Outsourcing Security Analysis with Anonymized Logs

Jianqing Zhang
Department of Computer Science
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801
jzhang24@uiuc.edu

Nikita Borisov
Department of Electrical
and Computer Engineering
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801
nikita@uiuc.edu

William Yurcik
National Center for
Supercomputing Applications (NCSA)
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801
byurcik@ncsa.uiuc.edu

[1] ***Abstract*— As security monitoring grows both more complicated and more sophisticated, there is an increased demand for outsourcing these tasks to Managed Security Service Providers (MSSPs). However, the core problem of sharing private security logs creates a barrier to the widespread adoption of this business model. In this paper we analyze the logs used for security analysis with the concern of privacy and propose the constraints on anonymization of security monitor logs. We believe if the anonymization solution fulfills the constraints, MSSPs can detect the attacks efficiently and protect privacy simultaneously.**

I. Introduction

As security monitoring grows both more complicated and more sophisticated, there is an increased demand for outsourcing these tasks to Managed Security Service Providers (MSSPs). MSSPs follow a long trend of outsourcing organizational functions. They leverage economies of scale by assembling skilled security professionals and a security support infrastructure that can be shared across multiple organizations [6]. MSSPs can also correlate attacks across organizational boundaries to provide a more effective response [22]. However, MSSPs must handle sensitive data that is either protected by privacy laws, such as employee and customer data, or highly valuable to competitors, such as volumes, applications, or potentially useful to malicious attackers, such as network and system configuration information. For this reason, many organizations are reluctant to form such a close and high-risk connection with an outside security provider and have to either hire expensive security professionals or sacrifice on the level of security protection. This concern over data privacy can serve as a barrier to the growth of the MSSP market.

Our proposed solution to this problem is to perform anonymization on security monitoring logs before they are sent to the MSSP. The logs form the main communication channel between the organization and MSSPs, and by applying anonymization to them, we hope to be able to limit the loss of sensitive information while permitting the MSSPs to perform security analysis. The key challenge will be balancing these two competing needs.

[1]Second International Workshop on the Value of Security through Collaboration (SECOVAL'06), Baltimore MD, USA September, 2006

Our vision for the architecture of future privacy-preserving MSSPs is shown in Figure 1. Communications with outside agents, both benign and malicious, will cause events to be recorded in the security monitoring logs. These events can range from network-level statistics to application-level logs, and can potentially reveal much about the organization's partners, customers, or operations. Anonymization techniques will remove sensitive information, such as identities of communication partners, from these logs before sending them to the MSSP. The logs are sent either in real-time or periodically. The MSSP then performs security analysis on the logs and sends alerts back to the organization. The alerts may reference anonymized identities that the organization can translate back into true ones and take appropriate action against them.

To evaluate the feasibility of this approach, we perform several case studies of common attack types. We analyze what information in NetFlows logs is necessary to detect these attacks, and what may be anonymized away. Based on this, we derive a common set of anonymization criteria for our approach: (1) retaining time interval dependence between records, (2) pseudonymizing the external IP address such that the organization can re-identify them when alerts are submitted, and (3) pseudonymizing the internal IP addresses so that they can be re-identified, as well as preserving some topology information. These techniques still allow the MSSPs to find the attacks, and provide what we believe is a sufficient level of privacy to the organization. We did find that some information, such as port numbers, that may be sensitive to the organization can also be helpful in analysis. In this case, there is a trade-off between the privacy achieved and the security monitoring ability of the MSSPs requiring a more detailed examination of the privacy concern of the particular organization.

Our results are encouraging, and we are hopeful that these techniques can successfully be used to defend against attacks other than those analyzed in our case studies. However, our work is only a first step in developing an architecture for privacy-preserving MSSPs. We identify three key research directions in this area. First, analysis of other attacks than those we studied will help confirm whether the anonymization techniques we defined are compatible with a broad range of

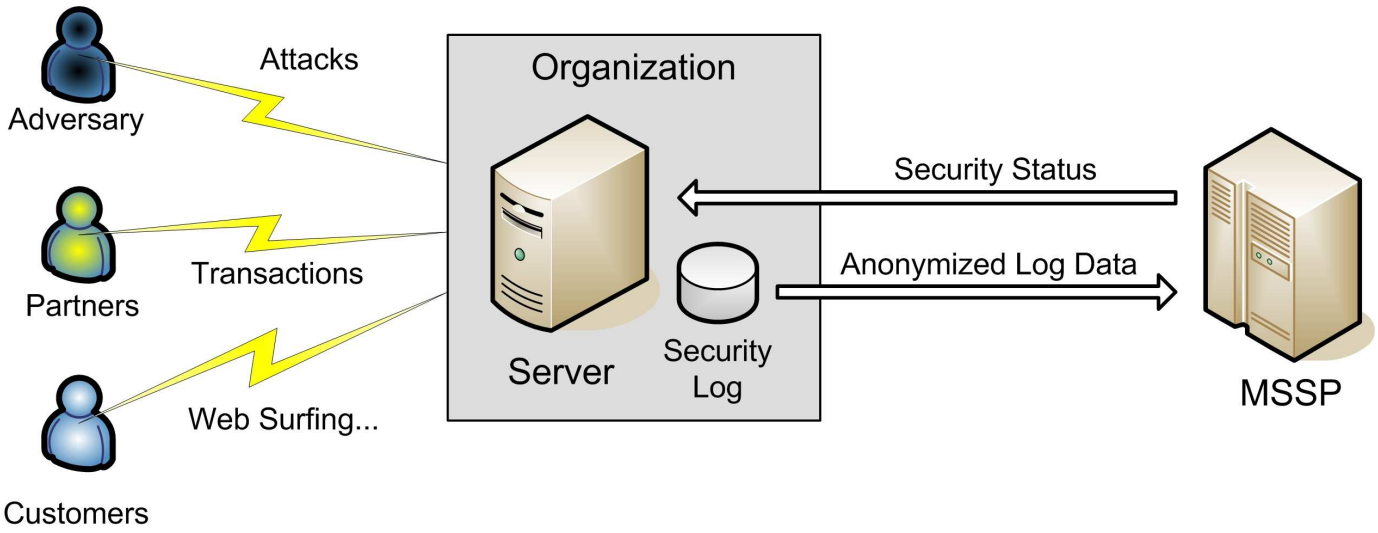

Fig. 1. Future Privacy-Preserving MSSP Architecture

attack types. Second, a study of other log formats and types is needed to identify appropriate anonymization techniques to be used. The third direction is to analyze how MSSPs may correlate attacker activity among logs from multiple organizations, and how the anonymization techniques can be made compatible with such analysis.

The rest of the paper is organized as follows. Section II presents some background on log types and anonymization techniques. Section III presents case studies of attacks and which anonymization techniques are appropriate to detect them. Section IV contains some discussion about the results of the studies and suggests directions for future research. Section V lists related work and Section VI concludes.

II. Background

A computer network contains a variety of different infrastructure devices each of which may be instrumented to produce an audit log file. Though it is possible to detect simple attacks by performing signature matching on byte patterns in streaming network traffic, it is an open problem beyond the scope of this paper how to perform sophisticated security monitoring on streaming data in real-time. As a result, off-line analysis of logs is the common mechanism for security engineers to monitor networks. However, this off-line analysis may approach near-real-time security monitoring, at NCSA we use five minute intervals for NetFlow log analysis which we find is satisfactory.

Logs are not heterogeneous. Although some fields may be common between logs, in general each log type has a different format. Heterogeneity is advantageous for security monitoring since it provides multiple views for enhanced attack discovery [2] and robustness against attack [3]. However, heterogeneity also eliminates possible data management synergies for faster processing from uniform formats.

For the purposes of this paper, we focus on two different logs types, NetFlows as a network-based log and syslogs as a host-based log. This small set of logs are from orthogonal sensors - NetFlows logs describe network traffic activity and syslogs describe host-based operating system events. Together these two logs may provide a relatively complete security view of networked systems depending on sensor placement.

A. NetFlows and Syslogs

a) NetFlow logs: have become the preferred source of security information from network traffic. One NetFlow record is generated for each flow independent of the size of the flow thus providing valuable metadata volume compression for efficient data management. Each flow may be unidirectional or bidirectional depending on the type of NetFlow sensor (for example Cisco NetFlows are unidirectional while Argus NetFlows are bidirectional).[24] The following fields are the minimum set found in each NetFlow record: IP address pairs (source/destination), port pairs (source/destination), protocol (TCP/UDP/ICMP), packets per second, byte counts, and timestamps. While we focus on the use of NetFlows to facilitate security-at-line-speed, other applications for NetFlows include accounting for billing, network management, and network capacity planning.

b) Syslog: is a standard for distributing information about systems by transmitting messages to a remote log at different levels of granularity (*e.g.* warning, error, emergencies).[4] In addition to pattern-matching syslog entries for known attack signatures, other examples of suspicious activity found in system syslogs include critical events (system reboots), unsuccessful login attempts, storage overload when a log fills its allocated disk space, and cessation of logging messages

[2]different logs will detect different attacks and together different logs may detect a wider range of attacks

[3](information in logs lost to a successful attack can be compensated by redundant information found in remaining logs not lost to attack

[4]IETF Working Group on Security Issues in Network Event Logging http://www.ietf.org/html.charters/syslog-charter.html

(may indicate the logging process has either been deleted or a Trojan logging process installed).

B. Common Log Anonymization Techniques

In this section we introduce different log anonymization techniques that have been implemented in tools developed at NCSA – the CANINE anonymization tool for NetFlows logs, the Scrub-PA anonymization tool for process accounting logs.[13], [16]. We also include discussion of how each anonymization technique may affect security analysis. In the practical implementation of these log anonymization techniques, it was an objective to make sure the output anonymized logs are consistent and indistinguishable from the internal format of an arbitrary input log. For this reason, metadata of some form (filename, annotation, etc.) must be used to identify whether a log is an original log or an anonymized log – and if it is an anonymized log then which anonymization techniques were used (with accompanying selection parameters). For CANINE and Scrub-PA we facilitate metadata annotation by providing a print summary option that contains all information about operations executed on an input log file.

1) IP Anonymization: IP address anonymization seeks to hide the source or destination host of a connection. An MSSP would typically be seeking to identify a source IP that may be an external attacker or an internal compromised host. In the security context with the most privacy implications, a destination IP may be an internal compromised host and sharing its identity may reveal vulnerabilities shown in the log that invites future attacks. For the purposes of this paper we focus on three types of IP address anonymization: (1) Truncation, (2) random permutation, and (3) prefix-preserving pseudonymization.

a) Truncation: is the most basic type of IP address anonymization. Here a user selects the number of least significant bits to truncate from each IP address in a log. For example, truncating the rightmost IP address bits that identify individual hosts on a subnet would leave only the network domain address. If bits of the network domain are truncated then some information about the network domain may still remain. While truncation can be used to obfuscate host addresses, a brute force attack can be used to effectively break this technique for a relatively small network domain. Despite this brute force attack scenario, against most adversaries truncation is a satisfactory technique to anonymize host IP addresses on relatively large network domains while still retaining some information about the network domain.

b) Random Permutations: applies a random permutation on the set of possible IP addresses to translate each IP address. A 32-bit block cipher would represent a subset of permutations on the IP space. In this way, it is possible to generate any permutation, not just one from a subset of the possible permutations. However, hosts from the same network in the input log will not be anonymized consistently to the same network in the output anonymized log so some potentially important information structure is lost. A seed can be used to replicate IP address random permutations so the same output log can be replicated from the same input log using the same random permutation algorithm with the same seed. This technique may not be practically reversible if the seed is lost.

c) Prefix-Preserving Pseudonymization: is a special class of permutations that have a unique structure preserving property – two anonymized IP addresses will match on a prefix of n bits if and only if the un-anonymized addresses match on n bits. This allows multiple IP address on the same network in an input log to be mapped consistently to corresponding anonymized hosts on the same anonymized network in the output log. This technique retains potentially important network information structure in that activity can be identified per host within each unique network domain. It has been observed at NCSA that attackers often launch attacks from entire networks or entire subnets they own either legitimately or by compromise so prefix-preserving pseudonymization retains enough information to reveal whether attack sources are coordinated or independent events.

2) Timestamp Anonymization: Timestamps identify events in the time dimension including start and end times, duration, and individual event times. In the security context with the most privacy implications, reporting timing information may allow attackers to probe with self-known patterns that can be correlated with timing information of otherwise anonymized logs in order to map a network for attack (while simultaneously avoiding sensor detection). For the purposes of this paper we focus on three types of timestamp anonymization: (1) time unit annihilation, (2) random time shifts, and (3) enumeration.

a) Time Unit Annihilation: Timestamps and duration can be broken down into the units of year, month, day, hour, minute, and second. Any subset of those units can be annihilated. Timestamps and duration are obviously related such that if a timestamp is partially annihilated then the duration must be adjusted to match and vice versa (if a duration is partially annihilated then the start and end times must be adjusted to match).

b) Random Time Shifts: In the initial discovery of isolated security events, relative timing information is more important than knowing the exact day and time the events actually occurred. With the random time shift technique, all timestamps in a log can be shifted by a random number leaving the relative time intervals between events and time durations intact. The relative time information is critical in determining the type of attack while the exact day and time information of when the event actually occurred lost using this technique is most useful when correlating between isolated events from different sources.

c) Enumeration: In this technique all time information is removed except for the sequence order in which the events occurred. The enumeration algorithm simply chooses a random time for the earliest record and then shifts all starting times equidistant from each other retaining the same sequence order. Corresponding ending times are calculated from the original flow duration. Implementation of this method can be problematic if records within a log are not pre-sorted in time (e.g. NetFlows are typically not entirely presorted in time). With

the enumeration method of timing anonymization, the causal sequence between events remains intact even though the timing interval between events is lost.

3) Port Number Anonymization: Port numbers identify services. Although services do not have to run services associated with their standardized port numbers, for interoperability services typically do run on standardized ports. Port numbers are arguably the most valuable single unit of information to detect attacks – while the use of certain ports are a reliable indicator of distinct attacks, most ports have dual-uses for both legitimate use or malicious attack so context information is important for validation (context information such as IP address, packet or byte counts, and timing information). In the security context with the most privacy implications, reporting port numbers will reveal all services running on a network as well as revealing which hosts are running these services. However, if port numbers are anonymized, MSSP security analysis becomes much more difficult with only context information to infer possible attacks. For the purposes of this paper we focus on three types of port number anonymization: (1) bilateral classification, (2) black marker, and (3) random permutation.

a) Bilateral Classification: Ports numbers range from 0 to 65,535 but can be bilaterally classified as being below port 1024 (well known ports) or above port 1024 (ephemeral ports). This method of anonymization is similar to the truncation of IP addresses in that a subset of ports is collapsed to a single representative within that set. In the security context, knowing whether services on a network are well known services or ephemeral services can be revealing without having the ability to identify the exact service. Capable adversaries can use brute force attacks to effectively break this technique, however, bilateral classification is a satisfactory technique to obfuscate port numbers (and thus network services) against most adversaries.

b) Black Marker: The black marker anonymization technique is the same, from an information theoretic view, as printing the logs and blacking out all port information. In a digital form, we replace all ports with a constant such that no port number information remains.

c) Random Permutation: This technique applies a random permutation on the set of all possible ports (0 to 65,535) to map each port number. A 16-bit block cipher represents a subset of permutations in port space. If port activity can be characterized by a uniform distribution then this mapping provides a measure of obfuscation against most adversaries, however, most port activity has distinctive patterns such that capable adversaries will be able to identify port numbers (and thus network services) using a brute force attack given enough log data containing patterns.

III. Anonymized Traffic Traces for Attack Analysis

In this section, we discuss how to anonymize different fields of NetFlow log entries to detect two types of attacks: port scanning and (distributed) denial-of-services. We examine the information required for MSSPs to detect such attacks and derive constraints constraints on anonymization techniques.

A. Port Scanning

Port scanning is used both by system administrators and attackers to examine the configuration of hosts on the network. Port scans often reveal potential weaknesses by showing which insecure services are running on what computer. Adversaries frequently perform a scan of open ports on a single hosts or a collection of hosts; such scans are commonly followed by attacks targeted to particular applications. We consider two common port scans: portmap scans and individual host port scans.

1) Portmap Scan: The portmap service is a dynamic port assignment daemon for RPC services. It assigns ports to services from a pool and lets clients look up the port assigned to a particular service. A malicious portmap scan can be used to obtain a map of services running on a computer and their port numbers. This map can then suggest the existence of vulnerabilities that may be exploited.

Figure 2 shows an example of log entries for a portmap scanning based on NetFlow. For a typical portmap scan, there are a number of flows from a same source address with different source ports to a variety of destination hosts with a same destination port (portmap daemon) in short time. Usually, the hosts are in the same domain, *i.e.* they have the same subnet number. Each flow contains a single packet using TCP protocol (P=6).

2) Individual Host Port Scan: This scan simply scans all ports on a single host to find running services with exploitable vulnerabilities. Figure 3 shows NetFlow log entries for an individual host port scan. It contains a set of TCP (P=6) flows from a single source host to a single destination that happen within a short span of time and contact incrementing destination port numbers.

B. DoS/DDoS

In a basic denial-of-service (DoS) attack, packets are sent at a high rate from a single source address to a single destination address. Since blocking a single source address with a firewall or router can stop a DoS attack, distributed denial-of-service (DDoS) attacks emerged, where packets are sent from many source addresses to converge on a single destination address with the same result. DoS and DDoS attacks may target the same port or different ports in the destination address. There are many variations of DoS and DDoS attacks, for our purposes we only consider a “SYN flood” DoS and distributed DoS attacks.

1) SYN Flood: An attacker sends a succession of SYN requests to a target system without sending the following ACK messages. Because TCP is a three-way handshake protocol, many half-open connections will be established on victim’s side and the victim’s resources will be exhausted quickly. Some systems may malfunction badly or even crash if other operating system functions are starved of resources this way.

Start time	SrcIPaddr	SrcPort	DstIPaddr	DstPort	P	Pkts
10:53:42.50	165.132.86.201	9781	128.146.0.76	111	6	1
10:53:42.54	165.132.86.201	9788	128.146.0.6	111	6	1
10:53:42.54	165.132.86.201	9791	128.146.0.11	111	6	1
10:53:42.55	165.132.86.201	9381	128.146.0.10	111	6	1
...						

Fig. 2. NetFlow logs of Portmap Scan

Start time	SrcIPaddr	SrcPort	DstIPaddr	DstPort	P	Pkts
18:56:23.916	130.241.53.23	902	128.146.38.15	4138	6	1
18:56:23.924	130.241.53.23	900	128.146.38.15	4139	6	1
18:56:23.936	130.241.53.23	893	128.146.38.15	4140	6	1
18:56:23.944	130.241.53.23	891	128.146.38.15	4141	6	1
...						

Fig. 3. NetFlow Logs of Individual Host Port Scan

Figure 4 shows log entries of a typical SYN Flood attack. In a short time frame, a very large number of SYN packets are sent from the same source host with the same port number, all destined for the same destination host and port (the web server on port 80 in this example) using TCP. The packet size is 40 (SYN request). Each flow is only one packet long since each SYN packet starts a new flow.

If the SYN Flood is launched by a distributed denial of service attack, neither the source IP address nor the source port number will be same. Figure 5 shows the SYN Flood by DDoS attack.

C. Anonymization Constraints on Traffic Traces Logs

From above classic cases we can see that the following attributes of NetFlow log entries are relevant to detect port scanning: 1) Start Time; 2)Source IP Address; 3) Source Port; 4) Destination IP Address; 5)Destination Port; 6) IP protocol type: ”6” stands for TCP and ”17” stands for UDP.

To keep the logs informative enough for MSSP to detect port scanning, the correlation of the above fields value should be retained even if the values are anonymized.

Although the time stamp of *Start Time* could be anonymized, the interval of time between two events must be retained, since an MSSP needs to check whether a series of events happened within a small interval (the threshold of the interval is defined by MSSP, which is out of scope of this paper). This eliminates the *Time Unit Annihilation* and *Enumeration* strategies, as they destroy time interval information, but *Random Time Shifts* preserve the interval information even though the original time value is hidden.

The anonymized value of *Source IP Address*(SrcIPaddr) must be re-identifiable, since an organization must be able to identify and block the adversary when presented with forensic information from the MSSP. This can be accomplished by encrypting the IP address with a key known only to the organization. Since encryption functions offer a permutation operation, it will preserve equality comparisons and allow for an MSSP to detect unusual activity from a single host, which can then be decrypted by the organization for the purposes of taking action.

The *Source Port*(SrcPort) is not useful in attack analysis, but it also tends to carry little privacy value; an organization may choose any anonymization policy depending on its own privacy needs.

The *Destination IP Address*(DstIPaddr) field reveals internal addresses of an organization, which are sensitive. Therefore, anonymization on this field is necessary. If the MSSP reports attacks or suspicious events to an organization’s site security officer (SSO), the SSO needs to figure out which hosts or which subnets are under attack to monitor for weaknesses and guide the deployment of better security measures; therefore, the addresses need to be re-identifiable. In addition, an MSSP can operate more efficiently if the anonymized network addresses preserve the domain structure, as many scans are easy to identify because they sequentially probe all hosts within a single domain. So, we suggest to use a prefix-preserving anonymization strategy.

Destination Port(DstPort) usually indicates particular applications. These applications, such as web server (port 80), FTP server (port 21) and telnet (port 25), are often attack targets. The true values of those ports help MSSPs identify application-specific attacks. On the other hand, the port values also reveal applications running inside the organization, which may be sensitive information. In this case, an SSO might decide to anonymize some more sensitive port values (perhaps only on sensitive hosts), while preserving the more common publicly known port numbers.

Although [13] suggests a simple method to anonymize *IP Protocol Type* (P) by replacing the protocol number with the unused, but IANA reserved, number of 255, we think it unnecessary to hide it. On Internet, most network traffic consists of TCP and UDP. Anonymization of this attribute does not significantly improve the organization’s privacy but it weakens the MSSP’s capability to detect attacks.

We summarize the attributes list, anonymization constraints and recommended anonymization mechanisms in Table I.

D. Active Operating System Fingerprinting

We perform another case study using a different type of log to detect operating system fingerprinting, a method to detect the type of operating system a host is running. Based on the OS type detected, the adversary can select corresponding attacks to exploit the known vulnerability. This method includes sending crafted, abnormal packets to the remote host, and analyzing the replies being returned from the remote host. Different TCP

Start time	SrcIPaddr	SrcPort	DstIPaddr	DstPort	P	Pkts	B/Pk
21:47:11.670	165.132.86.201	514	128.146.97.7	80	6	1	40
21:47:11.854	165.132.86.201	514	128.146.97.7	80	6	1	40
21:47:12.198	165.132.86.201	514	128.146.97.7	80	6	1	40
21:47:12.338	165.132.86.201	514	128.146.97.7	80	6	1	40
...							

Fig. 4. NetFlow logs of SYN Flood in DoS

Start time	SrcIPaddr	SrcPort	DstIPaddr	DstPort	P	Pkts	B/Pk
19:08:40.492	192.1.6.69	77	194.20.2.2	1308	6	1	40
19:08:40.532	192.1.6.222	1243	194.20.2.2	1774	6	1	40
19:08:40.720	192.1.6.108	1076	194.20.2.2	1869	6	1	40
19:08:40.764	192.1.6.159	903	194.20.2.2	1050	6	1	40
...							

Fig. 5. NetFlow logs of SYN Flood in DDoS

TABLE I
ANONYMIZATION CONSTRAINTS ON NETFLOW LOGS

Attr. List	Anonymization Constraints	Recommended Anonymization
Start Time	Retain events interval and time dependence	Random Time Shifts
Source IP Addr.	Anonymized and Re-identifiable	Pseudonyms *e.g.* Shamir's threshold
Source Port	–	–
Dest. IP Addr.	- Retain virtual network topology - Re-identifiable anonymized	Pseudonyms + Prefix-preserving
Dest. Port	More efficient if retained	–

stacks will give different replies and thus allowing the analyzer tool to recognize a particular OS. Active OS fingerprinting is a fast process and a large number of hosts can be scanned in a short time frame. `NMAP` and `queso` are the two most widely used tools for active OS fingerprinting. If the remote host's network is being protected by IDS or firewall devices, such attacks will be detected. To illustrate our work, we show an example of pseudonymized audit entries generated by `tcplog`. The `tcplog` program logs TCP flag combinations that should not occur in regular TCP traffic. Additionally `tcplog` is able to detect `queso` OS fingerprinting scans.

Figure 6 shows the syslog `tcplog` entries. After logging a series of abnormal TCP packets, `tcplog` detects a `queso` OS fingerprinting scan.

From this example, we can see that active OS fingerprinting attacks happen in a short time frame. So the dependence of time of each record should be retained. Therefore, just like logs in traffic trace, `TS` should be anonymized using *Random Time Shifts* or other similar methods.

The host name (`H`) corresponds destination IP address, which reveals internal address of the organization. Because SSO needs to identify which host was scanned after it receives security status report from MSSP, the attribute of `H` should be re-identifiable.

The source port number and destination port number are encoded into the entries. If the destination port is a well known reserved port for particular application, it will be converted to the application name immediately (`ftp` in Fig 6). If it is shared directly, the applications which the system hosts would be revealed. However the MSSP can detect attacks more efficiently because MSSP can correlate the application with well-known attacks. So the anonymization of the destination port number is a tradeoff. It depends on organization's policy. For source port, the situation is same because it may reveal the sensitive information of the partners and/or customers but MSSP can determine the attacks more quickly if the attack is well-known and often use some typical port number. On the other hand, the destination port should be re-identifiable because SSO needs to figure out which part of the system is being attacked.

Like the cases in traffic trace, Source and destination IP addresses should be pseudonymized and re-identifiable. The internal virtual network topology also should be retained. [9] shows an example how to anonymize the source IP address using [2] (see Fig 7). Note that the actual pseudonyms for IP address `192.168.1.4` are linkable by the identifier `d2petb50CXOun4CuLPhluM8` (for simplicity, we only select the case where the source IP addresses are pseudonymized). [9] gives more details about re-identifiable pseudonymization and the example of recovery pseudonymized IP.

The rest of the entry is the encoded message. In active OS fingerprinting example, the attributes between two colons are TCP events (see Fig 6). Because `tcplog` is able to detect `queso`, it indicates that these events form a fingerprinting scan. Since these `TCP` events could be considered sensitive under some circumstance and `tcplog` gives the conclusion at last, it seems reasonable to anonymize them. If the logs are recorded by a common TCP monitor application rather than the one with capability of detecting some attacks, such as `tcplog`, however, these events sequence are important clues to detect the attacks just like `queso`. MSSP needs

```
02:23:37 gary tcplog[1567]:  FIN  SYN  RST                    URG                  : port 41067 from 217.82.199.102 port 42312
13:42:06 gary tcplog[1040]:       SYN                         URG  RES2            : ftp from 192.168.1.4 port 45691
13:42:06 gary tcplog[1040]:  FIN  SYN            PSH          URG                  : ftp from 192.168.1.4 port 45693
13:42:06 gary tcplog[1040]:  FIN  SYN            PSH          URG                  : ftp from 192.168.1.4 port 45693
13:42:06 gary tcplog[1040]:  FIN                 PSH          URG                  : port 34513 from 192.168.1.4 port 45697
13:42:06 gary tcplog[1040]:  FIN                 PSH          URG                  : port 34513 from 192.168.1.4 port 45697
13:42:06 gary tcplog[1040]:  QUESO: port 34513 from 192.168.1.4 port 45697
```

Fig. 6. Example of un-anonymized audit records generated by `tcplog`

```
02:23:37 gary tcplog[1567]:  FIN  SYN  RST                    URG                  : port 41067 from a1 port 42312
13:42:06 gary tcplog[1040]:       SYN                         URG  RES2            : ftp from a2 port 45691
13:42:06 gary tcplog[1040]:  FIN  SYN            PSH          URG                  : ftp from a3 port 45693
13:42:06 gary tcplog[1040]:  FIN  SYN            PSH          URG                  : ftp from a4 port 45693
13:42:06 gary tcplog[1040]:  FIN                 PSH          URG                  : port 34513 from a5 port 45697
13:42:06 gary tcplog[1040]:  FIN                 PSH          URG                  : port 34513 from a6 port 45697
13:42:06 gary tcplog[1040]:  QUESO: port 34513 from a7 port 45697
02:23:37 gary pseudonymizer: Short=a1 Long=RW4gGPc...:ph6kOhKLZBsMw!H7H2ztcsd
13:42:06 gary pseudonymizer: Short=a2 Long=5QgfV3!...:d2petb50CXOun4CuLPhluM8
13:42:06 gary pseudonymizer: Short=a3 Long=sW7bl67...:d2petb50CXOun4CuLPhluM8
13:42:06 gary pseudonymizer: Short=a4 Long=wdEey!p...:d2petb50CXOun4CuLPhluM8
13:42:06 gary pseudonymizer: Short=a5 Long=Fi7o9GJ...:d2petb50CXOun4CuLPhluM8
13:42:06 gary pseudonymizer: Short=a6 Long=rog6EOb...:d2petb50CXOun4CuLPhluM8
13:42:06 gary pseudonymizer: Short=a7 Long=it0SIYe...:d2petb50CXOun4CuLPhluM8
```

Fig. 7. Pseudonymized audit records of Fig 6

these entries to analyze the event sequence by itself. So we think these TCP events should be retained. The constraints for anonymized `tcplog` logs are listed in Table II.

E. Dummy Log Records

In section I we mentioned that the number of events records could reveal the overall resource usage or user behavior in the system, which can be considered sensitive under certain circumstances. One possible remedy is to add "noise" to the logs. Fake entries could be added for existing users [15]. Consequently, the system workload will not be revealed by the number of events entries.

However, these mechanisms decreases the efficiency of MSSP detecting the attacks. First, MSSP may miss some attacks due to the "noise". For example, in the case of active OS fingerprinting, if fake entries are inserted among the `queso` events, the sequence of TCP packets of `queso` will be broken. Thus, MSSP can not detect it (assume the monitor program can not detect it independently). Second, the false positive rate will be increased. For example, if fake entries of SYN packets or failed logins with wrong password are added and the number exceeds the threshold of the security analysis tools used by MSSP, the false alert will be raised. A careful crafting of noise policies can minimize interference with detection at an MSSP; we plan to explore this in more detail in the future.

IV. Discussion

A. Organization Privacy

Based on our analysis, a single anonymization strategy for NetFlows and traffic trace logs is flexible enough to permit analysis of several common classes of attack. However, for this strategy to be useful in an outsourced-MSSP scenario, the strategy must also sufficiently protect the privacy of the organization. So we start by examining what sensitive data may still be left in the logs after anonymization.

Anonymization preserves data about traffic volumes, which may be sensitive. In particular, if logs are sent to the MSSP in real-time, the MSSP will know the instantaneous volumes of incoming traffic processed by the organization. If this data is sensitive, we would recommend a batched upload strategy that, with randomized time shifts, leaves only aggregate volumes visible; this trades off the timeliness of analysis for privacy of traffic volumes. Hiding aggregate volumes is more difficult and perhaps unnecessary for a large number of businesses: frequently this data is not sensitive and may in fact be available through other means. Dummy log records may help hide aggregate traffic volumes, at the cost of sacrificing efficiency at the MSSP.

In addition to volumes, the pseudonymized IP addresses can reveal more detailed information about the organization. Counting the distinct external IP addresses will allow estimates of the size of the customer base, and tracking the appearance of the same pseudonym over time will let the MSSP to estimate customer retention; both potentially highly sensitive metrics. To make this kind of analysis more difficult, we recommend periodically changing the pseudonym mappings. This does not impact detection of attacks we outlined above, as all of the detection schemes focus on identifying repeated instances of an IP address within a short timespan.

Pseudonymized internal addresses can similarly reveal the size of the internal network. In addition, prefix-preserving maps can reveal some information about the internal network structure. Since prefix-preserving mappings are only useful to detect a small subset of scanning patterns, if this information is sensitive to organization, simple pseudonyms may be a better choice. And once again, periodic rotation of pseudonyms can be used to hide more information.

Finally, the internal port numbers can reveal what services are running within an organization. Port numbers can also be eliminated from the logs if this information is sensitive, though we believe in many cases it is not.

To summarize, the log anonymization strategy we develop removes most, but not all sensitive information from logs. We believe that for many organizations, this strategy will be sufficient to allow them to trust their anonymized logs to an MSSP. In other cases, a more restrictive policy can be used, based on the privacy needs of a particular organization, at the expense of sacrificing some analysis capability at the MSSP.

TABLE II
ANONYMIZATION CONSTRAINTS ON META-DATA LOGS

Attr. List	Anonymization Constraints	Recommended Anonymization
Time Stamp	Retain events interval and time dependence	Random Time Shifts
Source IP Addr.	Anonymized and Re-identifiable	Pseudonyms *e.g.* Shamir's threshold
Source Port	- More efficient for MSSP if retained	Pseudonyms
Dest. IP Addr.	- Retain virtual network topology - Re-identifiable anonymized	Pseudonyms + Prefix-preserving
Dest. Port	- More efficient for MSSP if retained - Re-identifiable if anonymized	Pseudonyms
Msg.	Retained	–

B. Future Work

Our results demonstrate that outsourcing security monitoring MSSP by using log anonymization is a promising approach. However, looking ahead, there are several important questions that need to be answered. The first question is whether our anonymization strategy is compatible with other types of attack analysis. A good strategy for answering this question may be to examine common rules used in intrusion detection systems, as well as the techniques used by security administrators to detect and analyze attacks. Our belief is that a with a thorough analysis, it will be possible to not only demonstrate the anonymization strategies appropriate for a wide range of specific attack analysis approaches, but also derive patterns of attack detection and make a general statement about what information is necessary for such analysis.

Secondly, NetFlows and traffic traces are only a subset of all the logs used for security monitoring. Logs recording host-level events, such as process accounting and system logs, as well as application-specific logs, are also invaluable for attack detection. These logs have different fields than NetFlows, and different anonymization strategies will need to be explored. Often, analysis involves correlations between several types of logs, so the anonymization of all logs must be performed in such a way that correlation is possible. (At the minimum, time field translation must be performed in the same fashion for all logs.)

Correlation of different logs is also potentially useful across different organizations. An MSSP can exploit its vantage point of observing logs from multiple organizations to better detect attacks. However, anonymization techniques make such detection much more complicated, since, for example, a pseudonym used for a source address will be different among different organizations. The feasibility of coordinated but privacy-preserving attack detection, therefore, remains an important open question.

V. RELATED WORK

While there exists a large body of economics literature on outsourcing as applied to many industries, there has been little work specific to outsourcing Internet security. While in this paper we focus on MSSP outsourcing for security monitoring, the MSSP market also provides other services such as vulnerability assessment, compliance auditing, threat intelligence, and individual protection services (network, Email). For a comprehensive survey of the MSSP market see [6]. [10] provides an overview of the evolution of the Storage Service Provider (SSP) market as well as two illustrative case studies. We consider the SSP market a subset of the larger MSSP market since storage security is the primary motivating factor in its development.

The use of anonymization as privacy-enhancing technique for facilitating the sharing of data for security monitoring is a growing area of study. Motivations for sharing data for security purposes are summarized in [22]. While there is a consensus for the exchange of logs as the data sharing medium, there are on the order of 20 commonly implemented network system logs so selecting which logs to share is an important question citesiam03,ictsm03.

Pseudonyms are an important building block to later log anonymization techniques. In 1981, Chaum proposed using public keys as pseudonyms [4]. It was not until 1997 when Sobirey et al. first suggested using pseudonyms for privacy-enhanced intrusion detection of syslogs [23]. From 2000-2002, Biskup and Flegel extended this work in a series of papers [2], [3], [9].

Recently, anonymization techniques using strong encryption has supplanted work on pseudonyms. Even though the use of strong encryption does not prevent re-identification from determined attackers, it is commonly felt that anonymization may raise the bar to attackers high enough that organizations will feel more confident sharing logs. At the USENIX Security Symposium in 2004, SRI researchers proposed a repository to which sensors would send anonymized alerts which are then analyzed and publicly announced [14]. While there are potential problems with their proposed encryption schemes noted in [22], more importantly the level of coordination across the Internet for this type of scheme is likely impractical as well as any public repository being an open target for attackers to evade, subvert, or disable. At the USENIX Security Symposium in 2005, two papers were presented on attacker detection and subversion of public repositories of alert information. In [1], [21], the authors demonstrate (with simulation and case studies respectively) that although

sensor locations of public Internet threat monitors may be concealed, they can be revealed with probe-response attacks. While [1], [21] propose the use of access control, query limiting, throttling information, sampling, time delay, and added noise (and other schemes) to prevent sensor locations from being discovered, they also note these schemes are all easily broken. For the purposes of related work, we note that that the authors in [1], [21] explicitly omit the potential use of anonymization which we believe can be used to accomplish the desired goal to a satisfactory, although not perfect, level of privacy protection while still enabling significant security analysis.

The latest related work is from Pang et al., who present a packet trace anonymization case study. While this work is an example of the difficulty of mapping a security policy to a packet trace anonymization implementation, the anonymization techniques applied to selected fields in the packet trace are simplistic[5] and the fundamental tradeoff between privacy protection (via anonymization) and the use of the resultant packet traces for security analysis is not discussed.

For practical implementations beyond research, several log anonymization tools are currently available on the Internet for download including:

- CANINE: NetFlow anonymization [13]
- TCPdpriv: prefix-preserving IP address anonymization [17]
- Crypto-PAn: prefix-preserving IP address anonymization [8]
- ip2anonip: filter with IP address anonymization based on TCPdpriv [19]
- ipsumdump: ascii conversion of tcpdump with IP address anonymization based on TCPdpriv [12]
- Scrub-PA: process accounting anonymization [16]

Lastly, the SANS Internet Storm Center (ISC) [20] and DShield [7] currently provide public Internet security monitoring based on gathering data (logs and alerts) and then sharing information larger Internet community in summarized form. For instance, both ISC and DShield provide geographic "weather" maps of Internet security events and the ISC has a "Handler Diary" prepared by one of its volunteers summarizing major events. ISC and DShield are typically one of the first public sources of new attacks. MSSPs provide similar security monitoring services for a fee, for example Symantec's DeepSight [5] and ISS's Managed Security Services Virtual-Security Operations Center (Virtual-SOC) [11].

VI. Conclusions

We propose a new architecture for outsourced security monitoring, where anonymized logs are sent to a monitoring provider in order to protect the security and privacy of an organization. It can remove an important barrier to the use of managed security services providers (MSSPs). We explored several case studies of logs and attack analysis and developed anonymization strategies for those logs that leave sufficient information for security analysis while protecting other, sensitive information.

Our studies show that our proposed architecture is a promising approach for future security monitoring. We therefore define three areas of future research, exploring different types of attacks and analysis, different kinds of logs, and correlating logs across multiple organizations. We believe that such research will help deployment of outsourced MSSPs become both more widespread and more effective.

[5]in [18] each field is either anonymized or not anonymized; this is in contrast to the more flexible multi-level anonymization schemes implemented in existing tools [13], [16]

References

[1] J. Bethencourt, J. Franklin, and M. Vernon. Mapping Internet sensors with probe response attacks. In *USENIX Security Symposium*, 2005.

[2] J. Biskup and U. Flegel. Threshold-based identity recovery for privacy enhanced applications. In *7th ACM Conference on Computer and Communications Security (CCS)*, 2000.

[3] J. Biskup and U. Flegel. Transaction-based pseudonyms of audit data for intrusion detection. In *Recent Advances in Intrusion Detection (RAID)*, 2000.

[4] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[5] Symantec Corporation. DeepSight. `http://www.symantec.com`.

[6] W. Ding, W. Yurcik, and X. Yin. Outsourcing internet security: Economic analysis of incentives for managed security service providers. In *Workshop on Internet and Network Economics (WINE)*, 2005.

[7] DShield.org — Distributed intrusion detection system. `http://www.dshield.org/`.

[8] J. Fan, J. Xu, M.H. Ammar, and S.B. Moon. Prefix-preserving IP address anonymization. *Computer Networks*, 46(2):253–272, 2004.

[9] U. Flegel. Pseudonymizing Unix log files. In *Infrastructure Security Conference (InfraSec)*, 2002.

[10] R. Hasan, W. Yurcik, and S. Myagmar. The evolution of storage service providers: Techniques and challenges to outsourcing storage. In *1st International Workshop on Storage Security and Survivability (StorageSS)*, 2005.

[11] Internet Security Systems Inc. Virtual-security operations center. `http://www.iss.net/`.

[12] E. Kohler. ipsumdump. `http://www.cs.ucla.edu/~kohler/ipsumdump/`.

[13] Y. Li, A. Slagell, K. Luo, and W. Yurcik. CANINE: A combined conversion and anonymization tool for processing NetFlows for security. In *International Conference on Telecommunication Systems Modeling and Analysis*, 2005.

[14] P. Lincoln, P. Porras, and V. Shmatikov. Privacy-preserving sharing and correlation of security alerts. In *USENIX Security Symposium*, 2004.

[15] E. Lundin and E. Jonnson. Privacy vs. intrusion detection analysis. In *Recent Advances in Intrusion Detection*, 1999.

[16] K. Luo, Y. Li, C. Ermopoulos, W. Yurcik, and A. Slagell. SCRUB-PA: A multi-level multi-dimensional anonymization tool for process accounting. Technical Report cs.CR/0601079, ACM Computing Research Repository (CoRR), 2006.

[17] G. Minshall. TCPdpriv. `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`.

[18] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *Computer Communication Review*, 36(1):29–36, 2006.

[19] D. Plonka. p2anonip. `http://net.doit.wisc.edu/~plonka/ip2anonip/`.

[20] SANS. Internet storm center (ISC). `http://isc.sans.org/`.

[21] Y. Shinoda, K. Ikai, and M. Itoh. Vulnerabilities of passive Internet threat monitors. In *USENIX Security Symposium*, 2005.

[22] A. Slagell and W. Yurcik. Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization. In *IEEE/CREATENET SecureComm*, 2005.

[23] M. Sobirey, S. Fischer-Hubner, and K. Rannenburg. Pseudonymous audit for privacy enhanced intrusion detection. In *IFIP TC11 13th International Conference on Information Security (SEC)*, 1997.

[24] W. Yurcik and Y. Li. Internet security visualization case study: Instrumenting a network for NetFlow security visualization tools. In *21st Annual Computer Security Applications Conference (ACSAC)*, 2005.