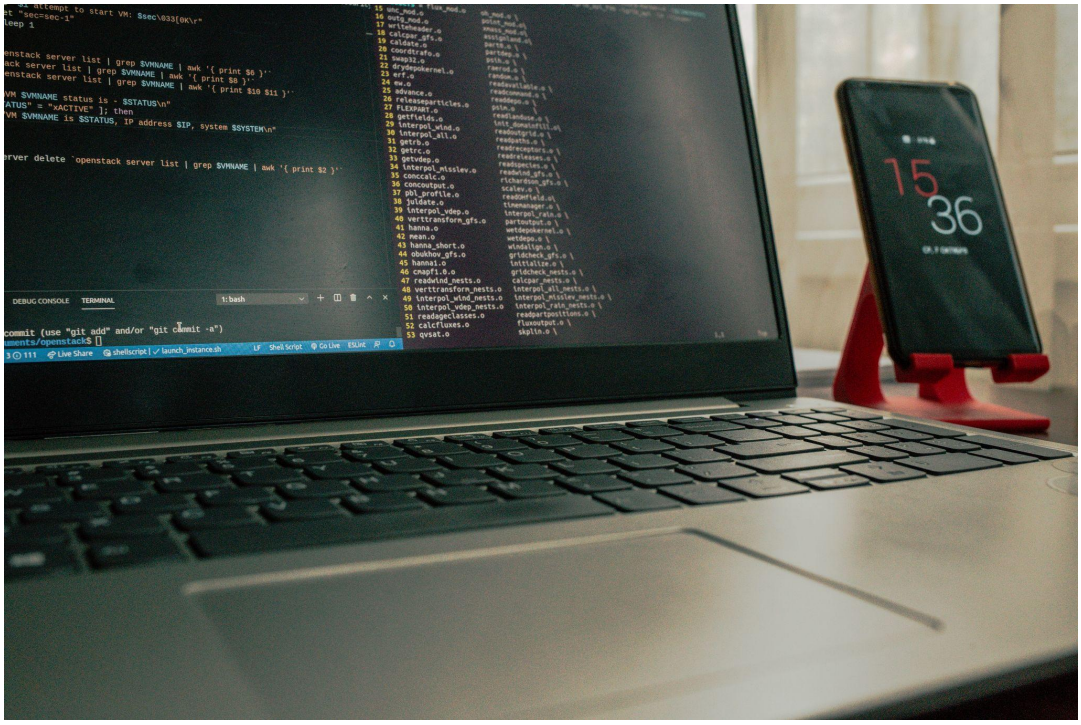
Cybersecurity

Penetration Test Report



Rekall Corporation

Dtureo Security Services, LLC

Confidentiality Statement

This document contains confidential and privileged information from Rekall Corporation 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

Contact Information

Company Name	Dtureo Security Services, LLC
Contact Name	Daniel Tureo
Contact Title	Penetration Tester
Contact Phone	813 143 0432
Contact Email	dtureo@dss.com.au

Document History

Version	Date	Author(s)	Comments
001	29/06/2023	Daniel Tureo	v0.1
002	16/07/2023	Daniel Tureo	v0.2

Introduction

In accordance with Rekall's policies, Dtureo Security Services, LLC (henceforth known as DSS) 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. The project was conducted on a number of systems on Rekall's network segments by DSS during June of 2023.

For the testing, DSS 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.

DSS used its 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

DSS begins 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 Nessus.

Identification of Vulnerabilities and Services

DSS uses custom, private, and public tools such as Metasploit, Nessus, 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

DSS's 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.

IP Address/URL	Description
192.168.13.0/24 172.22.117.0/24 *.totalrekall.xyz	Rekall internal domain, range and public website

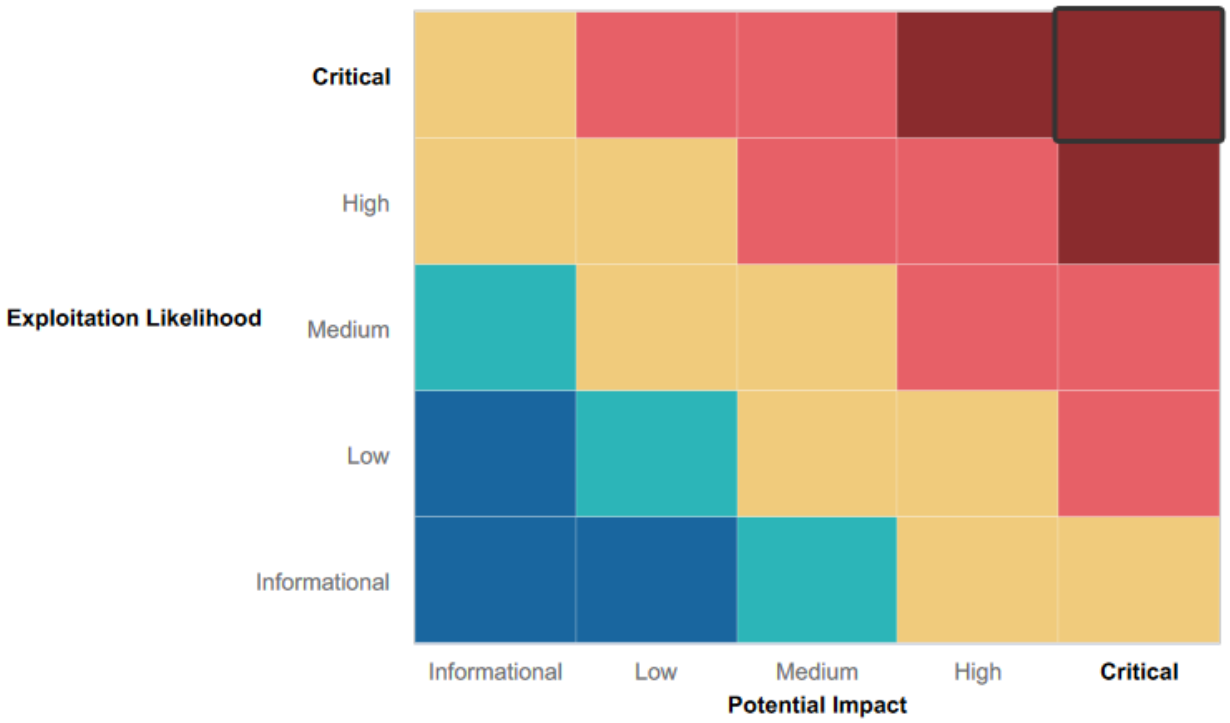
Executive Summary of Findings

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 positive aspects highlight the successful prevention, detection, or denial of various attack techniques and tactics. Some of the strengths observed include:

- Effective Incident Response: The organization's incident response protocols proved efficient, allowing for timely detection and mitigation of security incidents, minimizing potential damage.
- System Patching and Updates: Rekall demonstrated a consistent approach to system patching and updates, reducing the risk of known vulnerabilities being exploited.
- Security Awareness Training: Employees exhibited a strong understanding of security best practices, reflecting the success of the organization's security awareness training program.
- Regular Security Audits: Rekall's proactive approach to conducting regular security audits ensured vulnerabilities were identified and addressed promptly, reducing the attack surface.

Summary of Weaknesses

During the penetration test, DSS identified several critical vulnerabilities across the network. These weaknesses pose a significant risk and demand immediate attention to prevent potential exploitation. The vulnerabilities encompass a range of issues, including:

- XSS, SQL, and PHP Injections: These code injections expose the application to malicious scripts and unauthorized database access, jeopardizing data integrity and user privacy.
- Sensitive Data Exposure: Information such as login credentials and sensitive website details were found publicly accessible, potentially leading to data breaches and unauthorized access.
- Local File Inclusion (LFI): LFI vulnerabilities enable attackers to upload and execute arbitrary PHP files, posing a direct threat to the system's security.
- Command Injection: The vulnerabilities allow attackers to execute malicious commands on the system, potentially leading to unauthorized access and data manipulation.
- Brute Force Attack: Weak password policies allow attackers to systematically guess passwords, granting unauthorized access to accounts and compromising system security.
- Session Management Weakness: Poor session management enables unauthorized users to gain access to privileged information and user accounts.
- Directory Traversal: The vulnerability enables attackers to access sensitive files and directories, risking data exposure and unauthorized modification.
- Nessus / Nmap Scans: Vulnerabilities identified through scanning tools indicate potential entry points for attackers.
- Remote Code Execution (RCE) Exploits: These exploits allow attackers to execute arbitrary code on the target system, leading to complete compromise.
- Scheduled Task Vulnerabilities: Poorly configured scheduled tasks expose the system to potential manipulation and unauthorized access.

These vulnerabilities demand immediate remediation through patching, secure configuration, and regular security audits. By addressing these weaknesses proactively, the organization can strengthen its security posture and mitigate the risks of potential cyberattacks.

Executive Summary

The penetration test on Rekall's environment uncovered critical vulnerabilities and identified strengths in its cybersecurity defenses. Web applications, Linux and Windows OS, and network infrastructure were assessed, revealing numerous weaknesses such as XSS, SQL Injection, LFI, Command Injection, Brute Force Attacks, and PHP Injection. Additionally, sensitive data exposure and session management issues were detected.

Popular programs like Nessus, Nmap, and Metasploit and also OSINT tools were instrumental in identifying vulnerabilities and exploiting weaknesses, showcasing the potential impact of these security gaps.

Despite the vulnerabilities, Rekall demonstrated strengths in its proactive incident response, regular security audits, software updates and patches, robust network perimeter, advanced endpoint protection, efficient access controls, and employee security awareness.

To mitigate risks and enhance security, Rekall should prioritize patching critical vulnerabilities, implement secure coding practices, and strengthen access controls. Regular security audits and network monitoring are essential to proactively identify emerging threats. Additionally, investing in employee security awareness training and prompt system patching will bolster their cybersecurity posture and prevent future attacks. Continuous vigilance and proactive security measures are crucial in the ever-evolving threat landscape.

Summary Vulnerability Overview

Vulnerability - Host	Type	Severity
Reflected XSS - 192.168.14.35/Welcome.php	Web App	High
Reflected XSS Advanced - 192.168.14.35/Memory-Planner.php	Web App	High
Sensitive data exposure - 192.168.14.35/About-Rekall.php	Web App	Low
Local File inclusion (LFI) - 192.168.14.35/Memory-Planner.php	Web App	Critical
LFI (Advanced) - 192.168.14.35/Memory-Planner.php	Web App	Critical
SQL Injection - 192.168.14.35/Login.php	Web App	Critical
Sensitive data exposure - 192.168.14.35/Login.php	Web App	High
Sensitive data exposure - 192.168.14.35/robots.txt	Web App	Medium
Command injection - 192.168.14.35/networking.php	Web App	Critical
Command injection (advanced) - 192.168.14.35/networking.php	Web App	Critical
Brute force attacks - 192.168.14.35/networking.php	Web App	Critical
PHP injection - 192.168.14.35/souvenirs.php	Web App	Critical
Session management - 192.168.14.35/admin_legal_data.php	Web App	Critical
Directory traversal - 192.168.14.35/disclaimer.php	Web App	Critical
Open-source exposed data - totalrekall.xyz	Linux OS	Low
Ping totalrekall.xyz - 3.33.130.190	Linux OS	Low
Open-source exposed data - Certificates for totalrekall.xyz	Linux OS	Low
Nessus scan - 192.168.13.12	Linux OS	Critical
RCE exploit Apache Tomcat - 192.168.13.10	Linux OS	Critical
Shellshock - 192.168.13.11	Linux OS	Critical
Additional findings in host - 192.168.13.11	Linux OS	Critical
RCE exploit Struts - 192.168.13.12	Linux OS	Critical
RCE exploit Drupal - 192.168.13.13	Linux OS	Critical
SSH user - 192.168.13.14	Linux OS	Critical
Totalrekall GitHub page - github.com/totalrekall	Windows OS	Medium
Nmap scan FTP - 172.22.117.20	Windows OS	Medium
Nmap scan SLMAIL - 172.22.117.20	Windows OS	Critical
Scheduled task - 172.22.117.20	Windows OS	Critical

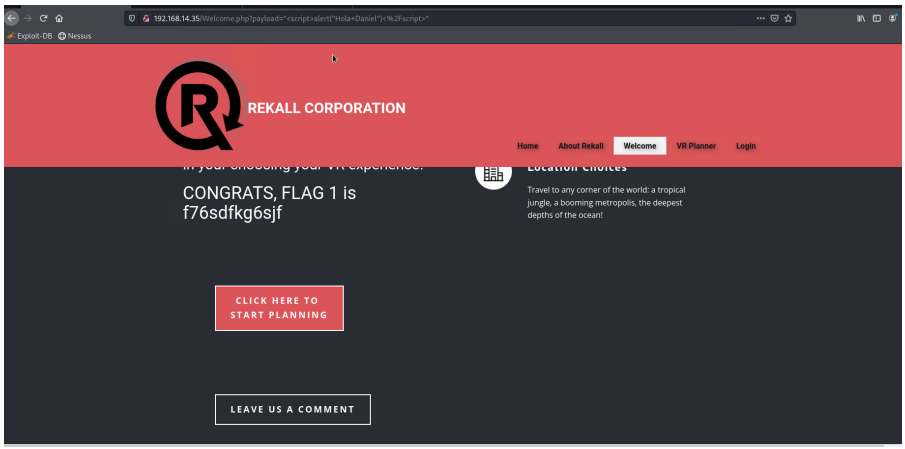
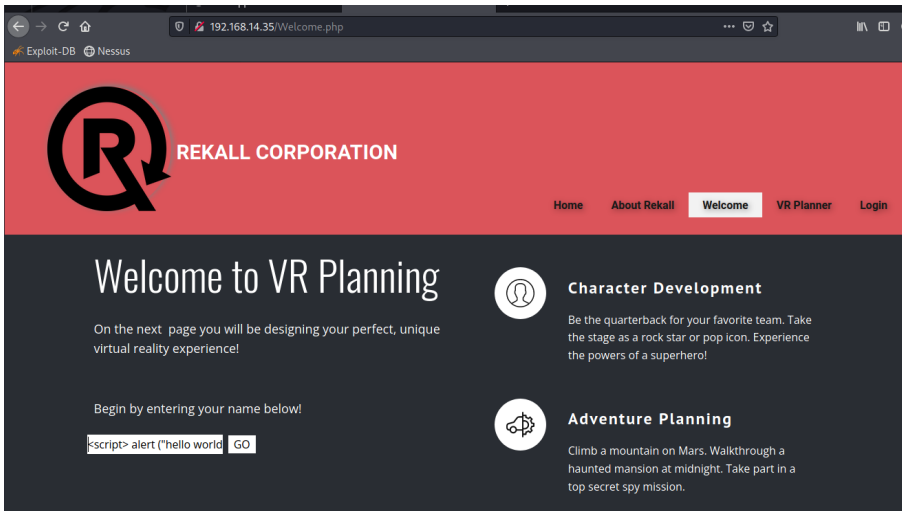
The following summary tables represent an overview of the assessment findings for this penetration test:

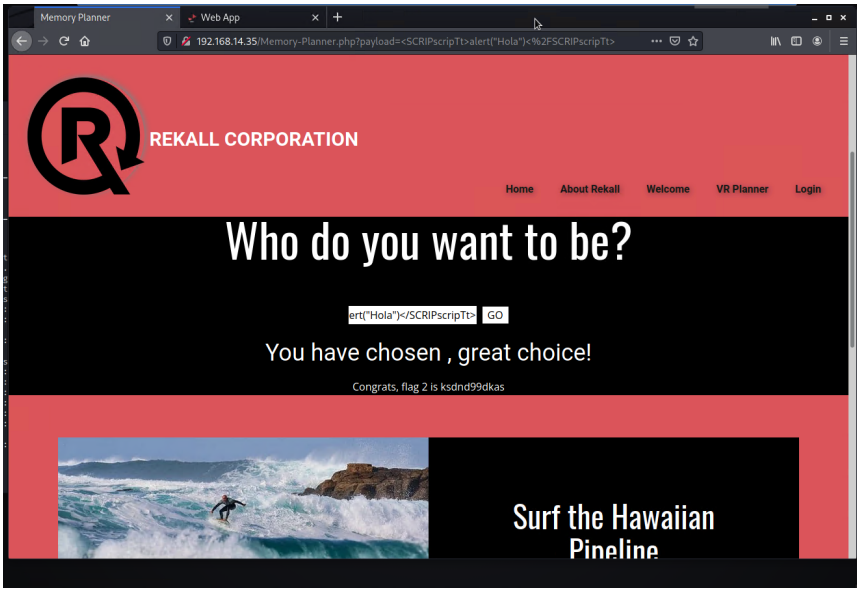
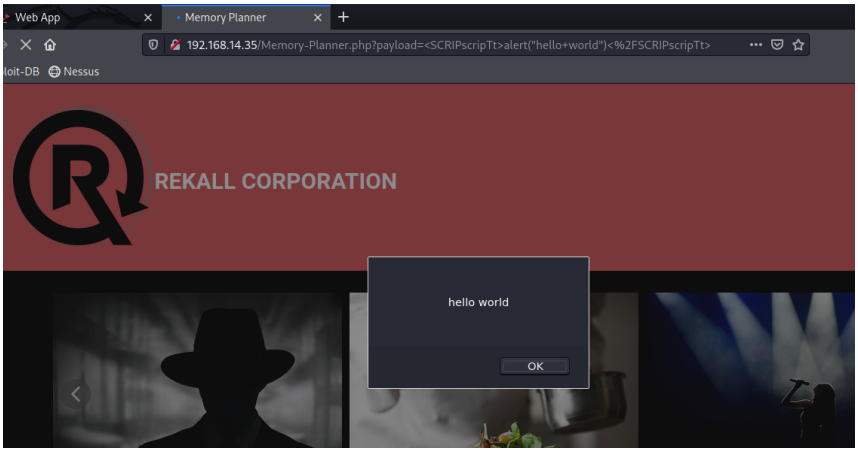
Scan Type	Total
Hosts	192.168.13.10 (Linux) 192.168.13.11 (Linux) 192.168.13.12 (Linux) 192.168.13.13 (Linux) 192.168.13.14 (Linux) 172.22.117.10 (Windows) 172.22.117.20 (Windows) 3.33.130.190 (Web server) 192.168.14.35 (Website)
Ports	21 22 25 80 110 8080 4444

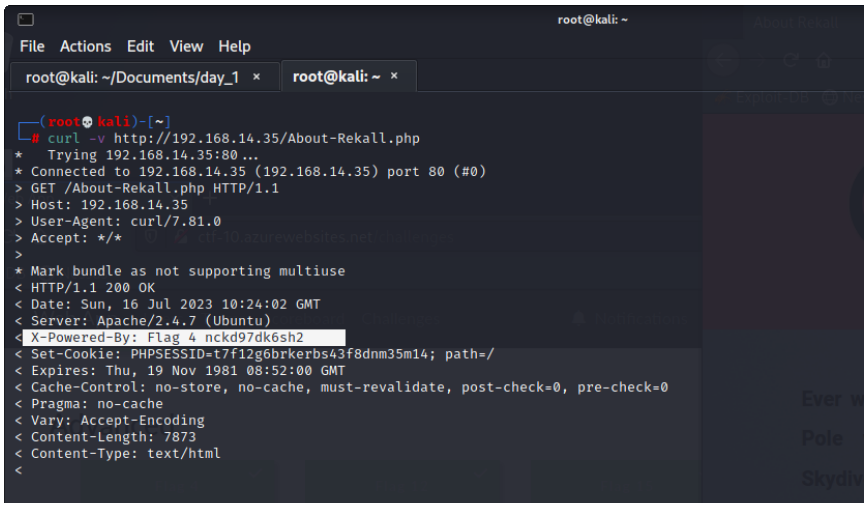
Exploitation Risk	Total
Critical	18
High	3
Medium	3
Low	4
Informational	0

Vulnerability Findings

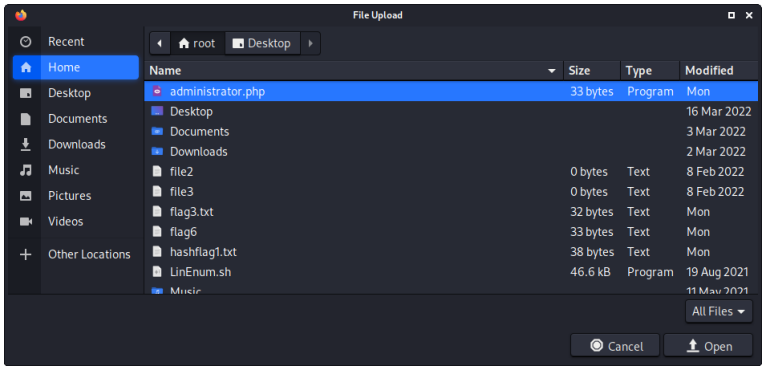
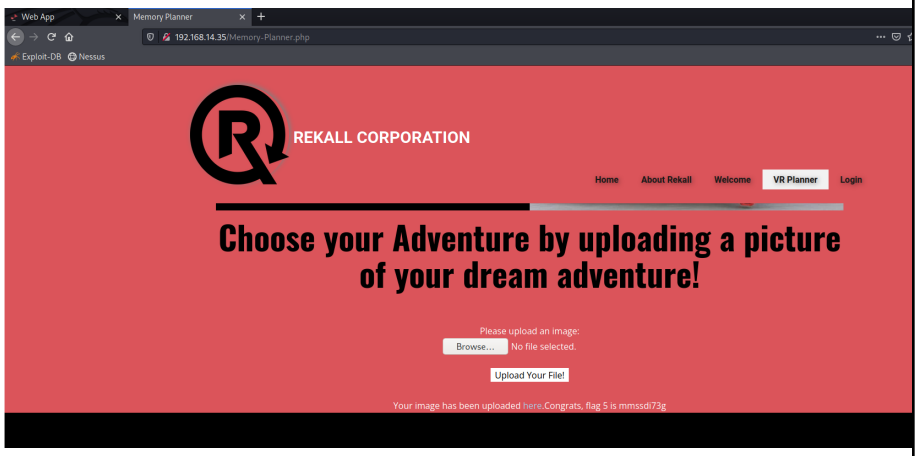
First penetration test findings

Vulnerability	Findings
Title	Reflected XSS - Flag 1
Type	Web App
Risk	High
Description	XSS vulnerability found in Welcome.php. This script was used to reveal the issue “<script>alert(“Hello World”)</script>”.
Founded in or affected host	http://192.168.14.35/Welcome.php
Remediation / Recommendation	Input validation and output encoding techniques should be implemented in the application code to properly sanitize user-supplied data and prevent the execution of arbitrary code.
Images	 

Vulnerability	Findings
Title	Reflected XSS advanced - Flag 2
Type	Web App
Risk	High
Description	Similar to the previous vulnerability, but in this case there is a input validation which looks and removes the word “script” only. To exploit this vulnerability a script like this was used “<SCRIPscripTt>alert(“Hello world”)</SCRIPscripTt>”.
Founded in or affected host	http://192.168.14.35/Memory-Planner.php
Remediation / Recommendation	A more robust input validation and output encoding approach should be implemented. The validation should consider various bypass techniques and properly sanitize user-supplied input to prevent the execution of any arbitrary code.
Images	 

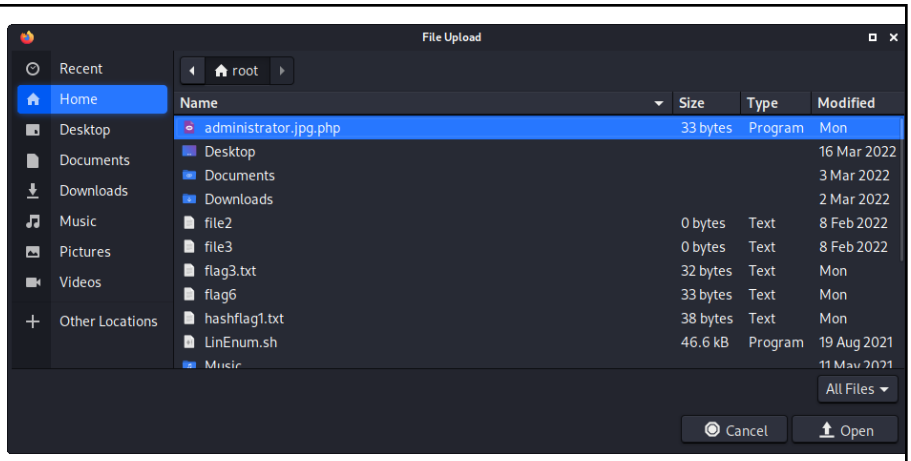
Vulnerability	Findings
Title	Sensitive data exposure - Flag 4
Type	Web App
Risk	Low
Description	Command curl -v to http://192.168.14.35/About-Rekall.php Shows on the header important information about the website
Founded in or affected host	http://192.168.14.35/About-Rekall.php
Remediation / Recommendation	It is recommended to review and modify the server configuration or application settings to ensure that sensitive information is not exposed in the response headers. This may involve adjusting server configurations or modifying the application code to sanitize or obfuscate sensitive details.
Images	

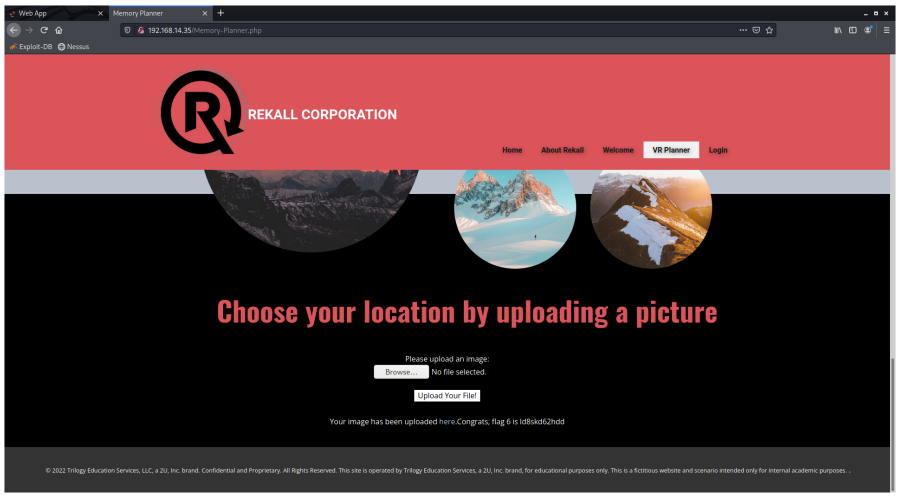
Vulnerability	Findings
Title	Local File Inclusion (LFI) - Flag 5
Type	Web App
Risk	Critical
Description	It is possible to upload any PHP file to the server, in a box with a description only for upload images. Once uploaded, the PHP files can be accessed and executed on the server, enabling them to carry out various malicious activities.
Founded in or affected host	http://192.168.14.35/Memory-Planner.php

Remediation / Recommendation	Strict input validation and security controls should be implemented during the file upload process. The server should only accept specific file types (images) and reject all other file types, especially executable files like PHP. Additionally, file permissions and access controls should be set properly to prevent unauthorized execution of files.
Images	 

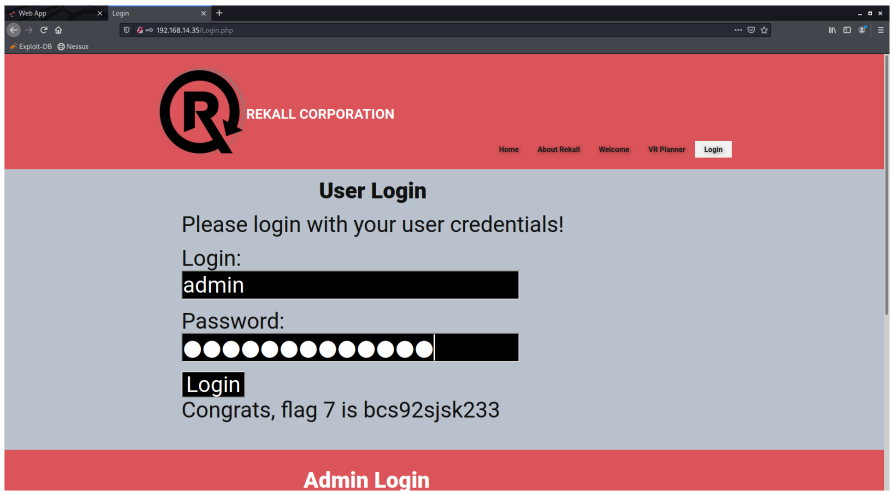
Vulnerability	Findings
Title	LFI (Advanced) - Flag 6
Type	Web App
Risk	Critical
Description	Similar to the previous vulnerability, but there is some input validation for the ".jpg" file. It was only necessary to modify a file to ".jpg.php" to pass over that basic validation.
Founded in or affected host	http://192.168.14.35/Memory-Planner.php
Remediation / Recommendation	Similar to the previous case, Flag 5.

Images	

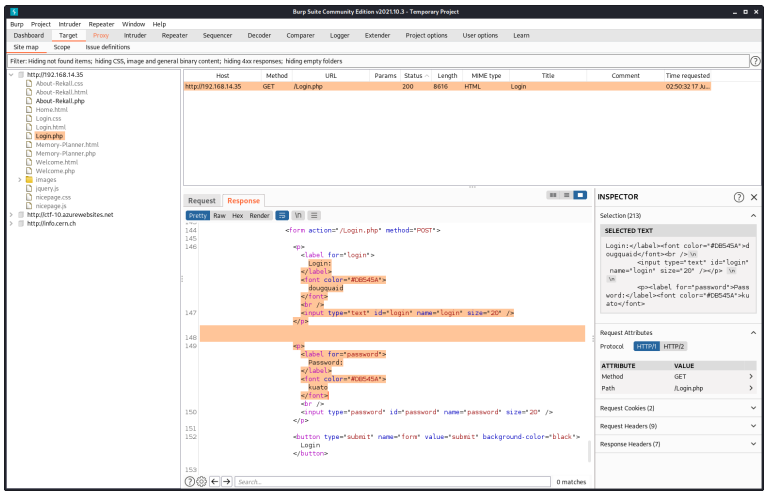
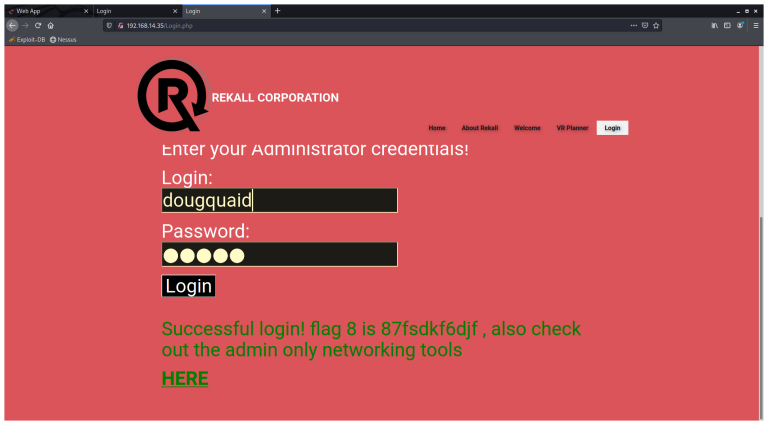




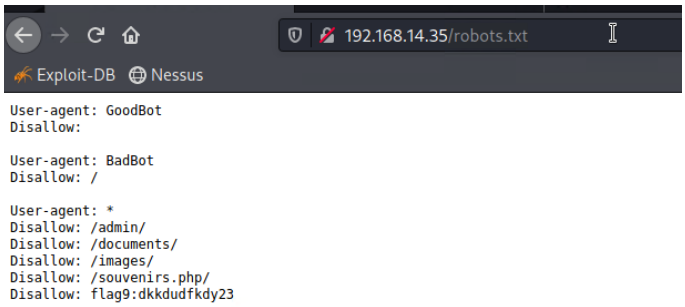
Vulnerability	Findings
Title	SQL Injection - Flag 7
Type	Web App
Risk	Critical
Description	In the password field, admin' OR '1' = '1 was used as a payload. This vulnerability allows the attacker to manipulate the SQL query executed by the application.
Founded in or affected host	http://192.168.14.35/Login.php
Remediation / Recommendation	To address this vulnerability, the application's code must be updated to use parameterized queries or prepared statements to handle user inputs safely. Proper input validation and sanitization techniques should be applied to prevent malicious SQL code injection.

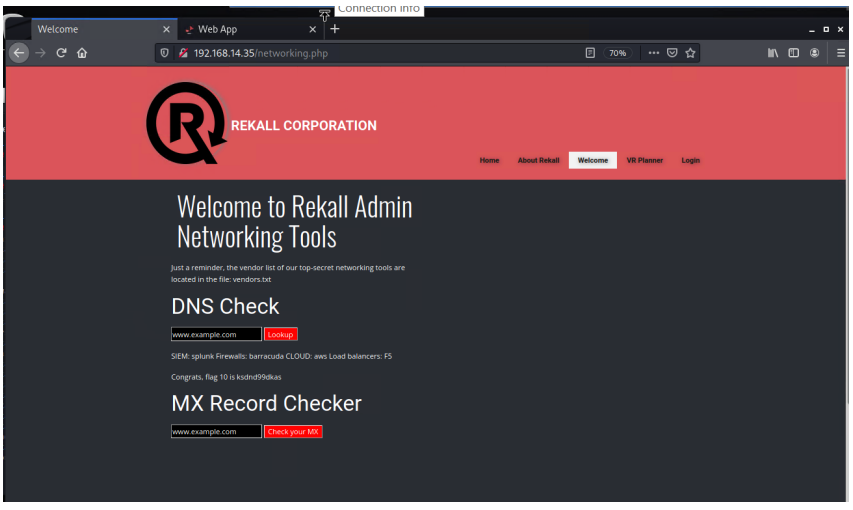
Images	

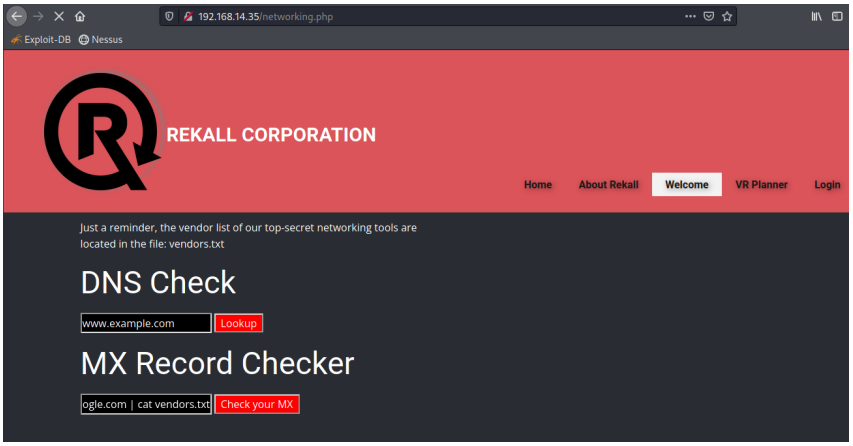
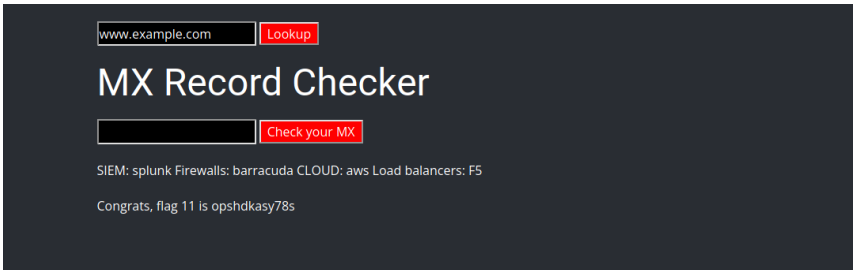
Vulnerability	Findings
Title	Sensitive data exposure - Flag 8
Type	Web App
Risk	High
Description	Using Burp on http://192.168.14.35/Login.php it was possible to visualize login credentials. This is considered sensitive information, is not properly protected and can be viewed by unauthorized individuals. Crealry they were admin credentials. Unauthorized access to administrator accounts can lead to significant data breaches
Founded in or affected host	http://192.168.14.35/Login.php
Remediation / Recommendation	Similar as for Flag 4, sensitive data must be identified beforehand and not expose it to the public. In this case, a revision of the file Login.php would be highly recommended.

Images	 

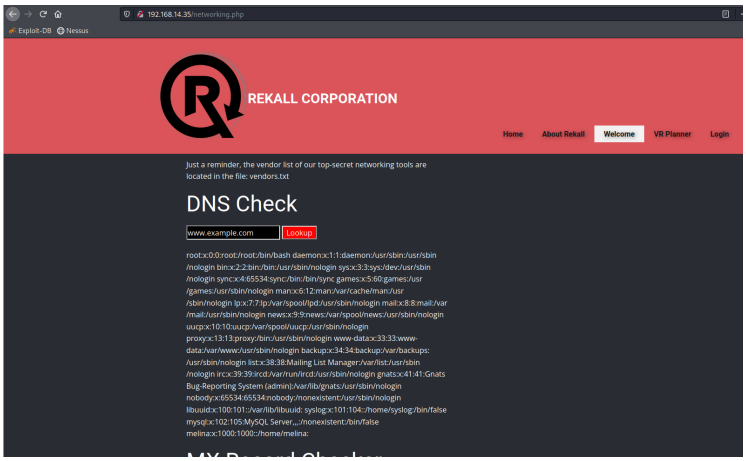
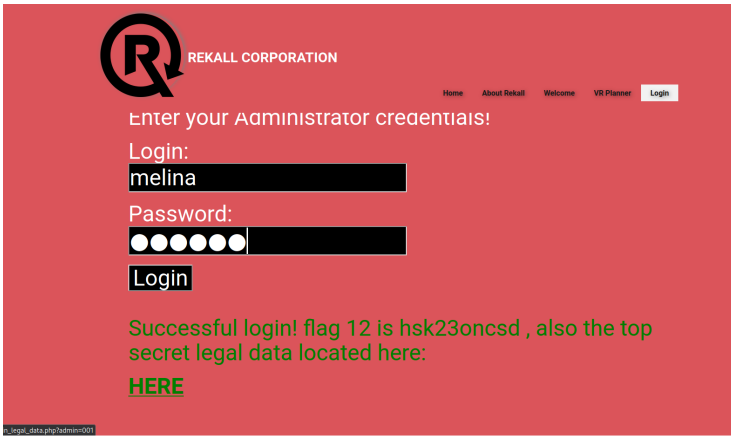
Vulnerability	Findings
Title	Sensitive data exposure - Flag 9
Type	Web App
Risk	Medium
Description	robots.txt file is accessible for the public. The "robots.txt" file is a standard used by websites to communicate with web crawlers and search engine bots, providing instructions on which parts of the site should be crawled or indexed. In this case, the file contains information about additional hidden websites, which should not be publicly disclosed.
Founded in or affected host	http://192.168.14.35/robots.txt
Remediation / Recommendation	Review the content of the "robots.txt" file and ensure that it does not disclose sensitive information, hidden resources, or any other sensitive details. Like in previous findings, sensitive information should not be included in files accessible by the general public.

Images	

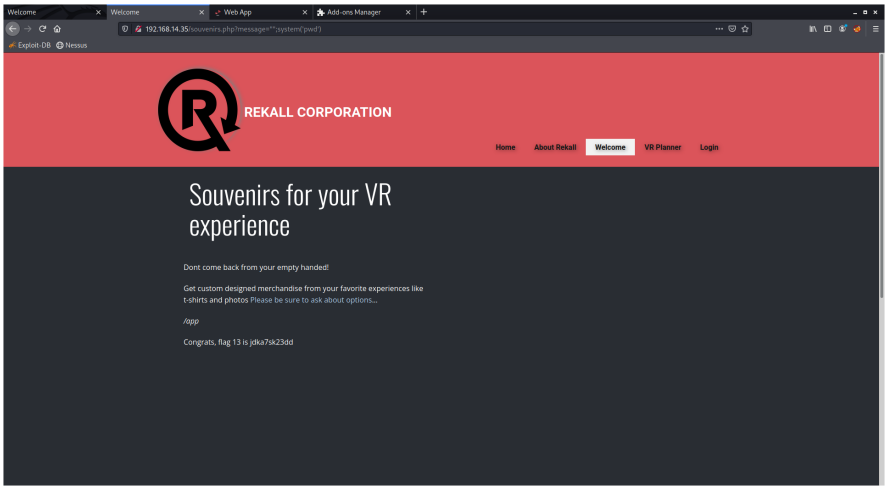
Vulnerability	Findings
Title	Command injection - Flag 10
Type	Web App
Risk	Critical
Description	After finding Flag 8, there was a hint to check a networking tool for the admin. In the DNS Check box field, it is possible to use “www.google.com | cat /etc/vendors.txt” (‘|’ pipe symbol) or other special characters like “&” or “;” and allows an attacker to execute arbitrary commands on the server.
Founded in or affected host	http://192.168.14.35/networking.php
Remediation / Recommendation	Proper input validation and sanitization for this field will be required. User input should not be directly used in command execution. Instead, parameterized queries or prepared statements should be used to ensure safe handling of user input.
Images	

Vulnerability	Findings
Title	Command injection (advanced) - Flag 11
Type	Web App
Risk	Critical
Description	Similar to Flag 10, in the section MX Record Checker it is possible, to used only "|"
Founded in or affected host	http://192.168.14.35/networking.php
Remediation / Recommendation	Same as for Flag 10.
Images	 

Vulnerability	Findings
Title	Brute force attacks - Flag 12
Type	Web App
Risk	Critical (High)
Description	Following findings from Flag 10, it was possible to expose the file passwd using "www.google.com | cat /etc/passwd". This gave access to a user called melina. Brute forcing its password, which in this case, is the same as the username, it was possible to login. This lack of complexity in the

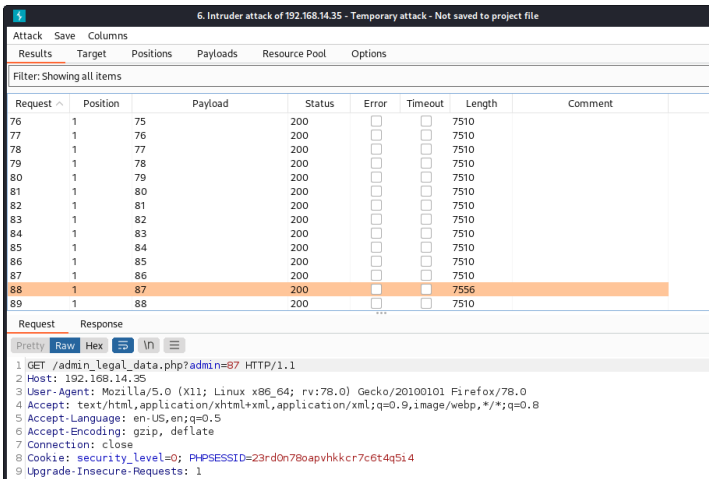
	password successfully gained unauthorized access to the system.
Founded in or affected host	http://192.168.14.35/networking.php
Remediation / Recommendation	Strong passwords policy and multi-factor authentication are strongly recommended. Additionally, account lockout and rate-limiting mechanisms should be put in place to prevent brute force attacks.
Images	 

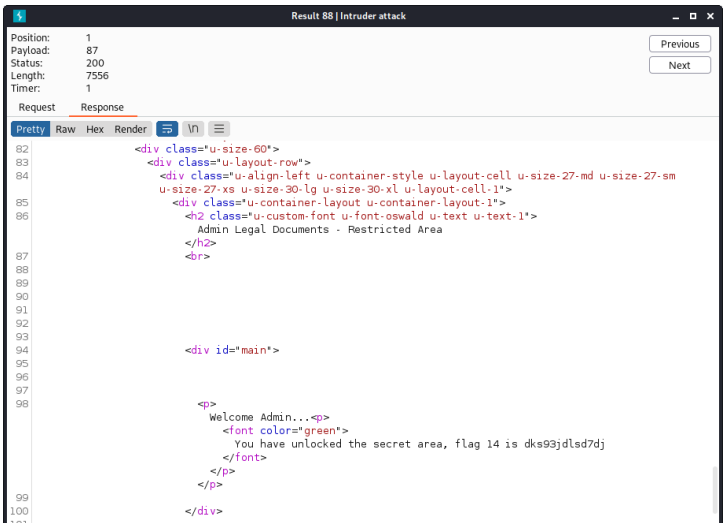
Vulnerability	Findings
Title	PHP injection - Flag 13
Type	Web App
Risk	Critical
Description	Following findings from Flag 9, it was possible to access http://192.168.14.35/souvenirs.php. In this case, the "message" parameter is vulnerable to PHP code injection. A provided payload like “message=””; system(‘pwd’)” executes the “pwd ”command on the server. Eventually, any other PHP command might be executed in the system.
Founded in or affected host	http://192.168.14.35/souvenirs.php

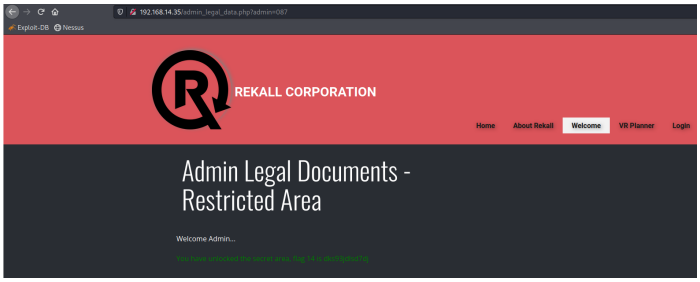
Remediation / Recommendation	Similar to other vulnerabilities, it is crucial to implement strict input validation and output encoding. User-supplied data should never be directly executed as PHP code. Instead, user input should be validated and sanitized before being used in any code execution or evaluation functions.
Images	

Vulnerability	Findings
Title	Session management - Flag 14
Type	Web App
Risk	Critical
Description	Following a hint on Flag 12, accessing http://192.168.14.35/admin_legal_data.php?admin=001, using Burp and intruder, it was possible to payload for sequential id numbers for admin and find another admin (id 87).
Founded in or affected host	http://192.168.14.35/admin_legal_data.php
Remediation / Recommendation	New sessions for users are required to be created for each login, this will avoid accessing other sessions.

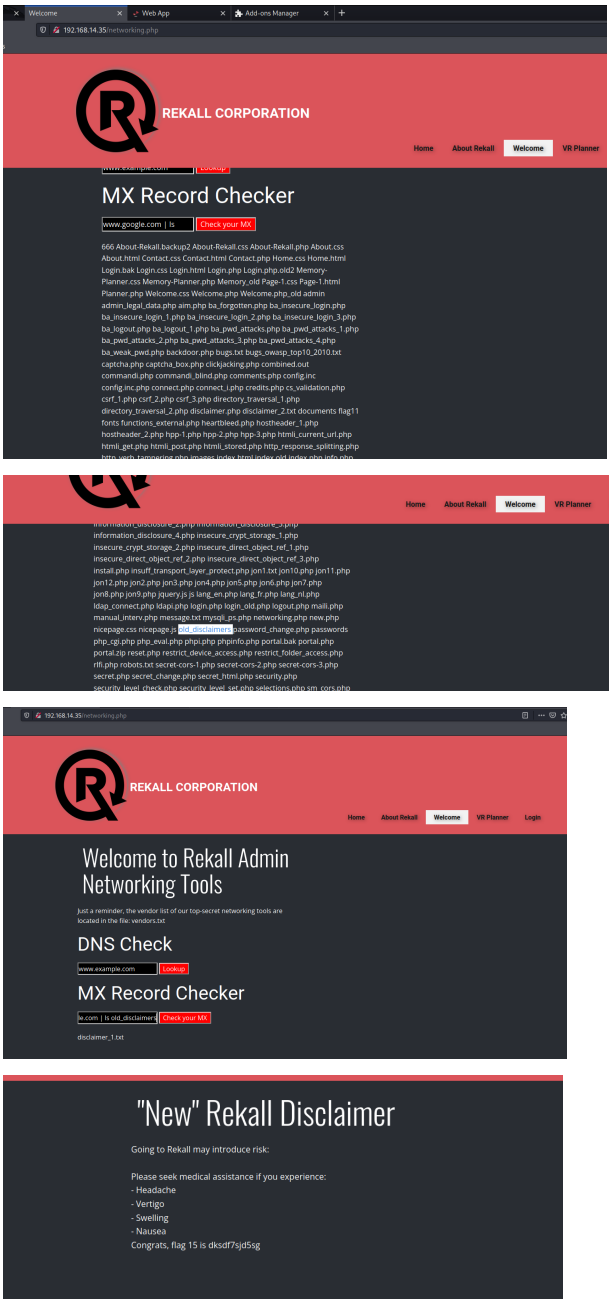
Images	



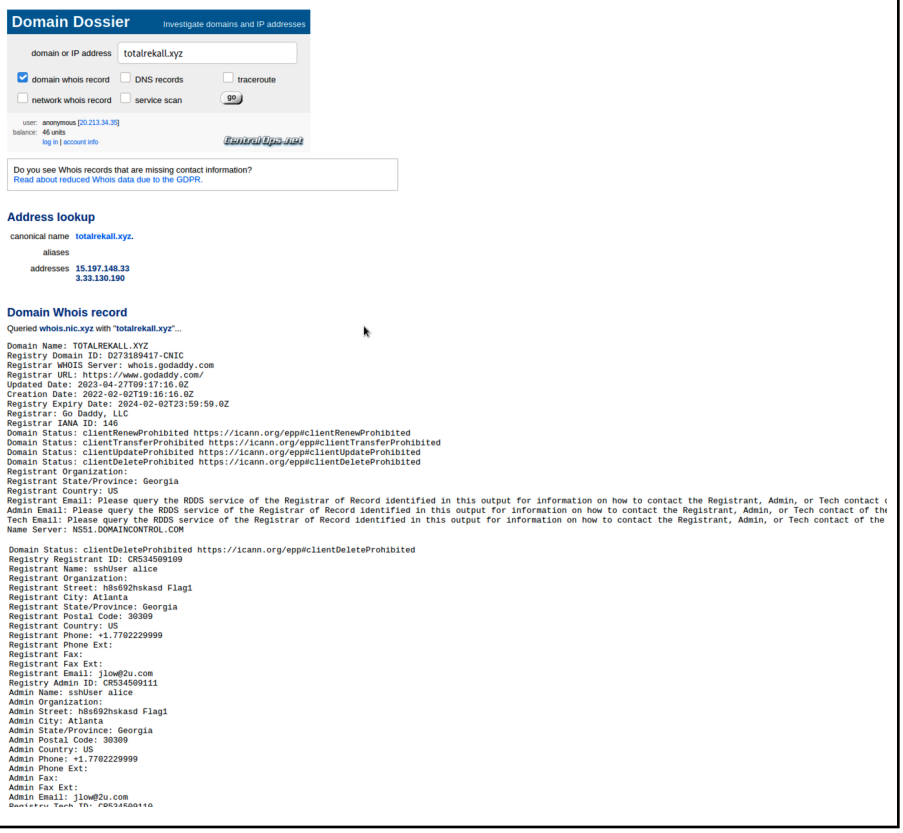


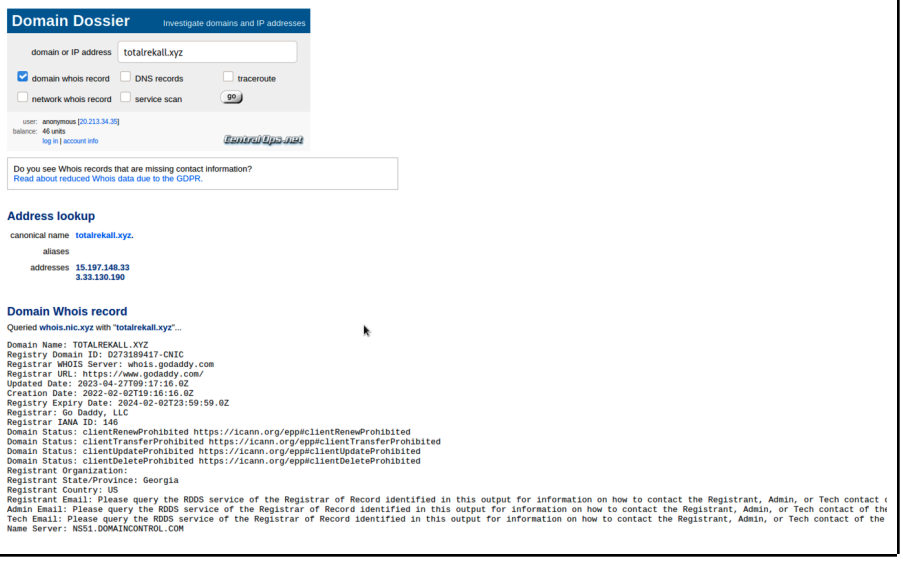


Vulnerability	Findings
Title	Directory traversal - Flag 15
Type	Web App
Risk	Critical
Description	Using findings from Flag 11, it was possible to use a command “www.google.com | ls” to get all the files and directories in the folder. A directory called “old_disclaimers” shows up and using another command “www.google.com | ls old_disclaimers” it was possible to access sensitive

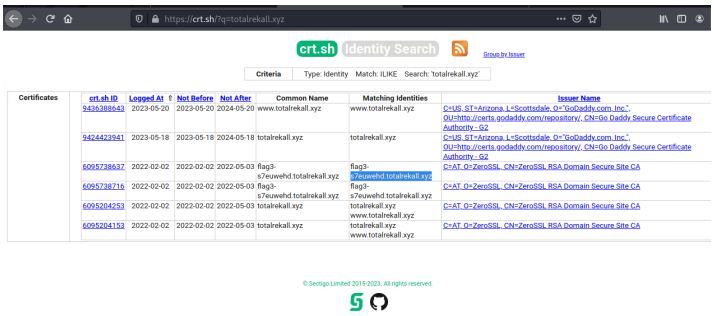
	information. In this case a file called disclaimer_1.txt. Manipulating the URL to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt , the issue was found which might also lead to data breaches, disclosure of sensitive information, and potential legal or compliance issues.
Founded in or affected host	http://192.168.14.35/disclaimer.php
Remediation / Recommendation	Essential to implement strict input validation and enforce proper access controls on file and directory requests. Any user input, such as the "page" parameter in the URL, should be carefully validated and sanitized to prevent directory traversal attempts.
Images	

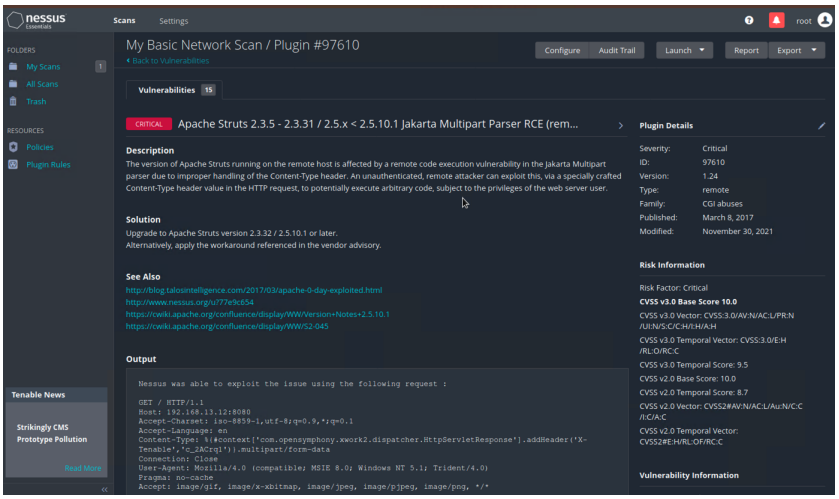
Second penetration test findings

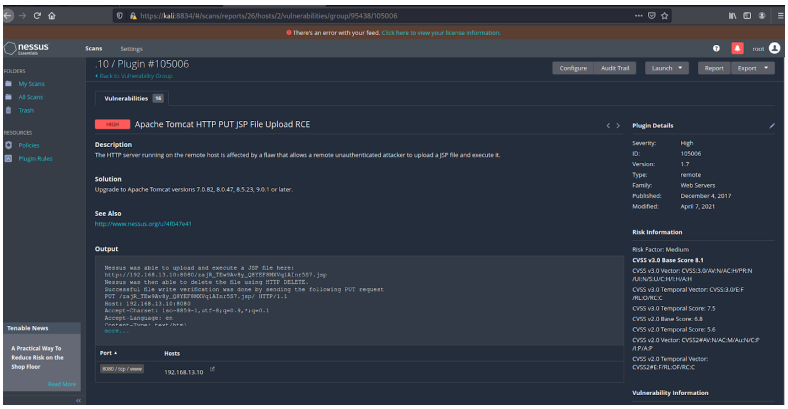
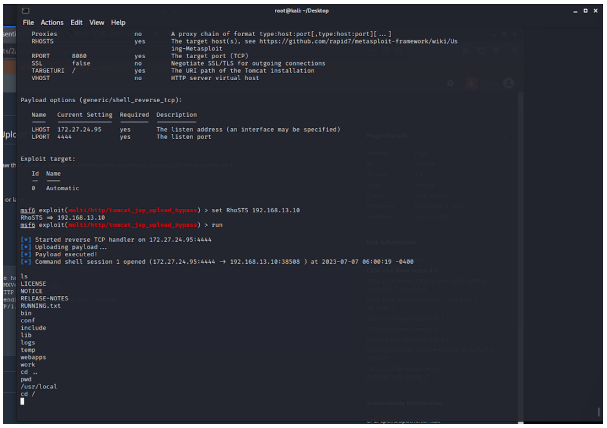
Vulnerability	Findings
Title	Open-source exposed data - Flag 1
Type	Linux OS
Risk	Low
Description	Using http://centralops.net/co/DomainDossier.aspx website with totalrekall.xyz was possible to extract some sensitive data (flag1 and a ssh user) which are exposed to the public.
Founded in or affected host	totalrekall.xyz
Remediation / Recommendation	Similar to Flag 4 in the previous section, amend the data disclosed for the domain. Also clean WHOIS records.
Images	

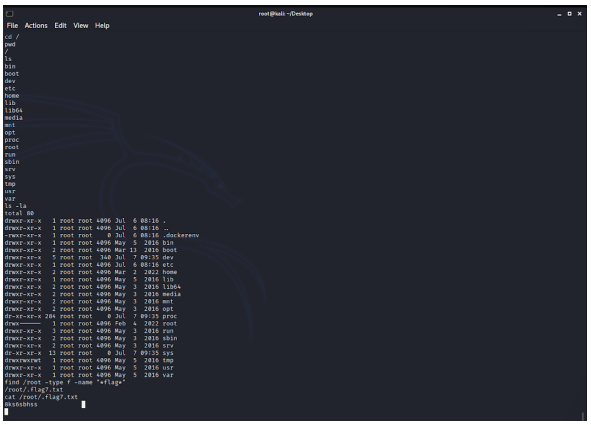
Vulnerability	Findings
Title	Ping totalrekall.xyz - Flag 2
Type	Linux OS
Risk	Low
Description	Ping to totalrekall.xyz exposed the corporate IP address. This could provide valuable information to potential attackers during the reconnaissance phase of an attack.
Founded in or affected host	totalrekall.xyz / 3.33.130.190
Remediation / Recommendation	Ping requests might be blocked (ICMP) or they can be identified by a firewall and create alerts when they become suspicious.
Images	

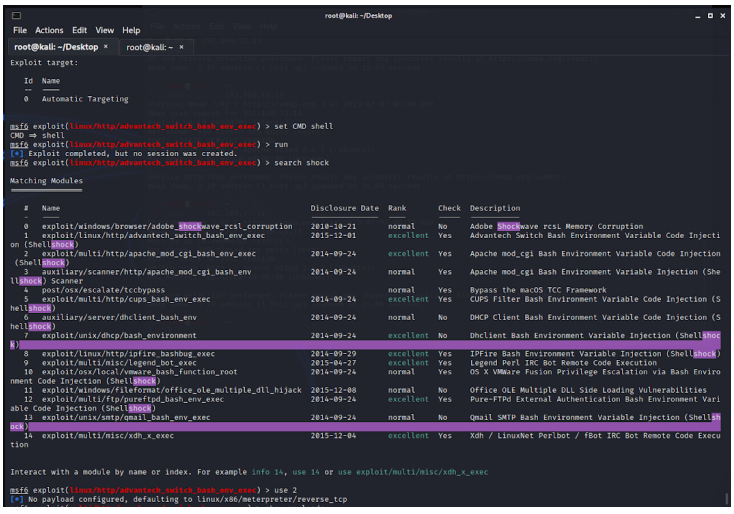
Vulnerability	Findings
Title	Open-source exposed data - Flag 3
Type	Linux OS
Risk	Low
Description	Open source website https://crt.sh/ gives information about the company's certificates.
Founded in or affected host	Certificates for totalrekall.xyz
Remediation / Recommendation	Similar to Flag 4 in the previous section

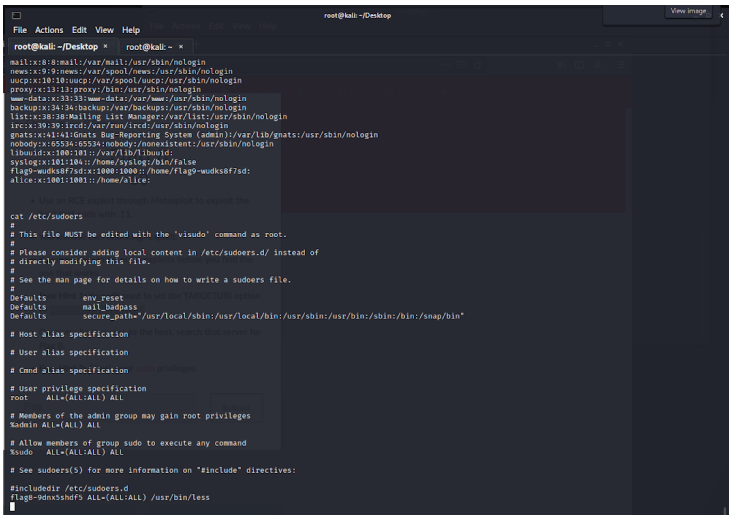
Images	

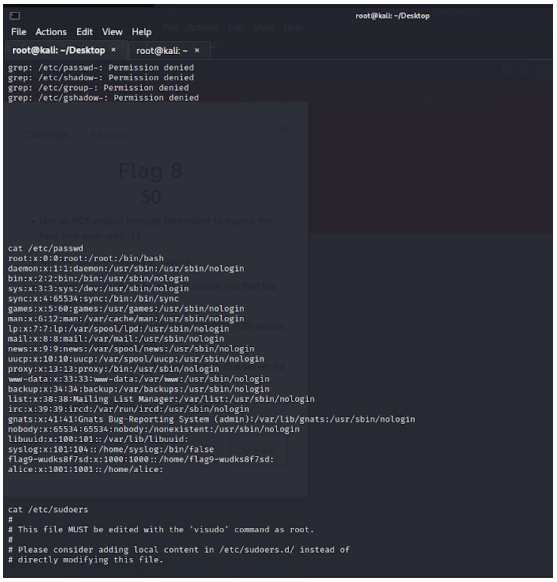
Vulnerability	Findings
Title	Nessus scan - Flag 6
Type	Linux OS
Risk	Critical
Description	Critical vulnerability found in the actual Apache Strut (a web application framework) version using Nessus tool. This could be exploited by attackers allowing them to execute arbitrary code on the server, leading to full system compromise.
Founded in or affected host	192.168.13.12
Remediation / Recommendation	Patch or update software, as often as possible, to avoid attacks.
Images	

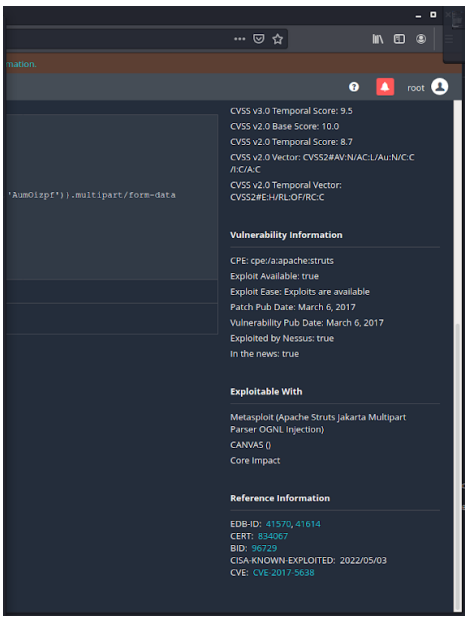
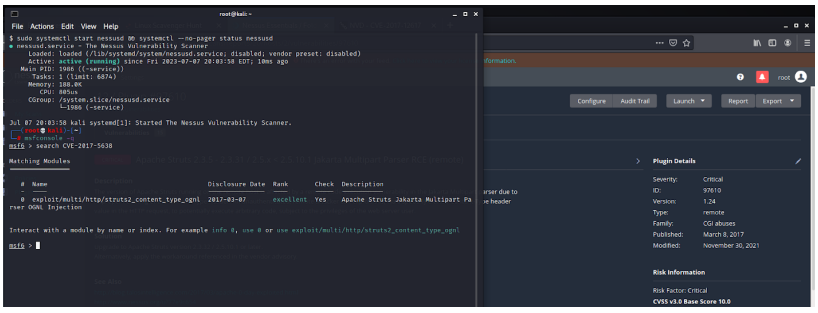
Vulnerability	Findings
Title	RCE exploit Apache Tomcat - Flag 7
Type	Linux OS
Risk	Critical
Description	Using Nessus to scan 192.168.13.10 it was possible to find a critical vulnerability on Apache Tomcat for a remote code execution (CVE-2017-12617). Using Metasploit, it was possible to find an exploit suitable for this vulnerability (multi/http/tomcat_jsp_upload_bypass) This basically allows uploading a malicious JSP file to the server and executing arbitrary code with the privileges of the Tomcat service. By setting up the right parameters, it was possible to create a meterpreter session, writing “shell”, a linux command line appeared and the possibility to execute commands.
Founded in or affected host	192.168.13.10
Remediation / Recommendation	Similar to Flag 6, it is very important to keep software updated with the latest security patches. Also vulnerability scanning, and intrusion detection systems are essential components of maintaining a secure web server environment.
Images	 

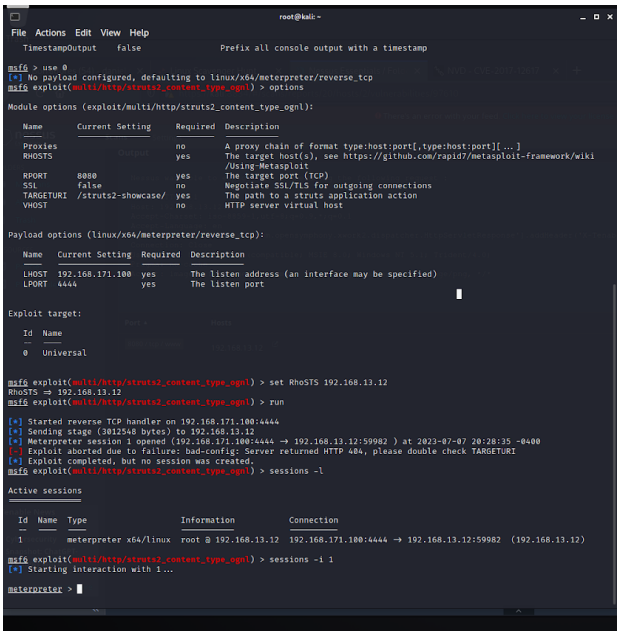


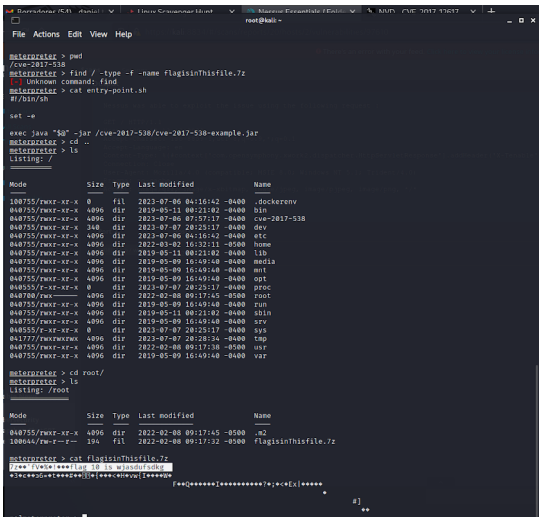
Vulnerability	Findings
Title	Shellshock- Flag 8
Type	Linux OS
Risk	Critical
Description	Using Metasploit, searching for shock, it was possible to find an exploit for this vulnerability. Shellshock is a critical vulnerability in the Bash shell which allows an attacker to execute arbitrary commands remotely. Similar steps to Flag 7, in this case the exploit was "exploit/multi/http/apache_mod_cgi_bash_env_exec", by setting up the right parameters, with a target machine on 192.168.13.11 (Rhost), running a command line in the victim was possible. Access to "cat /etc/sudoers" gave extra details.
Founded in or affected host	192.168.13.11
Remediation / Recommendation	Similar to Flag 6, it is very important to keep software updated with the latest security patches.
Images	



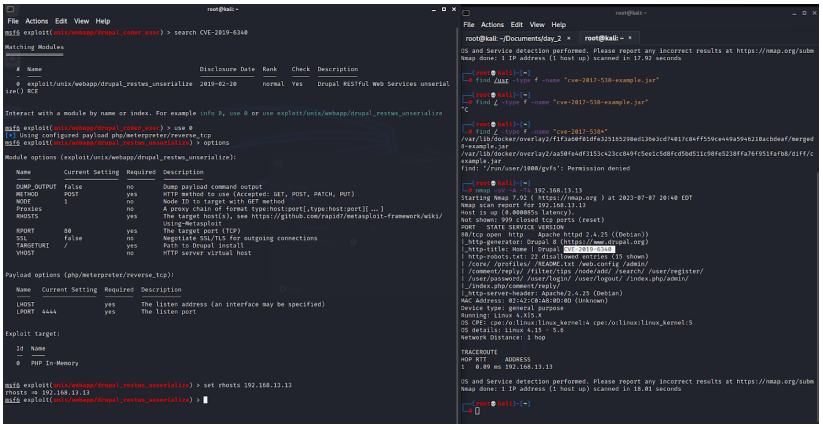
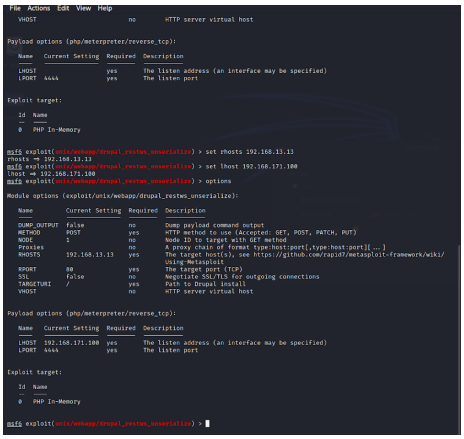
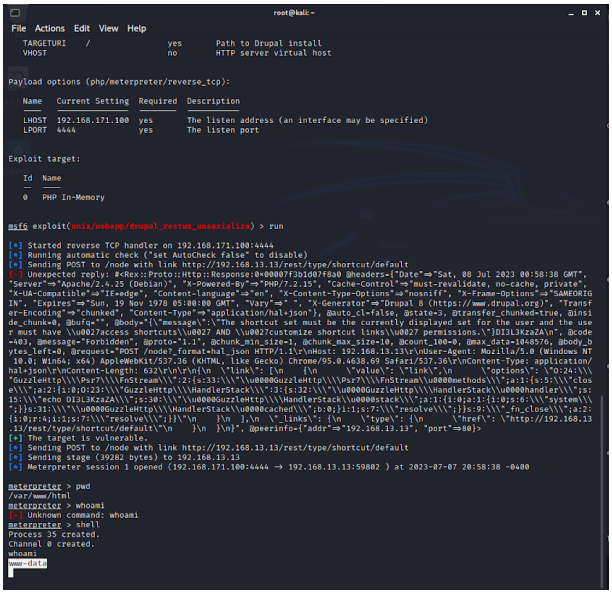
Vulnerability	Findings
Title	Additional findings in host - Flag 9
Type	Linux OS
Risk	Critical
Description	Following the same steps as in Flag 8, it was possible to access the content for the file “/etc/passwd”.
Founded in or affected host	192.168.13.11
Remediation / Recommendation	Same as Flag 6.
Images	

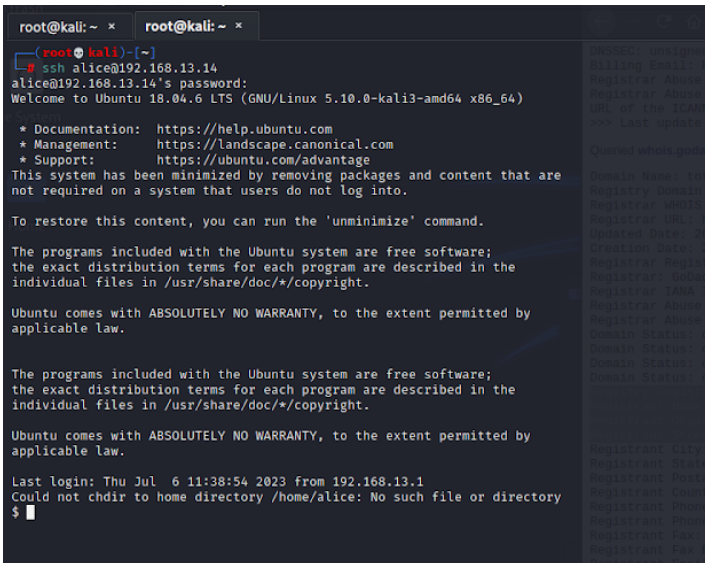
Vulnerability	Findings
Title	RCE exploit Struts - Flag 10
Type	Linux OS
Risk	Critical
Description	In Nessus a specific vulnerability was shown, CVE-2017-5638 which llows remote attackers to execute arbitrary code on the target server. A search in Metasploit gives the exploit. Similar steps to Flag 7, but only accessing meterpreter, it was possible to access folder /root. The "/root" directory typically contains sensitive files and configurations accessible only to the root user. Important to mention that this exploit needed some additional configuration, shown in the images.There are required to use the active session created after running the exploit.
Founded in or affected host	192.168.13.12
Remediation / Recommendation	Same as Flag 6, immediately update to the latest patched version.
Images	 

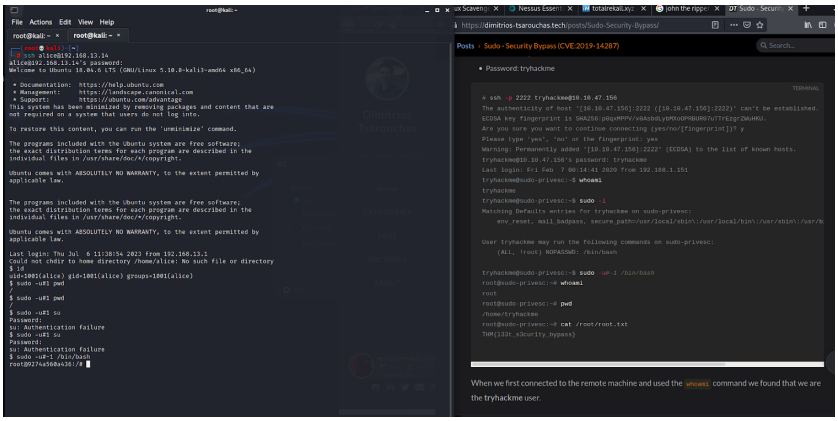


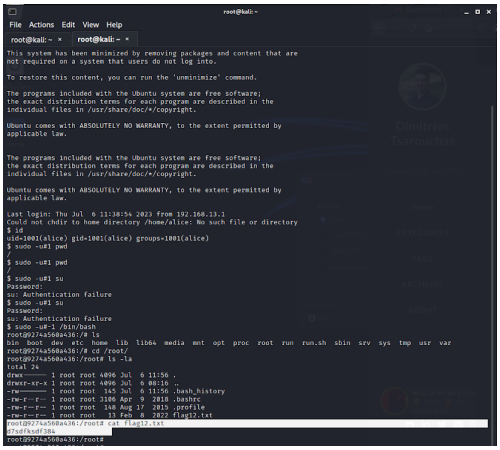


Vulnerability	Findings
Title	RCE exploit Drupal - Flag 11
Type	Linux OS
Risk	Critical
Description	A “nmap -Sv -A -T4 192.168.13.13” scan shows the vulnerability code for this machine (CVE-2019-6340). Drupal is a content management system and this vulnerability allows remote attackers to execute arbitrary code on the target system. A search in Metasploit for that code, shows the appropriate exploit. Once the session is open with a meterpreter, “shell” command gives access to the command line. A “whoami” gives the information needed.
Founded in or	192.168.13.13

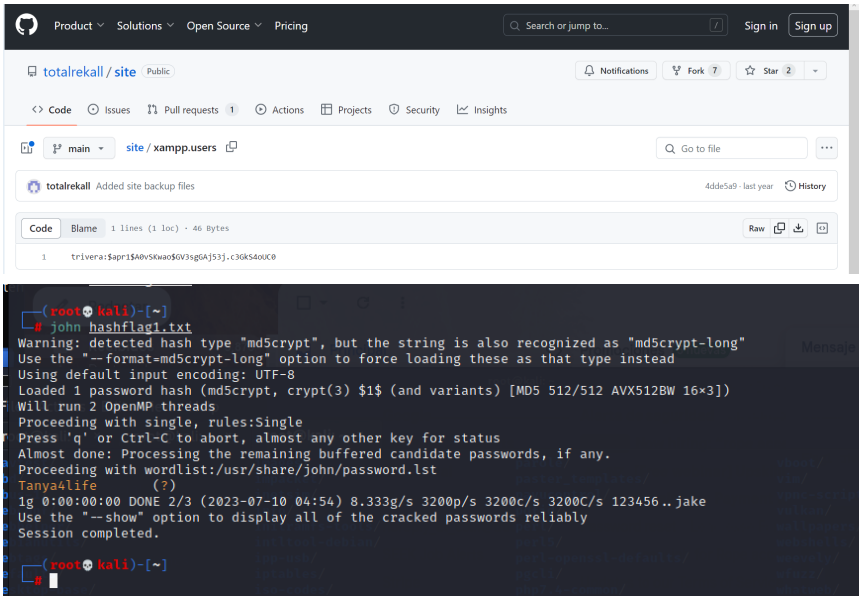
affected host	
Remediation / Recommendation	Same as Flag 6, immediately update to the latest patched version.
Images	  

Vulnerability	Findings
Title	SSH user - Flag 12
Type	Linux OS
Risk	Critical
Description	From Flag 1, the ssh user “alice” was recovered. Password in this case was identically as the username. It was possible to use CVE-2019-14287 which is a vulnerability that affects the sudo command in Unix-based systems and allows unauthorized users to execute commands as root or other privileged users. Based on the information found in the website https://hackersploit.org/sudo-security-bypass-vulnerability-cve-2019-14287 a method to exploit the vulnerability. Basically any user can run commands using “sudo -u#-1”. After applied “sudo -u#-1 /bin/bash” a root console was possible to access. Finding was in the /root folder.
Founded in or affected host	192.168.13.14
Remediation / Recommendation	Same as Flag 6, security updates and patches to the sudo package.
Images	

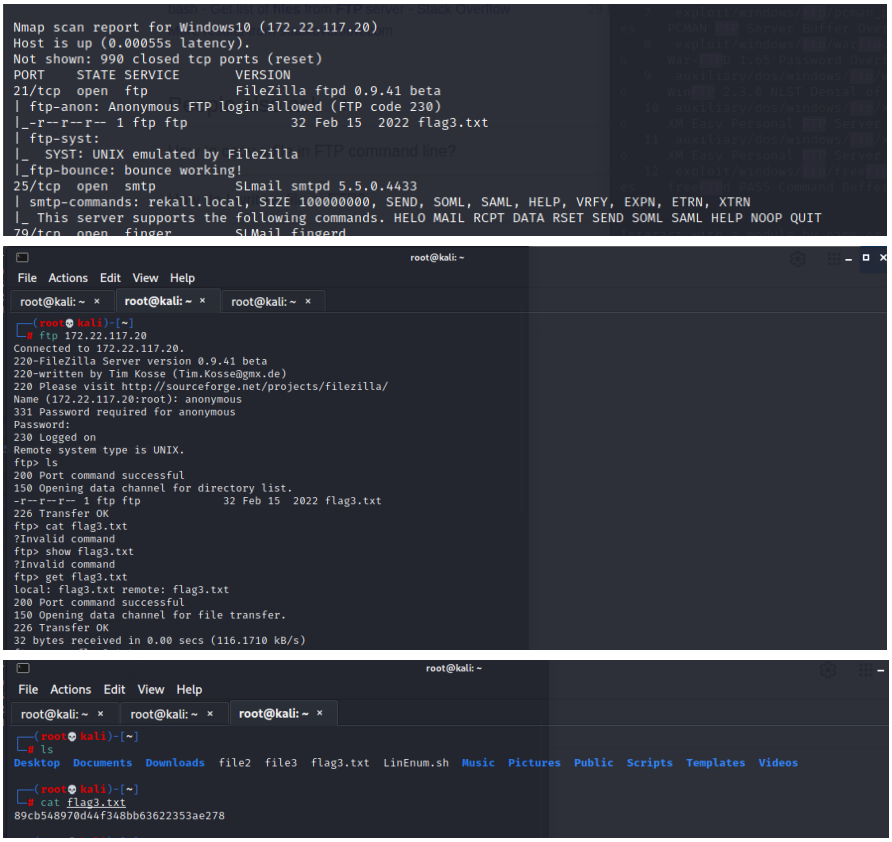

Posts › Sudo - Security Bypass (CVE:2019-14287)
Password: tryhackme
When we first connected to the remote machine and used the whoami command we found that we are the tryhackme user.



Third penetration test findings

Vulnerability	Findings
Title	Totalrekall GitHub page - Flag 1
Type	Windows OS
Risk	Medium
Description	User credentials found in the repository open to the public. In this case, it was found in the file xampp.users. User is trivera and to crack the hash ($apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0) John the Ripper was used.
Founded in or affected host	https://github.com/totalrekall
Remediation / Recommendation	Going back to Flag 4, sensitive data should not be exposed to the public.
Images	

Vulnerability	Findings
Title	Nmap scan FTP - Flag 3
Type	Windows OS
Risk	Medium
Description	Using a scan “nmap -sV 172.22.117.0/24”, it was possible to get information about a FTP server on port 21 running in the machine 172.22.117.20 with anonymous login allowed.

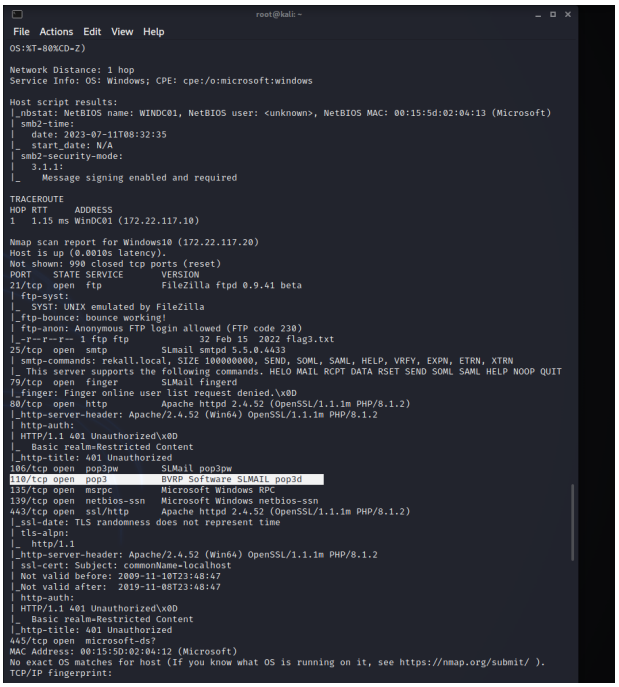
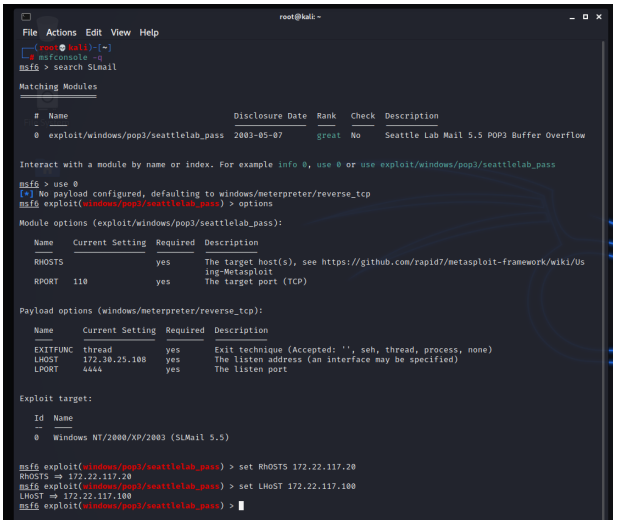
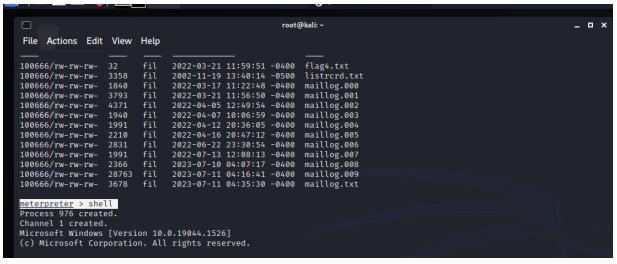
	After connecting to the machine through FTP with an anonymous username and no password, it was possible to download a specific file (flag3.txt). Content of that file was displayed on the local machine.
Founded in or affected host	172.22.117.20
Remediation / Recommendation	FTP is a protocol barely used in the present, basically because connection is not secure, data is transmitted in clear text. Additionally, allowing a user with no credentials represents a danger for any system. It is recommended to use a secure protocol like SSH and allow only specific users to connect to the server.
Images	

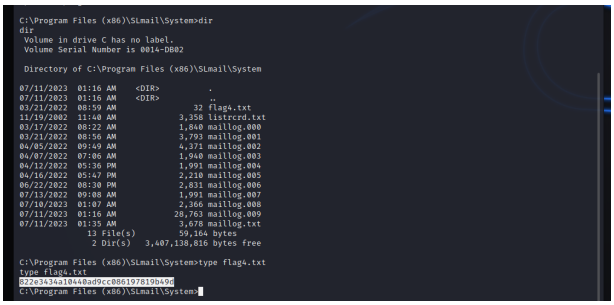
```
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00055s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
|_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 fingerd
```

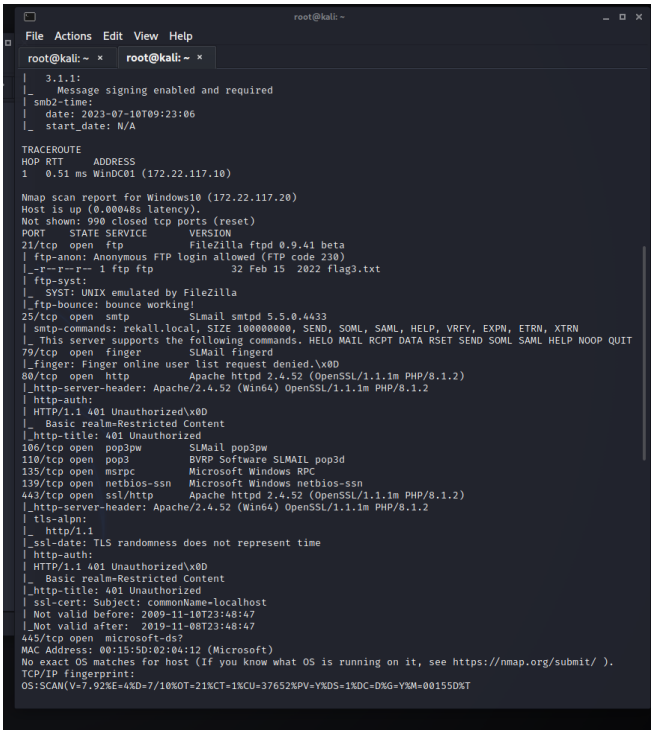
```
root@kali: ~
File  Actions  Edit  View  Help
root@kali: ~ ×   root@kali: ~ ×   root@kali: ~ ×
┌──(root㉿kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> cat flag3.txt
?Invalid command
ftp> show flag3.txt
?Invalid command
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (116.1710 kB/s)
```

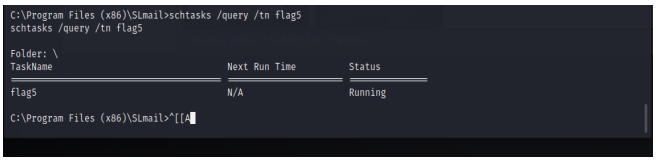
```
root@kali: ~
File  Actions  Edit  View  Help
root@kali: ~ ×   root@kali: ~ ×   root@kali: ~ ×
┌──(root㉿kali)-[~]
└─# ls
Desktop  Documents  Downloads  file2  file3  flag3.txt  LinEnum.sh  Music  Pictures  Public  Scripts  Templates  Videos

┌──(root㉿kali)-[~]
└─# cat flag3.txt
89cb548970d44f348bb63622353ae278
```

Vulnerability	Findings
Title	Nmap scan SLMAIL- Flag 4
Type	Windows OS
Risk	Critical
Description	Using a scan “nmap -sV 172.22.117.0/24”, it was possible to identify a program, SLMAIL or Seattle Lab Mail, running on port 110 in a Windows machine (IP 172.22.117.20). Using Metasploit, searching for an exploit for that specific program, it was possible to find one. By setting up the right options, running the exploit, a remote connection to the machine was possible. After running the

	command “shell” a DOS console appeared. Going through some of the folders, a sensitive file was found (flag4.txt).
Founded in or affected host	172.22.117.20
Remediation / Recommendation	Same as Flag 6 from the previous section, keeping the systems updated and in this specific case, patch the email servers for the company.
Images	  



Vulnerability	Findings
Title	Scheduled task - Flag 5
Type	Windows OS
Risk	Critical
Description	Following a previous exploit in Flag 4, it was possible to use meterpreter with the command “schtasks /query” to show all the tasks in the 172.22.117.20, and from there identified flag5 as a task. Using “schtasks /query /tn flag5 /v” it is possible to get the value.
Founded in or affected host	172.22.117.20
Remediation / Recommendation	Same as for the previous vulnerability Flag 4, once connection to a machine is completed, accessing any kind of files is possible.
Images	


C:\Program Files (x86)\SLmail>schtasks /query /tn flag5
schtasks /query /tn flag5
Folder: \
TaskName Next Run Time Status
flag5 N/A Running
C:\Program Files (x86)\SLmail>^[[A


root@kali: ~
File Actions Edit View Help
root@kali: ~
root@kali: ~
root@kali: ~
C:\Program Files (x86)\SLmail>type flag5
type flag5
The system cannot find the file specified.
C:\Program Files (x86)\SLmail>schtasks /query /tn flag5 /v
schtasks /query /tn flag5 /v
Folder: \
HostName TaskName Next Run Time Status Logon Mode
Last Run Time Last Result Author Task To Run Start In
Comment Scheduled
Task State Idle Time Power Management Run As User
Delete Task If Not Rescheduled Stop Task If Runs X Hours and X Mins Schedule
Schedule Type Start Time Start Date End Dat
e Days Months Repeat: Every Re
peat: Until: Time Repeat: Until: Duration Repeat: Stop If Still Running
WIN10 flag5 N/A Ready Interactive/Background
7/10/2023 2:53:45 AM 1 WIN10\sysadmin C:\Windows\System32\WindowsPowerShell\v1.0\powersh N/A
54fa8cd5c1354adc9214969d716673f5
Enabled
Only Start If Idle for 1 minutes, If Not Stop On Battery Mode ADMBob
Disabled 72:00:00 Scheduling data is not
available in this format. At logon time N/A N/A N/A
N/A N/A N/A N/A N/
A N/A N/A
0
72:00:00 Scheduling data is not
available in this format. At idle time N/A N/A N/A
N/A N/A N/A N/A N/
A N/A N/A