On Attack Correlation and the Benefits of Sharing IDS Data

by

Sachin Katti

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[September 2005]
August 2005

Author .....
Department of Electrical Engineering and Computer Science
19 August, 2005

Certified by .....
Dina Katabi
Assistant Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

On Attack Correlation and the Benefits of Sharing IDS Data

by

Sachin Katti

Submitted to the Department of Electrical Engineering and Computer Science
on 19 August, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

This thesis presents the first wide-scale study of correlated attacks, i.e., attacks mounted by the same source IP against different networks. Using a large dataset from 1700 intrusion detection systems (IDSs), this thesis shows that correlated attacks are prevalent in the current Internet; 20% of all offending sources mount correlated attacks and they account for more than 40% of all the IDS alerts in our logs. Correlated attacks appear at different networks within a few minutes of each other, indicating the difficulty of warding off these attacks by occasional offline exchange of lists of malicious IP addresses. Furthermore, correlated attacks are highly targeted. The 1700 IDSs can be divided into small groups with 4-6 members that do not change with time; IDSs in the same group experience a large number of correlated attacks, while IDSs in different groups see almost no correlated attacks. These results have important implications on collaborative intrusion detection of common attackers. They show that collaborating IDSs need to exchange alert information in realtime. Further, exchanging alerts among the few fixed IDSs in the same correlation group achieves almost the same benefits as collaborating with all IDSs, while dramatically reducing the overhead.

Thesis Supervisor: Dina Katabi
Title: Assistant Professor of Computer Science and Engineering

Acknowledgments

This thesis is the result of joint work done with Dina Katabi and Balachander Krishnamurthy.

Contents

Chapter 1

Introduction

Many attacks on the Internet are mounted from botnets. Every day about 30,000 new machines are compromised [30]. Many of these machines are rented by the hour on IRC channels [15], with the result that the same machines are involved in multiple attacks against different networks [30]. Hence, it should be beneficial for the victim networks to collaborate to exchange information about attacks and attackers. The intrusion detection system (IDS) or the firewall at each network can learn about recent alerts and offending IPs from other IDSs. Future packets from suspicious source IPs can then be flagged to be dropped or scrutinized.

This thesis studies **correlated attacks**. Two attacks are said to be correlated if they are mounted by the same source IP against different networks. Exchanging information about attacks and attackers is most effective when it happens between networks with highly correlated attacks. Although several prior research efforts have looked at collaborative intrusion detection [33, 34, 25, 10, 31, 26], none has studied the prevalence of correlated attacks in the current Internet and its implications on how to pick collaborators.

This thesis addresses two questions:

- *How prevalent is attack correlation in the current Internet?* Although collaboration to detect common attackers seems plausible, there is no quantification of the potential benefits. Measurements of the frequency with which different networks become victims of a common attacker, the types of shared attacks, and the quantity of resulting IDS alerts are important to gauge whether collaboration is worth the effort.

- *How can an IDS pick trusted and effective collaborators?* Allowing IDSs to exchange

alerts to collaborate against common attackers requires addressing two issues: overhead and trust. Exchanging alert data with thousands of IDSs in realtime is a daunting task. Processing this data in a timely manner to detect and protect against in-progress attacks is highly resource intensive, and can easily overwhelm any IDS/firewall. Thus, an IDS needs to pick its collaborators intelligently to minimize the overhead and maximize the utility of the collaboration. Furthermore, two networks need to establish trust before they can exchange IDS data. Otherwise, a network cannot ensure the information it receives is correct and not maliciously manipulated to make certain IP addresses look as attackers. Also, it cannot ensure that the information it provides will not leak internal vulnerabilities to malicious entities.

This thesis analyzes logs from 1700 IDS/firewalls deployed in US and Europe. The data is rich; in addition to sanitized logs from DSHIELD [2] and multiple universities, it contains *detailed* attack logs from 40 IDSs maintained by a Tier-1 provider to protect its customer networks. The logs cover 1-3 months, and a big chunk of the IP address space including the class A address space of a large provider and many class B and C networks.

1.1 Contributions

This thesis constitutes the first large scale empirical investigation of attack correlation in the Internet. It results in 4 major findings.

(a) The extent of attack correlation: Correlated attacks are prevalent in the Internet. More than 20% of the offending IP sources attack multiple networks–i.e., generate alerts at multiple IDSs–and these common attackers are responsible for 40% of the total alerts in the dataset. Correlated attacks exploit vulnerabilities in widely deployed services like RCP, SMTP, Windows NT servers, IIS servers, IBM Tivoli etc. Further, shared attackers attack different networks within a few minutes of each other, emphasizing the advantage of realtime IDS collaboration, as opposed to sharing attack logs offline.

(b) Reducing collaboration overhead by exploiting correlation structure: This thesis analyzes the spatial structure of attack correlation. It shows that the 1700 IDSs in the dataset can be divided into small groups of 4-6 members (about 0.4% of the IDSs in the dataset); IDSs in the same correlation group experience highly correlated attacks, whereas IDSs in different groups see uncorrelated attacks. By collaborating with only IDSs in its

correlation group, an IDS achieves the same utility obtained form collaborating with all IDSs, while dramatically reducing the collaboration overhead.

These small correlation groups seem to arise from recent attack trends. In particular, victim sites in the same group may be on a single hit list, or might be natural targets of a particular exploit. For example, the Santy worm uses a vulnerability in popular phpBB discussion forum software to spread and uses Google to find vulnerable servers [6]. Such targeted attacks are far from random and likely to create small correlation groups of sites that run particular software or provide a particular service. We have examined the correlated attacks in each group for cases where full attack details are available. Indeed, each group seems to be characterized by a specific attack type, i.e., there are SMTP groups, NT groups, IIS groups, etc.

(c) Scalable Trust Establishment: Measurements also reveal that correlation groups are fairly stable and their membership persists for the duration of the dataset (1-3 months). Thus, each network needs to collaborate with only 4-6 *fixed* networks in its correlation group. The small number of IDSs in a correlation group and the persistence of their identities allow a network to check their credibility offline and establish trust using an out-of-band mechanism such as legal contracts or reputation.

A network still needs to learn who is in its correlation group. This service can be provided by a few trusted nonprofit organizations, like CERT [1] and DSHIELD [2], or commercial entities. They receive sanitized alert data, (containing only time and offending source IP), from participating networks, analyze it for attack correlation, and inform the participating networks about others in their correlation group. The process is scalable because correlation groups are persistent for long intervals (months) and do not need frequent update. DSHIELD already has the means to provide this service to its participant networks.

(d) The importance of picking the right collaborators: This thesis provides rough estimates of the overhead and detection capability obtained via different choices of collaborating IDSs. The following schemes are compared: (1) local detection with no collaboration; (2) collaboration with all IDSs in the dataset; (3) correlation-based collaboration (CBC), where each IDS collaborates with only IDSs in its correlation group; (4) random collaborators, where an IDS collaborates with the same number of IDSs in its correlation group but picks the identity of its collaborators randomly.

Term	Definition
Correlated Attacks	Two attacks are correlated if they are mounted by the same source IP.
Alert	An alarm raised by a sensor when it encounters a suspicious event, e.g. a packet or set of packets that contain a known exploit.
Correlated IDSs	Two IDSs are said to be correlated if more than 10% of their attacks are correlated.
Correlation group of IDSs	A set of IDSs whose attacks are highly correlated.
Correlation Vector of IDS i	is $\vec{v_i} = (v_{i1}, ..., v_{ij}, ..)$, where $v_{ij} = 1$ if $j \in$ correlation group of i, and otherwise $v_{ij} = 0$.
Blacklist	A list of suspicious IP addresses whose packets are dropped or given unfavorable treatment.

Table 1.1: Definitions of terms used in the thesis

The above schemes are compared using a trace driven simulation, where attacks are rerun as they appear in the logs. The focus is on collaboration to quickly blacklist malicious IP sources. Each IDS maintains two thresholds: a Blacklisting Threshold, B, and a Querying Threshold, Q, where $B > Q$. When the alert rate of a source crosses Q, the IDS queries its collaborators about the source. If the source aggregate alert rate at all collaborators exceeds B, all collaborators blacklist the source. This collaboration scheme may not be optimal, but it is enough to provide *rough* estimates of the benefits and overhead of the considered schemes. The entire dataset is divided into two halves, the first half is used as a training dataset to create the correlation groups in CBC and the second half is used as a test dataset for evaluating the collaboration schemes.

The results of the evaluation emphasize the importance of picking the right collaborators. Mainly:

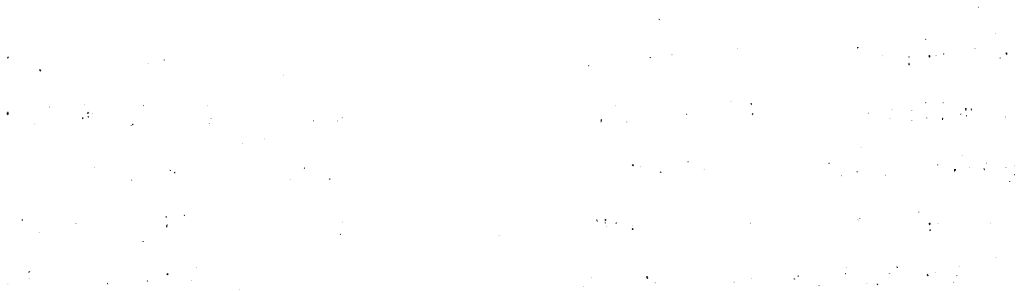
- Correlation-based collaboration has almost as good detection capability as collaborating with all IDS, but generates less than 0.003 of the traffic overhead. It detects 95% of the attackers detected by collaborating with all IDSs and reduces alert volumes by nearly the same amount.

- In comparison with local detection, correlation-based collaboration increases the number of detected common attackers at an IDS by 30% and speeds up blacklisting for about 75% of the common attackers. As a result of the blacklisting, correlation-based collaboration reduces the size of the log that the administrator has to examine by an

additional 38%.

- Replacing the IDSs in the correlation group by random collaborators reduces the detection capabilities dramatically and does not add much beyond local detection.

Table 1.1 defines the terms used in this thesis.



Chapter 2

Dataset and Method

This chapter describes the dataset used in the measurement study. It also discusses the reasons for the various definitions used in the thesis.

2.1 Dataset

The dataset consists of logs collected at 1700 different IDSs deployed in US and Europe. They can be divided into 3 distinct sets based on their origin: (1) 40 IDSs on different networks in a Tier-1 ISP; (2) DSHIELD Logs; (3) University logs. The logs cover periods of 1-3 months. They span a relatively large fraction of IP address space. In addition to a /8 ISP space, the DSHIELD data contain logs from many /16 and /24 networks. This data is at least as large and representative as those used in prior work [33, 20, 9, 29, 19, 21]. Further, it is the first studied dataset of its size that provides detailed alert information from deployed IDSs in the commercial Internet.

Table 2.1 provides a summary description of the dataset. A detailed description follows.

2.1.1 ISP Logs

This set consists of logs from 40 IDS deployed in a large ISP with a /8 address space. The IDS boxes protect different customer networks and span a large geographic area, but they are all administered by the ISP and hence have identical characteristics and signature sets. The signature set is large and diverse consisting of over 500 different alerts. The logs contain full unanonymized packet headers for all suspicious packets, as shown in Fig. 2-1a. Hence unlike the DSHIELD data described below, they provide access to the offending packet as

a) ISP Dataset log record

Time	Direction	Source IP	Destination IP	Alert Type	Attack information	Sensor ID
10:00:07	[In]	164.120.83.253	10.0.0.1	RPC:PROTOCOL-EVADE	(tcp,dp=32789,sp=20)	(ABCDEF)

b) DSHIELD log record

Date	Time	Provider Hash	Alert Count	Source IP	Source port	Destination IP	Destination port	TCP Flags
2004-12-20	10:00:07	12345678	10	164.120.83.253	20	*.0.0.1	32789	S

Figure 2-1: Log records for the ISP dataset and the DSHIELD dataset. The ISP dataset also has packet headers for each log record. The DSHIELD dataset has the destination IP anonymized.

	ISP dataset	DSHIELD	University datasets
# of IDSs	40	1657	3
Address space	Class A	5 Class B, 45 Class C and several smaller networks	2 Class B, 1 Class C
Period	July 1 - August 30, 2004 Dec. 15, 2004 - Jan. 15, 2005	Dec. 15, 2004 - Jan. 15, 2005	Dec. 15, 2004 - Jan. 15, 2005
Richness	Detailed alerts, unanonymized	Dest. IP addresses anonymized	Detailed alerts, unanonymized
Avg #alerts/day/IDS	40000	15000	30000

Table 2.1: Description of the 3 datasets

well as the nature of the offense. The logs cover two separate periods: one period from December 15, 2004 to January 15, 2005 and the other from July 1 to August 30, 2004. The data exhibits a large amount of variation in the kind of attacks seen (over 100 different attack types) as well as the distribution of attacking IP addresses (over 100000 unique source addresses) and 40000 alerts/day/IDS.

2.1.2 DSHIELD Logs

DSHIELD is a global repository set up as a research initiative as part of the SANS institute. Participating organizations provide IDS/firewall logs, which DSHIELD uses for detection and analysis of new vulnerabilities, and blacklist generation. Since the IDS systems which participate in DSHIELD employ widely varying software, DSHIELD uses a minimal record format for its logs and scrubs the high order 8 bits of the destination IP address, as shown in Fig. 2-1b. The entities participating in DSHIELD vary in size from several Class B networks to smaller Class C networks and are distributed throughout the globe [33, 2]. The

logs are of substantial size with nearly 15000 alerts/day/IDS. We have collected DSHIELD logs from 1657 IDS for the period from Dec. 15, 2004 to Jan. 15, 2005 corresponding to the ISP dataset.

2.1.3 University Logs

The last set of logs is from IDS/firewall systems deployed at 3 universities U1, U2 and U3. Of these we have access to raw data complete with packet headers and nature of offense detected in U1. The second university U2 provided logs from the Bro IDS [22] installed on their network, but with protected addresses anonymized. The signature set deployed is different and the alerts consist mostly of scans of IP addresses as well as port-scans. The third university U3 provided firewall logs which consisted of blocked connection attempts. The University logs generate 30000 alerts/day/IDS on the average.

A few limitations are worth mentioning. Except for the ISP logs, the other IDSs in the logs are largely independent. Their configurations, the signature sets used, or even the platforms they operate on, are unknown. This means that some of the attack correlation may be hidden because of differences between IDS signature sets. Second, there is no information about the nature or the business of the protected networks, and thus it cannot be ascertained whether these issues play a role in attack correlation.

2.2 Method

This section describes some of the pre-processing that was done to clean the data from obvious false positives. The data was then analyzed to find a meaningful definition of the term "attack correlation".

2.2.1 Filtering

IDS logs are prone to flooding with alerts, many of which are innocuous alarms. For example, the ISP and University data sets contain innocuous alarms triggered by misconfigurations, P2P applications like eDonkey, malformed HTTP packets etc. Many of these were already flagged as false positives by the security administrators in the ISP dataset. Since these are not actual attacks, they do not help in detecting attack correlation among different sites. Hence such known false positives are filtered out from the ISP and univer-

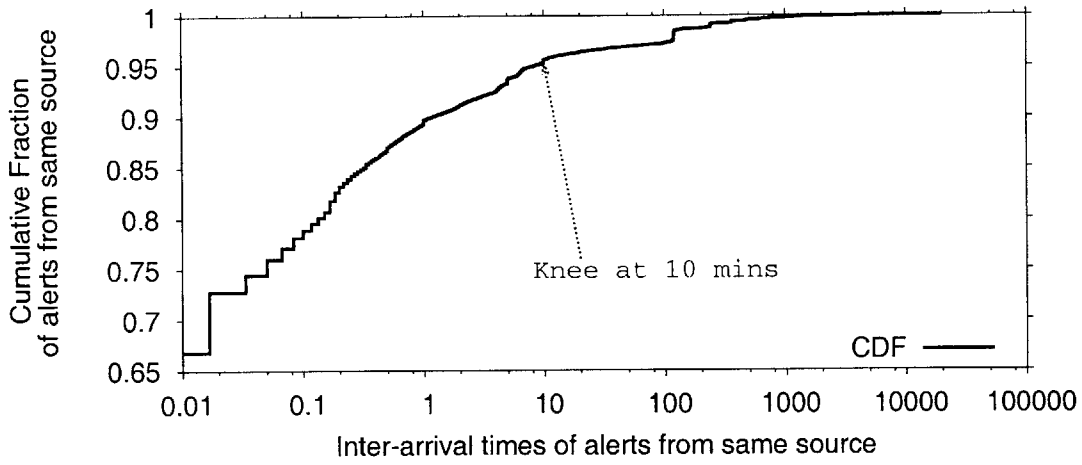


Figure 2-2: CDF of inter-arrival times of consecutive alerts from a source in minutes. The CDF is taken over the sources. 95% of consecutive alerts from a source arrive within 10 minutes of each other, the rest are separated by several hours.

sities logs. All the remaining alerts are considered to be parts of valid attacks. Of course this cannot be done for the DSHIELD dataset, since the nature of the alert is unknown.

2.2.2 Attack Durations

To carry out this study, attacks need to be extracted from IDS logs. A stream of suspicious packets from the same source to an IDS with an inter-arrival smaller than 10 minutes is considered as an attack. The following paragraphs explain why a separation window of 10 minutes is reasonable.

To find a meaningful separation window, a CDF of inter-arrival times of consecutive alerts from the same source at an IDS is plotted in Figure 2-2. The CDF shows that 90% of the alerts from a source arrive within a minute of each other, these are likely to belong to the same attack event. The knee in the CDF happens at 10 minutes, inter-arrival times larger than 10 minutes are spread out to several hours. Hence 10 minutes is picked as the window because about 95% of the alerts from the same source arrive separated by less than 10 minutes and the other 5% have widely-spread interarrivals.

2.2.3 Defining Attack Correlation

How should one define attack correlation? Should all fields in the IDS alerts received at different IDSs be the same, or is it enough to consider one or two fields? Furthermore, how long can the interval between the two attacks at two different IDSs be for them to be still

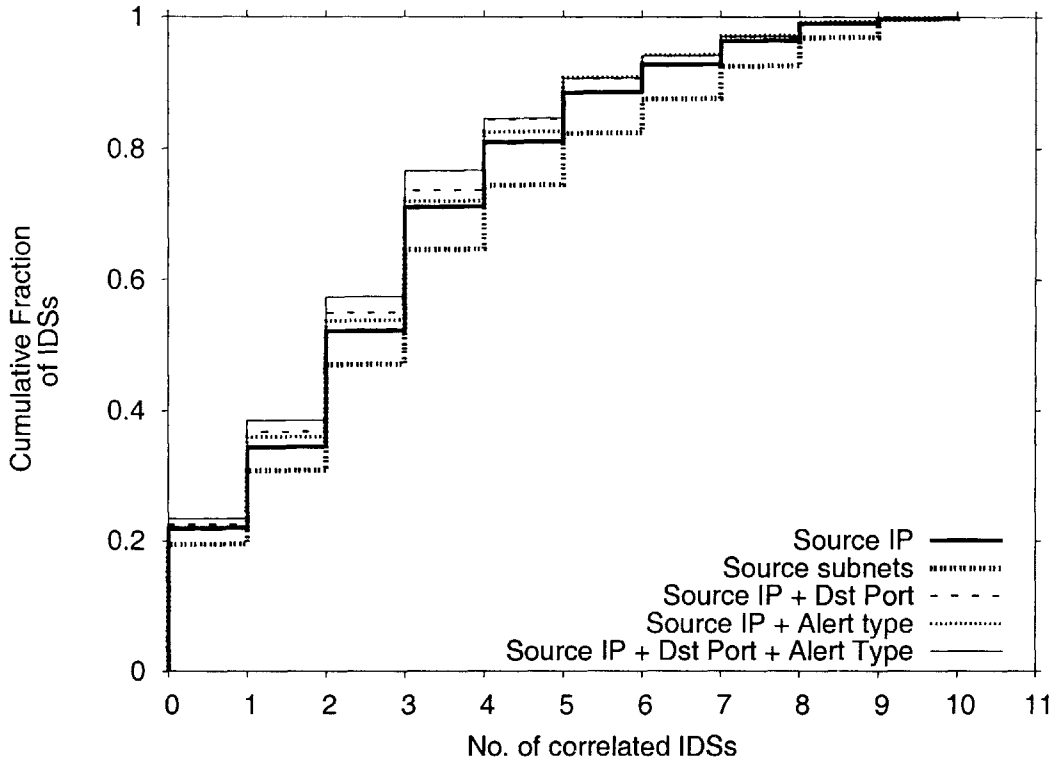


Figure 2-3: CDF of the size of the correlation groups for different definitions of attack correlation for the ISP and U1 datasets. The CDFs are taken over the IDSs. They show that the correlation is insensitive to the additional information obtained from the alert type and port, and can be discovered based solely on source IP.

considered correlated?

Attack correlation can be parameterized by the set of correlated header fields and the time window used to compute the correlation. *Two attacks are said to be correlated if they share the source IP address, and attack correlation is computed over windows of 10 minutes.* Both choices are based on detailed analysis of the data that showed almost no sensitivity to including additional fields in the correlation beyond the source IP and using time windows larger than 10 minutes. This analysis is described in detail below.

Picking the correlation fields

Defining attack correlation based on the destination IP address is not useful since attacks seen by a particular IDS will have their destinations in the local network. Also the source port is likely to be picked randomly and is not useful for defining attack correlation.

The following definitions of correlated attacks are considered: 1) source based, 2) source subnet based, 3) source and the destination port combined, 4) source and alert type com-

bined, 5) and source, alert type, and destination port combined. This analysis is conducted for the ISP dataset and the U1 datasets, since these provide access to all the necessary fields.

Since the main interest is to find who is correlated with whom, the effect of different attack correlation definitions on the size of the correlation group of a IDS (see Table 1.1) is considered. Correlated groups are explained further in 3, but for the purposes of this analysis they are simply the set of other IDSs which experience correlated attacks.

Figure 2-3 plots the cumulative distribution functions (CDFs) of the size of the correlation set, for various correlation fields. The figure shows that, except for the CDF for source subnets, all the other CDFs are very close together. Classification based on the attacking source subnet results in slightly higher correlation, but the difference is not substantial. Further, classifying based on source subnet carries the danger of blacklisting an entire subnet resulting in innocent sources being blocked. Since including extra fields in the definition of correlation in addition to the source IP has an insignificant impact on the correlation CDF, we define attack correlation based solely on the correlation of the offending source IP address.

The above lead to an interesting result: performing attack correlation analysis requires minimal information, namely attack time and offending source IP.

Picking the maximum time window between correlated attacks

Unless stated differently, a 10 minute window is used for determining correlated attacks at different IDSs. Different time windows in the $[5, 30]$ minutes range were considered. Windows less than 10 minutes resulted in decreased attack correlation while there was not much difference for windows greater than 10. Hence the minimum window possible i.e. 10 minutes was picked. Thus if two attacks at two IDSs start within 10 minutes of each other, then they are considered correlated.

Correlation threshold

Two IDSs are correlated if more than 10% of their attacks are correlated. The threshold is justified below. The CDF of correlation taken over all IDSs with non-empty correlation groups (i.e., IDSs that are correlated with at least one other IDS) is plotted in Figure 2-4. For 90% of the IDS, the correlation (percentage of correlated attacks w.r.t all attacks) was

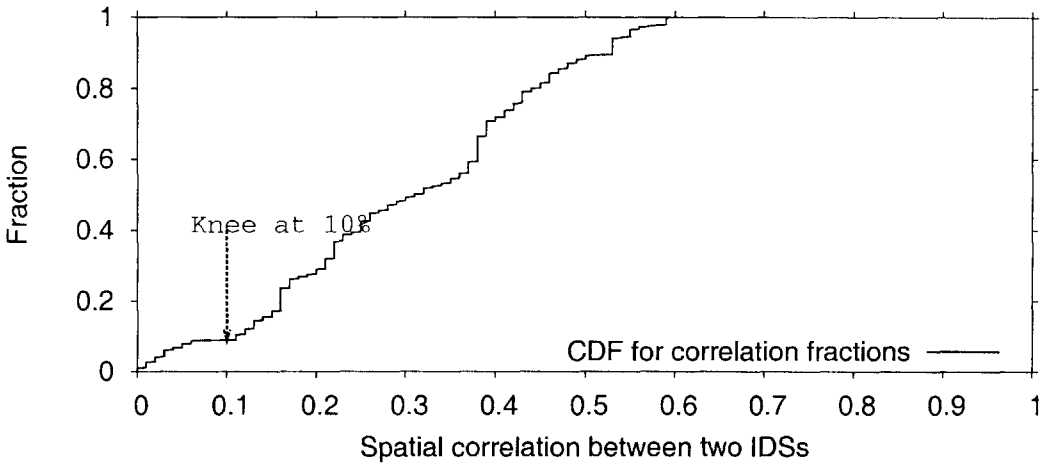


Figure 2-4: Cumulative distribution of spatial correlation exhibited by IDS for all 3 datasets.

higher than 10% ranging upto 57%. For the remaining 10% of the IDS, the correlation was slightly higher than 0%. Such small values are due to a few attacks being shared and do not reflect any significant correlation between the two IDSs.

Chapter 3

Extent and Structure of Attack Correlation

This chapter examines whether correlated attacks happen and if so to what extent. Specifically it analyzes the percentage of attacks that are correlated, what is the average number of common victims for each attacker and the frequency of these attacks. It then studies the structure of attack correlation and investigates the reasons behind the structure.

3.1 Do IDSs see common attackers?

The average number of common and uncommon attacking IP addresses for each IDS per day is computed. A common attacker is an IP address that generates alerts at two or more IDSs. Figure 3-1 compares the CDF of common attacking source IPs with the uncommon/local ones. The CDF is taken over all IDSs. The graphs show that on average an IDS sees 1500 shared offending IPs per day, and 6000 unshared offenders. Thus, about 25% of the suspicious source IP addresses observed at an IDS are also seen at some other IDS in the dataset. The averages are for each IDS per day, hence the total amount of correlated attacks is much larger for the entire logs. These common source IP addresses account for 40% of all alerts in the logs. *Thus, correlated attacks happen quite often and constitute a substantial fraction of all attacks.*

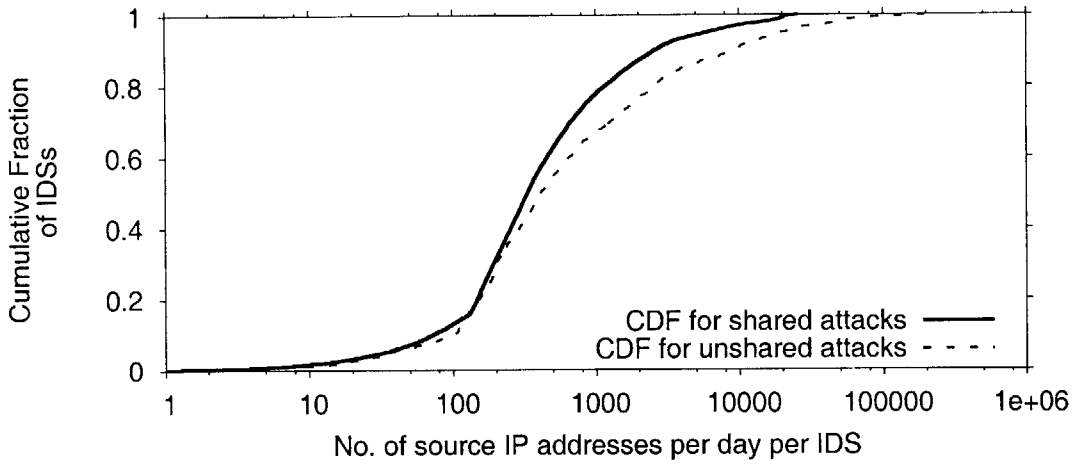


Figure 3-1: Prevalence of common attackers. Figure shows the CDFs of the average number of common attackers and local attackers per day per IDS, A common attacker is a source IP that is flagged as suspicious at two or more IDSs. 90% of the studied IDSs see more than 100 common attacking IPs per day. The average number of common attacking IPs at an IDS is about 1,500 while the maximum can be as large as 25,000.

3.2 How Many Victims does a Common Attacker Attack?

The previous section quantified how many source IP addresses at each IDS are common attackers, here the focus is on the number of victims of a common attacker. Figure 3-2 plots the CDF of the number of IDSs targeted by a common attacker. The CDF is taken over all common attacker IPs. On the average a common attacker appears at 10 IDSs–about 0.006% of all IDSs in the dataset. The high average of 10 victims seems to comply with recent trends in using botnets to mount multiple attacks against many target networks [30].

3.3 Time Between Correlated Attacks

How long does it take a common attacker before he attacks the next network? If this time is long then the exchange of alert data can be offline, but if it is short then effective collaboration against common attackers requires realtime exchange of information. The interarrival times of attacks from the same source at multiple IDSs, i.e., the difference between when the first time the attacker is observed at different IDSs is computed. Figure 3-3 shows the CDF of these interarrival times. More than 75% of the time, a common attacker attacks the next IDS within 10 minutes from the previous IDS. Attackers therefore mount multiple attacks within a span of a few minutes, suggesting that collaborative detection of such attackers has to be in realtime.

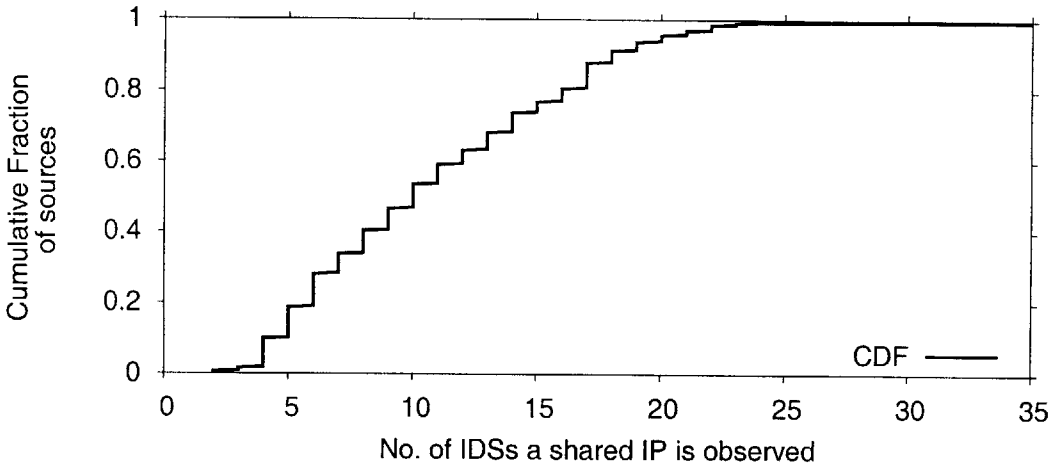


Figure 3-2: Figure shows the CDF of the number of different IDSs targeted by a common attacker. Common sources are detected at 10 different IDSs on the average, implying that such sources are employed to mount a large number of attacks at different victims.

3.4 Attack Correlation Structure

Why is the structure of attack correlation important? Since correlation is prevalent, it would be beneficial for IDSs to collaborate to speedup the detection of common attackers. However 3.3 showed that common attackers attack their victim networks within a few minutes of each other. Thus, to effectively collaborate against common attackers, the IDSs need to exchange information in realtime. on the average, an IDS generates 1500 alerts/hour in the dataset examined. Exchanging alerts at this rate in realtime with thousands of IDSs creates an unacceptable overhead. This section examines how many collaborators each IDS needs to have in order to achieve the benefits of collaboration without incurring much overhead. To answer this question, this section analyzes the spatial and temporal structures of attack correlation, i.e., how many IDSs are usually correlated with each other and how often does the set of IDSs a particular IDS is correlated with change over time?

3.4.1 Correlated IDSs

For the objective of detecting common attackers, an IDS benefits from exchanging alerts with only those IDSs whose attacks are correlated with its own. This set of IDSs is called its correlation group. If correlation groups are small, i.e., much smaller than all IDSs, then by focusing only on the IDSs in its correlation group, an IDS can achieve most of the benefits of the collaboration at little overhead.

Figure 3-4 plots the CDF of the number of IDSs with which an IDS is correlated (i.e.,

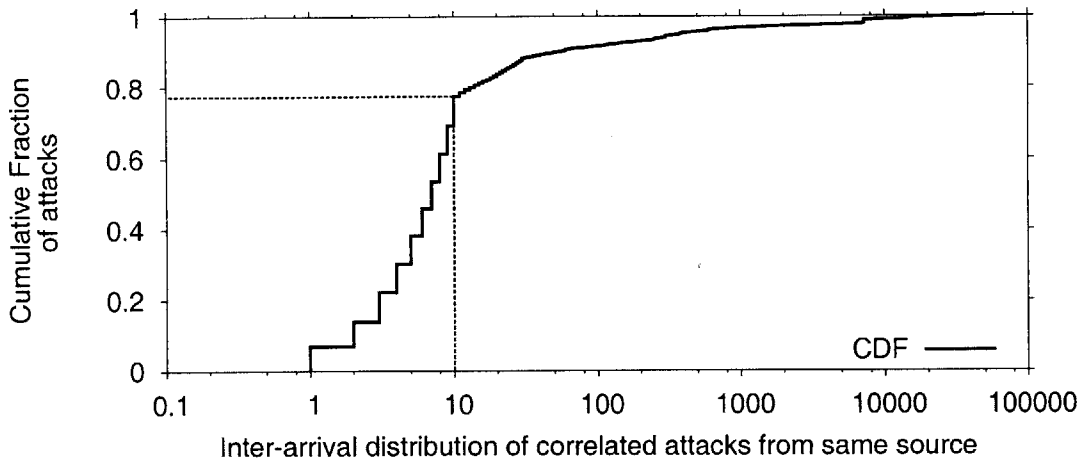


Figure 3-3: Figure shows the CDF of the interarrival times of correlated attacks at different IDSs. More than 75% of the correlated attacks arrive within 10 minutes of each other. This emphasizes the need for realtime exchange of attack data.

the size of its correlation group) for all 1700 IDSs in the dataset. Two cases are considered: simultaneous correlation, in which two attacks are correlated if they share the same source IP and happen within 10 minutes of each other, and general correlation, in which two attacks are correlated if they share the source IP. The former helps detect distributed attacks, while the latter helps detect malicious sources which should be blacklisted. General correlation is by definition greater than simultaneous correlation. The figure shows that on average each IDS is correlated with $4-6$ other IDSs, i.e., less than 0.4% of the total number of IDSs. Further, 96% of the IDSs are correlated with less than 10 IDSs.

Note that the plots for simultaneous and general correlation are fairly similar. Though the average number of IDSs with which an IDS shares attacks increases to nearly 5, the CDF does not change much. Again, this shows that when correlated attacks happen at different locations in the Internet, most likely they happen with a short period.

3.4.2 Persistence of IDS Correlation

The second question analyzed is how often does the correlation group of an IDS change? If the membership of the correlation group of an IDS is stable then each network can spend the time to identify its correlation group offline. Once the correlation group is identified, the actual exchange of alerts is done in realtime. On the other hand, if the members of an IDS' correlation group keep changing over short intervals, collaboration will be hard as it requires re-examining attack correlation and deciding in realtime whether to collaborate.

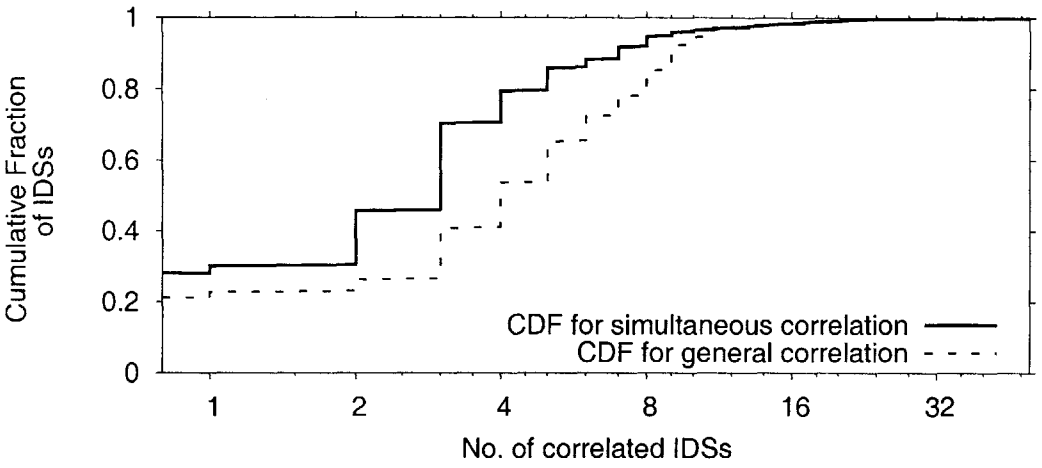


Figure 3-4: Cumulative Distribution of the number of IDSs with which any IDS exhibits correlation for all 3 datasets. Figure shows most IDSs are correlated with 4-6 (among 1700) IDSs with the average being slightly higher than 4.

A measure of how a set of IDSs is changing is needed in order to quantify shifts in membership of each correlation group. Each of the IDSs are assigned consecutive IDs. For each IDS i in the dataset, a correlation vector $\vec{v}_i(n)$ whose length is equal to the total number of IDSs in the dataset is created. The components are set as follows; $v_{ij}(n) = 1$ if IDS i is correlated with IDS j, and 0 otherwise based on the alerts they generate on day n. For example, $\vec{v}_i(16) = (0, 1, 1, 0, 1, 0, ..., 0)$ means that IDS i and IDSs 2,3, and 5 see correlated attacks on the 16th day in the dataset.

The difference vector for two days for a given IDS is the vector obtained by subtracting the corresponding correlation vectors for those days. For example, the difference $v_i(17) - v_i(0)$ indicates how the correlation group of IDS i changes over a period of 17 days, starting on day 0 in the logs.

The persistence of attack correlation as a function of time is then measured using the following metric:

$$f_{m-n} = \frac{1}{N} \sum_i \frac{||\vec{v}_i(m) - \vec{v}_i(n)||}{||\vec{v}_i(n)||}, \tag{3.1}$$

where $N = 1700$ is the number of IDSs; v_i is the correlation vector of IDS i; and $||\vec{v}||$ is the Euclidean norm of the vector. Thus, f_{m-n} is the average change in the norm of the correlation vector between day n and day m where $m > n$, normalized by the size of that vector.

Figure 3-5 plots the measure of the difference in attack correlation f_i as a function

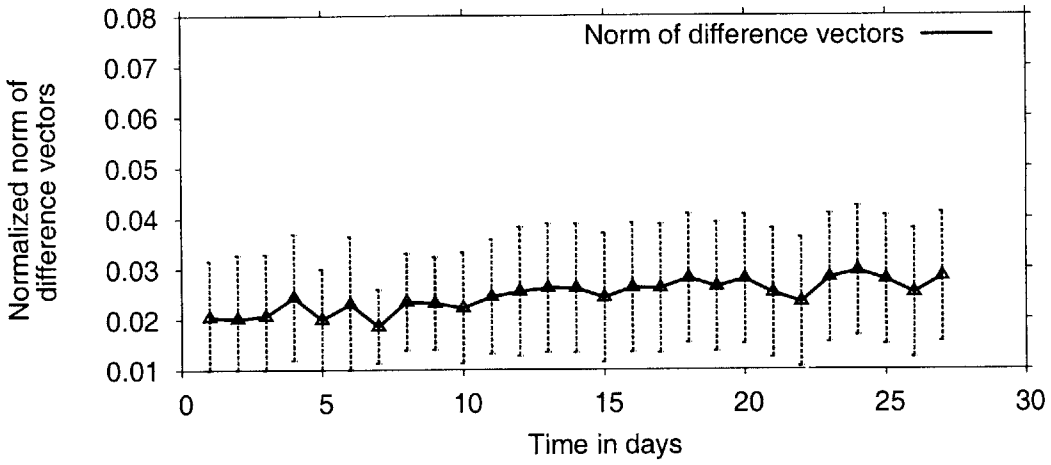


Figure 3-5: Figure shows that the group of IDSs which experiences attacks correlated with attacks at a particular IDS does not change for the duration of the 3 datasets (about a month). The y-axis is the normalized difference in the correlation vector defined in Equation 3.1.

of time in days along with the standard deviation. It shows that, the correlation vector does not change significantly with time. In particular, on average the correlation vector changes by less than 0.025 of its original value over a period that spans a whole month. The insignificant change shows that *correlation happens consistently with the same group of IDSs and is persistent over time.*

3.4.3 Robustness to Source Spoofing

The correlation shown above considers all attacks, including those which could be from spoofed source addresses. Intuitively, one would expect that source spoofing does not affect the correlation structure as it is usually done randomly, and thus unlikely to create a well-defined structure. In order to estimate the effect of spoofed sources on the results the logged attacks are divided into two classes:

- *Connection oriented attacks*: Attacks which require establishing a TCP connection. This includes most non-flooding attacks and application layer attacks (e.g SQL server, MS IIS server etc) and formed 68% of all attacks.

- *Connectionless attacks*: Attacks which get flagged due to incomplete TCP connection attempts or those which do not require a TCP connection. (e.g. SYN floods, UDP packet floods etc).

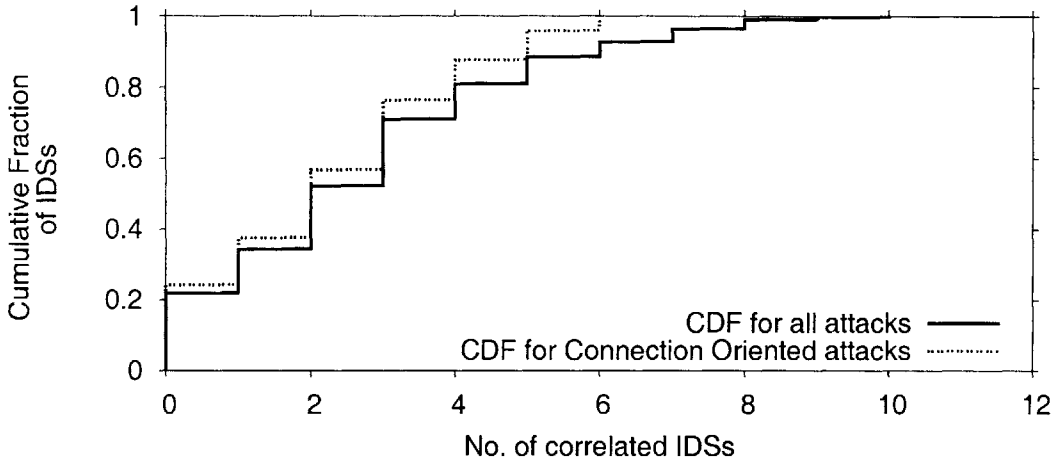


Figure 3-6: Comparison of attack correlation among connection-oriented attacks and all attacks for the ISP dataset. The figure plots the CDF of the number of IDSs that experience correlated attacks to a particular IDS. The two CDFs are very close indicating that the results are robust against source spoofing.

This classification is performed only on the ISP data and one of the University logs (U1). The rest of the logs do not contain the necessary information. Connection-oriented attacks should not have spoofed IP addresses since they require the attacking machine to respond to the TCP ACKs sent by the victim.

Figure 3-6 compares the correlation exhibited by the connection oriented attacks to that exhibited by the combination of all attacks. The figure plots the CDF of the size of the correlation group for each IDS for each kind of attack. *The figure shows that the two CDFs are very close, indicating that the correlation structure is highly robust to source spoofing.* Similarly, the correlation persistence test in 3.4.2 is performed on connection oriented attacks and found the results to be compatible with those in 3.4.2.

3.4.4 Is the Structure Due to Random Scans?

The fact that each IDS in the dataset shares attacks with only a small and persistent set of IDSs is intriguing. Why do certain IDSs share attacks? An additional test is performed to ensure that the spatial structure of attack correlation is not random. Suppose each worm or attacker picks for victims a random subset of all destinations, could this be responsible for generating the attack correlation structure seen in the data? The test described below shows that the answer to this question is "no". The correlated attacks seen are likely targeted attacks, i.e., the victims are not randomly chosen; the same group of correlated

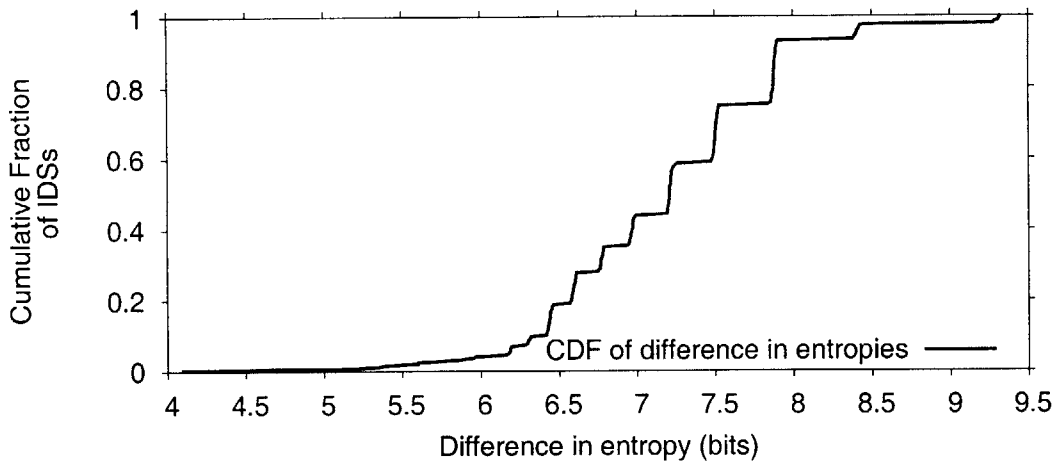


Figure 3-7: Figure shows that the set of IDSs with which an IDS is correlated is far from random. We compare the distribution of correlated IDSs in the dataset with that generated by having each common attacker target a small random group of IDSs. The difference in entropy between random targeting and the empirical data is plotted. The empirical distribution has, on average, has 7.2 bits less entropy than the one generated by random targeting; correlated IDSs are therefore far from random.

victim networks are chosen repeatedly as one group, probably because they are on one hit list circulating among the attackers, or because they run the same software (as in the case of the Santy worm [6]).

The test considers the distribution of IDSs with which a particular IDS is correlated. This distribution which is empirically obtained from the data is compared with the corresponding distribution generated by random targeting. Random targeting is simulated as follows. An IDS i is picked and all of its correlated attacks are examined. For each correlated attack, the set of IDSs with which IDS i shares this attack is replaced with a random set of IDSs of the same size. This process is repeated for each attack at IDS i. For each IDS j, where $j \neq i$, the number of correlated attacks with i, after proper normalization, represents the probability that IDS j is correlated with IDS i. This probability distribution generated by random targeting is compared with the one empirically generated from the data. The empirical distribution is found to be highly biased, i.e., an IDS i is correlated with a few other IDSs and uncorrelated with the rest of IDSs. Since the objective is to measure how far the empirical distribution is from random targeting, the entropies of the two distributions are compared. The entropy of the distribution of a random variable X is defined as:

$$H(X) = -\sum_{x_i} P(x_i) log(P(x_i)). \tag{3.2}$$

This analysis is repeated for each IDS and the difference in entropies are computed for each IDS. Figure 3-7 shows the CDF of these entropy differences. The figure shows that the set of IDSs with which an IDS is correlated is far from random. It shows that the empirical distribution has, on average, 7.2 bits less entropy than the one generated by random targeting. Note that number of IDSs in the system (dataset) is 1700, hence the maximum entropy is 10.73 bits. The difference in entropy is also bounded by the same value. Thus, an entropy difference of 7.2 bits is very high, which shows that the set of correlated IDSs in the data is far from random.

3.4.5 Origin of IDS Attack Correlation

So why two IDSs share correlated attacks? This section investigates two possible reasons: 1) closeness in the protected IP space, 2) similarity in the software and services run on the two sites. The results show that the latter is the likely reason of attack correlation between two IDSs.

Closeness in IP space

Some attackers employ scanning techniques to discover vulnerabilities. They start from a randomly selected IP and then scan sequentially. If the scanned address spaces belong to different sites, the IDS at the respective sites are likely to show attack correlation. Thus, closeness in the IP space could be a reason for attack correlation.

The distance between two prefixes P_1 and P_2 of equal length is computed as the decimal value of the bit-string produced by taking XOR of P_1 and P_2. If the prefixes are of unequal length, the shorter prefix is bit-shifted to the left to equalize the lengths. The distance in IP space between two IDSs i and j, D_{ij}, is defined as the IP distance between their protected address prefixes. Also for each IDS pair we generate the vector of correlation $\vec{C}_{ij}$, where c_{ij} is the percentage of attack at i which are correlated with some attacks at j. If proximity in the IP space is a reason for attack correlation, then the more the distance between IDSs i and j is, the less likely they share correlated attacks–i.e., $\vec{D}_{ij}$ and $\vec{C}_{ij}$ should be inversely correlated. Thus, the cross correlation between these two vectors is computed. The cross correlation is defined as:

$$r_{xy} = \frac{\sum_i (x(i) - \bar{x})(y(i) - \bar{y})}{\sqrt{\sum_i (x(i) - \bar{x})^2} \sqrt{\sum_i (y(i) - \bar{y})^2}} \tag{3.3}$$

where r_{xy} is the cross correlation, x and y are vectors of equal length, and $\bar{x}$ and $\bar{y}$ are the corresponding means. Note that a cross correlation around zero means independence.

Figure 3-8 plots the cross correlation between attack correlation and distance in IP space. The x-axis is the IDS id. The correlation with IP space hovers around zero, indicating that attack correlation is independent from the distance in IP space. Thus, having nearby IP prefixes does not have a visible impact on sharing correlated attacks.

Similarity in Software and Services

Small correlation groups may be due to recent attack trends. In particular, two IDSs may share correlated attacks because they are on a single hit list, or they run software or a service that is targeted by the common attacker. For example, the Santy worm uses a vulnerability in popular phpBB discussion forum software to spread and uses a search engine to find vulnerable servers [6].

Unfortunately, except for the university logs (U1), the identities of the protected networks, the type of software they run, or the services they provide are unknown; hence attack correlation cannot be compared with that information. Instead two indirect tests are performed.

First, the correlated attacks in each group are examined for the case of the ISP data where full attack details are available. Indeed, except for one group, each group seems to focus on a specific shared attack, i.e., more than 60% of the correlated alerts in that group are of a particular type. There are SMTP groups, NT groups, IIS groups, etc. This should not be surprising as recent attacks obtain a list of networks that run a software with the targeted vulnerability via a search engine or other ways and send only to those sites [6].

Second, the test indirectly infers the software and services run on the correlated networks by comparing the type of alerts they generate. The distribution of alert types generated by each network is generated and compared against each other. Alerts are divided into 13 broad categories: alerts due to attacks on DNS servers, web servers, ftp, RPC services, Windows Server 2003, servers running RPC, mail servers, servers using SQL (both MS and MySQL), telnet and ssh servers, attacks on routers, IRC servers, CIFS (SMB) servers and miscellaneous. The fraction of alerts of each type in the IDS log is computed. This distribution is considered to be characteristic of the network itself. The test now checks if attack correlation is correlated with correlation in this distribution.

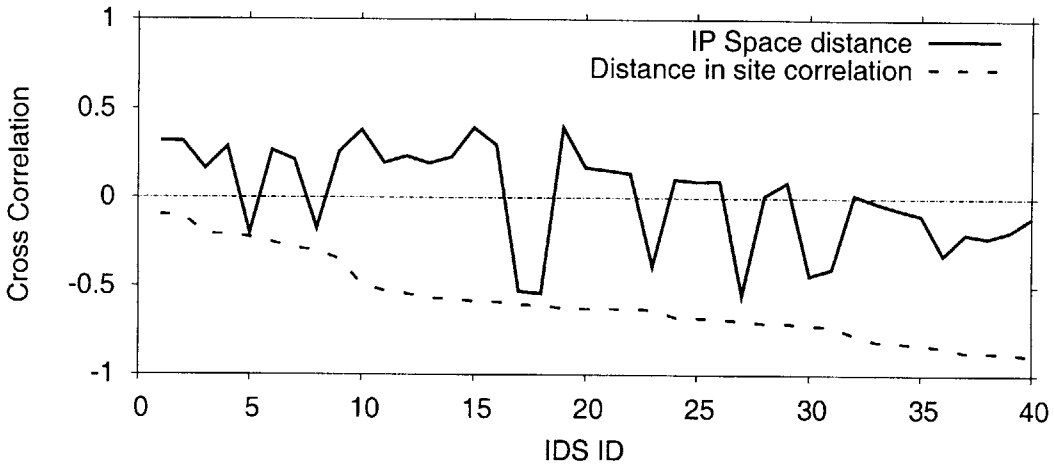


Figure 3-8: Cross correlation between attack correlation and: 1) distance in IP space, 2) an indirect measure of site's software and services. Figure shows that attack correlation is independent of closeness in IP space. In contrast, attack correlation seems to decrease with decreasing similarity between the software run on the protected networks. The figure is for the ISP and U1 datasets for which we have detailed alert logs.

The alert distribution is expressed in a vector $\vec{V}_i$ with 13 elements. For example, $\vec{V}_i = (0.03, 0.2, ...)$ means that 0.03% of the alerts generated by IDS i are of category 1, etc. The distance between the alert distributions at IDS i and j is quantified by the difference $\vec{D}_{ij} = ||\vec{V}_i - \vec{V}_j||$, where $||.||$ is the Euclidean norm. Similarly to the analysis in 3.4.5(a), $\vec{D}_{ij}$ is compared with $\vec{C}_{ij}$, where c_{ij} is the percentage of attack at i which are correlated with some attacks at j. If similar software and services are reasons for attack correlation, then $\vec{D}_{ij}$ and $\vec{C}_{ij}$ should be inversely correlated. The cross correlation of these two vectors is computed to understand how they are related. Note that a cross correlation around zero means independence, whereas a negative cross correlation means that an increase in the distance, $\vec{D}_{ij}$, is correlated with a decrease in attack correlation $\vec{C}_{ij}$. Figure 3-8 plots the cross correlation between attack correlation and the indirect measure of the similarity of the software and services on the protected networks. Attack correlation is negatively correlated with the measure of the distance between the software and services on the protected networks–i.e., an increase in this distance results in a decrease in correlation. Thus, it seems that one origin of attack correlation across different networks is the similarity in the software and services they run.

3.4.6 Summary of Empirical Results

The results of the study of attack correlation can be summarized as follows:

- Correlated attacks mounted by common attackers against multiple networks happen quite often. 20% of the unique sources in the dataset attack generate attacks at multiple IDSs, and common/correlated attacks account for an average of 40% of all attacks observed at an IDS.

- A network experiences attacks correlated with only a few other networks. On average an IDS shares attacks with 4-6 other IDSs which is just 0.4% of the total number of IDSs, and 96% of the IDSs share attacks with less than 10 other IDSs.

- Attack correlation persists over time–i.e., the sets of IDSs that experience correlated attacks did not change for the duration of the study (1-3 months).

- Though all origins of attack correlation are not known, the data shows that similarity in the software and services run on the protected networks plays an important role in making them endure correlated attacks.

- Common attackers tend to attack different networks within a few minutes of each other. Thus, there are considerable advantages for realtime sharing of alerts.

- Discovering the correlation group of an IDS (i.e., who shares with whom) requires minimal IDS-related information, namely attack time and offending source IPs.

- The study of the correlation for connection-oriented attacks shows that the correlation groups and their member IDSs are robust to IP spoofing.

The above results have important implications to collaborative intrusion detection of common attackers. First, they show that most of the benefits of sharing IDS alerts can be obtained form collaborating with a small set of IDSs that experience correlated attacks. Second, because attack correlation persists over time, the privacy of the exchanged IDS data can be protected using out-of-band mechanisms such as legal contracts. Third, since correlated attacks happen at different sites within a few minutes, there are considerable advantages for realtime sharing of alerts.

Chapter 4

Efficient Collaboration with Trusted Partners

This chapter discusses an architecture for an efficient collaborative intrusion detection system which exploits attack correlation structure. It then presents an evaluation of the architecture using a trace driven emulation and examines its efficacy, accuracy and overhead.

4.1 Correlation Based Collaboration

The major impediments to having independently administrated IDSs collaborate on detecting common attackers are: overhead and trust. Since common attackers attack different networks within a few minutes from each other, the IDSs need to exchange their alerts in realtime. But exchanging alerts with thousands of IDSs in realtime is impractical because of the resulting overhead, and the potential of having malicious IDSs incriminating innocent hosts or using the alert data to discover the vulnerabilities of other networks.

The structure of attack correlation can be exploited to solve the above two problems. This chapter proposes a correlation-based method for picking collaborators. By exchanging alert data with only those IDSs in its correlation group, an IDS minimizes the overhead of the collaboration while maximizing its chances of detecting common attackers. Furthermore, since the size of a correlation group is small and its membership is stable, an IDS can check using out-of-band mechanisms the reputability of each of the IDSs in its correlation group. It can use this information to decide whether to collaborate. If needed, the IDS can use legal contracts to enforce trust and privacy. If the IDSs choose to collaborate, they use a

secure channel to exchange information so that eavesdroppers cannot snoop.

IDSs need to know which other IDSs are in their correlation group. Non-profit organizations (like CERT and DSHIELD) or commercial entities can discover attack correlation across IDSs and report to each network the identity of the other networks in its correlation group. These entities are termed Attack Correlation Detectors (ACD). A network may participate in one or more ACDs. The choice of ACD may depend on the number and types of networks participating in the ACD, its reputation, etc. The ACD occasionally collects logs from participant IDSs. The logs cover a particular period that can be as small as a single day randomly chosen by the ACD. The logs have minimal sensitive information. Each record in the log provides the following fields: (Time, Source IP, Packet Count). The analysis in 2.2.3 shows that these fields are enough for detecting attack correlation. The ACD performs the correlation analysis and informs each network of its correlation group, expressed as a list of the following records: (correlated IDSs, level of correlation). The correlation analysis is not intensive, the average time taken to analyze a days worth of logs is just 4 hours on an Intel Itanium 1.5 GHz SMP machine with 2 GB of memory. Further since IDS correlation is persistent over atleast a month, the analysis is repeated only after such long periods of time. Once organizations know their correlation group, they can independently decide with whom to collaborate, basing their decisions on the level of correlation and the identity of the peer network.

Integrating new IDSs and updating participant IDSs about changes in their correlation group can be performed incrementally. A new IDSs provides logs from the same collection point so that its correlation group can be found. Updates are incremental, since IDSs need to be informed only if their correlation group changes. Due to the persistence of group membership in these correlation groups (a month or more), the update process can be performed in a lazy fashion with the cost amortized over long periods of time.

It should be noted that acting as an ACD is relatively simple. Indeed, DSHIELD already has the means to provide this service to its participant networks.

4.2 Discussion

4.2.1 Scalability

Correlation-based collaboration ensures scalability by the small size of the groups and the persistence of correlation among IDSs across long timescales. In particular, over 96% of the IDSs in the dataset are correlated with less than 10 other IDSs. The overhead of setting up peering and exchanging information is therefore relatively small. Additionally, the persistence of correlation over months ensures the scalability of ACDs. The ACDs analyze correlation at these timescales, amortizing the cost of the analysis.

4.2.2 Privacy

Recall that for discovering its correlation group an IDS provides the ACD with logs of attacking IP addresses, alert time, and packet count. Thus, none of the sensitive alert fields such as the attack type, the destination, and the destination port, are needed. Also the data is revealed only to the ACD and does not get published. On the other hand, privacy of the data exchanged with one's collaborators is provided largely because IDSs have the ability to independently decide which IDSs to collaborate with, and what to reveal. Further, the persistence of correlation allows the collaborators to use legal contracts to protect their data, if necessary.

4.2.3 Protecting Against Spreading Lies

An IDS that lies about its attackers to the ACD does not harm the system. Such lies are unlikely to be correlated with any of the attacks seen at other IDSs, even if they do, each IDS checks independently the credential of each of its collaborators before sharing any alert data with them. Lying to one's collaborators is unlikely as their reputations are carefully checked and information exchange is protected by legal contracts.

4.3 Picking the Right Collaborators

This section presents a rough evaluation of the overhead and the enhancement in detection capability obtained via various choices of collaborating IDSs for detecting correlated attacks. The following 4 schemes for picking collaborators are compared:

- **Collaborate With ALL IDS:** An IDS collaborates with all other IDSs in the dataset.

- **Correlation-Based Collaboration (CBC):** Each IDS collaborates with only those IDSs in its correlation group.

- **Random Collaboration:** An IDS picks a random set of IDSs to collaborate with. To ensure the comparison with CBC is fair, each IDS collaborates with as many IDSs as there are in its correlation group.

- **Local Detection:** in this scheme, detection is based on local alerts with no collaboration with other IDSs.

4.3.1 Blacklisting Malicious Sources

In order to compare the above schemes, a protocol has to be specified for exchanging alerts and processing the acquired information. The simple approach described below is used. This approach is not necessarily optimal, but it suffices to evaluate the relative benefits of the different methods of picking collaborators.

The IDSs collaborate to detect low rate attackers and speed up the detection of moderate rate attackers. Each IDS maintains a `Blacklisting Threshold` and a `Querying Threshold`. A source IP address is blacklisted when the number of suspicious packets from it crosses a `Blacklisting Threshold`. An IDS queries its collaborators when the number of malicious packets from a source IP address crosses the `Querying Threshold`. If the aggregate rate of the offending source at all collaborators exceeds the `Blacklisting Threshold` the source is blacklisted. Once a source is blacklisted it is set apart for further investigation and an alarm is triggered to all collaborators.

The time taken to blacklist a source depends on two factors; the rate at which the source is attacking as well as the chosen `Blacklisting Threshold`. In picking a particular threshold, there is an inherent tradeoff between false positive ratio and false negative ratio. A low `Blacklisting Threshold` will result in a high false positive ratio while a high threshold will miss many moderate rate attacks resulting in a high false negative ratio. The right value for the `Blacklisting Threshold` is site specific and should be picked to optimize the false negative and false positive ratios.

The ISP and U1 datasets are used to find a good value for the thresholds because these logs contain enough information to distinguish many cases of false positives. The `Blacklisting Threshold` is set to 1000 malicious packets/day because in the dataset, this rate results in a false positive ratio less than 1%. The `Querying Threshold` is set to 50 malicious packets/day. The `Querying Threshold` has to be substantially lower than the `Blacklisting Threshold`, but there is nothing special about the value of 50 packets/day. In reality, these thresholds will vary depending on the local sites configuration as well as the nature of the alert itself. The above thresholds seem reasonable for those IDSs in the dataset for which there is detailed attack information.

To simulate the attacks, the traces in the datasets are replayed. The traces are divided into two equal parts, corresponding to 15 days each. The correlation groups are generated from one set (the training set), while the various schemes for picking collaborators are tested on the other set (the test set).

4.3.2 Detection Speedup

Figure 4-1 plots the time it takes to blacklist a source in each of the four approaches: CBC, Local Detection, Random Collaboration and Collaboration with All IDSs. The time to blacklist a source is defined as the time difference between the instant the source is blacklisted by some IDS and the instant the source was first detected by any of the collaborators. The plots are only for sources detected at more than 1 IDS, because localized sources always require the same time to detect under all four schemes. The malicious sources on the x axis are sorted according to their detection time by Local Detection. Note that for this figure, we set `Blacklisting Threshold` to a total of 1000 packets, rather than 1000 packet/day, so that each approach will eventually detect the malicious source.

The figure shows that, for fast sources which can be detected locally in 5 minutes or less, there is no significant difference among the four schemes. These sources form nearly 25% of all classified common attackers. The curves diverge for most slower sources which take longer to blacklist locally. Random collaboration offers no benefit, i.e., the time taken to blacklist is the same as Local Detection except for a few sources. In contrast, CBC speeds up detection for about 75% of the studied sources, and performs nearly as well as collaborating with all IDSs. There are a small number around 5% of slower sources which take longer to detect in CBC because of them being correlated across IDSs which do not

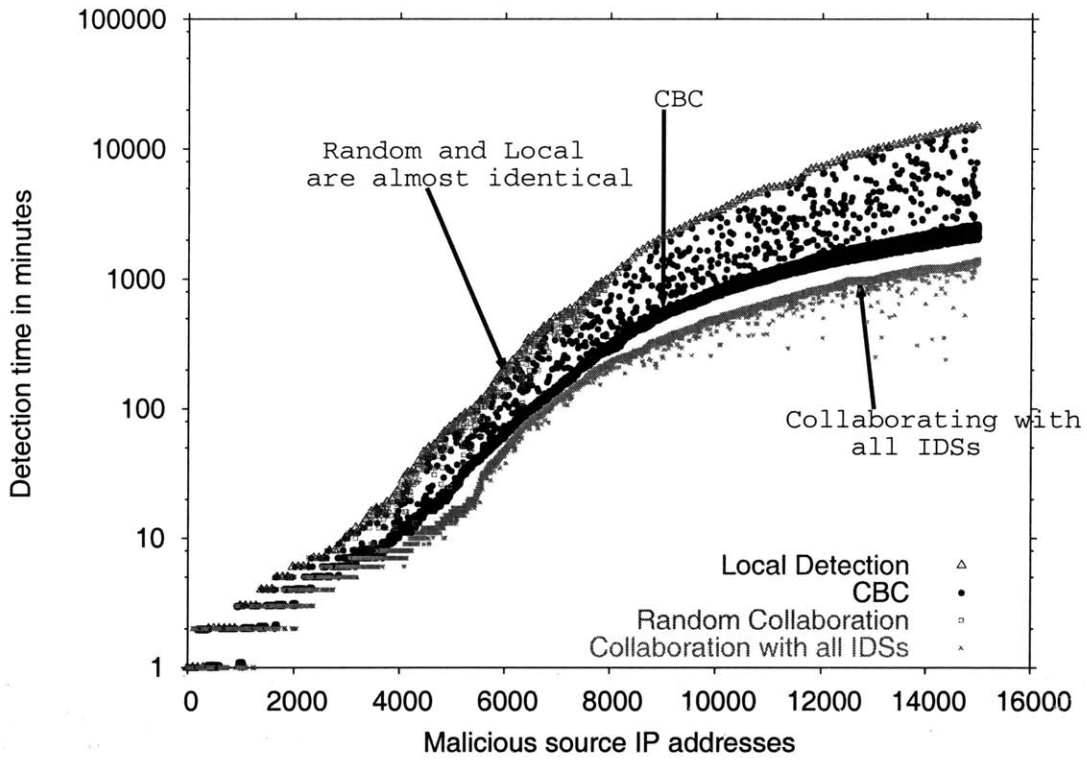


Figure 4-1: Comparison of time taken in minutes for blacklisting a shared malicious source for CBC, Local Detection, Random Collaboration and Collaboration with All IDSs. Short duration attacks (under 5 minutes) which number around 25% do not show significant difference, local detection works nearly as well. CBC performs nearly as well as collaboration with all IDSs in detecting longer duration, slower attacks. Random collaboration offers no benefit except for a few sources.

belong to the same correlation group.

4.3.3 Overhead

In comparison with Local Detection, the speedup in detecting malicious sources is obtained at the cost of communication among the collaborators. The average query rates in CBC and Random Collaboration are fairly close. They both have an average of about 1.3 query/minute/IDS, with a standard deviation of 2.9. In comparison, collaborating with all IDSs has a very high overhead; the average query overhead is 454.9 query/minute/IDS, which is 2 orders of magnitude higher than CBC.

	CBC	Local Detection	Random Collaboration	All IDSs
Alert Reduction	73.44%	35.48%	37.77%	80.56%
Sources missed	5.02%	38.65%	36.69%	0%

Table 4.1: Comparison between 4 schemes for picking collaborators in terms of the reduction in alert volume and the number of malicious sources missed.

4.3.4 Effectiveness

Faster detection of malicious sources also results in significant reduction in alert volume. Table 4.1 lists the average reduction in alert volume from blacklisting under CBC, Random Collaboration, Collaborating with all IDSs and Local Detection. On average, CBC results in 73.44% reduction in alert volume (i.e., the size of the log that admin should examine). The value is close to the one obtained by collaborating with all IDSs, which is 80.56%. Local Detection, on the other hand, performs significantly worse; it reduces the alert volume only by 35.48%. There is no discernible difference between Local Detection and Random Collaboration, the reduction in alert volume is only marginally better at 37.77%. The above numbers are for all attacks, correlated and uncorrelated. Thus by being able to quickly detect correlated attacks, CBC reduces alert volume by a further 38% over Local Detection.

Table 4.1 also lists the number of correlated malicious sources missed by CBC, Local Detection, and Random Collaboration in comparison to Collaborating with all IDSs. A malicious source is missed if the scheme is unable to blacklist it due to incomplete information, though it is blacklisted when each IDS collaborates with all other IDSs. CBC misses only 5.02% of the malicious sources, while Local Detection misses 38.65% of the sources. The Random collaboration scheme is almost similar at 36.69%. All these values are dependent on the thresholds used, but the values demonstrate the order of magnitude improvement obtained in CBC and the relatively insignificant difference between CBC and collaborating with all IDSs. In summary, CBC improves significantly over Local Detection. It increases the number of detected sources by 33%, and reduces the volume of alerts by an extra 38% beyond Local Detection. It performs almost as well as the collaborating with all IDSs. In contrast, a random choice of collaborators is as bad as not collaborating.

Chapter 5

Related work

Recent research has produced a number of proposals for building distributed intrusion detection systems, measurement studies of global attack properties and new intrusion detection systems. This chapter examines each of these related areas in detail and explains how this thesis differs from them.

5.1 Distributed Intrusion Detection Systems

Several proposals exist for building collaborative and distributed intrusion detection systems [25, 10, 11, 31, 23, 27, 26, 8, 33], but none of them studied attack correlation. This thesis extends many of these proposals with a mechanism for picking collaborators, and maximizes the benefit of collaboration while limiting its overhead.

Early distributed intrusion detection systems collect audit data from distributed component systems but analyze them in a central place (e.g., DIDS [25], ISM [10], NADIR [11], NSTAT [31] and ASAX [8]). DIDS [25] focuses on data volume reduction at the local monitors and centralized aggregation and analysis. It also presents some heuristics for tracking attackers which move across networks.

NSTAT [31] models intrusions as an anomalous set of state transitions. Each state corresponds to a computer state, an anomalous sequence triggers and intrusion alert. It can also model state transitions in a distributed filesystem, albeit as a centralized process. ASAX [8] presents a rule based language for audit trail analysis. Further it introduces RUSSELL, a query language for efficiently expressing audit trail queries.

Recent systems have paid more attention to scalability (e.g., EMERALD [23], GrIDS [27],

AAFID [26], and CSM [32]). The Collaborative Intrusion Detection System [17] involves dynamic groups of nodes that rapidly change and exchange information. The set of nodes exchanging information is not constant and is changed continuously to cover all nodes in the system which limits its scalability.

COSSACK [12], another collaborative response framework, is concerned more with alarm propagation than detection itself. It uses signal processing techniques to do blind attack detection without issuing a distributed query. Attack mitigation is done using a topology database to contact the offending network and shut down the attackers.

DOMINO [33] relies on a hierarchy of nodes with different levels of trust and aims to exchange blacklist information. The nodes are placed such that IDSs protecting networks with close destination address spaces are close together. Trusted nodes known as axis nodes do the exchange of alert data, the untrusted nodes are used only for local blacklisting. DOMINO [33] does not discuss how this topology of axis nodes should be constructed, hence it is unclear how efficient alert propagation will be.

The Distributed Intrusion Detection System (DIDS) [25] addresses system attacks across a network. Attacks such as doorknob, chaining, and loopback could be detected when data from hosts within a given network was combined under centralized control. Clever attackers could still subvert DIDS by reducing the volume of attacks for a given network.

EMERALD addresses intrusions within large separately administered networks [23]. EMERALD includes features for handling different levels of trust between the domains from the standpoint of a centralized system: individual monitors are deployed in a distributed fashion, but still contribute to a high-level event-analysis system. EMERALD appears to scale well to large domains. The Hummer project [18] focuses on the relationships between different IDSs (e.g., peer, friend, manager/subordinate relationships) and policy issues (e.g., access control, cooperation policies).

Finally, there has been work on specification and event abstraction to allow multiple IDS boxes to share attack information and collaborate on detection and protection [5, 28, 7].

5.2 Attack Measurements & Analysis

A lot of work has been done in characterizing attack characteristics. Yegneswaran et al. [33, 21] study the global characteristics of intrusions as well as Internet background radiation.

Intrusions are shown to have a zipf distribution, a few attackers contribute most of the alerts. Further attackers were shown to exhibit significant spatial trends, underscoring the need for global intrusion detection.

Network telescopes have been used to study DoS activity in [20]. The paper presents a new technique, *backscatter* analysis which estimates attack activity by observing residual fallout from attacks in unused Internet address spaces. The technique assumes a uniform distribution of attack activity throughout the IP address space, but recent trends in denial of service attacks which use botnets make that assumption weak.

Placement of blackholes in a distributed Internet setting for global threat detection is addressed in [3]. The paper discusses the optimal location of telescopes to ensure coverage of the entire Internet address space. This is for attacks which use spoofed sources only, but a large number of recent attacks have moved away from plain bandwidth flooding to higher level resource consumption attacks.

5.3 Analysis of Intrusion Alerts

GrIDS [27] collects traffic and connections data. It analyzes TCP/IP network activities using activities graphs and reports anomalies when activity exceeds an user specified threshold. Methods of discovering intent by correlating alerts from different IDSs are presented in [14]. Algorithms for sharing of alerts [16] in a privacy-preserving manner could be a future avenue of research. Alert correlation to reduce the number of alerts to be manually examined is discussed in [4]. Alerts are inserted into a relational database to be aggregated and the summarized alert is presented to the operator. These are orthogonal to the present thesis and can be easily integrated.

Automatic Signature Detection: Work orthogonal but integrable into this thesis' framework include Autograph [13], Earlybird [24] which do blind signature detection using fast Rabin fingerprinting on packets.

Chapter 6

Concluding Remarks

This thesis presented the first wide-scale study of attack correlation in the Internet i.e., attacks that share the source IP but occur at different networks. The dataset, consisting of alert logs collected at 1700 IDSs, show that correlated attacks are fairly prevalent in today's Internet; 20% of all the attacking sources are shared attackers, and they are responsible for 40% of all alerts in the logs. Shared attackers attack different networks within a few minutes of each other, emphasizing the advantage of realtime collaboration between victim networks as opposed to sharing attack information offline.

The results also show that the 1700 IDSs can be grouped into small correlation groups of 4-6 IDSs; two IDSs in the same correlation group share highly correlated attacks, whereas IDSs in different correlation groups see almost no correlated attacks. Furthermore, the correlation groups are stable and their membership persists for months. Though not conclusive, the analysis indicates that similarity in the software and services running on the protected networks causes their IDSs to show attack correlation.

The empirical results have important implications for collaborative intrusion detection of common attackers. They show that it is quite important that each network/IDS picks the right collaborators. Exchanging alerts with thousands of IDSs in realtime is impractical because of the resulting overhead and the lack of trust between these networks. But, using a trace-driven simulation the thesis shows that picking at random a smaller and fixed set of collaborators has almost no benefits beyond local detection. In contrast, collaborating with the 4-6 IDSs in one's correlation group has almost the same utility as collaborating with all 1700 IDSs in the dataset with 350 times less overhead.

Finally, these results reflect the state of the Internet at the end of 2004 and the beginning of 2005. It is hard to predict the extent of attack correlation in the future and the continuous existence of correlation groups. Future research should investigate these characteristics and track their evolution.

Bibliography

[1] Computer Emergency Readiness Team. http://www.us-cert.gov/.

[2] Distributed Intrusion Detection System. http://www.dshield.org/.

[3] Evan Cooke, Michael Bailey, David Watson, Farnam Jahanian, and Danny McPherson. Towards understanding distributed blackhole placement. In *The 2nd Workshop on Rapid Malcode (WORM) Fairfax, Virginia, October 29, 2004.*

[4] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *2002 IEEE Symposium on Security and Privacy.*

[5] D. Curry and H. Debar. Intrusion detection message exchange format: Extensible markup language (xml) document type definition, 2001.

[6] F-SECURE. F-secure virus descriptions : Santy. http://www.f-secure.com/v-descs/santy_a.shtml/.

[7] B. Feinstein, G. Matthews, and J. White. The intrusion detection exchange protocol (idxp), 2003.

[8] Naji Habra, Baudouin Le Charlier, Abdelaziz Mounji, and Isabelle Mathieu. ASAX : Software architecture and rule- based language for universal audit trail analysis. In *ESORICS*, 1992.

[9] J. W. Haines, R. P. Lippmann, D. J. Fried, E. Tran, S. Boswell, and M. A. Zissman. 1999 DARPA Intrusion Detection System Evaluation: Design and Procedures. In *MIT Lincoln Laboratory Technical Report.*

[10] L. T. Heberlein, B. Mukherjee, and K. N Levitt. Internet security monitor: An intrusion detection system for large-scale networks. In *Proceedings of the 15th National Computer Security Conference*, 1992.

[11] J. Hochberg, K. Jackson, C. Stallings, J. McClary, and J DuBois, D.and Ford. NADIR: An automated system for detecting network intrusions and misuse. In *Proceedings of Computers and Security 12(1993)3*, 1993.

[12] A. Hussain, J. Heidemann, and C. Papadopoulos. COSSACK: Coordinated Suppression of Simultaneous Attacks. In *DISCEX*, 2003.

[13] H. Kim and B. Karp. Autograph: Automated, Distributed Worm Signature Detection. In *Usenix Security 2004*.

[14] C. Krugel, T. Toth, and C. Kerer. Decentralized Event Correlation for Intrusion Detection. In *4th International Conference on Information Security and Cryptology 2001*.

[15] John Leyden. The illicit trade in compromised PCs, 2004. http://www.theregister.co.uk/2004/04/30/spam_biz/.

[16] P. Lincoln, P. Porras, and V. Shmatikov. Privacy-Preserving Sharing and Correlation of Security Alerts. In *Usenix Security 2004, San Diego, CA*.

[17] M. Locasto and et al. Collaborative Distributive Intrusion Detection. In *CU Tech Report CUCS-012-04*, 2004.

[18] J. McConnell, D. Frincke, D. Tobin, J. Marconi, and D Polla. A framework for cooperative intrusion detection. In *NISSC*, pages 361–373, 1998.

[19] D. Moore, C. Shannon, and J. Brown. Code-Red: a Case Study on the Spread and Victims of an Internet Worm. In *Internet Measurement Workshop, 2002.*

[20] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring internet Denial-of-Service activity. In *USENIX Security 2001.*

[21] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet Background Radiation. In *Proceedings of the IMC 2004.*

[22] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(23–24):2435–2463, 1999.

[23] P. A. Porras and P. G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *Proc. 20th NIST-NCSC National Information Systems Security Conference*, 1997.

[24] Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage. Automated worm fingerprinting. In *Proceedings of the 6th ACM/USENIX OSDI, San Francisco, CA, December 2004.*

[25] A. Snapp and et. al. Distributed intrusion detection system - motivation, architecture, and an early prototype. In *Proceedings of the 14th NCSC*, 1991.

[26] E. Spafford and Zamboni. D. Intrusion detection using autonomous agents. In *Computer Networks, Volume 34*, 2000.

[27] S. Staniford-Chen and et. al. GrIDS – A graph-based intrusion detection system for large networks. In *19th National Information Systems Security Conference*, 1996.

[28] B.; Staniford-Chen S.; Tung and D. Schnackenberg. The common intrusion detection framework (cidf). In *Information Survivability Workshop, Orlando FL*, 1998.

[29] The Honeynet Project. http://www.honeynet.org/.

[30] The Honeynet Project. Know your Enemy: Tracking Botnets. http://www.honeynet.org/papers/bots/.

[31] G. Vigna, S. Eckmann, and R. Kemmerer. The stat tool suite. In *In Proceedings of DISCEX*, 2000.

[32] U. White, G. B.; Pooch. Cooperating security managers: distributed intrusion detection systems. In *Computers & Security, Vol. 15, No. 5*, pages 441–450, 1996.

[33] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: Global characteristics and prevalence. In *In Proceedings of ACM SIGMETRICS,*, 2003.

[34] Stefano Zanero. Behavioral Intrusion Detection. In *ISCIS 2004.*