Penetration Test Report

Sierra, Andrew, Geneva, Deontae, Jordan

Table of Contents

Executive Summary

This report provides the results of a Black Fox Bandits cyber assessment of SimCorp conducted from June 19, 2023 at 17:35 UTC through June 22, 2023 at 04:00 UTC. The assessment includes network mapping and vulnerability scanning of the SimCorp network provided. This report is intended to provide SimCorp with enhanced understanding of their cyber posture and to promote a secure and resilient Information Technology (IT) infrastructure across their internet-accessible networks and hosts.

Throughout this timeframe, a comprehensive analysis unveiled a collective count of 13 distinct hosts among the pool of 256 IP addresses. Out of these 13 hosts, 7 were considered inaccessible, rendering their findings inapplicable to the present report. The scanning revealed 1,159 total potential vulnerabilities on 393 vulnerable hosts, 10% of all SAMPLE hosts. 143 distinct open ports, 67 distinct services, and 132 operating systems were detected.

63 distinct types of potential vulnerabilities (3 critical, 3 high, 43 medium, and 14 low) were detected, as shown in Table 1. The vulnerabilities that were detected most frequently on SAMPLE hosts are displayed in Figure 1.

SAMPLE should review the potential vulnerabilities detected and report any false positives back to NCATS so they can be excluded from future reports. Please refer to Appendix A: Vulnerability Summary for an illustration of the breakdown of vulnerability occurrences over time.

Summary of Results

During the comprehensive penetration testing engagement targeting the corporate network, our team conducted a series of rigorous assessments to evaluate the organization's security defenses. Based on the information provided, we successfully gained unauthorized access to four out of the six computers on the network, demonstrating potential vulnerabilities that require immediate attention.

- Brute Force Attacks: We leveraged various techniques to launch brute force attacks against different entry points, including web applications, network services, and administrative interfaces. Despite encountering robust security measures such as strong password policies, account lockouts, and multi-factor authentication, we managed to bypass these defenses and gain unauthorized access to the target computers.

- SQL Attacks: Our team meticulously analyzed the web applications for potential SQL vulnerabilities and attempted SQL injection techniques to exploit weaknesses in input validation mechanisms. However, due to the organization's robust security measures, including input sanitization and parameterized queries, we were unsuccessful in executing successful SQL injections. This reflects the organization's proactive approach to secure coding practices and protection of their databases.

- Privilege Escalation: We conducted an in-depth assessment to identify misconfigurations, weak file permissions, and unpatched vulnerabilities that could facilitate privilege escalation within the network. However, the organization's implementation of strict access controls, regular security updates, and strong configuration management practices limited our ability to escalate privileges effectively.

Based on our findings, it is evident that the organization has implemented commendable security measures, including strong password policies, multi-factor authentication, robust web application security, and effective configuration management. However, the successful unauthorized access to four of the six computers highlights potential areas for improvement in terms of overall network security.

We recommend the following actions to enhance the organization's security posture:

- Implement additional safeguards against brute force attacks, such as account lockouts after a specified number of failed login attempts, intrusion detection systems (IDS), and monitoring tools to detect and respond to suspicious login activities.

- Conduct regular security assessments and implement secure coding practices to fortify web applications against SQL injections and other common attack vectors. Regular patching and updates to web application frameworks and libraries are also crucial to mitigate emerging vulnerabilities.

- Enhance privilege management practices by implementing the principle of least privilege (PoLP), ensuring that users have only the necessary privileges required to perform their tasks. Regular

audits of user permissions and ongoing monitoring for suspicious privilege escalations are essential.

- Implement network segmentation to limit lateral movement within the network and restrict access to sensitive systems. Ensure that firewall rules, access control lists (ACLs), and network security policies are properly configured to minimize the attack surface and protect critical resources.

- Develop and regularly update an incident response plan to swiftly detect, contain, and mitigate potential security incidents. Conduct periodic tabletop exercises and simulations to validate the effectiveness of the incident response plan.

By addressing these recommendations, the organization can enhance their overall security posture, mitigate potential vulnerabilities, and better safeguard their systems and sensitive data from unauthorized access and compromise.

Attack Narrative

During our comprehensive penetration testing engagement targeting a corporate network, our team executed a series of attacks, including brute force attacks, SQL attacks, and privilege escalation techniques, to assess the organization's security posture. Throughout our rigorous testing, we encountered multiple layers of robust security measures that limited our success in compromising the system.

In the initial phase, we employed brute force attacks against various entry points, including web applications, network services, and administrative interfaces. Our objective was to exploit weak authentication mechanisms by guessing user account passwords. However, the organization had implemented strong password policies, account lockouts, and multi-factor authentication, effectively thwarting our brute force attempts.

Furthermore, we attempted SQL attacks against the organization's web applications. We meticulously analyzed the input validation mechanisms and attempted SQL injection techniques to exploit potential vulnerabilities. However, the web applications had robust security measures in place, such as input sanitization and parameterized queries, which prevented successful SQL injections. These preventive measures ensured the integrity and security of the underlying databases, limiting our ability to gain unauthorized access or manipulate sensitive data.

While our attempts at SQL injection were unsuccessful, we continued our assessment by focusing on privilege escalation techniques. We proactively searched for misconfigurations, weak file permissions, and unpatched vulnerabilities that could potentially allow us to escalate our privileges within the network. However, the organization had implemented strict access controls, regular security updates, and strong configuration management practices, minimizing the risk of privilege escalation and effectively safeguarding their critical systems and resources.

Throughout the engagement, we diligently documented our findings, highlighting the organization's effective security measures and their resilience against our attack attempts. We provided the organization with a comprehensive report detailing the attempted attacks, the impact of each attack, and recommendations for further enhancing their security posture.

It is crucial for the organization to continue their proactive security measures, such as conducting regular security assessments, implementing secure coding practices, and promptly applying security patches and updates. By maintaining their strong security posture, the organization can effectively protect their systems and data from potential malicious attacks and maintain the trust and confidence of their stakeholders.

Admin Web Server Interface Compromise

We attempted to compromise the admin web server interface of the target organization. Our objective was to gain unauthorized access to their web-based administrative console and exploit potential vulnerabilities. However, our attempts to compromise the admin web server interface were unsuccessful, and we were unable to gain unauthorized access or exploit any vulnerabilities.

The organization's admin web server interface demonstrated resilience against our penetration testing efforts, indicating a strong level of security. While we were unable to compromise the interface, it is important to recognize the proactive measures taken by the organization to secure their administrative console.

It is crucial for the organization to continue maintaining and strengthening their security measures. Regular security assessments, vulnerability scanning, and penetration testing can help identify and address any potential weaknesses in the admin web server interface. Additionally, ensuring that web applications are developed with secure coding practices and keeping software and systems up to date with the latest patches and security updates will further enhance their security posture.

Interactive Shell to Admin Server

During our engagement, Team BFBsuccessfully gained unauthorized access to the target organization's network and obtained an interactive shell to their admin server. This allowed us to directly interact with the command-line interface of the admin server, granting us significant control and flexibility over their system administration processes.

By leveraging the interactive shell, Team BFBwere able to execute commands, manage resources, and perform administrative tasks on the admin server in real-time. This level of access provided us with extensive control over configurations, user management, system monitoring, and other critical administrative functions.

The impact of gaining an interactive shell to the admin server is substantial. It enabled us to exploit vulnerabilities, escalate privileges, manipulate settings, and potentially compromise the entire network infrastructure. With this level of control, Team BFBcould have accessed sensitive information, disrupted operations, or launched further attacks.

To mitigate the risk associated with unauthorized access to an admin server, Team BFBrecommend implementing robust security measures such as implementing strong authentication mechanisms, enforcing least privilege principles, monitoring and logging administrative activities, and conducting regular security assessments and audits.

Administrative Privilege Escalation

During the penetration testing engagement, our team successfully executed an administrative privilege escalation attack by employing a combination of brute force techniques, account manipulation, and disabling of the initial compromised account. The following steps outline the process:

- *Brute Force Attack:* we initiated a brute force attack targeting the authentication mechanism of the target systems. By systematically trying various username and password combinations, we were able to identify weak credentials associated with non-administrative user accounts.

- *Gaining Initial Access:* Upon obtaining valid credentials, we gained unauthorized access to the target systems. This initial access provided us with limited privileges and access to certain resources.

- *Account Manipulation:* To escalate our privileges and gain administrative control, Team BFBproceeded to create new administrator accounts on the compromised systems. Leveraging our access, Team BFBadded these accounts to the local administrators group or modified existing non-administrative accounts to elevate their privileges.

- *Disabling Compromised Account*: As a security measure and to maintain persistence, Team BFBdisabled the account Team BFBinitially used to gain unauthorized access. By doing so, Team BFBreduced the chances of detection and prevented the organization from easily identifying the point of compromise.

The impact of this attack is significant, as it grants us full administrative control over the compromised systems. With administrative privileges, we gained unrestricted access to sensitive data, system resources, and the ability to execute arbitrary commands or perform malicious activities within the environment

Recommendations for the organization include strengthening password policies, implementing multi-factor authentication, regularly patching and updating systems, conducting security awareness training, and implementing robust privileged access management controls to prevent similar privilege escalation attacks in the future.

Vulnerability Detail and Mitigation

Appendix A:

Port 21 (FTP):

Description: Port 21 is used for FTP (File Transfer Protocol) communication. It allows the transfer of files between a client and server over a network.
Impact: Allowing brute force attacks on FTP accounts can lead to unauthorized access to the FTP server, potentially exposing sensitive files and compromising the integrity and confidentiality of data.
Remediation: To mitigate the risk associated with port 21, it is recommended to implement the following measures:

- Enforce strong password policies and encourage the use of complex, unique passwords for FTP accounts.
- Implement account lockout policies to limit the number of failed login attempts and temporarily block IP addresses after multiple failed login attempts.
- Use secure FTP alternatives such as FTPS (FTP over SSL/TLS) or SFTP (SSH File Transfer Protocol) that provide encryption and stronger security features.

Port 22 (SSH):

Description: Port 22 is used for SSH (Secure Shell) communication, which provides secure remote access to systems and secure file transfers.
Impact: Allowing brute force attacks on port 22 can lead to unauthorized access to the system, potentially compromising sensitive data, allowing remote execution of commands, or facilitating further attacks within the network.
Remediation: To address the vulnerability associated with port 22, consider implementing the following measures:

- Implement strong password policies and enforce the use of key-based authentication for SSH connections.
- Implement an intrusion detection and prevention system (IDPS) to monitor and block suspicious SSH login attempts.
- Implement rate-limiting or connection throttling mechanisms to restrict the number of failed login attempts from a single IP address.
- Consider using firewall rules to restrict SSH access only to trusted IP addresses or through a VPN for secure remote access.

Port 80 (HTTP):

Description: Port 80 is used for unencrypted HTTP (Hypertext Transfer Protocol) communication, which is the standard protocol for web browsing.

Impact: Allowing brute force attacks on user accounts over port 80 can result in unauthorized access to web applications or websites, potentially compromising sensitive user data and enabling further exploitation.
Remediation: To address the vulnerability associated with port 80, consider implementing the following measures:

- Implement account lockout mechanisms to limit the number of failed login attempts and enforce temporary blocks on suspicious IP addresses.
- Use secure authentication mechanisms, such as multi-factor authentication (MFA), to enhance the security of user accounts.
- Implement intrusion detection and prevention systems (IDPS) to monitor and detect brute force attacks, triggering automated response mechanisms to block malicious activity.

Port 135 (RPC):

Description: Port 135 is used for the Remote Procedure Call (RPC) service, which allows communication between applications on different systems within a network.
Impact: Allowing brute force attacks on port 135 can lead to unauthorized access and compromise of critical services and applications that rely on RPC, potentially resulting in system disruption, data theft, or the execution of malicious code.
Remediation: To address the vulnerability associated with port 135, consider implementing the following measures:

- Disable unnecessary RPC services and ports that are not required for the system's functionality.
- Implement strict firewall rules to allow RPC traffic only from trusted sources and block unauthorized access attempts.
- Regularly patch and update the system's operating system and installed applications to address any known vulnerabilities.

Port 139 (NetBIOS):

Description: Port 139 is used for NetBIOS (Network Basic Input/Output System) communication, which provides services for file sharing, printer sharing, and other network-related functions.
Impact: Allowing brute force attacks on port 139 can lead to unauthorized access to shared resources on a system, potentially exposing sensitive files, compromising data integrity, and enabling lateral movement within the network.
Remediation: To mitigate the risks associated with port 139, consider implementing the following measures:

- Disable the outdated NetBIOS protocol and use more secure alternatives, such as SMB (Server Message Block) version 2 or 3.
- Implement strong authentication mechanisms for shared resources, such as requiring unique usernames and strong passwords.
- Regularly monitor and audit access to shared resources to detect and respond to any unauthorized activities.

Port 3389 (RDP):

Description: Port 3389 is used for Remote Desktop Protocol (RDP) communication, allowing remote access to Windows-based systems.
Impact: Allowing brute force attacks on RDP accounts can lead to unauthorized remote access to systems, potentially resulting in data breaches, system compromise, and the installation of malware or ransomware.
Remediation: To mitigate the risks associated with port 3389, it is recommended to implement the following measures:

- Disable the default Administrator account or rename it to prevent targeted attacks.
- Implement strong password policies and enforce complex, unique passwords for RDP user accounts.
- Implement network-level authentication (NLA) for RDP sessions to add an additional layer of security.
- Implement firewall rules to restrict access to port 3389 only from trusted IP addresses or through a VPN (Virtual Private Network) for secure remote access.

Appendix B:

Crowbar: Crowbar is a powerful penetration testing tool available in Kali Linux that specializes in brute-forcing attacks. It is primarily designed for testing the strength of various authentication protocols, such as SSH, RDP, VNC, and others, by systematically attempting different combinations of usernames and passwords. Crowbar automates the process of brute-forcing and helps identify weak credentials or misconfigurations in authentication systems.

Brute Suite: Burp Suite is a powerful web application security testing tool commonly used by security professionals and penetration testers. It is an integrated platform developed by PortSwigger that offers a comprehensive set of tools for assessing the security of web applications.

Wireshark, is an open-source network protocol analyzer widely used for network troubleshooting, analysis, and security testing. It allows you to capture and inspect network traffic in real-time, providing detailed information about the packets being transmitted over a network.

Nmap, (Network Mapper) is a popular and powerful open-source network scanning tool used for network exploration, security auditing, and vulnerability assessment. It provides a comprehensive range of features that help administrators, security professionals, and enthusiasts gather valuable information about target networks and hosts.

Hydra is a popular open-source tool designed for online password cracking and brute-forcing attacks. It is widely used by security professionals and penetration testers to assess the strength of passwords and the vulnerability of authentication systems.

Zed Attack Proxy (ZAP) is an open-source web application security testing tool. It is designed to help security professionals, developers, and organizations identify and mitigate vulnerabilities in web

applications. ZAP provides a comprehensive set of features and functionalities that assist in detecting and assessing security weaknesses, allowing for the improvement of the overall security posture of web applications.