ALERT-ID: Analyze Logs of the network Element in Real Time for Intrusion Detection

Jie Chu[1], Zihui Ge[2], Rick Huber[2], Ping Ji[3], Jennifer Yates[2] and Yung-Chao Yu[2]
[1]Graduate Center, CUNY, jchu1@gc.cuny.edu
[2]AT&T Labs - Research, {gezihui, rvh, jenniferyates, yungyu}@att.com
[3]John Jay College of Criminal Justice, CUNY, pji@jjay.cuny.edu

August 23, 2011

Abstract

The security of the networking infrastructure (e.g., routers and switches) in large scale enterprise or Internet service provider (ISP) networks is mainly achieved through mechanisms such as access control lists (ACLs) at the edge of the network and deployment of centralized AAA (authentication, authorization and accounting) systems governing all access to network devices. However, a misconfigured edge router or a compromised user account may put the entire network at risk. In this paper, we propose enhancing existing security measures with an intrusion detection system overseeing all network management activities. We analyze device access logs collected via the AAA system, particularly TACACS+, in a global tier-1 ISP network and extract features that can be used to distinguish normal operational activities from rogue/anomalous ones. Based on our analyses, we develop a real-time intrusion detection system that constructs normal behavior models with respect to device access patterns and the configuration and control activities of individual accounts from their long-term historical logs and alerts in real-time when usage deviates from the models. Our evaluation shows that this system effectively identifies potential intrusions and misuses with an acceptable level of false alarms.

1 Introduction

A fundamental aspect of network security is securing the networking infrastructure itself, which can be particularly challenging in a large scale enterprise or ISP (Internet service provider) network. In such networks, hundreds or thousands of routers and switches are widely dispersed among a geographically diverse set of offices and are typically managed by a large team of network operators. It is imperative that the networking infrastructure and the information contained therein be fully protected against any malicious priers and attackers. For example, information available at networking devices, such as router configuration and traffic statistics, may contain confidential business data of tremendous value to a business competitor. Divulging such information will likely result in a significant disadvantage to the ISP's business. Leakage of some critical security information in the router configuration such as QoS policy or firewall/ACL (Access Control List) settings may subject the network to crafted and targeted attacks such as DDoS (Distributed Denial of Service) attack. Or in an even more devastating scenario, malicious attackers gaining privileged access to the networking device might alter the network configuration to create havoc and paralyze the entire network and the services it supports.

Given the risk of severe consequences, large scale networks typically devise and deploy a range of security and protection measures for their networking devices. One common practice is to utilize the combination of *periphery protection* and centralized *authentication and authorization* for communication to networking devices. By restricting premises access, unauthorized persons are blocked from gaining physical access to networking devices. Through careful configuration of ACLs at all network edge routers, unauthorized network traffic is also blocked from reaching network devices. And finally, technologies such as TACACS+ (Terminal Access Controller Access-Control System Plus) [1] and RADIUS (Remote Authentication Dial In User Service) [2], ensure that only authenticated users (i.e. authorized network operators/administrators) have access to routers and switches (either directly or remotely over the network).

The architecture above is very effective against threats from external attackers when working properly. However, there is always the possibility that building security is breached, allowing physical access to router hardware, or that ACLs on an edge router are misconfigured, admitting attacking traffic. Furthermore, with a large team of network operators, compromised users or compromised user accounts can be a critical source of potential security troubles arising inside the network.

In this paper, we propose to add another layer of defense for networking infrastructure by overseeing *all* operations being done in the network, and automatically detecting and raising alarms for "suspicious" activities. We leverage the existing authentication and authorization framework and collect router/switch access logs in real-time. We develop an anomaly detection system that compares on-going router/switch access activities against a set of patterns or profiles constructed from historical data, and once an anomaly is identified, triggers an alarm to network security managers for further investigation of potential intrusions and misuses.

Although the concept of intrusion detection system is well established in computer system security, applying the idea in networking device management remains unexplored, interesting, and challenging. To detect abnormal activities, we must obtain data on routine/normal network management activities in a large scale network, analyze that data, and determine what features best distinguish normal activities from abnormal ones. In our study, we base our analysis on the real network data from one of the largest ISP networks, which comprises tens of thousands of routers distributed worldwide. We conduct an in-depth analysis on a wide range of different characteristics about operators' access patterns and identify useful features. The effectiveness of an intrusion detection system is known to be limited by noisy baseline behavior and hence high false positives. Thus, when developing the detection methodology and the prototype system for capturing potential intrusions and misuses, we focus on managing false positives to be well within an acceptable range. Any given attack is likely to come from a small number of source subnets or accounts. Thus we aggregate detected "threat scores" by their origin source addresses and login accounts. This allows us to amplify the signal of offense and hence be able to detect offenders while they are still exploring the network before large-scale damage is inflicted.

Our contribution in this paper can be summarized as follows:

- We propose to systematically monitor and analyze the networking device access logs to protect the networking infrastructure. To the best of our knowledge, this is the first study that focuses on monitoring and auditing networking device access and control logs to catch anomalous activities.

- We analyze TACACS+ logs collected over more than six months from a tier-1 ISP network and identify a set of features that can be utilized to distinguish suspicious activities from normal operations — such as the login ID and originator IP prefix association pattern, the

daily number of distinct routers accessed, and the number of hops over which an operator logs on to a router from a different router.

- we develop a system tool for the ISP network. Our controlled experiment shows that it successfully identifies injected "malicious" activities – with corresponding threat scores significantly higher than those of day-to-day operational activities.

The rest of the paper is organized as follows. In Section 2, we provide an overview of operational management activities in large scale IP networks and a brief introduction of the authentication, authorization, and accounting system from which we collect logs. Section 3 presents our analysis result on the characteristics of normal operation activities. Section 4 describes the rules and detection system that we build for detecting and alerting on suspicious router accesses and controls. We evaluate our overall system performance in Section 5. We discuss related work in Section 6 and finally conclude the study in Section 7.

2 Background

2.1 Managing IP networks

We first provide an overview of the various types of management activities in large scale IP networks. We describe these in the setting of a global ISP network although many of them are fundamental to large enterprise networks or regional ISP networks as well.

Managing a global ISP network requires a large team of network operators. These operators are typically organized in a tier structure – lower tier operators respond to more routine issues following a set of predefined MOPs (Maintenance Operation Protocols), while more complex matters are escalated to upper tier operators, who have more profound knowledge and deeper understanding of the network. Truly complicated ones are further passed to a small group of experts, possibly including support teams of vendors of involved devices.

Different tiers of operators have different functional roles. Some may be dedicated to the care of a high profile enterprise customer, in which case they will frequently access provider edge (PE) routers but seldom touch backbone routers. Some operators may be responsible for servicing the metropolitan area network for a certain region. Others may oversee control plane health (e.g., router CPU utilization) for the entire network. Depending on their role, operators are expected to have distinct patterns of network management activities.

Network operators often exercise control over routers and switches by logging on to the device. Today, nearly all networking devices support console access via direct connection to the device and remote access via `ssh` or `telnet`. Control is exercised by invoking a sequence of commands through the Command Line Interface (CLI) of the device's operating system. For example, on Cisco IOS, typing

```
ping 1.2.3.4
```

triggers a ping test from the router to the IP address. And typing

```
enable
configure terminal
interface Ethernet0
shutdown
exit
```

administratively shuts down the interface *Ethernet0* at the router.

Note that Cisco IOS supports two different access levels – user level and privileged level. The `enable` command in the above example enters the privileged level, in which configuration change

(`configure terminal`) is allowed. Such capability is widely supported on other vendor systems such as Juniper JunOS as well. In addition, AAA systems (described in the next subsection) support finer grained command groups. A user cannot invoke commands outside of his/her predetermined access levels or command groups.

In addition to operators typing commands via the CLI, ISPs rely on a broad range of automated tools for their network management activities. These tools are typically designed to achieve a specific type of function. For example, automation tools/systems that perform configuration auditing periodically sweep through the entire network issuing a `show running-config` command to collect active router configurations. Another tool might collect hardware, traffic, or protocol status and statistics information by logging into the routers of interest and invoking commands such as `show process CPU history`, `show interfaces POS 1/0`, and `show ip bgp summary`. These software systems may use designated logins when requesting access to networking devices.

Using the combination of function-level controls via various automated systems and manual command-level controls, operators are able to accomplish a wide range of network management tasks including provisioning and decommissioning customer services, troubleshooting networking and service problems, performing device life cycle management, conducting measurements, and monitoring the health of the network and services.

2.2 Authentication, Authorization and Accounting

The networking infrastructure in large scale networks is typically protected by an AAA (Authentication, Authorization, and Accounting) system. There are two mainstream AAA frameworks widely used commercially – TACACS+ (Terminal Access Controller Access-Control System Plus) [1] and RADIUS (Remote Authentication Dial In User Service) [2]. While differing in some specifics, such as whether authentication and authorization are separately maintained in user profiles, both systems use one or more centralized servers to verify a user's identity (authentication) on login, verify access privilege (authorization) on a per command basis, and record all users' activites in their logs (accounting). The log entries contain critical information which includes

(i) the timestamp of the access request
(ii) the IP address of the targeted network device (e.g., the router Loopback address)
(iii) the IP address of the remote user requesting access
(iv) the user's login ID
(v) the command line executed
(vi) other information such as user terminal, user privilege level, and timezone.

To enable a deployed TACACS+ or RADIUS system in the network, all routers in the network need to be configured with the IP addresses of the servers – typically there are multiple replicated servers for redundancy. A large network can further be divided into multiple zones, for example, by the device type or by the autonomous system (AS) that they belong to. Different zones may have different sets of TACACS+/RADIUS servers, which may contain different user account and privilege settings.

3 Characteristics of Normal Operation Activities

As described in the Introduction, our objective is to monitor all operations activities in the network and detect potential intrusions and misuses. This serves as an additional security protection for the networking infrastructure. Hence, we do **not** assume network periphery protection and AAA

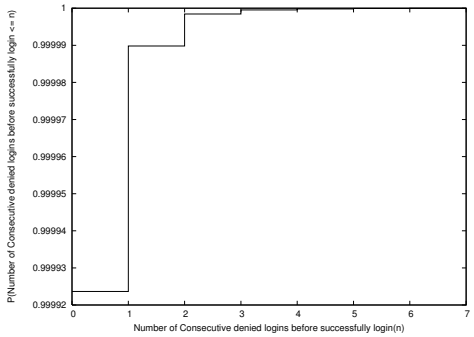

Figure 1: CDF of the number of login attempts before a success

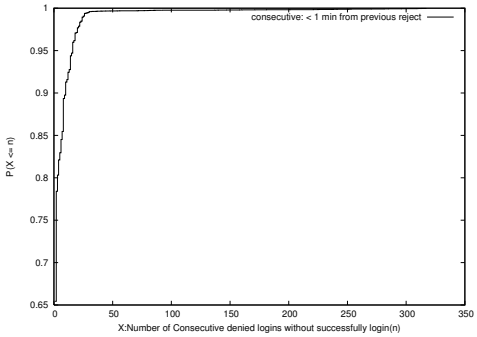

Figure 2: CDF of the number of consecutive login failures from a common originator IP

are working to their fullest extent – e.g., all users' login credentials are kept confidential to the operator only, and the periphery protection is effectively blocking all external attackers.

We start by examining normal network operational activities recorded in the AAA logs. We focus on aspects that would best distinguish normal activities from actions that an external or internal attacker might take. In the following analyses, we use data collected from a global tier-1 ISP network which generates tens of millions of TACACS+ log entries accessing tens of thousands of routers per day.

3.1 Failed login attempts

The most intuitive way to separate potential attacks from legitimate accesses is to check whether they can readily pass authentication. Attackers may expose themselves by inputting the wrong login credentials. However, it is also expected that legitimate users sometimes "fat finger" their login ID or password. Thus, we examine failed login attempts in normal TACACS+ logs (using one month's data).

Figure 1 plots the cumulative distribution function (CDF) of the number of consecutive login attempts before a successfully authenticated login. We consider a login request within one minute of a preceding one with the same originator IP, the same login ID, and the same target networking device as a consecutive login.

We observe that more than 99.992% of logins pass authentication the first time. More than 85% of the remaining ones input the correct credentials the next time, and it is extremely rare that a user fails more than five times before finally getting it right. The ratio of login failure is considerably lower than that typically seen in computer systems [3]. This is likely due to the predominance of logins generated by automated network management tools in our data – a unique characteristic of network infrastructure operations. Figure 1 demonstrates the potential of generating intruder alarms when a small number (e.g., 6) of repeated failed logins is seen.

This type of monitoring can be defeated if the attacker has a list of valid login IDs and device names – they can use a different login ID or target a different device when an attempt fails. We can improve detection by looking for consecutive login failures from a common originator IP irrespective of the login ID used. Figure 2 plots the CDF of the number of consecutive (i.e., less than one minute apart from the preceding one) login failures. We observe that around 85% of the rejected login attempts are either rectified or abandoned in six times or fewer. However, there are some login attempts going as high as a few hundreds in a row. Manual inspection finds that they are due to network management scripts running out of sync with router CLI (e.g. sending password when login

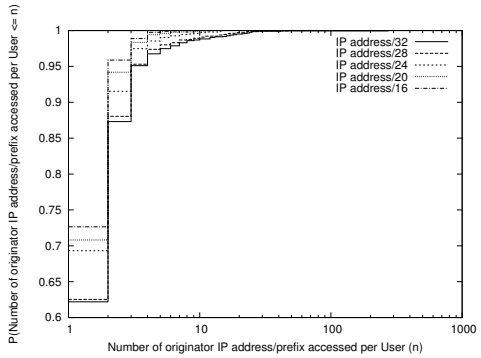

Figure 3: CDF of the number of originator IP (prefix) per login ID

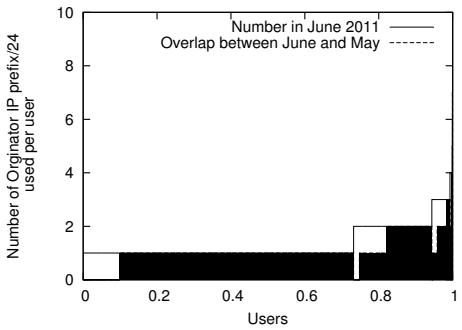

Figure 4: Stability of the login ID and originator IP prefix association

ID is expected or *vice versa*). This rarely occurs, but when it does it produces many consecutive login failures – and correctly generates an alarm.

3.2 Login access pattern

As described in Section 2, an AAA log entry contains login access information characterized by the user login ID, the originator IP address, and the target router IP address. We define a *login session* as the network management activities sharing the common triple and being close in time (e.g., with an idle timeout of 10 minutes).

The login access information can be valuable in capturing attackers. For example, an originator IP that is not part of a block of addresses previously seen as an originating address in the logs is a strong indication that the network periphery protection may have a hole. Furthermore, each network operator typically has a rather stable set of work locations from which he/she manages the network, and due to his/her role, there can be a fixed set of network devices that the operator typically manages. So source and destination IP addresses will tend to be consistent over time for many operators.

We first look at the association of the operators' login and the originating IP address. Figure 3 plots the CDF of the number of distinct originator IP addresses associated with a login ID in a month. We observe that 62% of login IDs manage the network from only one IP address. If we consider common originator subnets (with varying size), the number rises to 69% for /24 IP prefixes and 72% for /16 IP prefixes. In the rest of the paper, we will use /24 IP prefixes when aggregating originator IP address – it is not excessively large, yet can accommodate most of logins from the same facility/office.

Figure 3 also shows that even with /24 originator IP prefixes, about 1% of the login IDs access the network from more than 10 distinct IP prefixes. Looking into those, we find that there are cases when an operator first logs on from a gateway server to a router, and then logs on to other routers from that router. The loopback IP address of the first hop router appears as the originator IP for the second access session. While such "stepping-stone" access sessions are not common, they do occur – operators use this either for convenience or under certain network conditions, for example, when direct access to the other routers is unavailable. We can tighten our rule to deal with this situation by excluding sessions which originate on a router or switch. This removes "stepping stone" sessions from the analysis. The solid line in Figure 4 plots the number of distinct originator IP prefixes against the rank of login IDs – to protect proprietary information, we normalize the rank of login IDs to be between 0 and 1. We find above 73% of users have only one (non-stepping-stone)

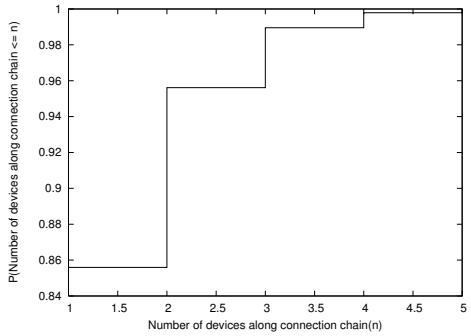

Figure 5: CDF of the number of routers in a "stepping-stone" chain

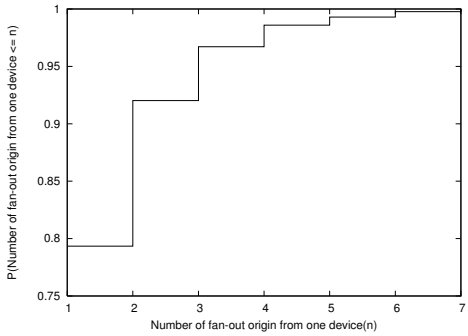

Figure 6: CDF of the size of outbound fan-out in a "stepping-stone" chain

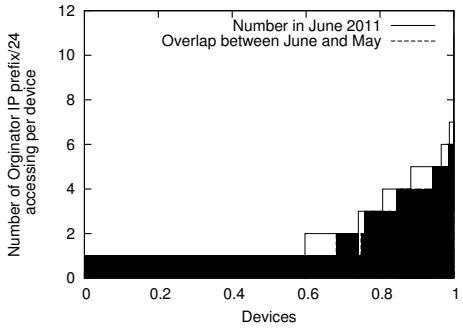

Figure 7: Stability of the networking device and originator IP prefix association

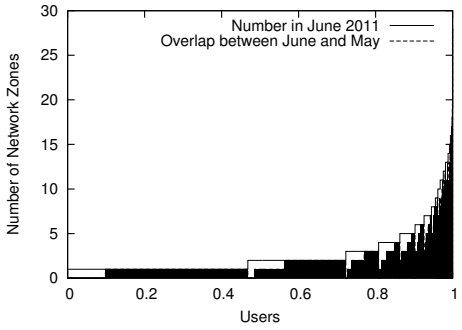

Figure 8: Stability of the login ID and network zone association

originator IP prefix and no one logs on from more than 5 distinct IP prefixes. This indicates that there exists a strong stability in the access pattern characterized by login ID and originator IP prefix combination – deviating from it can be a symptom of attacks. Figure 4 also plots the stability of this access pattern month by month – the shaded area indicates that the same login ID and IP prefix association has appeared in the preceding month. This demonstrates strong predictability based on past access behavior pattern (the unshaded area is mostly due to new users or infrequent users who only access the network in the second month).

Going back to the "stepping-stone" sessions, by matching the *ssh* command on the first hop router and the remote login log request on the second router, we can reconstruct the chain of stepping-stones. Figure 5 and Figure 6 plot the distribution function of the length of these chains and the outbound fan-out of these chains. It is evident that both attributes are bound by a small number (e.g., 7) in normal operational activities. In contrast, an intruder working from a compromised router may attempt to gain information from a large number of other routers, which is likely to produce long chains or high fan-outs. Watching those attributes closely can be an effective way to catch the intruder.

We next turn to the association between the networking device and the IP prefix from which management control activities originate. The solid line in Figure 7 plots the number of distinct originator IP prefixes versus the rank of the networking device IDs. As in Figure 4, we normalize the ranks to be between 0 and 1 to protect proprietary information. We find that 60% of the routers are only controlled from hosts within one /24 IP block during a one month period. These control activities are likely routine network auditing and health monitoring. A small portion of the

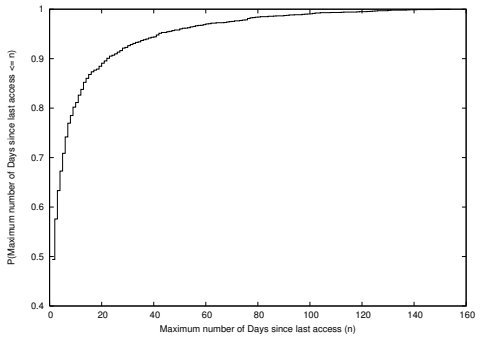

Figure 9: CDF of the maximum gap in days between consecutive logins per ID

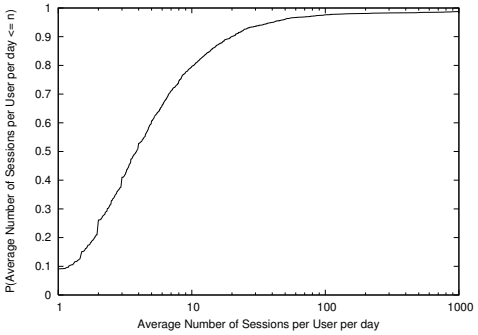

Figure 10: CDF of the average number of sessions initiated per login ID per day

network devices are managed from a small number of (e.g., 2-7) IP prefixes which correspond to the network operation centers responsible for those devices. In normal operations activities, some access across regions is unexpected. While a major Network Operations Center may deal with equipment anywhere in the world, it is suspicious if an operator from a regional office in Japan requests access to a router serving IPTV in the USA. Catching abnormal associations between routers and originator IP prefixes can be an effective way of identifying such cases. The shaded area in Figure 7 shows the overlapping associations that have appeared in the preceding month; this demonstrates the predictability of these associations. The unshaded area is mostly due to the limitation of using one-month data for comparison. This can be greatly reduced when we consider longer historical data.

Finally, we examine the association between login IDs and network devices. Many users or software tools have limited scope in terms of the networking devices managed. The solid line in Figure 8 plots the number of distinct network zones (described in Section 2) that each login ID has accessed in a one month period. We again normalized the x-axis to avoid disclosing the size of the operator work-force. We observe that the majority of login IDs have a very limited scope (e.g., less than 3 zones) while a small number of high-tier operators or software tool IDs access many zones. The stability of the login IDs' access pattern is depicted by the month over month comparison shown in the shaded area. We observe a strong month-by-month predictability that can be utilized for detecting intrusions or misuses.

3.3 User behavior

As mentioned in Section 2, different login IDs (corresponding to different operators or network management tools) have different roles/functions. Each user is likely to have a roughly stable behavior in access schedule (frequency), type of control (e.g., monitoring, or troubleshooting, or configuration change), and class of commands (e.g., SONET controller settings versus ACL configurations). Significant deviation from normal behavior can be a symptom of an account becoming compromised and an intruder impersonating the owner of the login account. In this subsection, we examine the properties of such user behaviors exhibited in normal network management activities.

We first examine the inter-session time distribution. Figure 9 plots the maximum difference in days between two consecutive logins from the same ID in a six-month period. We observe a wide variability among different login IDs. Many login IDs access the network on a regular basis, with at most a few days gap. But there are a considerable number of login IDs that only access the network occasionally. This suggests that it may be helpful to profile login IDs in different groups

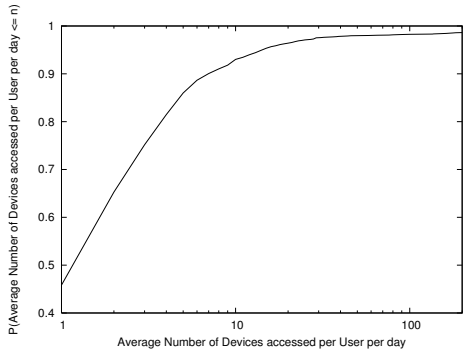

Figure 11: CDF of the average number of devices accessed per login ID per day

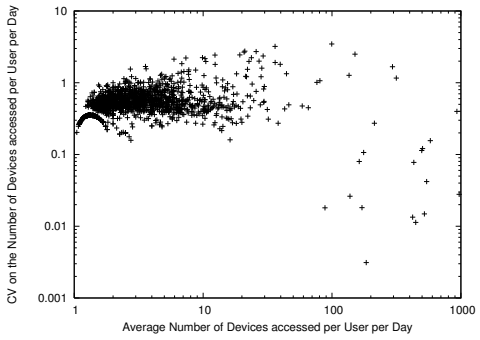

Figure 12: CV versus mean of the number of distinct devices accessed per login ID per day

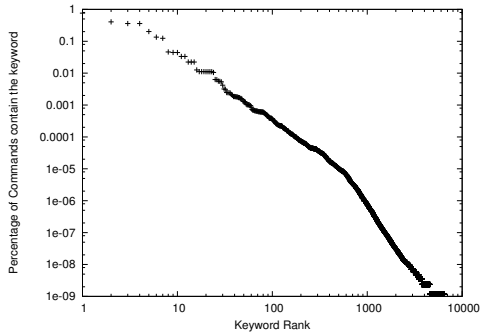

Figure 13: Keywords Frequency Distribution

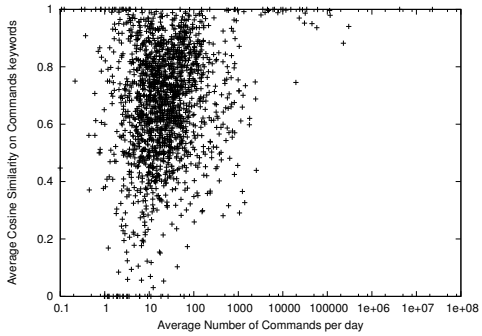

Figure 14: Average Cosine similarity on the keywords frequency per login ID

according to their access frequency.

Figure 10 shows the CDF of the average number of login sessions per login ID per day in a representative month. Here we exclude the days when the login ID is not active in the average statistic. The tail part of the curve, which goes several order of magnitude larger, is cut off so that we do not disclose the exact number of devices in the network – similarly for Figure 11. We observe that the majority of the login IDs have only a few login sessions per day. For example, 65% of IDs log onto the network no more than 5 times daily (on average). There are also many software tools and network management scripts producing over a hundred login sessions on daily basis. A login account suddenly changing its behavior, especially from having a small number of login sessions daily to a large number of them on a given day, is unusual or abnormal behavior and should be examined to see if it indicates a problem. Similarly, Figure 11 shows the CDF of the average number of distinct networking devices accessed per login ID per day. Compared to Figure 10, it shows even more concentration – 65% of IDs log on to no more than 2 networking devices daily (on average). The tail portion of the curve again is dominated by software tools monitoring a large number of devices regularly, such as network configuration auditing tools. A surge in the number of distinct networking devices that a user initiates in a short period of time might be an intruder scouting for information. To understand the variability on this metric, Figure 12 shows a scatter plot of the coefficient of variation (CV) versus the mean – each point represents one login ID. We find that most of the CVs are bound by a small number (e.g., 3), while the login IDs with large number of average daily device accesses typically have much smaller CVs – suggesting that they can be more tightly bounded.

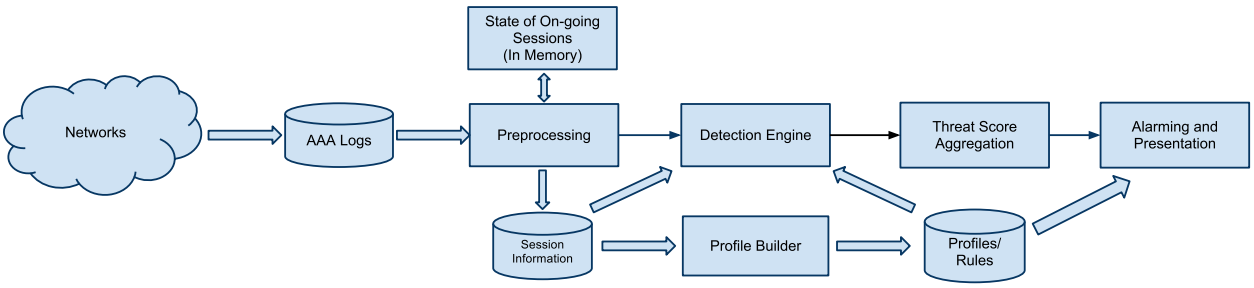

Figure 15: Detection System Architecture

The set of router control commands and configurations used by a login ID is expected to exhibit some stability too. For a login account used by a software tool, the set of commands is determined by programming and rarely changes. For an operator's login, the subset of commands should be subject to his/her privilege level and tightly related to his/her job role. However, obtaining the exact association between login ID and the subset of the commands is a challenging task – for example through code analysis of the software tool.

In contrast, we take an approach that is detached from the semantics of router control commands, as follows: (a) we tokenize the commands (i.e., separate words in the command by white space); (b) we consider the tokens that contain any number as parameters (e.g., IP address) and remove those tokens; (c) we remove any non-alphabetic characters in each of the tokens and convert the remainder into lower case letters – we will refer to these as the *keywords*; (d) we profile each user with the set of keywords used.

Figure 13 shows the likelihood that a keyword is present in a command (sorted in decreasing order) based on one month of logs. We observe a strong skewness in the distribution, which can be well modeled by the Zipfs' distribution. The high ranked keywords are those used in monitoring network health (e.g., ping, vrf, show). And most of the bottom ranked ones are some arbitrary tokens (such as customer name) referenced in the description field of certain router configurations or some misspellings (due to typos by operators) of other keywords. It is sufficient to keep track a subset of keywords (e.g., top 1000) and represent the remainder simply as the *other* keyword. Figure 14 shows the stability of the use of keywords per login ID, which plots the average cosine similarity of the keyword frequency distribution comparing one day against the previous active day. Login IDs with a high number of daily log entries trend to have high predictability one day to another. Deviation from the regular command keywords, especially for a software tool account, can be a symptom of an intruder impersonating the owner of the account.

4 Design of an Online Intrusion Detection System

Based on the analysis from the previous section, we design an online intrusion detection system that oversees the network management activities of the ISP network and detects and alarms on anomalous patterns. Figure 15 shows the system architecture. We collect the logs from AAA servers in near real-time. The logs are fed into an online preprocessing module, which extracts critical information and updates on an entry by entry basis the running states of sessions, login IDs, originators and commands that are required for the different intrusion detection rules. Periodically (e.g., every day) the running states are fed into an offline profiling module in which the different profiles required by the rules are updated – the initial profiles can be constructed via offline analysis of an extended period of historical data. The online rule checkers examine the running states against

the profiles and rules and tag the corresponding log entries with a threat score. An aggregation module then sums the threat score in a window according to the login ID or originator IP and finally an alarming and presentation module makes the information available to network security operators.

4.1 Domain knowledge-based rules

We first define a set of rules that is specific to the network under study. We maintain a list of the IP address blocks that belong to the ISP network. We check the originator IP of each AAA log entry against the list. An IP address from outside of the network indicates a breach of the periphery protection that the ISP has deployed, and consequently the log entry is given a high threat score.

We also track the timestamp of the last login failure from each originator IP address and if a new failed login attempt is observed within T_1 seconds we update the timestamp and increment the count of consecutive login failures for the originator IP. Once the count exceeds a threshold N_1, we output the entries of these login attempts and assign a threat score to each of them. The timestamp and failure counts are reset when a successful login from the originator IP is made or the timeout T_1 is exceeded. With such a rule in place, an intruder that attempts to stay under the radar has to significantly slow down its attack, reducing the efficacy of the attack and prolonging the exposure.

Another rule in this category is for stepping stone sessions. We trace stepping stone accesses as they occur and when the length of the access chain becomes greater than a threshold N_2 or the fan out becomes greater than N_3, we assign a threat score to the sessions involved.

4.2 Rules based on access pattern profiles

In our daily association profile we keep track of the following attributes: (1) originator IP prefix (2) login ID (3) $\langle$ login ID, originator IP prefix $\rangle$ (4) $\langle$ login ID, device zone $\rangle$ (5) $\langle$ originator IP prefix, device zone $\rangle$. For each entry we track the most recent date of appearance and the cumulative number of appearances. We delete an entry when the most recent appearance is more than T_2 (e.g., 180) days and add new entries to the long term profile once their count is sufficiently large.

We assign a threat score to sessions that do not match the existing profile. Note that if a session is from a new originator IP or new login ID, we do not include the threat score due to the lack of associations in (3), (4) and (5). The weight of the threat score of new associations of (3), (4) is set to be higher as the cumulative count of the login ID increases – our confidence to assert suspicious activities increases with more history data. Similarly, the weight of the threat score for (5) increases when the cumulative count for the corresponding originator IP prefix increases.

4.3 Rules based on statistical models of the access profile

We keep track the mean and variance of the following attributes: (1) daily number of sessions per login ID (2) daily number of distinct routers accessed per login ID (3) daily frequency count of command keywords per login ID for top N_4 and the *other* keywords.

We use the EWMA (Exponentially Weighted Moving Average) algorithm in estimating the running statistics for attribute X on day t:

$$Mean_t = \alpha X_t + (1-\alpha) Mean_{t-1}$$

$$Var_t = \alpha (X_t - Mean_t)^2 + (1-\alpha) Var_{t-1}$$

When computing the daily average, we exclude the case where the corresponding attribute is zero on day t – for example, when the user is inactive on the day.

If at any time, the daily cumulative counts reach or exceed some pre-calculated threshold for the attribute, we will assign a threat score to the access sessions involved. The thresholds are determined as follows.

Since the login IDs that have a high number of daily sessions (i.e., the highly active accounts), exhibit low variability as shown in Section 3.3, we set the thresholds in the same way as anomaly detection in Gaussian random variables: Threshold $= Mean + N_4 \times Var^{1/2}$. For the login IDs that have a moderate number of daily sessions, we set the threshold to be the product of a constant factor and the mean value: Threshold $= Mean \times N_5$. This is based on the observation that the coefficients of variation are bound by a small constant. And finally for the large portion of login IDs that only access the network occasionally, we set the threshold to be a small constant: Threshold $= N_6$. Once an attribute exceeds the defined threshold, a threat score is assigned. The value (weight) of the threat score is set according to a sublinear function of the corresponding attribute (the daily cumulative count) value.

4.4 Aggregation of threat scores

Using the rules above, in the course of a day, we maintain an updated set of AAA log entries that are assigned non-zero threat scores. Those AAA log entries are tagged with information on the rule of violated to aid further examination by network security operators. Suspicious log entries can be noise (i.e., triggered by abnormal activites of low interest to security) – for example, an operator starting to use a new set of commands can trigger a violation detection by the user-keyword-rule. To reduce the chance that a network security operator has to investigate a non-critical violation, we further aggregate these log entries by login ID and by originator IP. The idea is that real attackers may be caught by multiple rules and by aggregating the threat scores on a per login ID or per originator IP basis they can be further distinguished from non-critical anomalies.

To achieve this, we use a moving window of T_3 (e.g., 1 day), and sum up the threat score within the window for all login IDs and originator IPs. We then set a threshold N_7 based on historical data. When we observe an aggregate threat score exceeding N_7, we generate an alarm to the network security operators. We also display all suspicious activities on a dashboard report from which network security operators can pull information on demand.

5 Evaluation

We evaluate our system from two perspectives – the rate of anomalies detected from day-to-day network management activities and the effectiveness of detecting artificially injected anomalous activities. The former quantifies the resources required to investigate potential misuses and intrusions. The latter quantifies the chance that an anomaly goes undetected by our system.

5.1 Running system performance

Figure 16 shows the distribution of the aggregate threat score in a month using two types of aggregations – by login ID (solid line) and by originator IP address (dashed line). We observe that about 91% of login IDs and 82% of originator IPs pass the system without raising any threat score. Meanwhile, there exist a small number of cases in which the system reports a high threat score. By manual inspection, we find most of them correspond to unusual network changes such as a newly deployed network management center or a major software upgrade on an existing network

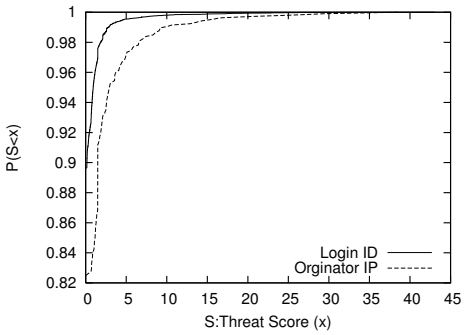

Figure 16: CDF of aggregate threat score by login ID in one month

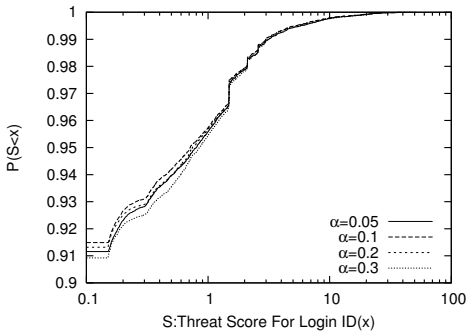

Figure 17: CDF of aggregate threat score by login ID in one month w.r.t. different α values

management tool. We will see an example of this in Section 5.3. We also find that most of the low threat scores (e.g., less than 3) correspond to a small number of log entries from either a new or a very infrequently used account. Profiles for such accounts are difficult to construct based on history, and they do not generate many activities to drive the threat score high.

Figure 16 defines the tradeoff curves between the alarming rate and the sensitivity to anomalous activities. Raising the alarm threshold (the N_6 in Section 4.4), reduces the number of cases that security operators have to investigate but also reduces the chance of catching a stealth intrusion. For a concrete example, setting N_6 to 5 would produce a few alarms per day on average, which is quite manageable for the network security operators.

We note that the above N_6 and several others as described in Section 4 are parameters used in the system. We do not present the exact values for the parameters in our running system due to security considerations. Instead, we show through an example our reasoning on parameter selection. Figure 17 shows the solid line in Figure 16 with a varying α value used in the EWMA estimate. Note the x-axis is in log scale. Different α values effectively factor in different amounts of history data. Setting $\alpha = 0.05$ effectively ignores (e.g., weight less than 0.01) data more than 90 active days old while $\alpha = 0.3$ effectively ignores data more than 13 active days old. However, Figure 17 shows there is little difference in the threat scores among the four different α values – indicating that a short history is sufficient for the system.

5.2 Controlled experiment

Using Figure 16 as a reference point, we design a controlled experiment as follows. We first randomly select 50 pairs of non-overlapping login IDs. We then take one day's worth of AAA logs from our running system and substitute the login ID field in the log entries such that the two login IDs in each of the chosen pairs are switched. Finally, we feed the manipulated AAA logs into our running system and monitor the output.

Figure 18 presents the CDF curves of the aggregate threat score based on the original AAA logs (dashed line) and the synthetic data (solid line) respectively. We find that our system is able to detect many of the behavior changes introduced by the login ID swapping. 72 out of the 100 login IDs report non-zero threat score and among them, 30 login IDs have a threat score higher than 10. Compared to the baseline threat score distribution from the original logs, the result using the synthetic data stands out significantly.

For a closer look at how our system detects anomalous behavior changes, Figure 19 and Figure 20 compare the contribution to the threat score from rules based on access pattern profiles with the

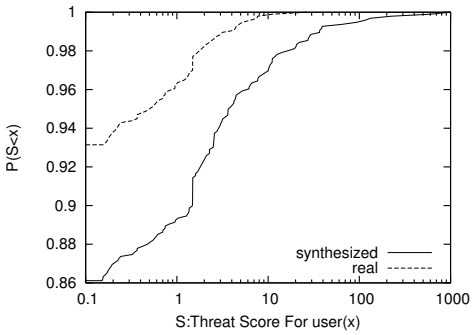

Figure 18: CDF of aggregate threat score on real data and synthesized data

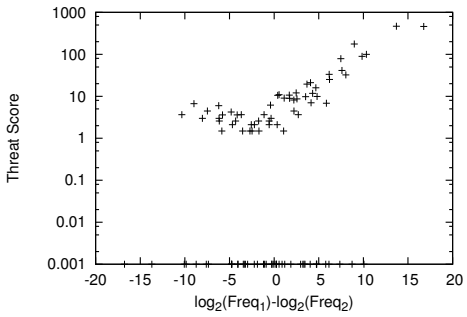

Figure 19: Threat score by rules based on access pattern profiles

contribution from rules based on statistical models of the access profile respectively. For each of the 100 login IDs, we plot their threat score against the difference in the daily average access frequency to the substituting login ID in the AAA logs. For example, a login ID *abc*, with 16 sessions per day on average, which is replaced by login id *xyz*, with an average of 1024 sessions per day, would have its threat score plotted at 6 (i.e. $\log(1024) - \log(16)$) on the x-axis. We observe that both the access pattern changes and the access statistics changes have contributed to the high threat score. The higher the difference in the amount of access activities between the pair of swapped login IDs, the higher the resulting threat scores – with the exception in Figure 20 on the negative side of the x-axis. The exception arises because the rules based on the statistical models are one-sided, i.e., we do not alert on a "busy" user suddenly becoming less active, as this behavior change does not seem to pose any security threat.

5.3 A case study

We now look at an example in which our system alerts with a high threat score. Figure 21 plots the aggregate thread score of a particular login ID over the course of four days. The login ID is used by a software tool that periodically initiates `ping` commands among the various provider edge (PE) routers of the VPN customers to monitor their VPN health.

Starting in the afternoon of day 2 of the plot, we observe a fast increase in the aggregate threat score by the login ID. In less than two hours, the threat score passed the 99.5% alarming threshold and kept rising. It turns out that the software tool was upgraded that day and the new control sessions include a `show version` command that collects the router OS version across the network – similar to what might occur if an intruder attempted to collect information as preparation for attacks. After validating the change of behavior due to software upgrade, we include the pattern change in the profile update at the end of day, which greatly reduces the threat score on day 3. The corresponding statistical models are further updated at the end of day 3 and the new pattern then gets fully captured by the profiles. Hence, there is no more threat score on day 4.

6 Related Work

Our work falls into the area of IDS (Intrusion Detection Systems) in computer and networking security, which dates back to 1980 when Anderson [4] first proposed a computer security surveillance system. Over time, the research area has become more active as the Internet grew in scale and

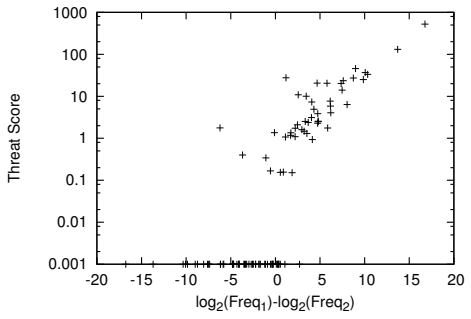

Figure 20: Threat score by rules based on statistical models of the access profile

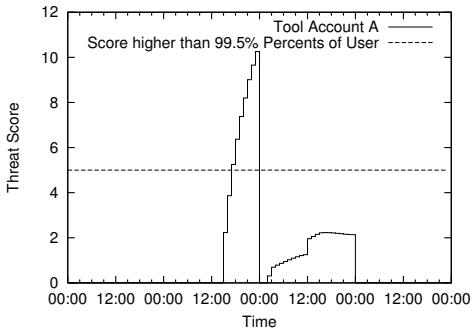

Figure 21: Threat score For Tool account A

application diversity and new security threats constantly emerged.

IDS broadly divide into two categories: host-based (HIDS) [5,6,7,8,9] and network-based (NIDS) [10, 11, 12, 13] — HIDS typically rely on information about running processes to catch intrusions to computer host(s); NIDS typically analyze network traffic in order to detect attacks. Another taxonomy of IDS is based on detection principles [14]: anomaly-based IDS (AIDS) [5,6,7,8,9,11,12] capture anomalous traffic or processes based on analysis of normal patterns. Signature-based IDS (SIDS) [10, 13, 15, 16] use known signatures of attacks to alert on viral activities. Our work aligns with AIDS in principle.

Masquerader detection is a branch of IDS. A masquerader is an attacker who obtains a user's password, penetrates the access control system and impersonates a legitimate user. Lunt et al [5] designed IDES as the first IDS handling masquerader detection, using a simple yet effective statistical model. Recently, different machine learning techniques such as Genetic Algorithm [7], Naive Bayesian classification [8], Support Vector Machine [9] have been applied in this area. In this study, we build user behavior models from access and command invocation patterns using statistical methods and alert based on deviation from the model. It remains as future work to evaluate whether more sophisticated machine learning algorithms can improve sensitivity and accuracy in our problem setting.

7 Conclusion

In this paper, we have studied the problem of protecting the networking infrastructure and the information available therein for large scale enterprise or ISP networks. We have proposed to enhance existing security measures with an intrusion detection system overseeing all network management activities. By analyzing device access logs collected via AAA systems, particularly TACACS+, in a global tier-1 ISP network, we have gained a tremendous insight on the features that distinguish normal operational activities from rogue/anomalous ones. We have further developed a real-time intrusion detection system that builds statistical models to profile normal operational activities and alerts in real-time on any deviation from the profiles. Our evaluation demonstrates that this system effectively identifies potential intrusions and misuses with an acceptable overall alarm rate.

For future work, we would like to explore using more sophisticated machine learning techniques in additional statistical methods to capture anomalous network access activities. We are also interested in further introducing automated mitigation control based on detected anomalies to the AAA system such that an attack or intrusion can be stopped as early as possible.

References

[1] L. G. D. Carrel, "The TACACS+ protocol," http://tools.ietf.org/html/draft-grant-tacacs-02, January 1997.

[2] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote authentication dial in user service (radius)," United States, 2000.

[3] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: A survey," in *Proceedings of the 21st Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 463–472. [Online]. Available: http://dl.acm.org/citation.cfm?id=1106778.1106849

[4] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report James P Anderson Co Fort Washington Pa*, p. 56, 1980. [Online]. Available: http://www.citeulike.org/user/animeshp/article/592588

[5] T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, H. S. Javitz, A. Valdes, T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, H. S. Javitz, and A. Valdes, "Ides: The enhanced prototype - a real-time intrusion-detection expert system," SRI International, 333 Ravenswood Avenue, Menlo Park, Tech. Rep., 1988.

[6] F. Maggi, M. Matteucci, and S. Zanero, "Detecting intrusions through system call sequence and argument analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, pp. 381–395, 2010.

[7] J. A. Iglesias, A. Ledezma, and A. Sanchis, "Creating user profiles from a command-line interface: A statistical approach," in *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: formerly UM and AH*, ser. UMAP '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 90–101. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02247-0_11

[8] R. Maxion, "Masquerade detection using enriched command lines," *2003 International Conference on Dependable Systems and Networks, 2003. Proceedings.*, no. June, pp. 5–14, 2003.

[9] M. B. Salem and S. J. Stolfo, "A comparison of one-class bag-of-words user behavior modeling techniques for masquerade detection," *Security and Communication Networks*, pp. n/a–n/a, 2011. [Online]. Available: http://dx.doi.org/10.1002/sec.311

[10] Z. Li, G. Xia, H. Gao, Y. Tang, Y. Chen, B. Liu, J. Jiang, and Y. Lv, "Netshield: massive semantics-based vulnerability signature matching for high-speed networks," in *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 279–290. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851216

[11] Y. Song, A. D. Keromytis, and S. J. Stolfo, "Spectrogram: A mixture-of-markov-chains model for anomaly detection in web traffic," in *NDSS*. The Internet Society, 2009.

[12] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: methods, evaluation, and applications," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, ser. IMC '03. New York, NY, USA: ACM, 2003, pp. 234–247. [Online]. Available: http://doi.acm.org/10.1145/948205.948236

[13] H. Dreger, A. Feldmann, M. Mai, V. Paxson, and R. Sommer, "Dynamic application-layer protocol analysis for network intrusion detection," in *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*. Berkeley, CA, USA: USENIX Association, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1267336.1267354

[14] A. Stefan, "Intrusion detection systems : A survey and taxonomy," *Technical Report*, vol. 99, no. Technical report 99-15, pp. 1–15, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.6603&rep=rep1&type=pdf

[15] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Comput. Netw.*, vol. 31, pp. 2435–2463, December 1999. [Online]. Available: http://portal.acm.org/citation.cfm?id=337967.337972

[16] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX conference on System administration*, ser. LISA '99. Berkeley, CA, USA: USENIX Association, 1999, pp. 229–238. [Online]. Available: http://portal.acm.org/citation.cfm?id=1039834.1039864