BLEKINGE INSTITUTE OF TECHNOLOGY

SSH and Bitrate Monitoring Using DPMI

Software Requirements Specification

Version 1.2

Group – 4

Date:19-10-2017

Team Members:

1.GURRAM KARTHIK

2.HOSSEN SADDAM

3.JANAGAM ANIRUDH

4.KONDEPATI DIVYA NAGA KRISHNA

5.MAMIDI SAI PRAKASH

6.NAMANA SAI KIRAN KUMAR

7.SATHI SANTHOSH REDDY

8.YALAVARTHI SRI LEKHA

9.KARUMANCHI MAHESH

1.INTRODUCTION:

This project describes the SSH (Secure Shell) dictionary attacks detection where we are required to monitor SSH connection attempts to avoid possible dictionary attacks. Dictionary attacks are a type of fraudulent attempts made to compromise a system with SSH connection. This can also involve trying many username and password combinations on the (remote) system to gain access to it. This type of attack can be successfully recognized from analyzing network data. In this subpart, the main objectives will be to configure SSH such that if more attempts than X/Minute, the connection must be blocked for no further attempts for Y Minutes. SSH must be configured in such a way that the blockage time increases every x/minute's attempts made and the increases by 2*Y minutes, 3*Y minutes etc. Suitable IP tables will be configured to block/unblock automatically.

We are also required to monitor bitrates using measurement streams The bit rates are to be stored into Influx database and visualized graphically in Grafana with a sample time of one second.

In DPMI, the measurement points capture traffic between clients, often sent live as measurement streams for consumers. Each MP has set of filters e.g.: only IP traffic. The captured measurement streams are saved as a CAP file. The PCAP streams are extracted from measurement streams using a CAP2PCAP tool from libcap_utlis, in a way readable by us.

2.User requirements:

Identification String	Requirements	Assignee
UR1	Identifying and logging number of SSH connection attempts during a time interval (Ti),	Mahesh, Krishna
UR2	Identify and count the occurrence of unique source IP (SSH originators).	Sai Kiran,Karthik
UR3	Identify the source IP which exceeds count > threshold and not found in the whitelist.	Karthik, Santhosh
UR3(a)	If first occurrence block source IP for X minutes, then unblock.	Santhosh, Karthik
UR3(b)	If ‘n’ occurrence block the source IP for ‘n*X’ minutes, then unblock.	Prakash,Santhosh
UR4	Configure firewall to block and unblock IP’s	Mahesh
UR5	Ability to remove the blocked IP	Prakash,Krishna
UR6	Add instance to calculate bitrate using DPMI consumer filters and store into Influx DB with user specified tag.	Anirudh, Saddam,Lekha
UR7	List instances and show what they calculate.	Sai Kiran,Anirudh
UR8	Kill or remove instance.	Saddam,Anirudh
UR9	Visualize bitrates in Grafana.	Lekha,Prakash
UR10	Restful API to configure or operate system.	Karthik,Krishna,Saddam

3.Test cases:

T1:

Test: To Identify and log number of SSH connection attempts during a time interval (Ti)

Purpose: To be able to identify a total number of ssh connections attempts made from the source Ip address in a suitable time interval like x/minute and to log them.

Requirement: UR1

Test Environment: Libcal_utils needed to be installed as a prerequisite. Capshow command is used to identify the number of ssh connections and log them to a file.

Result:

Comment: Functionality in progress

T2:

Test: To Identify and count the occurrence of unique source IP (SSH originators).

Purpose: To be able to identify and count the number of unique IPs originating from source IP address.

Test Environment: Libcal_utils needed to be installed as a prerequisite. Capshow command is used to identify the number of ssh connections and log them to a file.The trace file can be viewed using the utilities to identify the unique source IP's.

Requirement: UR1, UR2

Result:

Comment: Functionality in progress

T3:

Test: To Identify the source IP which exceeds count > threshold and not found in the whitelist.

Purpose: To be able to identify the source IP which exceeds the connections attempts than the allowed threshold limit and also to identify if they are found in a list of Whitelisted Ip address.

Test Environment: Libcal_utils needed to be installed as a prerequisite. Capshow command is used to identify the number of ssh connections and log them to a file.The trace file can be viewed using the utilities to identify the unique source IP's. Using the firewall script it is able to identify the IPS exceeding the counts than the allotted threshold.

Requirement: UR1, UR2

Result:

Comment: Functionality in progress

T3a:

Test: If first occurrence block source IP for X minutes, then unblock.

Purpose: To be able to identify the first occurrence of source Ip addresses, block them for x minutes and unblock them after the time limit for the block is complete.

Test Environment: Libcal_utils needed to be installed as a prerequisite. Capshow command is used to identify the number of ssh connections and log them to a file.The trace file can be viewed using the utilities to identify the unique source IP's.Using the firewall script it is able to identify the IPS exceeding the counts than the allotted threshold and block for X minutes and unblock after Y minutes.

Requirement: UR1,UR2,UR4

Result:

Comment: Functionality in progress

T3b:

Test: If ‘n’ occurrence block the source IP for ‘n*X’ minutes, then unblock.

Purpose: To be able to identify the next occurrence of source Ip addresses, after the first occurrence, block them for n*X minutes then unblock

Test Environment: Libcal_utils needed to be installed as a prerequisite. Capshow command is used to identify the number of ssh connections and log them to a file.The trace file can be viewed using the utilities to identify the unique source IP’s.Using the firewall script it is able to identify the IPS exceeding the counts than the allotted threshold and block for X minutes and unblock after Y minutes.If the connections attempts continue after the unblock the firewall script will be used to block for n*X minutes and unblock after Y minutes.

Requirement: UR1,UR2,UR3a,UR4

Result:

Comment: Functionality in progress

T4:

Test: Configure firewall

Purpose: To be able to configure firewall to block and unblock IP’s by running the firewall script

Requirement: UR4, UR2

Test Environment: A firewall is configured using a firewall.sh script and suitable rules will be added at the ssh instance.

Result:

Comment: Functionality in progress

T5:

Test: Ability to remove the blocked IP

Purpose: To be able to remove the blocked source IP which exceeds the connections attempts than the allowed threshold limit manually through Rest API or

WebUI.

Requirement: UR3a, UR3b,UR4

Test Environment: A firewall is configured using a firewall.sh script and suitable rules will be added at the ssh instance.Blocked IPs can be removed through rest API.

Result:

Comment: Functionality in progress

T6:

Test: To add an instance to calculate bitrate using DPMI consumer filters and store into Influx DB with user-specified tag.

Purpose: To be able to calculate bitrates using DPMI consumer filters using the measurement streams 01::71 and 01::72 to store these values into influx DB with user-specified tag.

Test Environment: An instance is launched to calculate bitrates and suitable scripts are used to store the bitrates into influx database by running consumer-bitrate in parallel.

Requirement: UR 6

Result:
Comment: Functionality in progress

T7:

Test: List instances and show what they calculate

Purpose: To be able to list instances and also to show what they are calculating

Requirement: UR6, UR7

Test Environment: Suitable instances will be launched in OpenStack for performing ssh connections and for calculating bitrates utilizing those instances.

Result: Success

Comment:

T8:

Test: Kill or remove instance

Purpose: To be able to kill or remove instance

Test Environment: Suitable instances will be launched in OpenStack for performing ssh connections and for calculating bitrates utilizing those instances.Instances can be killed or removed based on the requirement to decrease the load.

Requirement: UR6, UR7, UR8

Result: Success

T9:

Test: Visualize bitrates in Grafana.

Purpose: To be able to visualize bitrates graphically in grafana using the data from influx database

Test Environment: Using the influx database script the bitrates will be stored into influx database and using the rest API the bitrates are visualized in grafana by suitable commands.

Requirement: UR9, UR7, UR6

Result:

Comment: Functionality in progress

T10:

Test: Restful API to configure or operating system.

Purpose: To be able to configure Restful API in such a way to control the entire system from the command line interface.

Requirement: UR3A,UR3B,UR9,UR10

Test Environment: Using the command line rest API will be tested to perform various operations like To show streams, Delete streams etc.

Result:

Comment: Functionality in progress