Implementation of an Intrusion Detection System (IDS) Based on Statistical Access Pattern Analysis [*]

Onur Kocak
Bilkent University
onur.kocak@bilkent.edu.tr

Omer Faruk Aktulum
Bilkent University
omer.aktulum@bilkent.edu.tr

ABSTRACT

Network layer access control is one of the most important aspects of security in an enterprise network architecture where multiple servers and services are available. Firewall devices are the main components of the network for access control of traffic to these servers and services from internal and external sources. Most of the devices provide a passive protection against different attack models. In real world scenarios, a new system is necessary to provide proactive protection to increase the security level. In this study, we propose a method as an Intrusion Detection System (IDS) which analyses statistical access patterns in a proactive manner. Our proposed method addresses the major weaknesses of passive firewall devices and provides better security against advanced DDoS attacks with real time proactive control.

Keywords

Intrusion prevention system, intrusion detection system, DDoS attacks, access pattern analysis.

1. INTRODUCTION

Enterprise network infrastructures have multiple layers of network devices, servers and services. These devices can create data flow on the network. For security purposes, network data flow is monitored through firewall devices which filter external and internal traffic based on preset rules. The source and destination addresses of TCP and UDP packets, size and number of the packets forms a pattern. On a typical enterprise network, traffic flow among the servers generally shows a regular pattern. Scheduled tasks such as backup operations, cron jobs, NTP updates, log rotation and archiving, file synchronization and time dependant business operations may cause the network traffic to increase and decrease on regular base.

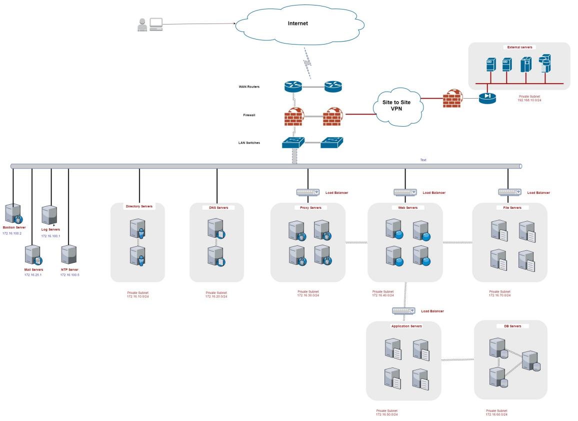

Figure 1: Infrastructure Diagram of an example Enterprise Network

Figure 1 shows a typical enterprise network infrastructure diagram which is designed to serve a secure and high available service to its clients. In the remaining part, we go into more detail to summarize the structure of the diagram. The transactions coming from the Internet are filtered on firewalls and transferred to internal network based on the routings and NAT rules. Depending on the design, there might be multiple layers of servers to respond and process transactions. Figure 1 represents an example of an enterprise network infrastructure which has proxy server, web server, application server and database server that are clustered for core operations. There are also supporting servers such as NTP server, log server, Bastion host, etc.

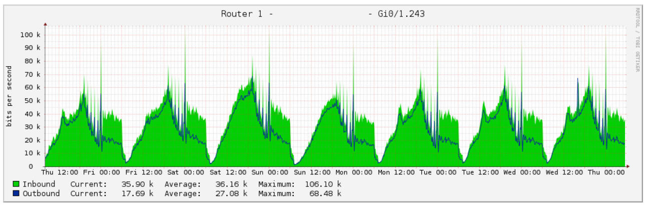

Figure 2: Multi Router Traffic Grapher (MRTG) graph for 1 week span of particular link

The flow of a network traffic, which is taken from a monitoring system belongs to a company, is illustrated in Figure 2. As you may realized, the network traffic and number of transactions follow a regular pattern as discussed above.

[*]This is a midterm report for CS577 Data Privacy course project in Spring 2017 at Bilkent University.

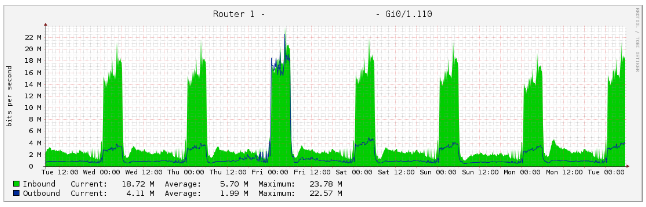

Figure 3: MRTG graph for 1 week span of particular ISP link with a typical anomaly

Figure 3 shows an typical traffic anomaly on the network that can be caused by a DDoS attack. As it may seen, on Friday there is an unusual increase on the traffic. By analyzing the number of valid and invalid sessions, TCP packets and application logs, the increase in the traffic can be identified as safe traffic or a possible DDoS attack. Enterprise systems should be robust and reliable against the attacks. Attackers might try various ways depending on their capabilities and knowledge on the target system. In many cases, infrastructures have security holes caused by misconfiguration of hardware or bugs in software. Due to these reasons, the best way of securing the infractures is to have a proactive access control system that will block the unwanted network traffic in realtime regardless of the weaknesses of the system. This is possible by analyzing the network behavior and traffic with proper methods and implementing a self-developing and proactive Intrusion Prevention and Detection System (IPS and IDS) that is integrated within a firewall device. In this project, we are going to propose and implement an IPS system based on statistical access pattern analysis on firewall traffic log files.

2. PROBLEM STATEMENT AND LITERATURE SEARCH

According to Akamai's Quarterly State of Internet - Security Report of 2016 [1], there has been a 125 percent increase in distributed denial of service (DDoS) attacks year over year. To protect against the DDoS attacks, the network traffic should be monitored actively and unknown and unwanted traffic should be dropped before it reaches the servers. Firewall devices are capable of filtering the traffic based on the predefined rules. However, as it may estimated, attackers have various methods to scan and find the publicly open ports and services on the target system. Therefore, having a static filter does not actually provide a caution for securing the system against the DDoS attacks. Moreover, if a firewall device if misconfigured, for instance an unnecessary port left open, attacker can easily start a massive TCP SYN flood attack to make the service unavailable. A better approach might be to monitor and analyze the network traffic and create an autonomous and proactive security framework to detect and prevent the unwanted traffic without an interaction by the system users. Based on the severity of traffic volume, notifying the system users should also be desired.

There exists many work about anomaly based intrusion detection systems. Most of them proposed various data mining techniques to learn the network behaviour. In [2], an incremental learnable model for detecting anomalies and preventing the Zero-day attacks is proposed. They have introduced LMAD/PZ algorithm where they integrate the intrusion detection with intrusion prevention plan. By training clustering algorithms, they distinguished the traffic as normal or anomalous. They have suggested Algorithm Quasi-optimal (AQ) and Cobweb which is an unsupervised hierarchical conceptual clustering algorithm. LMAD/PZ performed well in the experiment for detecting the Zero-day attacks, and they outperformed the Cobweb clustering algorithm in terms of accuracy, detection rate and false alarm rate.

In intrusion attacks, attacker aims to break through the system and take control by creating malicious TCP packets. However. in DDoS attacks, attackers aim to make the service unavailable without intrusion. In a typical DDoS attack, since packets are valid in term of content, IPS systems can not detect an anomaly. To fill this gap, Beitollahi and Deconnick [3] suggest a method called ConnectionScore which is a statistical technique to protect from most common application layer DDoS attacks such as HTTP flood [5]. They scores the traffic based on the statistical data, which includes the browsing behaviour, connection time and status of the resources on the system. With the introduced method, 24% of legitimate connections get negative scores.

Pathan deeply approaches to intrusion detection and prevention systems in his book [4]. In Chapter 2, he covers methods for intrusion detection such as pattern matching, protocol analysis, statistical and probabilistic approach and neural networks. Detection methods have been introduced in Chapter 13, that are signature based intrusion detection, anomaly based intrusion detection and specification based intrusion detection. Applications of machine learning is also covered in this section. However, he claims that traditional supervised machine learning approaches usually require a large number of labeled data in the phase of training whereas in real-world applications the number of available labeled data is very small. Therefore, we decided on to continue with a statistical analysis model in our project. We will refer to this book in the next phases of our project.

In our project we will implement a method that is similar but not limited to ConnectionScore. We are going to analyze the historic data based on the source IP distribution, number of sessions and port variance.

3. PLANNED METHODS

Our project plan contains following tasks to be done. Firstly, the logs will be gathered from a real company and sanitized and parsed for further analysis. Then, we are going to implement some scripts to preprocess large sized log data. In this phase, the database tables will be populated with the traffic patterns and counts. After the preprocessing of log data completed, we need to develop a statistical method for identifying the secure and insecure traffic patterns. Our current idea is to have a policy table in our database for known traffic patterns that are exactly known as secure or insecure. The rest of the traffic can be analyzed based on the historic data that we gathered. After we designed our statistical method, we are going to implement a basic web interface for management of our IPS system. The web interface should allow us to view identified traffic patterns, edit the policies and report events. Finally, a simulation tool will be implemented to test some known scenarios and attack types and what percent of the malicious packets would be dropped will be reported.

4. IMPLEMENTATION

In this phase, 1 week of firewall logs belongs to a payment gateway company with 400K POS (Point of Sale) devices, have been gathered. Daily 35 million transactions are processed and forwarded to upstream servers of banking companies by the company.

Our first aim is to analyze the traffic by counting network packets in different source and destination addresses in 10 minutes resolution for each day and create a statistical model for the normal behaviour of network traffic. We completed two main preprocessing implementations that we planned for the midterm report at the beginning of the project.

In the first phase, the log data is sanitized and parsed by implementing AWK and PHP scripts, and extracted the necessary information which will be used in analysis of regular flow of network traffic. As it may estimated, a log file includes many lines with many features/columns which are separated by comma. In this project, we are interested in only nine of those features as date, time, source ip, source interface, destination ip, destination port, destination interface, action and service. After the preprocessing phase, each line of the log file includes the features/columns as in following sample log;

```
date=2017-02-11,time=22:51:18,srcip=172.31.31.17,
srcintf=”172.31.31.16/28”,dstip=172.31.32.10,
dstport=8000,dstintf=”NW_172.31.32.0”,
action=close,service=”TCP_8000”
```

In the second phase, the preprocessed log data was populated into a database to execute necessary queries to analyse the regular network traffic by implementing a script in Python, and upload the data into a MySQL database. In Python script, we scan through the large files of the preprocessed data, parse it and upload the data into the database by executing SQL queries.

In the database design of the project shown in Figure 4, currently, there exist traffic, log, reputation and policy tables. traffic table stores the identified traffic patterns which are formed by the Source IP, Destination IP, Destination Port and Service (Protocol) of network packets. Each traffic pattern constitutes a composite unique index linked with an integer primary key. In the log table, the number of packets for each traffic pattern is the historic log data stored in 10 minutes resolution in column wise. The policy table is created for exactly and explicitly known to be secure or insecure traffic patterns . This table will contain ALLOW or DENY, rules pre-defined by the network administrator for known cases. We are going to keep track of IP addresses and their trust score in reputation table in real time.

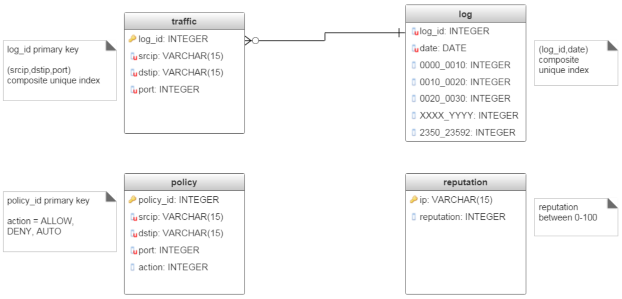

Figure 4: ER diagram of designed database

5. NEXT STEPS

We are going to develop a statistical model for analysing existing logs. Up to now, 1 week of logs belongs to a real production environment have been preprocessed in 10 minutes resolution and database tables populated based on traffic patterns. Next, we will use a statistical model to detect a given network packet might be secure or not. Our current idea is to analyze the traffic by the time range, distribution of source IPs, distribution of the destination IPs, port variance, number of sessions and reputation of IP addresses. Then, we are going to define percentage threshold limits. We will keep track of reputation of hosts and traffic patterns on run time. By analyzing the historic data in a given time range, we can drop or allow the packet by also checking the predefined rules in the policy table. By this technique, we are going to prevent methods like port scanning and flooding attacks. A basic web interface for managing the system will also be developed. The web interface should allow us to view the traffic patterns, counts, view events and create and edit the policies.

Finally, a simulation tool that basically generates random traffic towards our system needs to be developed. The simulation tool will provide us a test environment for the project. In the simulation tool, we will try to implement well known DDoS attack methods and intrusion types.

6. EXPECTED OUTPUT

Our simulation tool should initiate safe and malicious traffic towards our system based on given IP list, pattern, time range and volumes. We expect our system to detect the malicious traffic and create DENY rules for them in order to keep the internal network safe. In attack scenarios, we will focus on port scanning, TCP flood, UDP flood. To measure the success rate of our system, we are going to calculate the DENY percentage of the number of packets coming from malicious sources.

7. REFERENCES

[1] State of the Internet / Security Report. (n.d.). Retrieved March 15, 2017, from `https://www.akamai.com/us/en/our-thinking/state-of-the-internet-report/`

[2] Abdurrahman A., N., Mohamed M., E., & Mohamed Z., A. (2016). An Intrusion Detection and Prevention System based on Automatic Learning of Traffic Anomalies. International Journal Of Computer

Network And Information Security , Vol 8, Iss 1, Pp 53-60 (2016), (1), 53. doi:10.5815/ijcnis.2016.01.07

[3] Beitollahi, H., & Deconinck, G. (n.d). ConnectionScore: a statistical technique to resist application layer DDoS attacks. Journal Of Ambient Intelligence And Humanized Computing, 5(3), 425-442.

[4] Pathan, A. K. (2014). The State of the Art in Intrusion Prevention and Detection. Boca Raton: Auerbach Publications.

[5] T.Yatagai, Isohara, T., Sasase, I.: Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior. In: Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 232 - 235 (2007)