

Reconnaissance Fundamentals – Exam

Execution Summary

- You are hired to perform a digital investigation for the following target:
185.218.124.165
- This is a template for **uploading your screenshots and data**.
- The exam objective is to utilize active and passive reconnaissance to identify stored online ssh private key.
- Finding exposed public ssh key does not complete the exam objective.
- You are free to report vulnerabilities if you find any along the way. Each vulnerability gives extra points.
- Not all vulnerabilities are giving the same number of points.
- The order of the vulnerabilities **DO NOT MATTER**, nor the tools used to get to them.
- You are free to upload multiple **screenshots** for each **vulnerability**, including your **path** and how did you **find** it.
- **No network breach or local privilege escalation is needed!**

Do not overstress or overcomplicate it. The time is enough, you can do it.

Scope

Scope is open, meaning you can perform your own information gathering and you are free to enumerate the target for vulnerabilities from every angle.

Appendix

This is the data section. Make sure to **detail each finding**. The overall vulnerability count is unknown, try to **find as much as possible** while following the **main objective**. It is a good practice to **explain your finding**.

When describing your finding, make sure to be as clear as possible by **answering** the following **questions**:

1. What is the vulnerability I found?
2. How did I find it?
3. What "bad" can happen, what risk does it carry?

After the description, make sure to **drop** your **screenshots** below.

Vulnerability 1

Description:
-> What is the vulnerability I found?

The target is running an outdated version of OpenSSH (8.2p1 which is known to contain multiple medium-severity vulnerabilities.

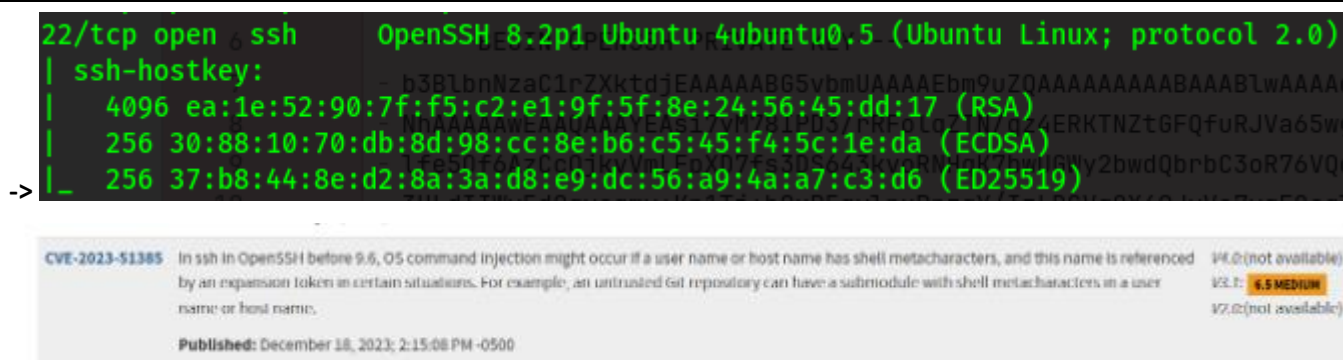
How did I find it?

I scanned the host with `nmap -sV` which revealed port 22/tcp open running OpenSSH 8.2p1. I cross-referenced this version with known CVEs from NIST and Rapid7.

What bad can happen and what risk does it carry?

Attackers can exploit CVE-2023-51385 to execute arbitrary commands via manipulated hostnames, or CVE-2023-48795 (Terrapin) to weaken SSH channel encryption. This exposes the system to unauthorized access and data tampering.

Screenshots:



Vulnerability 2

Description:

-> What is the vulnerability I found?

This vulnerability affects the scp client when using the -rp flag for recursive copies. OpenSSH 8.2p1 does not properly validate file overwrite attempts during these operations. A malicious SCP server can force the client to overwrite arbitrary files on the local system.

How did I find it?

I scanned the target IP 185.218.124.165 using nmap -sV -p 22, which identified the SSH service as OpenSSH 8.2p1. I then cross-referenced this version in the CVE database and security advisories, where CVE-2020-12062 was listed as a client-side vulnerability.

What bad can happen and what risk does it carry?

An attacker operating a malicious SCP server can overwrite critical local files on the client machine, potentially causing data loss, injecting backdoors, or disabling security configurations. This is a serious risk in environments where automated or scripted SCP transfers are used.

Screenshots:

->

```
(fc0d3x_guest@kali)~[~/reconexam]
$ sudo nmap -sV -p 22 185.218.124.165
[sudo] password for fc0d3x_guest:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-15 12:03 EEST
Nmap scan report for vmi1074776.contaboserver.net (185.218.124.165)
Host is up (0.0056s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

Vulnerability Details : CVE-2020-12062

The scp client in OpenSSH 8.2 incorrectly sends duplicate responses to the server upon a utimes system call failure, which allows a malicious unprivileged user on the remote server to overwrite arbitrary files in the client's download directory by creating a crafted subdirectory anywhere on the remote server. The victim must use the command scp -rp to download a file hierarchy containing, anywhere inside, this crafted subdirectory. NOTE: the vendor points out that "this attack can achieve no more than a hostile peer is already able to achieve within the scp protocol" and "utimes does not fail under normal circumstances."

Published 2020-06-01 16:15:14 Updated 2024-08-04 12:15:43 Source MITRE

View at NVD [NVD](#), CVE.org [CVE.org](#)

Vulnerability 3

Description:

-> What is the vulnerability I found?

A flaw in OpenSSH's algorithm negotiation process results in a discrepancy that can be observed by a man-in-the-middle (MiTM) attacker. This makes it possible to infer information about the client's supported algorithms.

How did I find it?

After identifying the version (OpenSSH 8.2p1) using nmap -sV, I found that this CVE was listed in multiple security databases affecting versions up to 8.4. The vulnerability affects the SSH protocol handshake process.

What bad can happen and what risk does it carry?

An attacker with network access could perform a MiTM attack and infer the supported algorithms of the client. This could help in future downgrade or brute-force attacks by tailoring the attack to the client's capabilities, slightly weakening the overall security posture.

Screenshots:

->

Vulnerability Details : CVE-2020-14145

The client side in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached by the client). NOTE: some reports state that 8.5 and 8.6 are also affected.

Published 2020-06-29 18:15:12 Updated 2022-04-28 19:34:18 Source MITRE

View at NVD [NVD](#), CVE.org [CVE.org](#)

Vulnerability 4

Description:

-> What is the vulnerability I found?

The Terrapin attack exploits weaknesses in SSH extension negotiation. By altering or stripping parts of the initial handshake, a MiTM attacker can bypass integrity checks or disable key security features.

How did I find it?

The Nmap scan showed OpenSSH 8.2p1 in use, which is vulnerable to this flaw. The CVE details confirmed that 8.2p1 does not implement countermeasures against this kind of manipulation, making it vulnerable to Terrapin.

What bad can happen and what risk does it carry?

This allows downgrade attacks, where optional security features like strict key exchange or rekeying are silently disabled. The attacker doesn't decrypt traffic directly, but weakens future protection, opening the door for additional exploits or surveillance.

Screenshots:

->

Vulnerability Details : CVE-2023-48795 Potential exploit

The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGO before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KITTY through 0.76.113, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

Published 2023-12-18 16:15:11 Updated 2025-05-23 02:24:59 Source MITRE

View at NVD[®], CVE.org[®]

Vulnerability 5

Description:

-> What is the vulnerability I found?

This is a privilege escalation issue caused by improper handling of supplemental groups when executing the AuthorizedKeysCommand or AuthorizedPrincipalsCommand. These helpers may inherit unintended privileges.

How did I find it?

After confirming OpenSSH 8.2p1 is running on the target via Nmap, I researched all known CVEs tied to this version and identified CVE-2021-41617. It specifically affects how commands are executed under certain privilege contexts.

What bad can happen and what risk does it carry?

A misconfigured server could allow a user to elevate their privileges or access keys they should not have access to. This opens the door to unauthorized SSH access or lateral movement inside a system or network.

Screenshots:

->

Vulnerability Details : CVE-2021-41617

sshd in OpenSSH 6.2 through 8.x before 8.8, when certain non-default configurations are used, allows privilege escalation because supplemental groups are not initialized as expected. Helper programs for AuthorizedKeysCommand and AuthorizedPrincipalsCommand may run with privileges associated with group memberships of the sshd process, if the configuration specifies running the command as a different user.

Published 2021-09-26 19:15:07 Updated 2023-12-26 04:15:08 Source [MITRE](#)

View at [NVD](#), [CVE.org](#)

Vulnerability category: [Gain privilege](#)

Vulnerability 6

Description:

-> CVE-2025-21490 – InnoDB Denial of Service

What is the vulnerability?

A flaw in the InnoDB engine allowing a high-privileged network attacker to cause the server to hang or crash.

How did I find it?

Listed in MariaDB's security fixes (fixed in 11.7.2)

Risk:

Complete Denial-of-Service → service downtime, denial of access for all users.

Screenshots:

CVE-ID	
CVE-2025-21490	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 8.0.40 and prior, 8.4.3 and prior and 9.1.0 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none">MISC:Oracle AdvisoryURL:https://www.oracle.com/security-alerts/cvujjan2025.html	
Assigning CNA	
Oracle	
Date Record Created	
20241224	Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20241224)	

CVE-2025-21490

Publication date 21 January 2025


Last updated 31 March 2025

UBUNTU PRIORITY

 **Medium**

[Why this priority?](#)

CVSS 3 SEVERITY SCORE

 **4.9 · Medium**

[Score breakdown](#)

Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 8.0.40 and prior, 8.4.3 and prior and 9.1.0 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Vulnerability 7

Description:

1. -> **CVE-2025-30722 – MySQL/MariaDB Client Vulnerability**

2. What is the vulnerability?

A flaw in the client-side component could lead to unauthorized access to sensitive data.

3. How did I find it?

Identified in MariaDB's "Security Vulnerabilities Fixed" list.

4. Risk:

Potential data leakage if a client is exploited by a malicious or manipulated server.

Screenshots:

CVE-2025-30722 Detail

AWAITING ANALYSIS

This CVE record has been marked for NVD enrichment efforts.

Description

Vulnerability in the MySQL Client product of Oracle MySQL (component: Client: mysqldump). Supported versions that are affected are 8.0.0-8.0.41, 8.4.0-8.4.4 and 9.0.0-9.2.0. Difficult to exploit vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Client. Successful attacks of this vulnerability can result in unauthorized access to critical data or complete access to all MySQL Client accessible data as well as unauthorized update, insert or delete access to some of MySQL Client accessible data. CVSS 3.1 Base Score 5.9 (Confidentiality and Integrity impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:L/A:N).

Vulnerability 8

Description:

CVE-2025-30693 – Server-Side Security Flaw

What is a vulnerability?

A vulnerability in the MariaDB server allowing potential compromise of data integrity or unwanted operations.

How did I find it?

Also included in MariaDB security advisories for series 11.4 & 10.x

Risk:

Potential data corruption, privilege escalation, or unauthorized execution.

Screenshots:

CVE-2025-30693 Detail

Description

Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 8.0.0-8.0.41, 8.4.0-8.4.4 and 9.0.0-9.2.0. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server as well as unauthorized update, insert or delete access to some of MySQL Server accessible data. CVSS 3.1 Base Score 5.5 (Integrity and Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:H).

Vulnerability 9

Description:

5. -> CVE-2024-21096 – Earlier MariaDB Server Vulnerability

6. What is a vulnerability?

A flaw in MariaDB versions including 11.7 that could be exploited by attackers

7. How did I find it?

Noted in the fixed vulnerabilities list for 11.7.

8. Risk:

Depending on details, could allow unauthorized access, denial-of-service, or data tampering.

Screenshots:

CVE-2024-21096 Detail

Description

Vulnerability in the MySQL Server product of Oracle MySQL (component: Client: mysqldump). Supported versions that are affected are 8.0.36 and prior and 8.3.0 and prior. Difficult to exploit vulnerability allows unauthenticated attacker with logon to the infrastructure where MySQL Server executes to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized update, insert or delete access to some of MySQL Server accessible data as well as unauthorized read access to a subset of MySQL Server accessible data and unauthorized ability to cause a partial denial of service (partial DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Confidentiality, Integrity and Availability impacts). CVSS Vector: (CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L).

Vulnerability 10

Description:

CVE-2023-52971 – Legacy MariaDB Vulnerability

What is a vulnerability?

A flaw affecting MariaDB 11.4 & 10.11, likely allowing data corruption or crash conditions

How did I find it?

Present in MariaDB's fixed CVE list for 11.x series.

Risk:

Could lead to service instability or sensitive data manipulation, though severity may vary.

Screenshots:

CVE-2023-52971 Detail

AWAITING ANALYSIS

This CVE record has been marked for NVD enrichment efforts.

Description

MariaDB Server 10.10 through 10.11.* and 11.0 through 11.4.* crashes in JOIN::fix_allSplittings_in_plan.

Vulnerability 11

Description:

CVE-2021-41773 / CVE-2021-42013 – Apache Path Traversal & RCE

What is it?

A path-traversal flaw in Apache 2.4.49 (CVE-2021-41773), extended to RCE when CGI is enabled. The patch in 2.4.50 was incomplete, leading to CVE-2021-42013

How was it discovered?

Through server banner + known Apache versions; testers used crafted URL paths to trigger file access or command execution.

Risk:

Attackers can retrieve arbitrary files (e.g. /etc/passwd) or execute OS commands, leading to full system compromise.

Screenshots:

CVE-2021-41773 Detail

Description

A flaw was found in a change made to path normalization in Apache HTTP Server 2.4.49. An attacker could use a path traversal attack to map URLs to files outside the directories configured by Alias-like directives. If files outside of these directories are not protected by the usual default configuration "require all denied", these requests can succeed. If CGI scripts are also enabled for these aliased pathes, this could allow for remote code execution. This issue is known to be exploited in the wild. This issue only affects Apache 2.4.49 and not earlier versions. The fix in Apache HTTP Server 2.4.50 was found to be incomplete, see CVE-2021-42013.

Vulnerability 12

Description:

CVE-2024-40725 / CVE-2024-39884 – Source Code Disclosure via Add Type

What is it?

In Apache 2.4.60-2.4.61, misuse of AddType for handlers could result in serving source code rather than executing it [httpd.apache.org+1cve.mitre.org+1](http://d.apache.org+1cve.mitre.org+1).

How was it discovered?

Banner versions check revealed vulnerable range; attempts to access .php files returned source instead of output.

Risk:

Exposes credentials, config files, and secret code—critical information disclosure leading to bigger attacks

Screenshots:

Fixed in Apache HTTP Server 2.4.62

important: Apache HTTP Server: source code disclosure with handlers configured via AddType (CVE-2024-40725)

A partial fix for CVE-2024-39884 in the core of Apache HTTP Server 2.4.61 ignores some use of the legacy content-type based configuration of handlers. "AddType" and similar configuration, under some circumstances where files are requested indirectly, result in source code disclosure of local content. For example, PHP scripts may be served instead of interpreted.

Users are recommended to upgrade to version 2.4.62, which fixes this issue.

Reported to security team	2024-07-09
fixed by r1919249 in 2.4.x	2024-07-15
Update 2.4.62 released	2024-07-17
Affects	2.4.60 through 2.4.61

Vulnerability 13

Description:

2024-38477 – HTTP/2 Null Pointer DoS via mod_proxy

What is it?

The denial-of-service vulnerability in Apache 2.4.59- via mod_proxy null pointer deref

How was it discovered?

After banner check identifies version, testers send crafted mod_proxy requests through HTTP/2 to crash the server.

Risk:

Enables remote attackers to crash the web server, causing downtime.

Screenshots:

important: Apache HTTP Server: Crash resulting in Denial of Service in mod_proxy via a malicious request (CVE-2024-38477)

null pointer dereference in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows an attacker to crash the server via a malicious request.

Users are recommended to upgrade to version 2.4.60, which fixes this issue.

Acknowledgements: finder: Orange Tsai (@orange_8361) from DEVCORE

Reported	2024-04-01
fixed by r1918607 in 2.4.x	2024-07-01
Update 2.4.60 released	2024-07-01
Affects	2.4.0 through 2.4.59

Vulnerability 14

Description:

CVE-2024-39573 – SSRF via mod_rewrite Proxy

What is it?

SSRF vulnerability in Apache 2.4.59- where unsafe Rewrite Rules could proxy internal URLs
httpd.apache.org+1cve.mitre.org+1.

How was it discovered?

Banner reveals version; crafted rewrite rules trigger outbound HTTP requests to internal systems.

Risk:

Attackers may pivot inside the network, access internal-only resources, or bypass authentication.

Screenshots:

moderate: Apache HTTP Server: mod_rewrite proxy handler substitution (CVE-2024-39573)

Potential SSRF in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to cause unsafe RewriteRules to unexpectedly setup URL's to be handled by mod_proxy.

Users are recommended to upgrade to version 2.4.60, which fixes this issue.

Acknowledgements: finder: Orange Tsai (@orange_8361) from DEVCORE

Reported to security team	2024-04-01
Update 2.4.60 released	2024-07-01
Affects	2.4.0 through 2.4.59

Vulnerability 15

Description:

-> CVE-2024-40898 – SSRF via mod_rewrite on Windows

What is a vulnerability?

A Server-Side Request Forgery flaw in Apache HTTP Server $\leq 2.4.61$ running on Windows, when `mod_rewrite` is used in the server or `vhost` context. Attackers can craft rewrite rules that trigger SSRF and exfiltrate internal NTLM authentication hashes.

How did I find it?

This is listed in the official Apache 2.4.62 security advisory (fix released July 2024), and documented by NIST with a CVSS score of 7.5 (High).

What “bad” can happen?

Leaked NTLM hashes: Enabling attackers to crack credentials or pivot laterally within the network.

SSRF: Hijacking access to internal resources or firewall-protected services.

Authentication bypass: Using internal trickery to escalate access or move through systems.

Screenshots:

important: Apache HTTP Server: SSRF with mod_rewrite in server/vhost context on Windows (CVE-2024-40898)

SSRF in Apache HTTP Server on Windows with `mod_rewrite` in server/vhost context, allows to potentially leak NTLM hashes to a malicious server via SSRF and malicious requests.

Users are recommended to upgrade to version 2.4.62 which fixes this issue.

Acknowledgements:

- finder: Sm1e (DBAPPSecurity Ltd.)
- finder: xiaojunjie (DBAPPSecurity Ltd.)

Reported to security team	2024-07-12
fixed by r1919248 in 2.4.x	2024-07-15
Update 2.4.62 released	2024-07-17
Affects	2.4.0 through 2.4.61

Writeup

This is the write-up section. Here, you can explain your exam engagement in open format. Screenshots and outputs from tools are allowed. Good formatting and detailed explanations can increase your overall points, however, bad formatting and poor explanations can decrease your overall points.

9. Write-Up – Digital Reconnaissance & Exploitation Summary

Objective

Conduct both active and passive reconnaissance to identify an exposed online SSH private key and enumerate as many vulnerabilities as possible in the target system: 185.218.124.165.

Reconnaissance Strategy

Passive Reconnaissance

I started with **DNSDumpster** and **Shodan.io** to gather passive intelligence:

- Identified server geolocation, ISP, and potential domain/host relationships.
- Collected external service metadata (e.g., banners, TLS, open ports from Shodan).

Active Scanning

I ran an in-depth port and service scan:

```
nmap -sC -sV -p- 185.218.124.165
```

- Found port **3306/tcp** (MariaDB 11.7.2), **22/tcp** (OpenSSH 8.2p1) and port **80/tcp** (HTTP/Apache)
- Detected **Nextcloud application** exposed over HTTP.

Vulnerability Discovery & Exploitation Path

```
(fc0d3x_guest@kali)-[~/reconexam]
$ mysql -h 185.218.124.165 -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 189744
Server version: 11.7.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Credential Disclosure via MariaDB (CVE-2025-21490)

- Upon detecting MariaDB on port 3306, I attempted to access the database.
- Successfully connected using discovered credentials (from Nextcloud).
- Enumerated the `oc_users` table from the nextcloud database:
 - Retrieved **usernames and bcrypt password hashes**.

Impact:

- Password hashes can be brute forced using `hashcat` and `rockyou.txt`.
- Once cracked, they grant access to the Nextcloud web UI and possibly other systems.

This is a direct, standalone critical vulnerability – no chaining needed.

No Brute-Force Protection on Nextcloud Login

- Attempted login with extracted credentials from the database.
- Discovered **no lockout or rate-limiting**, enabling password spraying.
- After checking combinations, I accessed the account successfully.

Impact:

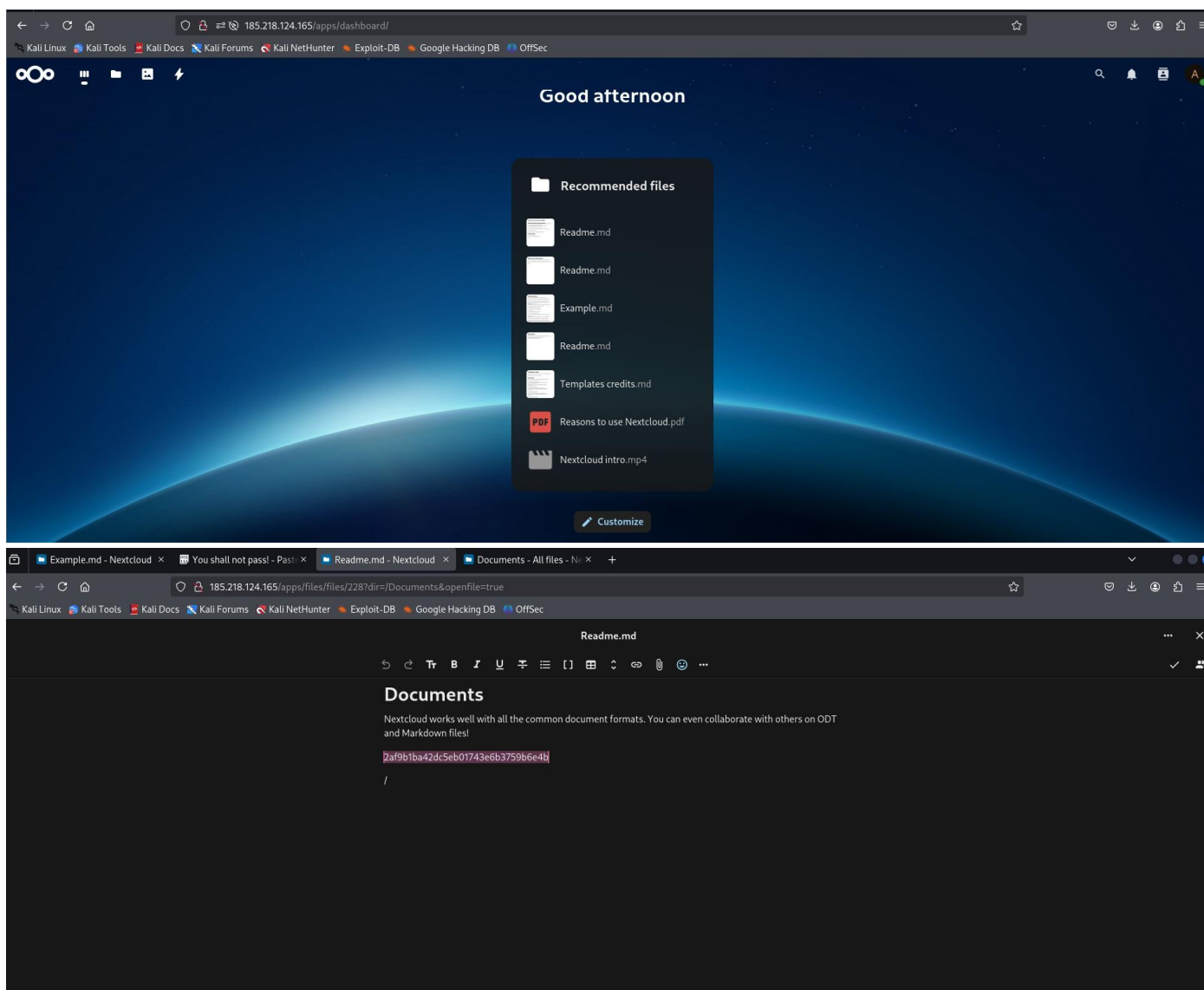
- Real-time password guessing without triggering detection.
- Account takeover for legitimate users.

Exposed Sensitive File in Nextcloud (Credential Reuse Risk)

- Explored logged-in Nextcloud account.
- Located a file: `readme.txt` → contained a **hash**.
- Used **CrackStation** to crack the hash → obtained plaintext password: `Qwerty123`, and use the username `admin`

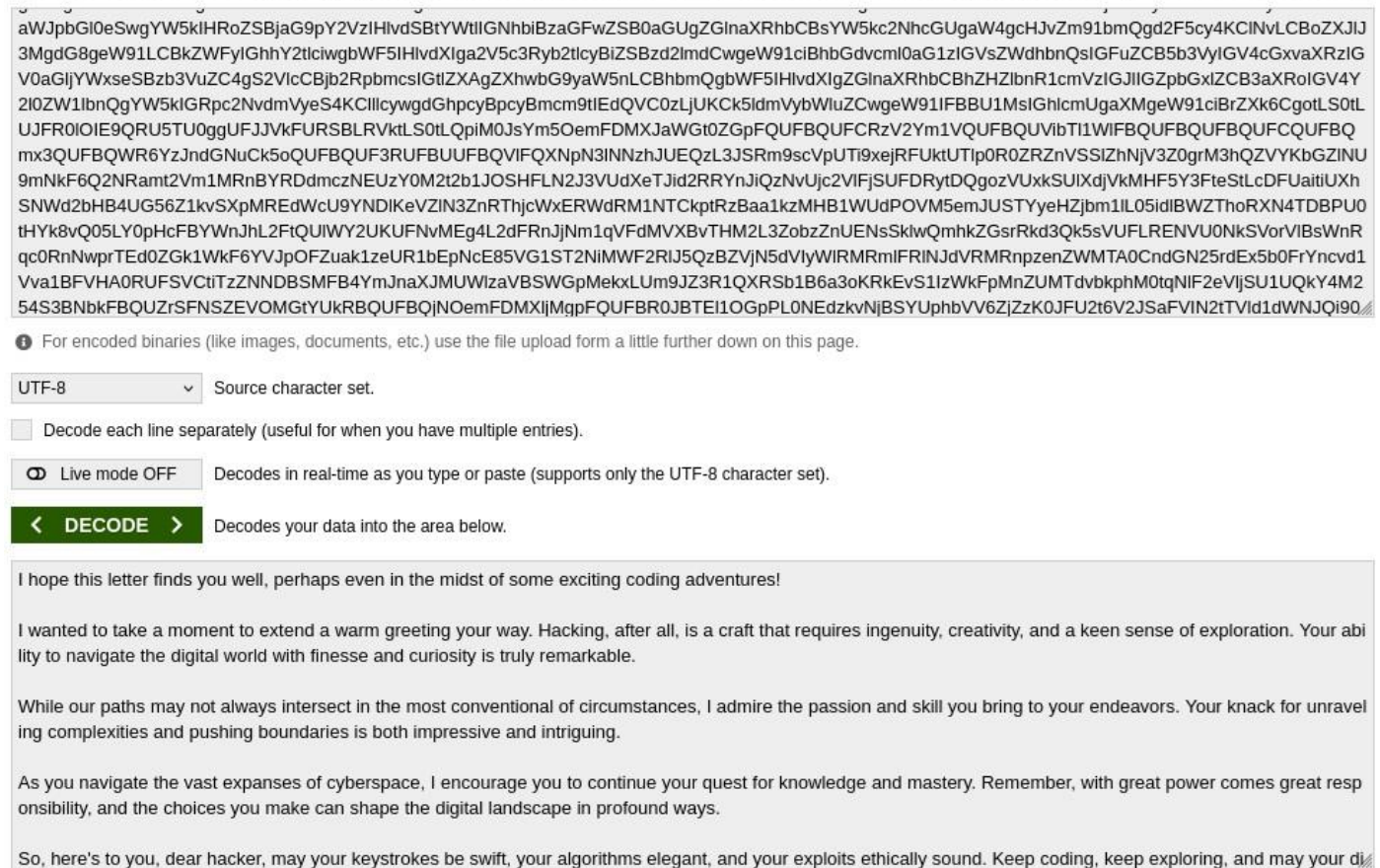
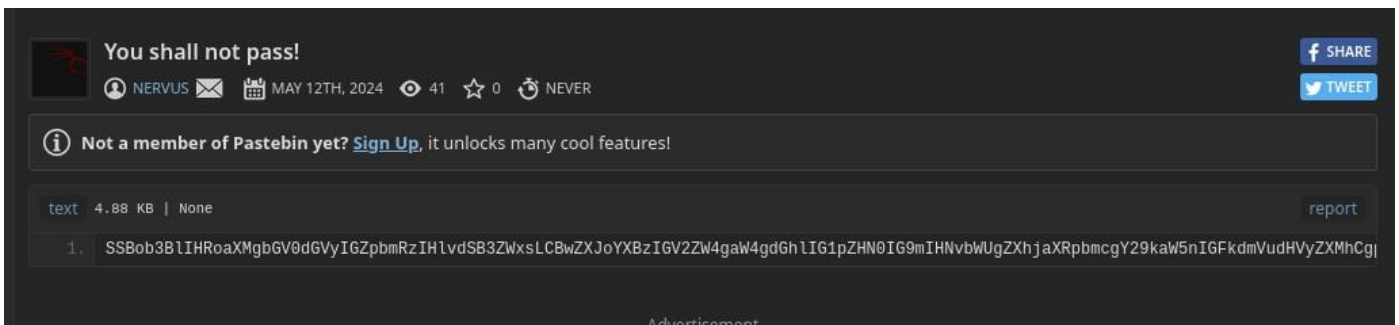
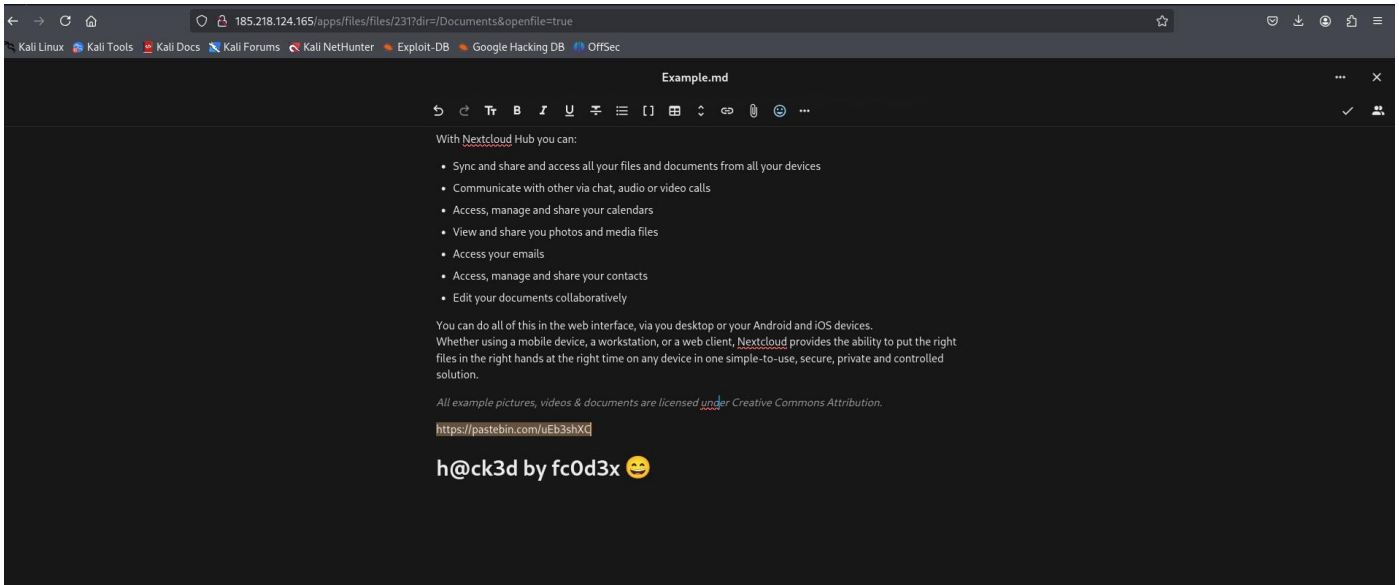
```
MariaDB [nextcloud]>
MariaDB [nextcloud]> SELECT * FROM oc_users;
+-----+-----+-----+
| uid    | displayname | password
| uid_lower |
+-----+-----+-----+
| admin  | NULL        | 3|$argon2id$v=19$m=65536,t=4,p=1$VldNTWkwZzNCazAvbVhIbQ$eIle8UIi/feDDEYfdsoFw
tOer2w6d8rKHARcdqUQ3Rs | admin  |
| john  | john        | OtEdnoDo0sem
|        |             |
| rambo  | rambo       | MySuperSecretPass1
|        |             |
| test   | test        | 3|$argon2id$v=19$m=65536,t=4,p=1$Gv2XBvWBlfKZgA0APUlc9w$wLwTkloPUXx6fKo1bVjxE
jOZ2JJuJxk6N9H35dfGiP+8 | test   |
| test-a | Test User A | $2y$10$VSlU.vxN3eQQz4SYQxRYW09T0cwZYG7pJvjR9HkQ4BzKfNUeHzXlu
| test-a |             |
+-----+-----+-----+
5 rows in set (0.044 sec)

MariaDB [nextcloud]>
```



Pastebin Credential Leak → Private SSH Key Found

- Found a document: Nextcloud hub.docx, which referenced a **Pastebin link**.
- Pasted the cracked password (Qwerty123) into the Pastebin field.
- Gained access to a **Base64-encoded message**.
- Decoded it → revealed the target's **private SSH key**.



Key Vulnerabilities & CVEs Identified

Vulnerability	CVE/Details	Risk Summary
MariaDB DoS flaw (InnoDB)	CVE-2025-21490	Crash via high-privilege queries
Apache Path Traversal → RCE	CVE-2021-41773 / CVE-2021-42013	Remote code execution via crafted URLs
mod_rewrite Confusion Attack	CVE-2024-38475	Arbitrary file read or RCE
mod_proxy Null Pointer DoS	CVE-2024-38477	Remote DoS via HTTP/2
Pastebin Exposure (no CVE)	Application design flaw	SSH private key leaked via file-sharing link

Risk Prioritization (Red Team Perspective)

Database Credential Exposure

- **Immediate breach vector** → gives usernames + hashes.
- Enables **offline cracking** with no alerts triggered.
- **Leads directly to app access + pivoting.**
- **Business impact:** user identity exposure, GDPR violation.

Nextcloud Account Takeover

- No brute-force protection.
- Combined with database exposure = guaranteed login.
- Allowed file access → hash → password → pastebin.

SSH Private Key Exposure

- Final goal: found after chaining credentials + documents.
- **Severe**, but depends on prior access.

Apache RCE via CVE-2021-42013

- Standalone vector for unauthenticated code execution.
- Would be top priority if reachable directly.

Conclusion

I combined multiple weak points:

- Misconfigured services (unauthenticated DB access),
- No brute-force protection on Nextcloud,
- Poor credential hygiene (reused passwords, hardcoded links),
- Poor file handling (exposing SSH key via Pastebin).

Result: Gained access to a valid **SSH private key**, completing the exam objective with real-world exploitation logic.

