CS658A: Topics In Malware Analysis & Intrusion Detection

Team: Four Unknown Signatures

Project Title :
Network Attack Detection and Prevention using TCPdump, Grafana and Prometheus

Team Members

- Punit Chaudhari - 21111050 - punitac21@iitk.ac.in
- Akshay Kumar Chittora - 21111007 - akshay21@iitk.ac.in
- Jeet Sarangi - 21111032 - vishwasc@iitk.ac.in
- Piyush Gangle - 21111046 - piyushg21@iitk.ac.in

1 Abstract

An Intrusion Detection System (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. It is a software application that scans a network or a system for the harmful activity or policy breaching. Any malicious venture or violation is normally reported either to an administrator or collected centrally using a security information and event management system.
In this project we have built an intrusion detection, visualisation and resolution tool that can detect intrusion attacks in real time and can resolve them using some predefined rules.We have used multiple open source tools in this in order to monitor the network traffic and detect if any thing goes wrong.

2 Problem Statement

With recent advances in network based technology and increased dependability of our every day life on this technology, assuring reliable operation of network based systems is very important.During recent years, number of attacks on networks has dramatically increased and consequently interest in network intrusion detection has increased also the area of network traffic visualisation is of great importance since for monitoring of packets in a network can solve many of the issues in field of security.
In this project we have implemented a system for network packets monitoring and visualisation which will help in detection of common attacks related to network traffic and intrusion.

3 Introduction

The Goal of this project to detect intrusions like DDOS, TCP SYN flooding, ARP spoofing, Smurf DOS, Nmap and others and to take actions according to some predefined rules so that systems can be made intrusion free and proper responses can be given to the clients to function properly.
We have used various open source softwares tools to monitor the traffic in a network and propose possible solution to the clients in order to prevent these kinds of attacks.

4 Data Collection

The network packets data is collected using TCPdump tool which takes some formatting in order to model the data so that it can be used as useful information in monitoring network packets.
All local machines will have the tool install on there system and the server will get the data in regular interval based on a pre-defined time interval.The server

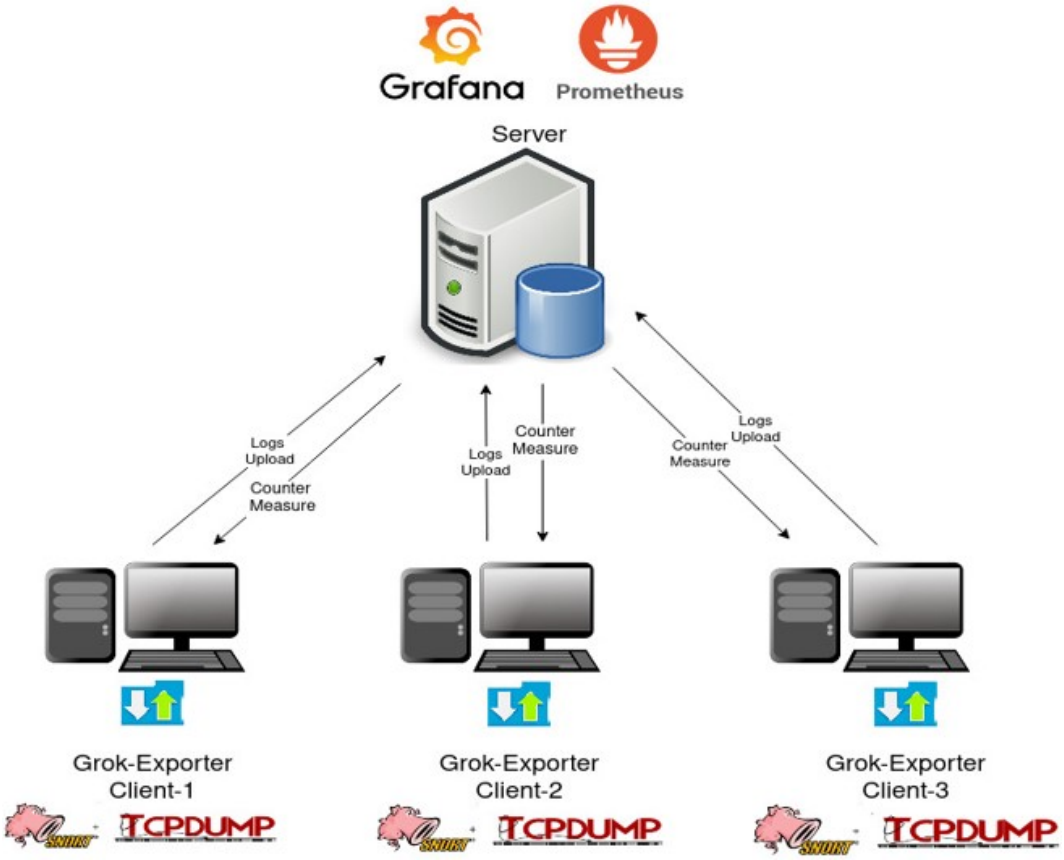
will ping the client machines to send data collected so far and each will then reply with the dump collected by that time.

5 Motivation

In traditional intrusion detection system (IDS) environments, little activity has been applied to usingvisual analysis as an aid to intrusion detection. With more information systems being attacked and attacktechniques evolving, the task of detecting intrusions is becoming an increasingly difficult. Efficientinformation visualization is an important element required for urgent detection of intruders. The Goal of this project was to create a useful data visualising system to monitor the network and detect attacks efficiently using visual techniques.

6 Architecture

System broadview:



6.1 Client

For the client machines Snort and tcpdump will be used as a packet sniffer.

6.1.1 Snort

- foremost Open Source Intrusion Prevention System (IPS),snort can be used as a packet sniffing tool as it can be configured to catch various network activities.

- Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users.

6.1.2 Tcpdump

- We have also used Tcpdump as a packet sniffer which is a data-network packet analyzer tool that allows the user to sniff and anlyse the TCP/IP and other packets being transmitted or received over a network to the client machine on which it is placed.

- Snort and tcpdump outputs are processed using bash scripts using tools like **sed, awp, grep** that will save the output in a file named netlog.txt and we will use that file to process further and keep useful data for us in a particular format.So that the information we want for analysing network vulnerabilities can easily be implemented.

  Log file before preprocess:

```
IP 18.66.63.7.443 > 172.23.130.166.38944: Flags [P.],
IP 172.23.130.166.38944 > 18.66.63.7.443: Flags [.],
IP 172.20.160.23.53 > 172.23.130.166.44444: 49862 1/1/1
IP 172.23.130.166.35783 > 172.20.160.23.53: 17111+ [1au]
IP 172.20.160.23.53 > 172.23.130.166.35783: 17111 0/1/1
IP 172.20.160.23.53 > 172.23.130.166.50535: 44235 5/0/1
IP 172.23.130.166.50425 > 172.20.160.23.53: 45562+ [1au]
IP 172.20.160.23.53 > 172.23.130.166.50425: 45562 0/1/1
IP 172.23.130.166.53840 > 13.224.16.126.80: Flags [S],
IP 13.224.16.126.80 > 172.23.130.166.53840: Flags [S.],
IP 172.23.130.166.53840 > 13.224.16.126.80: Flags [.],
IP 172.23.130.166.53840 > 13.224.16.126.80: Flags [P.],
IP 13.224.16.126.80 > 172.23.130.166.53840: Flags [.],
IP 13.224.16.126.80 > 172.23.130.166.53840: Flags [P.],
IP 172.23.130.166.53840 > 13.224.16.126.80: Flags [.],
IP 172.23.130.166.53840 > 13.224.16.126.80: Flags [F.],
IP 13.224.16.126.80 > 172.23.130.166.53840: Flags [F.],
IP 172.23.130.166.53840 > 13.224.16.126.80: Flags [.],
```

  Preprocess script:

```
tcpdump -nn arp or icmp or udp or tcp | awk '{print(NR,$2,$3,$4,$5,$6,$7)}'
```

6.1.3 Grok exporter

Grok exporter is used as a tool to parse this log data into something structured and queryable form. The Grok exporter here take this netlog.txt file and process it using it config file, The config file of the Grok is written such that it extracts out tcp,udp,icmp and arp packets requests and responses from all the machines connected to this computer. Used regex to match the queries and this information is then forwarded to the server machie for further processing.

The Grok exporter is also installed on all the client machines in order to keep creating the dumps of tcp,icmp, arp and udp packets based on the configuration provided to a buffer.

After the dump is transmitted or sent to the server the dump buffer is then cleared and again fresh dump is created for next time.

6.2 Server

The server machines will display all the useful information using a visualization tool and will alert and send appropriate response to the clients regarding the malicious activities in the system if any.
For the server machine prometheus is used for all data collection from the client machine to the server in the specific formats.

And for visualisation Grafana is used as a tool for showing the admin all the network activities going on in the network.These tools together work on the server for doing some visual monitoring on the network traffic and also to do some rule based or condition based fixed protocol implementation.

6.2.1 Prometheus

Prometheus is the open source data collection software used for event monitoring and alerting.It records real-time metrics in a time series database (allowing for high dimensionality) built using a HTTP pull model, with flexible queries and real-time alerting.
We have integrated prometheus into our server and configured it using a yml file, the configuration file will be saved with configuration info for the prometheus like what are the targets machines and what will be the scrape interval, the prometheus will ask for the data from the clients and the Grok exporter on clients will send the infomation to promteheus based on the fixed interval.
We can query that data using the Prometheus query language and extract out relevant information from that. This information that is extracted from prometheus using queries will work as input for visualisation and also for using it as input to rule based protocol implementer.

- Prometheus has four kind metrics (Counter, Gauge, Histogram, Summary) to collect various kind of information from log file

- Counter: is used to count number network packets satisfying certain rule(example: count tcp packets, count icmp packets)
- Gauge: is used to store numeric value for a certain attribute(ex. cpu utilization of a instance).

Counter Metric

```
- type: counter
  name: prot_tcp
  help: Destination IP Addresses
  match: '%{INT:ts} IP %{IP:sip}\.%{INT:sp} > %{IP:dip}\.%{INT:dp}\: Flags.*'
  labels:
    ts: '{{.ts}}'
    sip: '{{.sip}}'
    sp: '{{.sp}}'
    dip: '{{.dip}}'
    dp: '{{.dp}}'
```

Gauge-Metric

```
- type: gauge
  name: cpu_usage
  help: CPU utilization
  match: '%{INT:ts} IP %{IP:instance} > %{INT:cpu}.*'
  labels:
    ts: '{{.ts}}'
    instance: '{{.instance}}'
  value: '{{.cpu}}'
```

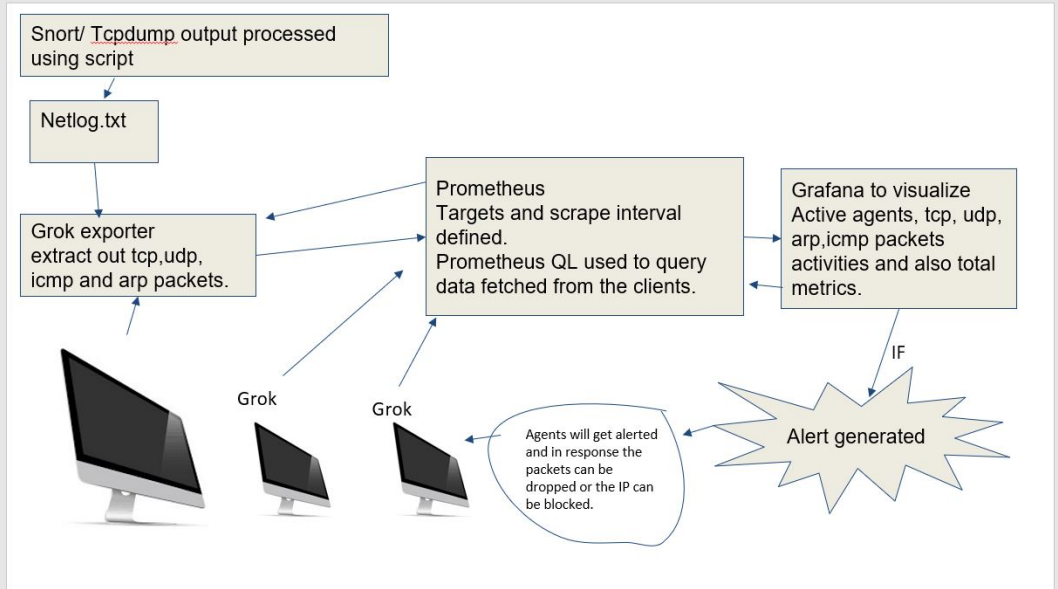
6.2.2 Grafana

Grafana is a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources. Visualization will be done from the data collected by the Prometheus using Grafana on web based interface. We are doing the following visualization on the data.

- Active Agents: It will show the number of online agents connected to this server.
- Memory and CPU utilization: It will be showing the Cpu utilization and the memory utilization of each of the machines connected.
- GPU utilization: It will be showing the GPU utilisation of various hosts.

- TCP: This panel will show rate of TCP packets incoming in to a particular host for different host different color scheme is used to differentiate.
- UDP : This panel will show rate of UDP packets incoming in to a particular host for different host different color scheme is used to differentiate.
- ICMP : This panel will show rate of ICMP packets incoming in to a particular host for different host different color scheme is used to differentiate.
- ARP : This panel will show the movement of ARP packets in the network for different hosts different color schemes are used to differentiate.
- Total Packets count:This show total count of packets in whole network for all clients summed as it can help in detection of how packets are flowing in the network.

The below figure explains the architecture in detail:



7 Methodology

- Prometheus server will be connected to all the agents and it will collect logs from them at particular scrap interval(1s).
- Prometheus will build a single data structure which will aggregate all the logs of every agent.
- Grafana Visualization Engine is established at server and it will query the Prometheus with the help of PromQL(Prometheus Query Language) in-order to fetch the records and perform visualization.

- Grafana Alert Notifications are used to generate an alert if any suspicious behaviour is encountered.
- We have set certain thresholds like(tcp-packet count¿1500) based on which alert will be generated.
- Grafana will keep an eye on network packets if certain packet count goes beyond threshold then it will generate and send a alert message in JSON format to Webhook(at agent end).
- Snort and TCPdump will execute on agents, collect network activities and generate logs.
- Grok-exporter will be running on each agent in-order to collect logs and send those to Prometheus server.
- Client at agent side consisting of webhook will be listening at a defined port for Grafana Alert Notifications.
- Counter-measure like(block ip, drop packets ,etc.) will be applied by the client for certain time duration with the help of firewall.

7.1 Method for intrusion detection

For intrusion attack analysis , prevention and visualisation we can create some policies for in order to prevent these:

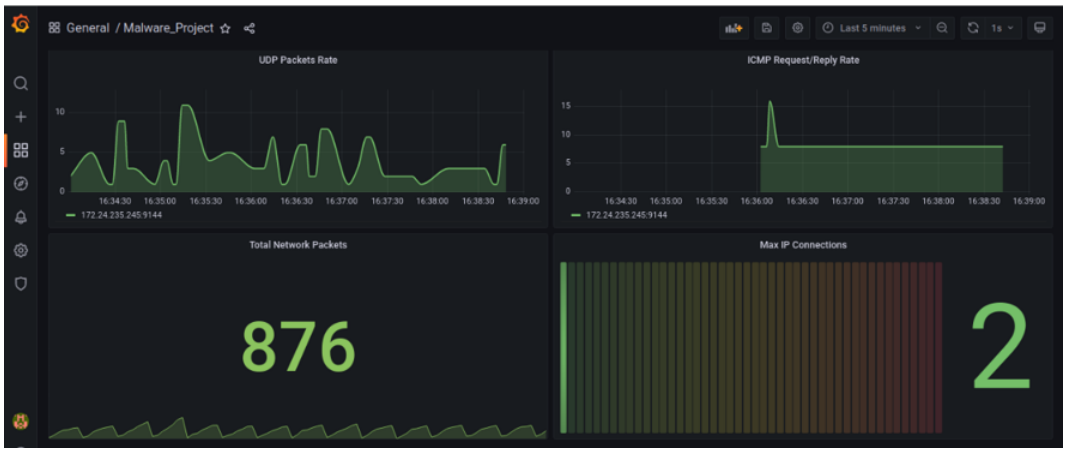
- **TCP Flooding SYN, DDOS and NMap Scan**:TCP packets will show high increase.
- **ARP Spoofing**:ARP packets shoot in number.
- **Smurf Dos**:ICMP packets increase.

Solutions proposed are as follows:

- **TCP Flooding SYN, DDOS and NMap Scan**:TCP packets will show high increase.
- **ARP Spoofing**:ARP packets shoot in number.
- **Smurf Dos**:ICMP packets increase.

8 Results

Grafana Dashboard:



- The proposed system can detect prevent(applying counter measure) network intrusion attacks like DDoS,TCP flooding,ARP Spoofing,Nmap , etc.
- Visual representation of the network state helps the administration to take action against vulnerability quicker.

9 Discussion

In this project we have implemented a system that works on network packets sniffing we got to learn about various tools and techniques which could be used for intrusion detection.
We got insights about various common attacks and there patterns also got insights on how to avoid by implementing proper organisation policies.

10 Future Direction

For the next version of this project following things can be thought of:

- Limited number of protocols are considered(tcp,udp,arp,icmp)simplicity more could be added.
- Machine learning algorithm can be applied to predict unseen threats from data collected.
- Better recovery measures could be added to reply to client on detection of a attack.