


2008

Reference models for network trace anonymization

Shantanu Gattani
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Computer Sciences Commons](#), and the [Library and Information Science Commons](#)

Recommended Citation

Gattani, Shantanu, "Reference models for network trace anonymization" (2008). *Retrospective Theses and Dissertations*. 15304.
<https://lib.dr.iastate.edu/rtd/15304>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Reference models for network trace anonymization

by

Shantanu Gattani

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Co-majors: Computer Engineering,
Information Assurance

Program of Study Committee:
Thomas E. Daniels, Major Professor
Douglas W. Jacobson
Brett M. Bode

Iowa State University

Ames, Iowa

2008

Copyright © Shantanu Gattani, 2008. All rights reserved.

UMI Number: 1453120

UMI[®]

UMI Microform 1453120

Copyright 2008 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

DEDICATION

I dedicate this thesis to my late grandfather, Sri Badri Prasad Gattani. He taught me the true value of education. I miss him every day.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
ACKNOWLEDGEMENTS	viii
ABSTRACT	ix
CHAPTER 1. OVERVIEW	1
1.1 Introduction	2
1.2 Motivation for Sharing Network Traces	3
1.3 Problems in Sharing Network Data	5
1.3.1 Privacy Vs. Usability	5
1.4 Current Efforts of Sharing Network Data	7
1.4.1 CRAWDAD	8
1.4.2 PREDICT	8
1.4.3 DatCat	9
1.5 The NetBottle Approach	9
1.5.1 Probing, Scanning and Characterization	10
1.5.2 Model Construction	11
1.5.3 Deployment	11
1.6 Thesis Contribution and Organization	12
CHAPTER 2. RELATED WORK	13
2.1 Anonymization Tools	13

2.1.1	Anonymization Tool Requirements	14
2.1.2	Multi-Level Anonymization Tools	15
2.1.3	Multi-Layer Anonymization Tools	16
2.1.4	Issues and Insights	17
2.2	Attacks against Anonymization Techniques	18
2.2.1	Active Data Injection Attacks	19
2.2.2	Known Mapping Attacks	19
2.2.3	Network Topology Inference	20
2.2.4	Cryptographic attacks	21
2.3	Frameworks	22
2.3.1	Evaluation Framework	22
2.3.2	Attack Framework	24
CHAPTER 3. REFERENCE MODEL FOR TRACE ANONYMIZA-		
TION		26
3.1	Entities	27
3.1.1	Data Remitter	27
3.1.2	Data Interpreter	28
3.2	Reference Model	29
3.2.1	Data Model	30
3.2.2	Auditor Model	31
3.2.3	Collector Model	33
3.3	Privacy Metric for Sanitized Network Logs	38
CHAPTER 4. INFORMATION FLOW MODEL		40
4.1	Information Flow	41
4.1.1	Case 1: $U_O == R_O$	42
4.1.2	Case 2: $R_O \subset U_O$	42

4.1.3 Case 3: $R_O \not\subseteq U_O$	43
4.2 Probabilistic Interpretation	44
CHAPTER 5. FUTURE WORKS	47
5.1 Roadmap	47
CHAPTER 6. CONCLUSION	49
BIBLIOGRAPHY	51

LIST OF TABLES

Table 1.1	Examples of Methods to Anonymize IP Addresses	6
-----------	---	---

LIST OF FIGURES

Figure 1.1	Privacy Vs. Usability Tradeoff	7
Figure 1.2	NetBottle 3-Layer Paradigm	10
Figure 2.1	Evaluation Framework suggested by Coull. et. al.	24
Figure 3.1	Reference Model for Network Trace Anonymization	32
Figure 3.2	Functional Architecture for Trace Collection and Sanitization	36
Figure 3.3	Procedure to Derive Privacy Metric	38
Figure 4.1	Information Flow Model	41
Figure 4.2	Information sets	42
Figure 5.1	Roadmap for Future Work	48
Figure 6.1	Updated Threat Model	49

ACKNOWLEDGEMENTS

More than six years have passed since I first stepped on the campus of Iowa State University, young and clueless, a freshman in computer engineering. Now, as I prepare to step over the threshold of the ivory tower into the real world, I must pause and give credit where it is due.

Research does not happen in isolation. Rather, numerous people have contributed to this thesis through their ideas, feedback and support. First and foremost, I would like to express my sincere gratitude to my major professor, Dr. Thomas Daniels, or as his students call him, Dr. D. He took me under his guidance in my senior year at Iowa State and has been a constant source of inspiration ever since. His mentorship over the past years has been invaluable, just as his patience and help with this thesis. I am grateful for his direction, especially in the wake of my original work getting scooped! Thanks Dr. D, you provided me a wealth of all things technical and philosophical.

Next I would like to thank my colleagues from lab. Their constructive criticism and insightful feedback helped me shape this thesis. A special note of thanks is due to Benjamin Anderson. His wonderful *words of wisdom* kept me sane through graduate school.

To Dr. Jacobson and Dr. Bode, I extend thanks for participating on my program of study committee and providing much needed feedback.

Finally, I am grateful for all of the support given to me by my family and friends. The reality of school and life would be insurmountable if it were not for their constant support and company.

ABSTRACT

Network security research can benefit greatly from testing environments that are capable of generating realistic, repeatable and configurable background traffic. In order to conduct network security experiments on systems such as Intrusion Detection Systems and Intrusion Prevention Systems, researchers require isolated testbeds capable of recreating actual network environments, complete with infrastructure and traffic details. Unfortunately, due to privacy and flexibility concerns, actual network traffic is rarely shared by organizations as sensitive information, such as IP addresses, device identity and behavioral information can be inferred from the traffic. Trace data anonymization is one solution to this problem. The research community has responded to this *sanitization problem* with anonymization tools that aim to remove sensitive information from network traces, and attacks on anonymized traces that aim to evaluate the efficacy of the anonymization schemes. However there is continued lack of a comprehensive model that distills all elements of the sanitization problem in to a functional reference model.

In this thesis we offer such a comprehensive functional reference model that identifies and binds together all the entities required to formulate the problem of network data anonymization. We build a new information flow model that illustrates the overly optimistic nature of inference attacks on anonymized traces. We also provide a probabilistic interpretation of the information model and develop a privacy metric for anonymized traces. Finally, we develop the architecture for a highly configurable, multi-layer network trace collection and sanitization tool. In addition to addressing privacy and flexibility concerns, our architecture allows for uniformity of anonymization and ease of data ag-

gregation.

CHAPTER 1. OVERVIEW

Network security research can benefit greatly from testing environments that are capable of generating realistic, repeatable and configurable background traffic. In order to conduct network security experiments, researchers require isolated testbeds that can recreate actual network environments, complete with infrastructure and traffic details [30]. Ideally, such background traffic would be realistic, repeatable and tunable to allow for accurate and repeatable security testing. For our purposes, realistic can be defined as being indistinguishable from actual traffic collected from the environment being simulated. Unfortunately, due to privacy and flexibility concerns, actual network traffic is rarely shared by organizations as sensitive information, such as IP addresses, device identity or other information can be determined from actual traffic [1, 18]. In order to solve this problem, several network data anonymization tools and techniques have been developed which intend to sanitize sensitive data such that identity information cannot be inferred from a published trace. Reassured by these developments, recent years have seen the emergence of a few network data cataloging repositories [6, 5, 20]. However due to the lack of explicit confidence metrics in the efficacy of these tools, the availability of accurate and usable enterprise network trace data remains scarce.

In this thesis we introduce a comprehensive functional reference model that identifies and binds together all the entities and components required to formulate the problem of network data anonymization. We aim to tie in all relevant past work into our model and build a new information flow model that illustrates the overly optimistic nature of attacks on anonymized traces. We also provide an information theoretic interpretation

of the information model and develop a privacy metric for anonymized traces.

1.1 Introduction

With the Internet revolution of the nineties, our communication mechanisms, commerce, economy, national security, and even mundane chores of our daily lives are becoming more and more intertwined with technology and its reliable functioning. As this interdependency increases, it becomes vital to ensure the reliability and security of this vast infrastructure. However, with its growing complexity it is becoming increasingly difficult to develop and test new security solutions. Such a gargantuan task requires new methods and techniques along with the development of versatile testbeds that allow modeling of large enterprise networks.

In order to secure a large organization, not only is it important to develop improved security solutions, it is equally necessary to development new testbed environments that accurately simulate the target network. The development of such virtual worlds represents a paradigm shift in the area of security research making them invaluable to industry and academia alike.

This network-modeling problem, however, is non-trivial, not only because of the scale and complexity of modern networks, but also due to the lack of network trace data and the reluctance of organizations to share this information with security researchers. In the past we have seen various efforts that address portions of this problem, such as traffic characterization, traffic and environment reproduction and various testbed designs [7, 30, 23]. However, since these efforts only address individual parts of the problem, the interdependencies and details tend to be lost in translation from data collection and characterization, to deployment of the test environment, thereby reducing the accuracy of the deployed environment.

The “Network in a Bottle”, or NetBottle, Project is an ongoing effort at Iowa State

University that employs a new paradigm to preserve these important details to accurately recreate the desired environment. This new paradigm divides the problem into three layers - data collection and anonymization, model construction and emulation and deployment. By addressing the entire process, the interdependencies and desired details can be preserved, while sensitive information within the actual network traffic can be anonymized and removed, as required, during model construction, preventing that information from appearing in the testing environment. It is hoped that this approach will allow for more accurate recreation of a real-world environment within a testbed, allowing for more accurate experimentation to occur. However the development of the NetBottle project is constricted by the lack of accurate and usable network traces, which are essential for the development of the statistical models that feed any realistic background traffic generator. Industry and academic organizations are like-minded in their reluctance to share network trace data, which has resulted in fewer entities sharing them.

1.2 Motivation for Sharing Network Traces

Past and current research has laid significant emphasis on the development of tools and techniques to prevent and detect cyber exploits. Three broad categories of such tools are Firewalls, Intrusion Detection Systems and Intrusion Prevention Systems. Intrusion detection and intrusion prevention system monitor network traffic for suspicious activity and takes appropriate actions such as raising alarms and fingerprinting the attack/perpetrator. Since these systems identify intrusions by differentiating the anomalous activity from normal network activity, the description of ‘normal traffic’ assumes vital importance in the quality of intrusion detection systems.

While studying the behavior of attack programs such as worms and viruses, which are let go without further human intervention, is feasible without realistic background traffic, testing security tools against human motives and complex multi-phase attacks is

infeasible in the absence of an accurately modeled network with realistic traffic within a sandboxed environment. Simulating realistic network traffic takes into account various attributes that affect observed traffic in real networks. For example, modeling host behavior consists of usage habits as well as ordering of the applications used. Similarly, application protocols, their types, their versions and protocol versions play an important role in behavior of generated traffic. Therefore it is necessary for security operators to share logs of network activity data with researchers to enable them to accurately map networks within testbed environments and conduct security experiments and analysis.

While it is typical in enterprise environments to use network logs to locally optimize network throughput and security, current privacy and security policies effectively prevent organizations from sharing network logs with third party research institutions. While *security through obscurity* is frowned upon by the academic community, it is a common, perhaps unintended, practice in industry. For example, following a security breach, an organization might scan its networks for vulnerabilities and patch open holes, but the results from such reactionary behavior would typically not be shared with others, perhaps, even with departments within the same organization (to safeguard against malicious insiders). The use of such practices has led to a culture of pushing attackers away from oneself without any consideration of poor overall security resulting from a lack of co-ordination among organizations [1].

Adversaries and malicious attackers on the other hand collude and share attack information quite commonly. Gone are the days when *hacking* was for bragging rights only. A whole underground economy exists, that surrounds and fuels cyber crime and computer criminals. In order to dampen the growth of this parallel economy and safeguard oneself, it has become ever more important for industry to share network logs with research institutions. Such cooperation would facilitate building of precise and realistic testing environments for verification and validation of security tools.

1.3 Problems in Sharing Network Data

Network activity logs in the hands of a malicious attacker can be as invaluable as a map to a treasure trove. Network logs can be used for reconnaissance and network fingerprinting. Raw network traces contain information that can be utilized to determine vulnerabilities and other attack vectors and also pin point defense mechanisms. At the very least, raw trace logs can divulge private user information such as usernames, passwords, IP addresses, protocol usage, etc. Adversaries can utilize all this information to launch a range of diverse attacks, on enterprise networks as whole or individual users within the network. It is privacy and security concerns like these that prevent organizations from sharing data and coming together in a mutual effort towards network security research.

In order to circumvent these privacy concerns several tools have been developed which sanitize network traces and attempt to remove sensitive data from them. However, the anonymization tools and techniques present today provide few guarantees on privacy and do not provide the publisher any means to quantify the privacy risk of making a sanitized dataset public. Another problem inherent in the nature of network data is that it is difficult to know a priori the fields that should be considered sensitive and appropriately sanitized. Finding such information leads us to the tradeoff between privacy concerns and the usability value of a network activity trace [1, 14].

1.3.1 Privacy Vs. Usability

Network trace anonymization can be performed at several levels of granularity. Taking IP addresses for example, the crudest form of anonymization, referred to as *black marking*, removes the entire IP address from the IP header. *Gray marking* on the other hand removes host identification information from the IP address, but leaves subnet information unperturbed. It trivially follows that such crude forms of anonymization do

not provide sufficiently useful data for network security and performance research.

Two forms of sanitization that do prove useful are *prefix-preserving* or *pseudo-anonymous transformation* and *random* or *fully-anonymous transformation*.

Table 1.1 Examples of Methods to Anonymize IP Addresses

IP Address	Black Marking	Gray Marking	Randomization	Prefix-Preserving
129.186.200.110	-	129.186.0.0	69.72.15.50	151.165.120.15
129.186.200.195	-	129.186.0.0	129.186.200.195	151.165.120.125
129.186.50.50	-	129.186.0.0	129.170.7.200	151.165.80.20
129.170.7.200	-	129.186.0.0	192.168.0.100	141.185.35.72
192.168.0.100	-	129.186.0.0	129.186.50.50	145.182.10.10
69.72.15.50	-	129.186.0.0	69.62.50.50	12.29.150.10
69.72.50.50	-	129.186.0.0	129.186.200.110	12.29.150.191

While pseudo-anonymous transformations map all instances of a particular raw data field to the same unique anonymized identifier in the target namespace, random transformations maintain no such relations and simply map each instance of a raw identifier to a different identifier in the target space. Though pseudo-anonymization provides analysts data with preserved cross identifier correlations and behaviors, the concern remains that expert adversaries can extract sensitive information, such as protocol usage, network topology and even true host identities from distributions over several packet flows that are preserved by pseudo-anonymization techniques.

Therefore, we can visualize the capability of anonymization techniques as a normal distribution across a spectrum of privacy/usability. This tradeoff between privacy and utility is illustrated in Figure 1.1. As indicated in the figure, it is equally simple to optimize trace anonymization for privacy (black marking of sensitive fields) and utility (minimal sanitization). However the degree of difficulty increases rapidly when balance is desired between privacy and utility.

Remitters of network trace data prefer to stay within region 1, while data interpreters/analysts desire data with higher level of utility. Hence we propose a framework which takes into consideration a privacy metric, a usability metric and an accuracy met-

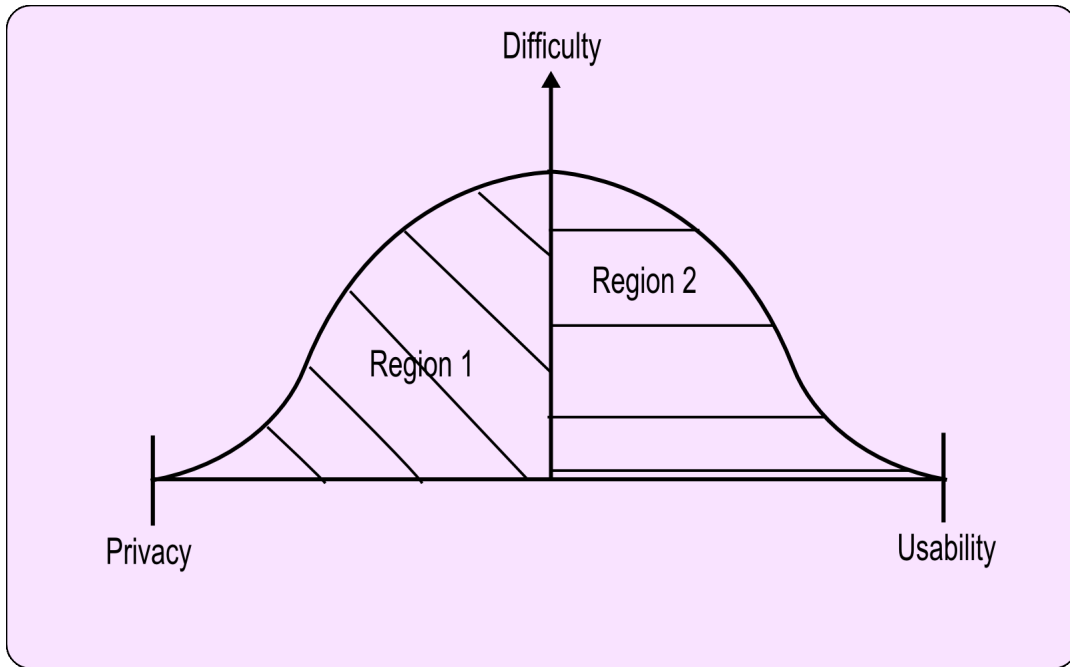


Figure 1.1 Privacy Vs. Usability Tradeoff

ric for network trace anonymization. Together these metrics would form a confidence metric for the publisher and the interpreter and facilitate a greater degree of cooperation between industry and academia.

1.4 Current Efforts of Sharing Network Data

Over the past few years the continuing dearth of real network traffic data has fueled research initiatives towards developing new and improved data anonymization techniques. This growth in the area of sanitization research has spawned a few network data repositories and catalogues. In this section we list and briefly describe a few such major efforts.

1.4.1 CRAWDAD

CRAWDAD [6] is a National Science Foundation (NSF) funded effort at Dartmouth University. This project collects network activity data from their campus-wide wireless infrastructure and utilizes their in-house tools for anonymizing and analyzing the collected data.

While this is a move in the right direction, this project is still not mature and needs to develop into a true community resource wherein it would archive trace data from several locations. Also, due to the nature of wireless data communication networks, data models generated from their datasets would not be applicable to emulation of more generic enterprise networks.

1.4.2 PREDICT

The Protected Repository for the Defense of Infrastructure Against Cyber Threats, also known as the PREDICT [20] project, is an initiative by the Department of Homeland Security Science and Technology Directorate. This project aims to provide a central repository of network traffic data accessible through a web-based portal. The repository consists of three entities - data providers, who submit data to PREDICT, data hosts that maintain independent systems to store data submitted from providers and researchers, who utilize the aforementioned data.

The PREDICT project is still in its nascent stages and promises to grow into an excellent research resource. However at this point access to this repository is extremely restricted (difficult to register and available only to researchers based in United States) and data sets are limited in number.

1.4.3 DatCat

The Internet Measurement Data Catalog [5], also known as the DatCat project, is a searchable registry of information about network measurement datasets. This repository has been developed by the Cooperative Association for Internet Data Analysis (CAIDA). While it is similar to CRAWDDAD in nature, it indexes datasets from various sources and allows users to find, annotate and cite data contributed by others, alongside providing means to upload new data. The datasets indexed under this project represent a mix of wireless and wire-line Ethernet data.

Most datasets available through DatCat focus on macro-level data and as such lack the level of detail necessary for host level modeling and simulation. These datasets also suffer from the problem of undefined anonymization standards and available traces were found to be anonymized to different levels.

1.5 The NetBottle Approach

Various efforts have been made to address portions of the sanitization and emulation problem, such as traffic characterization, traffic and environment reproduction and various testbed designs. However, since these efforts only address portions of the problem, some of the interdependencies and details are lost between data collection and deployment of the test environment, reducing the accuracy of the deployed environment.

In this section we introduce the NetBottle project, a comprehensive effort, much larger in scope and vision than related projects, which considers the emulation problem as a whole and models the big picture, preserving the important details in order to accurately recreate the target environment. As mentioned in Section 1.1, this new paradigm divides the problem into three layers - data collection, model construction and emulation and deployment. The lower layers provide data to the layers above, and various methods to evaluate the artifacts generated at a specific layer against the data

provided by the lower layers. This feedback is critical to ascertain the accuracy of the models or the deployed environment, while preventing sensitive data from being used in the creation of the models, or recreated in the generated traffic.

We examine the three layers in the NetBottle paradigm from lowest to highest.

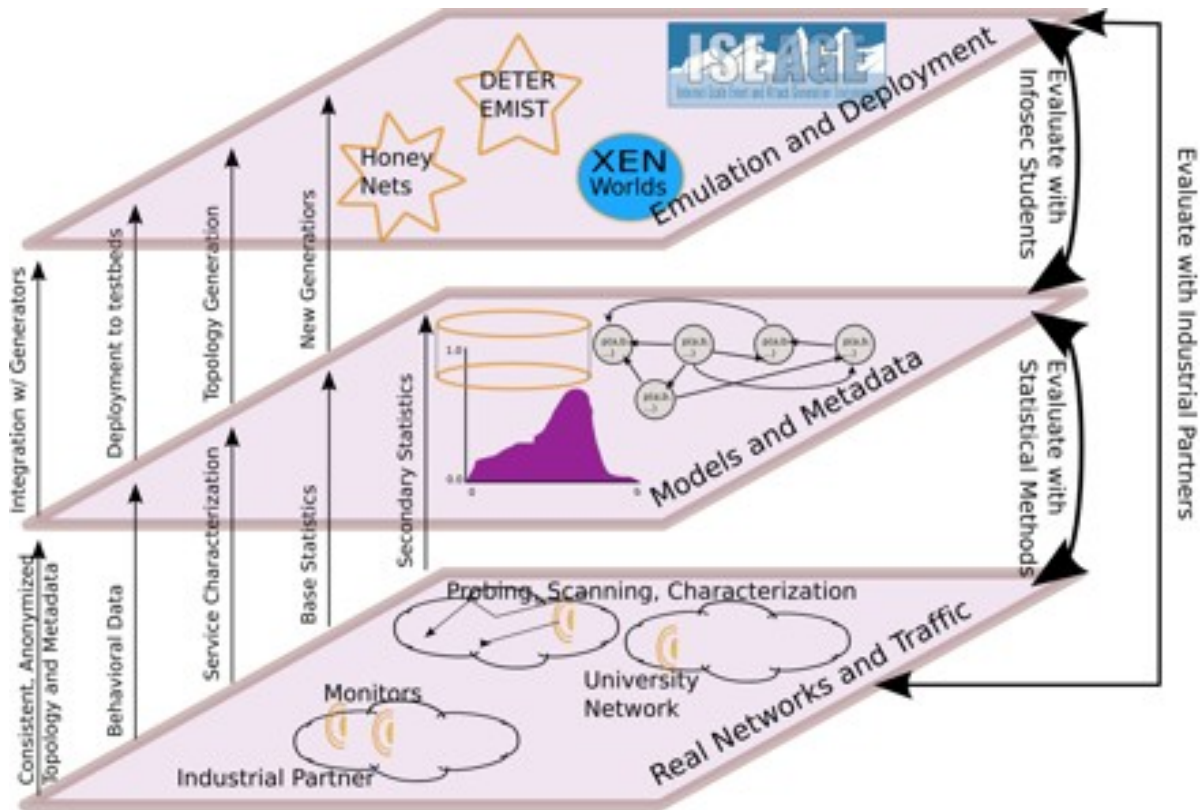


Figure 1.2 NetBottle 3-Layer Paradigm

1.5.1 Probing, Scanning and Characterization

At this layer, data collection is performed on the real-world network that is going to be emulated. The key network edges are identified, and traffic transmitted over these edges captured. Since this traffic may contain passwords and other sensitive information, the traffic captures are anonymized and sanitized to prevent this information from being leaked. However, different from related works, at this layer a privacy and utility audit is

conducted which results in privacy, utility and accuracy metrics. These metrics in turn drive changes in privacy and usability policies of the data remitter.

There is also the need to deal with multiple capture points within the network, creating the need for aggregation into a single dataset. This aggregation is required to prevent the skewing of the analysis due to single packets being captured multiple times throughout the network. This data is then analyzed to determine the network topology, key services offered by the hosts in the network, and other desired information. The results of this analysis are passed up to the second layer for use in model construction.

1.5.2 Model Construction

At this layer, the analysis data from layer one is used to construct various models of the network topology, services running and network traffic. It is important to note that the models must remain “environmentally neutral” as they are to support a variety of deployment environments. Also, since any sensitive information is removed at the previous layer, no sensitive information should exist within the models, allowing for use by several researchers and, potentially, for release to the public.

1.5.3 Deployment

At this layer, the models constructed in the middle-layer are used to construct the systems and traffic generators to emulate the desired network. The systems are configured to provide the services identified during traffic capture so appropriate responses may be generated.

Since there is the risk of some key dependencies or details being lost during the transitions between layers, or from the choice of deployment environment, it is also possible to capture data in the deployment environment and compare that against the original data capture. The differences between the two datasets can be examined to determine the overall accuracy of the testbed environment.

1.6 Thesis Contribution and Organization

This thesis provides an extensive study of anonymization techniques for published network trace data, focusing particular attention on qualitative issues of collection and anonymization. In this thesis we introduce a comprehensive functional reference model that identifies and binds together all the entities and components required to formulate the problem of network data anonymization.

Chapter 2 of this thesis presents all the relevant past work that has been done in this area. In Chapter 3 we define the functional models for data collection and introduce the concept of anonymization audits. We also develop privacy metric for anonymized logs in Chapter 3. Chapter 4 builds a new information flow model that illustrates the overly optimistic nature of previous attacks on anonymized traces and provides a probabilistic interpretation of the information model. Chapter 5 discusses future work and possible extension of this research and we draw conclusions in Chapter 6.

CHAPTER 2. RELATED WORK

Typically the anonymization process seeks to achieve three [10] goals: (i) protect privacy of monitored users, (ii) hide the internal infrastructure information of the network, and (iii) create realistic and useful anonymized traffic traces. Traditionally, the last of these three goals has been the most difficult to attain as leaving too much information in the traces allows sensitive data to be leaked, but if too much information is removed, the trace provides no practical benefit.

In this chapter we discuss relevant past work that aims to achieve the goals mentioned above. We divide this description into a discussion of tools and techniques that have been developed to achieve desired anonymization, individual attacks and inference methods that have been used against anonymization schemes, and more recent mathematical frameworks which attempt to formalize the de-anonymization process.

2.1 Anonymization Tools

Several sanitization tools have been developed over the last few years. They employ different techniques for network trace sanitization. Diverse as these tools are, they adhere to certain common requirements forced upon them by the research uses that the sanitized traces are subjected to. These requirements, as outlined by Coull et. al. in [27], are listed below:

2.1.1 Anonymization Tool Requirements

2.1.1.1 Pseudonym Consistency Requirement

Areas of research that require counting of distinct hosts within a trace, e.g. traffic characterization, require that empirical evaluation be done on a per host basis. Such research endeavors would be rendered futile if the unique host identifiers within a trace, such as IP addresses and hardware addresses, were not perturbed consistently within a trace and even across multiple traces within the same enterprise. Without such consistency, such research would only be applicable to a particular trace and not the entire organization.

2.1.1.2 Protocol Header Requirement

Most research endeavors require the use of transport, network and link layer headers. As such, through the same reasoning as above, the preservation and appropriate sanitization of these headers is central to any anonymization tool. Payload destruction, on the other hand may be allowable for certain types of research purposes.

Additionally, transport layer traffic records must be maintained within the dataset.

2.1.1.3 Port Number Requirement

Port numbers must be maintained in their unperturbed state in order to facilitate protocol classification schemes. While this requirement might seem like a red flag, it must be noted that recent research has indicated that that application layer protocols, and hence port numbers, can be accurately identified through the use of timing and size information [37].

Tcpmkpub [24] and Tcpdpriv [15] are the two most prominent trace anonymization tools. While tcpdpriv acts as a simple filter, removing or modifying packet header based

on network level rules, Paxson and Pang in [22] have developed a high level environment that allows cross layer packet anonymization. Recently Slagell et al. concluded work on FLAIM (Framework for Log Anonymization and Information Management) and CANINE (Converter and Anonymizer for Investigating NetFlow Events), both of which are multi-level anonymization tools that support several anonymizing algorithms [2].

Upon analysis of these works, we discovered that while they fulfill some expectations that are desired of an ideal anonymization tool and provide important insights, they are lacking in several key areas. Before we identify these issues, it is important that we establish a classification for anonymization tools. While researching these works two classes of anonymization tools were identified: (i) multi-level anonymization tools and (ii) multi-layer anonymization tools. Tools that fall in the former category strive to allow the user to make fine-grained tradeoffs between information loss and privacy/security concerns. They achieve this goal by supporting a diverse set of anonymization algorithms of different strengths, which work to conceal sensitive information at different levels. However, these tools primarily work at the network layer and deal with packet headers only. The multi-layer anonymization tools, in addition to supporting a variety of algorithms, reconstruct packets into data stream flows and anonymize packet payloads as well. These trace anonymization tools are cross layer and are capable of identifying and anonymizing application layer packets as well.

2.1.2 Multi-Level Anonymization Tools

Tcpdpriv, FLAIM and CANINE are all multi-level trace anonymization tools. While all three of them honor the pseudonym consistency, transport layer records and port number requirements, only Tcpdpriv meets the header requirement (FLAIM and CANINE operate on NetFlow logs). Tcpdpriv provides multiple levels of anonymization, from leaving fields untouched up to performing complete black marking of header fields. Tcpdpriv, however completely discards TCP and UDP payloads. It also provides very

limited anonymization primitives such as prefix-preservation and sequential mapping [16]. Ipsumdump [12] dumps packets into ASCII format and then uses tcpdpriv to anonymize the IP addresses as specified by the user. FLAIM and CANINE build on the same concept though they provide a much wider set of anonymization primitives and support several log formats and input sources. FLAIM can also be easily integrated with passive traffic monitoring, though it is not optimized for real time anonymized trace output. While FLAIM and CANINE have come a long way since Tcpdpriv, they suffer from some of the same issues that plague their older counterpart. All of these tools are extremely memory intensive and are marked by their lack of parallelizing options for trace anonymization.

2.1.3 Multi-Layer Anonymization Tools

The Tcpmpub programming environment proposed by Paxson and Pang is a multi-layer anonymization tool. As mentioned above, Tcpmpub constructs the packets into data stream flows upon which user defined anonymization policies are executed. Tcpmpub is not limited to the network layer and is capable of anonymizing application layer payloads as well. However, Tcpmpub also provides a limited set of anonymization primitives. Also, the framework defined in [22] is currently implemented in Bro [24], a network intrusion detection system scripting language for Unix. Bro is a rather obscure scripting language, which limits the usability of Tcpmpub, since it requires users to define their own functions due to the limited number of primitives that Bro, being an intrusion detection system, provides.

AAPI [10] (Anonymization Application Programming Interface) is another multi-layered anonymization framework. AAPI offers a wide range of anonymization capabilities that can be applied to any field of a packet up to the application level. AAPI simplifies the process of defining anonymization policies by providing an intuitive, C-like API for the user. Also, AAPI supports several input formats and flow logs and

has been partially integrated with passive network monitoring systems. While AAPI is a comprehensive tool, which possesses many of the desired characteristics of an ideal anonymization tool, its still requires seamless integration with passive network monitoring systems, optimizations for real time operation and parallel processing of trace data. Both Tcpmkpub and AAPI meet the requirements outlined above.

2.1.4 Issues and Insights

While the multi-layered tools are developments in the right direction, they still lack some architectural attributes which make them less than ideal. As described by Pang, Allman, Paxson and Lee [24], complete anonymization requires at least 2 passes through the data as some IP addresses need to be mapped across the entire IP address space without collisions. The solutions implemented by the tools describe above, anonymize as much information as possible before writing to persistent storage, and then complete the anonymization with a second pass.

However, such anonymization in isolation cannot occur, as it would prevent a proper analysis of the collected data. For example, we can imagine packets that are having their timestamps altered to prevent the fingerprinting attack proposed by Kohno, Broido, and Claffy [34]. The approach presented by Pang, Allman, Paxson and Lee would be to simply apply a counter as the timestamp, maintaining the order of the packets, but not the actual timing information. Unfortunately, without accurate timing information, it would be impossible to determine the proper ordering for packets collected at two different locations in the network. This illustrates the need to maintain some information for the packets to be properly aggregated. However, to prevent sensitive information from being propagated into the model, it is necessary to have a multi-tiered classification and categorization system. Another drawback of existing tools is the lack of an explicit privacy and usability policy. These policies should be formally defined and utilized to construct the anonymization policy. While Tcpmkpub and AAPI, as described above,

do require an explicit anonymization policy, neither tool has support for generating or updating this policy based on privacy and utility policies or feedback from an auditor.

In Chapter 3 we introduce a functional model for network data anonymization. This model requires multiple anonymization passes, interleaved with analysis steps. We can then construct an anonymization policy, based on the analysis needs and the desired level of privacy and utility, to determine when in the analysis process certain pieces of information are removed or anonymized.

2.2 Attacks against Anonymization Techniques

Anonymization techniques aim to transform network logs such that host identity, network topology, user behavior, and security mechanisms cannot be inferred from them. While it has proven quite challenging to develop a tool that provides strong anonymization without hampering usability [1, 21], several techniques have been devised to subvert the anonymization transforms and infer sensitive information from sanitized logs.

The same anonymization requirements that facilitate research utility also open several holes in the process for an attacker to exploit. The goal of the malicious attacker or adversary is to extract as much of the original data as possible from the sanitized data. The recent advancements in sanitization technologies have happened as a reaction to the growing privacy threats and increasing need for trace utility [17, 36]. Several recent works have shown that it is possible to infer network topology [27], deanonymize unique host identities [9, 26, 27], and profile user protocol usage [3, 28, 27, 25].

Despite the development of state-of-the-art sanitization techniques and tools to implement them, several attack vectors remain that an adversary may utilize to achieve circumvent anonymization. In the following section, we identify a few such attack vectors that form the building blocks of more complex hybrid attacks. We borrow the categorization developed by Pang et. al. in [24] for this purpose.

2.2.1 Active Data Injection Attacks

Pang et al. define data injection attacks as an adversary injecting information within a network to be logged with the purpose of later recognizing that data in an anonymized form. This type of attack is similar in form to a known or chosen plaintext attack in cryptography. The attacker creates specific markers in unsanitized traffic, and if these markers remain recognizable after anonymization, then they allow the adversary to derive original data, not only for the corresponding marker but also for other raw data sanitized using the same transforms. The target space of this attack is, however, limited to the organizations that publish anonymized data on a regular basis over short intervals.

In [32], Brekne et. al. empirically demonstrate the effectiveness of this attack against prefix-preserving anonymization and suggest remedies that might limit the damaging capacity of this attack. However, they concede that this type of attack is nearly impossible to defend against without tenacious human investigation and its efficacy is limited only by the expertise of the adversary. Pang et. al., as a remedy, suggest the removal of all scanning data from a trace before the sanitization and release of the dataset.

2.2.2 Known Mapping Attacks

This category of attacks exploits the discovery of a mapping between raw and sanitized data in one instance, to subvert the anonymization of that same data in multiple instances. Therefore, if IP address anonymization is consistent multiple logs, if the adversary is able to discover the mapping in one log, he or she can use the information gained to recover IP addresses in other logs.

It is easy to observe that this kind of an attack can be used effectively with data injection attacks discussed in the previous section to obtain a great deal of identity information from a sanitized network trace.

2.2.3 Network Topology Inference

Network topology consists of nodes that make up the vertices of the network, the connectivity between them as the edges of the networks and routers along the way. Several fingerprinting attacks have been suggested that attempt to deanonymize hosts and accurately infer the topology map from a trace. Pang et al. formally define fingerprinting attacks as the process of matching attributes of an anonymized object against attributes of a known object to discover a mapping between anonymized and unanonymized objects.

In order to provide intuition for such an attack, consider the case of targeting an organization's web server with such an attack. If an adversary can isolate all anonymized IP addresses which respond to large amounts of traffic on port 80, and compare them to unanonymized external data available from several public databases, the adversary will be able to create a mapping between the anonymized and raw IP addresses.

Koukis et al. [9] and Coull et al. [27] have done significant work in this area and have achieved striking deanonymization percentages. Koukis et al. in their work show that, given the pseudonym consistency and header requirements, statistical identification techniques can be used to uncover the identities of users and their surfing activities from sanitized traces. They utilize HTTP request and response payload sizes (from multiple sources) to construct a signature of a webpage and then match these signatures to similar profiles extracted from the sanitized trace.

Coull et al. on the other hand apply an information theoretic approach to identifying 'heavy hitters' in the sanitized trace. They employ an iterative algorithm to obtain the set of most frequently occurring IP addresses in the trace. They monitor the normalized entropy of the trace to determine the heavy hitters. Intuitively, upon removing frequently occurring IPs from a trace, the normalized entropy of the trace should stabilize above a threshold. They then utilize the Dominant State analysis proposed by Xu et al.

[36] to develop a behavioral profile of the heavy hitters. These behavioral profiles can then be utilized to fingerprint and uniquely identify hosts based on their behavior. In order to determine classless subnets they utilize the k-means clustering algorithm to automatically determine the best subnets.

Towsley et. al. utilize the IPID field from the IP header to infer the network path and determine end system characteristics. They exploit the IPID to determine— the amount of internal (local) traffic generated by a server, the number of servers in a large-scale, load-balanced server complex, and the difference between one-way delays of two machines to a target computer. Their work however is applicable only to 16-bit IPIDs that are generated by a global counter in the IP stack of the source machine.

While a great deal of effort has been spent on topology inference attacks, no attack completely infers the topology of a network. Above attacks can be used to identify the important servers within the network, cluster subnet information and profile hosts. However, it was unclear within [27], and other similar works, as to how this information would be used to determine connectivity between identified hosts as well as subnets. Till such a connectivity map can be accurately determined, topology inference attacks remain incomplete.

2.2.4 Cryptographic attacks

Cryptographic functions are at the heart of most anonymization schemes. This intrinsically makes the anonymization techniques only as secure as the cryptographic functions that are used to implement them. Cryptographic attacks, such as chosen plaintext attacks, known plaintext attacks, etc. are therefore equally applicable to sanitization functions. Furthermore, a cryptographic compromise affects not only a single entry or log, but also the entire data set, since these attacks usually reveal the secret keys that are used to sanitize data. Fan et al. in [16] present a complete empirical security evaluation of prefix-preserving IP address anonymization.

However, it must be noted that these attacks require expert knowledge and are much more difficult to perpetrate than other attacks that have been listed above. Also, cryptographic attacks typically require additional information that must be first gathered through other attacks.

2.3 Frameworks

The attacks described above, while effective in most cases, lack a formal structure and methodology. They are mostly implementations that exploit independent vulnerabilities, and as such, can be independently and sometimes easily mitigated. Also, they view the sanitization problem solely from the eyes of an adversary and as such provide little help to a data contributing enterprise. While they demonstrate the immaturity of sanitization schemes, they do nothing to provide a metric that might be used to gauge the efficacy of a sanitization scheme.

In light of these observations, two attack and evaluation frameworks have very recently been developed. These frameworks take into account the interdependencies between different fields in a trace and infer sensitive information from relations between these interdependencies across traces. Above all, they mathematically formulate the attack process and attempt to derive a privacy metric that a data remitter can utilize prior to publishing a dataset. We discuss these frameworks below and extend the first one in Chapter 3.

2.3.1 Evaluation Framework

In [25], Coull et. al. develop a mathematical, information theory centric, framework for the analysis and evaluation of network trace anonymization. They utilize the concepts of entropy and mutual information [33] to derive probability distributions of identities of unique objects within an anonymized trace.

In their analysis, they define an object by a set of distributions on the features of the network data (header attributes), which represent its presence in the data. Therefore, within their framework, the goal of the adversary is to create a bijective mapping between an anonymized object and its unanonymized counterpart. In order to do so, they perform an implicit time-series analysis on the anonymized data. Note that they assume that the adversary possesses the unanonymized trace. This is done to model the attacker's external network information.

The first step after the isolation of unique objects from the trace is feature extraction and selection. In order to do this, they record the inter-record and intra-record correlations between the various occurrences of the object, thus creating an implicit time series. Once the feature extraction is complete, they select a mutually independent set of the features. They do this by grouping together the fields whose mutual information across their marginal distributions was higher than a defined threshold.

In order to quantify the degree to which the anonymized objects are distinguishable, they compare the features of an anonymized object to those of all unanonymized object using the L1 similarity metric. They use this comparison to derive a probability distribution on the potential true identity of the object. Finally, they use the resultant probability distribution to calculate an entropy measure that provides an intuition on the efficacy of the sanitization scheme.

Though this framework elegantly captures the interdependencies between various attributes of a sanitized trace, its flaw lies in its design. This work assumes that the adversary has access to the raw, unsanitized trace. While this is done for calculating worst-case probability, it defeats the purpose of evaluation. This scheme will always provide an underestimation of the sanitization scheme. This scheme also does not take into account the notion of universal information that we introduce in this thesis. Lastly, Coull et. al. fail to provide an explicit privacy metric. We address this in Section 3.3.

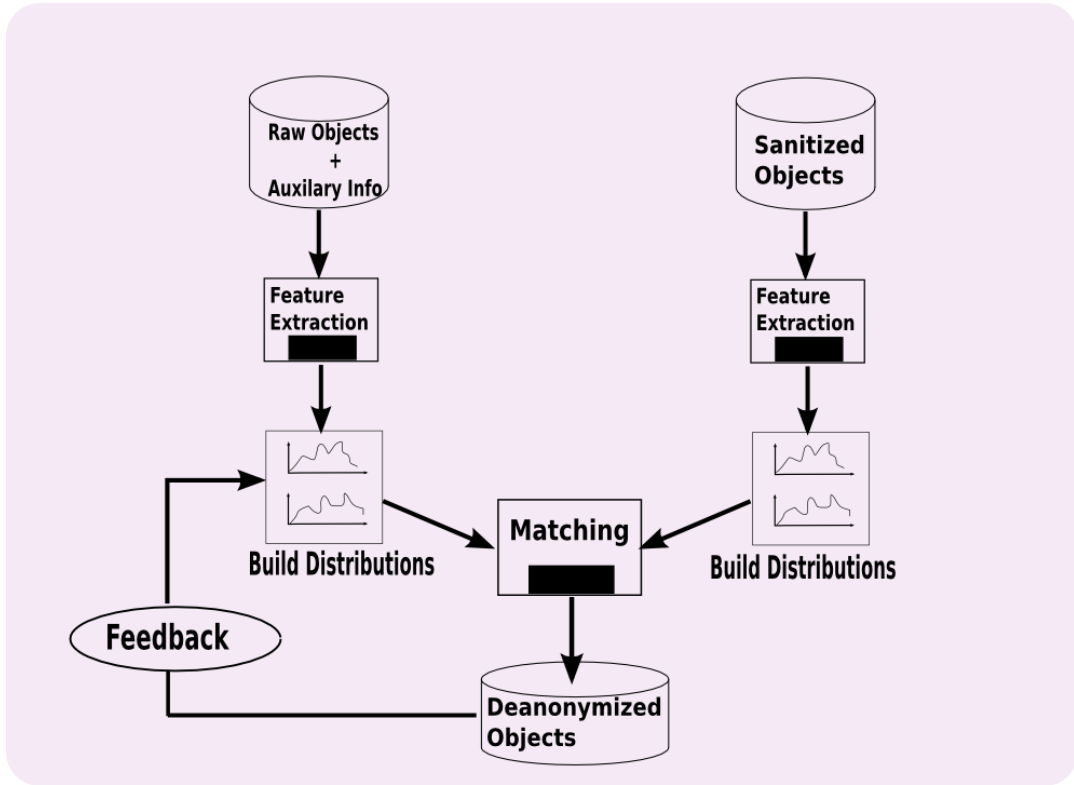


Figure 2.1 Evaluation Framework suggested by Coull. et. al.

2.3.2 Attack Framework

Inspired by the same factors as above, Towsley et. al., in [4] develop a novel, systematic, mathematical attack framework applicable to prefix-preserving anonymization. They develop a general attack that can be executed by an adversary in possession of a modest amount of public information about the target network. We present their attack algorithm below. Note that in interest of brevity, we omit the mathematical details of the framework.

Equally applicable to full and partial prefix preservation, the attack formalized by Towsley et. al. consists of three distinct steps. In step one, the adversary derives traffic information for each host in the anonymized trace. In this step the adversary utilizes attacks outlined in Section 2.2 to compile a trace fingerprint of each unique object in the anonymized trace.

In step two, the adversary gathers external information about the real network of hosts believed to be present in the trace. The host properties collected are similar to those presented in step one. Since these external fingerprints can be imperfect, they define a cost function which is applied to pairs of fingerprint attributes and acts as a measure of the adversary's belief in the mapping between a trace and an external fingerprint. They define the cost function to be zero when the trace fingerprint equals the external fingerprint. They visualize these fingerprints as unordered trees constructed using the natural subnet hierarchy.

Step three produces a set of de-anonymization functions that are good approximations to the actual de-anonymization function. The trace and external fingerprints along with the cost function are given as input to step three. They model this attack as a constrained tree edit distance problem for unordered trees.

Once again, this framework assumes that the information gathered from external sources and that contained the network trace is an accurate representation of the target environment. It thus suffers from the ignorance of the information model that we introduce later. Also, the choice of a good cost function is a limiting factor for this framework.

CHAPTER 3. REFERENCE MODEL FOR TRACE ANONYMIZATION

We have established, thus far, that in order to conduct network security experiments, researchers require isolated testbeds that can recreate actual network conditions under controlled conditions. Ideally, the background traffic in such testbeds would be realistic, repeatable and tunable to allow for accurate and repeatable testing. Perhaps the most difficult and constricting problem that researchers are facing today is the scarce availability of actual network traffic. This data is required to construct statistical models that lie at the core of background traffic generators. Even the grand unifying view of the NetBottle project is constricted at its first layer due to the lack of realistic data, and the continued reluctance of enterprise organizations in sharing their network activity logs.

While several attempts have been made to address these privacy concerns, we contend that they attend only to parts of the problem and lack formal structure that is required to solve the data collection and sanitization problem. The tools that we have discussed so far are accurate and robust implementations of sanitization techniques, but they provide no privacy guarantees on the efficacy of these schemes against inference attacks. The attack and evaluation frameworks discussed in Chapter 2 provide a formal methodology of perpetrating attacks against sanitized traces and claim to provide worst-case privacy metrics. They do so however under questionable assumptions, due to which the utility of their work seems overly optimistic. Thus, past work has failed to build a comprehensive framework that is capable of formally modeling the sanitization problem and identifying

the interfaces between the various entities in the system. Especially, from the data contributor's viewpoint.

In this chapter we offer such a reference model that formalizes the sanitization problem from the remitter's viewpoint. We identify the entities involved in the sanitization problem and provide a functional model for a data collection and sanitization tool. Based on [25], we also provide a usable information theoretic privacy metric for a sanitized trace. With the aid of our model we are able to identify open problems that must be addressed and pave a roadmap for future work in the area.

3.1 Entities

Traditionally formalization attempts at the sanitization problem have only considered three entities in the system [25, 4, 21], namely: *collector*, *analyst*, and *adversary*. While these three entities are adequate for analysis and attack research purposes, they represent an incomplete view of the system for the data publishing organization. Ideally the organization that collects and sanitizes activity logs from their network, requires an internal auditing entity that emulates the adversary in order to evaluate the privacy, accuracy and usability of the network trace.

In order to allow for such a body in the system, we categorize entities in our model into two broad classes: *remitter of data* and *interpreter of data*.

3.1.1 Data Remitter

This category comprises of two entities: the *collector* and the *auditor*. Collectively these two entities are responsible for network trace collection, sanitization, and publication.

3.1.1.1 Collector

Given the scale of modern network, it is not feasible to collect data at every edge within the network. The collector's goal is to identify the most important network edges, collect and aggregate data across these edges and uniformly sanitize the data in a way that maximizes the efficiency and accuracy of the analyzer's task, while minimizing the efficiency and accuracy of the adversary's attempt to infer the raw, sensitive data.

3.1.1.2 Auditor

While the anonymization tools used by the collector provide guarantees on the cryptographic security of the functions used for anonymization, no anonymization system provides privacy and utility guarantees on the output trace. Therefore the need arises for an explicit entity that can provide guarantees on the privacy, usability and accuracy of sanitized activity logs to the collector before a trace can be published.

The auditor shares a threat model with the collector and utilizes the privacy policy and usability policy, to determine the efficacy of the anonymization scheme and the utility of the trace to the analyst. This is a feedback-controlled mechanism between the collector and the auditor which makes the privacy and usability policies dynamic.

3.1.2 Data Interpreter

Analysts, who intend to use published network traces for security experimentation and *adversaries*, whose goal is to recover as much of the original, sensitive information from the sanitized trace, together form the category called interpreters.

3.1.2.1 Analyst

The analyst gathers network trace data from one or several remitters and utilizes it to construct various statistical models of the network topology, services running and

host/user behavior. Analysts are expert, cost-effective researchers who can utilize realistic network data from an organization in order to accurately map the organization's network in a safe sandboxed environment and run security experiments.

3.1.2.2 Adversary

The adversary, while also an interpreter of data, has malicious goals. The adversary aims to create a bijective mapping between the unanonymized trace and the sanitized trace, and has expert knowledge to perpetrate attacks of the nature described in Section 2.2 against the published data.

3.2 Reference Model

We offer a reference model for network trace anonymization that adequately captures all the entities that form the sanitization problem. Our model harnesses past work that has been done in this area and unifies it in a functional way along with identifying missing pieces of the puzzle and completing the big picture. As mentioned above, we divide the problem into four basic entities, collector, auditor, adversary, and analyst and build functional models for these entities. We build a complete functional architecture for the collector/anonymizer, one that supports both offline and online anonymization, and develop a reference model for the auditor. Adversarial models have been developed in [9, 26, 27] and [4], though not without some questionable assumptions, and as such have been given due attention in Section 2.3.2. We show in Chapter 4 how some of these assumptions, while not stated explicitly in past work, inadequately represent information flow in the system and as such illustrate the optimism inherent in past work. It must be noted that the inclusion of the analyst in our reference model is only for the sake of completeness and a comprehensive formal model of the analyst is beyond the scope of this thesis.

Figure 3.1 below depicts our reference model for network trace anonymization. Before we dive into a discussion of each of the elements of this reference model, we provide a data model for the sanitization problem.

3.2.1 Data Model

Our data model begins with *universal information* of the enterprise network where the trace data is to be gathered. This universal information represents the accurate and comprehensive truth regarding the state of the network. This information can be viewed as comprising of complete knowledge of network topology (vertices and edges connecting them), unique host identity and operating system, user/service information, and security devices within the network. The *raw data* collected by an organization (depicted by R in Figure 3.1) is a subset of this universal information. There are various elements of the universal information that cannot be captured in the raw trace during collection phase.¹ The collector transforms this raw trace data to obtain the *sanitized trace* (represented by $T(R)$ in Figure 3.1) and the *metadata* associated with this sanitized trace (represented by M in Figure 3.1). The sanitized trace is produced under the constraints of the privacy policy and the usability policy and as such the data publisher must define the characteristics and semantics of each field in the sanitized trace. This ensures that an analyst can treat each field within the anonymized trace appropriately with respect to the value that it contains. The accompanying metadata must contain several pieces of information to impart research value to the anonymized trace. The metadata must provide the IP prefixes of the local subnets that appear in the data along with their anonymized counterparts. It should describe the packets that had incorrect checksums, if any, and the edges within the network which were tapped to collect the data. The metadata must also include the explicit anonymization policy that was used to gather

¹We expand on this notion of information in Chapter 4 and provide a probabilistic interpretation of this idea and its implications

the trace data. This data is necessary to inform the analyst of the type of transformation that was applied to a field of data.

3.2.2 Auditor Model

The auditor is an entity, internal to the contributing organization, which works with the collector to guarantee the privacy, accuracy and usability of a sanitized trace. Within our model, the auditor is the only entity that has access to the universal information of the network and its components. This information feeds the functions utilized by the auditor to develop the privacy, utility and accuracy metrics.

The auditor emulates the role of an adversary in order to derive the privacy guarantee for a sanitized trace. We utilize the framework developed by Coull et al. in [25] to model the auditor as the adversary, augmenting the information base of the auditor with universal information of the network. The auditor performs a time series analysis on the objects present in the anonymized as well as the raw trace to derive probability distributions of the features of the objects. An object within a trace can be either a unique host or a particular user, and its features are the shared, semantically meaningful relationships between its occurrences in the time series that is the data set. Once the auditor has derived probability distributions for all anonymized and unanonymized objects, statistical similarity metrics can be utilized to compare the feature distribution of an anonymized object with that of all unanonymized objects. Using this approach, the auditor can create a probability distribution of the true identity of the anonymized object. Thus the auditor can derive a pessimistic bound on the privacy of the anonymized trace assuming the availability of complete raw information.

It must be noted however, that since the raw trace is only a subset of the universal information, even under perfect deanonymization, a higher degree of uncertainty remains in the probability distribution of the unanonymized object than is acknowledged in [25]. However, since the auditor possesses the universal information, he is able to further

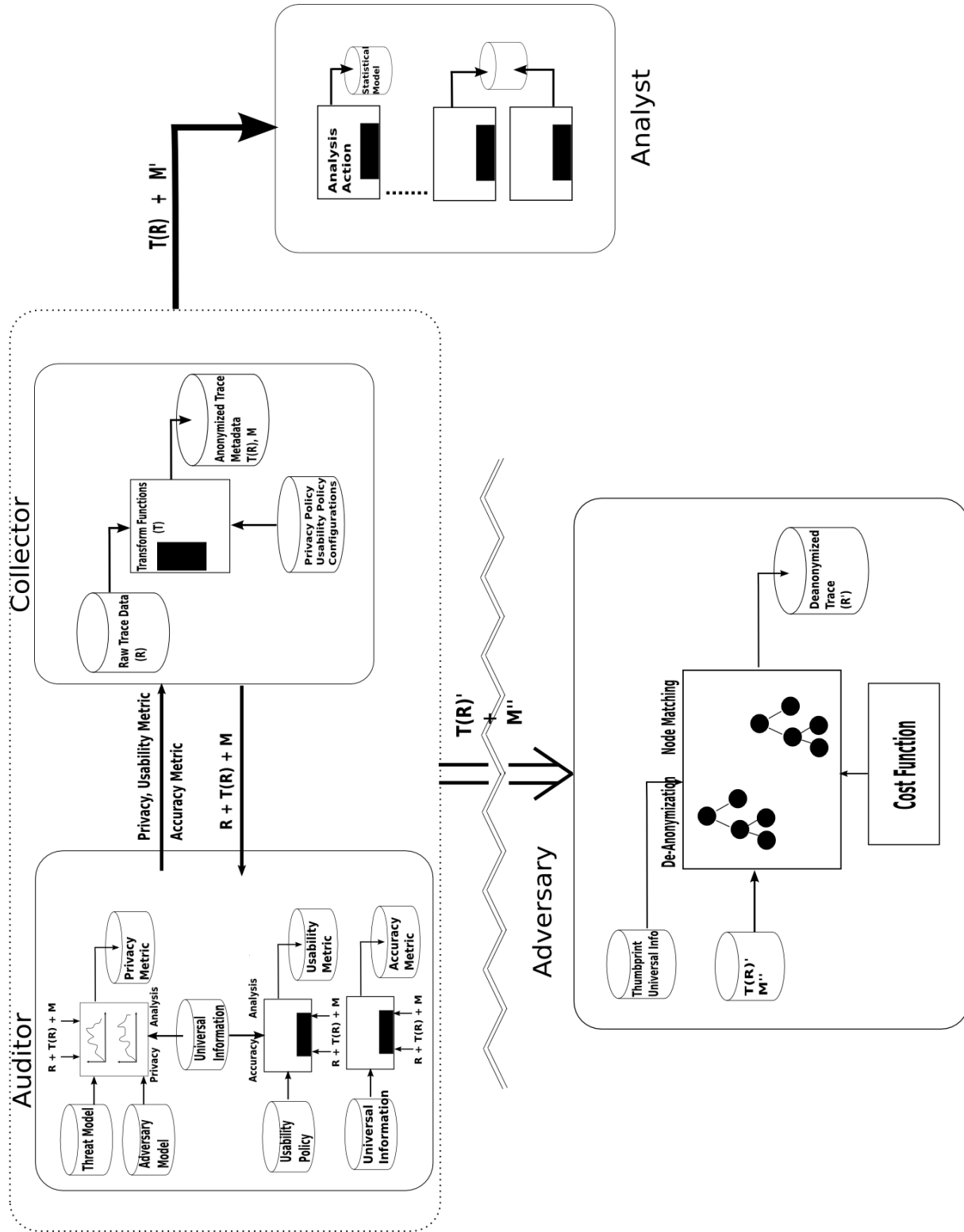


Figure 3.1 Reference Model for Network Trace Anonymization

interpret the true privacy guarantee provided by the sanitization process. We provide a detailed understanding of this notion in Chapter 4.

While privacy guarantees are important to the publisher of a sanitized log, it is equally important for the publisher to provide the analyst a guarantee regarding the utility of the trace. Therefore it falls to the auditor to calculate a utility metric of the anonymized trace. The auditor leverages the usability policy, defined by the collector in conjunction with the analyst, to derive such a metric for the trace. A simple method [21] of deriving such a metric is for the auditor to assign a negative score for each field in the trace that is anonymized, and a positive score for a field left in its raw state. Care must be taken to assign appropriate weights to these scores, as each field is not of equal importance in the trace. The usability policy can prove handy for this purpose. However, even with weighted scoring, this method fails to capture the interdependencies between attributes and hence provides a flawed metric. Further work, beyond the scope of this thesis, is warranted in order to develop a robust utility metric.

Just as privacy guarantees for the sanitized trace are of paramount importance to the data publisher, so is the accuracy of the trace to the analyst. The sanitized trace must accurately represent the environment in which the activity log was captured, or else any experimentation done with the use of such data would be misguided and flawed. Once again it is up to the auditor, in possession of the universal information, raw and sanitized traces and the metadata to provide accuracy guarantees to the publisher and the analyst. Comparing the anonymized trace with the raw trace, and cross-referencing the matching results with the universal information can reveal such an accuracy guarantee.

3.2.3 Collector Model

Privacy preserving traffic collection is at the core of our reference model. As we have discussed in this work, network monitoring has severe drawbacks in terms of privacy infringements, even when data capture is restricted to the header part of the transmitted

packets and excludes the user payload data. The offline anonymization paradigm — gather raw data first, then anonymize before distribution, is plagued by the fact that the collector domain might itself be vulnerable to attackers and malicious insiders and that trace logs might need to be stored for long periods of time. Therefore it is necessary to investigate further the problem of collection and sanitization within this domain. To address this problem and the problem of data aggregation, we propose a functional architecture for a collection and sanitization tool based on the security principles of, *least privilege, minimum time in raw state, restricted access, complete mediation, and fail safe defaults*. Our architecture, in its simplest distributed form, is presented in Figure 3.2.

3.2.3.1 Dirty State

We divide our architecture in to two tiers, namely, *dirty state* and *clean state*. This nomenclature is used to identify the condition of data passing through the functional modules of the system. Dirty state, as its name implies, signifies the presence of private and sensitive information within the data that needs to be sanitized as it percolates through the model. This state comprises of a *sniffer*, a *classifier* and an interleaved stage of *data analysis actions*.

Sniffer Given the scale of modern networks, it is not feasible to collect data at every edge within the network. The problem, then, is given a certain level of resources, to determine the edges within the network that are the most important to creating the emulation environment. It is possible to borrow work from network attack attribution, to determine the edges most likely to receive the bulk of the traffic. This component then is the classic network traffic sniffer that is at the heart of several network protocol analyzers like tcpdump, wireshark, etc. This component takes into account persistent state information, such as software services, operating system services and protocol filters

as required by a network administrator.

Classifier Engine The sniffer module feeds network frames to the classifier engine. This module is an enhanced protocol analyzer that examines network protocol states and according to the classification rules, outputs protocol events. Functions such as TCP de-fragmentation and stream reconstruction are performed at this level to support multilayer analysis and sanitization.

Analysis Actions Since the NetBottle approach requires data to be aggregated, anonymization in isolation cannot occur, as it would prevent a proper analysis of the collected data. For example, we can imagine packets that have their timestamps altered to prevent fingerprinting attacks. The approach presented by Pang, Allman, Paxson and Lee would be to simply apply a counter as the timestamp, maintaining the order of the packets, but not the actual timing information. Unfortunately, without accurate timing information, it would be impossible to determine the proper ordering for packets collected at two different locations in the network. This illustrates the need to maintain some information, for the packets to be properly aggregated. However, to prevent sensitive information from being propagated into the model, it would be beneficial to only have the information critical to analysis be present within the datasets.

In our model, we extend the idea of anonymization requiring 2 passes proposed by Pang et al., to conducting multiple anonymization passes, interleaved with the analysis steps. It is then possible to construct an anonymization policy, based on the analysis needs and the desired level of privacy, to determine when in the analysis process certain pieces of information are removed or anonymized. We omit the depiction of this feedback mechanism in Figure 3.2 in favor of simplicity.

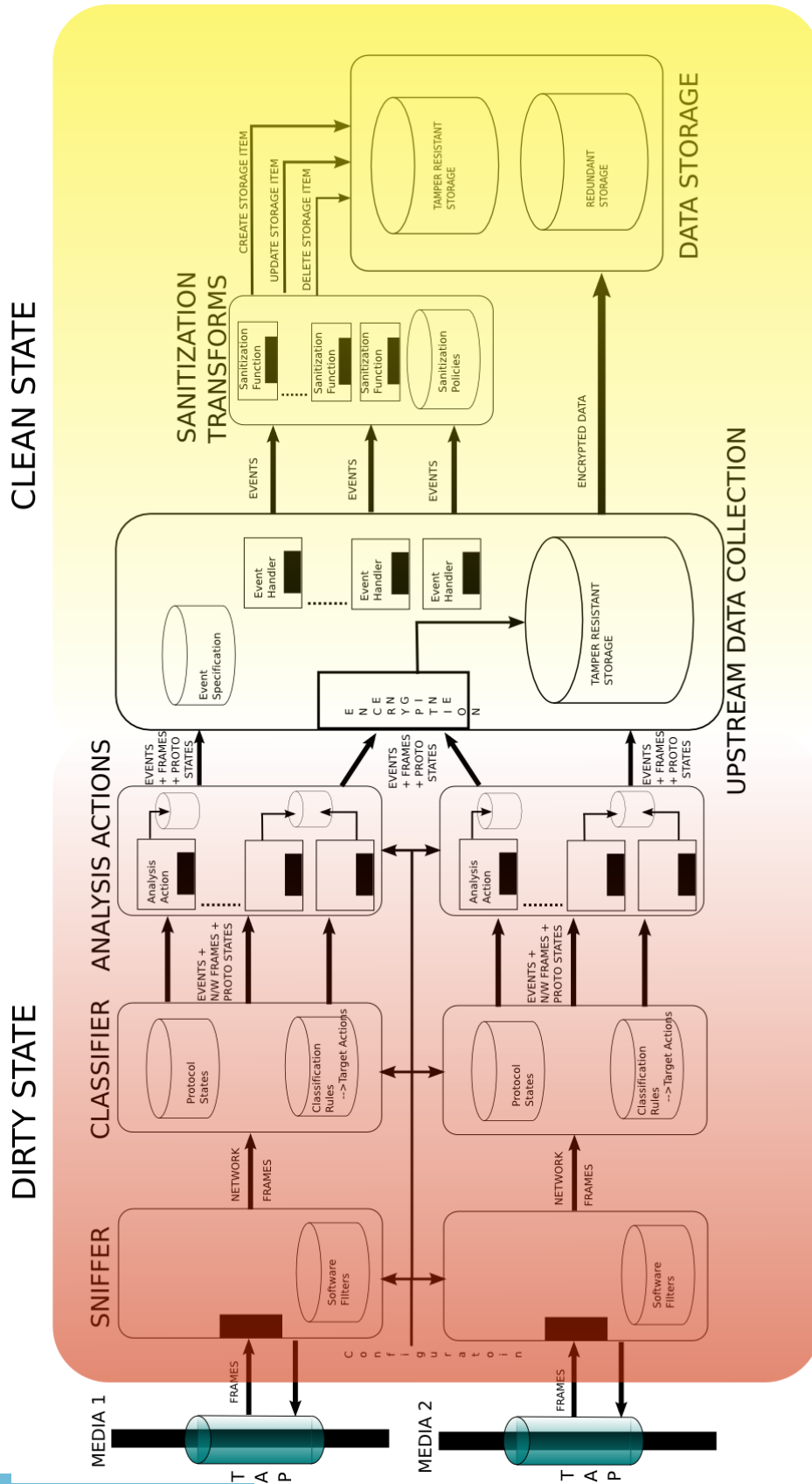


Figure 3.2 Functional Architecture for Trace Collection and Sanitization

3.2.3.2 Clean State

As implied by its name, data in this state is secured and sanitized and the anonymized trace is output at the end of this stage.

Upstream Data Collection Even though offline anonymization can introduce several privacy and security problems, it is a desired feature in any collection and anonymization tool due to its simplicity and ease of data aggregation. The upstream data collection module in our architecture supports this feature and provides the functionality split that is required for online anonymization. At this stage, the raw events, frames and protocol state information generated by the classifier engine can take two directions. This module can be configured to aggregate and encrypt this traffic and collect it in a tamper resistant storage as outlined in [29]. The data, which was in its raw form up till this point, is then secure in a cryptographically secure storage and available, only through restricted access, to the remitter for in house analysis or offline anonymization and aggregation. On the other hand, the aggregated raw data can be fed to event handlers, which utilize the event specification definitions to concurrently feed the raw events to the next module to be appropriately sanitized.

Sanitization Transforms and Data Storage The sanitization module takes pre-defined events as inputs from the upstream data collection function and utilize cryptographically secure anonymization transforms to them. This is conceptualized as an extendable and flexible layer that can be augmented with new and improved sanitization functions as new research discovers them. The sanitization functions are driven by a user defined anonymization policy which the collector derives from the privacy policy and the usability policy.

Once the data is appropriately sanitized, it is fed into redundant data storage. This data storage is similar in design to the tamper resistant storage defined above.

3.3 Privacy Metric for Sanitized Network Logs

The reference model for trace sanitization provides for an auditing entity that can derive privacy, usability and accuracy guarantees for a sanitized trace. By doing so the auditor provides the collector and the analyst a measuring stick against which they can gauge their confidence in the value of a published dataset. In order to develop such a metric, we utilize the work done by Coull et al. in [25]. We extend their approach with our notion of universal information and express privacy in terms of bits of entropy lost post deanonymization.

The privacy metric we propose follows the procedure depicted in Figure 3.3.

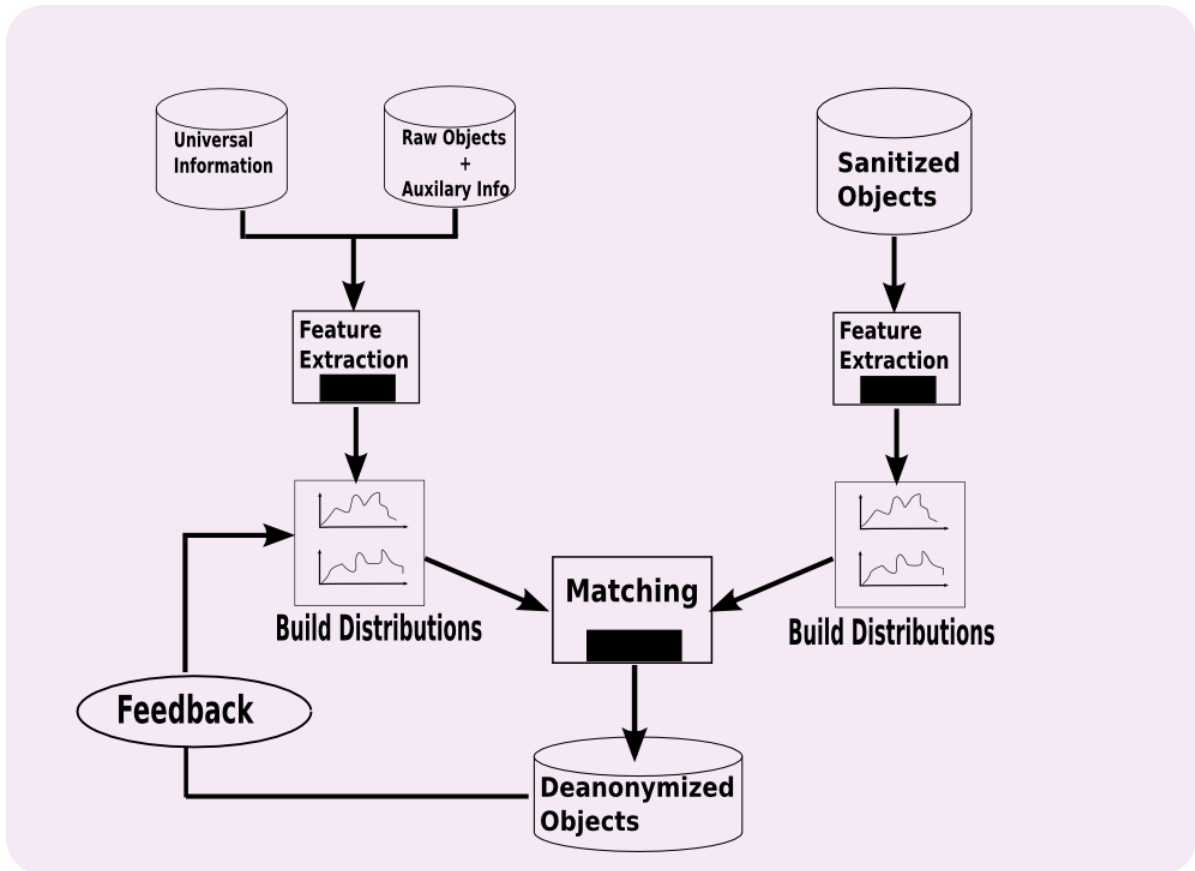


Figure 3.3 Procedure to Derive Privacy Metric

The auditor models an adversary's auxiliary information regarding the organization's

network and builds initial probability distributions of the identity of each anonymized object. We call this probability distribution the *a priori identity distribution*. Next, as described in Section 2.3.1, the auditor identifies objects from the anonymized and unanonymized traces and extracts the probability distributions of each feature for every object. Upon extracting these feature distributions, the auditor matches the complete feature distribution of each anonymized object with the feature distributions of all unanonymized objects. Using a similarity metric, the auditor then derives a probability distribution of the true identity of the object. We call this probability distribution the *a posteriori identity distribution*. For an object (O), the entropy of the a posteriori identity distribution (H_T) will be less than the entropy of the a priori identity distribution (H_P).

$$\implies H_T(O_i) \leq H_P(O_i)$$

The information gained about the identity of an object can then be specified in terms of the difference in the a priori and a posteriori entropies of the object.

$$\implies \Delta H(O_i) = H_P(O_i) - H_T(O_i)$$

Therefore the total number of bits of information gained is expressed by:

$$\sum \Delta H(O_i) = \sum_{i=1}^N (H_P(O_i) - H_T(O_i))$$

Where N is the number of objects.

The auditor can use this estimation of information gained from a deanonymization attempt to bind it to a ceiling defined by the privacy policy. However, it must be noted that since the data present in the raw, unanonymized trace is only a subset of the universal information, even perfect deanonymization does not reduce the entropy of the trace to zero. There is still a degree of uncertainty present in the deanonymized trace due to the inherent lossy nature of the raw trace.

CHAPTER 4. INFORMATION FLOW MODEL

Traditionally, research in the area of network traffic sanitization has utilized a very optimistic view of the actual information captured within a log of network activity. The attacks identified in past research implicitly assume that the network information captured in the raw trace is complete, and accurately represents the entire environment of an enterprise network. While this assumption simplifies attack formulation and analysis, eventually the deanonymization results derived from such analysis overestimate the accuracy and applicability of the attacks.

In this section we expand upon the idea of *universal information* that we introduced and briefly described in Chapter 3. The notion of universal information implies that the inferable information captured in a raw network activity log is an incomplete, and possibly inaccurate, representation of the original enterprise network where the trace was recorded. The universal information is the actual, complete truth regarding the components of the network and the users of these components. The universal information is a complete record of the true identities of all vertices in a network, the edges that connect these vertices, users that operate the nodes, their behavioral profiles, and the protocol statistics for all the nodes.

The raw data captured from an enterprise network, even at best, is only a proper subset of the universal information for the network and cannot capture the universal information set in its entirety.

4.1 Information Flow

Recent attack research, attempting to extract sensitive information from a sanitized trace, implicitly assumes the universal information set to be completely represented by the raw activity capture trace. This assumption is very questionable since under the notion of universal information, even perfect deanonymization leaves a degree of uncertainty in the deanonymized trace. This is due to the intrinsic limitation of the raw trace. Figure 4.1 represents the true information flow model.

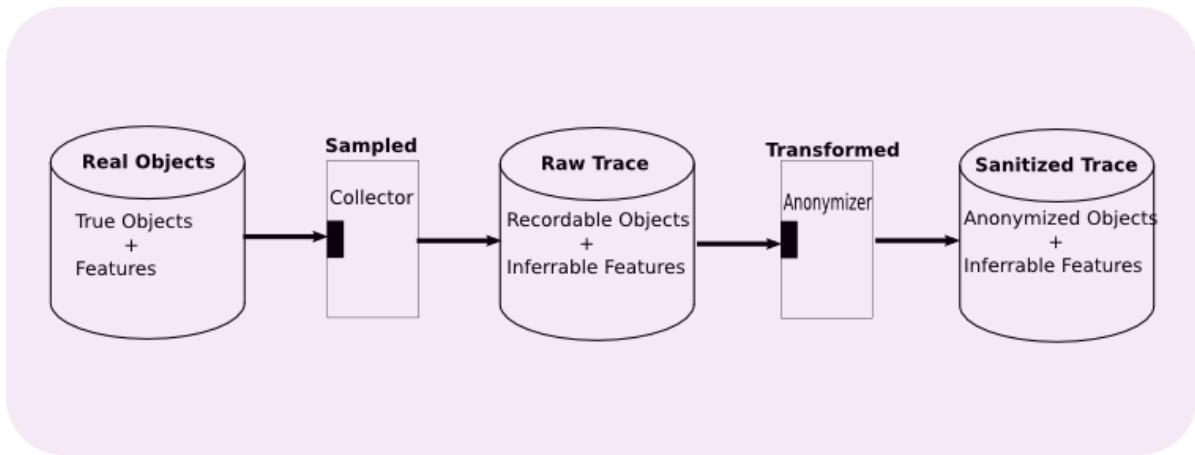


Figure 4.1 Information Flow Model

This model shows the actual flow of information from the universal information set to the captured and sanitized trace. The information contained in the sanitized trace is bound by the information within the raw trace, which in turn is sampled from, and hence a subset of, the universal information of real objects. Figure 4.2 utilizes venn diagrams to represent the three possibilities of information objects, as captured in traces. U_O represents universal information for all objects and R_O represents the raw recordable information for all objects within the network.

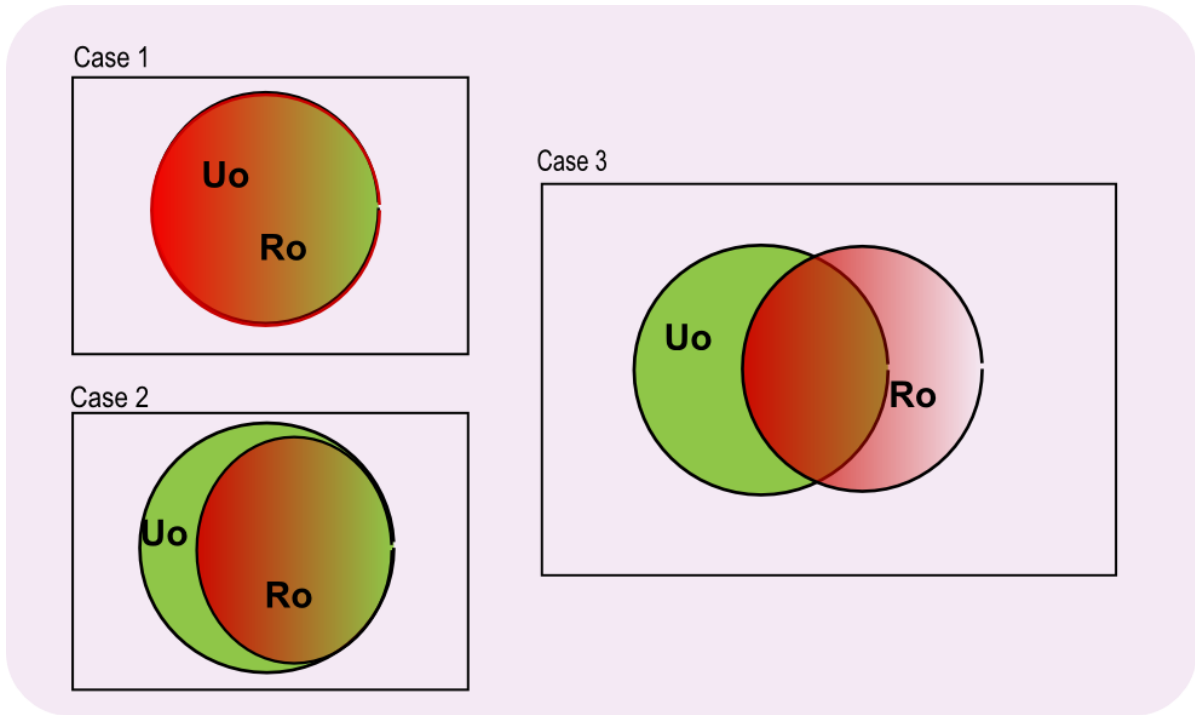


Figure 4.2 Information sets

4.1.1 Case 1: $U_O == R_O$

This case, while implicitly assumed in all past research work, is least likely to happen, and is almost never observed in any raw network activity log. In this case the universal information for all objects in an enterprise network is completely captured within the raw trace recorded by the collector.

4.1.2 Case 2: $R_O \subset U_O$

This case, while theoretically possible, is still much less likely to appear in a real dataset. In this case, the real information is a proper subset of the universal information. While this case is a more accurate assumption to make for modeling purposes, this case does not yet accurately represent an enterprise environment in the collected dataset.

4.1.3 Case 3: $R_O \not\subseteq U_O$

This is the most likely to happen case that represents the information captured within a network activity log. As indicated in Figure 4.2, any raw network trace captures only a part of universal information. Due to physical attributes of networks and the nature of network usage, the activity logs also capture features that are not present in the universal information. This presents a skewed picture of the actual universal information as inferred from a raw trace.

In order to drive home this notion, we provide a few concrete examples that illustrate the cases of information that is present in universal set but not in raw capture set and vice-versa.

4.1.3.1 Example 1: $|U_O - R_O| > 0$

Several elements of the universal information set are either omitted from or inaccurately represented in raw trace captures. As mentioned previously, this causes an added degree of uncertainty to anonymized traces which the past work does not take into account. Quiet hosts are the simplest examples of omitted information from this equivalence class. The universal information set accounts for all hosts within the network. It is highly likely for several hosts to be *quiet* (lunch hour!) or even *offline* (maintenance) during trace collection. These hosts will not appear as objects within an anonymized trace or even the raw trace. Hosts that are behind a *Network Address Translator* (NAT) will only ever appear to be one single unique object in a raw trace, especially in the absence of close manual investigation. Therefore, the raw trace will fail to capture the presence of several hosts that appear in the universal information set. Similarly, any *changes made to host characteristics* (operating system changes, starting/stopping services) will be reflected in the universal information. These changes will be mirrored in adversary's external database, but will not appear in a raw trace captured before such

changes were made.

4.1.3.2 Example 2: $|R_O - U_O| > 0$

Counter to natural intuition, the raw trace information set can easily over-run the universal information set. This leads to the same end result, an unaccounted increase in the uncertainty of the raw trace, as the case in example one does. The one-to-many mappings between a single *DHCP'd* host and its many IP addresses, causes the raw information set to record multiple identities for one host, while the universal information set only records the host's true identity at a given time. *Mobile or guest hosts* that might join the network while activity logs are being collected will also be recorded in the raw information set. The universal set however, will have no persistent record of such entities. Similarly, any *changes made to the network infrastructure* (removed servers or clients) will remain in a raw trace that was captured before the said change, but will not exist in the universal information. An extreme case of this example is when the organization liquidates its network after a trace is published. In this case the universal information set is empty whereas the raw information set is still populated.

4.2 Probabilistic Interpretation

We begin with the universal information set that describes all unique object identities (I) and sensitive attributes (s) belonging to the multi-set of all sensitive attributes.

$$U = (i_k, s_j) : i_k \in I, s_j \in AttributeValue$$

Working from the simplifying Case 2 (4.1.2) above, the network data collected by an organization is a subset of this universal information set.

$$R \subset U$$

We consider an adversary who has access to a released anonymized trace $T(R)$ as well as some subset of U which forms the *external thumbprint* of the objects. The adversary utilizes this information to perpetrate attacks against the sanitized trace with the goal of inferring mappings between the identifiers within the trace and the sensitive attributes. Since it is impractical and limiting to model the external thumbprint, we take the pessimistic view that the adversary is armed with the complete universal information set U .

Therefore the adversary's knowledge of U and $T(R)$ can be modeled in terms of a set of pairs:

$$\{(I_1, S_1), \dots, (I_d, S_d)\}$$

In this set, S_d denotes a multi-set of sensitive attributes of the d^{th} type and I_i denotes the set of identifiers that can be logically associated with the d^{th} attribute type.

Given the published trace $T(R)$ and the adversary's external thumbprint U , the adversary's belief that a sensitive attribute is associated with a unique identifier is quantified by the conditional probability:

$$P((i, s)|T(R), U)$$

As defined previously, the adversary's goal is infer the best mapping between feature/attribute distributions for each unique object. Therefore, A mapping in $(T(R), U)$ is an assignment that matches each element of S_d with I_d . Assuming that $(T(R))$, is a high entropy sanitized trace, the adversary cannot cumulate deanonymized identities to rule out future mappings, making each mapping equally likely. Therefore, for all mutually independent mappings M_1, \dots, M_n under above assumption:

$$P((i, s)|T(R), U) = \frac{|M_k: M_k \text{ having } (i, s)|}{n}$$

The goal of the data publishing organization is to limit the adversary's confidence that a particular feature distribution is associated with any particular object. This implies that anonymization with respect to a sensitive attribute should place an upper bound on the adversary's belief in (i, s) . Thus, $T(R)$ is adequately sanitized if the following condition holds;

$$P((i, s)|T(R), U) \leq c$$

Where c is a parameter ($0 < c < 1$) specified by the auditor and agreed upon by the collector. However since the raw dataset is a subset of the universal information set, we can define an upper bound for this *breach probability*.

$$\forall(i, s) : P((i, s)|T(R), U) \leq P((i, s)|T(R), R)$$

Therefore we can safely conclude that the vulnerability of an anonymized trace is implicitly bounded by the information contained within the raw trace. This information being a subset of the universal information set, automatically introduces a degree of uncertainty within the anonymized trace that is irreducible even under perfect deanonymization.

CHAPTER 5. FUTURE WORKS

In this work we have envisioned a comprehensive reference model that aims to capture the network trace sanitization problem. We have laid theoretical foundations for all the elements required to complete our model. While previous work in this field has addressed parts of this problem, our model identifies several new avenues of future work that are required to solve the problem in its entirety. Below we present a roadmap for future innovations in the field of network trace anonymization.

5.1 Roadmap

Figure 5.1 summarizes our vision of future research work. The red areas indicate those elements that have received their due attention, either in this work or in past work, whereas the green areas indicate the components that have been theoretically described in this thesis and merit more attention in the future.

Special attention is due, to the formal development of a usability guarantee function and metric. Equally important is the formalization of the accuracy metric. Very little work has been done in the past on derivation of explicit privacy and usability policies. For the sanitization problem to be solved, it is imperative that a formal specification language for privacy policy and usability policy be developed. Continued research in the area of cryptographic functions that are at the core of sanitization transforms is also an open research avenue. Finally, the proposed architecture for a collection and sanitization tool merits careful implementation and rigorous testing.

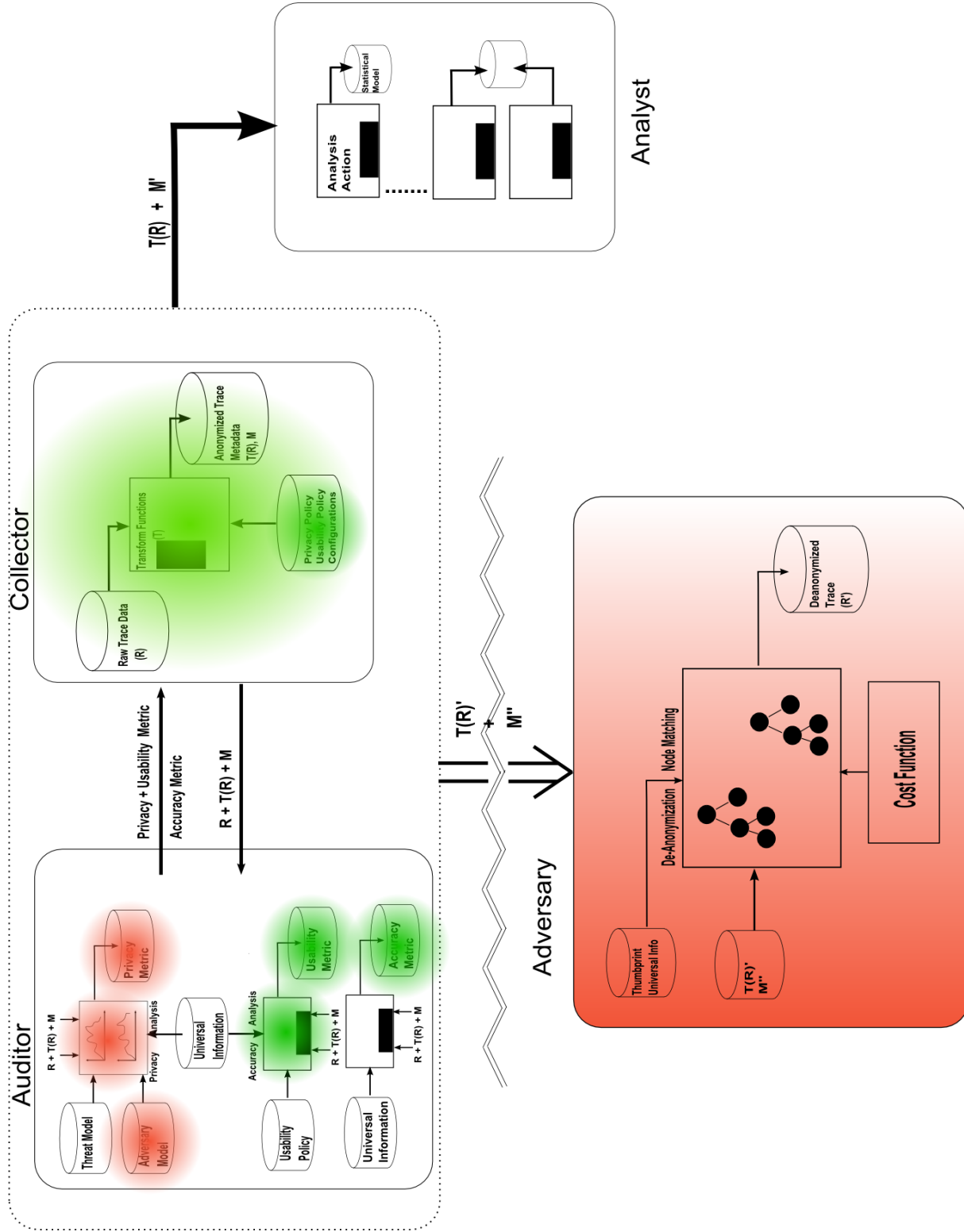


Figure 5.1 Roadmap for Future Work

CHAPTER 6. CONCLUSION

To facilitate network security research, the need for realistic traffic from enterprise networks has grown rapidly over the years. This thesis has addressed several of the issues that plague the availability of this data. We have described a reference model that distills the big picture of the sanitization problem and presents it from the point of view of the organization that is reluctant to share its network activity logs. Our information flow model and its interpretation have extended the threat model typically associated with the sanitization problem with the notion of universal information (Figure 6.1).

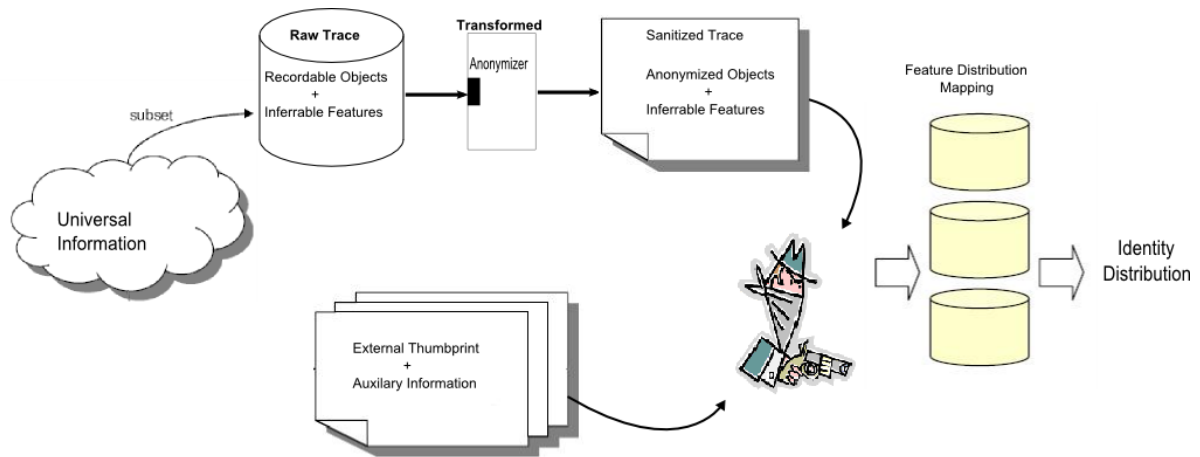


Figure 6.1 Updated Threat Model

This threat model illustrates the over-estimation that is implicit in the results published with previous works on subverting trace anonymization. We have also provided a functional architecture for a configurable and extensible multi-layered trace collection and anonymization tool and an information theoretic privacy metric that can be utilized

to quantify the strength of an anonymization scheme. Finally, we utilized our reference model to pave the way for future work required to solve the *sanitization problem*.

BIBLIOGRAPHY

- [1] A. Slagell, and W. Yurcik, Sharing Computer Network Logs for Security and Privacy: A Motivation for New Methodologies of Anonymization. In *SECOVAL: The Workshop on the Value of Security through Collaboration*, September 2005
- [2] A. Slagell, K. Lakkaraju, and K. Luo. FLAIM: A Multi level Anonymization Framework for Computer and Network Logs. In *Proceedings of the 20th USENIX Large Installation System Administration Conference*, pages 63–77, 2006.
- [3] A. W. Moore and D. Zuev. Internet Traffic Classification Using Bayesian Analysis Techniques. In *Proceedings of ACM SIGMETRICS*, pages 50–60, June 2005.
- [4] B. Ribeiro, W. Chen, G. Miklau, and D. Towsley. Analyzing Privacy in Enterprise Packet Trace Anonymization. In *Proceedings of the 15th Network and Distributed Systems Security Symposium*, February 2008.
- [5] C. Shannon, D. Mooore, and K. Keys. The Internet Measurement Data Catalog. In *ACM SIGCOMM Computer Communications Review*, 35(5):97–100, October 2005.
<http://imdc.datcat.org/browse>.
- [6] CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth.
<http://crawdad.cs.dartmouth.edu>.
- [7] D. Jacobson, ISEAGE Project Overview, May 2007.
<http://www.iac.iastate.edu/iseage/iseageoverview.pdf>

- [8] D. Katabi, I. Bazzi, and X. Yang. A passive approach for detecting shared bottlenecks. In *The 11th IEEE International Conference on Computer Communications and Networks (ICCN '01)*, October 2001.
- [9] D. Koukis, S. Antonatos, and K. Anagnostakis. On the Privacy Risks of Publishing Anonymized IP Network Traces. In *Proceedings of Communications and Multimedia Security*, pages 22–32, October 2006.
- [10] D. Koukis, S. Antonatos, D. Antoniadis, P. Trimintzios, E.P. Markatos. A Generic Anonymization Framework for Network Traffic. In *Proceedings of the IEEE International Conference on Communications (ICC 2006)*, June 2006.
- [11] E. Blanton. *tcpurify*, May 2004.
<http://irg.cs.ohiou.edu/eblanton/tcpurify/>.
- [12] Eddie Kohler. *ipsumdump*.
<http://www.cs.ucla.edu/kohler/ipsumdump>.
- [13] G. Lebanon, M. Scannapieco, M. R. Fouad, and E. Bertino. Beyond k-Anonymity: A Decision Theoretic Framework for Assessing Privacy Risk. In *Privacy in Statistical Databases*, December 2006
- [14] Garimella Rama Murthy, Fundamental Limits on a Model of Privacy-Trust Trade-off: Information Theoretic Approach, In *International Journal of Network Security*, Vol.3, No.3, pages 202–206, November 2006.
- [15] Greg Minshall. *tcpdpriv*, Aug. 1997.
<http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html>.

- [16] J. Fan, J. Xu, M. Ammar, and S. Moon. Prefix-preserving IP Address Anonymization: Measurement-based Security Evaluation and a New Cryptography-based Scheme. In *Computer Networks*, 46(2): 263–272, October 2004.
- [17] J. Xu, F. Fan, H. M. Ammar, and S. B. Moon, On the Design and Performance of Prefix-Preserving IP Traffic Trace Anonymization, In *ACM SIGCOMM Internet Measurement Workshop*, November 2001
- [18] M. Bishop, B. Bhumiratana, R. Crawford, and K. Levitt. How to Sanitize Data? In *Proceedings of the 13th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise*, pages 217–222, June 2004.
- [19] Phillip Porras and Vitaly Shmatikov. Large-scale collection and sanitization of network security data: Risks and challenges. In *New Security Paradigms Workshop*, September 2006.
- [20] PREDICT: Protected Repository for the Defense of Infrastructure Against Cyber Threats.
<http://www.predict.org>
- [21] R. Crawford, M. Bishop, B. Bhumiratana, L. Clark, and K. Levitt, Sanitization Models and their Limitations. In *Proceedings of the New Security Paradigms Workshop*, pages 41–56, September 2006.
- [22] R. Pang and V. Paxson. A High-Level Programming Environment for Packet Trace Anonymization and Transformation. In *ACM SIGCOMM*, August 2003.
- [23] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney. A First Look at Modern Enterprise Traffic. In *ACM SIGCOMM/USENIX Internet Measurement Conference*, October 2005.

- [24] R. Pang, M. Allman, V. Paxson, and J. Lee. The Devil and Packet Trace Anonymization. In *ACM Computer Communication Review*, 36(1): 29–38, January 2006.
- [25] S. Coull, C. Wright, A. D. Keromytis, F. Monrose, and M. K. Reiter. Taming the Devil: Techniques for Evaluating Anonymized Network Data. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium*, February 2008.
- [26] S. Coull, C. Wright, F. Monrose, M. Collins, and M. K. Reiter. On Web Browsing Privacy in Anonymized Net-Flows. In *Proceedings of the 16th USENIX Security Symposium*, pages 339–352, August 2007.
- [27] S. Coull, C. Wright, F. Monrose, M. Collins, and M. K. Reiter. Playing Devil’s Advocate: Inferring Sensitive Information from Anonymized Network Traces. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium*, pages 35–47, February 2007.
- [28] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE INFOCOM*, pages 1582–1592, March 2004.
- [29] Stefan Saroiu, Andrew G. Miklas, Alec Wolman, Angela Demke Brown, Tamper Resistant Network Tracing, In *Proceedings of the ACM Workshop on Hop Topics in Networks (HotNets)*, November 2007.
- [30] Song Luo, Gerald Marin, Generating Realistic Network Traffic for Security Experiments, In *Proceedings IEEE SoutheastCon*, 2004.
- [31] T. Brekne and A. Arnes. Circumventing IP-Address Pseudonymization. In *Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks*, October 2005.

- [32] T. Brekne, A. Arnes, and A. sleb. Anonymization of IP Traffic Monitoring Data - Attacks on Two Prefix-preserving Anonymization Schemes and Some Proposed Remedies. In *Proceedings of the Workshop on Privacy Enhancing Technologies*, pages 179–196, May 2005.
- [33] T. Cover, J. Thomas, and M. Burns. *Elements of Information Theory, Vol. 1, (revised edition)*. Wiley Series in Telecommunications and Signal Processing, John Wiley & Sons, Inc., 2006.
- [34] T. Kohno, A. Broido, and K.C. Claffy. Remote Physical Device Fingerprinting. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2005.
- [35] Weifeng Chen, Yong Huang, Bruno F. Ribeiro, Kyoungwon Suh, Honggang Zhang, Edmundo de Souza e Silva, Jim Kurose, Don Towsley, Exploiting the IPID field to infer network path and end-system characteristics, In *Proceedings of the Passive and Active Measurement Workshop*, March 2005.
- [36] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An Information-Theoretic Approach to Traffic Matrix Estimation. In *Proceedings of ACM SIGCOMM*, pages 301–312, August 2003.
- [37] Z. Zhang, K. Xu, and S. Bhattacharyya. Profiling Internet Backbone Traffic: Behavior Models and Applications. In *Proceedings of ACM SIGCOMM*, pages 169-180, August 2005.