Análisis de las vulnerabilidades en un servidor de pruebas

Contenido

Para esto voy a usar **OpenVAS**:

Se utiliza para identificar y evaluar vulnerabilidades en sistemas informáticos y redes. Ayuda a descubrir posibles debilidades en la seguridad que podrían ser explotadas por atacantes.

Funciones clave:

Escaneo de red: Identifica dispositivos activos en una red.

Detección de vulnerabilidades: Encuentra y evalúa debilidades en sistemas y aplicaciones.

Informes detallados: Genera informes con información sobre las vulnerabilidades encontradas.

Componentes:

OpenVAS incluye varios componentes, como el motor de escaneo (OpenVAS Scanner) y las bases de datos de vulnerabilidades (NVTs - Network Vulnerability Tests).

Greenbone Security Assistant (GSA):

Interfaz web: GSA es la interfaz web de OpenVAS que permite a los usuarios interactuar con el sistema de forma gráfica. Proporciona una interfaz fácil de usar para configurar escaneos, ver informes y gestionar resultados.

Características:

Configuración de escaneos: Permite personalizar los escaneos según las necesidades específicas del usuario.

Visualización de resultados: Muestra de manera clara y detallada las vulnerabilidades encontradas.

Gestión de informes: Facilita la generación y revisión de informes de evaluación de vulnerabilidades.

Acceso remoto:

Al ser una interfaz web, GSA facilita el acceso remoto a OpenVAS, lo que es útil para la administración y supervisión desde ubicaciones diversas. En resumen, OpenVAS es una herramienta de seguridad que se centra en la evaluación de vulnerabilidades, y Greenbone Security Assistant proporciona una interfaz web intuitiva para interactuar con OpenVAS, facilitando la configuración de escaneos, la revisión de resultados y la generación de informes de seguridad. Ambos trabajan juntos para ayudar a proteger sistemas y redes al identificar posibles puntos débiles.

Instalación y configuración de OpenVAS

Decidimos hacerlo con dockers por ser más sencillo:

 # docker pull mikesplain/openvas; Para descargarse el docker.

 # docker images; Para ver si se había descargado bien.

 # docker run -d -p 443:443 -p 9390:9390 --name openvas mikesplain/openvas; Para runnearlo en el puerto 443 y 9390 en la local host 127.0.0.1.

#docker ps -a; Para ver mis docker que estan corriendo.

#docker top openvas; Para ver los procesos abiertos y si esta corriendo adecuadamente. **#docker exec -it openvas bash**; Para abrir el bash de openvas, a partir de aqui en el bash ejecutamos estos comandos para actualizar la base de datos del contenedor.

>greenbone-nvt-sync

>openvasmd --rebuild --progress

>greenbone-certdata-sync

>greenbone-scapdata-sync

>openvasmd --update --verbose --progress

>/etc/init.d/openvas-manager restart

>/etc/init.d/openvas-scanner restart

Pongo en Firefox https://localhost:443

Verificamos la instalación, todo está adjuntado en fotos.

Análisis de amenazas

Lo primero fue instalar el servidor de pruebas, instale por Docker mutillidae, luego saque la ip, la cual era 172.17.0.3.

Una vez todo instalado le di a configuración y cree un nuevo target llamado Mutilldae con la ip antes mencionada, 172.17.0.3, y le hice los 6 tipos de análisis posibles, full and fast, full and fast ultimate, full and very deep, full and very deep ultimate, host discovery y system discovery, adjunto captura.

Como se puede observar con los exámenes host discovery y system discovery no se ha sacado nada, por lo cual deducimos que no se ajustan a nuestras necesidades, así que nos centraremos en los otros 4 antes mencionados.

Los otros 4 han sacado exactamente las mismas 12 vulnerabilidades, aunque los que tienen la terminación final han examinado 126 en lugar de 125 posibles vulnerabilidades, así que por eficacia y agilidad voy a hablar todo el rato del scan full and very deep ultimate, aunque lo mismo puede ser aplicado para las otras 4 antes mencionadas: Adjunto captura.

Voy a hablar desde el menos peligroso hasta el más peligroso siguiendo ese orden claro, y en cada una de las vulnerabilidades la identificare, desarrollare, explicare el impacto que pueden tener y posibles exploits.

En las capturas de cada vulnerabilidad se observa el código CVE, por lo que no me parare a comentarlo.

TCP Timestamps: Se detectó que el host implementa RFC1323 .

Las siguientes marcas de tiempo fueron recuperadas con un retraso de 1 segundo entre : Paquete 1 : 732837250 Paquete 2 : 732838466

Un efecto secundario de esta función es que a veces se puede calcular el tiempo de actividad del host remoto.

Puede haber sido un fallo en las marcas de tiempo, por lo que bastaría simplemente con desactivarlas, hay múltiples tutoriales en google que te enseñan a hacerlo.

Esto puede ser una vulnerabilidad por que se pueden llegar a falsificar paquetes IP, las marcas de tiempo son una medida contra eso, como en ataques man in the middle.

SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability: la clave temporal del servidor tiene un tamaño menor a 2048 bits, el cual es 1024, esto, dependiendo de algunos parámetros, puede hacer que con 1024 bits sea descifrable, aunque con mucho esfuerzo, por lo que no es un peligro muy alto, la solución sería simplemente usar una clave de 2048 la cual se puede considerar prácticamente indescifrable.

SSL/TLS: Certificate Signed Using A Weak Signature Algorithm: El servicio remoto está utilizando un certificado SSL/TLS en la cadena de certificados que ha sido firmado utilizando un algoritmo hash criptográficamente débil, esto se puede solucionar usando un SHA-256 por ejemplo ya que el que usamos en este caso es un simple e inseguro SHA-1, esto ya es más peligroso por lo que lo pongo por delante del anterior punto ya que no es excesivamente difícil romper un SHA-1, con cierta información o máquinas potentes puedes romperlos.

SSL/TLS: Report Weak Cipher Suites: Hay un conjunto de cifrado catalogados como débiles que estan siendo aceptados por este servicio a través del protocolo TLSv1.0:

TLS_ECDHE_RSA_WITH_RC4_128_SHA

TLS_RSA_WITH_RC4_128_SHA

TLS_RSA_CON_SEED_CBC_SHA

Suites de cifrado "débiles" aceptadas por este servicio a través del protocolo TLSv1.1:

TLS_ECDHE_RSA_WITH_RC4_128_SHA

TLS_RSA_WITH_RC4_128_SHA

TLS_RSA_CON_SEED_CBC_SHA

Suites de cifrado "débiles" aceptadas por este servicio a través del protocolo TLSv1.2:

TLS_ECDHE_RSA_WITH_RC4_128_SHA

TLS_RSA_WITH_RC4_128_SHA

TLS_RSA_CON_SEED_CBC_SHA

Esto significa un escalón más de peligro ya que no solo falla un protocolo o capa, aqui hay un conjunto de suites vulnerables, por lo que la solución pasaría por modificar la configuración de este servicio para que no acepte estos conjuntos catalogados como débiles antes enumerados.

FTP Unencrypted Cleartext Login: El host remoto está corriendo en un servicio FTP el cual no encripta las conexiones, entre otros problemas acepta logins sin hacerles un comando previo 'AUTH TLS', osea no son sesiones anónimas, el atacante podría descubrir el nombre y contraseña de los usuarios, esto claramente pasa a ser una vulnerabilidad contra el propio usuario y la integridad de su información, por ello lo considero más peligroso todavía.

Esto se soluciona activando FTPS o forzando una conexión vía 'AUTH TLS'.

El certificado SSL/TLS: ya ha expirado, En el puerto 443, el certificado remoto expiró hace 3 años, y es peligroso realizar la conexión con un certificado expirado, adjunto detalles del certificado en las imágenes, la solución es simple, reemplazar el SSL/TLS por uno nuevo, ya que un expirado TLS expirado puede generar todos los siguientes problemas relacionados con la seguridad y la conectividad en una aplicación o sitio web.

SSL/TLS: Report Vulnerable Cipher Suites for HTTPS:

Esta rutina informa de todas las suites de cifrado SSL/TLS aceptadas por un servicio en el que existen vectores de ataque sólo en servicios HTTPS, parecido a uno de los puntos anteriores, en este caso acepta:

Suites de cifrado "vulnerables" aceptadas por este servicio a través del protocolo TLSv1.0:

TLS_DHE_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

TLS_ECDHE_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

TLS_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

Suites de cifrado "vulnerables" aceptadas por este servicio a través del protocolo TLSv1.1:

TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (SWEET32)

TLS_ECDHE_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

TLS_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

Suites de cifrado "vulnerables" aceptadas por este servicio a través del protocolo TLSv1.2:

TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (SWEET32)

TLS_ECDHE_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

TLS_RSA_CON_3DES_EDE_CBC_SHA (SWEET32)

Habria que cambiar la configuración de este servicio para no aceptar estos cypher suit.

SSL/TLS: Untrusted Certificate Authorities: El servicio utiliza un certificado SSL/TLS de una autoridad de certificación conocida que no es de confianza. Un atacante podría usar esto para ataques MitM, acceder a datos sensibles y otros ataques, esto es especialmente peligroso ya que se sabe que es uno de los ataques más frecuentes y fáciles de hacer, con un proxy podríamos ejecutar un ataque.

La solución es cambiar el certificado SSL/TLS por uno de confianza.

HTTP Debugging Methods (TRACE/TRACK) Enabled 443/tcp: Las funciones de depuración están habilitadas en el servidor web remoto, por lo que un atacante podría usar esto para engañar a sus usuarios legítimos de la web para que den sus credenciales por ejemplo, parecido en lo que peligro se refiere debido a que con ambos ataques puedes obtener toda la información de los usuarios para usarla luego.

La solución es desactivar el TRACE and TRACK methods en la configuración de servidor. El servidor web remoto soporta los métodos TRACE y/o TRACK. TRACE y TRACK son métodos HTTP que se utilizan para depurar las conexiones del servidor web.

HTTP Debugging Methods (TRACE/TRACK) Enabled 80/tcp: Igual pero para el puerto 80.

Result: phpinfo() output Reporting http: Muchos tutoriales de instalación de PHP instruyen al usuario a crear un archivo llamado phpinfo.php o similar que contenga la sentencia phpinfo(). Dicho archivo suele dejarse en el directorio del servidor web y suele poseer información potencialmente

sensible, como la clave del administrador, dirección ip del host, versión del servidor o del SO.

Puntuo esto como lo más peligroso porque no sólo puedes acceder como administrador al servidor y robar todos los datos si no que puedes tirar cualquier parte del servidor, o incluso comprometer la seguridad del administrador dejando expuesta su IP.

La solución es borrar el archivo, restringir el archivo o no poner información comprometida ahí.

Result: phpinfo() output Reporting
https: Igual pero para https.

Plan de mitigación

Hay varias formas de mitigar un ataque de este tipo que haya sido exitoso, la primera y más obvia es con copias de seguridad, por ejemplo con copias de seguridad semanales, recomiendo:

https://www.pccomponentes.com/kingston-a400-ssd-480gb por su precio, facilidad de poner al ser sata y rapidez al ser SSD, también se puede usar uno externo como:

https://www.pccomponentes.com/adata-se900g-disco-duro-ssd-externo-1tb-usb-c que es más caro pero más cómodo y con más almacenamiento, también podemos optar por hacer un RAID con diferentes discos duros, como un RAID 1 o disco espejo que nos puede servir para tener un backup en caso de pérdida de datos por un ciberataque.

Para restaurar sistema siempre recomendamos tener un disco de instalación de windows 10 y otro con ubuntu con un boot manager, para en caso de querer o hacer un formateo rápido o una recuperación del sistema sea facil, además es un método barato simple y rápido.

Es importante siempre tener claro un protocolo a seguir en caso de ciberataque, Microsoft nos ayuda con numerosas guías como esta que te enseña a crear un punto de restauración del sistema, lo cual puede ser muy útil en caso de ataque o perdida de información.

https://support.microsoft.com/es-es/windows/crear-un-punto-de-restauraci%C3%B3n-del-sist ema-77e02e2a-3298-c869-9974-ef5658ea3be9 o esta que te enseña lo mismo pero bastante más explayado: Manual punto de restauración