Flipkart GRID 3.0

Problem Statement: Asset Discovery Tool

Team Name : HACKIITG

Institute Name: Indian Institute of Technology (IIT), Guwahati

Team members details

Team Name	HACKIITG		
Institute Name	Indian Institute of Technology (IIT), Guwahati		
Team Members >	**1 (Leader)**	**2**	**3**
Name	SHUBHAM KR GUPTA	SATYENDRA DHAKA	UJJWAL RUSTAGI
Batch	2022	2022	2022

Deliverables/Expectations for Level 2 (Idea Submission)

An algorithm/approach with block diagrams and detailed explanation to accomplish the below:

Commands being used with Screenshots to be included in the report. A detailed explanation of what the command is meant to do with explaining all the options/switches of tool.

Running the scanning tool on a remote network would need formation of an SSH tunnel from the ground zero system. The scanning tool needs not to be installed on remote system, rather scan has to be initiated from ground zero system and the scan probes are to be tunneled through SSH tunnel and scan will be performed on the remote network. This will simulate as if the remote system (running SSH server) has initiated the scan. Scan results will be tunnelled back to ground zero system and will be written to a database as per choice below-

- ELK can be used to record the results, provided there is enough scope of correlation of records related to a particular asset. Correlation method also is to be documented.
- In case, a Database/ file is to be used for storing the results, a frontend GUI for interacting with records is also to be mentioned and functioning to be documented.

NOTE:Please provide definitions and working of the core components of the system. You can give references if any part of work is inspired by some previous work. - The solution should work for both a returning user as well as a new user. Considerations of different settings, edge cases will be given extra points.

Glossary

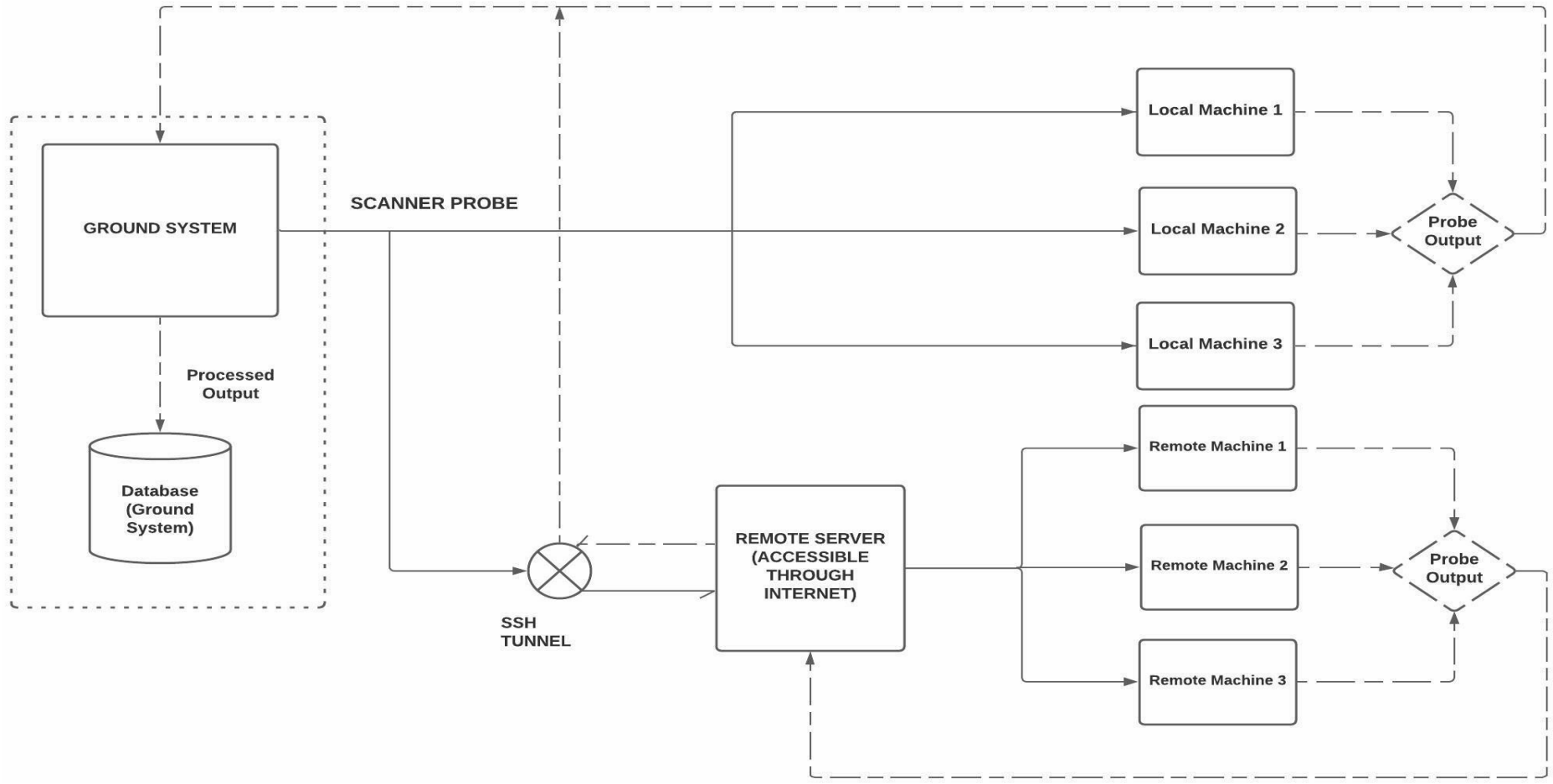
- **NAT** : Network Address Translation
- **ARP** : Address Resolution Protocol
- **NMAP** : Network Mapper
- **DNS** : Domain Name System
- **ELK** : Elasticsearch, Logstash, and Kibana
- **OS** : Operating System
- **IP** : Internet Protocol
- **SSH** : Secure Shell
- **TCP** : Transmission Control Protocol
- **UDP** : User Datagram Protocol
- **ICMP** : Internet Control Message Protocol
- **PING** : Packet InterNet Groper
- **ETH** : Ethernet

Use-cases

Priority Wise Classification:

- **P0 : *Essential***
  - Identify the relationships between asset clusters usage, the network, and devices
  - Asset Discovery will help organizations to take data driven based decisions.
  - Important Information on Network-Connected Devices.
  - Helps in visualising network topology.
- **P1 : *Important, but could be done without***
  - Helps in understanding vulnerabilities in the network.
  - Provide statistics of time interval for which a specific IP was turned on.
- **P2 : *Can wait, but nice to have***
  - Efficient Management of SSL Certificates.
  - Efficient Management of Domain License Expiry.

Solution statement/ Proposed approach



Details about workflow (EDIT THIS)

- Initially, SSH tunneling between the local computer and the remote server was established.
- Using Python, we created a script that would scan the network for vulnerabilities on both local and remote networks.
- Nothing is installed on the remote server because all of the scan probes for the remote network are tunneled through SSH to the local server.
- All information gathered from scans of the remote network and the local network is processed and stored in CSV data file formats.
- Then, using Python, we process our csv sheet and create a correlated datasheet.
- The processed datasheet is then used to display the scan results using Kibana and Elastic search to index the data and display it.

COMMANDS USED (SSH TUNNEL)

REMOTE SERVER	LOCAL PC
• tunctl -t tap0 • ifconfig tap0 10.100.100.101 netmask 255.255.255.0 • ethtool tap0 • echo 1 > /proc/sys/net/ipv4/ip_forward • iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE • iptables -t nat -A POSTROUTING -o tap0 -j MASQUERADE • iptables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT • iptables -A INPUT -i tap0 -m state --state RELATED,ESTABLISHED -j ACCEPT • iptables -A FORWARD -j ACCEPT	• tunctl -t tap0 • ifconfig tap0 10.100.100.100 netmask 255.255.255.0 • ssh -o Tunnel=ethernet -f -w 0:0 root@remoteip true • ethtool tap0 • ip route add **remoteIp/subnet** via 10.100.100.101

COMMANDS DETAILS (SSH TUNNEL)

- **tunctl -t tap0**
  - Tunctl will setup tap0 as virtual device which behave like real interfaces, so our scanner will not know the difference.

- **ifconfig tap0 10.100.100.100 netmask 255.255.255.0**
  - We're assigning ip addresses to tap0 the virtual interface

- **ifconfig tap0 10.100.100.101 netmask 255.255.255.0**
  - We're assigning ip addresses to tap0 the virtual interface

- **ssh -o Tunnel=ethernet -f -w 0:0 root@remoteip true**
  - It will establish tunnel and execute "true" command , and send process into background.

- **ethtool tap0**
  - Checks if link is established between remote and ground system

- **echo 1 > /proc/sys/net/ipv4/ip_forward**
- **iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE**
- **iptables -t nat -A POSTROUTING -o tap0 -j MASQUERADE**
- **iptables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT**
- **iptables -A INPUT -i tap0 -m state --state RELATED,ESTABLISHED -j ACCEPT**
- **iptables -A FORWARD -j ACCEPT**
  - This commands will enable ip forwarding through NAT (masquerading)

- **ip route add remoteip/subnet via 10.100.100.101**
  - This will add route and enable packets to travel to remote subnet

COMMANDS USED (SCREENSHOTS)

REMOTE SERVER

```
[root@16[redacted] ~]# tunctl -t tap0
Set 'tap0' persistent and owned by uid 0
[root@16[redacted] ~]# ifconfig tap0 10.100.100.100 netmask 255.255.255.0
[root@16[redacted] ~]#
```

```
[root@1[redacted] ~]# ethtool tap0
Settings for tap0:
        Supported ports: [ ]
        Supported link modes:   Not reported
        Supported pause frame use: No
        Supports auto-negotiation: No
        Supported FEC modes: Not reported
        Advertised link modes:  Not reported
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Advertised FEC modes: Not reported
        Speed: 10Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 0
        Transceiver: internal
        Auto-negotiation: off
        MDI-X: Unknown
        Current message level: 0xfffffffa1 (-95)
                               drv ifup tx_err tx_queued intr tx_done rx_s
tus pktdata hw wol 0xffff8000
        Link detected: yes_
```

```
[root@[redacted] ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@[redacted] ~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
[root@[redacted] ~]# iptables -t nat -A POSTROUTING -o tap0 -j MASQUERADE
[root@[redacted] ~]# iptables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
[root@[redacted] ~]# iptables -A INPUT -i tap0 -m state --state RELATED,ESTABLISHED -j ACCEPT
[root@[redacted] ~]# iptables -A FORWARD -j ACCEPT
```

LOCAL PC

```
┌──(root💀kali)-[~]
└─# tunctl -t tap0
Set 'tap0' persistent and owned by uid 0

┌──(root💀kali)-[~]
└─# ifconfig tap0 10.100.100.101 netmask 255.255.255.0
```

```
┌──(root💀kali)-[~]
└─# ssh -v -i id_rsa -o Tunnel=ethernet -f -w any root@1[redacted] true
OpenSSH_8.4p1 Debian-5, OpenSSL 1.1.1k  25 Mar 2021
```

```
┌──(root💀kali)-[~]
└─# ethtool tap0
Settings for tap0:
        Supported ports: [ ]
        Supported link modes:   Not reported
        Supported pause frame use: No
        Supports auto-negotiation: No
        Supported FEC modes: Not reported
        Advertised link modes:  Not reported
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Advertised FEC modes: Not reported
        Speed: 10Mb/s
        Duplex: Full
        Auto-negotiation: off
        Port: Twisted Pair
        PHYAD: 0
        Transceiver: internal
        MDI-X: Unknown
        Current message level: 0x00000000 (0)

        Link detected: yes
```

```
┌──(root💀kali)-[/home/dhaka/swc/ssh_keys]
└─# ip route add [redacted]0/24 via 10.100.100.101

┌──(root💀kali)-[/home/dhaka/swc/ssh_keys]
└─#
```

Commands used details info (Local Network Scan)

- **arp -a**
  - arp command manipulates the System's ARP cache. It also allows a complete dump of the ARP cache. ARP **stands for Address Resolution Protocol**. The primary function of this protocol is to resolve the IP address of a system to its mac address, and hence it works between level 2(Data link layer) and level 3(Network layer)

- **arp-scan --interface=eth0 -localnet**
  - The ARP Scan Tool (AKA ARP Sweep or MAC Scanner) is a **very fast ARP packet scanner that shows every active IPv4 device on your subnet.** Since ARP is non-routable, this type of scanner only works on the local LAN (local subnet or network segment). Devices cannot hide from ARP packets like they can hide from Ping.

- **nmap -sC -sV -sn -A -O -Pn <localIP> --script smb-enum-domains.nse**
  - Nmap is **Linux command-line tool for network exploration and security auditing.**Here, we are scanning ports, ip sweeps, dns enumeration, OS discovery, hostname, mac address scan,etc

COMMANDS USED (SCREENSHOTS: LOCAL PC)

```
┌──(root💀kali)-[/home/dhaka/swc/ssh_keys]
└─# arp-scan -l
Interface: wlan0, type: EN10MB, MAC: 34:e1:2d:4b:f5:aa, IPv4: 192.168.11.44
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.11.39   b2:5b:b8:fd:6a:5a       (Unknown: locally administered)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.096 seconds (122.14 hosts/sec). 1 res
ponded

┌──(root💀kali)-[/home/dhaka/swc/ssh_keys]
└─# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
1[redacted].unified              (incomplete)                              tap0
_gateway                 ether   [redacted]          C                     wlan0
10.100.100.100           ether   aa:e7:07:b3:27:b1   C                     tap0

┌──(root💀kali)-[/home/dhaka/swc/ssh_keys]
└─# nmap -O 192.168.11.44 -Pn
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times wil
l be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-22 00:39 IST
Nmap scan report for 192.168.11.44
Host is up (0.000073s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE
7070/tcp open  realserver
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.95 seconds
```

Commands used details info (Remote Network Scan)

- **nmap -sC -sV -sn -A -O -Pn <RemoteIP> --script smb-enum-domains.nse,dns-nsec-enum**
  - Nmap is **Linux command-line tool for network exploration and security auditing.**Here, we are scanning ports, ip sweeps, dns enumeration, OS discovery, hostname, mac address scan,etc
  - -sC :
  - -sV :
  - -sn :
  - -A :
  - -O : OS detection
  - -Pn :
  - dns-nsec-enum : dns enumeration
  - smb-enum-domains.nse: domain enumeration

COMMANDS USED (SCANS INTRANET REMOTE NETWORK via LOCAL PC)

```
┌──(root💀kali)-[/home/dhaka/swc/ssh_keys]
└─# nmap -O [redacted] -Pn
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times wil
l be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-22 00:29 IST
Nmap scan report for [redacted]
Host is up (0.21s latency).
Other addresses for [redacted] (not scanned): [redacted]
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
Warning: OSScan results may be unreliable because we could not find at least 1 open
 and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Linux 3.X|4.X (86%)
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4.9
Aggressive OS guesses: Linux 3.10 - 3.16 (86%), Linux 4.9 (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.11 seconds
```

Why we used ELK Stack?

- The ELK stack is an acronym that refers to a software stack that is made up of three popular projects: Elasticsearch, Logstash, and Kibana, which are all open source projects.
- IP addresses, operating system information, mac address information, manufacturer, hostnames and domain names, workgroups and domain information as well as status and ports are all contained in a csv file that we have available.
- All IP addresses now have correlations between ports and the various services that run over them.
- We have a data csv file containing the results of a full local network scan and a remote subnet scan.
- To effectively interact with the displayed data, we must sort, search, and interact with it in a specific manner that is highly interconnected with one another.
- We use Kibana as our database and frontend system, rather than any other database or frontend system.
- In addition, we have a scan that is running continuously for an extended period of time (see below).
- This information is needed for visualization and management of the data collected, as well as the generation of alerts for various purposes and applications.

Limitations

- Due to the fact that this **tool only scans the remote system's subnet**, it is possible that **it will miss some assets on the remote network**.

- In some cases, depending on how the scanner system operates, **requests or pings made by the scanner system may be denied by the firewall** because of their nature.

- Due to the frequency with which this tool scans, it may cause the network to become **overburdened**, **degrade user expectations**, **increase response time**, and even increase the cost of service upkeep and maintenance.

Future Scope

- Updates to our scanning probe may aid in standardization, and this testing may aid in performance optimization and early detection of problems.
- Our current scanning system may benefit from updates, which will aid us in the identification and resolution of security threats.
- Improvements in asset monitoring may make it possible for businesses to implement new technologies and system upgrades more efficiently.
- Because network traffic capacity is constantly being increased and improved, our scanner can be updated to provide statistics for a relatively short time interval for which a specific IP was turned on.

References:

- https://isc.sans.edu/forums/diary/Tunneling+scanners+or+really+anything+over+SSH/24286/ (ssh tunnel)
- NMAP: https://nmap.org/
- arp: https://en.wikipedia.org/wiki/Address_Resolution_Protocol
- arp-scan: https://linux.die.net/man/1/arp-scan
- Active Directory reconacense : https://exploit.ph/active-directory-recon-1.html