EJERCICIOS ANÁLISIS Y GESTIÓN DE VULNERABILIDADES CON OPENVAS Y NESSUS

Prerrequisitos

Kali Linux
Metasploitable2

Ejercicio 1 - OpenVAS

Realizar un análisis de vulnerabilidades sobre el equipo Metasploitable2 utilizando OpenVAS.

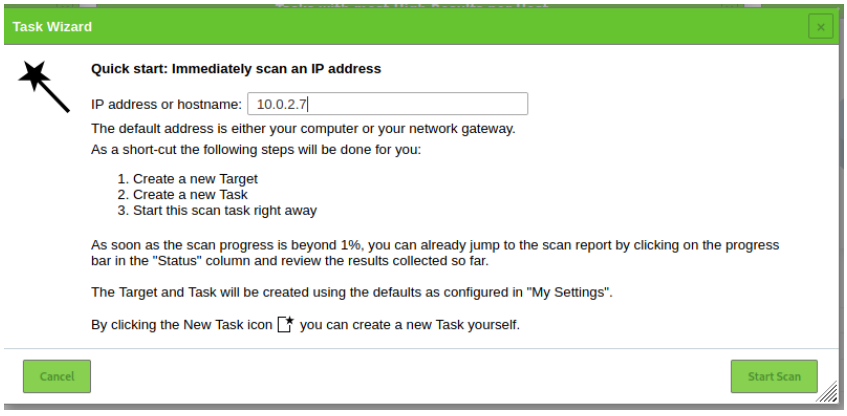
```
┌──(root💀kali)-[/home/kali]
└─# gvm-start
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>]  Web UI (Greenbone Security Assistant): https://127.0.0.1:9392

● greenbone-security-assistant.service - Greenbone Security Assistant (gsad)
     Loaded: loaded (/lib/systemd/system/greenbone-security-assistant.service; disabled; vendor preset: disabled)
     Active: active (running) since Tue 2023-10-10 15:35:40 CEST; 112ms ago
       Docs: man:gsad(8)
             https://www.greenbone.net
    Process: 1217 ExecStart=/usr/sbin/gsad --listen=127.0.0.1 --port=9392 (code=exited, status=0/SUCCESS)
   Main PID: 1219 (gsad)
      Tasks: 2 (limit: 2300)
```

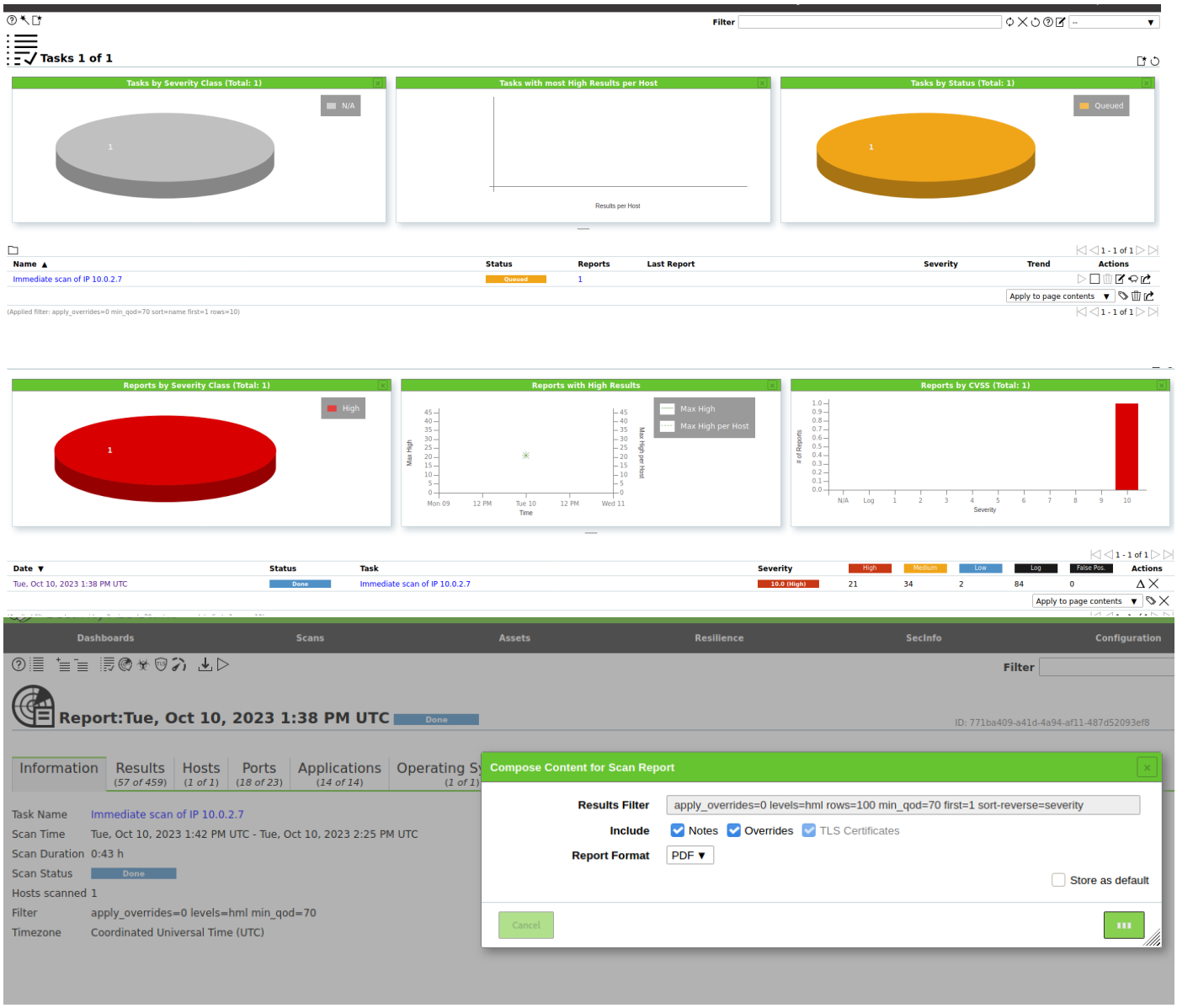
```
    CGroup: /system.slice/ospd-openvas.service
            ├─1131 /usr/bin/python3 /usr/bin/ospd-openvas --config /etc/gvm/ospd-openvas.conf --log-config /etc/gvm/ospd
logging.conf --unix-socket /run/ospd/ospd.sock --pid-file /run/ospd/ospd-openvas.pid --log-file /var/log/gvm/ospd-openva
.log --lock-file-dir /var/lib/openvas
            ├─1133 /usr/bin/python3 /usr/bin/ospd-openvas --config /etc/gvm/ospd-openvas.conf --log-config /etc/gvm/ospd
logging.conf --unix-socket /run/ospd/ospd.sock --pid-file /run/ospd/ospd-openvas.pid --log-file /var/log/gvm/ospd-openva
.log --lock-file-dir /var/lib/openvas
            ├─1210 openvas --update-vt-info
            └─1211 openvas --update-vt-info

ct 10 15:35:26 kali systemd[1]: Starting OSPd Wrapper for the OpenVAS Scanner (ospd-openvas)...
ct 10 15:35:27 kali systemd[1]: Started OSPd Wrapper for the OpenVAS Scanner (ospd-openvas).
>] Opening Web UI (https://127.0.0.1:9392) in: 5... 4... 3... 2... 1...
```

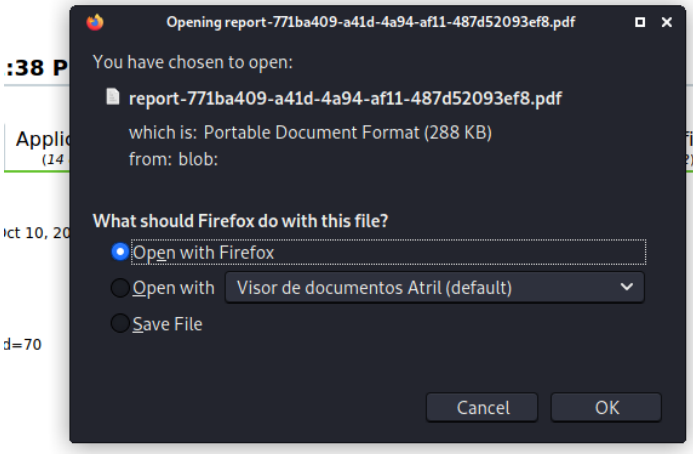
Le damos click a *task wizard* que viene a ser la varita mágica



Exponemos la IP que queremos estudiar



Una vez acabado el proceso le doy justo donde estaba la barra de proceso, donde ahora pone *done*



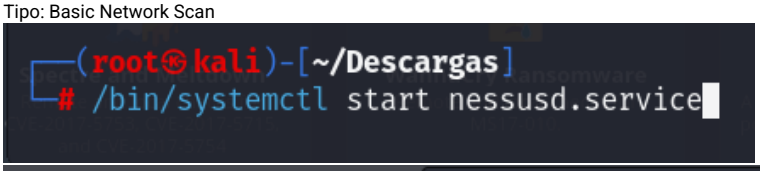
Descargo el archivo y lo envío al sistema operativo nativo

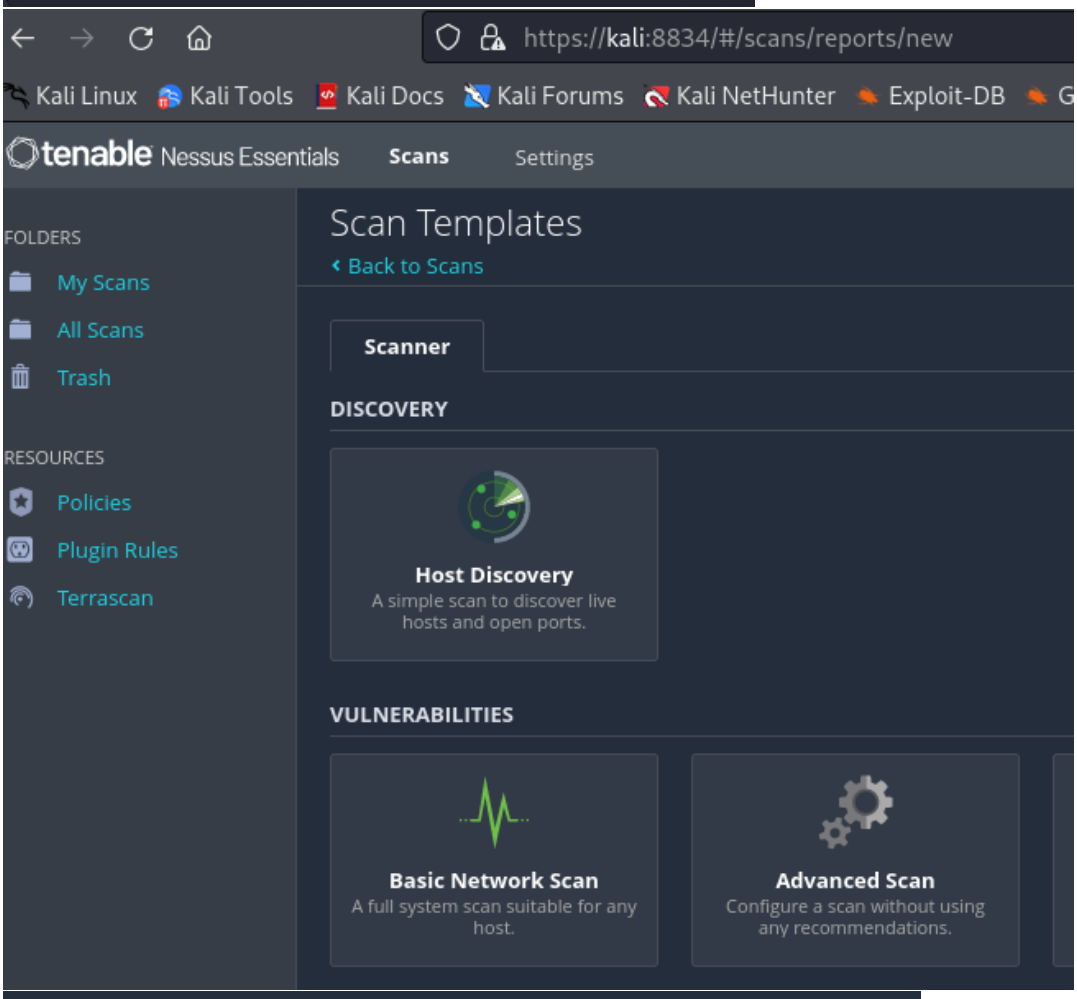
Exportar informe PDF.
Subir el informe a classroom.

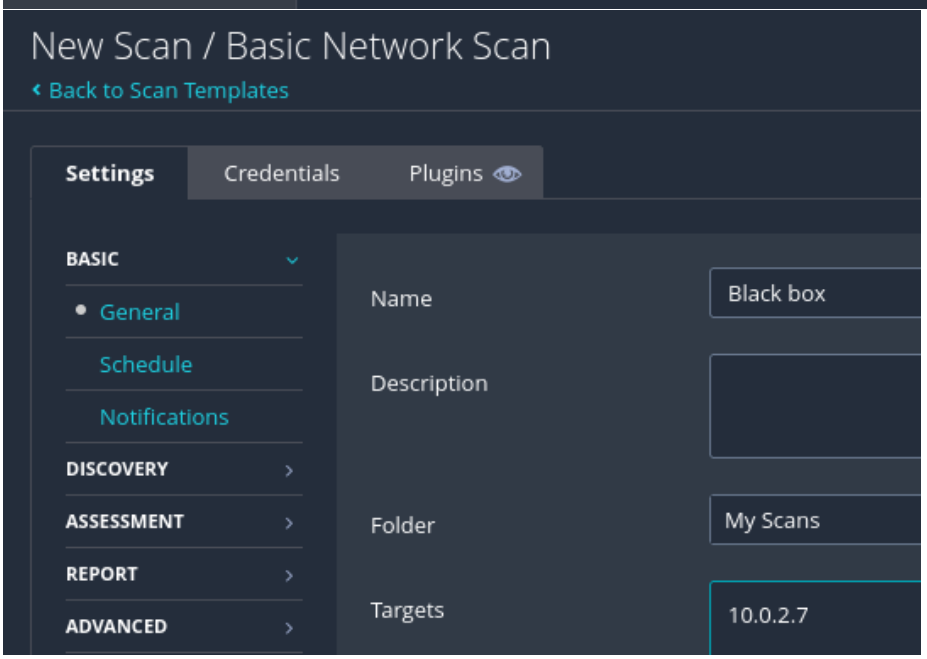
Ejercicio 2 - Nessus

Realizar un análisis de vulnerabilidades sobre el equipo Metasploitable2 en modo Black Box:

Tipo: Basic Network Scan

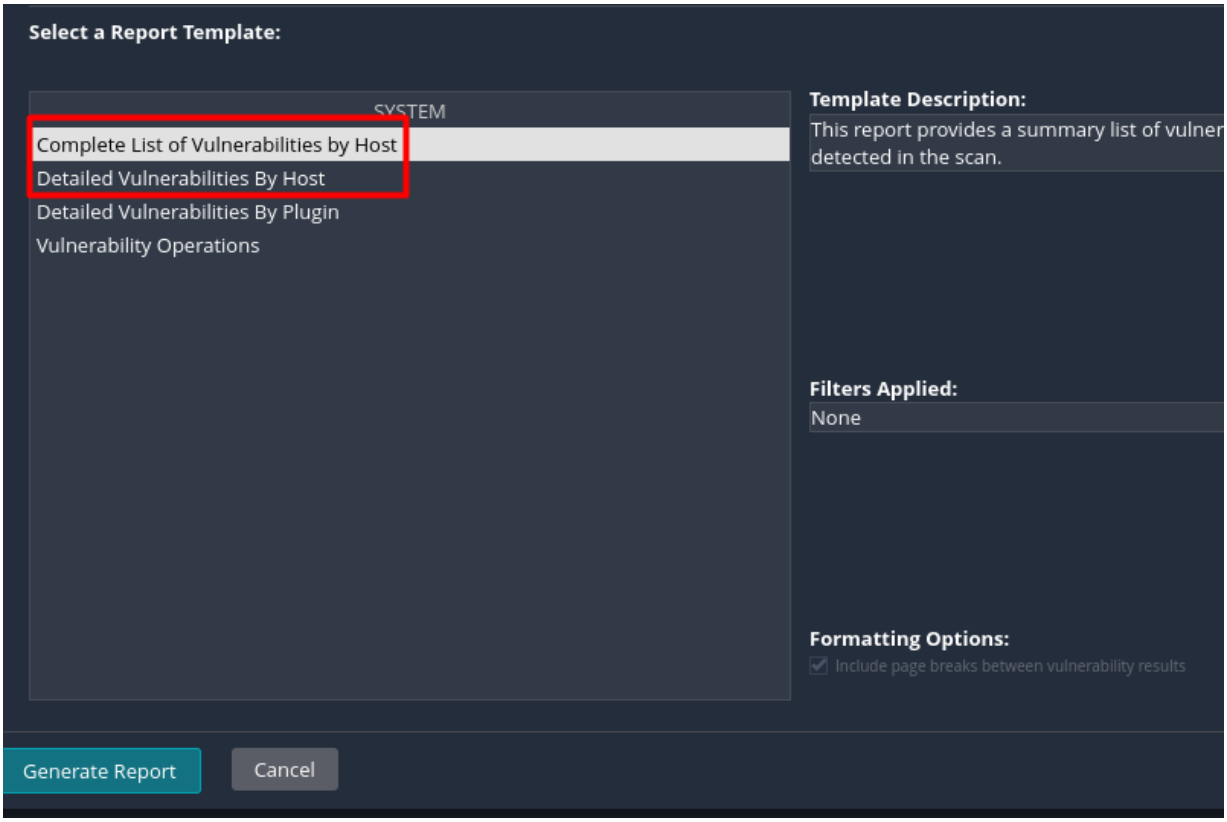






En *credentials* no tengo que modificar nada

Una vez acabado el escaneo descargamos los PDF's correspondientes en el apartado *report*

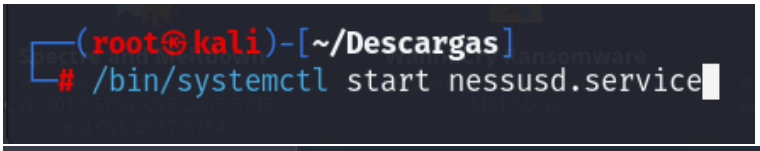


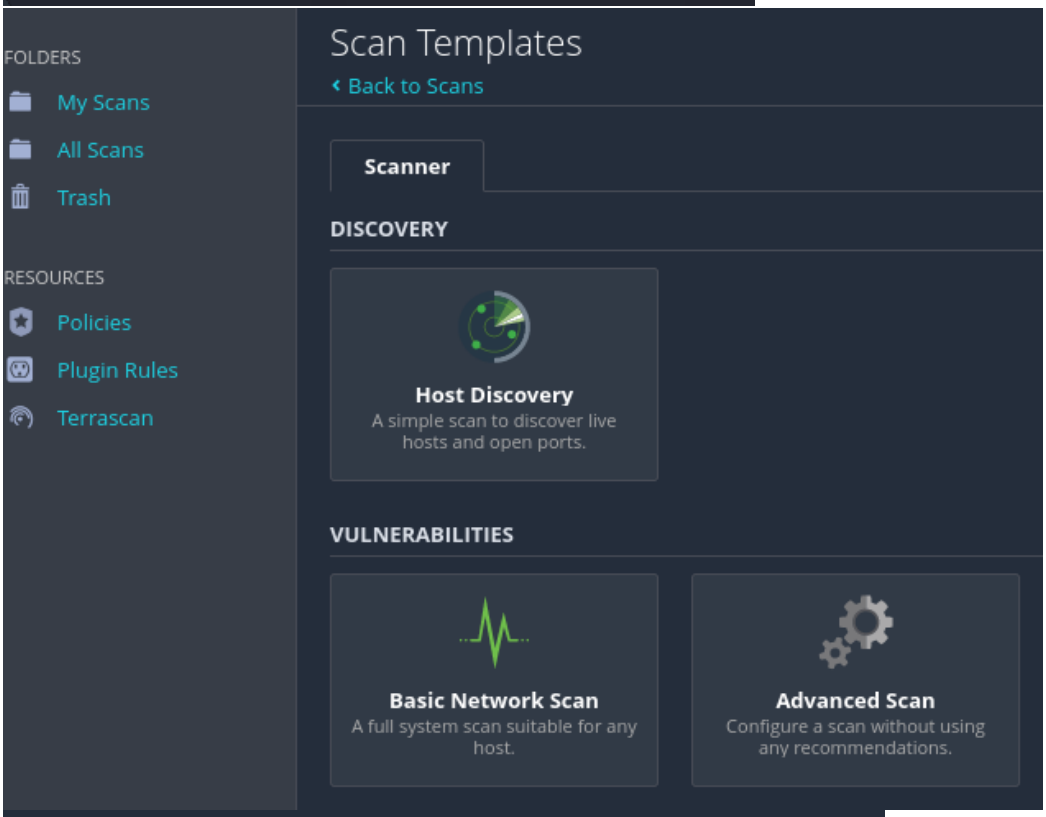
Seleccionamos los dos tipos de archivos y se hará una descarga automática.

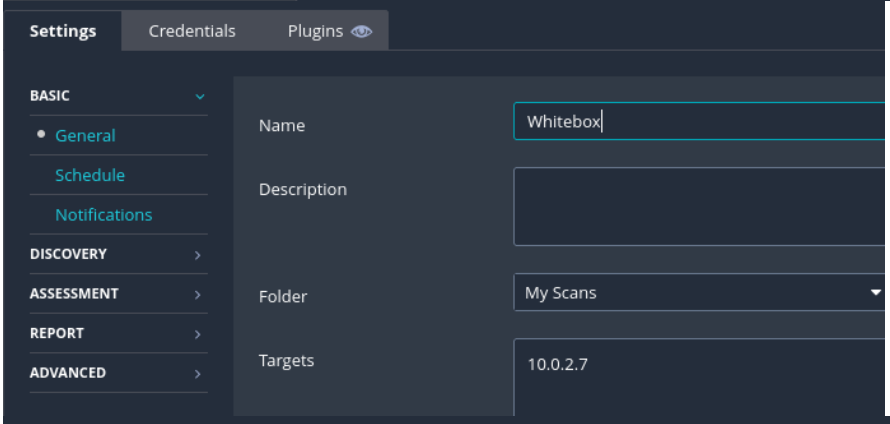
Sin credenciales
Exportar informe PDF:
Ejecutivo: Complete List of Vulnerabilities by Host
Técnico: Detailed Vulnerabilities by Host
Subir el informe a classroom.

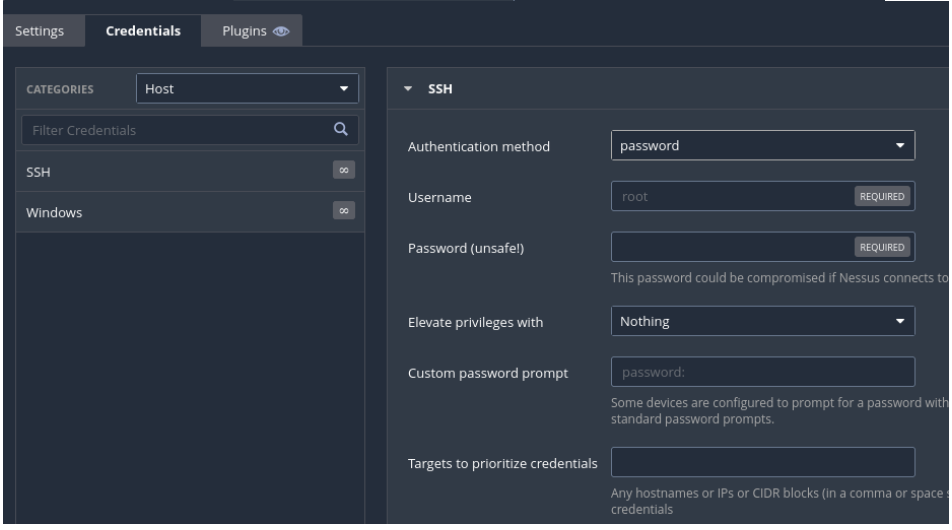
Ejercicio 3 - Nessus

Realizar un análisis de vulnerabilidades sobre el equipo Metasploitable2 en modo White Box:
Tipo: Basic Network Scan
Con credenciales > username: msfadmin - password: msfadmin

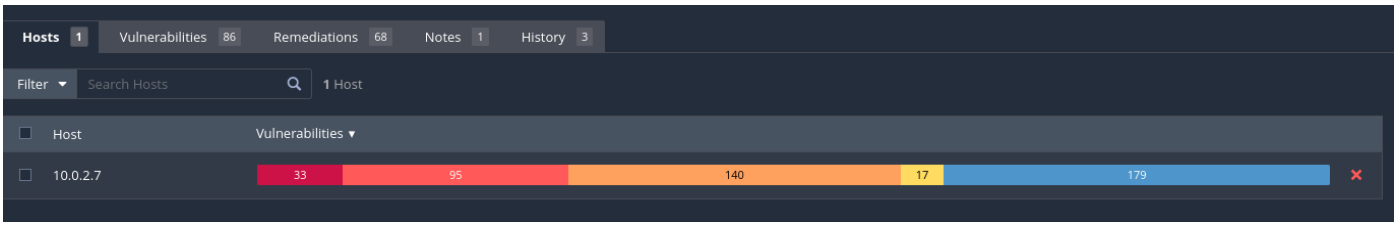




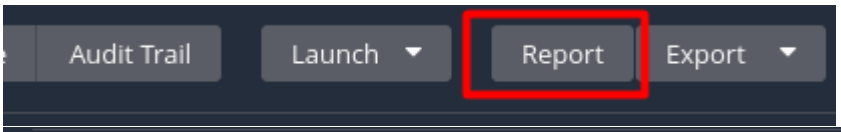


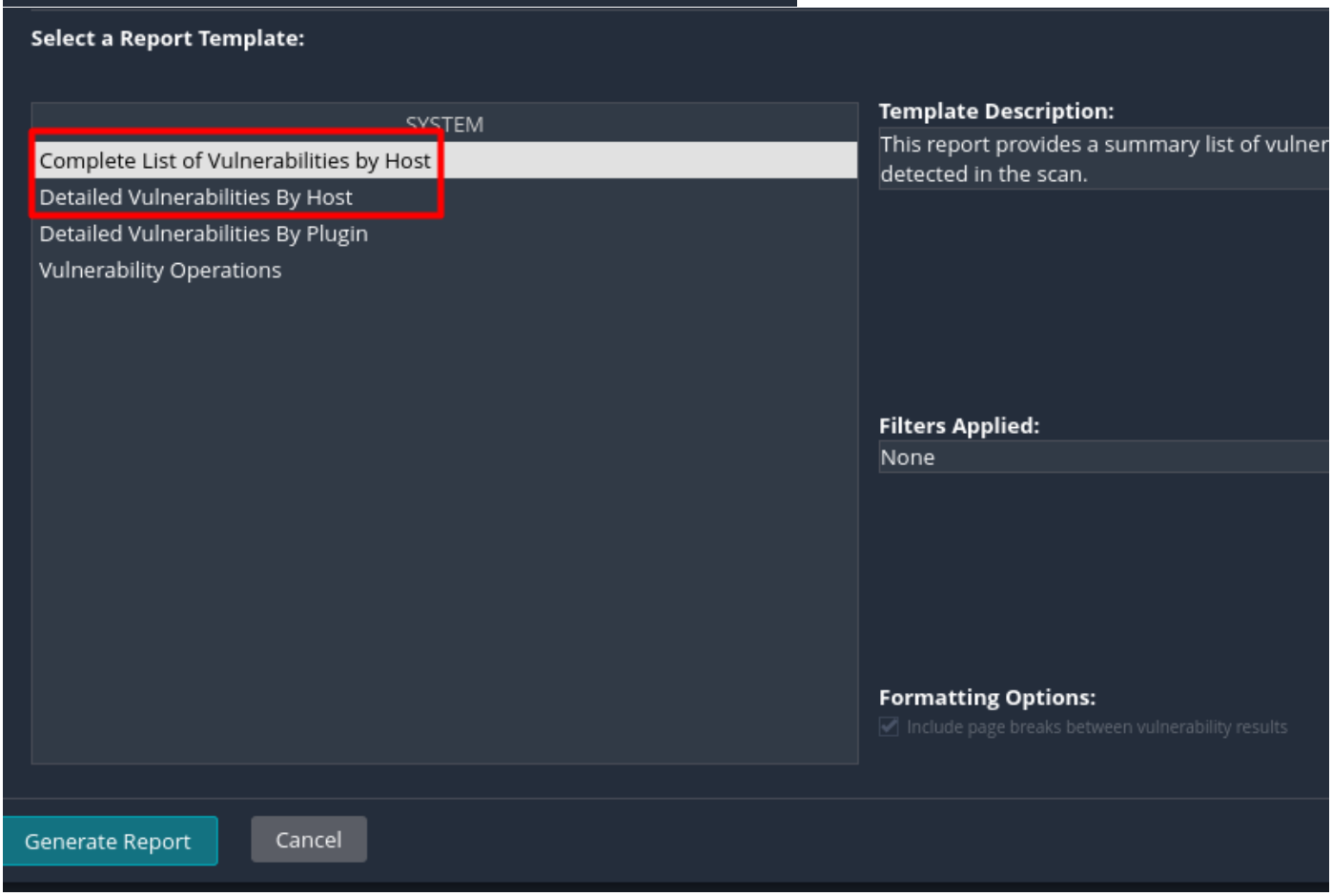


En el apartado de *Credentials* en username ponemos las credenciales antes descritas, tras esto le damos a Save y empieza el escaneo



Una vez se haya completado el escaneo entramos en él.





Descargamos estos dos archivos como pdf y lo tendríamos

Exportar informe PDF:
- Ejecutivo: Complete List of Vulnerabilities by Host
- Técnico: Detailed Vulnerabilities by Host

Subir el informe a classroom.

Ejercicio 4 - Caso Real

Imagina que un cliente te solicita por correo información detallada de alguna de las vulnerabilidades de tu informe realizado OpenVAS. En base a esto, desarrolla una explicación de la vulnerabilidad que elijas.

CVE-2008-5305

- **Fecha de publicación:** 12/09/2008
- **Código CWE:** CWE-94
- **CVSS 3.x:** N/A
  - Vector: N/A
- **CVSS 2.x:** 10.0 High
  - Vector: (AV:N/AC:L/Au:N/C:C/I:C/A:C)
- **URL de CVE:** https://nvd.nist.gov/vuln/detail/CVE-2008-5305
- **Productos y versiones vulnerables**
  - cpe:2.3:a:twiki:twiki:4.1.1:*:*:*:*:*:*:*
  - cpe:2.3:a:twiki:twiki:4.0.1:*:*:*:*:*:*:*
  - cpe:2.3:a:twiki:twiki:4.0.3:*:*:*:*:*:*:*
  - cpe:2.3:a:twiki:twiki:4.0.4:*:*:*:*:*:*:*
  - cpe:2.3:a:twiki:twiki:4.2.1:*:*:*:*:*:*:*
- **Descripcion de la vulnerabilidad:** Vulnerabilidad de inyección "eval" en TWiki y versiones anteriores a 4.2.4 que permite a los atacantes remotos ejecutar arbitrariamente código Perl a través de la variable %SEARCH{}%.
- **Recomendaciones:** esta vulnerabilidad tiene más de una década y se espera que la mayoría de los sistemas hayan sido actualizados para abordarla, las mejores prácticas de seguridad siempre incluyen asegurarse de que el software esté actualizado y protegido contra vulnerabilidades conocidas.
  - **Actualizar el sistema**
  - **Configurar el firewall:** configura las reglas del firewall para permitir solo el trafico necesario y bloquear lo demás
  - **Monitoreo de red y trafico**
  - **Limitar privilegios**
  - **Aplicar parches de seguridad**
  - **Auditoria y registro de eventos**
  - **Implementar soluciones de seguridad:** Utilizar herramientas y soluciones como sistemas de prevención de intrusiones basados en host (HIPS), soluciones de análisis de vulnerabilidades y sistemas de gestión de eventos e información de seguridad (SIEM), para una mayor protección contra amenazas.

Ejercicio 5 - Caso Real

Imagina que un cliente te solicita por correo información detallada de alguna de las vulnerabilidades de tus informes realizados con Nessus. En base a esto, desarrolla una explicación de la vulnerabilidad que elijas.

CVE-2019-17571

- **Fecha de publicación:** 12/20/2019
- **Código CWE:** CWE-502
- **CVSS 3.x**: 9.8 Critical
  - Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
- **CVSS 2.x:** 7.5 High
  - Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P)
- **URL de CVE**: https://nvd.nist.gov/vuln/detail/CVE-2019-17571
- **Productos y versiones vulnerables**
  - cpe:2.3:a:apache:log4j:*:*:*:*:*:*:*:*
  - cpe:2.3:o:debian:debian_linux:8.0:*:*:*:*:*:*:*
  - cpe:2.3:o:debian:debian_linux:9.0:*:*:*:*:*:*:*
  - cpe:2.3:o:debian:debian_linux:10.0:*:*:*:*:*:*:*
  - cpe:2.3:o:canonical:ubuntu_linux:18.04:*:*:*:lts:*:*:*
- **Descripcion de la vulnerabilidad:**
  En Log4j 1.2 se incluye una clase SocketServer que es vulnerable a la deserialización de datos no confiables que puede explotarse para ejecutar código arbitrario de forma remota cuando se combina con un dispositivo de deserialización al escuchar el tráfico de red no confiable en busca de datos de registro. Esto afecta a las versiones de Log4j desde la 1.2 hasta la 1.2.17.
- **Recomendaciones:** algunas recomendaciones que se pueden seguir para mitigar los riesgos asociados son las siguientes:
  - **Actualización del software**
  - **Seguir las recomendaciones del vendedor:** encontradas en el siguiente enlace https://www.oracle.com/security-alerts/cpuapr2021.html
  - **Configurar el firewall**
  - **Principio de Privilegios Mínimos**
  - **Auditoria y monitoreo**
  - **Educación del usuario**
  - **Realizar auditorías de seguridad**
  - **Aplicar listas de control de acceso**
  - ***Backup* regular:** En caso de un ataque exitoso, tener copias de seguridad puede ser crucial para recuperar los datos
  - **Colaboración con Comunidades de Seguridad:** mantenerse al tanto de las discusiones en la comunidad de seguridad informática

.