3-21-2011

Catch, Clean, and Release: A Survey of Obstacles and Opportunities for Network Trace Sanitization

Keren Tan
Dartmouth College

Jihwang Yeo
Dartmouth College

Michael E. Locasto
Dartmouth College

David Kotz
Dartmouth College

Catch, Clean, and Release: A Survey of Obstacles and Opportunities for Network Trace Sanitization

Keren Tan, Jihwang Yeo, Michael E. Locasto, David Kotz

March 21, 2011

Abstract

Network researchers benefit tremendously from access to traces of production networks, and several repositories of such network traces exist. By their very nature, these traces capture sensitive business processes and personal activity. Furthermore, traces contain operational information about the target network, such as its structure, identity of the network provider, or addresses of important servers. To protect private or proprietary information, researchers must "sanitize" a trace before sharing it.

In this chapter, we survey the growing body of research that addresses the risks, methods, and evaluation of network trace sanitization. Research on the risks of network trace sanitization attempts to extract information from published network traces, while research on sanitization methods investigates approaches that may protect against such attacks. Although researchers have recently proposed both quantitative and qualitative methods to evaluate the effectiveness of sanitization methods, such work has several shortcomings, some of which we highlight in a discussion of open problems. Sanitizing and sharing network traces, however challenging, remains an important method for advancing network-based research.

1 Introduction

The sharing of network trace data provides important benefits to both network researchers and administrators. Sharing traces helps scientists and network engineers compare and reproduce results and the behavior of network tools. The practice of sharing such information, however, faces a number of obstacles. Network traces contain significant amounts of sensitive information about the network structure and its users. Thus, researchers wishing to share traces must "sanitize" them to protect this information. We distinguish the terms "anonymization" and "sanitization": "anonymization" attempts to protect the privacy of network users, and "sanitization" attempts to protect the privacy of network users *and* the secrecy of operational network information. In contrast, freely sharing full-capture traces happens rarely and usually requires either close, pre-established personal relationships between researchers or extensive legal agreements (as in the PREDICT repository [51]). Furthermore, most real-world traces contain a large volume of information

Table 1: A Taxonomy of Anonymization-related Papers

Category	*Example*
Anonymization	Coull [18], Slagell [59], Zhang [65], Xu [61], Fan [26], Pang [48], Harvan [30], Zhang [66], Ramaswamy [52], Ylonen [62], Brekne [8], Overlier [46], Koukis [36], CPdPriv [44], TCPurify [6], CANINE [41, 42]
De-anonymization	Koukis [35], Brekne [8], Pang [48], Coull [18], Coull [15]
Evaluation	Coull [15], Yurcik [64]

with features along many different dimensions, making the problem of identifying and masking sensitive data non-trivial.

It remains difficult to precisely specify a policy regarding the type and structure of information that should be sanitized, let alone provide a reliable method that ensures the conclusive suppression of such information in the shared trace. Thus, two main categories of concerns arise: (1) legal and ethical obstacles to capturing information derived from human interaction for research purposes and (2) operational difficulties arising from a lack of effective tools and techniques for suppressing sensitive information. In this chapter, we survey a selection of both seminal and recent papers to summarize the reasons for these concerns, identify the work that has been done to help address or overcome them, and frame what we have come to view as the next major problem in this space: the invention of metrics describing the *quality* of a particular sanitization or anonymization technique on a given dataset.

We find that network researchers face a dilemma: although they can hypothesize about network data properties and prototype sanitization tools, they find it difficult to obtain real network traces to test these hypotheses and tools and verify whether they are correct, or to operate with any utility on real-world networks, respectively. Fortunately, network research is far from stagnant because researchers have put a significant amount of effort into (or find creative ways of) obtaining access to large, meaningful traffic traces from real production networks.

1.1 Challenges for trace collection and sharing

The daunting challenge of creating and maintaining a network monitoring infrastructure involves obtaining legal and administrative approval, reaching out to the campus or corporate community, implementing extensive security and control measures, maintaining internal records and documentation, and (sometimes) undergoing external security audits (see Section 6). This investment of time and effort can restrict the ability to capture meaningful amounts of network data to larger or well-funded organizations. In such an environment, *sharing becomes an essential feature of networking research*. Yet, the legal, ethical, and privacy issues of capturing and sharing production network traces threatens to chill such sharing and to eliminate this form of applied research.[1]

Many of the relevant laws are unclear about the legality of capturing and releasing network traces [58]. Even if such laws were amended to include specific exceptions for research use of network traces, as some advocate [11], individual privacy would still need protection and organizations would still wish to protect operational details. For example, network administrators may wish to share data for operational, not research purposes, but privacy concerns remain. Moreover, the network operator may wish to protect other information of proprietary or operational significance, such as the structure of the network, the identity of important servers, or how the network itself responds to particular types of threats.

We recognize the inherent trade-off between **privacy** and **usefulness**. Sanitization methods intentionally degrade the quality of a network trace to protect against trace users who actively seek to extract sensitive information from the trace, and inevitably reduce the type and content of features useful for non-malicious research. It is difficult to simultaneously achieve privacy and usefulness. A relationship exists between the amount of information shared and the level of risk an organization or individual assumes in sharing that information. *Methods of sanitization or anonymization seek to bound the level of risk as information sharing increases, but they can also bound the utility of the resulting data.*

1.2 Real-world network trace sharing efforts

Although they may seem abstract, privacy concerns are far from theoretical, and recent incidents involving real data sets have increased such concerns. The release of and subsequent de-anonymization attacks against

[1] Some point out that simulation provides an alternative to using real traffic data. For certain types of research (*e.g.,* anomaly-based intrusion detection), simulation is unlikely to prove useful, as the details of a real data sample are important, not just those properties derived from aggregate statistics.

the AOL data set [29], the release of the Enron email archive [25], and the de-anonymization attack on the Netflix competition data set [45] show how easily simple methods of content anonymization can be broken and highlight the risk posed by data once considered "private" or confidential.

Yet, the utility of sharing traces is so compelling that several efforts exist to share varying amounts and types of network trace data, including CAIDA, CRAWDAD, and PREDICT.

CAIDA (Cooperative Association for Internet Data Analysis) [12] collects several different types of network data (including topology, security, traffic characteristics, routing, real time monitors, and performance related data) at geographically and topologically diverse locations. CAIDA makes this data available to the research community while preserving the data donors' privacy. Currently its data repository has more than 230,000 data files. DatCat [21] is a CAIDA project providing the Internet Measurement Data Catalog (IMDC), a searchable registry of information about network measurement datasets. It aims to provide a searchable index of available network datasets and to enhance the documentation of the dataset via a public annotation system.

CRAWDAD (Community Resource for Archiving Wireless Data At Dartmouth) [19] provides a collection of trace data from wireless-network and mobile-computing researchers around the world. As of July 2010, CRAWDAD.org has over 2,337 users from 73 countries. It now makes 60 data sets and 23 tool sets available through the archive, with several more in the pipeline. Over 200 papers have been published with or about CRAWDAD data sets. In addition, a Dartmouth-wide wireless monitoring infrastructure has contributed to the CRAWDAD repository since 2001.

PREDICT (Protected Repository for the Defense of Infrastructure Against Cyber Threats) [51] is sponsored by the Department of Homeland Security (DHS) Science and Technology (S&T) directorate. The datasets in this repository, which include security-relevant network traces and host logs, are only available to qualified cyber defense researchers and only in the United States.

The DSHIELD [24] repository of firewall logs is one of the earliest examples of sharing intrusion alert information. Newer sharing efforts like OpenPacket.org tend to have a more limited number of datasets.

1.3 Terminology

Since this chapter assumes a focus on network traces, rather than other types of data collections (notably databases), our terminology reflects this bias by referring to packets, headers, and other network-related terms. Within the world of network traces, however, many specific types exist (such as SNMP logs, IP packet dumps, or Netflow traces), each with their own organization, data types, and information peculiarities. The content of a network trace includes not only the information from a network protocol (such as IP and MAC addresses, or port numbers), but also other metadata such as timestamps, session duration, or a wireless device's geographical coordinates. Such diversity of network traces makes it difficult or impossible to construct a universal algorithm for uniformly sanitizing all types of network traces. Moreover, due to a lack of understanding of (or documentation about) what information a trace might contain, trace sanitization can be much more challenging than it might initially appear.

We assume a general model for a network *trace* that holds a series of *records*. Each record contains a tuple of several *fields*. Each component represents a specific *feature*, such as source and destination MAC address, source and destination IP address, and timestamp.

Sanitization techniques can be applied independently to specific fields or sets of fields, and can include intra-record methods (hiding correlations between fields of a single record), inter-record methods (hiding correlations between multiple records in a trace), and inter-trace methods (hiding correlations between traces captured from different devices or at different times). Section 2 gives a detailed review of the sanitization techniques, especially IP-address anonymization techniques, and introduces several state-of-art network trace sanitization tools.

De-sanitization techniques extract sensitive information from the sanitized network traces. These techniques can be classified into two categories: direct de-sanitization attacks and indirect de-sanitization at-

tacks. While a direct de-sanitization attack exploits the flaws and limitations of some sanitization techniques, an indirect attack often leverages implicit information from the sanitized trace or auxiliary information from other sources. As an interesting example of the indirect attack, a CRAWDAD user suggested that the characteristic scanning behavior of a well-known Internet worm could be used to reverse the anonymization of IP addresses in some traces. Section 3 introduces current de-sanitization techniques and demonstrates several sucessful de-sanitization practices.

1.4 Database sanitization and privacy-preserving data mining

A large body of research has been also conducted from the database and data mining community on sanitization metrics (and techniques) and privacy-preserving data mining [1].

Anonymity is widely used as a key measure of privacy in sanitized databases [49]. One specific anonymity metric is "k-anonymity" [55], which in the database setting is defined such that a system provides k-anonymity protection if each record in the database cannot be distinguished from at least $k-1$ other records, with respect to every set of quasi-identifiable non-sensitive attributes. Machanavajjhala et al. demonstrate some severe problems with k-anonymity, however, especially when the attacker uses background knowledge, and propose "l-diversity" as a more powerful privacy definition than k-anonymity [43]. Li et al. show some limitations of l-diversity, in that it is neither necessary nor sufficient to prevent attribute disclosure, and propose a privacy notion called "t-closeness" that protects against attribute disclosure [40].

Although from these metrics we may gain some insights for network trace sanitization, they have made some assumptions that are specific to the database setting. For example, each of these metrics assumes that the set of "sensitive" attributes are known *a priori*, which is difficult to assume for network traces [15, 17]. Moreover, the metrics are purely *static* and *syntactic* in that they only consider the distribution of attribute values in the sanitized database and do not aim to capture the *dynamic* change of the adversary's *semantic* knowledge [9].

Instead, Shannon's entropy is often used as a simple indicator of anonymity and a measure of the adversary's knowledge gain in a network trace. Many information-theoretic metrics have been proposed [22, 56, 14, 15], including the degree of anonymity [22, 15] and the measure of the adversary's knowledge gain [15].

Producing sanitized data that have "good" utility for various data mining tasks is an important research goal in privacy-preserving data mining [9]. There are two approaches for measuring utility: a workload-independent measure, i.e., a utility measure that can be used for any data mining tasks, and a workload-dependent measure. Although workload-independent measures of utility are ideal for broader uses of published data sets, they inevitably use "workload-independent" or "syntactic" properties, such as the amount of generalization and suppression [13], average size of quasi-identifier equivalence class [43], or preservation of marginals [33]. Such "syntactic" measures, however, are of little use for some specific data mining tasks such as classification algorithms and therefore several workload-dependent utility measures such as accuracy of data-mining algorithms have been also studied [9, 31, 39, 60].

We believe that both workload-independent (syntactic) and workload-dependent (semantic) approaches are applicable to *usefulness* metrics for network trace sanitization. For specific applications like network security analysis, some approaches define and exploit a workload-dependent usefulness metric [64, for example]. However, there is limited research on workload-independent metrics for the usefulness of a trace for network analysis. We discuss more details about the usefulness metric for network sanitization in Section 4 and 5.

1.5 Chapter organization

As network and security researchers, we have faced many obstacles, challenges, and problems in our efforts to share network trace information with others, be it wireless frames or intrusion alerts. Our experience building and maintaining CRAWDAD has shown us the promise of trace sharing. Similarly, our experience building the 200-sniffer Dartmouth Internet Security Testbed (DIST [23]) informs our opinion about the

cost to create such systems and their utility as a shared infrastructure for a wider community. Our experience led us to want a deeper understanding of the issues involved in safely sharing network traces.

We organize this chapter to reflect the structure of our own foray into this topic: a progression we hope will ease the reader's journey. We start by identifying other overviews of sanitization techniques and selecting those we believe provide a novel perspective. In particular, the work of Ohm et al. highlights the legal issues surrounding network monitoring for research [58]. That paper serves as a wake-up call for the wider networking community, because collecting and sharing network data has several subtle pitfalls that tend to get overlooked simply because computer scientists are rarely trained as social-science researchers or legal experts.

Gattani et al. define a comprehensive reference model that can capture anonymization problems [28]. They introduce the notion of *universal information*, which is the complete truth regarding the users and the network where the trace was recorded. They show that the raw trace is only a subset of the universal information and as such cannot contain the universal information in its entirety. They propose a new entity set consisting of *collector*, *auditor*, *analyst*, and *adversary*, where the *auditor* was missing in the traditional entity set. They define the auditor to be an entity internal to the organization, who works with the collector to guarantee the privacy, accuracy and usability of a sanitized trace. As the only entity that can access the universal information, the auditor emulates the role of an adversary, as demonstrated by Coull [15]. Their reference model is reasonable, and they demonstrate its utility by applying it to Coull's work; the comprehensiveness of the model has not been verified with enough examples, however. Therefore, we do not use their model in describing and comparing a variety of problems and methods in this paper.

Kelly et al. survey the state of the art in metrics for precisely quantifying the information leakage from anonymized network data [32]. They offer a comprehensive summary of existing anonymity metrics and compare them in terms of *applicability* (whether a metric is useful for data privacy or communication privacy), *complexity* (whether the method requires substantial computation), and *practicality* (reflecting the trade-off between practicality and mathematical rigor). In this paper, we not only address the issues and problems of anonymity metrics but also on research of *usefulness metrics* that quantify how useful the sanitized trace is for the researchers to analyze the trace.

Porras et al. propose nine risks and challenges [50]. They group these challenges into three categories: network sensors that generate data, repositories that collect data and make them available for analysis, and the network infrastructure which delivers the data from the sensors to the repository. Similarly, Bishop et al. pay special attention to the interactions between the multiple actors (collector, analyst, and adversary) involved in a sanitization problem [5, 20]. Coull et al. suggest that the research on anonymizing census microdata may also provide several useful insights on how to effectively anonymize network traces [17].

These surveys served as a starting point to explore various themes in the field. We organize our own report into three main sections bracketed by this Introduction and an argument about three critical open problems for trace sanitization (Section 5). The main sections consider, in turn, sanitization techniques (Section 2), methods of attacking these techniques (Section 3), and current proposals for evaluating the strength of sanitization and sanitization effects on datasets (Section 4). We seek to highlight the coevolution between ways to perform sanitization [18, 59, 65, 61, 26, 48, 30, 66, 52, 62, 8, 46, 36, 44, 6, 41, 42], de-sanitization techniques [35, 8, 48, 18, 15], and methods of measuring [15, 64] the success of both such efforts.

Finally, we close with a consideration of various "gaps" in the space of sanitization and sanitization techniques. We posit that the largest such gap is the difference between the type of information sanitization tools operate on (and thus report on) and the type of information meaningful to a human operator to help them assess the quality of a particular sanitization pass over a specific dataset. Although several researchers [5, 50] note that network data sanitization requires methods that simultaneously protect sensitive or private information and preserve information useful for analysis, there has been only limited development of usable quantitative metrics for measuring privacy protection for network data (e.g., the degree of

anonymity [15]).

2 Sanitization

To share network traces while preserving privacy, the trace publishers draft sanitization policies according to their specific privacy concerns. These policies explicitly or implicitly determine which sanitization methods to apply and how.

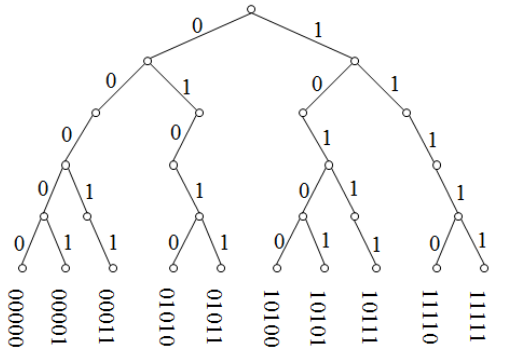
In this section, we review current research on sanitization, with a focus on techniques and tools. Here, "techniques" refer to specific methods or algorithms that solve a specific sanitization problem. Because different fields in the network trace possess different characteristics, they require different sanitization techniques; other techniques are needed to sanitize some inferable and implicit information, such as network topology. A sanitization "tool," on the other hand, provides a systematic solution for a range of applications. A sanitization tool usually implements a set of sanitization techniques and provides a convenient interface to its user.

2.1 Sanitization techniques

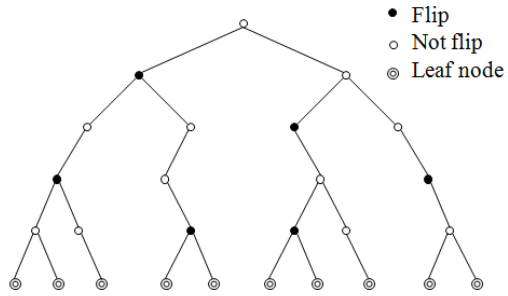
General techniques to sanitize specific network trace fields can be classified into a few categories [18, 59]: destruction, fixed transformation, variable transformation, and typed transformation. *Destruction* removes part of, or all, the information from a field, for example, complete removal of the TCP payload, or the removal of the least significant bits of the IP address. *Fixed transformation* uses a single pseudonym value to substitute all values appearing in a field, e.g., to replace the field with zero. Intrinsically this is same as destruction. *Variable transformation* provides more flexibility by using different pseudonym values according to the context of the field. One example is to substitute an original IP address with different pseudonym values according to the type of upper-layer protocols, such as HTTP or SMTP. *Typed transformation*, also called permutation in the most general sense, is a one-to-one mapping between a pseudonym value and a distinct value of the original field. "Prefix-preserving" address anonymization, a common technique, belongs to this category.

Among all the fields in the network trace, the IP address has received most research attention. There are several types of IP-address anonymization techniques based on different design considerations [65, 59]. IP-address partial destruction removes the rightmost IP-address bits, which identify an individual host on a subnet. Prefix-preserving anonymization (pseudonymization) is a special case of permutation that preserves the hierarchical nature of IP addresses and is often preferred to random permutations. There are two general classes of prefix-preserving IP address anonymization techniques: the strict bitwise-preserving approach [44, 61, 26], and Pang's "divide-and-conquer" approach [48].

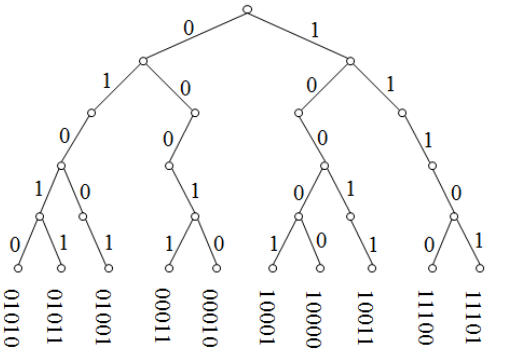
In the strict bitwise-preserving approach, two anonymized IP addresses will have a common n-bit prefix if and only if the un-anonymized IP addresses have a common n-bit prefix. Minshall implemented one approach to such prefix-preserving anonymization in TCPdpriv with the "A50" option [44]. Xu and Fan showed that such prefix-preserving anonymization functions all follow a canonical form. They proposed a cryptography-based, prefix-preserving anonymization technique, which is implemented in Crypto-PAn, without the need for a prefix table [61, 26]. A geometric interpretation of this prefix-preserving anonymization technique can be described as follows [30]. The collection of all possible IP addresses can be represented by a complete binary tree (see Figure 1). For IPv4 addresses the height of the tree is 32, and for IPv6 it is 128. Each leaf node of the tree represents a distinct IP address, and each non-leaf node corresponds to a bit position, indicated by the height of the node, and a bit value, indicated by the branch direction from its parent node. The set of distinct addresses present in the unanonymized trace can be represented by a subtree of this complete binary tree. This subtree is called the *original address tree*. A prefix-preserving anonymization function can be viewed as specifying a binary variable for each non-leaf node of the *original address tree*. This variable determines whether the anonymization function "flips" this bit or not. Applying this anonymization function to the *original address tree* results in an *anonymized address tree*. Based



(a) Original address tree



(b) Anonymization function fi



(c) Anonymized address tree

Figure 1: Geometric interpretation of prefix-preserving anonymization function: (a) represents nine addresses from a 4-bit address space as a binary tree; (b) shows a randomly chosen anonymization function, that is, a set of nodes in the tree are flipped to generate anonymized addresses; (c) shows the anonymized 4-bit addresses produced by applying the anonymization function from (b).

on Xu and Fan's work, Harvan [30] extended this algorithm to preserve SNMP's lexicographical-ordering property. Zhang and Li [66] observed that a trace is often used by different research groups at the same time. Since each group has a distinct trustworthy level, one network trace needs to be anonymized separately to fulfill each group's requirement. Thus, if there are n research groups, there will be n copies of anonymized trace from one original trace. They proposed a scheme that only generates one copy of an anonymized trace, but the users with different knowledge (secret keys) may recover different traces from this single copy. Ramaswamy [52] presented an online prefix-preserving anonymization algorithm– top-hash subtree-replicated anonymization– with low processing requirements and small space complexity.

Unlike the above bitwise-preserving approaches, Pang's approach [48] remaps the IP addresses differently based on the type of addresses, either *external addresses* or *internal addresses*. All *external addresses*– the IP addresses that do not belong to the trace publishing organization– are remapped using the IP address anonymization algorithm in Crypto-PAn. All *internal addresses*– the IP addresses that belong to the trace publishing organization– are divided into the subnet portion and host portion. These two portions are remapped independently and preserve only whether two addresses belong to the same subnet. This means that all hosts in a given subnet in the original trace will also appear in the same subnet in the anonymized trace. Note that this mapping does not preserve the relationship between subnets. For example, two 24-bit subnet numbers that share a 20-bit prefix in the original trace will not necessarily also have a 20-bit common prefix in the anonymized trace. Pang suggested that this anonymization approach can also be applied to the MAC address.

However, several researchers have criticized current prefix-preserving techniques. Ylonen demonstrates that the prefix-preserving anonymization in TCPdpriv with the "A50" option is not necessarily good enough to keep a well-informed adversary from determining where the data were collected [62]. In Crypto-PAn's prefix-preserving anonymization, any given bit of the anonymized address is dependent on all previous bits of the unanonymized addresses; Coull et al. argue that this kind of dependence causes a single deanonymization to affect all anonymized addresses that share a common prefix with the true unanonymized addresses [18]. Moreover, Brekne et al. present a set of attacks employing active packet injection and frequency analysis to systematically compromise individual IP addresses protected by the anonymization techniques implemented in TCPdpriv and Crypto-PAn [8, 46]. They propose transaction-specific anonymization schemes that use stream ciphers to encrypt each bit of an IP address and do not preserve the one-to-one mapping between the original and the anonymized IP addresses at all. By individually performing pseudorandom permutation on the subnet and host portions of internal IP addresses, Pang's approach reduces linkability among anonymized addresses more than Crypto-PAn's approach and is more robust against Coull's attack [18]. However, Coull shows that some sensitive information, such as network topology and network servers, can still be inferred from traces anonymized by Pang's approach [18].

2.2 Sanitization tools

Many network trace sanitization techniques have been proposed; some of these techniques are also implemented as software tools. As mentioned above, Crypto-PAn implements the cryptography-based prefix-preserving anonymization technique proposed by Fan and Xu [61, 26]. It uses the Rijndael cipher (AES algorithm) as its underlying pseudorandom function and has the following properties: a one-to-one mapping from unanonymized to anonymized IP addresses, a prefix-preserving mapping, and consistent address mapping across traces. TCPurify [6] is a packet-capture program with sanitization capabilities. After recognizing the Ethernet or IP header, it removes all data payload before storing the packet (except for certain protocols), and does a reversible randomization on IP addresses without preserving network prefix information. TCPdpriv [44] also anonymizes packet traces, with several options to process IP address and TCP/UDP port numbers. TCPdpriv provides prefix-preserving anonymization of IP addresses using a prefix table on a per-trace basis, and thus may not provide a consistent mapping: a particular address will likely be anonymized to different pseudonym addresses in different traces. CANINE (Converter and ANonymizer for Investigating Netflow Events) provides multiple format conversion utilities and integrates several sanitization methods, such as IP anonymization and timestamp sanitization, on NetFlow logs [41, 42]. AnonTool, an open-source implementation of a set of anonymization APIs, aims to build an anonymization assembly line, up to the application level, by expressing the anonymization policy as several sets of sequential function calls [36].

Compared to the sanitization tools above, tcpmkpub [48] and FLAIM [59] provide a more comprehensive and flexible solution. Both of them implement a generic framework for sanitizing network traces. These two frameworks have several common characteristics: (1) User-defined sanitization policies are de-

```
- <policy>
  - <field name="IPV4_DST_IP">
    - <BinaryPrefixPreserving>
        <passphrase>abracadabra</passphrase>
      </BinaryPrefixPreserving>
    </field>
  - <field name="IPV4_SRC_IP">
    - <BinaryBlackMarker>
        <numMarks>8</numMarks>
        <replacement>0</replacement>
      </BinaryBlackMarker>
    </field>
  - <field name="TS_SEC">
    - <RandomTimeShift>
        <lowerTimeShiftLimit>60</lowerTimeShiftLimit>
        <upperTimeShiftLimit>600</upperTimeShiftLimit>
        <secondaryField>NONE</secondaryField>
      </RandomTimeShift>
    </field>
```

Figure 2: PCAP header sanitization rules used in FLAIM

```
FIELD           (TCP_srcport, 2,       KEEP)
FIELD           (TCP_dstport, 2,       KEEP)
FIELD           (TCP_seq,     4,       KEEP)
FIELD           (TCP_ack,     4,       KEEP)
FIELD           (TCP_off,     1,       KEEP)
FIELD           (TCP_flags,   1,       KEEP)
FIELD           (TCP_window,2,         KEEP)
PUTOFF_FIELD        (TCP_chksum,       2,       ZERO)
FIELD           (TCP_urgptr,  2,       KEEP)
FIELD           (TCP_options, VARLEN, anonymize_tcp_options)
PICKUP_FIELD (TCP_chksum,      0,       recompute_tcp_checksum)
FIELD           (TCP_data,    RESTLEN, SKIP)
```

Figure 3: TCP sanitization rules used in tcpmkpub

scribed by a set of explicit rules using a dedicated language. Figure 2 gives an example of PCAP header sanitization rule used in tcpmkpub. The XML-based language used by FLAIM is called the *Module Schema Language*. A snippet of this language is shown in Figure 3. (2) The framework follows a modular design and is extensible. Many sanitization primitives and common algorithms, such as truncation and prefix preserving, are implemented and integrated into the framework. Users can also develop their new sanitization techniques as modules and plug these new modules into the framework. (3) tcpmkpub supports sanitization for multiple layers: link layer, network layer and transport layer. FLAIM supports sanitization for several types of logs. For each type of log, FLAIM implements a parser module respectively.

It is important to mention that all available sanitization tools can only do a "one-way job". That is, they can only sanitize a network trace, but they can not provide the user any feedback about the quality of sanitization, such as how much privacy information has been removed or kept in the trace. The goal of sanitization is to pursue a balance between protecting privacy and preserving trace's utility. Since no perfect sanitization techniques exist, choosing and tuning these techniques affects the final sanitization result greatly. As noted by many researchers [48, 59, 15, 61], exploring the relationship between the strength and utility of sanitization is an important task for future research.

3 De-sanitization

In security research, we often perform a *worst-case* analysis that assumes an adversary has almost unlimited resources and knowledge to launch an attack on the examined target. Due to the intrinsic complexity of network trace sanitization, we must admit that there are few, if any, available network-trace sanitization schemes that can provide a *water-tight* guarantee under the *worst-case* analysis. That we hold this opinion is not to degrade the value of sanitization research presented in Section 2 but rather to emphasize that

current de-sanitization research has been remarkably creative and successful.

According to the attack strategies employed by an adversary, we classify current de-sanitization research into two categories: *direct attacks* that exploit the limitations of an anonymization algorithm [8, 46, 18], and *indirect attacks* that use implicit information contained in the trace [18, 27, 2, 47, 38], auxiliary information obtained from other sources [38, 18, 47], and new techniques from other research fields, such as machine learning and pattern recognition [16, 7, 47, 2], to uncover sensitive information from the anonymized network trace.

One example of a direct attack exploits a flaw in Crypto-PAn [61]. As mentioned in Section 2, Crypto-PAn implements a strict bitwise-prefix-preserving anonymization algorithm, and any given bit of the address anonymized by Crypto-PAn depends on all previous bits of the anonymized addresses. This dependence enables a de-anonymization on one address to affect all anonymized addresses that share a common prefix with the true address [18]. For instance, if an anonymized address $a = a_1a_2 \ldots a_{n-1}a_n$ is deanonymized to reveal its true address $t = t_1t_2 \ldots t_{n-1}t_n$, then the adversary also learns that the anonymized address of another true address $t^* = t_1t_2 \ldots t_{n-1}t_n^*$. should be $a^* = a_1a_2 \ldots a_{n-1}a_n^*$. Because t and t^* have a common prefix of $n-1$ bits, their anonymized addresses a and a^* must also have the same $(n-1)$-bit prefix. Based on this idea, Brekne et al. proposed an attack against Crypto-PAn that uses packet injection and frequency analysis to compromise individual addresses in multilinear time [8, 46]. Pang's "divide-and-conquer" approach is regarded as an improvement over Crypto-PAn by processing the subnet and host portions of internal IP address respectively, and thus it decreases the linkability among anonymized addresses [48].

From a trace publisher's view, an indirect attack is probably more dangerous and much harder to defend than a direct attack. Although Pang's *tcpmkpub* [48] is regarded as one of the most state-of-art and comprehensive sanitization solutions, Coull's work [18] shows that this solution is far from enough to provide a "water-proof" protection for a lot of sensitive information. For instance, a "dominant state analysis" characterizes the behavior of each host and then classifies these hosts into logical groups, such as a possible server or an ordinary client, based on their behavior profiles. The subnet clustering algorithm takes advantage of the prefix-preserving anonymization to extract information about the underlying network topology. By associating the above information extracted from the anonymized trace with other auxiliary information, such as DNS records, SMTP traffic, ARP traffic and publicly available website information, their experiment shows that they can not only completely deanonymize some public hosts but also depict detailed traffic properties at some observation points.

Moreover, recent research has extended such indirect de-sanitization attacks to the wireless-network domain. Many researchers have shown that an IEEE 802.11 wireless device's chipset, the firmware or the driver can be identified by either passive fingerprinting [27, 2] (in which the adversary simply observes network traffic) or active fingerprinting [7] (in which the adversary sends out probes and observes the responses). Whether passive or active, these techniques work by building a database of the unique variations in protocol behaviors, as seen across different vendors or implementations, and later discerning the make and model of an unknown Wi-Fi network interface by observing this behavior in network traffic. Knowledge of the brand used by a Wi-Fi user may, when combined with other external information, allow an adversary to de-anonymize that user's traffic within a trace. Using auxiliary location information, Kumar's work shows the possibility to categorize Wi-Fi users based on their gender [38]. As a further step, Pang et al. demonstrate that by using so-called "implicit identifiers", such as network destinations, SSID probes, broadcast packet size and IEEE 802.11 MAC protocol fields, an adversary can accurately pin down user identities in some circumstances [47].

The above de-sanitization research shows that there is no one sanitization technique that can handle all situations. Any flaw in an anonymization algorithm can lead to disastrous privacy leakage. Beyond the robustness of the anonymization algorithm applied, a desirable outcome depends on the properties of the original unanonymized trace, such as the type and volume of implicit information contained in the trace.

To maximally defend against an indirect attack, the trace publisher should have a comprehensive view of the anonymization problem. This means that when sanitizing a trace, the trace publisher should not only focus on the trace itself but also take all auxiliary information into consideration. As shown above, the combination of auxiliary information plays a vital role in a de-sanitization. We regard the progress in network trace de-sanitization as a valuable and indispensable complement to anonymization research.

4 Evaluation of Sanitization

Methods to evaluate the efficacy of sanitization methods seem somewhat underdeveloped compared to the wide variety of actual suppression techniques and attacks. Most evaluation papers naturally focus on both quantitative and qualitative measures [15, 64, 10], but some consider the aspects of traces that can be exploited in the sanitization process. For example, the remote and local port features reveal more distinguishing information than the timestamp feature, and therefore re-examining the anonymization policy on these features may improve the efficacy of sanitization [15]. Settling on a particular metric of sanitization effectiveness is difficult, in part due to the variety of features in network traces. It is not immediately clear, for example, how one might meaningfully compare the sanitization of IP addresses with the anonymization of user browsing profiles.

There seem to be two broad types of metric. First, *sanitization metrics* measure how well the sanitization method has fulfilled predefined requirements. In view of the definition of sanitization, the predefined requirements may be those for privacy or secrecy. Second, *usefulness metrics* measure how well the sanitized traces remain useful to the researchers for the purpose of trace analysis.

Table 2 summarizes some representative evaluation papers with regard to evaluation metrics, evaluated sanitization methods, and evaluation methods.

Coull et al. [15] evaluate two well-known anonymization methods, CryptoPAn [61] and Pang [48], in terms of the privacy requirement. For the evaluation, they de-anonymize the sanitized data on a few selected fields to quantify the anonymity of the data. They defined the anonymity of each object (e.g., each host) with respect to a feature (e.g., port number) by calculating the entropy of the "similarity" distribution. For an object A, the similarity distribution consists of the probability $P_F(A, i)$ over N objects. This probability expresses how similar the object A is to an unanonymized object i with respect to the feature F. If the anonymity of the object A (in this case, the entropy of the object A) is close to its minimum value (zero), then there probably exists an unanonymized object that is similar to A with respect to the feature F. Otherwise, if the anonymity of the object A is close to its maximum value, $logN$, then the object is not more similar to any one unanonymized object than any other unanonymized object.

Yurcik et al. compare a variety of sanitization methods in terms of how they trade-off anonymity and usefulness [64]. They use the SCRUB-tcpdump network packet anonymization tool [63] to perform various anonymization methods on all fields of the data. The anonymization methods include replacing values with a predefined constant (*black marker*), mapping values to any valid permutation (*pure randomization*), hashing values with a small key (*keyed randomization*), and classifying values into groups (*grouping*). They examine, as the usefulness metric, the difference in the number of alarms from the Snort IDS, an intrusion-detection system [54], before and after a trace is anonymized.

They showed that some fields (Transport Protocol Number and Total Packet Length) have a zero-sum tradeoff, meaning that "the more network data is anonymized for privacy-protection, the less value the network data may be for security analysis" [64]. However, most of the other fields have a more complex tradeoff (not zero sum), suggesting that both privacy and usefulness can be achieved in certain cases.

More recently, Burkhart et al. investigated the tradeoff between data utility for anomaly detection and the risk of host identification for IP address truncation [10]. They evaluated the risk of de-anonymizing individual IP addresses using a metric based on conditional entropy [4, 37]. For measuring data utility, they compared the detection rates of the DoS and Scan attacks based on IP-based detection metrics (e.g., unique address counts and Shannon entropy) computed on the original traces with those based on IP-

Table 2: Representative evaluation papers

Paper	*Metric*	*Methods Evaluated*	*Evaluation Method*
Coull [15]	the anonymity of each object (e.g., network host) defined by calculating the entropy from the probability distribution on the object identity	CryptoPAn [61], Pang [48]	simulate adversary's behaviors and compare anonymization techniques by examining the impact on the anonymity of the data.
Yurcik [64]	tradeoff between privacy protection vs. research usefulness (security analysis capability)	black marker, pure randomization, keyed randomization, bilateral classification, and grouping	calculate the difference of intrusion-detection-system alarms before and after the anonymization.
Burkhart [10]	tradeoff between privacy protection (host identification) vs. research usefulness (anomaly detection capability)	IP address truncation	privacy risk is evaluated using a metric based on conditional entropy and usefulness is evaluated using ROC (Receiver Operating Characteristic) curve, i.e., true positive rate of anomaly detection.

based detection metrics computed on anonymized traces. According to their results, truncation effectively prevents host identification but degrades the utility of data for anomaly detection.

They found that the degree of utility degradation depends on the detection metrics used for anomaly detection (e.g., unique address counts vs. entropy) [10]. For example, the entropy detection metrics are more resistant to truncation than unique address counts because the entropy detection metrics better represent the distribution of flows per IP address than the unique address counts metrics, even when the IP addresses are truncated. They also noticed that the detection quality of anomalies degrades much faster in internal addresses than in external addresses. Specifically, the usefulness of internal address counts is lost even for truncation of only 4 bits while the usefulness of external address entropy is virtually unchanged even for truncation of 20 bits.

Research on methods to evaluate the efficacy of sanitization methods is obviously in its infancy, and many research questions remain. Among them, two issues draw our attention more than others: first, only a few evaluation metrics, either sanitization metrics or usefulness metrics, have been suggested that can precisely quantify the efficacy of network data sanitization. Second, even when a metric can give a precise measure of the sanitization efficacy, there may exist a large gap between the semantics of the metric and the semantics understood by users of the sanitization tool, or of the network trace. We discuss these two issues more deeply in Sections 5.2 and 5.3.

5 Challenges and open problems

Sanitizing network traces is about managing risk [48]. The amount of risk depends on both the trace publisher's policies and assumptions about the attacker's knowledge and capability. The trace publisher drafts sanitization policies according to his/her specific privacy or secrecy concerns, and these sanitization policies are mapped to a set of sanitization techniques and their configuration. Generally the trace publisher has a "benign wish" when sanitizing traces, that is, to preserve the trace's usefulness as much as possible.

However, there is a tradeoff between privacy and usefulness during sanitization. In choosing a sanitization approach, the trace publisher balances privacy and usefulness, informally evaluating the risk that an adversary will be motivated and capable of exposing sensitive information by leveraging benign information the publisher chooses to leave in the trace. Therefore, a top-level challenge for trace-sanitization research is to help trace publishers deal with the tradeoff between anonymity and usability.

In this section, we discuss several challenges to achieve this goal. The challenges include how to quantify private or sensitive information, what metrics to use for evaluating the sanitization result, and how to interpret the sanitization result.

5.1 Quantifying sensitive information

To protect personal or proprietary interests, the trace publisher would like to know how much private or proprietary information is contained in the trace. This may be difficult, however, for two reasons.

First, the boundary between sensitive and insensitive fields is obscure and changing over time. Some fields of a packet are obviously "sensitive," while others are not. It is well known that a port number is useful to identify a specific service, a MAC address is enough to identify a unique NIC, and an IP address may be useful to identify a specific host. For some other fields, the degree of sensitivity is not clear at first glance, but they actually may contain private information. For example, recent research shows that an attacker can fingerprint a physical host by using only the clock drift in TCP timestamps [34]. The length of an HTTP/TCP connection can identify the web server (if a well-known public server), and the order of IP addresses contained in SYN packets can be used to partly reconstruct the anonymized IP addresses [35]. The point here is that with the development of new techniques, fields that seem to be safe today may become sensitive in the future.

Second, in addition to the explicit values described in each field of a packet or an entry in a log, there is information "implicitly" contained in the network traces. For example, such information includes the traffic pattern of a host, the topology of the traced network, and the mutual relationships between hosts. Previous sanitization research mainly focused on anonymizing explicit values, and neglected this implicit information. As a result, some de-sanitization techniques, such as dominant state analysis and subnet clustering [18], can dig out valuable information.

We think a great amount of information exists intrinsically in the traces or is intentionally preserved by the specific sanitization technique, such as the network topology discovered by subnet clustering but preserved by Pang's prefix-preserving method [48]. Although trace publishers who intentionally preserve such useful information may be willing to take the risk of de-sanitization, others may not realize that they are preserving such information, or that new methods can extract more than they expect. Nevertheless, we regard this kind of de-sanitization research to be important, if only to inspire new sanitization methods and to help determine the privacy bounds of those methods.

5.2 Metrics for evaluating sanitization results

Trace publishers sanitize their traces to achieve both their "sanitization" goals (to protect both personal and operational information) and "usefulness" goals (to protect research value); after sanitization, presumably, they would like to know whether their goals are actually achieved. To evaluate whether their goals are achieved, we need metrics for measuring the degree of sanitization and usefulness of the sanitized traces. It would be even better if these metrics could also be used to help control the tradeoff between the degree of sanitization and usefulness.

Although several generic "anonymity" metrics have been suggested [55, 22, 56, 14], and some were specifically suggested for network traces [15], we have yet to find any generic metric for the "usefulness" of a trace for network analysis. For specific applications like security analysis, some approaches define and exploit a usefulness metric [64, for example].

Different research interests have different understandings of and requirements for the usefulness of net-

work traces. For example, network-security research for wired networks pays most attention to the TCP/IP layers and above, and does not often address the link layer. In the wireless-network security world, however, the focus is largely on the link layer [3]. Even for the same feature in a network trace, such as timestamp, Quality-of-Service research may require micro-second resolution [53], while other research, such as Delay Tolerant Networks, may be satisfied with minute-level resolution. Because of diverse research interests, it is infeasible to generalize the notion of "usefulness" by including the semantics of all possible usages. We think that there is another avenue for future research on the usefulness metric: we need a range of possible metrics that each apply to one or more trace features, then we need a framework that allows one to compose these per-feature metrics to provide an overall metric for the trace's usefulness.

5.3 Interpreting sanitization results

Although some evaluation papers report comparison results [15, for example], such as which sanitization method is most effective for a given trace and what kind of trace is most effectively sanitized when the same sanitization method is applied, there needs to be more research on how to develop an explicit evaluation stage that informs the trace publisher about the quality of the sanitization result in terms of various sanitization metrics, and on methods to effectively communicate the results to the trace publisher.

Indeed, trace publishers may find it difficult to interpret a sanitization result in terms of sanitization metrics. Generally, publishers are most interested in how to use sanitization methods or tools and how well the methods or tools achieve the publishers' initial goals, that is, for anonymity and usefulness. Therefore, they may prefer an intuitive interpretation of the sanitization result rather than rigorous metrics expressed in complicated mathematics. For example, although entropy has been used often in anonymity research [22, 56, 14], it may be difficult for trace publishers to intuitively interpret an entropy-based metric.

Thus, it remains an open problem to present the evaluation results as high-level feedback about the quality and limits of the sanitization. The feedback may report how well each of the user-specified sanitization goals is achieved, and if any goal fails, to identify a reason and to recommend a method that may resolve the conflict resulting from the tradeoff between anonymity and usability.

6 Case study: Dartmouth Internet Security Testbed

In Section 1.1 we noted the that it is often tremendously difficult to obtain permission to collect network traces on a production network, not to mention the logistical and technical challenges of establishing a robust and effective trace-capture system. In this section, we offer as a case study our experiences in deploying the Dartmouth Internet Security Testbed (DIST) [23]. We hope this case study offers practical lessons for others who may wish to collect network traces within their own enterprise.

Several years ago, in January 2006, we sought to build a large testbed for conducting network-security research at Dartmouth College. The testbed would contain both wired- and wireless-network components, and would cover a substantial fraction of the campus production network. The wireless-network infrastructure would be used initially for trace capture, but the hardware would also be useful for other wireless-network experiments including controlled studies of Wi-Fi network attacks. The wired-network infrastructure would only be used for capture and real-time analysis of traffic on the campus backbone network. Although research was the primary purpose of and motivation for the infrastructure, the Dartmouth network-operations group was enthusiastically interested in leveraging the infrastructure and the researchers' results for operational monitoring of the network.

The wireless-network infrastructure consists of about 220 Wi-Fi access points, of the same brand and model Dartmouth uses to provide its production Wi-Fi network on campus. We had developed a scalable network-monitoring and intrusion-detection software base in our MAP project [57], in which we reflash the Aruba AP70 access points with OpenWRT Linux and run our own software for sniffing on the Wi-Fi network interface. This software uses pcap to capture Wi-Fi frames and packs multiple frames into a custom format for transmission to our central server for real-time analysis and (optionally) storage.

The wired-network infrastructure includes a one-way feed of the campus network traffic, from a span port on one of the backbone routers, into a server located in the central computing facility. The research goal was to install and evaluate various network intrusion-detection systems, including several developed by Dartmouth researchers under previous projects, to evaluate their real-time performance on huge traffic flows.

Needless to say, the installation and operation of such an infrastructure requires careful planning and communication with the relevant campus departments. Although we had been collecting Wi-Fi network traces since 2001, the new wireless-network infrastructure was going to capture an order of magnitude more data, and the new wired-network infrastructure was going to capture data that had never been captured for research purposes at Dartmouth. Furthermore, the physical installation of over 200 Wi-Fi access points in about 10 large buildings around campus meant drilling holes into walls, leading to potential concerns about aesthetics.

The first step was to obtain permission from the Network Services group within Peter Kiewit Computing Services (PKCS), the central campus IT office. This step was easy, because we had developed the concept in collaboration with Network Services. We are fortunate to have a group of talented professionals who are also enthusiastic collaborators with researchers. We have repeatedly heard from colleagues, however, that this hurdle is very difficult in their organizations.

The next step was to obtain formal permission from Dartmouth's Committee for Protection of Human Subjects (CPHS). CPHS services as the Institutional Review Board (IRB) for Dartmouth; all universities with federal research funding are required to operate an IRB so that research involving human subjects can be evaluated to ensure that risks are acceptable and subjects provide informed consent where appropriate. Our Wi-Fi network tracing effort was approved by CPHS several years prior, and our proposed effort was a subset of what we had done earlier, so a simple renewal was sufficient. Our wired-network effort was new, however, so we submitted a new project application to CPHS.

Meanwhile, we set out to obtain permission from the department heads located in the buildings where we hoped to place Wi-Fi sniffers. These buildings included the main library complex, the school of engineering, the school of business, a gymnasium, a student center, several dormitories, and several academic buildings. In some cases, we chose sites where renovation was underway and our sniffers (and their wiring) could be easily installed in the construction process, requiring less cost and no inconvenience to the building residents. In all cases, we met personally with the lead staff in each department, describing what we planned to do. We walked through their building, sometimes repeatedly, discussing in detail the placement of sniffers and their wiring. Each building required several months of planning to obtain permission, choose sites, confirm the sites with department staff, obtain quotes from electricians, install the wiring, and install the sniffers.

During this process, Sicker et al. published a paper on the legal issues involved in network trace capture [58]. The paper provides a thoughtful review of the many issues involved, and yet concludes that the legal status of network trace-collection for research purposes is not entirely clear. We consulted with the university counsel in depth, and with outside consultants, concluding that such trace collection could proceed as long as the research activity was closely coupled with network-operations activity. We had involved PKCS Network Services from the start, but we adjusted our research program to more directly meet their needs; our trace-capture facility can now support both operational and research goals simultaneously.

Furthermore, because of the scale of this effort, and the sensitive issues related to the privacy of network users, we met with several leadership groups on campus to explain our plans, answer their questions, obtain their feedback, and ultimately seek official approval from the College to proceed with trace collection. In particular, we met with the high-level faculty committee responsible for sponsored research and the provost-level council that includes all campus deans. In both cases we obtained valuable feedback that helped us to clarify our operating parameters. We developed an increasingly crisp understanding of the privacy risks and our mechanisms for mitigating those risks.

Ultimately, we decided to conduct a careful, objective study of our trace-collection infrastructure and our privacy-protection mechanisms. The College hired an outside expert, a researcher with several years of network-tracing experience in academic settings, to visit campus, interview the research teams, and to study our trace-collection infrastructure in detail. This visit served as a tremendous help to us, providing a critical eye to help us recognize where our plans could be improved or become more specific. In the end, based on the expert's advice and internal deliberations, the College leadership decided that the risks posed by the wired-network infrastructure (given the type of data needed by the researchers) were not easily mitigated, and the wired-network capture will not proceed. For the Wi-Fi infrastructure, we decided to add additional layers of security— to ensure that the infrastructure itself can not be compromised by attackers— and additional layers of encryption and in-line anonymization to protect the privacy of network users. If, in the future, we make non-trivial changes to our data-collection infrastructure we will again ask the expert to evaluate our plans.

An important part of the process is communication and public notice, especially since informed consent is not feasible in an open wireless network covering numerous buildings and a shifting population. Every one of our sniffers is labeled with the URL of our website describing the project. We are posting notices at the entries to each building, informing visitors of the data-collection effort and directing them to the website for further information. At the request of the library, we are posting notices on every table in public areas of the library. Finally, we issued a press release describing our research and the scope of the data collection.

At this writing, our Wi-Fi sniffing infrastructure is ready to begin operation. We include several layers of security on the sniffers, including extremely limited services, narrow firewall openings, no crypto keys in persistent storage, and frequent defensive port-scans. We discard all but the MAC layer from each frame, then encrypt each packet of captured frames before sending them to the server; at the server they are decrypted and immediately anonymized before being used for inline analysis or storage for offline analysis. The anonymization map is generated anew for each experiment, using a random seed that is discarded after use. As a result, very little sensitive information is captured and the most sensitive components (MAC addresses and SSIDs) are thoroughly anonymized.

The result is, we expect, a highly secure, privacy sensitive, scalable capture system for Wi-Fi networks, larger and more secure than any other ever assembled. We intend to use the infrastructure to collect operationally useful data for Network Services, and to serve our own ongoing research in trace anonymization techniques.

7 Summary and conclusion

It can be technically and logistically challenging to collect large amounts of real network data. Sharing such data with the larger research community becomes an imperative for advancing scientific progress. Similarly, network operators and engineers look for ways to reliably share network traces to help analyze network problems. Unfortunately, legal, ethical, and privacy concerns complicate these sharing efforts.

In this chapter, we survey methods for sanitizing traces, methods for de-sanitizing traces, and methods for evaluating sanitization algorithms and tools. We discuss much of the research that describe methods to (or demonstrate the failure of methods to) protect the privacy of network users and the confidentiality of certain network properties.

Although this body of work contains numerous examples of methods for sanitizing a particular feature or set of fields (that is, identifying such information and blanking it out or transforming it in some way to suppress it), these methods are often bypassable by de-sanitization techniques that consider inter-feature or inter-record relationships or external information.

We hypothesize that, because researchers and network operators who want to share trace data have access to only a few tools for quantitatively assessing the *quality* of a particular sanitization technique or resulting data set, it is difficult for much substantial progress to be made on anticipating and defeating de-sanitization attacks. In essence, the risks of certain classes of sanitization methods are not well understood

because metrics for evaluating the efficacy of classes of de-sanitization attacks are in their infancy.

Efforts to improve our ability to measure the efficacy of sanitization are of paramount concern. As we note above, metrics, be they simple measures of a particular feature or complicated mathematical models, face an underlying problem: there is a large gap between their semantics and the semantics understood by users of the sanitization tool, or of the network trace. The key problem is that the semantics of sanitization success remains unclear and unintuitive for trace producers and (legitimate) trace consumers. Any such metric must simultaneously convey (1) how well sensitive information has been suppressed (that is, the level of effort for an attacker to recover this information) and (2) the potential loss for legitimate research or operational uses. Metrics that have these semantics can then be used with confidence in decisions about which traces, portions thereof, or derivative statistics can or should be shared with various consumers.

Acknowledgements

This chapter results from a research program in the Institute for Security, Technology, and Society (ISTS), supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, by the CRAWDAD archive at Dartmouth College (funded by Award 0454062 from the National Science Foundation), and by the NetSANI project at Dartmouth College (funded by Award CNS-0831409 from the National Science Foundation). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security or the National Science Foundation.

References

[1] C. C. Aggarwal and P. S. Yu. *Privacy-Preserving Data Mining: Models and Algorithms*. Springer-Verlag, 2008.

[2] K. Bauer, D. Mccoy, B. Greenstein, D. Grunwald, and D. Sicker. Using wireless physical layer information to construct implicit identifiers. In *Proceedings of HotPETS 2008*, July 2008. Online at http://petsymposium.org/2008/hotpets/mccoyd_hotpets2008.pdf.

[3] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, pages 15–28. USENIX, Aug. 2003. Online at http://www.usenix.org/publications/library/proceedings/sec03/tech/bellardo.html.

[4] M. Bezzi. An entropy based method for measuring anonymity. In *Proceedings of the International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm)*, pages 28–32, Sept. 2007. DOI 10.1109/SECCOM.2007.4550303.

[5] M. Bishop, R. Crawford, B. Bhumiratana, L. Clark, and K. Levitt. Some problems in sanitizing network data. In *Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 307–312. IEEE Press, 2006. DOI 10.1109/WETICE.2006.62.

[6] E. Blanton. TCPurify: A "sanitary" sniffer, 2000. Online at http://masaka.cs.ohiou.edu/~eblanton/tcpurify/.

[7] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of the ACM Conference on Wireless Network Security (WiSec)*, pages 56–61. ACM Press, 2008. DOI 10.1145/1352533.1352543.

[8] T. Brekne, A. Årnes, and A. Øslebø. Anonymization of IP traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies. In *Proceedings of the International Symposium on Privacy Enhancing Technologies (PET)*, volume 3856 of *Lecture Notes in Computer Science*, pages 179–196. Springer-Verlag, 2005. DOI 10.1007/11767831_12.

[9] J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 70–78. ACM, August 2008. DOI 10.1145/1401890.1401904.

[10] M. Burkhart, D. Brauckhoff, M. May, and E. Boschi. The risk-utility tradeoff for IP address truncation. In *Proceedings of the ACM Workshop on Network Data Anonymization (NDA)*, pages 23–30. ACM, October 2008. DOI 10.1145/1456441.1456452.

[11] A. J. Burstein. Toward a culture of cybersecurity research. Technical Report 1113014, UC Berkeley Public Law Research Paper, 2008. Online at http://ssrn.com/abstract=1113014.

[12] Cooperative Association for Internet Data Analysis (CAIDA). Online at http://www.caida.org/, visited 2008.

[13] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer-Verlag, 2007.

[14] S. Clauß. A framework for quantification of linkability within a privacy-enhancing identity management system. In G. Müller, editor, *Proceedings of Emerging Trends in Information and Communication Security*, volume 3995 of *Lecture Notes in Computer Science*, pages 191–205. Springer-Verlag, 2006. DOI 10.1007/11766155_14.

[15] S. Coull, C. Wright, F. Monrose, A. Keromytis, and M. Reiter. Taming the Devil: Techniques for evaluating anonymized network data. In *Proceedings of the Annual Symposium on Network and Distributed System Security (NDSS)*. IEEE Press, Feb. 2008. Online at http://www.isoc.org/isoc/conferences/ndss/08/papers/08_taming_the_devil.pdf.

[16] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter. On web browsing privacy in anonymized netflows. In *Proceedings of the USENIX Security Symposium*, pages 1–14. USENIX, 2007. Online at http://www.usenix.org/publications/library/proceedings/sec03/tech/bellardo.html.

[17] S. E. Coull, F. Monrose, M. K. Reiter, and M. D. Bailey. The Challenges of Effectively Anonymizing Network Data. In *Proceedings of the Cybersecurity Applications & Technology Conference For Homeland Security (CATCH)*, pages 230–236, March 2009. Online at http://www.cs.unc.edu/~scoull/CATCH09.pdf.

[18] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, and M. K. Reiter. Playing Devil's advocate: Inferring sensitive information from anonymized network traces. In *Proceedings of the Annual Symposium on Network and Distributed System Security (NDSS)*. IEEE Press, Feb. 2007. Online at http://www.isoc.org/isoc/conferences/ndss/07/papers/playing_devils_advocate.pdf.

[19] Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD). Online at http://www.crawdad.org/, visited 2010.

[20] R. Crawford, M. Bishop, B. Bhumiratana, L. Clark, and K. Levitt. Sanitization models and their limitations. In *Proceedings of the Workshop on New Security Paradigms (NSPW)*, pages 41–56. ACM Press, 2007. DOI 10.1145/1278940.1278948.

[21] Internet measurement data catalog (DatCat). Online at http://www.datcat.org, visited 2010.

[22] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Proceedings of the International Symposium on Privacy Enhancing Technologies (PET)*, volume 2482 of *Lecture Notes in Computer Science*, pages 54–68. Springer-Verlag, 2002. DOI 10.1007/3-540-36467-6_5.

[23] Dartmouth Internet Security Testbed. Online at http://www.cs.dartmouth.edu/~dist/, visited 2010.

[24] DHSIELD. Online at http://www.dshield.org/, visited 2008.

[25] The Enron email data set. Online at http://www.cs.cmu.edu/~enron/, visited 2008.

[26] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. *Computer Networks*, 46(2):253–272, 2004. DOI 10.1016/j.comnet.2004.03.033.

[27] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of the USENIX Security Symposium*. USENIX, 2006. Online at http://www.usenix.org/event/sec06/tech/franklin.html.

[28] S. Gattani and T. E. Daniels. Reference models for network data anonymization. In *Proceedings of the ACM Workshop on Network Data Anonymization (NDA)*, pages 41–48. ACM Press, 2008. DOI 10.1145/1456441.1456454.

[29] S. Hansell. AOL removes search data on group of web users. Online at http://www.nytimes.com/2006/08/08/business/media/08aol.html, visited 8 Aug. 2006.

[30] M. Harvan and J. Schonwalder. Prefix- and lexicographical-order-preserving IP address anonymization. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 519–526, 2006. DOI 10.1109/NOMS.2006.1687580.

[31] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288, New York, NY, USA, 2002. ACM. DOI 10.1145/775047.775089.

[32] D. J. Kelly, R. A. Raines, M. R. Grimaila, R. O. Baldwin, and B. E. Mullins. A survey of state-of-the-art in anonymity metrics. In *Proceedings of the ACM Workshop on Network Data Anonymization (NDA)*, pages 31–40. ACM Press, 2008. DOI 10.1145/1456441.1456453.

[33] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 217–228, New York, NY, USA, 2006. ACM. DOI http://doi.acm.org/10.1145/1142473.1142499.

[34] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 211–225, 2005. DOI 10.1109/SP.2005.18.

[35] D. Koukis, S. Antonatos, and K. G. Anagnostakis. On the privacy risks of publishing anonymized IP network traces. In *Proceedings of the International Conference on Communications and Multimedia Security (CMS)*, volume 4237 of *Lecture Notes in Computer Science*, pages 22–32. Springer-Verlag, 2006. DOI 10.1007/11909033_3.

[36] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos, and P. Trimintzios. A generic anonymization framework for network traffic. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 5, Istanbul, Turkey, June 2006. IEEE Press. DOI 10.1109/ICC.2006.255113.

[37] A. Kounine and M. Bezzi. Accessing disclosure risk in anonymized datasets. In *FloCon 2008*, January 2008. Online at http://www.cert.org/flocon/2008/presentations/Bezzi_Kounine_Flocon.pdf.

[38] U. Kumar, N. Yadav, and A. Helmy. Gender-based feature analysis in campus-wide wlans. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):40–42, 2008. DOI 10.1145/1374512.1374525.

[39] K. Lefevre, D. Dewitt, and R. Ramakrishnan. Workload-aware anonymization. In *KDD*, 2006.

[40] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 106–115, 2007. DOI 10.1109/ICDE.2007.367856.

[41] Y. Li, A. Slagell, K. Luo, and W. Yurcik. CANINE: A combined conversion and anonymization tool for processing netflows for security. In *Proceedings of the International Conference on Telecommunication Systems, Modeling and Analysis*, Nov. 2005. Online at http://laim.ncsa.uiuc.edu/downloads/li05.pdf.

[42] K. Luo, Y. Li, A. Slagell, and W. Yurcik. CANINE: A netflow converter/anonymizer tool for format interoperability and secure sharing. In *Proceedings of FLOCON Network Flow Analysis Workshop*, Sept. 2005. Online at http://laim.ncsa.uiuc.edu/downloads/luo05.pdf.

[43] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 24–85, 2006. DOI 10.1109/ICDE.2006.1.

[44] G. Minshall. TCPdPriv: Program for eliminating confidential information from traces, 2005. Online at http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html.

[45] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 111–125. IEEE Press, 2008. DOI 10.1109/SP.2008.33.

[46] L. Overlier, T. Brekne, and A. Arnes. Non-expanding transaction specific pseudonymization for IP traffic monitoring. In Y. Desmedt, H. Wang, Y. Mu, and Y. Li, editors, *Proceedings of Cryptology and Network Security (CANS)*, volume 3810 of *Lecture Notes in Computer Science*, pages 261–273. Springer-Verlag, 2005. DOI 10.1007/11599371_22.

[47] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 99–110. ACM Press, 2007. DOI 10.1145/1287853.1287866.

[48] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38, 2006. DOI 10.1145/1111322.1111330.

[49] A. Pfitzmann and M. Hansen. Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology, Feb. 2008. Version 0.31, Online at http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.

[50] P. Porras and V. Shmatikov. Large-scale collection and sanitization of network security data: risks and challenges. In *Proceedings of the Workshop on New Security Paradigms (NSPW)*, pages 57–64. ACM Press, 2007. DOI 10.1145/1278940.1278949.

[51] Protected Repository for the Defense of Infrastructure against Cyber Threats (PREDICT). Online at http://www.predict.org, visited 2010.

[52] R. Ramaswamy and T. Wolf. High-speed prefix-preserving IP address anonymization for passive measurement systems. *ACM/IEEE Transactions on Networking (TON)*, 15(1):26–39, Jan. 2007. DOI 10.1109/TNET.2006.890128.

[53] M. Raya, J.-P. Hubaux, and I. Aad. DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots. *IEEE Transactions on Mobile Computing*, 5(12):1691–1705, 2006. DOI 10.1109/TMC.2006.183.

[54] M. Roesch. Snort: A free, open source network intrusion detection and prevention system, 1998. Online at http://www.snort.org/.

[55] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International, 1998. Online at http://www.csl.sri.com/papers/sritr-98-04/.

[56] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the International Symposium on Privacy Enhancing Technologies (PET)*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer-Verlag, 2002. DOI 10.1007/3-540-36467-6_4.

[57] Y. Sheng, G. Chen, H. Yin, K. Tan, U. Deshpande, B. Vance, D. Kotz, A. Campbell, C. McDonald, T. Henderson, and J. Wright. MAP: A scalable monitoring system for dependable 802.11 wireless networks. *IEEE Wireless Communications*, 15(5):10–18, Oct. 2008. DOI 10.1109/MWC.2008.4653127.

[58] D. C. Sicker, P. Ohm, and D. Grunwald. Legal issues surrounding monitoring during network research. In *Proceedings of the Internet Measurement Conference (IMC)*, pages 141–148. ACM Press, 2007. DOI 10.1145/1298306.1298307.

[59] A. Slagell, K. Lakkaraju, and K. Luo. FLAIM: A multi-level anonymization framework for computer and network logs. In *Proceedings of the USENIX Large Installation System Administration Conference (LISA)*, Dec. 2006. Online at http://www.usenix.org/events/lisa06/tech/slagell.html.

[60] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 466–473, Washington, DC, USA, 2005. IEEE Computer Society. DOI 10.1109/ICDM.2005.142.

[61] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, pages 280–289, Nov. 2002.

[62] T. Ylonen. Thoughts on how to mount an attack on tcpdpriv's "-a50" option. Online at http://ita.ee.lbl.gov/html/contrib/attack50/attack50.html, visited 2009.

[63] W. Yurcik, C. Woolam, G. Hellings, L. Khan, and B. Thuraisingham. SCRUB-tcpdump: A multi-level packet anonymizer demonstrating privacy/analysis tradeoffs. In *Proceedings of the IEEE International Workshop on the Value of Security through Collaboration (SECOVAL)*, pages 49–56. IEEE Press, Sept. 2007. DOI 10.1109/SECCOM.2007.4550306.

[64] W. Yurcik, C. Woolam, G. Hellings, L. Khan, and B. M. Thuraisingham. Toward trusted sharing of network packet traces using anonymization: Single-field privacy/analysis tradeoffs. Technical Report 0710.3979v2, arXiv, 2007. Online at http://arxiv.org/abs/0710.3979.

[65] J. Zhang, N. Borisov, and W. Yurcik. Outsourcing security analysis with anonymized logs. In *Proceedings of the International Workshop on the Value of Security through Collaboration (SECOVAL)*. IEEE Press, 2006. DOI 10.1109/SECCOMW.2006.359577.

[66] Q. Zhang and X. Li. An IP address anonymization scheme with multiple access levels. In I. Chong and K. Kawahara, editors, *Proceedings of Information Networking: Advances in Data Communications and Wireless Networks (ICOIN)*, volume 3961 of *Lecture Notes in Computer Science*, pages 793–802. Springer-Verlag, 2006. DOI 10.1007/11919568_79.