction

Humans are, generally speaking, poorly equipped to distinguish situations in which it is safe to use their credentials from situations in which their credentials could be easily compromised. For example, when a user is asked to enter her password into a web page authentication form, she will likely judge the trustworthiness of the website based on its look and feel, qualities that are easily spoofable. Moreover, the passwords themselves are vulnerable when disclosed to an untrusted party; once the secret is out, there is no way for the owner to get it back. Some ways of shoring up the strength of the password mechanism have already been presented, such as Ross et al.’s hashed password scheme [42] that creates a custom password for every site to which the user authenticates, and which includes a hash of the website name in the password itself. SecureID tokens use a device-generated PIN for authentication. Users must provide their username, password, and current PIN to log in to the website; because the PIN is short-lived, many current phishing attacks are prevented [73]. Other schemes rely on a trusted third party and an extra device, such as Wu et al.’s [59] work with cell phones and proxy servers to bypass the workstation while entering private information.

A user authenticating with her public key cryptography credentials\(^2\) in a public key infrastructure (PKI) also does not disclose private information, even when the party she is authenticating to (the relying party) is untrusted. This safe authentication can combat many phishing attacks. (We review below the building blocks of a PKI, and explain how it can be applied to fight phishing.) However, PKI is difficult for the average user to understand and use [97], and thus is not currently a practical

\(^2\)Throughout this chapter I use the term “credentials” to refer to the data a user has or can generate to authenticate herself to a relying party. In PKI, this almost always means a certificate and a proof of possession of the private key that corresponds to the public key in the certificate. We assume that when PKI credentials are being used, there is an appropriate protocol in place for the authenticating party to offer proof of private key possession without actually revealing the private key. Such protocols are standard in public key cryptography.
replacement for passwords. Furthermore, it brings with it new vulnerabilities of key compromise that are at least as bad as password compromise. Also, PKI does not offer a solution to the fundamental problem of users being unable to judge the trustworthiness of the environment in which they are using their credentials: if they store their private keys on untrusted workstations, those keys can be compromised and abused by attackers.

In this chapter we consider the use of personal hardware devices (smart cards, USB dongles, and PDAs) to protect, transport, and use PKI credentials for authentication, with the goal of using a PKI-enabled hardware device as a full-on password replacement. We focus on the experimental system PorKI, developed in the PKI/Trust Laboratory of Dartmouth College\(^3\). PorKI not only provides secure authentication, but it also enables enterprise users and the relying parties to whom they are authenticating to make informed trust decisions about the workstation from which they are authenticating. Because PorKI and the specialized PKI devices we discuss in this section are particularly applicable in enterprise environments, they would be a good countermeasure against spear phishing or other context-aware attacks, although could also be deployed to prevent more generalized attacks as well.

### 3.2 The promise and problems of PKI

PKI researchers often talk about generic users Alice and Bob. Alice, as a user in a PKI, has a keypair composed of a *private key* and a *public key*. Alice can use her private key to decrypt data or generate digital signatures on data.\(^4\) (For this reason, we can also refer to a private key as a *decryption key* or a *signing key*). Another user, Bob, uses Alice’s public key to encrypt messages that only she can read, as well as verify digital signatures that she has generated. (We can thus call a public key a

---

\(^3\)Later work on the PorKI project that is not reflected here is discussed by Pala et al. [68].

\(^4\)In many cases, users will have separate keypairs for signing and encryption purposes.
encryption key or a verification key.) As long as Alice’s private key remains private, when Bob successfully validates a signature with Alice’s validation key, he can be sure that Alice really did the signing. If Bob has never seen Alice’s public key before, he can establish her identity via the keypair if she presents him with an identity certificate signed by a party he trusts (this is traditionally a Certification Authority (CA) in Alice’s PKI). This certificate contains Alice’s verification key, as well as some form of identifier, such as her name and email address. Through a carefully designed protocol, Alice proves to Bob that she has the corresponding private signing key; if her signing key has not been compromised, this shows that she is the person identified in the certificate.

Public key cryptography algorithms are designed such that, no matter how many times Alice authenticates to Bob, it would be very difficult for Bob to guess Alice’s signing key. As long as the cryptographic algorithms are sound, this means that even when she authenticates to a malicious party, her private credentials are not exposed.5

A PKI thus allows Alice to authenticate to remote services without compromising her credentials, and allows her to usefully sign and encrypt data, among other things.

However, its effective application in large enterprises requires several preconditions, including:

1. Private keys must remain private.

2. Private keys must be used only for the operations the users intend and authorize.

   (If the user does not have good control over the private key, an adversary can trick him into using it when for things he would not normally want to.)

3. The PKI must integrate with the standard desktop/laptop computing environments users and enterprises already employ.

5In the past, flaws with cryptographic padding functions have caused this assertion to fail. The development of efficient factoring today would affect the integrity of modern cryptosystems, too.
4. Users should be able to use PKI from any machine in the enterprise. Some users, such as system administrators, are responsible for machine maintenance; average average users may have more than one workstation, or one computer for each home and work.

5. The PKI must accommodate the fact that these machines may have varying levels of trustworthiness. (For example, in a university environment these machines may range from dedicated, well-maintained single-user machines all the way to de facto public access workstations.)

   The keyjacking work of Marchesini et al. [55] shows that the first two conditions fail in situations satisfying the third: storing and using private keys on a standard workstation opens then to compromise through the operating system’s credential access API. However, even if an enterprise solves the problem of securing a user’s private keys at one standard machine, the fourth condition requires that user PKI be portable: users should not be limited to using their private key to a single machine, but be allowed to use their credentials in a variety of environments as they desire. (Imagine if we told users who are used to working from home that they could only access their email account from the office.)

3.3 Smart cards and USB dongles to mitigate risk

Portable PKI devices can both protect credentials and allow them to be used on many different machines. They also provide two-factor authentication: users can prove to relying parties that they both have something (the hardware device) and know something (usually the password to unlock the device).

For example, smart cards (also known as chip cards) are thin plastic cards that resemble credit cards, only they have a computer chip embedded in them. Smart cards can be used for a variety of purposes; in Europe, they are the standard format
for debit cards. When used in PKI applications, they can store credentials as well as perform signing and verification operations. (In theory, this should prevent malicious workstations from gaining access to the credentials, although it is still possible to keyjack them in many implementations.)

Similarly, a USB dongle is a small, portable device that interfaces with a computer by plugging it into a USB port, similar to a USB thumb drive. Like smart cards, dongles can be used in PKI applications for storing credentials and perform cryptographic operations using them. The dongles’ larger form factor makes room for more credential storage and computation power, as well as additional functionality. For example, the Sony Puppy has an embedded fingerprint reader, and some manufacturers include the ability to store passwords in addition to PKI credentials.\(^6\)

An enterprise that uses smart cards or dongles for PKI will generate (or have the user generate) a keypair on the device, issue a certificate to the keypair, and put the certificate on the card. The user can use these credentials on any computer equipped with the appropriate reader (for smart cards, a special card reader is required; USB dongles require a USB port). Once the device is plugged into the workstation, the user must unlock it by typing in a password on the workstation. Dongles or card readers that have a built-in biometric reader allow for alternative means of unlocking, which does not require the user to remember a password and decreases the chance of that password being stolen by a workstation that has a keylogger installed.

Having the device be locked by default prevents someone from easily accessing the credentials if it is lost or stolen. Some devices will even lock themselves permanently if a wrong password is entered a certain number of times, which prevents an attacker from using a dictionary attack to gain access to the credentials. These devices are also usually resistant to physical attacks, by which an attacker might try to pry the data

---

\(^6\)This could help users remember secure passwords, and thus decrease the probability that they reuse passwords or choose insecure ones. However, this does not solve the problem of users being unable to recognize when their credentials are being phished.
from the physical memory using specialized tools. (Although, of course, designing a
commodity device whose hardware cannot be cracked by a dedicated adversary is a
very hard problem.)

Dongles are generally more expensive than smart cards, although they are also
more durable. They are easier to deploy for enterprises with a large number of
existing workstations, because they do not require specialized readers like smart cards
do. Both technologies make PKI credentials more portable, although dongles offer
greater security and functionality possibilities because they are not constrained by
the single-chip design and thin form factor of smart cards. For the rest of this section
we concentrate on dongles because their features are a superset of the smart card
features, and are thus the more interesting technology.

3.3.1 Experiences in PKI hardware deployment

Dartmouth College deployed its PKI in 2003. At that time, it started issuing cer-
tificates to users with a Dartmouth email address. These credentials can be used for
authentication to a number of online campus resources, as well as for email signing
and encryption. The College’s CA initially issued certificates online, and users stored
them in their operating system’s software keystore. However, in 2004 the College
started issuing dongles to new students and to the staff of certain departments. Ad-
ministrators decided to use dongles for credential storage and transport both because
they were concerned about the security of private keys on average workstations, and
because much of the campus population (the students) use public terminals as much
as they use their personal computers.

The USB dongles that Dartmouth employs are designed to be used in an enterprise
environment. Their manufacturer provides the College with two software packages,
one for the client side and one for the administrator side. The client software must
be installed on a workstation before the credentials stored on the dongle can be used
there. The token must be initialized with the administrator software (which includes loading it with credentials) before the credentials can be used, too.

In choosing to migrate to tokens, Dartmouth took great care in evaluating the usability of both software packages. Administrators wanted the technology to be both compatible with existing software and user-friendly enough that the average user could easily use their credentials anywhere on campus. While both packages were generally compatible and usable, administrators found that deploying a new dongle requires a complicated ceremony of passwords, ID cards, and mouse clicks for the administrator to identify the user and issue her credentials. This ceremony proved to be a bottleneck during the first year of dongle deployment, and few of the 1,000 freshmen who were supposed to receive the devices during orientation week actually got one. However, during orientation of 2005, administrators managed to overcome the bottleneck and reach about 75% of the students by issuing dongles to students at the same time they received their computers, and by having multiple dongle stations. The ability of the administrators to deploy to such large numbers of students bodes well for eventually requiring all Dartmouth users to have dongles.

Deployment to staff in the administrative departments went more smoothly than with the students. This was in part because it was easy to find the staff and have time to give them the tokens. However, staff seemed to have more trouble with learning how to use their tokens, particularly in choosing and remembering a password. Students are likely more used to signing up for various accounts on the web, which explains why they had less trouble with authenticating to their device using a new password.

### 3.3.2 Advantages and disadvantages

Although deploying these devices proved to be a logistical challenge, their use on campus offers distinct security advantages (and will offer even more when users can only authenticate using them). The alternative to dongle credential issuance (in
which the certificate is stored in a workstation’s software keystore) is done through a web interface to which a user authenticates using his email password. If that password is phished, the phisher could get credentials in the user’s name. Having users authenticate to administrators using a college-issued photo ID makes dongle-stored credentials much less likely to fall into the wrong hands. Furthermore, once a user is required to have a token to authenticate, she will notice and report when she loses it, because she will not be able to access anything she needs. Thus, someone who has the dongle and knows the password to unlock it has a very high probability of actually being the owner of the credentials it contains (although there are ways in which dongle credentials might be keyjacked, as discussed below).

In addition to providing an extra level of assurance over alternative credential storage schemes, USB dongles are much more portable than an operating system keystore or a smart card (the latter because the dongles only need a USB port to interface with the workstation instead of a specialized card reader). However, a workstation must be installed with the client-side software before a dongle can be used on it. This works well in enterprise environments where all machines are under the control of an administrator, but does not easily allow the credentials to be used on a home computer or outside of the normal enterprise network, for example a consultant working at off-site at a client’s location. Furthermore, the software currently available to support the dongles is only available for a limited number of operating systems. A user clearly cannot use her credentials if she cannot install the necessary software, or if the software is not available for the given platform. For a hardware authentication device to be a viable replacement to passwords, the user should be able to safely use it anywhere she wishes to.

Current implementations of dongles are also limited in their ability to replace passwords in that they cannot carry credentials from more than one organization. For example, a user cannot have her dongle loaded with credentials from both her
bank and her web email provider. Users may one day be able to authenticate to all remote resources using a single set of credentials, but in the immediate term we should assume that they may wish to keep sets from multiple issuers on the same device. The device must also then provide a user with a way to identify which set of credentials he would like to use for a given operation.

Credential security is one of the main motivators for considering the use of PKI devices. As we noted previously, the fact that the dongle provides two-factor authentication — requiring something you have and something you know in order to authenticate — increases the security of the system. Also, moving the credentials out of the operating system keystore prevent rogue programs from stealing them or using them improperly. In theory, moving the credentials to the dongle prevents this kind of keyjacking. In reality, because the operating system still provides the dongle with data to be signed or decrypted, a malicious program could still keyjack the credentials by tricking the user into authorizing unsolicited credential operations, which the keyjacking work in [55] shows. This means that relying parties must know that a workstation is totally trustworthy—that it is guaranteed to not have such malicious programs—before they can really trust authentication from a dongle user. Guaranteeing that a workstation has no malicious programs installed is a very hard problem.

This means that the security of current dongle systems rely on users’ ability to recognize a trustworthy machine. Since we know from the success of phishing attacks that users are not good at recognizing the trustworthy websites, it is a fair assumption that users will not always make good decisions when asked about workstations. It would be nice to have a solution that allowed for the portability of credentials, without exposing the credentials to keyjacking by potentially compromised workstations.
3.3.3 SHEMP: Secure Hardware Enhanced MyProxy

In an effort to improve upon the security of existing software and hardware credential stores, Marchesini et al. followed their keyjacking work with a partial solution [54], Secure Hardware Enhanced MyProxy (SHEMP). This system does not make use of a portable PKI device, but it motivates our discussion of the PorKI project below. SHEMP addresses the keyjacking problem by automatically and transparently limiting the damage that a weak client can cause. Instead of distributing long-term credentials for use on workstations, SHEMP keeps them in a central repository and uses them to issue short-term, temporary credentials called Proxy Certificates (PCs) [93]. Workstations have a set of credentials describing characteristics useful in making trust decisions—such as where the machine is located and how many people have access to it—and users have Key Usage Policies (KUPs) describing how their credentials should be used on workstations with certain characteristics. The SHEMP server limits the powers of PCs based on the KUP, depending on the characteristics of the workstation from which the user is working.

This means that the workstation never has access to the long-term credentials; only the temporary credentials can be keyjacked, which presents a compromise of limited duration. Because SHEMP allows users to automatically limit the capabilities of the temporary credentials based on the workstation environment, the system does not depend so much on users’ ability to recognize trustworthy machines. Limiting the capabilities of credentials on less trustworthy machines also mitigates the potential damage from keyjacking on those machines.

However, SHEMP requires the deployment and scalability overhead of this centralized repository. It also provides no effective way for a user to authenticate to the repository via a potentially untrustworthy client: in the SHEMP prototype, authentication is by password, which could be keylogged if the workstation is compromised. It would be nice if users could authenticate to the repository without going through the
workstation. Inspired by this idea and the portability of USB dongles, we designed PorKI as a portable implementation of many SHEMP concepts.

### 3.4 PorKI design and use

The goal of the PorKI system is to present users with a usable and secure way to store and use their private keys, without requiring them to purchase special devices, but while enabling them to make safe trust decisions in heterogeneous environments. PorKI accomplishes this by building on the SHEMP framework, but with some additions and modifications.

PorKI is essentially a software application for a Bluetooth-enabled PDA (personal digital assistant, or hand-held computer) that manages long-term credentials. This makes the repository portable, instead of centralized as in SHEMP. General-purpose PDAs are a departure from the specialized smart card and dongle devices that we considered above. However, as is indicated below, the additional computation power and display capabilities allow a more rigorous solution to the problem of portable PKI.

There may also be a workstation software component in the PorKI system, which offers information about the workstation to the PDA before credential transfer, and which can help manage the credentials when they are on the workstation. However, under the current design any workstation that accepts the Bluetooth file transfer protocol can interface with a PorKI-enabled PDA, as we indicated below. Any server that supports X.509 certificate-based authentication and accepts proxy certificates can interpret PorKI-generated credentials, although advanced functionality (parsing KUPs or workstation credentials from the proxy certificates) may require modifications to the server software, again discussed below.

Because the PorKI credential repository is kept on the PDA, the user can authen-
ticate to it directly instead of going through a potentially untrusted workstation, as in SHEMP. The PorKI program on the PDA then transfers a set of temporary credentials (again a temporary keypair and a proxy certificate) to the workstation over Bluetooth. As with SHEMP, if the workstation has itself a set of credentials attesting to certain characteristics, PorKI can limit the privileges of the temporary credentials according to a policy defined in terms of these characteristics. Instead of just allowing users to define a KUP, however, PorKI can also pass the workstation characteristics on to relying parties, who can also choose to limit the privileges of the user while she is authenticated from that machine. Often, administrators of the servers to which users are authenticating to are more qualified than the users themselves to define a usage policy, so it makes sense to offload the decisions of whether or not to trust a workstation to them.

We have implemented a basic prototype of the PorKI system, although it does not yet include the complete set of the features specified by our design. To better illustrate how PorKI might eventually be used, we consider two example cases of authentication using it according to its current design. In Section 3.5, we elaborate on the challenges of bringing PorKI to actual deployment in the real world.

### 3.4.1 Example 1: Bob and an untrusted machine

Bob is an XYZ Bank customer, and his bank has issued him an X.509 certificate he can use to authenticate to their online banking site. He is visiting a friend and wants to check his balance before going out to dinner. He sits down at his friend’s computer, takes out his PDA, and launches the PorKI application on it. He authenticates to PorKI by inputting a password into the PDA, which unlocks the credential repository. He makes sure that the Bluetooth device on the computer is set to be discoverable by other Bluetooth devices, and tells PorKI to search over Bluetooth for available workstations. When PorKI displays the list of available computers, Bob selects the
appropriate one, selects his XYZ Bank credentials from the list available in the PorKI repository, and chooses to “Generate Credentials”.

Because the computer does not have its own set of credentials, PorKI considers it to be the default “untrusted”. PorKI generates a new keypair and a proxy certificate (the latter of which specifies the value “untrusted”), and transfers these credentials to the workstation using the Bluetooth file transfer protocol. Because the workstation might have a malicious program that looks for unprotected credential files and tries to exploit them, PorKI protects the credentials with a one-time password before transferring them, and displays that password on the PDA screen once the credentials are transferred. On the computer, Bob imports the credentials into the operating system’s keystore by double clicking on the transferred file and entering the password that PorKI has chosen. He turns off his PDA, opens a web browser on the computer, and goes to XYZ Bank’s website. When he presses the “authenticate” button, the web browser detects his credentials in the operating system keystore and uses them to authenticate to the PKI-enabled web server.

Because Bob’s temporary credentials say he is authenticating from an untrusted workstation, XYZ Bank lets his view his balance, but does not display other personal information or allow him to transfer money out of his accounts. (If he were authenticating from a more trusted workstation, the bank site would likely give his more privileges.)

When Bob is done, he logs out of the website. He can choose to remove his temporary credentials from the computer, but they have a short default lifetime and he can be reasonably sure that no one else will try to use them. Even if someone else did use the computer and gain access to his credentials, the worst that the attacker could do would be view Bob’s bank balance, because the website compensates for the insecurity of the workstation by limiting the credentials’ privileges.

Because XYZ Bank uses PKI for authentication, its users’ credentials cannot be
phished. Because Bob has PorKI installed on his PDA, he can carry his credentials with his and use them on any Bluetooth-enabled computer without installing special software first.

3.4.2 Example 2: Alice and an enterprise workstation

Alice is an IT professional at a university who is installing new software on a workstation. The workstation already has a set of credentials issued to it, which identify it as being in one of the campus libraries and open to use by any campus user. It is also installed with a lightweight software program to interact with PorKI at a more advanced level than in the previous example.

Alice launches the PorKI software on the workstation, turns on her PDA, and instructs PorKI to search for available PorKI-enabled workstations. Instead of establishing a file-transfer connection with the workstation, PorKI instead establishes a PorKI-specific connection over Bluetooth.

The workstation passes its credentials (a certificate attesting that the workstation has certain characteristics, and proof that the workstation is in possession of the signing key that corresponds to that certificate’s verification key) to Alice’s PDA, and the PorKI application compares the characteristics they detail to the personal KUP that Alice has defined. She has specified that temporary credentials issued to workstations in public places should have a shorter lifespan than default, so this information, along with the workstation credentials, are put into the proxy certificate that PorKI generates and passes to the workstation along with the new keypair. PorKI again protects the temporary credentials with a one-time password, which it displays on the PDA for Alice to enter on the workstation when importing the credentials there.

When Alice goes to connect to the university’s software server, the web application interacts with the workstation PorKI software to perform the authentication. Because
Alice is authenticating from an enterprise workstation, but that workstation is public, the server might choose to allow her to download software, but limit the number of applications she can download. If she were trying to authenticate from an untrusted workstation, the server might let her view the applications she could download, but would likely not let her actually obtain copies. In this case, the administrators of the server would hopefully provide an informative message letting her know that the reason she could not download software was because of the characteristics of the workstation from which she was authenticating. Informing users in this way would help them figure out how to accomplish the tasks they want to.

3.5 PorKI evaluation

A key repository and delegation system like PorKI could better enable the deployment of PKI as a password replacement, because it improves the usability and portability of traditional PKI systems. Using PKI credentials offers significant security advantages over using passwords; the PorKI system also allows for additional information to be inserted into trust decisions programatically, thus reducing reliance on users’ judgment. However, as we saw with the PKI token deployment, even a well-designed system meets hurdles when implemented in the real world. In this section, we evaluate the design and the preliminary prototype in terms of issues that will be relevant in a real-world deployment.

3.5.1 Keypair generation

Generating a long-term keypair for a PKI device could take place in at least three different locations: the device itself, the workstation, or the certification authority’s server. The first option requires a mechanism to remotely certify the keypair; the second two options require a secure protocol to transfer the credentials. Section 3.6
elaborates on ways to enhance PorKI to better enable communication between the
PDA and remote parties.

Generating the temporary keypair could also be accomplished on either the PDA
or the workstation. The credential transfer protocol under which it is generated on
the PDA is significantly simpler (and enables PorKI to use the Bluetooth file transfer
protocol if it is interacting with an untrusted workstation not installed with PorKI
workstation software).

To determine whether it is feasible to expect a PDA to handle keypair generation
in a reasonable amount of time, we measured how long it takes to generate the
temporary credentials on preliminary PorKI prototype, which was implemented on
a notebook computer with a 1.33 GHz PowerPC processor. We found that it took
1.67 seconds to perform both the key and proxy certificate generation and signing,
with slightly over half that time being spent just on the key generation. If we assume
all other things being equal, transferring the PorKI software directly to a PDA with
a 200 MHz processor\(^7\) would yield credential generation time of roughly 11 seconds,
about half of which is dedicated to the keypair.

This amount of time is significant, and would likely frustrate users if they were
trying to authenticate quickly. Moreover, the battery life of portable devices becomes
an issue when performing compute-intensive tasks. For these reasons, we explore
ideas of offloading key generation to the workstation in 3.6. Perhaps one day hardware
improvements will not make this necessary. Boneh et al. [6] cited 1024-bit RSA key
generation (the same task being performed in PorKI) as taking 15 minutes on a
PDA in 2000; this improvement in five years indicates that the key generation lag
may evaporate in the near future, although the future of mobile battery lives is not
certain.

\(^7\)This was the speed of a lower-end Bluetooth-enabled business model at the time of this research.
3.5.2 Client and server integration

Integrating PorKI into existing client applications would likely prove to be a difficult problem. Authentication through web browsers, which have well-defined APIs for accessing the operating system’s keystore, is easier, so a system like PorKI would be more easily deployable for web-based resources. Client applications, into which password authentication is heavily embedded, would need to change to accept PKI authentication from credentials in the OS keystore (or user-defined keystores) before PorKI could be used with them.

In order for a remote server to accept authentication using PorKI temporary credentials, that server needs to be configured for general PKI authentication, as well as be able to parse extensions specific to proxy certificates and treat them accordingly during certificate validation. The proxy certificate IETF standard was developed to better enable distributed computing in grid environments. Because these certificates are of a standardized format already in use, it seems likely that commercial-off-the-shelf (COTS) software developers would be willing to integrate the necessary parsing into their products. However, this integration would likely take time.

Parsing PorKI-specific data in the proxy certificates, such as information on the workstation characteristics, would have to be explicitly implemented in the server software in order for the server to limit privileges accordingly. Many web servers provide APIs with which developers can create modules that manage access permissions; this indicates that we could develop a plug-in to allow major web browsers to understand the PorKI-specific data.

3.5.3 Repository protection

The major advantage of PKI-specific devices is that they are separate from users’ machines, and thus not as likely to be vulnerable to the same attacks just because they are not connected to the network. If a system like PorKI were to be deployed
on a PDA with wireless capabilities, it would become more open to attack, but there seems to be little malicious focus on PDAs in comparison to user machines. This will surely change if the number of people using PDAs grows.\footnote{With the advent and proliferation of smartphones, malicious software for mobile devices has certainly become more common than it was when this was first written. Time will tell whether mobile operating systems prove more resilient to malware than some desktop operating systems have thus far been.} However, because PDAs are frequently synchronized with a workstation in order to back up the data they contain, we could eventually envision having the workstation evaluate the PDA for changes between synchronization, thus reducing the risk of malicious programs that trick users into using their credentials for unintended operations.

In the case of device loss or theft, USB tokens and smart cards are designed to be resilient against both software and hardware attacks: they are protected by passwords (and USB tokens lock after too many failed password entries); they are designed to break and lose data if tampered with. The PorKI repository is also protected by a password; this prevents the possessor of the PDA from automatically having access to the credentials, provided that the password is non-trivial and not shared. However, entering a password into a PDA is not a very quick process, either by stylus or miniature keyboard. In the future it would be advantageous to consider alternatives that are more usable, such as a series of gestures with a stylus or other forms of “graphic passwords” [90] or a biometric password, if the PDA was equipped with a fingerprint-reading device (such as the Hewlett-Packard h5500 handheld is).

It is possible that mechanisms in the PDA’s operating system could be used to better protect the repository, although any solution implemented in pure software runs the risk of being compromised by some sort of rootkit. The only way to truly provide information assurance on a PDA would be for it to have a secure hardware component. No such device is currently available, although the U.S. Government has contracted to have one built [88]. Such a solution would offer much stronger assurance against advanced hardware attacks if the device were stolen, and would be
recommended for any truly high-security situations.

### 3.5.4 Bluetooth transfer

The Bluetooth protocol is a widely deployed and easy-to-implement for portable devices. However, despite these benefits, there is considerable concern regarding Bluetooth’s security; the pairing process and its use of PINs has come under particular scrutiny recently, with Shaked and Wool [80] describing a passive attack that cracks 4-digit PINs used in pairing in 0.06 seconds on a Pentium IV.) The current PorKI design does not integrate additional security measures for the credential transfer. However, this is one of the top priorities for future work, as is discussed in Section 3.6.

### 3.5.5 Usability

One of the most important design goals of PorKI is to make a system that is truly usable. On our notebook prototype, we did some rough measurements to estimate how many times a user would have to interact with the system in order to generate temporary credentials, transfer them, and import them into the workstation’s operating system keystore. The prototype required a total of four clicks and a password on the PDA end. The number of clicks required to import the credentials on the workstation took varied by operating system: Windows took eight and a password, where OS X took four and a password. Once the system is implemented on the PDA, the number of clicks could be reduced there; having PorKI client software on the workstation could similarly reduce the amount of interaction necessary.

This interface is not as seamless as that of a USB token or a smart card systems, which usually just require a few clicks and a password on the workstation. However, PorKI offers the added functionality of being able to store keys and certificates from many different sources, in addition to the added benefits associated with policy state-
ments about the workstations. Thus, we feel that any additional interaction is worth the increased functionality.

It is also important to consider overall usability and applicability of the PorKI model to the average user. While PorKI (like dongles and smart cards) seems well-suited to enterprise environments, it is also important that it be useful to individual users who need to authenticate to remote web resources. These users do not have a system administrator to install software for them, and they are less likely to have a machine that can securely keep a set of credentials. These users are the ones most susceptible to the widest-spread phishing attacks; can PorKI work as a password replacement well enough to protect them from these attacks?

When you consider the ease with which the home user uses passwords and that he usually does not check his email or bank statements anywhere other than his home computer, it would seem that PorKI is not an ideal scheme for him. The number of steps necessary generate and transfer temporary credentials would likely frustrate him when required for each individual authentication. Moreover, it would be unnecessary if his computer is properly protected: given a well-designed operating system that keeps his credentials in a truly secure keystore, with a trusted path to that keystore so he cannot be tricked into authorizing unwanted credential operations, there is no reason to try and move his credentials to a different device.

However, PKI is likely to be adopted for authentication before average user computers become this secure. When phishers can no longer target password credentials, they may turn to operating system exploits to get software-stored PKI credentials. PorKI could allow the user to store multiple credentials securely.

If we can develop an implementation of PorKI that is small enough to fit on a cell phone\(^9\) — much more common devices than PDAs — this solution would become even more applicable to the average user. Other expansions to the PorKI project,

\(^9\)Our lab later demonstrated the feasibility of adapting PorKI to smartphones by implementing it on an iPhone. [68].
particularly the stateful browsing and trusted path ideas discussed in Section 3.6, could also be useful in adapting PorKI for use in the home environment. Although the forms of protection they would provide are different, they could mitigate the risk of using private credentials from a poorly-protected workstation to authenticate to remote resources.

### 3.6 New applications and directions

In this section, we consider some areas of future work for PorKI.

#### 3.6.1 Offloading keypair generation

As we noted above, offloading the temporary keypair generation to the workstation could improve the usability of PorKI by reducing lag time and battery usage. Work by Boneh et al. [6] includes a scheme for generating keypairs on PDAs with the help of an untrusted workstation; luckily, because PorKI allows the workstation full access to the temporary key without risk, the entire task of key generation could take place on the workstation. In order for PorKI on the PDA to issue the proxy certificate using the long-term credentials, the public part of the temporary keypair needs to be transferred to the PDA, and the user needs to verify that the public key received is the same one the workstation transmitted (to prevent a man-in-the-middle attack). This would impact the usability of the system as a whole, as it would increase the amount of user interaction.

#### 3.6.2 Trusted path and stateful browsing

Programs such as SSH provide security in part by *remembering* the identity of the servers to which they have connected. (Identity is based on a server’s name as typed on the command line, paired with its public key.) The user is notified whenever she
is about to make a connection to a site that the program does not know, and asked if she wants to continue. As this message is rarely seen—only once for each server, unless the server is re-keyed or a man-in-the-middle attack is being enacted—it is a useful way to provide users with more information with which to make trust decisions.

If there were some way for PorKI to communicate directly with the remote party via a trusted path, PorKI could emulate SSH. Perhaps the first work to propose, prototype and evaluate trusted paths for web browsers was [100] by Ye and Smith; recent work described in [18] builds upon the principles presented therein. Previous work on trusted paths specific to portable devices includes [67]. When a user connects to what she thinks is her PayPal account, if PorKI notifies her that she is, in fact, connecting to a new server that has never before been seen, this could help her identify a phishing site. Garfinkel and Miller refer to the SSH metric for evaluating the security of a connection as *key continuity* [31]; in this application, we might refer to *stateful browsing*, which could grow to encompass not only useful security information, but also things like bookmarks or browser preferences. Efforts to put an entire web browser on a USB thumb drive include Portable Firefox [21]. Such an approach could limit the risk of a malicious program modifying the browser by ensuring that only parts of the filesystem (for example, the bookmarks) are modifiable. A trusted path between the PDA and the remote party could also eventually enable the use of the long-term keypair for highly sensitive applications, as well as digital signing and decryption, with the potentially untrusted workstation acting only as a transfer agent.

### 3.6.3 Wireless certificate issuance

In a PKI, a user’s certificate for their long-term keypair is issued by the PKI’s CA. With PorKI, once the certificate is issued it must join the keypair in the PDA repository to complete the long-term credentials. The current PorKI uses an intermediary
workstation to transfer the credentials, but it would be much cleaner to have them be transmitted in some streamlined fashion directly from the CA. Dongle deployment practices often require users to receive their tokens in person, so this would not be more of a burden than current technology. The user could interact with the PDA and the administrator with the server, and share data over Bluetooth, instead of trying to share a single interface (as with dongle deployment). It might even be possible for an administrator to issue certificates to a room full of people at once, given the proper protocol of real-word identification and measures to ensure that no one outside the room could receive a certificate.

3.6.4 Human-to-human delegation

Another application that could harness PorKI’s wireless interface as well as its capacity for issuing proxy certificates is the delegation of a user’s credentials. For example, a user might choose to delegate the right to access an online resource—e.g., a payroll server—to another user temporarily, while she’s on vacation. If both users are equipped with a PorKI-enabled PDA, this could be accomplished directly over a Bluetooth connection. The temporary credential could also be issued directly to a workstation as in the original PorKI design.

3.7 Conclusions

Phishing exploits the mismatch between human perception of a system and what is really happening under the interface; humans rely heavily on simple visual clues to judge trust, and are under-qualified to evaluate the trustworthiness of programs and machines they authenticate to or through. Using hardware devices, such as USB PKI dongles or PKI-enabled PDAs, affords a level of protection to users’ private credentials that the users themselves cannot. Expanded uses of these technologies,
including programmatic workstation and website evaluation, stateful browsing, and person-to-person credential delegation, enable users to re-use their intuitive notions of trust while making decisions and communicating in the digital world. Enabling our security systems to follow human trust processes reduces the mismatch between what a user expects and what the system is really doing—and with that mismatch also reduces the opportunity for phishing to exploit it.
Chapter 4

Qualitative methods and corporate study design

The body of work I describe as the Corporate Study started out as an effort to get inspiration for research like that presented in the previous chapter. As noted in Chapter 1, though, it quickly became apparent that the problems participants were having with access-control technology were far greater than we had anticipated. The engagement with participants thus morphed from a case study of a single financial institution into two\(^1\) full-on qualitative research efforts exploring the factors that drive and constrain the practice of access control in large and complex human organizations.

In this chapter I describe the qualitative research methods I employed in conducting these two studies, and present an introduction to the Corporate Study, whose results I describe in Chapters 5 and 6.

4.1 Qualitative methods

As described in Chapter 2, researchers studying human-computer interaction (and HCI in the context of information systems) have made significant use of qualitative

\(^1\)The Corporate Study and also the Clinical Study, which is described in Chapter 7.
methods to understand the human environments in which a computer system is to be deployed (e.g., to gather requirements for the design of that system), and to evaluate how well a computer system is meeting the needs of the humans who use it. In this section I detail the research goals I wished to attain in employing qualitative methods, and offer details on the specific methods I used in the course of the Corporate and Clinical studies.

4.1.1 Objectives in qualitative research

The goal of qualitative research is similar to that of quantitative research: to gather information that offers insight into the nature of a problem or phenomenon. Qualitative research methods achieve this goal by observing the qualities of (and interactions between) the factors and processes that contribute to the phenomenon, rather than by measuring their magnitude, frequency of occurrence, or other quantifiable attributes.

When embarking on this research I did not have a particularly well-defined set of questions I wanted to answer; I instead had a problem — the apparent brokenness of access control as it is practiced in real-world organizations — that I wanted to understand better. I chose to employ qualitative methods that seemed to offer a good path toward better understanding nature of this problem.

4.1.2 Semi-structured interviews

Both the Corporate and Clinical studies feature semi-structured interviews of individuals from participating organizations. This type of open-ended interview is designed to facilitate exploration of one or more themes. It starts with a series of questions, but emphasizes flexibility to enable researchers to ask unscripted follow-up questions, thus allowing participants and researchers to shape the discussion in whatever way seems most natural or interesting. This is in contrast with a structured interview, which relies on a pre-defined set of questions to elicit information.
Semi-structured interviews work best when the interviewer has identified a broad set of themes they wish to ask questions about; for this research a number of central themes quickly developed:

1. The tools participant organizations use to manage access control
2. How access-control tools come to be developed or purchased, configured, and deployed within the participant organization
3. Participants’ broad experiences managing and using access-control technologies
4. Anecdotes where participants were surprised or frustrated in their interaction with access-control technology — either by the design of the tool itself, or by unexpected interaction patterns exhibited by users
5. Participants’ vision of what it would mean for an access-control technology to be successful within their organization, and whether their current tools meet that standard
6. How participants perceived the community of access-control researchers, and the body of research literature on the subject

The open-ended nature of semi-structured interviews makes them well-suited to the discovery and exploration of an issue or phenomenon. When conducted with large participant populations, they can offer a relatively complete picture of participants’ experiences. In this research I sought to gather information that was both detailed and accurate. While I believe that the results I have highlighted apply to many organizations similar to those that participated in these studies — and practitioners from multiple non-participant organizations have informally affirmed this opinion — we should not take these results as being a comprehensive catalogue of the experiences such organizations face.
Like any data collection method that relies on self-reporting, we must also anticipate that the results revealed by semi-structured interviews may have been influenced by the individuals doing the reporting; I expect that it may even be possible to identify systematic biases in participant reporting. However, by seeking qualitative rather than quantitative data from participants — in seeking to document the existence of phenomena rather than to measure their frequency or prevalence — I believe that the potential for such biases does not reduce the importance of these results.

I used transcripts of recordings from interviews with software-industry participants to study their perspectives, but relied on my own and others’ field notes while analyzing the perspectives of other interviewees.

### 4.1.3 Ethnographic observation

In an effort to bypass some of the limitations of participant self-reporting one encounters while conducting interviews, I also engaged in a small amount of direct ethnographic observation of participants, particularly during the Clinical Study. Ethnography in general seeks to describe the nature of the participants, their interactions, and their environment by becoming an accepted (if temporary, and often relatively quiet) member of the community being studied. In both studies this included sitting in on regularly-scheduled meetings as a silent observer; in the Clinical Study, it also meant spending time at nurses’ stations watching them go about their jobs as normal.

The product of my ethnographic observations were field notes, which I then classified and analyzed using standard methods for qualitative data analysis, as described below.

### 4.1.4 Qualitative data analysis

A qualitative researcher who has gathered transcripts and field notes seeks to bring order to (and extract meaning from) her data using one or more qualitative coding
methods.

Data coding

The process of coding qualitative data consists of choosing words or phrases to represent chunks of source text; depending on the goals of the study and the nature of the data, researchers may choose from among a variety of coding schemes; Johnny Saldaña’s Coding Manual for Qualitative Researchers [74] offers a nice overview of major ones, and is the source of quotes below.

In this work I applied a mix of two coding methodologies to my raw data (my transcripts and field notes):

- **Descriptive coding**, which “summarizes in a word or a phrase — often a noun — the basic topic of a passage of qualitative data”. Saldaña notes that the topic of a passage is not the same as its content; for example, it is possible for two participants to discuss role-based access control but say very different things about it.

- **In vivo coding**, which “uses words or short phrases from the actual language found in the qualitative data record” to encapsulate meaning.

My process for applying codes to data thus consisted of breaking the data into logical passages, and identifying a phrase (either an in vivo quote or descriptive words of my own) that summarize each passage. To help me in my task of indexing my data, I often parameterized codes with additional information; I would thus tag a passage not just as pertaining to “provisioning” or “org structure”, but “provisioning: mechanisms” and “org structure: awareness of”. Figure 4.1 offers a small sample of the codes derived from Corporate Study data, to give a flavor of what they look like.
<table>
<thead>
<tr>
<th>business relations: buy-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>business relations: “language of the firm”</td>
</tr>
<tr>
<td>business relations: tech-savvy person to interface with</td>
</tr>
<tr>
<td>data ownership awareness</td>
</tr>
<tr>
<td>dogfood: internal product deployment as sales tool</td>
</tr>
<tr>
<td>entitlement granularity</td>
</tr>
<tr>
<td>entitlement review: drop-dead date</td>
</tr>
<tr>
<td>HIPAA</td>
</tr>
<tr>
<td>IT support: escalation</td>
</tr>
<tr>
<td>org maturity: inflated sense of</td>
</tr>
<tr>
<td>org scale: # employees</td>
</tr>
<tr>
<td>policy changes: testing</td>
</tr>
<tr>
<td>provisioning: copy/paste and adjust</td>
</tr>
<tr>
<td>provisioning: deprovisioning during transfer</td>
</tr>
<tr>
<td>provisioning: deprovisioning process</td>
</tr>
<tr>
<td>risk management: philosophy</td>
</tr>
<tr>
<td>role definition: “if it starts at IT it’s going to fail”</td>
</tr>
<tr>
<td>role definition: dependence on local laws</td>
</tr>
</tbody>
</table>

**Figure 4.1:** A small sample of the codes I used to organize raw qualitative data (field notes and interview transcripts) from the Corporate Study.

**Developing themes**

I then used these codes to develop a set of themes, using the following process:

1. Skim through the list of codes

2. Identify a theme or unifying idea that seems to connect a group of codes

3. Read carefully through the codes (referring if necessary to the sections of the transcripts and field notes to which they refer) to capture all the codes that are associated with the theme

4. If possible, identify connections between the current theme and others that have been previously identified

5. Iterate until all significant themes have been extracted

6. Review the resultant set of themes and their connections, with an eye toward splitting themes where necessary and combining them where possible
The set of themes that resulted from this process form the backbone around which I build my presentation of the study results in the next few chapters. While the experience of doing this analysis was somewhat laborious, it provided me the opportunity to examine my observations from a wide variety of angles, and ultimately has offered me a deeper understanding of the information that participants shared in taking part of these studies.

4.2 Corporate study design

The fieldwork I refer to as the Corporate Study\(^2\) consisted largely of semi-structured interviews of employees from three large companies (two investment banks and one software vendor). This section offers more an overview of the study: its focus, its structure, and the organizations who participated in it. This information sets the stage for the study results presented in Chapters 5 and 6.

4.2.1 Preliminary work

At the outset of the work that eventually became the Corporate Study, our research group was inspired by anecdotes from professionals in a variety of industries who reported users circumventing security controls in order to get their jobs done. As a computer security researcher, these accounts were surprising and troubling: were users really doing this? What types of controls were they circumventing, and why were not organizations trying to stop the circumvention? Was there something we, as researchers, could do to improve the security systems — perhaps make them more flexible, so users did not have to work around them?

I sought to interview information security practitioners to verify these reports of user circumvention, and identify any related phenomena. I anticipated my research

\(^2\)Here I use “corporate” to describe office-and-cubicle environments, which are in contrast to the hospitals and exam rooms of the Clinical Study.
path would consist of three broad phases: 1) Observe practitioners in the real world and document the challenges they encounter. 2) Build a new tool or solution (such as those discussed in Chapter 3) that addresses one or more of these challenges. 3) Take my tool back to practitioners and try to evaluate whether it would actually be useful to them.

**Focus and scope**

We first pursued collaboration with a high-level information security professional at a large investment firm. Preliminary discussions indicated that the firm did indeed struggle with providing access control that met the needs of their users, so we focused our interviews on identity and access management (IAM) systems.

In an effort to gain broader perspective on these problems, we also initiated collaboration with another investment bank of similar size to the first one. Although the scope of this second collaboration is considerably smaller than the first, its results do serve to validate that our findings are not unique to a single organization.

After collaborating with these financial organizations, we also had the opportunity to interview infosec practitioners at a large software company. This organization not only develops products in the IAM space, but uses its own products internally, and thus offered a new double perspective on access-control tools.

Section 4.2.2 offers more details on the individual participant organizations and our fieldwork with them.

**Evolution**

It quickly became apparent that participants were not concerned with — or even really interested in — many of the the issues academic researchers were focused on in this space. We entered the fieldwork expecting to offer novel insights and helpful solutions; instead, practitioners gave us lessons on the realities of their world. Academics were
looking at all the wrong problems. The basic forms of many of our pet technologies (e.g., end-user PKI) were neither used nor considered useful in practice. Our set of technical vocabulary was disjoint from their own. The formal literature of what we considered to be well-established access-control models (e.g., RBAC) was generally useless in building actual systems.

Because I started these interviews looking for problems I could solve with technology, I was initially surprised at how much participants talked about human and business factors in describing their challenges with identity and access management. While I expected that the usability of security tools would play a role in the organizations’ problems, I did not anticipate that the relevant human factors would extend so far from the individual user’s computer screen.

Given these early surprises, I quickly changed my approach from one of finding a solvable problem to one of documenting participants environment and experiential wisdom. I still wanted to build technology to help practitioners with identity and access management, but clearly needed to re-learn what IAM was all about first.

4.2.2 Participant organizations

We collaborated with three organizations during the course of this study: two investment banks and one software company. We made contact contact with key “gatekeeper” individuals at each organization through the professional networks of our PIs. These gatekeepers then helped us establish the collaborative relationship and get access to interview their colleagues.

Primary investment bank (P-Bank)

I refer to the investment bank that played the larger role in this study as the primary investment bank, or P-Bank. This bank is a large (approximately 30,000 employees worldwide) investment firm that provides a variety of financial services to organiza-
In 2006 an M.B.A. student and I conducted a two-week field study at the P-Bank headquarters. This included in-depth semi-structured interviews of 18 employees, including one Managing Director (a C-level executive), two Vice Presidents, and a variety of associate- and analyst-level participants. The minimum length of each interview was an hour, but we conducted multiple interviews with approximately a third of participants, including daily wrap-up interviews with our VP-level primary contact. Two participants occupied non-technical positions within the organization; the remainder were employed within various technical branches, including software development, general IT support, information security, and infosec policy definition. I base my analysis on the field notes of that the M.B.A. student and I each took during these interviews.

Employee participation was limited by professional availability during the site visit (e.g., employee vacations and meeting conflicts), and was arranged in all cases by our primary contact. Initial participants were selected by our primary contact based on pre-visit discussions of the study’s goals. Additional participants (most of whom were interviewed during the second week) were selected either by explicit referral from initial participants (e.g., “You should talk to Henry R., he is deeply involved in that process”) or by our primary contact in response to specific researcher requests (e.g., "Could we talk to a manager on the business side who had to perform an entitlement review?"). Researchers took field notes while interviewing participants, but the the interviews were not recorded using audiovisual equipment.

**Secondary investment bank (S-Bank)**

The second participant investment bank (which we refer to as S-Bank) is also a large (approximately 50,000 employees worldwide) investment firm that provides a variety of financial services to organizations and individuals.
We interviewed two S-Bank employees, and each interview lasted roughly an hour. We conducted the interviews during one of the weeks we spent doing deeper interviews at P-Bank. I base my analysis on the field notes that the M.B.A. student and I each took during these interviews.

The interviews did not go into as much depth as those at P-Bank, but we had the opportunity to both solicit their own perspectives on the challenges they faced and check that their experiences mirrored those reported by participants from P-Bank.

**SoftCorp.**

I conducted interviews at the participant software company (which we refer to as SoftCorp) in collaboration with a team of three researchers from another academic institution in 2009. I base my analysis on the transcripts of those interviews produced by my collaborators.

We interviewed a total of 5 SoftCorp employees. Participants reported that the organization had an employee population of approximately 14,000, give or take 5-10% for short-term contractors.

One participant was involved in the development of the identity management product that was the focal subject of the interviews. The other participants were part of the internal IT organization responsible for the operation and management of the internal deployment of that product.

### 4.2.3 Background observations

Chapters 5 and 6 reference a variety of systems and problems that came up during interviews. This section offers a brief overview of those topics, and serves to provide additional context that is useful in understanding observations and participants’ comments in the later chapters.
Organizational and interpersonal notes

As mentioned above, participants spent a considerable amount of time discussing the human externalities that impact the success of identity and access management systems. Participants brought up phenomena that are rooted both within individual humans (e.g., emotional state, perception of the IT department’s goals) and without (e.g., the organization’s philosophy and culture).

The organization. Participants often provided us with background information about the organization while discussing particular IAM topics; participants from P-Bank in particular had much to say about the driven, no-nonsense culture of their firm. All three organizations seemed to prioritize information security — but this is likely part of the reason they were willing to partner with us in this research. P-Bank participants espoused the belief that “cost is not the issue” when it comes to providing information security.

“The business would love to give [the infosec] group three times the funding if they could solve security issues completely transparently, without changing everything.”

The infosec team. Most interviews reflected a certain degree of cohesion among the organization’s infosec team, even when viewed from the outside by participants who were not themselves infosec professionals. These teams varied in size from approximately thirty people at SoftCorp to ninety-eight people at P-Bank. The high-level executive who headed the P-Bank infosec team said that the size of his organization was strategic. If he had a group larger than one hundred people, other groups might get the impression that he was there to do the heavy lifting; they might tell him that “you should be helping solve all my risk”. With a small team, he was able to force different units within the business to take ownership of their own security
issues instead of pushing it off on him:

“... other people can’t turn around and say, but you’ve got this huge team, you do it!”

The executives of P-Bank had also recently removed this participant from the technology management team in an effort to force other managers to engage in security management within their respective organizations.

“... the senior tech managers, any time security decisions came up, they turned to me. Once I left, the CIO grilled each of them on their security stuff, then they went back and called me to ask me and my team to help them.”

Infosec team relationships. Participants also spent significant time discussing the relationship of the infosec team to other players, including users, vendors, and regulatory compliance staff. The results chapters discuss some of the relevant aspects of interaction with users and vendors; it is also worth noting that participants from P-Bank had a particularly dim opinion of the competence and efficacy of regulatory enforcement. During an informal meeting early in the research collaboration, one participant told me a story about a compliance audit, and how easy it had been to sweep what he considered to be serious problems under the rug. At the end of the exercise — when he realized how little the auditors understood about the systems and the threats they faced — he wanted to tell them, “Hey, when you go to [the retail bank he used for his personal finances], you should ask them about these other things, too!”
Chapter 5

Corporate study results: the organization against itself

Popular understanding of information security holds that it aims to protect computer systems from the “bad guys,” be they external attackers or malicious users working from the inside. At the outset of this study I expected to observe a degree of opposition between the infosec staffers and the business users — I expected that on some level, in the effort to protect organization, its members would be working against each other. While there is some evidence of this dynamic — particularly in the interviews with lower-level infosec professionals — higher-ranking interviewees uniformly focused on cooperation and support of their organization’s business users and processes. As one information executive said of the challenges his group faced, “It’s not how to get stuff done, it’s how to not mess stuff up”. Rather than worrying about malicious users, interviewees said they “assume[d] people will do the right thing,” and strived to “make sure [the security systems] won’t annoy them in the process”. The goal of the team was thus to facilitate the organization in its larger efforts; in some ways, they viewed themselves as “stewards of the company”.

This study supports the belief that a deep understanding of business processes
helps IT groups achieve their infosec goals. However, in documenting the successes
of this cooperative philosophy, it also highlights ways in which well-intentioned but
insufficiently-informed infosec practices can put the organization at tremendous risk.
In this chapter I discuss specific examples: Section 5.1 considers instances in which
users have circumvented security controls in order to get their jobs done, and Sec-
section 5.2 explores why access-control settings consistently defy organizational policy
and tend over time toward a “system high” configuration. While troubling on their
own, these phenomena indicate that real-world organizations may suffer from a deep
mismatch between the official *de jure* and practical *de facto* policies, and thus a pro-
found disconnect between the the perception and reality of the organization's state
and security posture.

### 5.1 Users circumvent controls

Participants frequently mentioned the use of *controls* — mechanisms to constrain or
regulate users’ actions — to achieve the organization’s infosec goals. Some controls
are built into computer systems, including the set of technologies we refer to as “access
controls” and the separation of duty (SOD) constraint described by one participant
below. Organizations also implement “manual-type business controls,” which shape
user behavior by dictating processes to be followed; SoftCorp falls back on manual
controls when there are too few staff members within a department to provide the
preferred automatic SOD control.

“[The] treasury application, for example, they don’t want anybody at the
service desk handling stock. So there could be people in the business that
do it. There is a rule though. The person who approves — the person
that requests the person that approves and the person that executes [that
approval] cannot be the same person unless there is a staff issue and there
is only one person. In which case there has to be other mitigating controls in place — generally manual business-type controls, someone has to do a sign-off and whatnot.”

As their name implies, manual business controls depend heavily on the human users that interpret and enforce them, as in the case of controls placed on entitlement provisioning:

“So it is really relying on the service folks [...] to correctly identify the process in there as being acceptable and then the help desk people actually follow through with it.”

The vast majority of participants had encountered cases in which users circumvented controls (either automatic technological controls, manual business controls, or both). Many of the stories these participants relayed, however, involved situations where users were not censured for their circumvention: organizations often perceived that the user was acting not from malice, but in the best interests of the organization. Rather than being seen as breaking organization policy, users were often perceived as optimizing around unreasonable controls.

5.1.1 Routing work through the cloud

As a first example, let us consider a scenario in which users at one of our participant financial organizations circumvented a software vetting process by using a cloud-based document-sharing application.

This firm had a well-defined approval process for all software purchases; thus, for example, if a consumer investment department received a request from high-value client to communicate using a new video chat program, company policy required that the department request a review of the program before using it. This review process included an examination of the program’s security: e.g., is the video communication
encrypted? is it built on a peer-to-peer infrastructure, or is all video routed through the software company’s server in North Korea?

At the time of our interviews, most users shared documents within the organization over email or networked drives. However, a set of users — a participant who shared this anecdote conjectured that they were likely new employees fresh out of business school — started using SharePoint, a enterprise content management system provided (at the time) for free by Microsoft. Files uploaded to SharePoint were stored on Microsoft’s servers, and could be shared with a simple URL, which was much easier than waiting for a large file to load, as with email, or trying to navigate to the correct file in a networked filesystem. SharePoint files could also be accessed anywhere, including from machines not managed by the firm, and were thus were particularly convenient when sharing information with external partners.

Of course, when they discovered the use of SharePoint within the firm, members of the infosec team viewed these features as potential security threats, and believed that employees using it were in violation of the spirit (although perhaps not the letter, as SharePoint was not actually an application installed or running on users’ workstations) of the organization’s software review policy. However, their analysis of the situation focused less on the behavior of the users, and more on the drivers behind that behavior. Our conversation included questions such as the following:

- **Awareness:** Were users aware of the software review requirement? Did they realize that a security review of SharePoint would be in the firm’s best interest? Did they act knowingly, or in ignorance?

- **Cost of compliance:** Software review was tied to purchasing, and initiated by sub-organizations within the business, not individuals. If a user saw a free tool like SharePoint that they believed could improve their productivity, what steps did they have to go through to initiate a software review request? If the
software review process was faster or otherwise less onerous, would that have impacted users’ decision to not engage with it?

- **Actual outcome:** Would it be reasonable to provide the SharePoint functionality users desired, or were there some features that were fundamentally incompatible with the organization’s security goals? (E.g., under what circumstances is it acceptable for users to share internal documents with external partners, and is the organization willing to accept the risk posed by allowing users to make those decisions individually?)

Taken at face value, the organization’s software review policy would lead an auditor (or a literal-minded computer scientist) to believe that a user’s desire to use a particular program could be judged in stark, binary terms: either the user was using approved software, or she was not. Either the organization sanctioned the software, or it did not. Either the software was secure, or it was not. From the perspective of real-world practitioners, however, users’ circumvention of the software review policy existed in a grey area that begged further investigation.

### 5.1.2 Password use and abuse

Many participants shared stories about users circumventing security controls through the “misuse” of passwords; this is not surprising, as passwords are the single most commonly-used authentication mechanism in the corporate world. Many organizations have policies regarding password use: passwords must be of a certain length and complexity, they must be changed with a certain frequency, they must not be written down, etc. (Appendix A offers a more thorough summary and commentary on current password policy practices; I originally prepared it as a stand-alone document for one of our partner organizations in another study that was seeking to revise its password policy.)
One financial participant shared a story about his organization in which it had become apparent that users did not know the Active Directory passwords they were supposedly using to log in to their workstations each day; when asked to authenticate in another environment as part of a new business procedure, they were unable to do so. In the past the firm had trouble with passwords being written on sticky notes and attached to computer screens, but the monitors of the users in question appeared to be free of such policy-violating flags. After some sleuthing, the participant discovered that large numbers of users had adapted their strategy after the earlier crackdown, and were now taping password-laden sticky notes to the undersides of their keyboards.

From the perspective of a security professional, it would seem that users were not only going out of their way to violate the password policy, but were actively engaged in deception to hide the violation! However, a deeper investigation of the situation revealed that up until that point users had only needed those passwords for initial daily workstation authentication; they were still required to use a variety of other passwords to access email, internal databases, and other business resources. Users perceived that the value of the workstation passwords were comparatively low, and had a hard time remembering them because they were used comparatively rarely; in this situation, their decision to violate policy was actually quite reasonable.

Infosec professionals from participant organizations seemed to value this sort of insight into users’ decision-making, in part because it helped them craft security policies that made it easy users to do the right thing. Many participants addressed user password management by citing an ongoing focus on improving their organizations’ use of single sign-on (SSO) technology; although compromise of a SSO password poses a greater risk, it was easier for the infosec team to quantify and deal with that centralized risk than it would be to cope with the possibility that users throughout the organization were constantly finding new and creative ways to make single-system passwords vulnerable.
In the case of the users who cleverly hid their passwords under their keyboards, the organization sought to make passwords more memorable. Given that users had a hard time remembering infrequently-used passwords, it explored requiring users to log in multiple times a day. If implemented using inactivity timeouts, this approach would have the added benefit of reducing time when workstations were left logged in while unattended (i.e., when the user was at lunch). Timeouts and forced re-authentication were reasonable for some areas of the organization, but disastrous in others; for example, users in some departments spent long periods of time watching stock market data flow across one computer screen, using a different computer or a telephone to react to the market’s fluctuations. Users in that environment did not react well to being locked out of their real-time data feed because they had not moved a mouse or typed on a keyboard, and given the importance of those users to the profitability of the organization, executives mirrored their negative reaction.

The infosec team explored alternatives approaches to making workstation passwords more memorable — or at least manageable — in these departments, but their intense culture had offered some important insight: users in these environments worked in close contact with their peers (and would thus easily recognize an interloper), and it was vanishingly rare for the floor to be empty on a weekday during the business day. The organization was thus able to rely on informal social control to protect unattended workstations during operating hours, rather than resorting to a timeout.\footnote{Apparently the nature of the work (i.e., its dependence on the stock market being open) meant that users in those departments simply did not try to conduct business during off hours. The organization thus implemented an automatic logout as part of the daily machine refresh after the markets closed.}

**Password sharing**

Many participants reported encountering users who circumvented password-based controls by sharing passwords, although few were able to give concrete example sce-
narios. Of course, instances of password sharing come up periodically in the popular press. For example, Jérôme Kerviel made headlines in 2008 for conducting 50 billion euros’ worth of fraudulent trades at his then-employer, Société Générale; in 2010, he was sentenced to three years of jail time and monetary restitution of 4.9 billion euros, which is what the bank estimated it lost as a result of his actions [11]. Although the organization did not officially sanction password sharing, Kerviel claims that his supervisors largely turned a blind eye as long as his trades were profitable, and that he used passwords of fellow traders to disguise his activities in the official records [47]. If password sharing was indeed tolerated by Kerviel’s supervisors, the mismatch between the bank’s official policy (“Don’t share passwords”) and their functional policy (“It’s ok to share passwords if you make lots of money”) certainly contributed to auditors’ inability to detect the fraud taking place.

Although few of our study participants were able to give concrete details about password sharing within their organizations, one participant from the software industry did spend time on “generic” or “service” accounts, used “for letting various services and things that need to run that you need an account to run because Windows requires accounts to run”. Unlike user accounts, generic accounts are not affiliated with an individual human, but may be shared by several people who interact with a specific software program or business process. Because this represents a sanctioned form of password-sharing within the organization, we can use it to gain insight into some of the reasons users are motivated to share passwords illicitly, and into the challenges password-sharing poses for the organization.

The participant explained that generic accounts are popular because they allow users to transparently share access to data and functionality without having to provision individual user accounts: if Alice, Bob, and Carol need to manage the LDAP server, it is probably easier to create a single LDAP_admin account with the correct permissions than it is to add entitlements to each user’s existing account. If Susan
joins the team, or needs to help out in an emergency situation, it is quick and easy for Bob to give her the access she needs: he just tells her the password for the generic account.

The participant estimated that for the roughly 14k employees in his organization there were a total of 5-6k provisioned generic accounts. While not all of these generic accounts were used regularly by humans to log in to machines — some were used to run automated background processes — they represent a significant proportion of the total accounts in the organization. Despite this prevalence, the participant reported trying to phase out their use:

“They are hard to maintain — ownership rules change and the passwords are shared, sometimes a business support group and there might be five people that support a service and runs generic accounts. Now five people know the password. It could be owned by a person that changes roles now you have to keep track of that. [...] There’s a lot of overhead and we are looking for ways to better manage that. It’s one of the larger problems we have and it was a source of many findings for us in as well in the past.”

In other words, although generic accounts provide greater flexibility for users during their day-to-day activities, they also require additional management effort from administrators. The organization has well-defined processes that streamline the management of traditional user accounts, but these processes in turn represent overhead to users seeking to provision their peers for a specific task at hand. If the organization phases out generic accounts without taking additional measures to make individual accounts more flexible, it would seem likely that some users might start sharing passwords to accomplish their operational goals. I believe this would ultimately be counter-productive: although the organization might appear on paper to be more secure after removing generic accounts, ad-hoc and concealed password-sharing provides an effective backdrop for the rogue actions of the next Jérôme Kerviel.
5.1.3 Modeling the circumvention of controls

The factors that motivate users to share passwords — and, indeed, to circumvent controls in general — are too complex to address fully in this study (although I do touch on them again with the results of my clinical study in Chapter 7). However, we have sufficient information to provide at least a rough sketch of how users might make the choice to circumvent, and what the ramifications are for the organization.

From the user’s perspective

Figure 5.1 shows my model of what happens when a user Dana encounters a control while trying to accomplish a particular task. She may choose to attempt to satisfy the control (e.g., enter her password, place her eye in front of a retinal scanner), which carries with it a cost $C_S$ (e.g., the hassle of remembering the password or taking off her glasses); her attempt to satisfy the control may or may not succeed.\(^2\) Dana may also choose to give up, although this guarantees that she will not complete her task, and thus carries a cost $C_U$ (e.g., fallout for failing to do her job). If she does not think she will be able to satisfy the control — if, for example, she was not provisioned with sufficient entitlements and has failed to satisfy the control in the past — she may protest the control (or, more exactly, the control’s outcome), which carries a cost $C_P$ (representing lost time, embarrassment explaining the situation to her supervisor, frustration at having to call the helpdesk, etc.). Finally, Dana may choose to attempt to circumvent the control, which may or may not succeed, and which carries a cost $C_C$, which will likely include the time she spends trying to circumvent the control. $C_C$ also represents any potentially negative consequences for choosing to circumvent the control, scaled by the probability that her circumvention is detected. If she attempts but fails to satisfy or circumvent the control, she finds herself back in her initial state.

\(^2\)Note here that “success” is defined from Dana’s perspective: she aims to achieve a task, and failing to satisfy the control (e.g., entering an incorrect password) will not help her in that effort.
Figure 5.1: A user’s perspective on encountering a control. $C_P$ represents the cost the user perceives she will experience if she protests the control, $C_U$ the cost of abandoning her task, and $C_S$ and $C_C$ the costs of she attempting to satisfy or circumvent the control.

before the control,$^3$ minus the $C_S$ or $C_C$ “spent” enacting her choice.

**From the organization’s perspective**

When designing the system from the perspective of an infosec researcher, we intend to have users provisioned correctly, to have the controls be accurate in their verdicts, and — in the rare cases where a user is incorrectly unable to satisfy a control — that the process for rectifying the situation is swift and painless. In such a system we anticipate that Dana’s infosec team would try to minimize $C_S$ and maximize $C_C$, i.e., have users perceive that it is easy (inexpensive) to do the right thing and work with the control, and and hard (expensive) to try and circumvent it. They would also aim to have $C_P < C_C$, and scale $C_U$ to incentivize users to draw attention to problems rather than silently circumventing them or giving up (users who protest increase the organization’s ability to fix those problems and improve the effectiveness and efficiency of the system over time).

Unfortunately, my fieldwork indicates that users in real organizations frequently

---

$^3$This model does not reflect systems that keep track of multiple attempts to satisfy a control. A system that refuses access after three failed login attempts, for example, would likely include distinct costs $C_{S1}$, $C_{S2}$, and $C_{S3}$, and a path to a new failed end state after the third attempt.
encounter systems that do not meet these optimal specifications. Systems with incorrect entitlement assignments or inaccurate controls frustrate users, and increase the expected value of $C_S$ relative to $C_C$. Systems with higher $C_P$ relative to $C_C$ or $C_U$ make it harder for the organization to get the feedback it needs to improve the system. As modeled in Figure 5.2, a choice to circumvent or give up in such a system not only fails to provide the organization with accurate records or feedback, but can even corrupt the streams of information coming from users who attempt to satisfy or protest controls. (Unwinding Jérôme Kerviel’s fraud was likely difficult in part because the logs could not distinguish between actions initiated legitimately by his colleagues and actions initiated by him using their credentials.)

This means that getting the incentives balanced correctly to genuinely discourage users from circumventing controls is not just a desirable part of the design process, but a critical one: systems that facilitate circumvention are at constant risk of silent self-sabotage.
5.2 Nature abhors under-entitlement

The first half of this chapter considered ways in which an organization can work against itself by inspiring, tolerating, or ignoring user circumvention of security controls. We next consider another problematic phenomenon, one that seems practically universal among large organizations: user over-entitlement.

Professionals from the financial industry were the first study participants to point out that users not only tend to be over-entitled, but that they tend to become more so with time. They noted that it was possible to trace the career path of a senior users who had been with the company for many years by looking at their current entitlements. Figure 5.3 shows a sample of entitlements currently assigned to a hypothetical user, Carol Jones. Although she is currently an “Associate Vice President”, many of her entitlements are typically associated with positions she held earlier in her career. A person familiar with the organization’s entitlement descriptors could easily reconstruct Carol’s path from an entry-level user support person to her current job via multiple supervisory and management positions.

Being able to trace an employee’s career path is not in and of itself necessarily worrisome. However, it indicates that users are not being deprovisioned when they are promoted or transferred to new positions. In Carol’s case, her promotion from “Group manager” to “Associate VP” spanned a separation of duty constraint designed to control purchasing; because she did not lose the ability to perform \texttt{PurchaseRequest\_L3} when she gained \texttt{ApprovePurchase\_L3}, it may now be possible for her to both request and approve a high-level purchase by herself.

\footnote{I have been conducting an informal study of security professionals from large organizations wherever I encounter them; thus far, every one has said that their organization was either recently or currently struggling with rampant over-entitlement.}
<table>
<thead>
<tr>
<th>Entitlement Descriptor</th>
<th>Typical Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApprovePurchase_L3</td>
<td>Associate VP</td>
</tr>
<tr>
<td>ReconcileBudget-group</td>
<td>Group manager</td>
</tr>
<tr>
<td>PurchaseRequest_L3</td>
<td>Group manager</td>
</tr>
<tr>
<td>EmployeeRecords_supportGroup</td>
<td>Group manager</td>
</tr>
<tr>
<td>ReconcileBudget-team</td>
<td>Project manager</td>
</tr>
<tr>
<td>EditPublicPages</td>
<td>Web editor</td>
</tr>
<tr>
<td>EmployeeRecords_dnguyen</td>
<td>Direct supervisor</td>
</tr>
<tr>
<td>EmployeeRecords_agupta</td>
<td>Direct supervisor</td>
</tr>
<tr>
<td>EmployeeRecords_bmiller</td>
<td>Direct supervisor</td>
</tr>
<tr>
<td>SupportTicketApprove</td>
<td>Support lead</td>
</tr>
<tr>
<td>HelpdeskTicketsReview</td>
<td>Support lead</td>
</tr>
<tr>
<td>PurchaseRequest_L1</td>
<td>Support analyst</td>
</tr>
<tr>
<td>SupportTicketEscalate</td>
<td>Support analyst</td>
</tr>
<tr>
<td>HelpdeskTicketsAppend</td>
<td>Support consultant</td>
</tr>
<tr>
<td>HelpdeskTicketsRead</td>
<td>Support consultant</td>
</tr>
<tr>
<td>BasicIntranet</td>
<td>All users</td>
</tr>
<tr>
<td>EmployeeRecords_cmiller</td>
<td>All users</td>
</tr>
</tbody>
</table>

Figure 5.3: Participants reported being able to trace a senior user’s career through a company by her currently provisioned entitlements. In this sample of hypothetical user Carol Miller’s entitlements, we see evidence that she once worked a user support consultant, and progressed through several other positions since starting with the company.

5.2.1 A parallel with “system high”

The tendency toward over-entitlement in large organizations would seem to parallel the tendency of multi-level security systems to operate in “system high” mode, described here by Landwehr et al. in work on military message systems [52]:

Military message systems are required to enforce certain security rules. For example, they must insure that users cannot view messages for which they are not cleared. Unfortunately, most automated systems cannot be trusted to enforce such rules. The result is that many military message systems operate in “system-high” mode: each user is cleared to the level of the most highly classified information on the system. A consequence of system-high operation is that all data leaving the computer system must be classified at the system-high level until a human reviewer assigns the proper classification.
Operation at system high — which requires maximal user clearance — seems antithetical to the multi-level security approach, which aims to protect data while maximizing authorized users’ access to it. However, the phenomenon exists in military settings (among others) for entirely understandable reasons; certain facts about the organization’s goals and the capabilities of computing systems combine to make convergence to a state of system high somewhat predictable. Landwehr does not state these points explicitly, but I believe that they implicitly underlie his discussion.

**MLS Conjecture:** It is difficult for an automated computer program to accurately classify real-world data, but relatively easy to do so approximately.

**MLS Conjecture:** Accurate manual classification of data, or reclassification, is expensive (in person-hours) compared to approximate automated classification.

**MLS Conjecture:** Given a choice between assigning a data classification that is likely to high and assigning one that is likely too low, it is preferable in military environments to err on the side of higher classification.

**MLS Conjecture:** Resources (i.e., person hours) are limited.

The system high phenomenon sacrifices users’ access to data in exchange for protecting that data more rigorously. As I will now argue, I believe that organizations experience systematic over-entitlement because they make the opposite exchange: they sacrifice rigorous limitation of privilege, but guarantee that users have sufficient data access to fulfill their business expectations.
5.2.2 Entitlement errors

As it is difficult to accurately classify data in a multi-level security system, it is also hard to automatically provision users with appropriate entitlements. Participants, such as this infosec team member from the software industry, reported that the majority of provisioning in their organizations is done manually, but that they would like to see the process become more automated:

“[Provisioning] involves somebody figuring out what [the users] need access to, creating a ticket in Service Desk, and waiting for someone else to figure out which systems they are referring to and then going in and provisioning that access. And there is some document ultimately about what — who is the owner of the system, who is the access approver of the system. Here is the method that you give access to. That is largely not automated today and that is what the other products that we are trying to implement will help us do.”

In this instance, the participant was focused on automating the information-gathering aspects of the provisioning process, rather than the actual approval or rejection of the provisioning request they proceed. In his vision for the future, there would be a quick and easy way to identify what people “need access to,” to specify the system in question, and to identify the system’s owner or access approver to whom the request should be routed.

Assume for a moment that the participant’s organization achieved this perfect information availability. Like any manual task, the provisioning decision itself would still be subject to a certain error rate; we consider it normal that humans periodically make typos, misread instructions, or experience other imperfect operation. If

5At first I was surprised by how much trouble organizations seem to have keeping track of basic information about themselves. Now I believe that this self-understanding is a key factor in the success of organizations’ access management efforts, as I discuss at greater length in Chapter 6.
a provisioning error produces a change in users’ ability to access data, it will likely result in one of the following cases:

**Case 1:** The error prevents the user from accessing certain data where an error-free provisioning would have allowed such access; i.e., it results in under-entitlement. If the discrepancy makes it difficult or impossible for the user to complete her job, she will likely draw attention to this fact.

**Case 2:** The error allows the user to access certain data where an error-free provisioning would have prevented such access; i.e., it results in over-entitlement. If the discrepancy allows the user to access data that she does not need to complete her job, she may or may not become aware of the over-entitlement, and may or may not draw attention to it.

If the user does not draw attention to the error, it would likely remain in place until the next time the user’s access permissions are reviewed (e.g., the next time she requests additional entitlements). Because the user is more likely to draw attention to errors when they result in under-entitlement, *errors of over-entitlement are more likely persist.*

### 5.2.3 Goldilocks is hard: modeling provisioning pressures

Reports from participants indicate that provisioning errors are common in practice; it is easy to overprovision or underprovision, but hard to get entitlements “just right”. The case-based analysis of provisioning errors above is somewhat simplistic, because it assumes that errors of over-entitlement and under-entitlement are equally likely; in reality, Goldilocks is more likely to choose to overprovision, just as military systems are built to err on the side of overclassification.
Figure 5.4: My model of the process that Susan goes through in processing a provisioning request from manager Alice for user Bob. When deciding how many entitlements to assign, Susan weighs the possibility that incorrect entitlements will result in another (potentially more frustrated) request from Alice. Traditional security principles dictate that Susan should err on the side of under-entitlement.
The rest of this chapter is dedicated to describing the operational pressures and user perceptions that contribute to this “Goldilocks” phenomenon by modeling the decisions a provisioning user must make. Note that what follows is not an “access control model” in the traditional sense; it focuses less on the internal workings of authorization policy, and more on the operational impact of the humans’ decisions on the prevalence of over-entitlement in the organization.

**Scenario**

Susan is a system administrator charged with provisioning users at the Acme Company. We assume that Acme has a centralized provisioning system that allows Susan to easily see and modify all the permissions any given user has. (If she’s not able to provision everything, costs in this scenario are multiplied by the number of administrators involved. If Susan not able to see all permissions, costs in this scenario are multiplied by the number of different databases she has to consult.)

Figure 5.4 summarizes the process Susan goes through when she receives a request from Alice to provide user Bob with new entitlements following his promotion. Based on this outline, Figure 5.5 models the workflow and decisions Susan must make while provisioning a user. In this model, we represent the cost or time penalty associated with each action as *Susan perceives them* as $C_1...C_{10}$. This allows us to see how Susan’s perception of the value of each cost is likely to influence her choices while traversing this workflow throughout the day. Similarly, Figure 5.7 models the provisioning process from the perspective of Alice, including the costs as she perceives them.

**Discussion**

The costs $C_1, C_3, C_5, C_7$, and $C_9$ are associated with the basic operations that Susan must perform with the provisioning software (i.e., accepting a new request, creating a
Figure 5.5: My model of the workflow and costs that Susan (the system administrator) experiences in the given scenario. The table below contains a key to each cost $C_1...C_{10}$.

<table>
<thead>
<tr>
<th>$C_1$</th>
<th>Time spent on initial request processing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_2$</td>
<td>Time spent interacting with supervisor.</td>
</tr>
<tr>
<td>$C_3$</td>
<td>Time spent during initial role creation.</td>
</tr>
<tr>
<td>$C_4$</td>
<td>Time spent considering entitlements in the “too few” case.</td>
</tr>
<tr>
<td>$C_5$</td>
<td>Time spent assigning entitlements in the “too few” case.</td>
</tr>
<tr>
<td>$C_6$</td>
<td>Time spent considering entitlements in the “just right” case.</td>
</tr>
<tr>
<td>$C_7$</td>
<td>Time spent assigning entitlements in the “just right” case.</td>
</tr>
<tr>
<td>$C_8$</td>
<td>Time spent considering entitlements in the “too many” case.</td>
</tr>
<tr>
<td>$C_9$</td>
<td>Time spent assigning entitlements in the “too many” case.</td>
</tr>
<tr>
<td>$C_{10}$</td>
<td>Time spent processing the followup request.</td>
</tr>
</tbody>
</table>

Figure 5.6: Key to costs depicted in Figure 5.5.
Figure 5.7: My model of workflow and costs associated with provisioning from the perspective of Alice, the manager requesting provisions for her employee Bob. The table below contains a key to each cost $C_{11}...C_{14}$

| $C_{11}$ | Time spent preparing and submitting the request. |
| $C_{12}$ | Alice’s lost productivity. |
| $C_{13}$ | Bob’s lost productivity. |
| $C_{14}$ | Time spent interacting with the administrator. |

Figure 5.8: Key to costs depicted in Figure 5.7.

new role, and actually assigning entitlements to it). These are likely to be relatively small, because software developers perceive that they will be performed frequently and therefore try to optimize them for usability. Fieldwork indicates that they are dwarfed by the costs of figuring out which entitlements to assign ($C_1$, $C_4$, $C_6$, and $C_8$).

The cost $C_2$ represents the time that Susan and Alice will spend exchanging information about the provisioning request. Traditional access-control models assume that this interaction will iterate until Susan has enough information to provision Bob appropriately; in reality, Susan will likely minimize this interaction if she perceives it to be too costly (i.e., it takes too much of her time, of Alice’s time, or the interaction is unpleasant.) The cost that Susan perceives can depend in part on how senior Alice is in the organization (i.e., she will consider $C_2$ to be quite large if Alice is the CEO).

This model assumes that Susan’s actions will be evaluated based on Bob’s ability to get his job done (i.e., whether or not Alice must follow up with an additional
provisioning request). This reflects what I observed in the field: system administrators who provide services to business users are often evaluated based on how long the process takes \((C_1, C_2, C_3, C_5, C_7, \text{ and } C_9)\) and how many interactions are required before the issue is settled \((C_{10})\). Susan thus has a strong motivation to try and minimize \(P(\text{followup})\).

### 5.2.4 Compliance budgets

My models of provisioning pressures do not currently reflect the frustration or fatigue that administrators and users can experience while trying to make a computer system do what they need it to. As explored by Beautement et al.’s work on compliance budgets [4], hard-to-quantify costs such as these can dramatically impact users’ ability to comply with an organization’s policies over time. In the scenario discussed above, even if most of the other costs could be optimized to make the provisioning process easy, it is easy to see how Susan the system administrator might quickly deplete her compliance budget and err on the side of over-entitlement.

### 5.3 Implications and conclusions

The results I present in this chapter indicate that all is not well in the world of corporate access control. More worrisome still is that my observations likely under-represent the scope of the problem. Because the participants were almost exclusively drawn from the information-security staff, these reports represent just one side of the story. Participants were able to end most anecdotes by describing a modification they made that allowed users to both be in compliance and get their jobs done; in the clinical study, where we interviewed end-users in addition to executives, it is clear that solutions are not always quite so tidy.

Many of the problems documented in this chapter could probably be addressed by
better business processes. In Chapter 6 I examine some of the methods participant organizations have implemented to combat the tendency toward over-entitlement, and the specific mechanisms they have employed to bring current entitlements in line with what employees actually need in the course of their jobs. We cannot, however, focus our efforts solely on process-based solutions. Processes are implemented by human users, and I argue that users are not the real source of the problem. (If they were, better user education would suffice — but users already know that they should not be circumventing controls, so additional browbeating is unlikely to change that behavior.) We as system designers need to better understand the requirements of a situation and the limitations of the users who operate in it before we start on a design. A shallow usability analysis — i.e, one which looks at a graphical user interface and users’ interaction with it — is not sufficient to uncover the multitude of pressures that influence users in the real world. System design must be an iterative process, conducted in cooperation with real users in the context of their real jobs. (Of course, I recognize that it is much easier to call for this type of design process than it is to implement it in practice.)

However, better design processes and the tools they produce cannot eliminate all sources of error in organizations of this scale. Just as with other domains in which an organization faces risk, we must accept the presence of errors and craft our expectations accordingly. Right now, both the research community and the larger population of information security professionals seem to grow a reality distortion field around access-control systems. We design our controls, and our methods for auditing compliance, and happily declare victory when the latter uncover minimal violations. In documenting users’ circumvention and the tendency toward over-entitlement, I want most of all to emphasize that we should expect phenomena like these in all systems: rather than be surprised by them, we should seek them out, and hold them as evidence that our methods have room for improvement. Until we do so,
organizations will continue to fly blindly down a path of noncompliance, unable to see sources of massive risk until they bear disastrous fruit.
Chapter 6

Corporate study results: factors external to the user

Chapter 5 presented two important results of the corporate study:

1. Users of participant organizations circumvent access controls when they believe doing so will help them get their jobs done, and

2. Participant organizations tend toward a state of over-entitlement as a product of trying to maximize individual productivity.

Both of these phenomena result from the aggregate decisions of individual users, and can be better understood by modeling their decision-making process. However, the models presented in Chapter 5 are too simple to capture the great diversity of factors that can impact an organization’s access and identity management efforts. This chapter explores factors that can influence individual users’ decision-making, but that are themselves characteristics of the larger organization. It also explores some of the methods and processes that organizations use to shape their information security practices.
6.1 Complications: culture, compatibility, and clunkers

During my discussions with participants I learned about a variety of organizational factors that can influence the decisions individuals make in relationship to information security, including the organization’s culture, its desire to be perceived in a positive light by external entities, the technological limitations it experiences, and its process of growing through mergers and acquisitions. These observations focus on information security writ large, but apply to identity and access management within that domain.

6.1.1 Corporate culture

Participants from one financial organization had much to say about the firm’s culture and its influence on the infosec team’s efforts. In particular, they attributed much of their understanding of business users’ needs to those users’ willingness to share their opinions about new technologies. When we told these participants an anecdote from another industry about users who found it easier to silently circumvent controls than to complain to the IT department, they literally laughed out loud. They assured us that their users (many of whom achieve great professional success by being assertive in the workplace) do not hesitate to share their opinions with the IT staff, especially when they perceive a computer system is getting in the way of them making money for the firm.

The role of values  It makes sense that users who are highly attuned to an organization’s profitability would feel empowered to speak up when they saw an infosec control threatening that profitability — especially if their personal bonus was tied to it. Other industries value other principles more highly; for example, I imagine that members of the intelligence community consider it more important to preserve the
confidentiality of sensitive data than it is to minimize costs in handling that data.

**Variability within the organization** Participants from this organization reported that the culture and values of individual branches of an organization can also vary significantly. While users had a strong ethos for doing “what is right for the firm” on the whole, the responsibilities and management of individual departments impact users’ definition of “right”, and thus the way in which they react to infosec initiatives. In a similar fashion, an infosec team for a world-wide organization may experience additional challenges that stem from the diversity in the cultures of employees. For example, one participant said,

“In [this asian country], employees get very upset when they have to click “too many” times. They want to do the work, but they want it to be very easy to use. They also got pissed off when they didn’t have someone above them who is approving their access; they don’t want anyone below them in the company hierarchy to be approving them.”

6.1.2 Reputation risk

Another problem that taxes infosec teams is “reputation risk,” a term used by multiple financial participants to describe the consequences associated with negative publicity. (A participant from the software company also referred to this as “brand risk”.) When news of an organization’s infosec problems become public, it can reduce customers’ and shareholders’ confidence in the organization. This can have severe financial consequences, as in the case of Exxon-Mobile and the Exxon Valdez oil spill. One participant shared a conversation he had had with an Exxon executive:

“He said you could steal ten million dollars from me, I don’t really care — we’ll prosecute you and everything else — he goes like god forbid a
ship splits open because somebody fudged a maintenance log or the guy was drunk and we didn’t test him correctly — as they did in Valdez. He said I can lose ten percent of my stock overnight. He goes — that’s like a trillion dollars. I ain’t never taking that again.”

Unfortunately, the specter of reputation risk can reduce an organization’s willingness to sharing information with external parties about infosec problems they might be facing. One participant in the financial industry told us that he was part of an exclusive group of financial infosec professionals from around the globe who used private meetings to discuss common challenges and potential best practices. This secretive venue was necessary, he said, to protect his organization from the reputation risk it might face from an infosec problem leaking to the press.

Although we cannot remove the impact of reputation risk, we should be aware that it can have a strong influence on organizations (and their individual members) when they are deciding whether to report information security problems.

### 6.1.3 External systems and regulations

Participants also discussed ways in which external standards can pose additional challenges for the infosec team. For example, the diversity of international infosec regulations can make it difficult for multi-national organizations to develop a coherent strategy that is compatible with the requirements of all jurisdictions. Although the US and the EU have similar laws requiring organizations to regularly review users’ assigned entitlements, the penalty for not noticing small differences can be great. Even within the US, where individual states have different reporting requirements for data breaches, an organization may invest a lot of time and resources into understanding the patchwork of laws.

Above and beyond legal requirements imposed by regulatory bodies, participants described cases in which a decision by an external partner — for example, a com-
cial bank in a foreign country — could significantly impact the org’s efforts to roll out larger or more sophisticated infosec technology.

Participant: “Some of these systems [...] their access list is a screen shot. And I can’t do anything logically with a screen shot except look at it. That’s because it’s a hosted application lets say.”

Interviewer: “Are there any efforts to try and standardize the types of applications?”

Participant: “There is — there’s what they call an enterprise architecture — it’s really sponsored through here, but it’s got the support of all these other teams. And they try to standardize exactly that. But again we deal with I don’t know how many countries, fifty some odd countries or something. If the Bank of Japan or whatever, they say you must use this system and we don’t get a choice. If we don’t get an access list, well that’s pretty much it.”

Constraints imposed by well-intentioned regulators and contracted partners can dramatically limit an information-security group’s ability to build or buy products that actually meet its users’ needs. Even if the group can identify what those needs are, they may not be able to act on this insight in practice.

6.1.4 The technology ecosystem

In addition to cultural, social, legal, and contractual factors, participants reported that they struggle to develop identity and access management solutions that can work with the variety of technological systems present within a single large organization. I refer to the intricate web of interdependent programs, servers, and data feeds as the organization’s “technology ecosystem”. Organizations participating in this study have had computers at the heart of their business for several decades; after working
in the complex ecosystem that results, participants cited the following as challenging elements.

**System age** Some participants reported that their organizations depend on critical systems built on decades-old mainframe machines. In most cases the number of users who interact with these legacy systems was small, but as time goes on it becomes increasingly difficult to interface them with newer, more advanced technologies.¹

**Diversity** It is less expensive for an organization to maintain standardized systems (e.g., to have three basic builds from which an administrator might choose a new server), but participants reported a tremendous diversity of operating systems, applications, and hardware within their organizations. As one software-industry participant said, “We probably have every flavor of Unix […] somewhere in the company”. Part of this diversity is due to the presence of legacy systems, but a decentralized structure for IT purchasing and management can also exacerbate the problem.

**GUI usability** The usability of an individual interface can have a significant impact on its practical utility. However, participants did not always feel that a good visual and interaction design to be absolutely necessary for every application, especially if an application’s users are highly-trained specialists, and they have only rare occasion to use it. One participant did a demonstration of a policy-authoring tool that was strikingly ugly; when we asked whether people ever actually got it to do what they wanted, he responded “You wouldn’t think so — I know, I know — you would think different, but it works”. The usability of end-user interfaces is important, but if an organization has limited resources for improving interface designs, it should work to

¹The reasons participants gave for still having such ancient systems warrants a study of its own. In general, participants seemed to focus on the need for the critical systems in question to be highly reliable. Some participants said that their organizations have tried to build replacement systems — but that such efforts are in some cases a decade or more past their intended deployment date, and do not show any signs of producing a system that works as well as the aged one still chugging away in the basement.
evaluate where a re-design effort will have the highest impact in practice (and not just beautify designs that are functional but ugly).

**Certified software** Of course, this does not necessarily mean that a group can purchase and use any software it wishes; one participant noted that “I worked for a bank and it was... here are the five products you are allowed to buy because they are certified”. He did not go into detail as to the exact certifications his former employer required; what is interesting is that an information security professional can be constrained from the start by standards that may or may not make sense as a measure of the quality of an identity and access management product.

**Source code availability** I was surprised to learn that participant organizations rely on software for which the source code is no longer available. Whether developed in-house or purchased from external vendors (who then went out of business), having just the executable form of an application seems to be a common impediment to the development of patches to update the application’s functionality or interoperability.

**Interoperability** The previous factors — age, diversity, and source code availability — can combine and make it difficult to convince various computer systems within an organization to interoperate. Interoperability is particularly important when dealing with IAM products, as much of IAM benefit comes from automating data management across different applications in the organization. Participants report that vendors make grand claims about their products’ interoperability, but that these assertions must be tested before trusted. A participant from the software industry spoke with excitement about the company’s acquisition of technology to “make all our products talk to each other,” although I noted to myself at the time that I thought the problem was perhaps outside the scope of any single software solution.  

\[^{2}\text{This may be an example of an infosec professional having unrealistic expectations for the capability of a technical product to solve a problem that has both social and technical roots; Section 6.2.2}\]
**Credentials and roles**  One interoperability challenge that is of particular concern is the ability to reconcile different applications’ “local idea of roles” and credentials. Participants from the financial industry described significant efforts to standardize and disseminate entitlement data throughout the organization, so that applications which had been configured earlier in the life of the organization could be adapted to successfully operate using a common vocabulary.

### 6.1.5 Acquisitions and mergers

Another challenge participants reported their organizations facing is the speed at which the population and structure of the organization itself changes. In particular, participants from the software industry cited their organization’s acquisition of smaller companies as an impetus for change, as mergers require the combination of two different sets of technology, business processes, and cultures. Mergers frequently trigger a reorganization of the corporate structure, which can pose problems for infosec professionals, both locally — in cases where the infosec team itself is reorganized — and globally in their efforts to maintain secure systems that accurately meet the needs of their users. One participant offered the following as an example of the challenges mergers and acquisitions represent.

“[The company] basically grew through acquisition. So if you grow by eating five companies a year and those companies have their own — you know whose chief evangelist of technology — and now you bring them into your organization. What they did is they inherited the title. So your division was — we had a little piece we acquired and then from that you had your title — and now I had people that had — six people in one department [with similar jobs but] six different titles.”

*discusses other instances.*
6.2 Organization maturity

In addition to corporate culture, regulatory constraints, and the organization’s technology ecosystem, participants frequently mentioned issues that one infosec professional from SoftCorp described as relating to “organizational maturity”. This participant believed that an organization may need to achieve a certain degree of sophistication or phase of development before it can successfully deploy some IAM technology, and that “you have to look at the organization’s readiness to accept a solution”.

This seems to contradict a basic belief about computer systems. As technologists, we often view the systems we build as self-contained solutions; if an organization dedicates sufficient resources to a system’s configuration, deployment, and maintenance, the system will necessarily help them achieve the goals it was designed for. In the rest of this chapter I present observations indicating that participant organizations should never expect to solve issues of identity and access management with a purely technical or stand-alone solution — and that no deep solution can be deployed in a short timeframe.

6.2.1 Incremental and iterative philosophy

Computer programmers frequently use an iterative method in building software: first, one implements and tests basic functionality, then one repeatedly adds and tests features until the final program is reached. Study participants seemed to have an implicit understanding that they must apply a similarly iterative approach to engineering socio-technical systems, too. Choosing realistic incremental goals for policies is particularly important:

“But people who want to set a policy think all applications are going through our centralized authentication framework by this day. But the minute that day passes and that policy goes into effect and 50% of the
apps still don’t comply they’ll be writing up a lot of policy exceptions or handling it as auto findings which is an uncomfortable place to manage your IT business from.”

Basic functionality in the case of new IAM products is simple replication of old functionality:

“... don’t give me anything else I don’t need yet or want yet — just let me replace what I have so I can remain compliant.”

This attitude meshes well with the policy-creation methodology of one SoftCorp participant. In demonstrating a particularly confusing policy-authoring tool, we asked him how he knew what the order of operations was for the policy language (i.e., how actions were applied to the roles specified in the policy). He responded, simply, “trial and error”. Participants largely did not expect to write a policy and then deploy it in two convenient steps; they expected to pore over it in detail over a long period, making minute tweaks over time until the policy had the desired effect.

**Customization vs customizability**

One SoftCorp participant noted that product customizability was a double-edged sword:

“It’s very customizable. But it means that you have to customize it.”

Customization takes time, but is often necessary to achieve the full benefit of a product. An understanding of the potential pitfalls of customizability would seem useful in trying to shape executives’ expectations. While many products offer sample customization templates, they are not guaranteed to make the task of customization faster; the same participant noted of another product that

“You can do a lot out of the box, but you can’t do anything that we want.”
Another participant noted that developers understand and anticipate the need for customization of products when they are writing them:

“It’s impossible to write for everything. You have to write generalities when you deliver code so there is a lot of customization and in the end, if the expectation is set up that you are just going get it out of the box and tomorrow you are going to be provisioning. That is a false expectation. It is going to take some time. Now if we could do it in six months, we can get that first hit down from six months. We could probably get it down to say three months if you learn from that practice.”

6.2.2 Expectations of infosec

One of the most relevant aspects of an organization’s success with IAM systems seems to be its expectations of the infosec team and technology. Participants reported challenges rooted not only in software and in users, but in managing what corporate executives believed was possible to achieve with technology. One banking participant lamented that

“The business has no idea how long it takes to roll things out. They don’t understand all the implications of what they’re asking. The business wants to put more controls on email, wants to have it done right away. [Right now they’re] pushing for passwords on attachments. [It’s a] relatively non-technical change, a basic control on attachments... [but they have] no idea what they’re getting involved with, because compliance can’t surveil that. [They are] shocked to realize that it’s going to take out six months at minimum to roll out an email solution [or to] to change [any] major system like that. The business doesn’t totally recognize or understand that.”
One SoftCorp participant shared a story about the company’s internal adoption of its own identity management product. After failing to meet an important deployment deadline, members of the infosec team struggled in negotiating a new one with the company’s management team:

“But the sticking point around it was they had these unrealistic expectations of what we could do with the technology and what was really — you know — what our environment was ready to support and what the actual technology was ready to do.”

I imagine this must have been particularly challenging for SoftCorp because they were dealing with a product they themselves had developed. Participants in general indicated that vendors have a reputation for making grandiose claims about their products, especially before a potential client has decided to purchase licenses. Perhaps executives of SoftCorp had internalized the sales-pitch understanding of the product’s functionality — and perhaps this party-line understanding inhibited their ability to perceive root causes of the problems at hand. The participant agreed with this observation in explicitly attributing more blame to the software than to the organization as a whole:

“At the same time I would say there was an almost unrealistic and unreasonable expectation that this would be easy. I think there was a higher level of esteem in the maturity of our solution in this space and a — I won’t say it was an inflated sense of esteem about the maturity of our organization, I would say it was just assumptions. People did not understand the complexity of the problems as described.”

At the same time, another infosec professional from SoftCorp focused more on the fundamental limitations of IAM technology independent of implementation. He
seemed to believe that the company’s infosec team had also had unrealistic expectations for a software solution:

“We had been somewhat led to believe that identity policies could deal with all sorts of magical things, but you really can’t and you can see why they can’t.”

It is not surprising that an IT department generally struggles with managing executives’ expectations of technology. However, the task seems particularly challenging when managing expectations of access-control solutions. Infosec professionals from the banking industry reported that they have to lobby business executives to get buy-in for new security initiatives, and “sell” technology hard to secure funding for it. The academic literature and the professional press seem to agree that IAM is essentially a solved problem, one for which remaining bugs can easily be worked out during implementation.\textsuperscript{3}

While trying to sell an IAM product internally an infosec team could easily inflate its own expectations for that product (or at least suspend the skepticism with which any IT professional must approach the sales literature of a software vendor). Infosec teams that have been thus deluded by the promise of an IAM solution would be hard-pressed after the purchase has been made to offer balanced timelines for deployment; in this case, they have basically undercut their own efforts to manage executive expectations.

While this study reveals hurdles to reasonable executive expectations for IAM technology, it also indicates that an infosec team can work to foster an organizational philosophy that will transcend the hype of any one particular software product. One

\textsuperscript{3}During the course of my studies I have directly encountered this perception from a variety of academic quarters, including tenured computer science professors and one senior-level administrator of a governmental funding organization. Perhaps the depth of the IAM challenge is only comprehensible to people who have tried to deploy the vaunted solutions in practice: at the same conference where academics argued during the session that the organizations I was studying had simply failed to implement RBAC correctly, practitioners practically lined up afterward to tell me how strongly my observations resonated with their experience.
banking participant touched on this philosophy when he described what he considered to be executives’ overreaction to an understandable mistake:

“It was a vendor we’d just reviewed, they have very good practices, but it was just a minor screw up on their website, something we could’ve done. Business doesn’t understand that this is a risk management space, just like anything else.”

6.2.3 Fostering a vision of risk management

One banking participant described an ongoing effort to help the business view information security as an exercise in risk management. He said that executives initially expected the infosec team to be able to prevent all security problems; when a data exposure occurred, management folks said, “We’ll give you money to make this never happen again”. When he initially tried to explain that it was not possible to guarantee with perfect certainty that a data exposure or adverse event would never happen again, he was met with blank stares; these executives were used to requesting software features, and had a hard time differentiating features of security software from security guarantees.

Of course, there are other areas in which businesses are accustomed to managing risk, and many participants made reference to existing practices of risk mitigation in explaining the goals and capabilities of the infosec team. Roughly speaking, participants said that their organizations adopted a process composed of four stages: 1) Identifying the assets or things that are valuable, 2) Identifying the potential risks or adverse events that could happen to those valuable things, 3) Estimating the probability or likelihood that those events will happen, and 4) Making decisions on how to counter that risk.
Anticipating change through risk ownership

Participants at SoftCorp described a particularly well-defined process in which members of the staff took explicit “ownership” of risks that had been identified. This notion of ownership was useful because it tied risk mitigation to individual accountability; explicit risk ownership also provided a mechanism to periodically reevaluate risks even when the structure or composition of the organization changed. (Otherwise, risk mitigation for threats to legacy systems could easily fall by the wayside when focusing on risks posed to new systems, for example.) As another SoftCorp participant said, it was important that they constantly “readjust [their] calculations” of risk.

Perhaps one way in which an organization can experience maturity is by being able to recognize of this potential for organizational change. Rather than simply trying to mitigate the risks that an organization faces today, a mature organization anticipates its own evolution by designing processes that will allow it to constantly adapt its security posture to the new state of the organization. This aspect of maturity is connected to an organization’s understanding of itself, which is discussed in Section 6.2.4.

Security as cost savings

Participants reported that their position as infosec specialists was challenging because they could not produce direct financial benefits (profits or cost savings) to the organization. Other business units — including other IT units — could; for example, a non-security-related IT department might save the organization hundreds of thousands of dollars by building or buying software that streamlines a critical business process. Participants reported that the probabilistic nature of the threats they seek to mitigate makes it difficult for them to express the potential benefits of investing in their initiatives; if a security project saves the organization money, it will likely be
because it prevented an attack — but how to quantify the value of a thwarted attack, when one cannot know if such an attack has or will be attempted?

Because business executives are accustomed to measuring the value of an initiative in monetary terms, participants can find it hard to get buy-in for their funding requests. As one participant put it, “we’re just not a revenue center”.

However, members of one participant organization reported some success in operating not as the owners of the security initiatives, but as a consultant tasked with identifying problems that business organizations need to remedy with their own budgets. This subgroup within the organization’s security department operated largely without funding, for “[Our] program highlights risks. The funding issue [only] comes into play when we’re actually trying to mitigate the risks, not just identify and catalog them”.

Commoditization

Both of the previous examples — where risk ownership and risk mitigation were farmed out to business or other IT groups — are generalizations of the principle of commoditization. This idea was brought to a new level by P-Bank, whose infosec group went so far as to spin off ownership of entire risk-management systems to other teams within the organization. This allowed the infosec team to focus on new challenges — different, unaddressed areas of risk — while making sure the organization effectively managed the known risks associated with business operations.

6.2.4 Self-understanding and structure

The final aspect of organization maturity that this study revealed as being critically important to the success of identity and access management is the organization’s documented awareness of itself. This self-awareness both includes and depends on the structure of the organization; strong centralization lends itself to a standard
Although we often consider organizations as having a strict supervisory hierarchy (left), management literature [61] recognizes that project-based operation and informal supervisory relationships often contribute to a matrixed organizational structure in practice (right). Classification of users, for example, which helps in maintaining consistent records about the employees and their relationships across the organization.

**Identifying supervisors**

There are multiple levels of structure, however, and not all can be represented by tidy organization trees. Before starting the study I assumed that supervisory information—a mapping of employees to supervisors—would be readily accessible within an organization. In practice, participant banks report embracing a more fluid *matrixed* structure, which allows them to react more quickly to changes in the organization’s priorities and its markets. As reflected in Figure 6.1, a user in a matrixed organization may have multiple supervisors or project managers with whom she works daily (to whom she is connected by “dotted line relationships,” as one financial participant put it) in addition to her formal supervisor (whom she may see only a few times a year).

While conducting a firm-wide entitlement review\(^4\) the infosec team of one bank was challenged with coming up with an authoritative source for supervisor information. However, it was hard to identify exactly which supervisor should be able to approve a user’s entitlements: the official supervisor often had no idea of the user’s daily

---

\(^4\)I present these entitlement review in more detail later in the chapter; for now we note that they require a user’s supervisor to regularly review and approve the set of entitlements assigned to the user.
activities, but the informal supervisors who worked closely with the user were neither stable over time nor well-documented. Participants report that the infosec team was not even able to reliably leverage users’ knowledge to gather supervisor information, because the users themselves were unable to identify their supervisor when asked directly.

**Informal knowledge**

As one financial participant said, “It took a long time to figure out who all was working here”. Although the participant could find lots of information about various people within the environment, getting a clean source of data was challenging. As is often the case, much of the information that could be relevant to infosec professionals in their efforts to build and deploy systems is managed or held informally within the organization. For example, one SoftCorp participant was demonstrating a policy-crafting interface for us, and he noted that there were a number of different groups to which users could be assigned. He knew that although many users were assigned to group Y, that group was not in fact provisioned with entitlements deeper in the system; having these users in group Y was meaningless. The participant said “I just happen to know that nobody is [in group Y] anymore”. This bit of knowledge was not actually documented in the policy — although it may have been documented elsewhere — but it is an example of the type of informal knowledge that human organizations have about themselves.

As in the case of supervisors in the financial industry, the informal nature of much of this knowledge can makes it challenging for the infosec team to design systems that must integrate with the structure of the organization. Another SoftCorp participant noted that the infosec team could not serve as a source for the information it needed:

“If it starts at IT it’s going to fail. [...] It starts at the business, and the business is a mess.”
Not only was it difficult for infosec professionals to gather the knowledge they needed to deploy a policy, the nature of how that information is managed within the business itself makes it challenging to extract it in an ordered way.

**Formalizing knowledge**

Participants from SoftCorp were eager to share information about the organization’s recent centralization of human resources data, which contrasts with the struggles I observed in the financial industry. Although the software org was not able to centralize all of its provisioning and entitlement management systems — i.e., they maintained a separate accounts and access-control policies for individual users on different systems — they did architect a single, authoritative data feed for information that was useful to all access-control systems throughout the org. This feed, which was driven by a database maintained by their Human Resources department, provided basic “demilitarized” information about employees’ names, employment status, and geographical location, while restricting access to data on compensation, benefits, and other personal information.

One participant remarked that centralizing the data feed made it harder for users to circumvent provisioning controls: “Unless you are getting your human resources from one source... you will have people going around the process all the time”. If we consider provisioning as a potential opportunity for attack, it makes sense that standardizing and hardening the process might reduce the number of potentially exploitable interactions.

Along with its centralized data feed, SoftCorp maintained a master database of all assets, including ownership information and processes for changing a system’s configuration (including access permissions). One participant reported that “pretty much anything that goes into [the asset database] if it wants to change, it has to go through a change process. You can’t just go in and say okay now I’m changing this
Figure 6.2: Participants from SoftCorp reported that the organization had a centralized database that was fed by information from the HR department. Systems across the organization relied on the central database for basic user information. Another central database tracked information about technology assets, including computer systems, their business owners, and the procedure for changing their configuration.

BAR [Business Area Reviewer, or business owner] to somebody else. It has to go through an approval process — the information is highly managed”.

Although I am not convinced that centralization is absolutely necessary, I argue later in this chapter that accurate ownership information is a critical component of a successful access-control systems for our partner organizations.

Requirements gathering

Traditional software engineering methodology divides the software development process into a number of phases, including one dedicated to requirements gathering. However, participants expressed that it was tremendously difficult to define requirements for IAM systems before the software was actually built and being tested in their environment.

Part of the challenge in defining requirements is the process of getting specifications from business users. One participant from the financial industry reported asking business users “In this scenario, what would happen?” in an effort to root out
complicated use cases. To the participant’s frustration,

“The business folks come back and say, ‘We never expect that to happen, so don’t worry about it,’ but we can’t build code around that kind of functionality!”

Of course, the challenge of soliciting requirements from non-technical users is not new. What is more troublesome here is that even well-seasoned infosec professionals seem to have a hard time telling in advance what they need IAM systems to do. That same financial organization reports building long-term relationships with software vendors who seem competent but who have no products that are currently of interest to the infosec team. These ongoing relationships (and the frequent product demos they include) allow the team to 1) see if the vendor has thought up new functionality that seems useful, and if so, to 2) work with the vendor to push the development of their product to the point that the team might actually consider testing it internally. In other words, these infosec professionals are sufficiently dissatisfied with current IAM tools that they are constantly seeking better solutions from outside the organization. However, even when they find a potentially useful tool, they assume that it still requires significant improvement before they might actually consider using it. While this relationship likely proves beneficial to both the bank and the software vendors, it seems to go against orthodox software development methods; what if other fields developed requirements using this ad-hoc approach?

The challenge of developing requirements for IAM is also evidenced by the Soft-Corp’s enthusiastic practice of testing its identity management product in-house as a path both to validate the software and advertise the its validation to potential clients. Given that few clients will make a large purchase without evidence that the product has been successfully deployed in a real organization, internal validation is

Incidentally, participants from the bank report that these relationships have also proved useful when the venture capital arms of the firm are considering investing in the vendor; in such cases, the infosec team can offer a high-quality technical evaluation of the product’s market and viability.
particularly useful when launching new products; of course, SoftCorp participants report that internal deployment also provides them the opportunity to provide critical feedback to the product’s development team.

While it is encouraging that participant organizations seem to have found ways to develop IAM requirements and subsequently useful systems, I believe that they are unusually fortunate: both have an in-house development staff and large budgets dedicated to IAM. These approaches are not useful for all organizations — nor do their success answer the question of why requirements definition in this space is particularly difficult to begin with.

**Collaborative business insight**

Having programmers and actual business users in close functional proximity seems to make the iterative loops of IAM development tighter, and thus the overall process faster (and hopefully more accurate). Participant investment banks had large in-house development staffs that coordinated with business departments, and every SoftCorp participant noted the organization’s policy of “eating our own dogfood”. A close relationship is important for buy-in among business users:

“[We] need to make sure that encouraging folks to do security isn’t adding stress or additional information to their day. Even if you try to squeeze them, force them, fire them, you just extend the state. So, we take on the burden and build things, bootstrap it and prove it, then when folks are ready to take it on, they will be able to do so.”

This relationship can essentially allow an infosec team to anticipate the needs of its users, and would seem critical to the build-it-and-they-will-come approach adopted by participants:

“The same way when we built the single sign on system nobody wanted
single sign on. Now most of my demand from my resources are for new single sign on instances.”

My own experiences working at a large software company (not the same as SoftCorp of this study) have led me to believe that “dogfooding” is perhaps the single most valuable source of information on how customers will use and perceive a software product. Time and again I have watched teams of world-class engineers, designers, and product managers have their assumptions refined rapidly (sometimes brutally) by internal-user feedback. This process is easier when the product is broadly targeted but being dogfooeded by users who are themselves relatively technical, because they are willing, able, and numerous enough to provide copious feedback. Products targeted at non-technical users or at smaller populations of specialist users may be less likely to benefit by a widespread, open-ended dogfooding process as it is practiced at this company. However, with some additional effort, I believe the same principle of early experimental testing among real target users should still be useful in the development of most any software product.

6.3 Entitlement review and the IAM lifecycle

Participants often discussed entitlement management in terms of employee “lifecycle” events, such as initial hiring, promotion or transfer, and eventual termination. This perspective lends itself well to understanding IAM as a process: most actions in the process are triggered by a lifecycle event. One important IAM activity that does not fit into this schema, however, is entitlement review, which is mandated by Sarbanes-Oxley (SOx) in the United States, the Eighth Directive in Europe, and similar laws in other locales.

The study’s primary participant bank, P-Bank, was in the process of completing one of its first mandatory entitlement reviews during the period I conducted inter-
<table>
<thead>
<tr>
<th>Date</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>February</td>
<td>Work began on entitlement-review policies</td>
</tr>
<tr>
<td>April</td>
<td>Solicited feedback from users via questionnaires</td>
</tr>
<tr>
<td>June</td>
<td>Began building review trees</td>
</tr>
<tr>
<td>August</td>
<td>Entitlement snapshot taken</td>
</tr>
<tr>
<td>December</td>
<td>Corporate study interviews conducted</td>
</tr>
</tbody>
</table>

**Figure 6.3:** A timeline of the entitlement review as reported by P-Bank participants.

views there, and the rest of this chapter contains detailed information participants reported about that experience. The organization has likely completed an additional five entitlement reviews between that period and this writing, and should not be interpreted as being representative of an investment bank’s current practices; despite this, I believe this information offers important insight and perspective on the review process and the organization’s state before reviews were required.

### 6.3.1 Entitlement review timeline

The laws mandating entitlement reviews provide organizations with a fair amount of latitude in determining the specifics of the process, (e.g., the frequency of the reviews). P-Bank had chosen to conduct entitlement reviews on an annual basis. It had identified 200 applications and databases as containing data to which SOx applied, and which were thus part of the entitlement review.

P-Bank brought in external consultants to conduct the review. The consultants started working on policies to govern the review in February of 2006, started soliciting feedback from users via questionnaires in April, and started building review trees (mappings of employees to supervisors who would review their entitlements) in June. They took a snapshot of actual entitlement assignments at the end of August, and interviews for this study took place in December. The organization had originally scheduled a “drop-dead” date for November 15th, but they eventually decided to abandon it; especially with employee absences due to the holidays season, they feared that hard stop date would make the review “too messy”.

118
6.3.2 Entitlement review results

The entitlement review resulted in a fifteen percent reduction of application-level access entitlements across the 200 applications reviewed.

Managers who did not finish their reviews usually did not even start them; at the time of interviews, about fifty percent of P-Bank employees had actually participated.

One important production department (which made up approximately 30% of the firm’s population) did not participate at all; they said that they were “too busy”. One participant told us that there was a “subplot” to this occurrence: managers from the department in question had argued that “If we catch them on the way in, we don’t need to review them”. Our participant noted that those managers had

“... missed an opportunity to reduce risk today. While they were thinking of the long term, this deals with the risk today, and makes people responsible for understanding it today.”

Participants reported that the biggest challenge of the entitlement review was figuring out supervisor relationships:

“[Internal provisioning application] is there to address one item. Before, anyone could submit a request for app access, and there was no defined approval process. Yes, someone might have approved it, but there was no definite audit history. [...] In many other systems, users select their manager from the list of active users, and then that designated person can approve the user’s access. [...] This is one of the direct manager issues, you don’t know who your manager is, so we can’t easily say who it is. [...] It would be interesting to know if other firms have this same problem, or if they have more clearly identified managers that make this easier.”

Of the users whose access was broken, about one hundred were by the same manager, who had “disconnected them all from everything”.

119
Some app managers were pleasantly surprised at the results of the entitlement review, and reported being able to remove entitlements or users despite an earlier concerted effort to manage provisioning carefully.

While looking at the review results, we noted (and infosec team members affirmed) that many managers in the firm seemed to have over one thousand entitlements to review in total. Many departments’ users had an average of over 100 entitlements.

6.3.3 Infosec team observations

Participants offered a variety of observations on their experiences with the entitlement review process.

Success through broad support  Participants noted that the entitlement review came about not through executive mandate, but because members of the infosec team were able to convince a variety of stakeholders that it would be a valuable exercise: “There are market forces at play internally. It’s not all top down. [participant] and [participant] did a selling job”.

Deciphering entitlement descriptions  One problem that infosec team members noted was that non-technical users had a hard time deciphering the description applied to each entitlement. The importance of entitlement descriptions had been discussed in the past, and the organization had made a policy that

“When a new entitlement is added, it’s the business owner that runs the app that is responsible for a meaningful entitlement description.”

Business-understandable entitlement structure  Infosec team members also wanted to “Consider having business folks fill in the business role field, so it says what they do in their jobs, not what a technologist would put in”. One participant
suggested that it would be helpful if the app descriptions said “Only folks in this
department should have access to this,” or that “this app belongs to this division”.

“It would make much more sense to have reviewers be checking the state-
ment, ‘Alice is a salesperson’ instead of ‘Alice can do X and Y.’ ”

**Duration and organizational change** The entitlement snapshot took place in
August. One participant lamented “here we are in December. Much of the data has
changed.” An informal check revealed entitlement changes for 158 users in period of
just a few weeks, which caused uniform dismay among participants.

“If we had managed to get the thing done in three weeks, we wouldn’t
have to worry so much about people transferring, leaving, being hired,
etc. It’s a policy issue, and we have to get [policy specialist] in on it.”

**6.3.4 The next go-round**

Throughout our discussions, participants offered observations relevant to future in-
stances of the entitlement review effort.

**Automated deprovisioning** One alternative approach to reducing over-entitlement
could be to automatically “expire” an entitlement if the user has not exercised it re-
cently. When queried about the idea, participants stated that they had considered it,
but decided not to pursue it. “If you’re supposed to have the entitlement, you should
have it whether you’ve used it recently or not.” They gave the example of a user
is the backup administrator for an application who only exercised the entitlements
associated with this permission on rare occasions, when the other users listed as pri-
mary administrators for the application were out sick or on vacation. The participant
noted that “you don’t want her to lose access to that app in a month, especially since
when she needs to use it, she *really* needs to use it!”
**Automation**  We asked participants if it would be possible to use their recent experience to automate the entitlement review process. They said that for some aspects, yes: they were meticulous about keeping records of the whole process so they can deconstruct it later. “But, the source [review] trees haven’t been corrected”; they “created local copies of the trees and updated them”. This means that copies of the trees (the definitions of who reports to whom, and what permissions everyone has) must be manually reconciled into production systems. Automation would require additional work to streamline this process, or that the review itself occur on the continuously-evolving production tree rather than a static copy.

**System commoditization**  One participant noted that “we do the technical piece” — they make the system available — “we don’t own the business process”. She said that her team was “happy to own the tree as long as someone maintains it and there is a policy around it, not a free-for-all”.

This approach — offloading the business aspects of system management to appropriate business owners — is in line with the previously-discussed emphasis on system commoditization by infosec groups.

**Regulatory reaction**  During the “first year of [Sarbanes-Oxley] audits, they had [internal provision management application], and no one else in the industry had anything, so the auditors were happy. However, we can’t stagnate, because expectations are being raised all the time.” It is fascinating that participants perceived regulators’ expectations as being relative (how well the organization performed in comparison to its peers) rather than absolute (how well the organization actually protected its users’ interests).
6.4 Conclusions

Real-work technology work is messy and complicated. An organization cannot always just scrap old systems in favor of new ones. They cannot always find a vendor who sells the products they want, and even if they can, that does not mean the product is going to be compatible with the other systems the organization depends on. Other complications (i.e., acquisition of a new company) only add to the complexity of the problem. Academics often talk about a system in isolation: clean, fresh, and not-yet deployed — but in reality every such systems must find their place in an organization that is a delicately-balanced ecosystem of old and new, flexible and brittle, effective and suboptimal solutions.

Efforts to deploy software as part of an information-security initiative face an additional challenge; such initiatives cost money, but it is not always readily apparent how they will help an organization save or make money on that investment. The value they add— protection from data breaches — is realized when something does not happen. This makes it difficult for an information security group to get support for its initiatives from executives and leaders in other parts of the organization.

Even if that group has the wherewithal to manage its technical ecosystem and the financial and logistical support of the organization’s management, large corporate organizations face challenges in understanding the requirements that their systems should satisfy. This extends from basic self-awareness — who owns this system? — to deeper questions of functionality — in this scenario, how should the system behave?

This level of awareness cannot be achieved in a single sitting, but must be grown over time, and be able to evolve as the organization evolves. For this reason I believe that the heart of access control is an understanding that identity and access management cannot be achieved by deploying one or more products, but that it is the result of multiple socio-technical processes rooted in the organization’s core business. By approaching IAM with process-based solutions, the organization has the possibility of
accommodating its changing structure, its developing self-understanding, the evolving capabilities of technology, the expectations of business users, and the adapting external regulatory requirements. As is the case with the entitlement review, an organization can work over time to refine and automate its processes and thus reduce the work burden on human users, but in many cases it will not be possible to remove humans altogether, because the flexibility of human reasoning is the very thing that lends strength to the process.

While viewing access control as a process that requires constant care and feeding may not be attractive to financially-conscious corporate executives (or academics seeking provably secure, static solutions), I believe it is the only realistic way for a large organization to meet its access-control goals in the context of its messy, fast-changing environment.
Chapter 7

Clinical study design and results

7.1 Introduction

After documenting and cataloguing problems reported by information security professionals in the Corporate Study, I wanted to continue this line of research in a way that would allow me to better understand the users’ perspective. I was curious: were users aware that some of their actions worked against the organization? Did they perceive the cultural and technical factors I identified as impacting the organization’s identity and access management efforts? Could I find a way to share their experience in a way that would be useful to the technologists who design the systems that hold such sway over their daily professional lives?

It was difficult to arrange interviews with end users of the financial and software organizations we partnered with. This was in part a problem of logistics — identifying users, getting approval from their managers, finding time on their calendars — but I believe also a product of our partners’ generally circumspect approach to collaboration. An organization could be reasonably confident that its trained infosec professionals would not reveal information that would embarrass the company, but the sorts of users who were likely to share interesting anecdotes with us also had the
potential to let slip information that the organization might rather not get out. Although we had rigorous confidentiality agreements that prevented us from revealing any uniquely identifying information about participant organizations, I detected a degree of hesitancy in exposing us to information that they believed could be truly damaging to their reputations, particularly on the part of the investment banks. (I conjecture that this hesitance was due at least in part to worries about reputation risk, as discussed in Chapter 6.)

Happily, we were able to establish a relationship with an organization in the healthcare industry — specifically, a teaching hospital (details below) — that was willing to let us interview end users directly. Perhaps this organization was more open than corporate participants because it had greater confidence that its employees were doing the right things, or perhaps its intimate experience with deriving value from empirical research into human behavior helped it perceive that allowing us access could have wide-reaching benefits. In any case, this study generated observations that both support and supplement those I have discussed in prior chapters. In this chapter I touch on the observations that overlap, but focus principally on the new insights.

7.2 The study

My approach to the Clinical Study had a much stronger ethnographic flavor than the Corporate Study. This was in part because I had to engage emotionally with participants in order to convince them to trust me while observing them; if I did not, my presence made them visibly uncomfortable.

The results from this study are fewer than from the corporate study. This is in part because the previous study involved a larger number of organizations, and because the data gleaned from talking to IT professionals who deal extensively with
access control can provide a variety of insights more quickly than can be uncovered by direct observation of end users. However, I think that the access-control solution used by partner organizations is the largest determining factor: because the clinical partner puts comparatively few restrictions on users’ access to its systems, it faces fewer challenges in the technical management of access-control policies.

7.2.1 Teaching hospital

I refer to the healthcare organization that partnered with us in this research as Teaching Hospital; it is a tertiary care facility that has under 10,000 employees.

I conducted semi-structured interviews with a total of seven technical staff members (two of whom are also practicing physicians) and directly observed approximately twenty clinicians in their working environments. My observations were limited to activities not at the point of care (i.e., when clinicians were not interacting directly with patients), and focused on their use of computer systems. Of the twenty clinicians, approximately four were physicians (including residents), one a nurse practitioner, two medical students, and the remainder were nurses or other clinical workers. I was also able to conduct semi-structured interviews lasting more than fifteen minutes with approximately five of the participants I observed. (I categorize shorter interviews as part of observation.)

Participant recruitment  My sampling methodology — the process I used to enlist participants — is sometimes described in social science research as “snowball sampling”: new subjects are suggested (and sometimes directly recruited) from among the acquaintances of existing subjects. If I were to conduct further research at Teaching Hospital, there are a small number of participants whom I could clearly identify as potential snowball “cores” for further sampling. As it was, my initial set of participants were drawn from clinicians serving on an IT steering committee.
Data-gathering For about half of my interactions with clinical users I worked by myself, and relied on my fieldnotes for data. For the other half of interactions I was accompanied by a fellow graduate student or an undergraduate research assistant, and used their fieldnotes for data in addition to my own. One observation (of an IT professional conducting a compliance check on workstations spread throughout one wing of the hospital) was conducted by an undergraduate intern working under my supervision.

Software training I was also able to sit in on a three-hour training session that new employees attend to become acquainted with Teaching Hospital’s electronic medical record system, which I refer to hereafter as Teaching Hospital Electronic Record System (THERS).

7.3 Background observations

Structure Teaching Hospital is an academic medical center that includes a number of specialty medical clinics as well as a multi-specialty physician group practice. IT operations are overseen by a group of technical professionals and medical doctors. Each department or clinic operates with a fair degree of autonomy, although IT operations are centralized.

The medical cottage industry Domain experts from other institutions and participants from Teaching Hospital have described healthcare as a “cottage industry” in which each institution has its own distinct culture and practices. As one domain expert who had extensive experience digitizing another institution’s medical records said, “Once you know one academic medical center... you know exactly one academic medical center”. The number of processes and practices that are so institution-specific makes it hard to transfer personnel or knowledge from one organization to another;
this divide also makes it difficult for information security professionals to directly apply lessons learned in one place to another.

**Supremacy of patient care** Above all, Teaching Hospital values the safety of its patients and the quality of care it provides to them. This focus shapes many of the decisions they make.

**Data permanence** As one participant noted, a clinical organization “can’t ever throw information out”. Medical records, including electronic ones, are legal documents. If a clinician makes an error an amendment is added, but the error is preserved as part of the record.

### 7.3.1 Technical topics

Below is a sample of technical topics that came up during interviews with infosec professionals at Teaching Hospital.

1. Authentication systems, including passwords\(^1\) and biometrics

2. Deauthentication (making sure users log out when they are done with a computer)

3. Transferring patient data from old computer systems to new ones

4. Technical initiatives inspired by new regulatory requirements, including the encryption of all Teaching Hospital laptop hard drives

5. The role and challenges of new devices in the medical setting, including iPads and smart medical machines

\(^1\)Appendix A contains a brief discussion of password policies that was relevant to Teaching Hospital during its examination of its own password policy.
7.4 Results: user perception and experience

Participants in the Clinical Study use computers every day as part of their jobs. Teaching Hospital has migrated the majority of its record-keeping to the computer; as its central electronic medical record (EMR) system, THERS is used in some capacity by every clinician in the organization. Individual departments within Teaching Hospital also have specialty computer systems for handling inpatient data, obtaining medical images, and other tasks that are outside the scope of THERS.

Even though participants make regular use of computers as part of their jobs, the vast majority professed a degree of discomfort or perceived incompetence with the machines: in first being introduced to me and my research, they would invariably say something along the lines of “Oh, you shouldn’t talk to me, because I don’t know anything about computers!” or “Don’t ask me, I can barely get them to do what I want!” It was hard to sort out whether this reaction was actually a product of their experiences with the computer, a response to being introduced to me as “computer security researcher,” or simply a reluctance to take time away from their already-full professional schedules to talk to me. In any case, I often responded by saying the equivalent of, “Oh, don’t worry, I don’t need to ask you any technical questions. I’m trying to make computer systems more secure while also making them easier to use, so I’m actually looking for people to tell me about what annoys them or makes it hard to get work done”. Participants usually responded in turn with an enthusiastic or humorous attitude, and comments to the effect of “Well, I can certainly do that!” or “I’ve got enough to fill that whole notebook of yours!”

It is not surprising that most clinical participants seem less comfortable with computer systems than I do; after all, my work is focused on studying such systems, and computers have been central to my life for much of my life — so my net experience is greater than that of participants even ten or twenty years my senior. In observing their discomfort, however, I do not seek to paint them as bumbling technophobes;
in fact, most expressed that they enjoyed using computers in their personal lives. I believe that their discomfort with computers in the workplace is largely a product of fear: the fear of doing “the wrong thing,” of irrevocably messing up a critical clinical system, and — most immediately — the fear of taking an action that will make it hard for them to keep up the frenetic pace their professional environment demands.

I believe that this low-level fear is important to understanding many user behaviors, and particularly relevant in studying users’ interaction with identity and access management systems: after all, access-control mechanisms regularly “challenge” users by asking them to prove who they are, and when a user answers a challenge incorrectly, she is locked out of everything she needs to get her job done.

The rest of this section explores some of my observations on about clinical participants’ beliefs and experiences, whereas Section 7.5 considers some of the participant behaviors I observed surrounding identity and access management.

7.4.1 Relating to the machine

I observed one nurse manager in a small department for a period of about two hours: she spent the whole time at her desk, either talking on the phone with patients, using her computer, or doing both simultaneously. Although she did not employ keyboard shortcuts, her actions in the programs she used frequently were as quick and confident as those of any seasoned system administrator; it was clear that the volume of her interaction had brought her a certain degree of mastery.

Despite this apparent confidence, she expressed a high level of anxiety about any possible changes to the computer systems. In discussing updates to the interface of THERS, she exclaimed,

“When anything — even the colors — is changed, it freaks you out!”

Now, this participant was not inherently opposed to changes in the system, espe-
cially if she understood how the change could make it easier to do her job. At the same time, she depended heavily on being able to interact quickly with the IT, and dreaded the possibility that something would confuse her and slow her down.

While the nurse manager had a concrete understanding of the capabilities and behavior of the computer systems, other participants’ beliefs seemed to have been shaped by trial and error, and had the potential to cause significant unnecessary strife. One nurse who worked with medical images had to reboot her computer after a crash, and was frustrated because the other workstations in the area were already occupied. Normally, she said, she would use a different computer that was already running so she did not have to start up particular software program again. Although she had been using this program since early that morning on the machine in question, she anticipated that it would take a long time for the program to get going, because so many other people in Teaching Hospital were using it. She explained that “the system” slowed down the closer to noon you got, and if you tried to launch it after noon you were practically guaranteed to be locked out. When I asked her how she knew this, she related that this had been her exact experience when she was first using the program, but that she had avoided having to log on any time after 10:00 since. She was surprised when the program started up quickly, and I wondered how many times she had changed workstations because she had taken one negative experience to be indicative of the general case.

One participant — a doctor completing her residency — lamented the fact that she could not use her personal Apple computer to log in to THERS. She reported that residents are often responsible for entering notes into patient records, and felt that home access to THERS would allow her to both more happily complete her work

---

2When she lamented that her limited screen real estate made it difficult to use many programs at once (i.e., when coordinating the schedules of multiple providers while also entering data into THERS) I mentioned that many computer programmers use multiple screens at once. She jumped on the idea, and said that she would love to experiment with such a setup, but that since her department was not a high priority in the organization, she doubted she could convince the IT staff to go for it.
in the evenings and prepare in the morning for the upcoming day’s work. Strong personal preferences about computer systems are likely to become more common as the next generation of medical professionals enter their field. I wonder what impact their experience (and increased comfort) with computers at an early age will have on the role and expectations for clinical systems in the future.

7.4.2 The technology ecosystem

Participants frequently expressed frustration with aspects of computer systems that were close to working, but that failed in small or irregular ways. For example, THERS and department-specific systems were generally good about sharing data, but one program periodically failed to update THERS, particularly if a provider quit out of the program without explicitly closing a patient’s record first. Practitioners waiting for the data to appear in THERS could be stalled unnecessarily — and when they realized the problem, said they could not blame anyone other than the computer system.

Participants described another frustrating feature in the mechanism by which external lab results were imported into THERS. While it was convenient to have faxes scanned and uploaded to the appropriate patient record, all entries were labeled with a date and the generic title “Lab Result”. When a practitioner wanted to see the results of an external blood test taken the week before, she had to potentially choose from a potentially long column of generic “Lab Result” entries in the patient’s record.

An obvious source of assistance for users experiencing problems with computer systems is the organization’s helpdesk. However, one participant reported significant frustration even in dealing with helpdesk staff: she said that she had a favorite helpdesker, and that if anyone other than this favorite answered the phone, she would hang up immediately (only to try again a few minutes later). Given how much organizations rely on helpdesk metrics to gauge whether users are successful in getting
their jobs done with the software systems, this behavior — whether due to the user’s computer anxiety, her perception of the helpdesk staff’s general incompetence, or some other factor — would seem to be particularly problematic.

7.4.3 Of and for the patients

In the course of this study I came to believe that medical practitioners define and measure nearly every aspect of their profession in terms of their patients. In expressing a frustration, participants often grounded their complaint in its impact on patient care; in praising a new computer feature, they often explained how it made the lives of their patients better. Even though they are not computer experts, their wealth of experience dealing with patients provides them with a better understanding than any computer professional of how computer systems impact clinical settings. In discussing electronic medical records, one practitioner said that she was enthusiastic to see them more widely adopted. She noted that while some people believed that using computers in exam rooms would introduce distance between providers and patients, she found that the computers gave her the ability to share information — particularly charts and images — quickly, easily, and intuitively, which actually brought her closer to her patients. Once she said it, this made perfect sense — but it was a point that had not occurred to me all the times I’d heard the subject discussed.

Sometimes the overwhelming focus on patients added unusual constraints to participants’ patterns of computer usage. I observed the nurses in one department carefully negotiating and planning which laptop to use as a new round of patients were scheduled to arrive. When I asked why it was an issue — were not all the laptops the same? — they said that only one laptop had a music program installed on it. At first I assumed that they were listening to the music for their own amusement, and questioned their priorities to myself. They then explained that there was one long and uncomfortable diagnostic procedure that some of their patients had to go
through, and that they liked being able to offer music to help take their patients’ minds off the experience.

**Patient access to data**

As a computer scientist who takes an interest in the usability of security systems, I thought I had a good basic grasp of the needs and motivations that drive users in any environment. The insights that clinicians can offer about their computer usage — more importantly, the insights that stem from their fundamental and primary mission of patient care, which I did not understand but that they take for granted — surprised me time and again.

At the same time, some participating physicians made assertions about their patients that I would like to see studied further. In particular, multiple physicians expressed concern about the notion of providing a patient with the access logs of her medical record; these physicians, who were well versed in the structure of the computer system and the organization’s clinical, research, and billing processes, argued that a patient would not understand that many hospital employees legitimately need access to patient data in order for the patient to be seen, receive care, and have that care paid for. A patient without this understanding, they argued, would be alarmed to see access logs containing so many unfamiliar names. (The patient might expect to see the name of her doctor, but not of the billing clerk who sent paperwork to her insurance company.) While I agree that this data by itself could be alarming to many patients, I wonder whether it might be possible for a hospital to provide additional context to help patients understand what they are reading.

The subject of patients’ understanding of access logs came up in a discussion of how the organization handles cases where a patient is concerned that her record is being improperly accessed, e.g., by an ex-spouse who happens to work at the hospital. The current process for auditing access logs, which can be initiated by the
institution or by a patient if she has concerns, is conducted entirely in-house; at the end, the patient is given a simple yes-or-no answer depending on whether the internal auditor determined that inappropriate access had occurred. While this solution can be an effective way to address patients’ concerns in such a situation, it is neither particularly transparent (thus, some patients might wonder whether a hospital was hiding something to save face) nor scalable if the number of requesting patients grows large. Providing users with the access logs of their records would, on the other hand, be transparent and could scale better. If a patient suspects that an acquaintance is improperly accessing her medical record, she is likely to be better qualified to perform an audit, simply because she is the only one familiar with all of her acquaintances. In fact, this is the exact process that employees of the organization can use to audit access to their own medical record if they receive care from their employer. While I understand physicians’ desire to protect their patients from unnecessary confusion and concern, I worry that this position could, without further data to support it, be viewed someday soon as antiquated and somewhat paternalistic.

7.5 Results: compliance and circumvention

In comparison to participants in the Corporate Study, employees at Teaching Hospital seemed hesitant in extolling the role of security technologies while discussing their organizations. At the outset of the Clinical Study, one physician expressed skepticism bordering on defensiveness; he said that clinicians in Teaching Hospital were

“... highly interested in security... but not when it prevents us from getting our jobs done.”

Although participants believed information security was important, and something the organization should strive for, their experience indicated that it was very hard
to make systems that were both usable and secure. One infosec professional stated, with both frustration and chagrin, that

“Many of our clinicians work around [the security software, rather] than with it.”

The culture of Teaching Hospital differed from that of the Corporate Study organizations, especially P-Bank. Bankers were quick to speak up when they had a problem with a computer system, and often went so far as to blame the system and its administrators from reducing their productivity. As Section 7.4 notes, clinical participants often seemed afraid that they would make a mistake with computer systems. Even when users in clinical organizations were confident in their understanding of the system, they seemed less likely to vocalize their dissatisfaction through official channels; these users were perhaps more likely to vent their frustration in a way that resulted in unexpected confrontation or escalation. More often than not, though, users quietly did what they needed to do to get their jobs done.

During one observation I watched a clinician trying to log on to a system she rarely used retrieve her password from a binder in a nearby cupboard. She saw me watching, and told me that

“I know we’re not supposed to write them down, but if they want us to get in...”

She recognized that what she was doing was against organization policy, but I got the impression that if I had not been there she would not have thought twice about it.

7.5.1 Under-entitlement

Although participants from Teaching Hospital do not describe under-entitlement using the same terminology as participants in the Corporate Study, they recognize the
problems it poses. Indeed, given that timely access to medical data can allow clinicians to save lives, they consider avoiding under-entitlement to be of paramount importance.

In describing their goals for future security systems, one infosec professional said that they “want to increase control without losing flexibility”. The number of programmatic controls present in THERS was minimal in comparison to the systems discussed by participants of the Corporate Study. This surprised me: I expected (perhaps because I have read so many access-control papers that use medical examples to motivate the need for the new access-control models they describe) that access to an individual patient’s records would at least be restricted to the set of clinicians working to provide care to that patient. However, as discussed above, the number of hospital staff members who legitimately need access to medical data in the course of providing care is large, and includes a significant number of non-clinicians. Participants reported that these non-clinical users’ access was carefully restricted to a small subset of patient data using a classic RBAC scheme. (Note that I did not have the opportunity to interview, for example, an administrator tasked with billing insurance providers, so have no data on whether these users experienced under-entitlement or circumvented controls to get their jobs done.)

**Break-glass protection**

Participants also reported that restricting clinicians’ access to patient data without compromising patient care had proven to be unsolvable with the available tools. THERS thus implemented break-glass controls, which guarantee clinicians’ ability to view patient data without intervention by a third party (e.g., an administrator). Break-glass was implemented in the following cases:

1. For records containing sensitive information, such as HIV test results,

2. For records belonging to high-profile patients, such as celebrities,
3. For other records, determined by individual clinicians at their discretion.

Records in THERS were visually marked when under the protection of a break-glass control. Users who attempt to access a protected record encounter a dialog that 1) Notifies them the record is under break-glass protection, 2) reminds them that their access will be logged and responsible clinicians will be notified, and 3) Asks them to confirm that they want to proceed with the access.

Control challenges

Participants of this study reported that their organization experienced challenges in deploying more proactive controls for some of the same reasons reported in the Corporate Study. One basic criterion for controlling access to patient records — whether a clinician has a care relationship with a patient — is challenged as in finance by the tremendous churn that hospitals experience in staff assignments. Several classes of users change departments as a normal course of their work, including medical students rotating through departments as part of their training, and residents in specialties such as anesthesia which require them to roam from one inpatient department to another while on call. Even outside of teaching hospitals, “floating” nurses can change assignments daily, physicians are frequently asked to consult or provide a second opinion on one others’ cases, and all clinicians establish temporary care relationships with patients in a colleague’s absence.

Participants identified additional properties of the healthcare environment that would challenge an effort to control access to subsections of a patient’s medical record. Whereas a database — a resource commonly under access control in the financial...

---

3I unfortunately did not discuss further details on the break-glass feature during interviews with participants, and thus do not know answers to several questions that one might ask about its implementation. For example, if a user chooses to access a protected record, will she be prompted to break the glass upon future access attempts? When a clinician adds break-glass protection as in case 3, can he provide a whitelist of users who should not be prompted to break the glass? Aside from the physician who imposes protection in case 3, who is notified when the glass is broken, and can that list be specified or changed?
industry — is organized according to a well-understood schema, clinical data is often highly unstructured. More significantly, it is difficult to know ahead of time whether particular information from the patient’s record is going to be relevant in future situations. We might not expect that the notes from Jane’s podiatrist five years ago will help her neurologist diagnose her condition today, but participants report that the availability of any piece of data may be critical important in providing effective care.

7.5.2 Deauthentication and styrofoam cups

While the infosec team of Teaching Hospital largely perceived themselves successful in providing usable and secure authentication, they had identified the problem of deauthentication: how do you make sure people log out of a computer system when they are done using it? Here as in the high-stress financial environment mentioned in Chapter 5, the standard approach of applying inactivity timeouts was met with strong user opposition. For every timeout value the infosec team tried, it seemed like there were departments for whom the value was too short (users complained that they were logged out while still using the system) and others for which the same timeout value was too long (compliance checks in revealed logged-in computers that were left unattended).

At the same time, the nature of the healthcare setting left workstations exposed to large numbers of people not employed by the organization — and made it harder for employees to clearly identify strangers in their work environment. (The number of people regularly seen on an individual floor of a financial institution is easily both smaller and more stable than the crowds of patients and clinicians coursing through a hospital department in any given day.) Although financial organizations have stricter

---

4Participants reported that the large amount of unstructured data is a significant barrier in reaping the benefits expected from electronic medical records, but that clinicians in the past had ignored or actively fought efforts to impose increased structure on their data-collection process.
timeout policies, reliable de-authentication would thus seem to be more important in healthcare, where it can be harder to spot an opportunistic outsider.

**Proximity sensors**

In an effort to solve the deauthentication problem, the infosec team of Teaching Hospital deployed to a set of its workstations a special sensor designed to detect the close proximity of a human. Thus, when a user walked away the proximity sensor could detect her departure and sign her out if necessary. Like timeouts, this approach was met with widespread user complaints⁵; many said the sensors logged them out while they were still standing there, or were confused by people walking down the hall. Eventually some clever user realized that the sensors had no minimum detection distance, and thwarted one by putting a styrofoam cup over it. The solution spread throughout the organization, and shortly thereafter the infosec team removed the devices.⁶

**On closing tickets**

Both clinical and IT participants reported that the infosec team encouraged users to deauthenticate by closing their active Kerberos ticket⁷. Because the organization used Kerberos to mediate access to all its sensitive resources (electronic medical records, email, and department-specific systems), closing the ticket should, theoretically, be the best way to de-authenticate quickly. However, as one savvy clinical user pointed out (and others confirmed later), users who comply with this exact guidance can actually experience more frustration than users who de-authenticate from each indi-

---

⁵ Participants practically shuddered remembering the feedback they received during this experiment. It seems that users were vitriolic and quite aggressive in expressing their discontent — to the point that at least one device was literally ripped off of the workstation to which it was attached.

⁶ At least, the ones that angry users had not already broken.

⁷ Kerberos tickets manifested on the workstation’s screen as a small window containing the authenticated user’s name. Some users chose to minimize it, as it stayed in the foreground and they found it got in the way. It could be closed like any other window with a standard mouse click to the appropriate corner.
vidual application (i.e., using a Logout menu or button). As the observant clinical participant demonstrated to me, if user \( u \) terminates her session in THERS by some mechanism other than that application’s native logout flow, user \( n + 1 \) must wait while THERS restarts — a process that can take multiple minutes — before she can log in. Thus, at least in some departments, users exerted negative social pressure on their peers who followed the IT department’s deauthentication advice. I do not know how much this pressure impacted users’ deauthentication habits — are there some who gave up trying to authenticate in exasperation? — but I did observe cases of users who viewed this mismatch between official advice and practical realities as evidence that the IT department did not really know what it was talking about.

7.5.3 Compliance checks

Following the removal of the proximity sensors described above, the hospital staff decided to pursue a solution to the deauthentication problem that was not rooted in technology, but based instead on user education and periodic compliance audits. An intern who collaborated with me in this research joined a member of the hospital’s infosec staff in performing one such compliance audit; over a period of a few hours they checked roughly 225 workstations to identify ones that had been left unattended while still possessing a valid Kerberos ticket.

The compliance check was conducted as follows:

1. The staff member had a list of physical regions of the hospital he was going to target and a time period in which he was going to conduct the check. This allowed him to achieve systematic coverage in his audits over time.

2. The staff member walked around the designated region of the hospital. For each computer, he marked down its state:

   (a) in use by an employee
(b) unattended without a ticket open

(c) unattended and with a ticket open

3. For each machine that has been left unattended with an open ticket, he performed the following steps:

   I. Sent an email to the user from their own email account with a standardized message informing them of the situation and offering guidance for the future.

   II. Logged the user off (closed the Kerberos ticket).

   III. Used the workstation’s web browser to open to an internal webpage that explained why the unattended ticket was closed.

   IV. Placed a pen light printed with the message “Log off before you walk off / Thank you for protecting patient information” at the workstation.

The auditor’s definition of “in use” seemed a bit loose, in that machines with open tickets were considered to as such if there were clinicians in the same hallway — even if those clinicians were not directly interacting with machines. This situation represents one core of the organization’s de-authentication issue: throughout my observations, it became clear that clinicians frequently alternate between interacting with a stationary computer and performing non-computer tasks. A clinician believes that she is still \textit{using} the computer while performing other tasks, because she intends to come back to it within a matter of minutes. Similarly, the frequency with which she must alternate between computer and non-computer tasks is precisely what drives her to consider logging out and then logging back in again to be so burdensome.

Given this context, it is not surprising that the IT staff member chose to apply a somewhat broad categorization of “in use”: indeed, I think that his choice reflects an understanding of users existing perceptions of that term. Moreover, a more strict
interpretation of the de-authentication policy would be viewed by users as being antithetical to their goal of providing efficient patient care.

The intern who observed the compliance check noted the infosec staffer seemed highly conscious of clinicians’ perceptions and opinions in general, to the point that the intern noted him to be “on a bit of a PR mission”. The use of pen-light giveaways\(^8\) is consistent with this attention to user perception, and indicates a recognition that improving user de-authentication habits is difficult if one simultaneously inspires hatred for the IT department.

### 7.5.4 Compliance budgets

The compliance behavior that is documented here supports Beaumont et al.’s theory of *compliance budgets* (which is discussed in Chapter 2) [4]. As those authors observed of their study participants, the users of Teaching Hospital “value security, both for themselves and for the organization they work for”. At the same time, participants chose whether to comply with deauthentication policies based on the costs and benefits they perceived as being associated with that compliance — and we can tell from their behavior that they perceived differences in interacting with break-glass controls, reacting to timeouts vs. manual compliance checks, and using proximity sensors (which quickly maxed out their compliance budgets and inspired physical expressions of their frustration). Like Beaumont’s participants, hospital users expressed that “their primary work task dominates their perspective whereas the security goals of the organization are subordinate”: at a high level, the primacy of patient care dictates that under-entitlement is unacceptable, even if the only workable solution results in probable over-entitlement.

\(^8\)The staffer reported that they used to give out stress balls, but that “people threw them at” him in their frustration.
7.6 Results: the Phalanx Problem (or: bands of roving doctors)

In discussing access control — particularly authentication — with participants from Teaching Hospital, they brought up a number of usage scenarios that seemed to push the bounds of what current technology can cope with. As described above, they struggled with user deauthentication: neither inactivity timeouts nor the proximity sensors they deployed seemed to be good solutions; biometrics were appealing, but fingerprints were thwarted by medical gloves, facial recognition by surgical masks, voice recognition by noisy clinics, and the iris scans by the high equipment cost. The possibility of using RFID tags to solve deauthentication also came up, but there they raised the question: if you have a group of doctors within several feet of a computer when an authentication request is made, how could a system identify which doctor to authenticate?

While one can imagine technical solutions that address this particular conundrum, the question inspired a discussion about patient rounds, which in a teaching hospital can involve a whole team of doctors, residents, medical students, nurses, and other clinicians moving throughout an inpatient ward. During rounds the team makes its way from patient to patient, discussing each case, taking notes on the patient’s condition, quizzing clinicians-in-training, and generally providing care.

However, the dominant paradigm of human-computer interaction has long been focused on a single user: if there was a single computer being used during rounds to interact with the patient’s electronic medical record, which individual should be the one who authenticates to the EMR? Moreover, why should the system be limited to authenticating and logging the actions of a single user when it is in fact a group of clinicians who are actively participating? In these situations it is usually a resident who records the notes and orders, yet it is the attending physician who is legally
responsible for the group’s decisions and the subsequent care the patient receives. To complicate matters yet further, any clinician participating in rounds may reasonably need to access that patient’s data later, if just to further their professional development; in some sense they are authorized to see it by virtue of the simple fact that they were present for the examination.

As noted above, Teaching Hospital relied on break-glass controls to restrict clinicians’ access to sensitive data. A solution for this challenge of authenticating and dynamically provisioning groups of clinicians — which we have taken to calling the Phalanx Problem\(^9\) — could provide a more advanced authorization system with valuable data about which clinicians have been participating in a particular patient’s care. Indeed, without a mechanism for automatically gathering such data — and automatically provisioning them, or allowing one to “inherit” a subset of another’s entitlements — it would seem impossible to support more finely-grained controls on clinicians’ access to patient data without incurring tremendous administrative costs.

I believe that more time should be spent studying the data-access and data-generation patterns exhibited by clinicians conducting rounds, as well as other dynamic groups of professionals whose actions transcend the current one-user, one-machine authentication paradigm. Chapter 11 goes into a more detailed exploration of potential future work in this vein.

### 7.7 Conclusions

This study lends additional weight to many of the results of the Corporate Study, and offers further insights into the human factors that impact the success of access-control systems. The users of Teaching Hospital possessed a diversity of skill levels, depths of understanding, and degrees of comfort with computer systems, but discussions with a

---

\(^9\)A phalanx was an ancient greek military formation; the phalanges are the collection of bones that collectively form the fingers of a hand.
small set painted a coherent picture of the limitations of current access control technologies and policies in their environment. Although I believe these results support the hypothesis that human factors are at the root of why access-control systems fail, I believe they also indicate that human users cannot be blamed for this failure. The users I interviewed and observed at Teaching Hospital have the best interests of their patients and their organization at heart, and work consistently to comply with infosec policies while meeting the constraints of their professional environments. The failure of access control lies instead with us, the researchers and technologists who design access-control systems and policies that fail to take into account the basic realities of human workflows. It is thus that I propose in the next chapter that we fundamentally reboot our understanding of the access-control problem.
Chapter 8

Old assumptions and new wisdom

Chapters 4 through 7 have documented the findings of my fieldwork studying large organizations in a variety of industries, including a deep and systemic dysfunction in the practice of access control in the real world. In this chapter I attempt to diagnose the root of this dysfunction. Section 8.1, “Old Assumptions”, is a revised version of a text co-authored with my advisor that appeared as “What’s wrong with access control in the real world?” in IEEE Security & Privacy in 2010 [82]. The second section, “New Wisdom”, is new with this document.

8.1 Old assumptions

Access control is a fundamentally hard problem. Analog human systems (e.g., corporations, partnerships, families) use countless mechanisms to encode and enforce standards of resource use. Most of these mechanisms are what computer scientists would consider “informal”: together they form a complex ecosystem of sociological incentives and psychological motivators that are wholly divorced from the (often) provably secure computer protocols our community treasures.

At some point over the decades, the resources moved to electronic settings — and the problem of how to provide the appropriate access control came under the
domain of “computer security”. To gain traction representing these complex policies in formal computer terms, the infosec research community approached the challenge as any good scientist does: first one starts with a simplified model. One assumes that the world is less complex, convinces oneself that one can solve the problem in this simplified world, and then moves on to the complexity of the real world.

However, I worry that somewhere along the way our community forgot that the simplifying model was not the same as the real world. We (as a community) decided to implement the systems based on assumptions our simplifying model made, and forgot to make allowances for real-world messiness. (The academic research community, with its focus on occasionally provably secure systems and new, elaborate schemes for expressing increasingly complex access-control policies, is particularly responsible for this focus on the theoretically “secure”.)

Now, when a physicist’s empirical results deviate from those predicted by her theoretical model, she does not simply ignore the deviation: she mines it for information, because she knows it can help her validate and refine her model. In particular, she does not castigate the universe for being uncooperative or not sufficiently well-trained!

However, when our implemented computer security policies go awry, we often do not seek to understand why. First, we cope with the resulting crisis; next, we argue to interested parties that our system did, in fact, follow all accepted best practices, and that the failure was due to some uncontrollable, external factor, such misbehaving users.

**Organization-wide doublethink**

The result of this collective habit is nothing short of organizational doublethink. Everyone on the inside knows that the systems are not working, but admitting this knowledge in and of itself opens them to liability — the perceived success of the system depends on ignoring the problems.
Professionals from multiple large financial institutions have acknowledged this doublethink off the record; one (whom we refer to as Alice) related one particularly telling anecdote. While undergoing an audit by federal regulatory authorities, Alice’s team worked to demonstrate that the company met the government’s data security requirements; the auditors asked the expected questions, and ticked the expected boxes on their checklists. Alice was elated that the reviewers seemed satisfied her team’s answers — until it dawned on her that those auditors were asking the same easy questions when they evaluated the security of other financial companies, including the banks to whom she trusted her own personal finances. At that moment, she said, she wanted to grab the reviewers and tell them what questions they *should* be asking, to say “Hey, aren’t you curious about this information over here?” Of course, because she knew the organization that employed her did not have satisfying answers to some of these more revealing questions, she restrained her urge.

Clearly, the *actual* success of a security system depends on thinking honestly about it; in this situation, we must first admit the doublethink is doublethink. As a step in that direction, this section tries to enumerate some of the simplifying assumptions the security community has made in its effort to gain traction with the access control problem. For many environments, there seems to be a dramatic and painful mismatch between these simplifying assumptions and reality. I therefore argue in the second half of the chapter that effective security in these environments may require rethinking these assumptions.

### 8.1.1 The nature of policy

First, I offer a few words on what exactly we mean by “policy”. In the classical way of thinking about computer security, we think about *subjects* (the entities that do the acting) and the *objects* (the entities that get acted upon); we draw a matrix with rows for each subject and columns for each object, and fill in the boxes with the
permissions: what is a given subject allowed to do with a given object. (Yes, this is a simplifying model.) In many real-world enterprises, these permissions are called entitlements.

In the real world, the subjects are typically real people and the resources those things they need to use to get their job done. In the computer rendering, these become computer users and programs and data, governed by some type of access-control system. Our community worries about how to craft this system so that it does the right thing — that is, so that it matches the enterprise’s real requirements. Hence, we might start by calling out the implicit assumption that this goal in fact possible.

- **Assumption 1:** There exists a correct access-control policy for every organization.

Maybe such a policy exists. But maybe it does not — or maybe the language we use to render the computerized policy is ineffective at capturing the “it depends” gray areas in the real world.

- **Assumption 2:** The correct policy is human-decidable.

Our community gives enterprises an access-control system with a set of knobs. Even if a “correct” setting of the knobs exists, is it feasible for a human organization identify it? One infosec officer chortled about how computer security researchers believe that it is actually possible for an enterprise to stop what it is doing for two weeks, put everyone in a large room, and work out the policy.

Even weaker versions of this assumption can be problematic.

- **Assumption 3:** The correct policy is human-recognizable: a human can effectively audit a previous decision.
• Assumption 4: The correct policy is human-constructible: a human can say ahead of time whether \((\text{user, action, resource})\) should be allowed.

In the real world, access-control questions often lead to the answer “it depends”. “It depends” requires context. Will the context be available to the parties doing policy creation or auditing? (Our lab has heard of a critical control room door that was password-protected — because it needs to be secure — but has the password written on it — because if there’s an emergency, one must gain entry.)

8.1.2 Organizational structures

I now consider assumptions our community has made about human organizations themselves. The human components of secure systems are foreign territory for many computer scientists; unlike the finite instruction sets that guide the execution of deterministic machines, the principles that govern the action and interaction of a set of people are often beyond the understanding of any one individual.

• Assumption 5: Resources (and therefore policies) are managed centrally.

Our community typically expects centralized control — but reality often shows that the further one is from the action, the less one understands the real issues.

• Assumption 6: A corporate organization is structured like a tree, with a small number of decision-makers at the root and quantities of specialist employees at the leaves. Control and decision-making flows in a deterministic manner, one way, along the edges of this tree.

At a first approximation, modeling the structure of a human organization with a tree is appropriate; management researchers and corporations themselves often use it. However, a tree represents the relationships relevant only to the most formal of decision-making processes; for example, a tree can help understand the mechanism
by which a university crafts and approves its annual budget. In contrast, in many
domains the decisions made by most employees on a day-to-day basis proceed along a
more ad-hoc path; influence is determined more by a person’s effective job role than
their official title, and thus individuals who are not formally vested with power (for
example, administrative assistants) often have a surprising hand in the outcome of
small decisions.

The departure from the traditional tree view of an organization is also mani-
fest in larger sub-organizational units. For example, the clinical arm of a hospital
may theoretically be managed by a Medical Director or other executive to whom
individual departments must answer. However, no individual is qualified to directly
oversee the detailed operations of Radiology and Neurology and Oncology; each de-
partment requires a specialist leaders who can make decisions appropriate for the
work that department does. This distribution of authority (combined with cultural
factors common with a highly-educated workforce) makes the organization’s process
for choosing computer systems or crafting access-control policies deviate significantly
from the tree-shaped representation.

I have occasionally encountered computer security colleagues surprised by this
questioning the existence of a centralized hierarchy for policy making. I suggest such
colleagues contact the business management community, who have been discussing
such a “matrixed” organizational structure for years.

8.1.3 Knowledge of the users

Unfortunately, security professionals are so confident in their belief in the sufficiency
of existing schemes that they at times blame the user when things seem to go wrong.

- Assumption 7: With sufficient training and commitment to their jobs, well-
intentioned users will follow the organization’s formal rules, including information-
security policies.
• **Corollary:** Only ill-intentioned users will circumvent control mechanisms.

In the real world, I have repeatedly found users who circumvent the system not because they are evil, but because they are conscientiously trying to get their jobs done! One medical clinician even asked us “are you trying to build a better policeman, or do you want to help patients?” Inglesant and Sasse documented [41] that unusable password policies alone can decrease user productivity and negatively impact the organization; a recent medical journal article [48] provides a wonderful case study of a clever computerized resource control system — and all the ways that clinicians worked around it in order to save patient lives.

• **Assumption 8:** The possible negative repercussions of over-entitlement are far greater than the possible negative repercussions of under-entitlement.

Our community touts the principle of “least privilege”: the access permitted by a computerized policy should be a tight-as-possible superset of the access required. However, this thinking can easily lead to a computerized system that is too restrictive. Perhaps in national security environments, erring on the side of under-entitlement may make sense — but in the domains I’ve looked at, such errors may result in missed market opportunities and in patient death.

The computer security community laments when passwords are written on sticky notes or under keyboards — but I posit that this phenomenon is often an understandable effort by end-users to tune a suboptimal access-control system. An organization quashes this tuning at its peril; rather than punishing users, we need to develop systems and policies that help them get their jobs done efficiently without exposing sensitive data to passers-by.
8.1.4 Knowledge of the system

Is it even possible for security researchers and practitioners to understand the system? This conundrum is reflected also in the complexity of economic systems (and the economic meltdown that experts are still trying to understand and reverse). For access control to be manageable by humans, we implicitly simplify.

- **Assumption 9:** The information relevant to making access-control decisions (who has what responsibilities) changes slowly.

  In organizations where the tasks are complex and require professional judgement (e.g., clinical settings) or where the nature of the work is highly dynamic (e.g., investment banks, where bankers are constantly reassigned to new accounts), it may be simply not possible to know ahead of time whether a given operation is going to be acceptable. After his initial comments on policy crafting, the infosec officer I alluded to earlier chortled further that, even if a correct policy could be created, members of our community are deluded in thinking that the policy would remain correct for more than a few days.

  We need to adopt a practice of rigorous auditing. We must define boundaries of how much we trust an individual user, and allow them to operate broadly within those boundaries, but honestly evaluate what they are actually doing.

- **Assumption 10:** Supervisors (or even users) know what entitlements individuals legitimately need.

  IT professionals across domains recognize the practice of *copy-paste provisioning*: a manager Carla will admit that when David joined her group, he will clone the entitlements Bob has — since David’s job seems to be similar to Bob’s and Bob can manage to get his done. Neither Carla nor Bob actually know which are the magic combination of permissions, nor are they likely to take valuable time from their actual work to figure it out.
During one organization’s *entitlement review* (an effort to reduce over-entitlement), administrators presented users with a list of their entitlements, and incentivized them to voluntarily give up ones they deemed unnecessary. The administrators were thrilled when users reduced their permissions up to half — but, of course, dismayed when the permission changes went live and the users could not access the data they needed to get their jobs done.

### 8.2 New wisdom applied to old assumptions

Although I offered counter-examples for many of the assumptions presented in the previous section, I do not think we should abandon them entirely. In this section I seek to identify those that can still hold for some organizations, while also developing supplemental or alternative maxims for situations that the old assumptions do not cover. In general, I believe that some of the classic assumptions can hold true in settings that are well-suited to access-control schemes with classic pre-defined policies; in Chapter 9 I discuss alternative schemes.

#### 8.2.1 Fundamental properties of humans and organizations

Some of the assumptions discussed in the previous chapter apply to the very nature of human systems. Here I offer what I believe to be more accurate maxims on the topic, which I hope will shape our understanding in a more constructive way. Note that the order in which I consider the classic assumptions differs from the previous section, but I retain the same numbering scheme here to facilitate cross referencing.

**Assumption 7: User adherence to policy as an expression of intent**

Contrary to Assumption 7, even well-intentioned users will circumvent controls. They truly wish to do the “right” thing, but the system inadvertently rewards them for
circumventing those rules in cases where circumvention helps them get their jobs done. Some of these compliance problems may be addressed with user training, but there are other issues that user education cannot solve. Assumption 7 could only hold if all these issues were resolved and the system was guaranteed to allow legitimate users to always complete their jobs — a standard that no practical system is likely to achieve.

• **Maxim 1:** In the vast majority of cases, the vast majority of users truly want to do what is best for the organization.

• **Maxim 2:** An *unusual* policy violation may be a sign of malicious intent. *Frequent* policy violations are an indication that some aspect of the control system is broken (e.g., that there is an edge-case that the policy-maker did not take into consideration).

**Assumption 6: Organizational structure and decision-making**

Our society commonly uses tree structures to abstract aspects of a human organization’s operations, and this structure may accurately reflect some organizations; Assumption 6 may hold e.g. when applied to the military chain of command. However, the workflows and decision-making processes that financial, healthcare, and software-industry practitioners find challenging — the access-control edge cases and exceptions — are exactly those that do not fit into this rigid hierarchy metaphor.

• **Maxim 3:** Authority and decision-making in a human system can take on a variety of forms.

• **Maxim 4:** Humans are in many cases well-equipped to reason about the arbitrary and dynamic structures associated with human organizations, but computers rarely are.
**Corollary:** Access-control systems that depend exclusively on machine reasoning are fundamentally limited in comparison to systems that also rely on humans.

**Assumptions 1 and 2: The existence of a correct access policy, and its human decidability**

As discussed in the previous chapter, it can be difficult to identify a single person (or job function) within an organization that defines the optimal policy. A teaching hospital comprising diverse departments often struggles to reach consensus among its leadership; while an executive may choose a policy by fiat, the resulting acceptable policy may be far from optimal. Given these complexities, it is meaningless to talk about a “correct” policy for most organizations.

- **Maxim 5:** There may be no “correct” access-control policy, but there will likely be several “acceptable” policies.

- **Maxim 6:** Finding an acceptable access-control policy is the most expensive aspect of designing an access-control system.

Part of the expense inherent to policy development stems from the massive number of access scenarios that might arise in the organization. More bothersome still is the possibility that a policy-writer might effectively *ignore* scenarios. As the number of relevant environmental variables increases, it becomes functionally impossible for a human to anticipate every scenario. Assumption 2 can thus only hold in cases where the number of variables is small enough to guarantee a human can reason through each combination of their values; I expect that no large organization will fit this bill, but small organizations with relatively simple access-control needs might.

- **Maxim 7:** A human cannot always deliver an accurate access-control verdict before the request is made and the context is known.
8.2.2 Scoping the old wisdom and expanding with the new

The remainder of the assumptions hold some truth, at least in at least a subset of environments.

Assumptions 3 and 10: The recognizability of a correct access-control policy, and supervisors’ understanding of employees’ access needs

Even though my fieldwork indicates that large organizations’ optimal access-control policies are not human-decidable, they may be human-recognizable — i.e., it may be possible for a human to determine whether an access should be allowed at or following the moment of the request. Organizations whose policies are not reasonably decidable cannot depend fully on classic access-control schemes, but will likely benefit alternative approaches, as discussed in the following chapter.

- **Maxim 8:** Policies may not always be human-decidable, but they are human-recognizable.

  **Corollary:** It may not be possible for users to generate an acceptable policy, but they can recognize whether an access was acceptable after the fact.

Assumptions 5 and 9: The centralized management of policy and resources, and the rate at which access-control information changes

All human organizations change, and a majority of the challenges in access control relate to the need to evolve the policy to accommodate changes in the organization. Some organizations (or parts of organizations) change slowly, but others — particularly those in dynamic and highly competitive sectors — change quite rapidly. I think that it is possible for even fast-changing organizations (those that violate Assumption 9) to have effective access-control systems, as long long as those systems are designed to evolve with the organization.
• **Maxim 9:** All accurate access-control policies evolve.

A centralized management scheme will be more or less appropriate (and Assumption 5 more or less accurate) depending on the size of the organization, its rate of change, and its central management’s ability to react to that change in a reasonable way. Of the organizations I studied, not a single one observed a strictly centralized resource- or policy-management strategy. That said, none of them observed a strictly decentralized strategy, either: each organization found its own balance between the rigidity of the former and the relative chaos of the latter.

• **Maxim 10:** Some resources and policies within an organization will be managed centrally. Others will be managed in a decentralized fashion.

• **Maxim 11:** It takes non-trivial time to alter a centrally-managed policy. Decentralized, local policy management can react more quickly.

Crafting and maintaining an access-control policy incurs overhead costs based on the number of stakeholders involved, as each stakeholder must negotiate a policy that will be acceptable from her perspective.

• **Maxim 12:** Classic access-control policies are well-suited to being managed centrally, but are expensive to maintain in a decentralized fashion.

• **Maxim 13:** Policies that are likely to need change frequently should be managed in a decentralized fashion.

**Assumption 8: Repercussions of over-entitlement vs. under-entitlement**

While over-entitlement exposes an organization’s data to a certain risk of compromise, under-entitlement can prevent members of an organization from completing vital tasks. The subset of organizations that value the completion of such tasks over
avoiding data compromise will functionally tend toward over-entitlement rather than under-entitlement (and thus violation of Assumption 8).\(^1\)

- **Maxim 13:** It is sometimes better for a user to violate policy if it helps them get their job done. In these cases, no amount of education or grandstanding from a security professional will change this prioritization.

- **Maxim 14:** An organization that has an accurate record of policy violations can move toward improving its policy or its processes.

  **Corollary:** When an organization drives its users to hide their policy violations, it shoots itself in the foot.

### 8.3 Conclusions

Given that computer science is descended from mathematics, I find our community’s pursuit of rigorous formal models understandable: we like being able to say that a system is “secure”, and have proofs to that effect. However, pretending that security is a binary property — that it really is possible to sign off on a system or policy as being “secure” — is driving us to ignore the subtleties of the real world. We are making a best-guess effort when we make a security policy, but too often we then stick our heads in the sand before we can see what the results of our effort were. As Linda Cureton (CIO of NASA) notes, “policies in and of themselves do not eliminate cyber security compromises” [14]. If we want to achieve some measure of security, we need to instead observe our systems constantly, acknowledge their complexity, and admit to the fact that “security” is a constant process, not a product of finite action. Unfortunately, this reality seems incompatible with the bulk of the access-control research literature.

\(^{1}\)This tendency seems to be usually informal or organic — and also quite distressing to the organization’s compliance officers when they do not understand its root cause.
Chapter 9

Rebooting access control

My fieldwork documents problems with access control as it is practiced today in large, dynamic organizations. As I outlined in the previous chapter, I believe much of the blame for these problems lies not with those organizations or the users that compose them, but with the research community.

I argue that this community has not only failed to provide many practitioners with adequate tools, but that it has (inadvertently) misled many organizations into adopting an unnecessarily narrow view of access control. Despite these long-festering deficiencies, the research community seems to have largely stagnated into a state of continual but incremental elaboration on well-worn themes. While some work (such as studies into the usability of policy-management tools) truly strives to understand and mitigate the deficiencies of our current solutions, most contributions consist of small extensions to existing formal models. The authors in the latter camp often fail to motivate the need for their work in the context of real-world applications, and doubly fail to meaningfully evaluate it using any metrics other than its (mathematically) “provable” security properties.

In this chapter I aim to turn that pattern on its head. Rather than trying to improve the the state-of-the-art by adding yet another layer to the towering edifice
that is the literature on formalized access control, I propose we raze the ivory tower down to its narrow footing — or, at least, ignore the existence of the tower for a short time, so that we can attempt to craft a new foundation. This foundation should offer a clear definition of the basic access-control problem — the real-world conundrum that we try to address with access-control models and systems (and, thus, the fundamental goals of access control). It should have an expansive vision of the solution space — a recognition of the larger universe of solutions that real-world practitioners can employ to address their own instances of the access-control problem. This vision should, importantly, recognize and embrace the role that human users play in the maintenance and evolution of access-control systems. Finally, this new foundation for understanding access control should offer insight into the suitability of these solutions for different settings.

The rest of this chapter proceeds as follows.

- Section 9.1 recasts the problem as being one of access mediation, and describes — for the first time, to my knowledge\(^1\) — a functional representation (which I refer to as the Portunes Function\(^2\)) of access-mediation systems, which unifies understanding of previously disparate models (classic policy-based access control, Optimistic Security, and Reactive Access Control) under a single framework.

- Section 9.2 examines types of errors that can occur in crafting inputs to the Portunes function.

- Sections 9.3 through 9.6 describe the three broad classes of access-mediation

---

\(^1\)Although members of the research community have called for a “meta-model” to attempt to unify existing models [22], I think their interest is more oriented toward proving formal equivalence between these models than on providing on a high-level description of the space of possible solutions, which is what I try to do here.

\(^2\)Named after an ancient Roman god associated with keys, doors, and livestock. Although I am sure that frustrated information security practitioners might draw parallels between this last domain and their own efforts to herd members of their organization into the constraints of access-control policies, I choose to focus more on the keys and doors.
systems (ex ante, ex post, and uno tempore), their attributes, and how they relate to models from the literature that implement them.

• Section 9.7 offers concluding perspectives on the contributions presented in the chapter.

9.1 Reframing the problem

What is the basic problem that access-control systems are supposed to solve? To answer this question, consider for a moment what would happen if access control had never been invented: in their absence, what new concerns would P-Bank or Teaching Hospital have? I believe that we can summarize their worries thus: that users would enact accesses to the organization’s digital resources that are not advantageous to the organization (i.e., that either do not further the organization’s mission or outright hamper it).

How would organizations go about allaying these concerns? In particular, what functions or tasks could they build into computer systems$^3$ to achieve this goal? Unencumbered by the historical focus on “provably secure” solutions, and mindful of the complexities inherent in large socio-technical systems, I believe there are two basic strategies:

• **Monitoring**: track users’ accesses in order to hold them accountable for their actions afterward (and make sure they know that this monitoring happens)

• **Checkpoints$^4$**: make it difficult for users to perform the detrimental accesses in the first place, by serving as a gatekeeper for every access they wish to perform

---

$^3$Computer systems are not the only way to accomplish this goal, but a reboot of the alternatives (such as user education) is beyond the scope of this work.

$^4$Here I make reference checkpoints found at international borders, not those found in the fault-tolerance literature.
Given that an organization could choose to use either or both of these techniques, the task of deterring users from doing detrimental things is not one of just controlling access (which would seem to refer only to the “checkpoint” approach) so much as more generally mediating access. I thus use the term “access mediation” for the remainder of this document to refer to this more-inclusive vision of the potential solution space to the problem at hand.

9.1.1 A functional representation of access mediation

A good access mediation system will probably employ both strategies — checkpoint and monitoring — at once. We build toward a functional definition of a system that uses both by first offering definitions of systems that employ each strategy individually. We can then use the parts of this functional representation to discuss some of the deep challenges uncovered by my fieldwork.

The basics: subjects and accesses

Any system that mediates access must accept as one of its inputs a representation of the access it is to mediate. I therefore use \( A \) to denote an access, and define it to include three pieces of information:

- **Action**: the operation being undertaken (e.g., read, append, or sign)
- **Subject**: the entity performing the action
- **Object**: the resource upon which the action is being performed

There are a large number of ways that an individual system can represent each of these pieces of information. To facilitate later discussion and without loss of generality, we assume that \( A \)’s representation possesses the following properties:
• **Intelligible**: the representation uses units and formats that are comprehensible to a human; e.g., the representation of an object would consist of a filename rather than a filesystem-block identifier.

• **Specific**: while being abstract enough to achieve human intelligibility, the representation is particular enough to be unambiguous. For example, even if the system was able to determine that the user jane was part of the group students, A would include the username rather than the group identifier (which are conveyed in the context, discussed below).

• **Stable**: in addition to being intelligible and unambiguous, the representation of a given action, subject, or object remains relatively static over time; i.e., a given digital object will be represented by the same identifier throughout its existence.

**Checkpoint systems**

I now define $CP$ as a function that represents a checkpoint system. In addition to $A$, $CP$ accepts as input additional contextual information $C$ that the system needs to make a decision. $C$ can contain supplementary data about the subjects, objects, and actions referenced in $A$ (i.e., jane is a member of the students group, or has been assigned the TeachingAssistant role), but also information external to $A$ (i.e., the university is in its final exam period). Whereas $A$ contains information that is specific and stable, $C$ contains information that applies to multiple elements in the access-control system or the organization, and that can change on a regular basis.

Given an access request consisting of $A$ and $C$, the point of a checkpoint system is to evaluate that request and return a verdict $v$, which indicates whether the system says the access is permissible. We can thus represent a checkpoint system in the following manner:
\[ CP : (A, C) \rightarrow v \]

Note that this formulation does not offer any insight into how the mapping is actually accomplished. Most existing checkpoint systems are configured with a policy that defines a set of \((a, c)\) pairs that are allowable, and denies requests that are outside this set. In this case, we can say that the mapping definition occurs preemptively, before we even know whether a particular access query will ever be made at all.

However, as described in Chapter 2, reactive access control \cite{58} works on the principle that some access decisions must be made by humans in near real-time. While a human decider certainly uses heuristics in evaluating an access request, we cannot say that she uses a policy in the same way that a computer decider does.

Whether implemented as an interactive system with a real-time mapping or as a classic policy-based system with a pre-defined mapping, checkpoint systems are only part of the picture. To complete the functional definition of access mediation, we next explore how to describe monitoring systems.

**Monitoring systems**

A “pure” monitoring system would not attempt to control access in any way; it would merely observe the accesses that users perform and record those observations. Given that human decision-makers will be reviewing accesses at a later point in time (and holding a user accountable if they determine that the user acted inappropriately), it will likely be helpful for a monitoring system, like a checkpoint system, to record as much information as it can about the context in which the access occurs. This context in this case could include aggregate historical information about previous accesses, such as whether this is the first time that this user perform this access (data gathered by the system itself over the course of time). It could also conceivably act to notify human deciders in response to an access.
Our notation for such a pure monitoring system should thus reflect the system’s ability to both record its observations, and its ability to perform some sort of analysis and reactionary action in response to the data it gathers. To that end, we define $\mathcal{M}$ as mapping $A$ and $C$ to a reaction term $R$, which can represent one or more actions that the system takes in response to the access.

$$\mathcal{M} : (A, C) \rightarrow R$$

As with checkpoint systems, this description offers no insight into how or when the mapping is actually defined. Given that the point of a monitoring system is to gather enough information about the accesses that users perform to then be able to hold users accountable for their actions, it seems clear that the system design should focus first and foremost on producing a complete and accurate log of the accesses and the contexts in which they occur. Additional processing can be useful, but only if the logs are accurate to begin with.

**A unified function for hybrid systems**

As noted earlier, real-world systems are unlikely to follow just one of the checkpoint and monitoring paradigms; every organization I studied in my fieldwork had systems that were a blend of the two. We thus now define $\pi$, a function that combines the features of $\pi_{cp}$ and $\pi_m$ into a single, general-purpose representation of an access-mediation system, which I refer to as the *Portunes Function*:

$$\pi : (A, C) \rightarrow (v, R)$$

In other words, we can say that an access mediation system is one that takes

---

5We could also say that $\mathcal{M}$ maps *sequences* of $(A, C)$ to a reaction $R$, in which case $R$ can be constructed as to involve fewer bookkeeping actions — a nuance that future work involving this formalism should take into account. For simplicity, we continue here with the more constrained assumption that the right-hand-side of the mapping consists of a single $(A, C)$ tuple.
in a representation of an access (containing subject, action, and object identifiers) and a representation of the context in which it is to be performed, and returns both a verdict (indicating whether the access can go forward) and one or more reactive actions (recording the access in various ways or places).

We next explore the concepts of context and mapping more fully, and observe that they are the root of many of the real-world problems I have documented.

9.1.2 Mapping embodies organization objectives

Earlier in this chapter I framed the purpose of an access-mediation system as decreasing (relative to the absence of such a system) the probability of an organization’s users performing accesses that are detrimental to the organization. However, it can be practically challenging to decide whether an access is “detrimental to the organization”, especially since any particular access may offer both benefits and drawbacks. Moreover, ignoring this mixed nature and asserting that an access can be uniformly detrimental or beneficial to an organization would risk violating the “no absolutes” principle, and could mask some of the very challenges we are trying to understand.

The objective view

Let us thus get a bit more specific, and describe access-mediation systems not just in terms of discouraging nebulous-define “detrimental accesses”, but accesses that are detrimental with respect to a set of particular goals.

An organization seeks to mediate users’ access in order to attain a set of business objectives \( O \). An access \( A \) in context \( C \) is beneficial when \( A \) facilitates achievement

---

6For example, an access may earn the organization money, but may also expose it to financial risk. The organizations I observed often baked these competing interests into the organizational structure, such that different divisions (investment traders and compliance officers) were required to negotiate and determine what was ultimately “best” for the organization. As the financial crisis of the late 2000s indicates, this process is not perfect, and the “best” course of action can often prove to be far from optimal; yet again, it is important to remember that there are no absolutes in this game.
of $O$, and detrimental when it impedes that achievement. We can denote an access-mediation system that supports achievement of $O$ as $\pi_O$.

We note that $A$ in $C$ can be beneficial to some objectives while being detrimental to others: for different objectives $O[j]$ and $O[k]$, $\pi_{O[j]} \neq \pi_{O[k]}$. In other words, given the same $A$ and $C$, systems designed to implement the individual objectives would return a different $v$ or $R$.

The Portunes mapping is what much of the access-control literature refers to as policy, but I argue that my formulation presents a couple of important advantages.

First (and as noted multiple times earlier in this document), the classic conceptualization of access-control systems assume that there exists a single “right” policy. I do not necessarily expect an organization to sit down and explicitly list its diverse objectives while defining the Portunes mapping for an access-mediation system. However, it is a distinct positive step to simply recognize that there can exist competing objectives within an organization, and that figuring out what the “right” mapping is in the presence of such conflicts is one of the major challenges of access mediation.

Second, the “policy” terminology forces us to accept the view that only official policymakers can assert what is “right” for the organization, but the mapping terminology encourages us to consider other perspectives — those of the users — as valid, too.

The role of humans

As discussed in previous chapters, during my fieldwork I observed multiple instances where policymakers did not understand the real-world environment in which users were operating. Part of this mismatch comes from the fact that the sets of objectives

---

7In practice it is difficult for an organization to reliably enumerate its objectives beyond a high level: “make money for shareholders”, “stay in business”, “don’t get in legal trouble”, “maintain a positive reputation with customers”, especially as the organization slowly changes over time. The next chapter presents a framework to help an organization identify what its access-mediation needs are even without developing a detailed picture of its diverse objectives.
that individual, well-intentioned users are trying to achieve as part of their day-to-day jobs (e.g., $O_{Mark}$ or $O_{Alice}$) overlap with — but do not exactly match! — the set of objectives $O$ that the organization is trying to achieve as a whole. While we can try to help individual users see the big picture and act in the interests of the organization (i.e., align $O_{Alice}$ with $O$), practical realities conspire to make these sets of objectives diverge. When an individual’s perceived objectives diverge from that of the organization, so too does her expectations for how the access mediation system should behave ($\pi_{O_{User}}$).

In a situation where the actual system $\pi$ diverges from the user’s expectation $\pi_{O_{User}}$, my fieldwork indicates that there are two broad classes of outcomes (also depicted by Figure 9.1):

- In cases where the organization recognizes the user’s perspective as being valid (as was the case with the assertive investment bankers) and provides an easy mechanism for them to provide feedback that will be listened to, there is an opportunity for $\pi$ to be tuned to more closely match $\pi_{O_{User}}$ (or, if $O$ and $O_{User}$ are irreconcilably incompatible, to find some way to help the user adjust $O_{User}$).

- On the other hand, in cases where the organization does not expect $\pi$ to require constant tuning and does not assume that there is something to be learned from its mismatch with $\pi_{O_{User}}$, the user is incentivized to subvert the system to make it meet her expectations, or demoralized and frustrated because she cannot do her job properly.

By allowing for the possibility that users can also have valid opinions on access-mediation decisions, the mapping terminology I propose allows us to consider and describe the way that humans evolve the behavior of access-mediation systems over time (and helps us understand why some systems fall prey to subversion by well-intentioned users).

171
Figure 9.1: Potential outcomes when a user Alice perceives a conflict between the actual system $\pi$ and her expectation $\pi_{O_{\text{Alice}}}$.

**On timing**

Another nuance of the access-mediation problem that the mapping terminology allows us to discuss is the *timing* with which the mapping occurs. Before computer systems, the most basic form of access mediation involved having a human being calculate the $(A, C) \rightarrow (v, R)$ mapping in real time as accesses are attempted (providing *unomtempore* decisions). Digital access-control policies allows us to timeshift the decision-making, and define the mapping before the access even occurs — thus delivering *ex-ante* decisions. Monitoring systems that allow accesses when they are attempted but re-evaluate them later (with the goal of holding users responsible for harmful actions) can be viewed as conducting an after-the-fact remapping, or *ex-post* decision. I discuss the relative strengths and weaknesses of these different approaches — these different *verdict timings* — in Section 9.3.

**9.1.3 Context embodies organization state**

Just as the Portunes mapping embodies the organization’s objectives, context embodies the organization’s *structure* (who has what role, who manages whom, etc.) and *state* (who has accessed what, and which tasks are in the process of being accomplished).
The history of access control is paved with increasingly-expressive models that build toward this goal. From RBAC (which seeks to make access management more wieldy by informing the system about aspects of the organization’s structure) through ABAC (which incorporates current attributes of users and/or objects) and on to hosts of models that include additional descriptive details (e.g., GEO-RBAC’s geolocation [5], TRBAC’s temporal constraints [45], or the explosion of attributes available with XACML [64]), the community has seemed focused on the idea that more data will make for more perfect access mediation.

However, real-world practitioners have not adopted these increasingly rich models. By abstracting away the details of what information is contained in the Portunes context, I aim to focus our attention on the challenges that are inherent to the management of contextual information — challenges that I believe come to bear in all access-mediation systems.

**Context sourcing**

Perhaps one reason that real organizations have not adopted models as rich as those described in the literature is that the real challenge in teaching machines how to make use of complex contextual data is not in crafting the internal logic so much as it is in *gathering and channeling the data itself*. For a datum (“User Jane has the student role”, or “a smartphone belonging to user Alice reports a geolocation of [43.70708,-72.28712]”) to be useful to an access-mediation system, that datum must be made available to the system, either by having a human provide it as input directly (when a system administrator adds Jane to the student role) or by being measured by some

---

8There may be a deeper reason for this; work in psychology indicates that attentive deliberation — gathering data and making a carefully-reasoned conscious choice — may actually lead to less satisfactory outcomes for consumers than “unconscious deliberation” when making complex purchasing decisions [19]. It would be interesting to see whether this dichotomy between background and foreground reasoning could explain some of the challenges associated with crafting complex ex-ante policies, where the outcome of the decision is judged not just in terms of the individual’s satisfaction, but by its impact on the organization.
connected computer system (such as a smartphone’s GPS).

With the exception of human-to-computer authentications (which are by definition measured by machines), the majority of the contextual data used by the access-mediation systems in my fieldwork was input by humans.\textsuperscript{9} Furthermore, once input, most of this was also maintained (i.e., updated and/or removed) through manual human action.

\textbf{Errors}

As the amount of data that is maintained by humans grows large — because there are many types of data, because the organization’s dynamism requires it to be updated frequently, or simply because there are many humans and objects for which data is tracked — so too does the time investment necessary to maintain it. Humans will inevitably introduce errors while adding or updating data — because they are overburdened, bored by the repetitive work, or simply because humans are fallible creatures. Again, as the amount of human-sourced data grows large, so too will the number of errors that are (quite naturally and understandably) introduced.

System designs that rely heavily on human sources of contextual data must accommodate these inevitable errors. Organizations that deploy such systems must recognize that they place significant burden on the humans in question, and devise mechanisms to mitigate that burden. Where possible, the system design should be optimized around re-using data that has already been input (e.g., by sharing parts of a central HR database rather than creating a new one to track users’ names and office assignments) and on automating updates if at all possible.

Section 9.2 looks at the things that can go wrong with the context, including flaws that can be introduced by human users adding or updating it.

\textsuperscript{9}In several cases the data was relayed from one computer to another before it reached the access-mediation system, but its origin was with a human being taking manual action to create the data in the first place.
Human-driven updates

As with mapping, human users are going to disagree with the system's notion of context, which in turn is going to cause the user's perception of how the system should be behaving to be different than what the system is actually doing.

Also as with the mapping, when a machine and a human disagree, an astute organization will try to harness the user’s opinion as a signal that perhaps something has gone wrong in the context management — i.e., that a certain type of data should have been updated, or should generally be updated more frequently than it currently is.

Contexts through time

The amount of contextual information that is available to the system can change over time. One strength of an ex-post approach to access mediation is that there may be more information available to humans reviewing the access after the fact than there was to the system at the moment the request was made. On the other hand, if the access-mediation system is unable to capture all the relevant contextual information, waiting too long to perform a review risks having it fade in the memory of the humans storing it.

9.2 Types of input “errors”

Ideally, the data input to an access-mediation system will be the “right” data — i.e., the data that the system needs to deliver a verdict, or that the organization needs to do an accurate post-facto analysis of the access. Of course, there can be several problems with the information that is input. Some of these problems may have their root in the system design (or in the tradeoffs that the designer made). Others may be unanticipated by the system design, either because they are emergent phenomena or
because they result from a machine or human doing unexpected things. The following represents an initial attempt to identify the problems that can go wrong with inputs to an access-mediation system.

### 9.2.1 Omission

Errors of omission occur when a system fails to measure or record one or more types of information that are relevant (either in delivering a verdict at the time of the access query or in weighing whether the access was appropriate after the fact.)

### 9.2.2 Granularity

Errors in granularity occur when the correct type of information is being measured, but when the units of measurement are either too coarse (resulting in a loss of useful information) or too fine (resulting in redundant information, which overwhelms or confounds the human users ultimately responsible for making meaningful decisions with it).

### 9.2.3 Freshness

Data that is not updated frequently enough will produce verdicts that are based on a partially-outdated picture of the organization. Data that is updated too frequently generates unnecessary work for the system (or its human components).

### 9.2.4 Reliability

Independent of granularity and freshness, a data stream is reliable if the data it delivers is an accurate representation of the real-world phenomenon of which it is a measurement. Damaged or flawed equipment can introduce accuracy problems while measuring data, as can human error or intentional subversion.
9.3 Choosing a deterrence approach

In Section 9.1 I noted that we can describe access-mediation systems as coming in three flavors, according to when the organization decides whether the access is desirable:

- **Ex ante**: the organization decides before an access request is received by defining and encoding a Portunes mapping ahead of time. The system acts as a checkpoint, allowing some access requests but denying others.

- **Ex post**: the organization deters undesirable accesses by holding users accountable after-the-fact. Access decisions come after an access request is received and after the access is already executed. The system acts as a monitor, allowing all accesses with provisions to permit the organization to review them and make a decision later.

- **Uno tempore**: the organization makes decisions after an access request is received but before the access is executed. The system acts as a checkpoint in that it can allow or deny access requests, but it does not require a Portunes mapping to be defined before it can start operation.

To my knowledge, this categorization, which unifies a number of disparate access-mediation schemes\(^\text{10}\) into a unified framework, is both novel and useful for thinking about the different approaches an organization can take to the access-mediation problem.

The following sections explore the attributes of each type of system, with an eye toward highlighting the factors that an organization should consider in deciding which approach to take. With the exception of the last subsection, the discussion is focused

\(^{10}\)Classic policy-based access control, Optimistic Security, and Reactive Access Control, each of which is addressed through the course of the section.
on “pure” versions of each type (i.e., the attributes of a system that uses only ex-ante decisions, with no ex-post or uno-tempore ones). I expect that the majority of real organizations will find hybrid systems — systems that incorporate all three approaches for different types of data — to be most effective in practice, but this presentation will allow future researchers to perform a more informed and systematic consideration of hybridization approaches by first nailing down the basic forms.

9.4 Attributes of the ex-ante approach

The ex-ante approach is “classic” access control: it requires an organization to define an exhaustive Portunes mapping (traditionally called a policy) before the system begins operation.

9.4.1 Potential for rigorous protection

Ex-ante schemes were designed for the purpose of protecting sensitive data. When they operate as designed, they can do a good job at averting irreparable damage that other forms of deterrence are likely to prevent — e.g., preventing a terrorist from gaining access to nuclear launch codes.

9.4.2 Known quantities

Ex-ante schemes are good for organizations that are able to enumerate all the possible values of $s, a, o$ and $C$ — and, moreover, for organizations where the sets of subjects, actions, and contexts remain relatively constant over long periods of time. In other words, ex-ante schemes are good for organizations where the universe of legitimate access cases is well-known and relatively static.
9.4.3 Costs of mapping

A basic disadvantage of ex-ante systems is that the process of crafting good access-mediation mappings can be quite expensive, and so is cost-prohibitive to repeat frequently. This, in turn, can lead to mappings being out of sync with the current needs of the organization.

9.4.4 Organizational change

This means that ex-ante systems are not well-suited to environments in which there is a large amount of organizational change. Unix groups and later RBAC were designed to make it easier to manage ex-ante access policies as organizations underwent certain types of predictable changes. By grouping subject and access identifiers or creating abstract sets of users and permissions, implementing modifications associated with employee promotions and transfers became less tedious (and therefore less expensive and less prone to errors).

Thus far, however, there does not seem to exist a mechanism to increase the manageability of ex-ante policies for organizations that frequently redefine the tasks associated with a particular job function (or split one job function into two), or that regularly enter into new business domains (e.g., by acquiring another company). An environment in which a particular request can map to ALLOW one day and DENY the next — and where such changes are commonplace\textsuperscript{11} — can be ill-served by a purely ex-ante system.

9.4.5 Cost of false negatives

Ex-ante systems can also incur a high cost when false-negative verdicts are generated, largely because the process for overruling a verdict involves a human decider. This

\textsuperscript{11}E.g., in a hospital where nurses move from department to department every few months.
cost is experienced by the business, but most palpably by end users. Over time, this in turn drives overworked administrators to err on the side of over-entitlement in assigning permissions; an observer incorrectly believes upon cursory inspection that the rigorously classic “default-deny” wisdom is being followed, but in reality it is undermined with every small change the organization undergoes. In short, the organization falls prey to all of the problems I documented during my fieldwork.

9.4.6 Highest priority: minimize over-entitlement

Given the high cost of false negatives, ex-ante access schemes would seem to be suited to organizations where the damage that could result from over-entitlement is deemed to outweigh the cost incurred by under-entitlement.

9.5 Attributes of the ex-post approach

Ex-post systems are a good deterrence option when an organization already has a high confidence that their users will generally act in the interests of the organization, or when the cost of under-entitlement is greater than the cost of over-entitlement. Some of these properties, such as organizational culture, may be somewhat mutable through user education, etc., but a few are more intrinsically related to the nature of the data to which access is being mediated:

1. the magnitude of the benefit the user could reap from interacting with the data in a way that is against the interests of the organization,

2. the severity of the repercussions the organization can impose upon the user in comparison to (1),

3. the magnitude of harm to the organization that could result from users taking negative actions.
Ex-post systems are poorly suited to organizations for which the values of (1) or (3) are great, or the value of (2) is small — i.e., if the motivation for an individual to act against the interests of the organization is decidedly greater than the deterring consequences, or if great harm could come from negative user action.

9.5.1 Flexible without false negatives

Perhaps the greatest strength of an ex-post approach is that it enables an access-mediation system to seamlessly accommodate changes in the organization as it evolves over time. This is useful in situations where organizational change would induce false-negative verdicts with an ex-ante scheme; for that reason, the ex-post approach is advantageous where false-negatives are particularly costly or dangerous.

9.5.2 Potential for abuse

The flip side, of course, is that the default-allow scheme of ex-post approaches poses a higher risk of allowing malicious insider activity to go on unnoticed, especially if the organization’s system or processes for reviewing the accesses that occur is not comprehensive or takes a long time.

9.5.3 Costs for set-up and review

Ex-post systems do not deliver \texttt{DENY} verdicts, so do not require an initial investment to define a mapping between access query inputs and verdicts. However, this does not mean that the ex-post approach is some kind of low-cost panacea; investments are still required to:

1. build mechanisms for gathering appropriate contextual input data (i.e., to capture the circumstances under which the access occurred), which is necessary for meaningful ex-post access decisions
2. build systems implement the Portunes reaction to an access (i.e., to filter or pre-process data about accesses and reduce the burden of manual review)

3. build systems or define organizational processes to facilitate the manual reviews

4. conduct reviews

Items (1) through (3) represent largely up-front costs, but will certainly require additional investment as the organization, its digital resources, and the organization’s understanding of its access-mediation needs all evolve. The cost associated with item (4) will depend on the design of the previous items, but will need to be ongoing as long as the access-mediation system is in place — the need to at least periodically review accesses will never go away.

9.5.4 Break glass

The electronic medical system described in Chapter 7, THERS, uses ex-post access mediation to make sure that clinicians have access to the data they need, while also deterring them from accessing data for reasons other than patient care. The system is largely world-readable by clinicians, and the organization performs reviews of accesses performed on a given patient’s medical data if requested (e.g., by a patient who feels that her data may have been inappropriately accessed).\footnote{While Teaching Hospital felt that this type of “on-demand” review was sufficient for their purposes, I suspect that they might not be opposed to taking a more proactive approach if tools existed to help them perform additional analysis in a less labor-intensive way.}

This hospital system also uses a break-glass\footnote{Break-glass is described more extensively in Chapter 2.} mechanism to provide an extra layer of protection for particularly sensitive clinical data. Asking a user whether she wants to “break-the-glass” — and annotating the access log to highlight that the glass has been broken — is an implementation of a Portunes reaction to an access request. In this manner, break-glass mechanisms can also be used in ex-ante systems to allow
users to escalate their level of privilege and access things that the pre-defined verdict mapping would not normally allow — and thus create a hybrid system by adding a measure of ex-post mediation.

9.5.5 A history of clinical reviews

The driving factors in THERS’s access-mediation design were primarily 1) the difficulties of coming up with an ex-ante mapping and 2) the high costs of false negatives, but an ex-post approach corresponds to a permission model that has long been established in the medical community: clinicians are expected to operate independently, and allow their actions to be guided by their training and professional judgement. (In other words, organizations have a high degree of confidence or trust that clinicians will generally do the right thing.)

At the same time, regular reviews take place to make sure that the decisions being made are reasonable. In cases where the reviewers determine that a decision was not in the best interests of the patient, the clinician is subject to censure, including revocation of their medical license and legal action. These reviews are relatively rare, and end up covering a relatively small number of the medical decisions made by practitioners. However, it is still an interesting model of a non-digital system for performing ex-post reviews of actions taken by individuals within an organization — and holding individuals accountable when they have acted against the interests of the organization (which in these situations follow the interests of the patients). Perhaps the historical use of this model is part of why ex-post (or hybrid break-glass) approaches seem more psychologically acceptable to clinical organizations than to

---

14 The depth of clinicians’ knowledge about medicine is precisely what allows them to make informed tradeoff decisions while providing patient care. If we could similarly find a way to help security laypeople reason about parallel tradeoffs, perhaps a larger diversity of organizations could come to trust its members enough to adopt ex-post schemes. Unfortunately, the pedagogical expertise of the security community is centuries behind that of the medical establishment.

15 These include reviews after major negative events (like the death of a patient), which closely parallel the ex-post audits, but also second opinions, which have a somewhat uno-tempore flavor.
9.5.6 Optimistic security

A form of ex-post access control was described in the literature by Dean Povey as optimistic security [70]. This prescient work, which I discuss more in Chapter 2, is an almost-perfectly general model of ex-post access mediation. The one place that it falls down\textsuperscript{16} is in the requirement that all mediated actions must be “recoverable”:

“Accesses which write, modify or delete data must be able to be rolled back to ensure that a user cannot irreparably damage a system. Actions which have external behaviour (e.g. firing a missile, sending a letter) should be associated with compensating actions to restore the system to a stable state (e.g. abort the missile, send apology letter). [...] In general it is assumed that for any transformation on data or security properties of that data (confidentiality, integrity etc) there is a compensating transaction which exists to reverse this transformation. [...] Where this situation does not exist, it would be inappropriate to use an optimistic system anyway.”

While I think that this requirement is reasonable for many situations, and makes sense in the context of Povey’s effort to show that an optimistic security system can be “provably secure”, it is important to also note that there exist situations in which actions are not always recoverable, but where ex-post access mediation may still be the best option. Indeed, many of the actions that clinicians undertake are impossible to recover from, but that has not stopped their organizations from using an ex-post approach in both the digital and pre-digital worlds.

\textsuperscript{16}In its generality, not in its reasoning or overall quality.
9.6 Attributes of the uno-tempore approach

An uno-tempore system would theoretically strike a balance between the strengths and weaknesses of the ex-ante and ex-post approaches. In its pure form, such a system would require a human being responsible for making access decisions — i.e., a person who would otherwise define the verdict mapping in an ex-ante system or perform an ex-post review — to respond in real time to every access request the system receives.

This approach in its pure form is clearly not scalable in the settings I studied: with thousands of investment bankers accessing a financial database and dozens of clinicians and support people accessing an individual medical record, the latency of human decisions alone would cause these organizations to grind to a halt.

9.6.1 Reactive access control

However, it is possible to bring an uno-tempore flavor to ex-ante systems by making it easier for a user to request access to a resource, and by providing access deciders with a mechanism for modifying verdict mappings in real-time in response to an access request.

Described most recently by Mazurek et al. [58] as reactive access control, this approach is currently used by products such as Google Docs, which I have made extensive use of. When I navigate to a Google Doc for which I do not have read permissions, the application provides a link to “Request access” from the file’s owner.

Reactive access control is essentially a hybrid of the ex-ante and uno-tempore approaches. I am optimistic that it could be deployed to help mitigate some of the problems I have documented, but am cautious in this optimism because the research into its effectiveness has thus far focused on relatively limited situations, in which the following conditions held:

17 Discussed in Chapter 2.
18 A web-based word processor, currently available at https://docs.google.com
1. access deciders receive a relatively low volume\textsuperscript{19}

2. the resources being protected are discrete resources (text files, photos, physical resources, etc.) of which humans have a good intuitive understanding

3. the requestors are all people that the access deciders knew personally

4. the resources being protected belong to the decider, or the decider otherwise has the contextual knowledge to make a reasonable decision

I have found using Google Docs in a corporate setting that condition (3) has not always been true, since the company in question is large and collaboration with other teams is frequent (although the company provides tools for quickly learning about what projects other employees work on), and (4) is not strictly true as the documents belong to the company rather than to me personally (although I have been the document owner). Given that my experiences have largely met the conditions under which reactive access control has been studied, I am not surprised that I have had a generally positive impression of it; at the same time, I know that many of the resources to situations in which large organizations seek to mediate access will not meet these conditions. In Chapter 11 I call for further work to study the applicability of reactive access control in a broader diversity of settings.

9.7 Conclusions

In this chapter I have attempted to reboot our basic understanding of “access control” — abandoning that term in favor of the more realistically-minded “access mediation” — and presented a framework for thinking about solutions to this problem, one which accommodates models that were previously disjoint and unrelated.

\textsuperscript{19}A maximum of fifteen requests per day, as described by Mazurek et al.
This formulation allows us to meaningfully consider important factors — such as the role that users can play in helping a system evolve with an organization, and the limits of how much contextual information an organization can realistically gather and maintain about itself — that have hitherto been ignored by the literature. I believe that this ignorance is part of what has contributed to the problems documented in my fieldwork.

The next chapter addresses another neglected factor that plays a role in the challenges faced by real-world organizations: the organizational characteristics, types of organizational change, and external factors that should inform the design and implementation of an access-mediation system.
Chapter 10

Organization profiling

The task of evaluating the suitability of an access-mediation scheme for a particular organization requires that we check the compatibility of two radically different systems. On one hand we have the human organization — with its goals, its physical and financial resources, its sensitive data, and its human properties; on the other, we have an access-mediation scheme with its verdicts, information requirements, reactive action, and its technical limitations.

This chapter offers a catalog of the facts and properties of an organization that impact the success of any access-mediation approach. Combined with an understanding of the strengths and weaknesses of the ex-ante, ex-post, and uno-tempore approaches as described in the previous chapter, I believe that this catalogue will help practitioners understand the access-mediation problem and ultimately build systems that are easier to manage and that actually meet their organizations’ access-mediation goals.

Attribute categories

I start this catalog by first identifying categories of attributes that have the strongest role in determining the success of an access-mediation system. An organization can influence some of these properties (such as the budget and resources available for
access-mediation management), but are unlikely to incite significant change in others (e.g., the rate at which the organization’s mission and structure changes).

It is important to note that for some attributes (e.g., some aspects of false negative verdicts) we must focus on aspects of the access-mediation system, either as it operates today or as it is proposed to operate in the future. Other attributes (e.g., resources and budget) may exist largely outside of current or potential access-mediation systems, and can thus be evaluated independently as being more inherent to the organization as a social system.

**False positives** If the verdict delivered is ALLOW but an optimal verdict would be DENY, how much damage could the resulting access generate? Moreover, how easy is it to recover from that damage? For example, the consequences of a false positive in a military weapons installation could include the launch of long-range missiles against a hostile country, which would cause irreparable human, material, and diplomatic casualties.

**False negatives** Similarly, if Alice receives a DENY where an optimal verdict would be ALLOW, her ability to access data and take action will be sub-optimally limited. If she is a doctor attempting to provide care to a patient in critical condition, this limitation could negatively impact the patient’s health — in the worst case, it could be fatal.

**Organizational change** All organizations undergo change. Large companies that strive to react to dynamic market conditions are particularly susceptible: with each new opportunity comes the need for different data and access patterns. Even organizations whose mission and workflows remain static experience employee turnover, and users may come to be responsible for duties beyond the ones they are officially assigned.
User accountability  What disincentives are in place to discourage the user from taking damaging actions? Can the organization leverage some proportional control over the user’s future? (To draw an example from healthcare: when a doctor is found to have acted inappropriately, their medical license may be revoked.)

Contextual information availability  Access-mediation decisions rely on having an accurate picture of the state of the organization at the time of the access. Different types of this contextual information may be harder or easier for computer systems to track. Depending on what information is relevant to delivering accurate access-mediation verdicts, an organization may find that gathering some context is inexpensive, but gathering other context is functionally impossible. Context information availability is therefore determined by both the access-mediation system’s need for data and the cost of gathering and maintaining that data.

Resources and budget  Organizations that are able to dedicate more human-hours to the access-mediation problem will generally have a leg up on organizations whose resources are limited. Participants in healthcare struggled to fit all important projects in their limited priority queue, whereas participants from the investment banking industry seemed practically willing to burn money if it had the hope of reducing the firm’s exposure to risk.

Attributes and notation

In Sections 10.1 through 10.6 I break these categories down into individual factors or attributes, summarized in Figure 10.1. This presentation becomes more concrete in Sections 10.7 and 10.8, which offer profiles of a prototypical teaching hospital and investment bank, and discussion of each of these attributes in those contexts.

For each attribute or factor identified in Sections 10.1 through 10.6 I characterize a range of values that an organization might experience, from “high” to “low”. For
False positives

<table>
<thead>
<tr>
<th>A. Probability of damage</th>
<th>M. Available consequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Gravity of damage</td>
<td>N. Proportionality of consequences</td>
</tr>
<tr>
<td>C. Permanence of damage</td>
<td>O. Consistency of consequences</td>
</tr>
<tr>
<td></td>
<td>P. Awareness of consequences</td>
</tr>
</tbody>
</table>

False negatives

<table>
<thead>
<tr>
<th>D. Scope of obstruction</th>
<th>Q. Organizational understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>E. Magnitude of obstruction</td>
<td>R. Function homogeneity</td>
</tr>
<tr>
<td>F. Time to correction</td>
<td>S. Simplicity of necessary data</td>
</tr>
<tr>
<td>G. Cost of correction</td>
<td>T. Availability of contextual data</td>
</tr>
<tr>
<td>H. One-off corrections</td>
<td></td>
</tr>
</tbody>
</table>

Organizational change

<table>
<thead>
<tr>
<th>I. Turnover rate</th>
<th>U. Resourcing priorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Function turnover</td>
<td>V. Progressive metrics</td>
</tr>
<tr>
<td>K. Function dynamism</td>
<td>W. Trust and resource flexibility</td>
</tr>
<tr>
<td>L. Technical change</td>
<td></td>
</tr>
</tbody>
</table>

User accountability

<table>
<thead>
<tr>
<th>Contextual information availability</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Resources and budget</th>
</tr>
</thead>
</table>

Figure 10.1: A summary of the properties described in Sections 10.1 through 10.6.

example, if the attribute were “Number of employees”, we would use the following visual representation:

<table>
<thead>
<tr>
<th>The organization has hundreds of thousands of employees.</th>
<th>The organization has less than a dozen employees.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>

This conveys that an organization may find itself anywhere on the spectrum of states between the two extremes that are listed; in this example, an organization can have anywhere from “less than a dozen” to “hundreds of thousands” of employees.

10.1 False positives

If a particular false positive verdict $v$ has the potential of resulting in damage $d$, the organization must consider the probability of $d$ occurring, its gravity, and the organization’s ability to recover from it.

Note that for any given organization and any given piece of information there
may be many types of potential damage; a complete profile of these false-positive dimensions should ideally consider a variety of different threats.

10.1.1 Probability of damage (A)

In the case that false-positive verdict $v$ is delivered, how likely is it that $d$ will actually result? For example, imagine that a pharmaceutical company is worried about a competitor getting access to a drug formula. If a $v$ inappropriately gives Mildred access to that formula, what is the probability that Mildred will form an agreement with a competitor and successfully get the formula to them?

<table>
<thead>
<tr>
<th>Damage will almost certainly occur</th>
<th>Damage has a low probability of occurring</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>

10.1.2 Gravity of damage (B)

If damage were to occur, how consequential would it be? This could be measured in time, dollars, or more exigent currency. For example, the damages resulting from a false-positive verdict in a defense organization could impose a couple hours’ cleanup time (if an inexperienced user accidentally corrupted a database) or in the loss of human life (if a malicious insider leaked state secrets to a terrorist organization).

<table>
<thead>
<tr>
<th>Damage would be catastrophic (result in the death of a large number of people)</th>
<th>Damage would be a minor inconvenience to a small number of people</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>
10.1.3 Permanence of damage (C)

How well could the organization bounce back from the false positive and resulting damage? From the previous example, a corrupted database can likely be repaired with time and patience, but human deaths cannot be reversed.

\[
\begin{align*}
\text{Damage would be impossible to roll back or recover from} & \quad \iff \quad \text{It would be easy to return the system to a non-damaged state}
\end{align*}
\]

HIGH \quad LOW

10.2 False negatives

If a particular false negative verdict \( v \) has the potential of causing some obstruction \( o \), the organization must consider the scope of \( o \), its magnitude, the time and cost of the necessary corrections, and the probability of one-off corrections.

As with false positives, a complete analysis by a real organization should include a broad catalog of obstructions that can result from a particular false negative.

10.2.1 Scope of obstruction (D)

How many different people or tasks would the occurrence of \( v \) obstruct? Note that this is more a property of the system (current or candidate) more than a property of the organization, as the scope of false-negative obstruction can depend largely on whether the system uses groups or roles to help manage permissions. (In this sense, the scope acts as a sort of “multiplier” for the other variables associated with false negatives when evaluating their overall impact.)

For example, if a university’s access-mediation system is reconfigured to disallow all members of the “graduate student” role from modifying a database of student
grades, teaching assistants would be unable to enter grades. The scope of the obstruction would include all teaching assistants who would normally perform that task.

Many people would be obstructed in completing all of their activities ⇐⇒ A couple people would be obstructed in completing one or two tasks

10.2.2 Magnitude of obstruction (E)

What tasks would they be unable to do, and what would the larger effects of those tasks not being done have — what action would be altogether blocked while they could not perform that access?

In the previous example, teaching assistants would not be able to enter student grades, which could prevent students from accessing their final grades for the term through the online interface. However, teaching assistants could still inform students of their final grades via another channel, such as email.

In contrast, if an investment banker is obstructed from selling a stock that is rapidly losing value, the obstruction could cost the organization millions of dollars.

The obstructed tasks are of critical importance to the regular operation of the organization ⇐⇒ Failure to complete the obstructed tasks would have little to no effect on the organization

10.2.3 Time to correction (F)

How quickly could the system be righted to properly allow access? This dimension will depend on the nature of the access-mediation system — is there a mechanism in
place to allow users to appeal a verdict? How long does it take an administrator to evaluate the appeal and navigate the system interface? — and on the number and availability of system administrators.

A reactive system, such as Google Docs, is optimized to allow false-negative verdicts to be corrected quickly: the document owner is notified by email that a user has requested permission to access the document, so the time to correction will depend on the speed with which 1) the owner receives and checks email, 2) gathers additional information, if necessary (e.g., by looking the requestor up in the personnel database or chatting with them to determine why they want access), and 3) navigates the approval interface.

\[
\begin{array}{c|c}
\text{The obstruction is permanent, or takes months to remove} & \text{The obstruction can be cleared within a matter of minutes} \\
\text{HIGH} & \text{LOW}
\end{array}
\]

10.2.4 Cost of correction (G)

What would the cost of the correction \( c \) be? This will probably be determined primarily by the amount of time it takes the administrator(s) to weigh an appeal and navigate a software interface to make a correction (and thus depends heavily on the system being used), but in extreme cases could also include costs associated with re-issuing credentials like smartcards.

An organization might also weigh the cost of potential future errors that could be introduced when a correction is made. In the university example, if an administrator fixes the problem of teaching assistants not being able to access the undergraduate grades database by giving all “students” access (as opposed to just members of the “graduate students” role), over-entitled undergraduates could cause problems in the
future.

Correction $c$ would represent a sizable portion of the organization’s operating budget $\iff$ The cost of $c$ is negligible

10.2.5 One-off corrections (H)

Given that $v$ has already been generated and corrected, what is the probability of a similarly incorrect $v'\,\,\text{— another verdict generated on parallel inputs, modulo any changes made to the Portunes mapping or the organization — occurring in the future?}$

In general, a one-off correction could happen because the larger system is not able to learn and adapt to prevent similar false negative verdicts. In the context of the university example, imagine that Anthony is the first teaching assistant to discover that he cannot enter his grades. If the administrator reacts to his complaint by manually adding Anthony as an individual to the list of permitted accessors — rather than all members of the “graduate student” role, or more optimally (but more expensively from the administrator’s immediate point of view) to a new “teaching assistant” role — this one-off solution will not fix the situation for the other teaching assistants.

Incorrect verdicts similar to $v$ will almost certainly occur $\iff$ It is practically impossible for verdicts similar to $v$ to occur again

196
10.3 Organizational change

An organization should consider several factors relating to its own dynamism, including a couple different types of employee turnover and its rate of technical change.

10.3.1 Turnover rate (I)

What is the rate of employee turnover — how often do users leave the organization and are replaced? (Or, for organizations that are in a period of growth, how many new employees are joining the company?) This is significant because new employees must be provisioned, and that provisioning requires work. In cases where it is difficult to determine what provisions a new user should have, a high volume of new users can overload the administrative workflow and increase the probability of provisioning errors.

For general context, the national monthly rate of employee turnover in the United States varied between 2.8% and 4.3% from 2001 to 2011 [9]. I contrast this with the experience of one field study participant, who reported that a department in his organization experienced a turnover rate of about 25% (and that managing entitlements for this department was a source of tremendous difficulty). This is evidence that different organizations experience very different turnover rates.

<table>
<thead>
<tr>
<th>Annual turnover rates exceed 25%.</th>
<th>Employee turnover is so small as to be barely measurable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>

10.3.2 Function turnover (J)

How long do individuals stay in a particular function — i.e., how often do users transfer jobs within the organization? When transfers happen, how different are the
new functions?

In one large software company with which I have experience most employees seem to change positions within the company every couple of years (but are unlikely to change job titles other than being promoted — e.g., a software engineer usually remains a software engineer, although the software they are engineering may be very different). In teaching hospitals, medical students change functions every few months, nurses may change departments every couple of years (except for “floater” nurses, who may change daily!), but doctors rarely change their core job functions\(^1\).

<table>
<thead>
<tr>
<th>HIGH</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most employees change job functions every year or two, and some every few months.</td>
<td>Almost all employees are in the same function as when they joined the organization, with the bulk of changes being promotions within the same department or division.</td>
</tr>
</tbody>
</table>

### 10.3.3 Function dynamism (K)

How quickly does a particular function’s responsibilities evolve? In a retail bank, the daily tasks of a teller have remain largely unchanged for decades (although the technical systems he uses to perform those tasks may have evolved). In a consulting firm, the responsibilities of an individual consultant can change radically from week to week, depending on the needs of the client and the availability of other consultants in the organization.

\(^1\)A doctor in a teaching hospital may play a more or less active pedagogical or administrative role, but is unlikely to switch from one clinical specialty to another
The duties of individual job functions evolve slowly, and remain virtually unchanged for years at a time.

The duties associated with an individual function change every few months, such that a person's daily activities will be dramatically different from one quarter to the next.

10.3.4 Technical change (L)

How often does the organization deploy new computer systems? How often do existing computer systems receive major updates that change the workflows users traverse in performing their jobs? New systems or features can impose additional administrative costs (including managing new entitlements) and can make it difficult for users and administrators to know who should have what entitlements — and how to request or provide them.

The organization rolls out major new systems or introduces new features regularly.

The organization's technical systems are largely stable, and undergo only minimal maintenance updates.

10.4 User accountability

Several factors determine an organization’s position on the spectrum of user accountability, including the consequences it can inflict and its practices and culture around consequences. As described in Chapter 2, some of these factors were envisaged by Beaumé et al. in their discussion of compliance budgets [4].
10.4.1 Available consequences (M)

What consequences can the organization impose upon the user if she acts inappropriately? Are the possible consequences varied enough to encourage their application in cases of inappropriate user behavior? (An organization whose only recourse is to fire the user could be understandably hesitant to do so for a minor, first-time offense.)

\[
\begin{array}{|l|l|}
\hline
\text{The organization has few} & \text{The organization is easily able to impose a} \\
\text{consequences to choose from, and} & \text{variety of consequences, including those} \\
\text{their application is reserved} & \text{for both minor and severe infractions.} \\
\text{exclusively for severe cases.} & \\
\hline
\end{array}
\]

| HIGH | LOW |

10.4.2 Proportionality of consequences (N)

Given the magnitude of the potential damage\(^2\), are the consequences sufficient to deter users from taking the damaging action? The threat of losing one’s medical license may be enough to deter a clinician, but few consequences exist that could deter malicious access to military resources by individuals possessing a suicide-bomber mentality.

\[
\begin{array}{|l|l|}
\hline
\text{The potential damage from an} & \text{The organization has consequences} \\
\text{action is so severe that no} & \text{available that are significantly more severe} \\
\text{consequence could deter the type of} & \text{than the potential damage.} \\
\text{user drawn to take that action.} & \\
\hline
\end{array}
\]

| HIGH | LOW |

\(^2\)Of course, what the organization views as “damage” may be more accurately described as “profit” from the perspective of a malicious user.
10.4.3 Consistency of consequences (O)

Are users regularly held accountable for instances of inappropriate access or actions—even small ones? This depends both on the organization’s ability to actually enforce consequences (i.e., to notice there was an inappropriate access and catch the user responsible) and on whether the organization has a culture of following through with the consequences it has laid out.

Jérôme Kerviel [47] alleges that his managers knew about the trading activities that eventually lost Société Générale billions of dollars, and that the practice of exceeding one’s trading limits was commonly allowed because of the potential profit that could be made.

<table>
<thead>
<tr>
<th>HIGH</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>The organization is sometimes aware when inappropriate access occurs, but the consequences it implements vary wildly from one instance to the next.</td>
<td>The organization recognizes and reacts to the vast majority of inappropriate access with well-defined and consistent consequences.</td>
</tr>
</tbody>
</table>

10.4.4 Awareness of consequences (P)

Are users cognizant of the consequences that the organization will impose? This can tie into the organization’s cultural expectations for users; for example, based on personal conversations with members of the American intelligence community, it seems that they work in a culture that is keenly aware of the risks associated with inappropriate access, and the personal repercussions of violating organizational policies. On the other hand, healthcare participants in my study seemed to have no idea what (if any) penalty the organization would impose for violating policies by, e.g., writing down one’s password or walking away from a logged-in computer.
10.5 Contextual information properties

Of the dimensions on which an organization might define itself with respect to access mediation, I believe that its use and relationship with contextual information is one of the least understood. Important factors include the organization’s own understanding of its access-mediation decision, the degree of functional diversity among users with the same title, the amount of context information necessary to make a decision, and the availability and freshness of that data during the process of decision-making.

10.5.1 Organizational understanding (Q)

Can the organization identify the variables that are relevant to making a particular access decision? Are those variables well defined and documented, or are they only understood implicitly by the humans who deliver access decisions — or worse yet, guessed at during system implementation?

For example, what information is relevant in mediating access to a university’s registrar system (the repository for student grades and transcripts)? This type of data is relatively simple, highly structured, and usually managed by a single department. However, when one considers all the circumstances under which one might want to access it, the amount of information the system must maintain to both protect the integrity of the data and the student’s privacy grows large. For example:
• Submitting final course grades. (Can only professors submit them, or can a professor’s teaching assistant? Do they need to be reviewed before being committed to the system? Who can perform the review, and how is it noted as being performed?)

• Changing a grade in a student’s transcript. (Is there a time window in which a grade can be changed? Can anyone other than a professor of the course change a grade, i.e., if the professor has left the university? Does a department chair or dean have to be notified? What about the student?)

• Reviewing a student’s grades. (Is the student only allowed to see their grades after a certain period? What about a student’s parent, and how is a person authenticated as a being particular student’s parent? Can professors review their advisees’ grades? Can a department chair review a student’s grades — only for classes taught in their department, or for all classes? Under what circumstances can a law-enforcement official review a student’s grades, and how are those circumstances identified?)

For each additional action or restriction, more data must be maintained by the system to implement it: how easy is it for an organization to identify all these variables as being important to accurate access mediation?

<table>
<thead>
<tr>
<th>The organization has a hard time identifying what information is important to making all but the most trivial of access decisions.</th>
<th>System implementors and decision-making personnel can clearly agree upon and identify what information is relevant to making a particular access decision.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>
10.5.2 Function homogeneity (R)

Are users’ duties homogenous throughout a single function, or are there e.g. many “software engineers” occupied by disparate tasks depending on what software product they are currently engineering? In cases where job titles are not distinguishing, are there other sources of information within the organization that will help administrators or other users understand the nature of an individual’s work?

<table>
<thead>
<tr>
<th>HIGH</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job titles are vague, and it is difficult to determine what an individual’s job consists of.</td>
<td>Job titles are distinguishing, or other information about an individual’s tasks is readily available. It is easy to tell what a particular user works on.</td>
</tr>
</tbody>
</table>

10.5.3 Simplicity of necessary data (S)

How many different variables should go in to making an access decision? Are there some obscure factors that do not matter for most decisions, but when they occur cannot be ignored?

In the registrar example described above, I expect that several of the example actions (e.g., access by law-enforcement officials) are implemented not by providing direct access, but by requesting a copy of the data from the staff of the registrar’s office. This allows system designers to offload recognition of unusual situations that are hard to identify with confidence (whether there’s an emergency that warrants police access) to humans.
Decisions often require a host of variables, perhaps because the system must accommodate special cases (e.g., don’t provide access unless it is an emergency) that a smaller number of variables cannot capture.

Every access decision can be made with one or two variables (e.g., a username and a time of day).

10.5.4 Availability of contextual data (T)

How many contextual variables are automatically measured (e.g., “Jane’s RFID card has been scanned at the front door”), and how many are decided and manually updated by human administrators (“Jane is assigned to a weekend shift”)? For information that is input manually, how completely and how quickly does it propagate throughout the organization’s technical systems (must Jane’s shift assignment be manually entered at several different points)?

The majority of context data is gathered manually, and takes a period of days or weeks to propagate through the system.

Most contextual information is gathered automatically; even manual updates to contextual information propagate within a matter of seconds or minutes.
10.6 Resources and budget

Although many system administrators would embrace additional funding, the simple financial sum is but one among several resourcing factors that can impact the success access-mediation efforts. In particular, I believe that the relative priority the organization places on access mediation, the methods it uses for measuring the impact of its investment, and the flexibility in resourcing practices are also important.

10.6.1 Resourcing priorities (U)

The priority that an organization’s leadership places on funding its access-mediation efforts will likely be influenced by the relative importance of access mediation to achieving its central goals. Priorities may therefore be largely consistent across organizations within a single industry.

From my fieldwork, it seems that investment banks (for whom the protection of data provides a competitive advantage) place a higher priority on access mediation than healthcare institutions (for whom access mediation is viewed as a mechanisms for protecting patient privacy, but not a leading factor in their ability to provide quality patient care).

<table>
<thead>
<tr>
<th>The organization’s executives view access mediation at the bottom of a long list of priorities, and are consistently grudging in resource allocation.</th>
<th>Access mediation is critical to the central mission of the organization, and is at the top of the list during resource allocation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>
10.6.2 Progressive metrics (V)

Executive funding decisions are often driven by the expected return on investment (ROI): if we make improvement \(x\), how much money will we save in the long run? Organizations who recognize the inherent challenge of measuring the impact of infosec investments — the lack of catastrophic compromises that may never have occurred anyway — will fund access-mediation efforts differently than those who demand classic, concrete ROI evaluation. Furthermore, an organization that recognizes that access-mediation improvements can involve much more than simple software purchase and deployment — i.e., the restructuring of critical business processes — will estimate resource requirements more accurately than those who expect costs to resemble those of a server upgrade.

\[
\begin{array}{|l|l|}
\hline
& \text{The organization allocates resources according to classic, concrete ROI metrics, and funds access mediation similarly to how it funds software deployment efforts.} \\
\hline
\text{HIGH} & \text{Executives recognize that access mediation is a process, not a product, and budget accordingly. They make funding decisions based on long-term security posture rather than short-term return on investment numbers.} \\
\hline
\text{LOW} & \\
\hline
\end{array}
\]

10.6.3 Trust and resource flexibility (W)

An access administrator cannot always predict what resources she will need ahead of time — and it can be the timely response to unpredictable circumstances that determines the success of an access-mediation system. At the same time, organizations with limited resources are hesitant to offer arbitrarily abundant funding to teams without some ROI guarantees.

One investment bank addressed these dueling needs by employing 1) a small,
highly-trusted team of specialists responsible for identifying infosec problems and building appropriate tools, and 2) teams tasked with the regular operation and maintenance of those tools. The limited-engagement methodology of the first team restricted the resources dedicated to higher-risk IT ventures, but the adoption by operational teams allowed for regular budgeting and resource allocation for the experiments that worked out.

Budgets are strict and are rarely changed once approved. Previously unanticipated resource requests for improving access-mediation systems are not granted except in cases where there is a clear and significant security threat. ⇐⇒ Access administrators have built a positive reputation among the organization’s executives, and are quickly afforded wide resourcing latitude when they request it.

10.7 Example application: healthcare

I now use the characteristics described in previous sections to offer a preliminary profile of a prototypical American teaching hospital and its management of clinical (as opposed to research or administrative) data. This exploration is informed by my fieldwork in Teaching Hospital, but is intended to be broadly applicable to other similar organizations in the sector.

I offer this analysis as an initial example of how organizational profiling could be useful to practitioners who seek to identify suitable access-mediation systems for their organizations, and — in combination with the contrasting analysis of a hypothetical investment bank in the following section — useful to the research community in understanding the broad diversity of access-mediation needs among real-world orga-
Figure 10.2: Profile of a prototypical teaching hospital, particularly its use of clinical data. The organization is rated as “low”, “medium”, or “high” for each characteristic (labeled A-W and described in Sections 10.1-10.6). Where a characteristic is likely to vary greatly from one organization to another (or among departments within an organization) it is rated with a larger range (“low to medium” or “medium to high”).

As noted in Chapter 11, future work is needed to identify a concrete process to help organizations develop, validate, and make use of this type of profile.

Figure 10.3 depicts a graphical representation of the profile presented in this section. For each characteristic described in Sections 10.1 through 10.6, the organization is rated as “low”, “medium”, or “high”; characteristics that are likely to vary greatly from one organization to another — or among departments within an organization — are rated with a larger range (“low to medium” or “medium to high”).

10.7.1 Consequences of false positives

While I recognize that medical data is sensitive, and believe that all healthcare organizations should strive to protect it from inappropriate access, my fieldwork observations lead me to rate a teaching hospital as being “medium” or “low to medium” on each of the false positive scales.

I rate the probability that damage will result from a false positive verdict as
being “medium” because there are few cases in which a malicious user could reap significant financial or other personal benefit from gaining access to clinical data.\textsuperscript{3} One exception could be medical identity theft, in which one person uses another’s identity to gain access to medical care. Another exception is the case of celebrity patients; the EMR system used by the teaching hospital from my fieldwork allowed hospital administrators to place an extra layer of control on the records of patients deemed to be of public note.

In cases where damage does occur, I rate its gravity from the perspective of the organization as being currently “low to medium”. Repercussions that a healthcare organization can face from a privacy breach include 1) regulatory fines or other sanctions from governmental or professional bodies, 2) private lawsuits brought by patients, and 3) reputation damage done by news coverage on the breach. In general, I get the impression that the state of information security in healthcare is rough enough that regulatory agencies target organizations who seem to have a fundamental disregard for patient privacy. Compliance with HIPAA is somewhat self-determined, in that an organization defines its own plan for protecting patient data, and is measured against how well it adheres to that plan. And, as always, the imposition of repercussions — either institutional according to regulations or ad-hoc, like a lawsuit — depend on the breach being noticed in the first place.

I also rate the permanence of damage done by false positives as being “low to medium”. Financial sanctions or settlements could potentially bankrupt an organization, but are more likely to represent recoverable damage. Regulatory probation periods can restrict an organization for an extended period of time, but are designed to help the organization integrate better information security practices into its regular operation, not to prevent the organization from accomplishing its mission. While

\textsuperscript{3}I am targeting principally clinical settings, since that was the focus of my fieldwork, but other types of data may be of greater value to potential attackers. For example, a hospital participating in a field trial of a promising new drug could be a target of industrial espionage.
a hospital’s reputation is important, I believe that most people would be willing to overlook a past privacy breach if they felt that the hospital provided good medical care.

### 10.7.2 Consequences of false negatives

I rate the scope of the obstruction resulting from a false negative verdict in a clinical setting as being “medium”. As noted earlier, this property is highly dependent on the size of the system and whether it uses user groups or roles. In healthcare, large organization-wide systems that implement RBAC could easily experience a large scope of false-negative obstruction, whereas specialized departmental systems with few users would be likely to have a smaller such scope.

I rate the magnitude of the obstruction to be “high”. If a clinician is providing time-sensitive care and is unable to get access to the data she needs, the repercussions of the false negative on the patient’s health could be fatal.

I also rate the time to correction as “high”. I observed in my fieldwork that clinicians are incentivized (by the sum of their situation, not by an intentional design of policymakers) to work around the access-mediation system rather than pursue a fix through official channels. The time to correction will thus be delayed by users not reporting false negatives, and by organizational cultures that tend to reject user feedback as simply “users not complying with policy”. A further delay may be introduced if the user needs to receive explicit approval from one or more supervisors within the organization.

I estimate the relative cost of correction as being “medium to high”, and attribute it in large part to the general bureaucratic overhead (which are reflected more directly in the “resources and budget” category below) — does the administrator responsible for implementing the correction have all the data she needs to be sure it is an appropriate change? How many different stakeholders have to take part in the decision-making
process? Beyond the decision making process, the decision implementation also includes a certain amount of cost: how usable is the policy-authoring tool? If there is a QA process to validate the sanity of a new Portunes mapping with respect to a previous version, is that evaluation automated in any way, or completely manual?

Of course, the cost of an individual correction may be more on the “medium” side if the administrator simply adds a one-off exception to the mapping. In this case, however, the probability that the correction will percolate throughout the structured mapping such that it “sticks” for future users in similar situations is small — thus a “high” ranking for “one-off corrections”.

10.7.3 Organizational change

I rank the employee turnover rate for a teaching hospital as “medium”. Many clinicians and support staff may work at the same institution for decades, but this longevity is tempered by the constantly cycling population of students. I do not have data to back up this assumption, but it is something that an individual organization could easily measure about itself.

I rate the functional turnover as being “medium”. The student/long-term employee disparity will impact this metric among clinicians as it does the overall turnover rate; I imagine that this rate of change would be somewhat lower for support staff, but this remains to be verified in future work. However, I rate the functional dynamism in healthcare settings as being generally “low”, because the job descriptions of a user in a particular function (e.g., a dermatology nurse) are not likely to change much over time.

Finally, I rate the technical change experienced by the organization as “medium”, and here again it is an average of two extremes. On one hand the goals of clinicians’ workflows do not change rapidly, and the time and energy a hospital must put into the selection and deployment of a new large system makes the amount of technical
change small. On the other hand, the number of semi-autonomous departments within the hospital means that new small systems may come online frequently across the organization.

10.7.4 User accountability

Consequences for inappropriate action in a healthcare context could include incremental steps toward termination of employment (including verbal or written warnings), termination, revocation of one’s medical license, civil lawsuits, and criminal charges. All of these options would be reasonable if the action compromised the quality of the medical care that the organization provides. Large privacy breaches could warrant termination or civil lawsuits. However, only incremental steps toward termination seem appropriate for small privacy breaches (which my fieldwork indicates are likely to constitute the majority of inappropriate accesses), which are hard to detect.

For these reasons I rate the availability of consequences in a clinical settings as “low”, consequence proportionality as “medium”, and consistency of consequences as “low”. My fieldwork indicated that users were aware of cases where they were acting against the organization’s policy (like writing down passwords), but did not express awareness of how they might be held accountable for this choice. I thus also rate user awareness of consequences in clinical settings as “low”.

10.7.5 Contextual information availability

Based on my fieldwork, I believe that the structure of a teaching hospital — distributed into departments, each of which has specialist computer systems — makes it difficult for the organization to develop a good understanding of itself in detail. This specialty-oriented structure means that a “nurse” in one department may have a very different job description from one elsewhere. For these reasons, I rate categorize organizational understanding and function homogeneity as both being “low”.

213
The practice of health care is full of extenuating circumstances. Unexpected twists in a patient’s condition, the need for consultation by colleagues in other departments, and other factors make it difficult to predict what contextual information will be necessary to make a meaningful access decision. I thus categorize the simplicity of necessary contextual data as being decidedly “low” in a healthcare context.

The large amount of potentially-relevant contextual data combined with a poor degree of self-understanding makes it difficult for a teaching hospital to identify contextual data in the first place (much less make much contextual data available to the system in an automated fashion), so I also rate the availability of relevant contextual data in an access-mediation system as being “low”.

10.7.6 Resources and budget

I believe that healthcare organizations make patient care their top priority. Unless otherwise mandated by regulatory bodies, I believe that they will tend to place a “low” priority on resourcing access-mediation efforts. Given limited resources (included staff to perform analysis on security technology), I believe that these organizations are likely to adopt more classic metrics for judging the impact on patient care and the organization’s bottom line, and therefore rate their use of progressive metrics as tending to be “low”. I believe that the limited resources tend to also drive down the flexibility a healthcare IT group has with respect to resource use, but that the small size of IT organization can also allow them to foster a strong working relationship with the leadership. I thus rate the IT trust and resource flexibility in a healthcare setting as being “low to medium”.

10.7.7 Implications

The low ratings on false positive properties combined with the high ratings on false negative properties indicate that a teaching hospital would be a good candidate for
an access-mediation system that employs ex-post verdicts rather than ex-ante verdicts. This is further supported by the low levels of organizational understanding and context availability, both of which would make it hard for the organization to craft meaningful ex-ante policies. However, the low levels of user accountability would seem to indicate that a post-facto solution would not be appropriate: defaulting to `ALLOW` seems imprudent if one cannot incentivize users to choose to act in the best interests of the organization. On the other hand, the medical community has dealt with a situation where users (clinicians) are provided with relative autonomy in making decisions about patient care, and seems to be able to hold them accountable when they take damaging actions. I believe that this accountability comes less from the potential punishments of noncompliance, and more from the cultural norms imbued during the long process of user training.

As with any ex-post system, it would be important for the organization to be able to have sufficient data about the circumstances under which a particular access was performed in order to judge whether that access was appropriate. Given that the data necessary to make a decision is neither simple nor particularly available, it would be difficult for the access-mediation system itself to paint a sufficiently rich picture. Here, too, the medical community’s experience with evaluating the propriety of medical decisions could help an organization in its efforts to evaluate the propriety of a particular access.

Unfortunately, the organization’s generally “low” resources could indicate that it alone would not be able to develop sufficiently powerful auditing tools to analyze access logs and identify potential anomalies for further evaluation by humans. As I observed during my fieldwork, when the tools are insufficient and the number of access decisions very large, it can become impractical for an organization to conduct audits at scale (and lead them to only check access records when a complaint is filed). However, as discussed in Chapter 2, recent work (e.g., by Malin et al.) offers hope
In this section I offer another example profile, this time of a hypothetical investment bank. To avoid repetition of observations stated in the previous section, I focus primarily on the differences between an investment bank and a teaching hospital.

As with the previous example, this profile is extrapolated from my fieldwork observations, but not meant to be a profile of a single participant organization. Future work is needed to expand and refine this profile based on the experiences of practitioners from a variety of investment-banking institutions.

10.8.1 Consequences of false positives

In comparison to healthcare, false positives are generally more concerning to investment banks. I rate the probability of damage as being “medium to high” — medium because the sheer number of systems and entitlements mean that false-positive verdicts probably happen relatively frequently without causing any damage, but high because when users to get access to information they could exploit, the potential...
financial upside is so large that they are strongly incentivized to take advantage of it.

Similarly, I rate the gravity and the permanence of damage as being “medium to high”, because potential damage can range from minor corruption of data by unwitting users to the bank losing millions or billions of dollars — which can cause unrecoverable damage to the world economy if the bank is large.

10.8.2 Consequences of false negatives

In contrast to false positives, false negatives are somewhat less concerning to investment banks than they are to teaching hospitals. (Ironically, investment banks spend significantly more resources trying to reduce false negatives, these institutions put a high value on user productivity, in part because of the high opportunity cost when users’ actions on the financial markets are delayed even briefly). This is in part because the increased complexity of financial systems mean that there is likely to be a greater diversity of obstructions that can be experienced, depending on the type of data for which a false negative is delivered — unlike in healthcare, where simple systems (with relatively coarse access granularity) mean that a false negative can prevent a clinician from doing her job (i.e., accessing a patient’s record) altogether.

The scope and magnitude of obstruction depend heavily on the data in question, but I rate them generally as “medium to high” for investment banks. Factors that can increase the scope for some types of data include the matrixed interdependence between users in fulfilling their jobs and legally-mandated use of roles. The magnitude of the obstruction can vary from preventing a single user from performing a non-essential task to costing the company large amounts of money because a critical trade cannot be executed in time.

The time (“low”) and cost (“medium”) of correction depend heavily on the access-mediation system in place, but investment banks seem to try and optimize for these factors. however, more complex system can result in a higher cost of correction,
because it is harder for an administrator to find a fix that will not break other parts of the whole. This, in turn, increases the probability of one-off corrections (“high”).

10.8.3 Organizational change

Despite the medical-student turnover rates, I believe that organizational change is generally higher in investment banking than in teaching hospitals, as it is more common in the financial industry for an organization to evolve into new sub-sectors, take over other companies, and adopt new technologies. Individual users in finance are also more likely to hop from company to company, change specialties, or shoot up the corporate ladder than those in healthcare. I thus rate overall turnover, function turnover, and technical change as all being “high”, and function dynamism as being “medium to high” (this latter depending on the nature of the work the user was initially engaged in).

10.8.4 User accountability

I believe that user-accountability factors in investment banks are approximately on par with those in healthcare. The availability of consequences is slightly higher (“low to medium”) because it is a more competitive environment for employees, and getting written up for doing something bad can have a larger impact on promotion or bonuses. Proportionality of consequences are lower (“low”) because although some investment bankers professional licenses, they are not as critical as medical licenses for finding work in the sector. Also, the potential upside to malicious users (stealing money or getting promoted for making the company money) is large. I rate the consistency of consequences as “low” because banks do not seem to have a good way for detecting many undesirable user behaviors\(^4\). Awareness of consequences, especially external

\(^4\)This may be due more to technological lack than sociological truth, as improved tools might make it easier for financial organizations to detect undesirable behavior. On the other hand, I think that these organizations have already invested heavily in such detection, over a long period — with
consequences like being arrested, are slightly higher (“low to medium”) because investment bank employees receive more compliance training about the repercussions of being caught performing illegal acts such as insider trading.

10.8.5 Contextual information availability

As with healthcare, I rate the an investment bank’s organizational understanding, functional homogeneity, and simplicity of contextual data as being “low”. I believe that the major contributing factor is the size and complexity of the organization and its systems, and the agility of its workforce. I rate the availability of contextual data as being “low to medium” which is slightly higher than in healthcare (“low”) because investment banks have spent more time instrumenting business processes to make some of this information available.

10.8.6 Resources and budget

I rate an investment bank’s resources and budget factors as being generally more favorable than those of a teaching hospital. The financial industry perceives a competitive advantage in the confidentiality of some kinds of information, but also sometimes takes a check-off-the-checkboxes approach to designing its access mediation systems; I thus rate its resourcing priorities as “medium to high”. Because executives are habituated to thinking about many parts of the business in terms of return on investment (but some IT leaders are successful at making the case for systems that cannot be adequately evaluated this way), I rate the progressiveness of an investment bank’s metrics as “low to medium”. The more sizable budget of a financial IT department and its greater importance in protecting critical resources requires investment bank executives to place more trust in their IT managers, and I thus rate the organization’s trust and resource flexibility as being “medium”, rather than “low” as in healthcare.

only mixed success.
10.8.7 Implications

Given that the risks associated with false-positive verdicts are perceived to be greater than those associated with false-negative ones in an investment bank, a primarily ex-ante approach does seem advisable. However, for situations where a false-negative verdict could cause significant damage if not corrected quickly, introduction of break-glass controls could also be helpful.\(^5\) The addition of reactive access mediation in some situations — where data ownership could be distributed among the user population, where the number of access requests would be small (probably no more than a handful per week), and where data owners could easily gain understanding about the requestor’s function in the company and reasons for wanting access — could also make the system better able to cope with changes as the organization evolves, at least for some resources.

However, if such hybridization were to occur, an investment bank’s complexity and dynamism first necessitate better tools for understanding itself and the actions of its users.\(^6\) Rather than buying complicated role-mining technologies in an effort to shoe-horn the complexity of access mediation into a pure ex-ante approach, these organizations would do better to invest in solutions that help them analyze the patterns of user access. These tools could then evolve to help monitor the introduction and operation of schemes with a more ex-post or ex-ante flavor. With time, these tools could also be used by large organizations in other sectors that do not have the same degree of capital to invest in their development, such as healthcare.

\(^5\)I expect that, once introduced, the trick would be to make sure that break-glass escalation paths were not applied in situations where the damage from false negatives would not actually be critical.

\(^6\)I actually think such tools are critical even if no hybridization occurred; as documented in previous chapters, it is too easy for an organization to have a false sense of security in an ex-ante solution without noticing that it has gone completely off the rails.
10.9 Conclusions

This chapter represents an initial taxonomy of the factors that determine the success of access-mediation efforts in the real world. I have constructed this list based on my fieldwork observations in partnership with a handful of large organizations; as discussed in Chapter 11, additional work is necessary to further validate this taxonomy.

Additional work is also necessary to help practitioners learn how to build a profile of their organizations, and how to apply this profile in understanding what mix of access-mediation approaches will help them meet their organizational goals. In particular, several of the characteristics outlined here will vary from department to department or resource to resource within an organization; a framework to assist practitioners in cataloging such diversity and then combining it into a comprehensive profile would seem to be an important tool in applying this work in practice.

However, even though much work in this space remains, the taxonomy presented in this chapter is the first of its kind. Combined with the re-framing of the access-mediation problem presented in Chapter 9, it helps us understand not only why organizations experience the problems I documented in my fieldwork, but outlines a path for such organizations to avoid those problems in the future.
Chapter 11

Future work and conclusions

I believe that there is significant work still to be done if the research community is going to help real-world organizations build successful access-mediation systems; I outline in Section 11.1 several lines of future inquiry to that effect. However, as I argue in Section 11.2, I also believe that the work described in this document represents a significant first step in the right direction.

11.1 Future work

I offer here but a small sample of the work that could be done to further the cause of high-quality access mediation in the real world. The vast majority of these ideas require an interdisciplinary focus, as they would require both technical understanding and the use of usability studies or qualitative methods to understand the complexities inherent to socio-technical systems.

11.1.1 Direct extensions of my research

I propose further exploration of the ideas I have presented in the previous two chapters. In particular:
Validate the generality of the Portunes function. I believe that the Portunes Function presented in Chapter 9 is a foundational framework that can be used to describe any access-mediation system. Future work is needed to validate its generality, and identify whether there exist models\(^1\) it cannot encompass.

Characterize hybrid systems. I have distinguished the three “flavors” of access mediation, but expect — or at least hope — that the majority of real-world systems will come to implement different flavors to protect different types of data. A break-glass control is a good example of a hybrid ex-ante/ex-post system, and Reactive Access Control hybridizes ex ante and uno tempore. Are these the only hybridizations of these flavor combinations? What other schemes can be devised, and in what situations will they be useful? What framework should we use to characterize hybrid systems?

Existence of “pure” systems. Is it even possible for a purely ex-post, uno-tempore, or ex-ante system to be successful in a real organization? Do any examples exist in the wild, or is it the case that all organizations actually employ a blend of the three approaches — that pure ex-ante systems are in fact hybrids for which only the ex-ante aspects have been recognized or supported by management tools?

Profile-taxonomy completeness. I have constructed the taxonomy of factors described in Chapter 10 based on my fieldwork observations, but further work is necessary to determine whether there are other relevant factors that should be considered by organizations, particularly those in industries I have not studied.

\(^1\)Note that when I say “models”, I really mean “potentially practical models”. I am confident that it is possible to define a formal access-control model whose implementation could not be described by the Portunes Function, but models that will only ever exist for the purpose of proving theorems and publishing papers do not interest me.
Profile-taxonomy application. My taxonomy describes a large number of dimensions on which organizations can differ from one another, and I have characterized how I believe investment banks and teaching hospitals diverge on some of these dimensions. Are there other types of organizations that can be characterized using these two profiles? Can organizations in general be characterized by a small set of profiles, or will they diverge beyond any recognizable profiling pattern?

11.1.2 Ex-post evaluation

Aside from extensions of my own work, I would like to see the community put much more energy into exploring the use of ex-post access mediation schemes.

Usable tools. What tools, techniques, or other strategies could we develop to help organizations perform meaningful ex-post evaluation? Of the techniques that have been developed, how might we apply a deeper understanding of human factors to make them more useful?

Understanding log ignorance. The organizations I studied do not analyze their access logs except when a breach is suspected. Is this true for many other organizations? What incentivizes or discourages an organization to engage with meaningful logs analysis? Are there improvements outside of tools themselves that could make the practice more common?

Recognize existing instances. I have documented some real-world examples of ex-post access mediation, but there are surely others. In particular, existing efforts to detect data breaches seem to hold similar goals and may even involve the same types of system logs. Could we not therefore view these and other information-security efforts as a form of ex-post access mediation?
Patient auditing. As described in Chapters 2 and 7, systems have been proposed to provide patients with access logs for their medical data, yet clinicians are concerned that patients would not understand or be able to make use of such information\textsuperscript{2}. Is this true? Could filtering or improved usability of interaction design help improve patient understanding of access information? How many patients would take advantage of such a feature? Would it give them greater confidence in the care they were receiving, or in the overall security of their medical data?

General stakeholder auditing. In a similar vein: how well does ex-post stakeholder evaluation work in general, particularly when the resources in question may be more sensitive from the organization’s perspective than the perspective of the individual user? Systems that rely on uno-tempore mediation will fail noisily, because users will complain when they cannot access something or are receiving inappropriate access requests. Will the failure state for ex-post systems involving distributed manual evaluation be too quiet to be effective?

Ease of use: ex post vs. ex ante. It seems to make intuitive sense that it is easier for an administrator to reason about the acceptability of an access after it has occurred (if they are provided with sufficient contextual information) than it is to write an ex-ante policy that accurately covers all corner cases. Is this true? Could one find a way to meaningfully compare the ease with which we complete each task?

Policy rewriting. Although I hypothesize that it is easier to evaluate an access after the fact, it may still be easier for an organization to write an ex-ante policy (perhaps an imperfect one) than it is to define an efficient post-facto evaluation procedure. Could we create a tool for re-writing ex-ante policies into ex-post evaluation

\textsuperscript{2}I am also concerned that patients would have trouble if provided with the raw data, but as I noted in Chapter 7, I believe that patients may be better equipped to evaluate some access-log information than clinicians give them credit for.
schemes? This would be particularly useful for an organization considering a switch from an ex-ante approach to an ex-post one.

**Feeding other schemes.** There are myriad ways in which a small amount of ex-post evaluation could go a long way toward improving ex-ante or uno-tempore parts of an access-mediation system. Of course, the results of an organization’s ex-post analysis should feed into its ex-ante policy (but doing so in a reliably automated fashion is an open problem). Ex-post review of distributed uno-tempore access grants could help central administrators gain confidence that members of the organization are being judicious in authorizing access. Further work is necessary to integrate these workflows into existing and future access-administration systems.

11.1.3 Logs analysis

**Logs access** Although some researchers have been able to obtain organizations’ access logs for study (as summarized in Chapter 2), broader access to more data is necessary if researchers are to understand users’ access patterns — or identify meta-patterns that may exist across different organizations and industries. This involves a number of challenges, including data anonymization and some way to compare data from disparate sources. Could current efforts on logs analysis for data-breach detection be applied in this space, too?

**Organization self-study.** Once researchers have enough data to allow the development of meaningful tools, they might partner with organizations to perform self studies of access patterns. This, in turn, could further refine our understanding of the access-mediation problem, even for organizations that are not willing to share their data with external researchers.
Application to system definition. Given appropriate tools for studying an organization’s access logs, how should an organization go about translating its understanding into a useful access-mediation system? In particular, how could an organization currently practicing ex-ante mediation use logs analysis to identify places for mediation hybridization?

11.1.4 The Phalanx Problem

In Chapter 7 I defined the Phalanx Problem: how does one mediate access to digital resources not just by an individual, but by a dynamic group of individuals, whose membership is fuzzily defined? (The motivating example was a group of doctors and medical students doing rounds at a teaching hospital.) Wireless technologies such as Bluetooth proximity beacons might be used to detect the presence of clinicians, but other challenges remain.

Automated authentication. What tools can be developed to allow users to seamlessly join a phalanx — i.e., inherit at least a subset of the entitlements possessed by its members — without requiring them to be manually added to an access list?

Record-keeping and delegation. How does one record the actions of a phalanx when multiple members are present? Access logs that record an action taken by a resident are not meaningful if the attending doctor is the one who ordered the action be performed.

11.1.5 A grand challenge

Much of the work I have described in this section is driving at a deeper question: how does one measure the quality of an access-mediation system? How can an organization tell ahead of time whether a particular system will work well for them? My research
has documented many ways in which access-mediation systems work poorly today, but we still need an understanding of what it means for an access-mediation system to work well within a particular organization, and how to help organizations identify and choose systems that meet these criteria.

I suspect that over time we will come to understand that a “good” access-mediation system is one that is both correct (i.e., optimized to simultaneously facilitate desirable accesses and prevent undesirable accesses) and sustainable (i.e., optimized to minimize the setup and maintenance necessary to preserve correctness over long periods of time). I hope that future work will bear out my observations regarding the three flavors of access mediation, and that our community will find ways to help real-world organizations successfully hybridize these three approaches according to their individual needs and profiles — in part by recognizing the diverse ways in which uno-tempore and ex-post approaches are already being used.

11.2 Conclusions

To conclude, I now offer a brief summary of the contributions contained in this document, and a few last words of commentary. I structure the summary by grouping Chapters 2 through 10 into three broad themes: background observations, fieldwork report and analysis, and my reboot of access-control thinking.

11.2.1 Background content

Chapter 2 offers a broad overview of prior work related to my research, including observations on the development of access control since the “protection boom” of the 1970s and 1980s. It notes that the access-control models from that era were crafted to provide “provably secure” protection to critical military and intelligence information, at a time where the number of people interacting with digital data — and the amount
of data actually being protected — was small by today’s standards.

This chapter also notes that developments in access control since the protection boom have been (with a few exceptions) dominated by extensions to those first few models (MAC, DAC, and particularly RBAC), and built according to the same basic requirements of the (now decades-old) protection boom. Although the research community has been eager to propose new extensions, it has paid very little attention to validating whether these new models actually help real-world organizations achieve their access-control goals. More critically, almost no attention has been paid to understanding what organizations’ real access-control goals are, how well existing tools perform, or how one even goes about measuring whether an access-control solution performs well in a given setting.

As a precursor to the fieldwork chapters, Chapter 3 offers a survey of second-factor authentication devices. It also describes PorKI, a project from early in my graduate-school career, which sought to make users’ PKI-based authentication credentials portable by placing them on the user’s smartphone, and thereby make authentication via PKI more usable and relevant to real users. My goal in starting the studies described in later chapters was to try and build a similar solution — something that was more usable and relevant by real-world practitioners — for authorization management.

11.2.2 Fieldwork and analysis contributions

My fieldwork consists of two studies: a Corporate Study (described in Chapters 4 through 6) and a Clinical Study (Chapter 7). These studies use social-science methods to elicit information about access-control as it is experienced by administrators and end users at two large investment banks, one software-development company, and one teaching hospital. I interviewed and/or observed a total of 52 subjects across the four participant organizations.
Whereas other researchers have documented some challenges that real-world users experience with access control, the depth and breadth of my research offers a unique and valuable perspective on the state of the art as practiced in large, dynamic settings. My fieldwork observations lead me to believe that it does not suffice to say that access control is “challenging”: I believe that the problems I have documented to be so systemic, so deep, that it is more appropriate to say that access control is “fundamentally broken” for these types of organizations.

Some of the symptoms of this dysfunction that I present include:

- **Inevitable over-entitlement.** Operational realities (complex systems, unintelligible policies, and rapid organizational change) drive administrators of classic access-control systems to over-provision users, even if this is considered damaging by the organization.

- **Availability matters.** In some organizations and for some types of data, availability is more important than confidentiality (i.e., under-entitlement is more damaging than over-entitlement). Few solutions exist that recognize this prioritization.

- **Understandable circumvention by users.** Access controls prevent users from getting their jobs done, which drives users to circumvent the controls. Even non-technical users can find ingenious ways to break the system and achieve productivity, but the users who do so are rarely ill-intentioned.

- **Access control breeds a false sense of security.** Organizations remain blissfully ignorant of users’ circumvention, because the organizations have no scalable way of monitoring users’ actions. Access-control systems provide organizations (and regulatory bodies) a false sense of security.

- **Impossibility of self-understanding.** Organizations have very little understanding of themselves (e.g., what employees work on what tasks, who is re-
sponsible for what data). Organization dynamism makes it tricky to take a meaningful “snapshot” of the organization for an entitlement review, because by time the review is complete, the organization has changed out from underneath the reviewers.

- **Product vs. process.** Organizations expect to be able to buy software that will allow them to manage users’ access. However, some have learned the hard way that there is no perfect system; moreover, developing a good access-management solution requires iterative improvements and continuous investment — both in the technology and in the business processes — over time.

I argue in Chapter 8 that the present-day dysfunction is rooted in assumptions left over from the protection boom. The research community takes for granted that the requirements of that domain and era apply universally to organizations today, and this simply is not the case.

Some of the most damaging assumptions I identify — many of which conflict directly with my fieldwork observations — include:

- **Damaging assumption: organizational understanding.** The process by which an organization develops self-understanding may be somewhat expensive or time-consuming, but a comprehensive, centralized understanding is classically assumed to be at least possible.

- **Damaging assumption: organizations change slowly.** A slow rate of organizational change allows basic components that are expensive for administrators to update (e.g., the definition of what entitlements a role maps to) can change slowly, too.

- **Damaging assumption: confidentiality trumps all.** For the organizations that drove the protection boom, it was better to err on the side of under-entitlement than to risk over-entitlement.
• **Damaging assumption: a single policy.** It is classically assumed that it is possible to develop a single “correct” access-control policy for any organization.

Most importantly, I argue through these chapters that the dysfunction of access control is not the fault of the organizations or of their users. The fault lies with the research community, for not seeking to understand the full diversity of access-management needs experienced by these real organizations.

11.2.3 “Rebooting” contributions

In Chapter 9 I attempt to “reboot” our understanding of access control. Rather than continue to build on the protection-boom assumptions, I try to create a fresh and expansive definition of the problem that access control seeks to solve.

This includes abandoning the term “access control” in favor of the broader “access mediation”, which allows for schemes that seek to protect a digital resource by performing *ex-post* monitoring of users access instead of (or in addition to) asserting *ex-ante* control over it. Access mediation also allows for *uno-tempore* schemes that provide dynamic policy tuning, without the burden of ex-ante policy-writing or ex-post auditing. To accommodate this more expansive set of solutions, I introduce the *Portunes Function*, a generalized meta-model which describes access-mediation systems in terms of their data inputs and outputs.

The Portunes Function and its supporting terminology allow me to describe under a unified umbrella the three basic flavors of access mediation: ex ante, uno tempore, and ex post. Until now, discussions of alternative approaches (e.g., Optimistic Security, Reactive Access Control, and Break-Glass models) was largely disconnected from “classic” ex-ante understanding.

Of the Portunes Function’s inputs, I identify the *contextual data* as being critically important to making an accurate access-mediation decision (in either the ex-ante or ex-post timeframes.) The availability and manageability of contextual data were two
of the many factors that determine the success of an access-mediation scheme within a particular organization, and are included in the taxonomy of such factors that I present in Chapter 10. I also offer in this chapter an initial demonstration of how the taxonomy might be applied by an organization to help it understand its access-control needs. I believe that this preliminary foray into an organization to elicit requirements indicates that the three flavors of access mediation — ex ante, uno tempore, and ex post — might be blended to help organizations better blend their data-protection aspirations with the reality of their functional limitations. Furthermore, I believe that intentionally blending these approaches into a coherent whole would be far better than current informal or ad-hoc arrangements, which recognize only the ex-ante aspect as sanctioned, and thus allow other aspects to be practiced in an uncomfortable state of semi-approval.

11.2.4 Final words

In this document I catalog painful real-world experiences, call out false assumptions, and attempt to reboot a field that has been in development for the past forty years. In the first part of this chapter I identify directions for future work that I have uncovered during this research; among them I wish to emphasize that we — both researchers and practitioners — must focus more on understanding how users actually access data. If we were to invest (in both research-hours and organization-dollars) half as much in the development of innovative monitoring, auditing, or other ex-post technologies as we currently do in role-mining solutions, we might have a chance of making the reboot I propose successful: we might actually learn how to bring the full range of possible access-mediation solutions to bear in real organizations. Without data on users’ access patterns — for a large variety of resources, organizations, industries, and over a long enough period of time that we can watch the organization as it evolves — we are flying blindly. Without this understanding, we will be forced to stand idly
by while countless organizations invest staggering amounts of money in technologies that simply do not deliver what they promise.

Of course, even once we have taken the time to really understand users’ access patterns — even once we have made the critical and feasible improvements we have before us — I doubt that it will be possible to build a software product that makes the problem of managing users’ access to data easy. Access mediation is a problem firmly rooted in a socio-technical system, and its solution must therefore be sought not just among software products, but also among business processes. On some level I wonder whether this dual nature of the problem is why other computer scientists have not perceived the profound brokenness of current solutions until now: perhaps only through the (admittedly unplanned, in my case) application of both social-science and computer-science methods could one both recognize and diagnose what has been going so wrong for so long?

In that vein, I hope that other computer scientists who aim to study this problem seek out partnerships with researchers who are well-versed in social science methods. For any system that involves both humans and computers, it is critical that we inquire into the nature and operation of all the components, not just the ones with mathematically-provable properties.
Appendix A

An academic perspective on password policies

A.1 Common policy components

Password policies govern the entire password lifecycle, from creation to expiration. Password policy components fall in five broad categories, outlined below.

1. **Password Entropy**

   Passwords are knowledge-based authenticators; if an attacker guesses or deduces what a password is, the security of that password is compromised. We discuss the resilience of a password to this sort of attack in terms of the password’s entropy, or randomness [99]. In practice, password policies often substitute complexity for entropy, and proactively check users’ passwords to make sure they satisfy specific rules designed to increase password complexity.

   - **Password Length:** One way to increase the complexity of a password is by making it longer. The longer a password is, the more time it will take an attacker to crack (use a machine to repeatedly try to guess) it.
• **Password Content:** Another way to increase the complexity of a password is to require that it draws from a larger character set (i.e., numbers and punctuation in addition to letters).

• **Human Words:** Because it is easier for human users to remember strings of characters that are memorable or meaningful, such as words, many password cracking programs start by performing a *dictionary attack* against a password. Dictionary attacks often first target words that humans frequently use as passwords (e.g., “password”) and other strings derived from these words (e.g., “password123”).

All efforts to increase password complexity (or, ideally, password entropy) ultimately protect against the “cracked password” threat model, discussed below. **Increased password complexity does not protect against attacks involving recorded, phished, or reused passwords.** Many security researchers agree that these latter attacks are more concerning to organizations today than the traditional “cracked password” threat model.

2. **Login Attempts** An attacker rarely knows exactly what a user’s password is, and must thus try out a variety of different passwords before stumbling on the right one. (Attacks that rely on actually cracking a password require hundreds of billions of guesses.) Many password policies thus require that accounts with multiple failed login attempts be suspended or otherwise limited.

Limiting the number of failed login attempts can help protect against all of the threat models discussed below. While this sort of constraint is therefore critical, the number of failed login attempts that trigger suspension can be tuned to accommodate users who are forgetful or poor typists, without significantly compromising the security of the system [7].

3. **Password Change Frequency** The longer a password remains valid, the
larger the window an attacker has to perform an attack against it. **Requiring users to change passwords can protect against all the attacks discussed in Section A.2.** However, there is scarce hard evidence on the efficacy of mandatory password changes on improving system security, especially since frequent password changes encourage users to compromise password complexity in favor of memorability. This type of policy component therefore makes sense in principle, but it is **difficult to perform a cost-benefit analysis on its actual value in practice.**

4. **Password Re-Use** Some password policies require that users not re-use passwords across mandatory changes; if the user alternates between “1password23” and “7password89”, any benefit from the mandatory change is significantly reduced. However, **proactive password checkers usually cannot enforce rules against re-using passwords across systems;** this common practice can open a system to phishing and re-use attacks, as described below.

5. **Password Assignment and Reset**

   - **Initial Password Assignment:** Although initial password assignments may not be part of many organizations’ formal password policies, they can have a significant impact on the overall security of the system. (For example, assigning all users an initial password of “password123” could allow a knowledgeable attacker access to the system before an individual user has a chance to change her password.)

   - **Password Reset Mechanism:** The procedure an organization uses to reset a user’s password when she forgets it can also impact the overall security posture. Organizations that rely on human controls to enforce password reset policies (e.g., helpdesk staffers) are vulnerable to “social engineering” attacks. Following work that demonstrated vulnerabilities with relying on
the user’s mother’s maiden name as an authenticator (e.g., [33]), many organizations use a variety of “security questions” as part of the password reset process, or as a secondary authenticator in general. Some research has been done into the use of such questions by commercial banks, but the broader usability and long-term viability of this mechanism is not established [71].

A.2 Threat models

In building security systems, we identify attack scenarios that we want to protect against. We characterize these scenarios in terms of an adversary (traditionally named “Eve”) who possesses a defined set of data, and who is trying to accomplish a certain goal.

• The Guessed Password: Many users reason, “I’ll use my dog’s name and my daughter’s birthday. Eve won’t guess that!” Although there are cases of attackers successfully guessing passwords of high-value targets (e.g., Sarah Palin during the 2008 presidential campaign), the research community believes that this sort of attack is rare against average users.

• The Cracked Password: This is the threat model that the vast majority of password policy components strive to protect against. Password entropy makes this attack more difficult; as a result, the research community believes that higher-entropy passwords have forced attackers to turn to alternate attacks.

• The Recorded Password: As passwords have become more complex (and therefore less memorable), many users have tried to compensate by writing their passwords down. An attacker who has physical access to the user’s work environment may use these recorded passwords to gain access to the system.
Password policies that require users to push the boundaries of their memory capacity actually increase the organization’s vulnerability to this attack.

• **The Phished or Keylogged Password:** Unlike cryptographic keys, passwords have the “barn door property”: when Alice authenticates to Bob using a password, she has permanently revealed that password to him. Thus, if Eve can listen in on Alice’s authentication (e.g., using a hardware or software keylogger), or convince Alice to enter her password into a fake version of Bob’s login window (otherwise known as “phishing”), Eve knows Alice’s password until the next time she changes it. The phishing and keylogging threat models have quickly become a top priority for many organizations, and are generally considered harder to protect against than the traditional “cracked password” threat.

• **The Internally Re-used Password:** If an attacker knows or can discover a user’s old password, and the user re-uses that password across multiple mandatory password periods, the attacker has a higher probability of gaining access to the system using the old password. This may also be true for partial password reuse (e.g., “password1”, “password2”, etc.).

• **The Externally Re-used Password:** Users who re-use a password from one system to another may allow Eve to create a “pivot point”. With this model, Eve leverages her successful attack on a low-value system to compromise a higher-value system. For example, if Alice uses the same password for her Facebook account and her email account, Eve can gain access to Alice’s email by phishing her Facebook credentials.
A.3 Discussion

There exists a tension between the rigor of a password policy and the usability of the resulting system; academics have been discussing the associated tradeoffs for decades [62].) Recent studies indicate that requirements on password complexity in particular experience diminishing returns [16].

More generally, the academic community has come to doubt whether traditional password policy components really make sense in the current threat climate [26], especially given the rise of phishing and keylogging attacks over classic offline password cracking attacks.

Some work has been done to assist users in creating passwords that are both more secure and more memorable (e.g., mnemonic passwords [50]). While these efforts are interesting and promising, it is still not clear whether the improvement in overall security they bring is ultimately worth the time and effort to implement them [27–29].
Bibliography


