|  |  |  |  |
| --- | --- | --- | --- |
|  |  | How many shells (Core)  Otis Smith / Cybersecurity Professional / 12.7.23 |  |
|  |  |  |  |
| Pipette dropping liquid in a petri dish | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
|  |  | TABLE OF CONTENTS |  |
|  |  | Summary …………………………..  Discovery……………………………  Vulnerability……………………….  Exploitation………………………..  References.……………………….. |  |

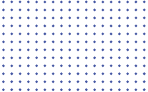


|  |  |  |
| --- | --- | --- |
| Summary |  | |
| This offensive security engagement aimed to exploit a target machine, Metasploitable3 Windows, providing a comprehensive walkthrough of the process. The focus was on discovering vulnerabilities in the GlassFish Server, leveraging a path traversal exploit to gain unauthorized access, and subsequently achieving a Meterpreter session.  A hand holding a glowing city  Description automatically generated | |  |
| Discovery   1. Network Enumeration: Conducted network discovery using nmap 10.0.2.1/24 on the Kali machine, identifying Metasploitable3's IP (10.0.2.9).A screenshot of a computer program     Description automatically generated      1. Detailed Enumeration: Employed nmap -A 10.0.2.9 to explore Metasploitable3, identifying interesting ports 4848 and 8181 associated with GlassFish Server.A screenshot of a computer program     Description automatically generatedA screenshot of a computer     Description automatically generated 2. Exploitable Vulnerability: Identified a known path traversal vulnerability (ExploitDB: 45196) in Oracle GlassFish Server Open Source Edition 4.1. A screenshot of a computer     Description automatically generated | |  |
|  | |  |

|  |  |  |
| --- | --- | --- |
| Vulnerability |  | |
| 1. Metasploit Framework: Utilized msfconsole to search for and select the GlassFish module. A screenshot of a computer     Description automatically generated 2. File Preparation: Created Username.txt and Password.txt files on Kali for required authentication.A screenshot of a computer program     Description automatically generated 3. Exploit Configuration: Set relevant options (RHOSTS, RPORT, USER\_FILE, PASS\_FILE)   A screenshot of a computer program  Description automatically generated   1. SSL Redirection: Encountered an error related to unexpected SSL redirection, resolved by setting SSL true. 2. Credential Retrieval: Successfully retrieved credentials (admin:sploit) for GlassFish Server.   A screenshot of a computer screen  Description automatically generated   1. Web Console Access: Logged into the GlassFish Console using obtained credentials, gaining access to administrative tasks.A screenshot of a computer     Description automatically generated | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exploitation | | |  | |
| 1. Update Tool Exploration: Explored GlassFish Console functionalities, particularly the Update Tool, revealing the server's file structure (C:\glassfish\glassfish4).A screenshot of a computer     Description automatically generated 2. Meterpreter Payload: Configured Metasploit with Meterpreter payload for Java (payload/java/meterpreter/reverse\_tcp). A screenshot of a computer     Description automatically generated 3. Exploitation: Executed the exploit, gaining a Meterpreter session on the target machine. 4. Verification: Confirmed access through commands like getuid, sysinfo, whoami and accessing the shell.   A screenshot of a computer program  Description automatically generatedA computer screen shot of a computer  Description automatically generatedA computer screen shot of a computer screen  Description automatically generated | | | |  |
| References |  |  | |  |

1. ExploitDB GlassFish Path Traversal
2. nmap 10.0.2.1/24
3. nmap -A 10.0.2.9
4. msfconsole, search GlassFish, use 1, set RHOSTS 10.0.2.9, set RPORT 4848, set USER\_FILE, set PASS\_FILE, set STOP\_ON\_SUCCESS true, set SSL true, run
5. GlassFish Console: admin:sploit, explored "Update Tool," and executed Meterpreter payload.
6. Meterpreter: getuid, sysinfo, shell, whoami, help

Mitigation: 

1. Patch and Update:
   * Regularly update and patch the GlassFish Server and associated software to address known vulnerabilities.
   * Stay informed about security updates from Oracle and apply them promptly.
2. Vulnerability Scanning:
   * Conduct regular vulnerability scans to identify and remediate potential weaknesses in the infrastructure.
   * Use automated tools to scan for known vulnerabilities in web applications.
3. Access Control:
   * Implement strong access controls on administrative interfaces to restrict unauthorized access.
   * Enforce the principle of least privilege to ensure that users have the minimum access required for their roles.
4. Network Segmentation:
   * Implement network segmentation to isolate critical servers and services from less secure parts of the network.
   * Use firewalls to control traffic between different segments.
5. Security Training:
   * Provide security awareness training for system administrators, developers, and users to educate them about common vulnerabilities and exploitation techniques.
   * Emphasize the importance of secure coding practices.
6. Web Application Firewall (WAF):
   * Deploy a Web Application Firewall to monitor and filter HTTP traffic between a web application and the Internet.
   * Configure the WAF to detect and block requests indicative of path traversal attempts.
7. Monitoring and Logging:
   * Implement robust monitoring and logging solutions to detect and alert on suspicious activities.
   * Regularly review logs to identify unauthorized access or unusual patterns.
8. SSL/TLS Configuration:
   * Ensure proper SSL/TLS configuration to prevent unexpected SSL redirection or vulnerabilities related to encryption.
   * Regularly review and update SSL/TLS configurations.
9. File and Directory Permissions:
   * Review and set appropriate file and directory permissions to prevent unauthorized access.
   * Regularly audit and update permissions based on the principle of least privilege.
10. Incident Response Plan:
    * Develop and maintain an incident response plan to handle security incidents effectively.
    * Conduct regular drills and training to ensure a swift and coordinated response.
11. Secure Coding Practices:
    * Follow secure coding practices to minimize the risk of path traversal vulnerabilities.
    * Input validation and output encoding should be implemented in web applications.
12. Regular Security Audits:
    * Conduct regular security audits to identify and address security weaknesses.
    * Engage in penetration testing to identify and remediate vulnerabilities proactively.

By adopting these mitigation strategies, organizations can enhance the security posture of their systems and reduce the risk of unauthorized access through path traversal exploits. A holistic approach to cybersecurity involves a combination of technical measures, user education, and proactive monitoring.

In conclusion, this engagement successfully demonstrated the exploitation of a known vulnerability in GlassFish Server, showcasing the importance of timely patching and securing web applications. The exploitation resulted in gaining unauthorized access to the target, highlighting the critical need for robust cybersecurity practices.