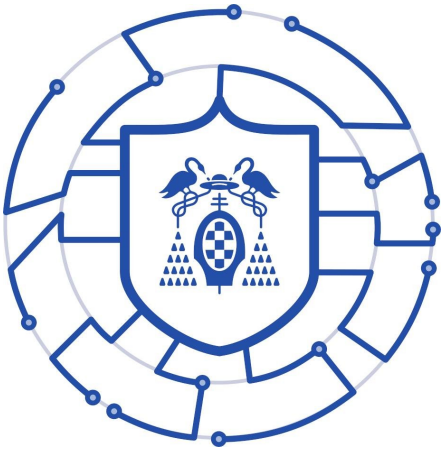
UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR



Seguimiento del estudio del entorno de investigación: Detection Lab

SEGURIDAD OFENSIVA

MÁSTER UNIVERSITARIO EN CIBERSEGURIDAD

José María de la Sen Molina

Febrero de 2021

Índice

Índice de figuras

1. Introducción

El entorno escogido es *Detection Lab*. El motivo principal por el que he escogido este entorno es tratar de familiarizarme con máquinas Windows y el famoso Directorio Activo o *Active Directory*. Me gustaría orientar mi trayectoria profesional hacia el mundo del *pentesting* y a la vez combinar esta idea con el mundo *cloud*, que como es lógico, está (y estará) pegando bastante fuerte. Actualmente trabajo como Ingeniero Cloud DevOps y estoy bastante familiarizado con muchos conceptos de las arquitecturas cloud (Azure), automatización de entornos con Ansible y Python, microservicios con Docker y tecnologías de orquestación como Kubernetes, sistemas de *monitoring* y *logging*, etc. Poco a poco, de forma independiente y autodidacta, me he introducido en el mundo del *pentesting* a través de la plataforma *Hack The Box*, pero siempre con máquinas Linux (Academy, Ready, Laboratory, Passage, etc) y algún que otro challenge sobre web, pero no he llegado a atreverme con máquinas Windows, quizás porque no estoy muy familiarizado con el sistema operativo y/o porque también siento que no sé por dónde cogerlo. Considero que esta es una buena oportunidad para "obligarme" a aprender sobre el sistema operativo Windows y conocer los elementos que componen su arquitectura (editor de registro, API functions, IAT, ADDS, etc). Por lo tanto, el objetivo es el comentado, aprender sobre Windows y su arquitectura, quitarme ese miedo o pereza, y ver si me siento motivado como para un posible TFM que sirva como parte de mi carta de presentación.

El entorno viene integrado con una gran cantidad de herramientas como son:

- *Atomic Red Team*, para simular TTPS (*tactics, techniques and procedures*) y observar métricas o crear nuevas detecciones.
- *AutorunsToWinEventLog*, que permite el análisis y búsqueda en el sistema de logs de eventos de Windows.
- *BadBlood*, para popular un *Active Directory* de forma rápida con cientos de objetos como pueden ser usuarios, ordenadores, etc.
- *Fleet*, *tool* para hacer gestionar, configurar y hacer búsquedas sobre múltiples *hosts*.

- *Microsoft ATA*, herramienta que permite desde la perspectiva *blue team* (defensores) lanzar tests para una futura protección ante ciberataques y mejorar los conocimientos sobre *Advanced Threats*.
- *Mimikatz*, aplicación open source que permite a los usuarios ver y guardar credenciales de autenticación como tickets de *Kerberos*.
- *osquery*, *framework* de instrumentación a bajo nivel del sistema operativo, permitiendo análisis y monitorización.
- *Splunk*, software empleado para centralizar logs, datos y métricas. Una alternativa al *stack ELK (Elastic + Logstash + Kibana)*.
- *Suricata*, robusto motor de detección de amenazas, cuenta con capacidades de IDS, IPS, NSM, etc.
- *Sysmon*, para monitorizar creación de procesos, de conexiones y otros eventos.
- *Velociraptor*, herramienta de monitorización orientada a análisis forense.
- *Windows Event Forwarding*, que sirve como *forwarder* de logs a Splunk.
- *Zeek*, un analizador pasivo de tráfico y además *open source*. Se emplea de forma conjunta con Suricata.

Como podemos ver son herramientas que facilitan la *observabilidad* de los sistemas por lo que considero que aprenderé y adquiriré una gran cantidad de conocimientos.

La arquitectura del entorno se puede apreciar en la figura 1, en ella queda reflejada la disposición de las herramientas anteriormente mencionadas.

Para la instalación del entorno se han seguido las instrucciones del repositorio Detection-Lab, las tecnologías empleadas para levantar la infraestructura han sido VirtualBox, Packer y Vagrant.

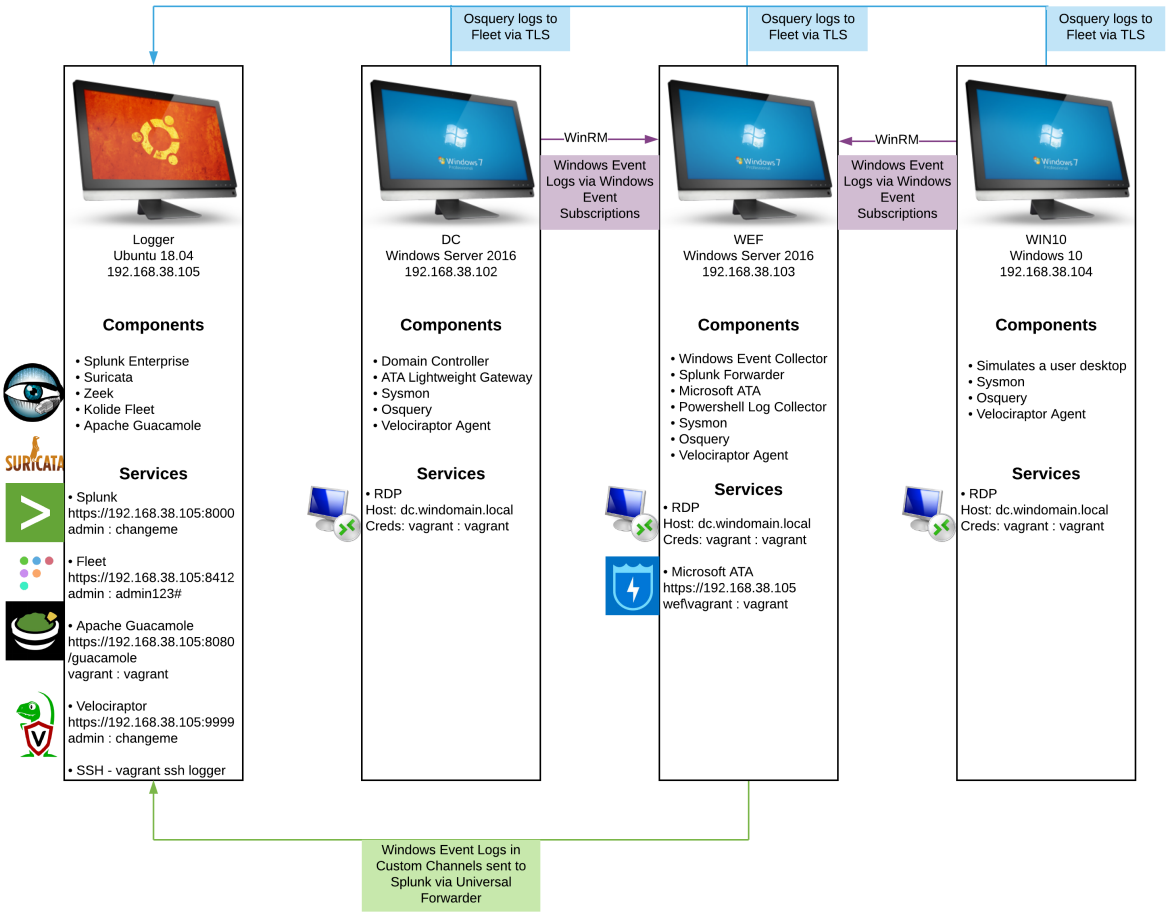


Figura 1: Arquitectura DetectionLab

2. Semanas

2.1. Semana nº1

Periodo: 1 de febrero de 2021 - 7 de febrero de 2021. La primera semana se ha dedicado al estudio y lectura de artículos, presentaciones y tutoriales relacionados con los conceptos, elementos y herramientas que integra el entorno *DetectionLab*. Son los siguientes:

- 2, Accelerating the Analysis of Offensive Security Techniques Using DetectionLab: Dónde Chris Long explica configuraciones y herramientas empleadas. De aquí salte a la aplicación de Splunk *ThreatHunting App*.

- 3, Endpoint Detection Superpowers with Sysmon + Splunk: En dónde se indica y describe la cobertura de técnicas de *Mitre Att&ck* que se puede realizar a través de las herramientas *Splunk*, *Sysmon* y *ThreatHunting*.
- 4, Endpoint detection superpowers on the cheap, Threat Hunting app
- 5, ThreatHunting wiki
- 6, Sysmon

Como he comentado en la introducción mi idea era orientar la investigación en este entorno al *pentesting* sobre máquinas Windows y sobre *Active Directory*. He encontrado un repo, Active Directory Kill Chain Attack & Defense, que incluye una gran cantidad de *TTPs (tactics, techniques and procedures)* empleados para compremeter un *Active Directory*. La idea sería emplear *TTPs* de este repositorio y observar qué eventos se generan en el sistema, cómo se clasifican estos *TTPs* dentro de *MITRE ATT&CK* con la app de *Splunk* de *ThreatHunting* y estructurar mejor la visión y el conocimiento sobre sistemas Windows y redes de *Active Directory*.

2.2. Semana nº2 - Semana nº3

Periodo: 8 de febrero de 2021 - 22 de febrero de 2021. Partiendo del repo mencionado en la primera semana (Active Directory Kill Chain Attack & Defense) y teniendo en cuenta que para este periodo había que orientar el ejercicio a explotar una vulnerabilidad de nuestros entornos decidí revisar los CVEs que venían incluidos. Entre ellos encontré el CVE-2020-1472 aka '*Netlogon Elevation of Privilege Vulnerability*'.

Por poner un poco de contexto, en agosto de 2020 Windows publicó una vulnerabilidad que afectaba a uno de sus servicios, concretamente *Netlogon Remote Protocol*, una interfaz RPC disponible en los controladores de dominio Windows que permite la autenticación de máquinas y usuarios en AD (active directory). Lo interesante de Netlogon es que emplea un protocolo criptográfico personalizado, a diferencia de otros servicios RPC, como consecuencia implementaciones antiguas del sistema operativo (Windows NT). En este procedimiento existe un uso inseguro de *AES-CFB8*, tanto cliente como servidor emplean este protocolo en una función conocida como *ComputeNetlogonCredential*. El protocolo *AES-CFB8* encripta cada byte del texto plano anteponiendo un vector de inicialización (IV) de 16 bytes, continuadamente aplica AES a los primeros 16 bytes de la concatenación IV+texto plano, tomando el primer byte de la salida de AES y realizando una XOR con el siguiente byte del texto plano, iterando de este modo hasta que todos los bytes del mensaje están encriptados, se puede ver este procedimiento en la figura 1. El protocolo parece seguro mientras el atacante no conozca la clave de sesión (previamente calculada/pactada por cliente y servidor), ya que no podrá ni calcular ni adivinar la salida en función de una entrada determinada.

En el caso de la función *ComputeNetlogonCredential* se establecen los vectores de inicialización siempre a 16 *zero-bytes*, violando los requerimientos para usar AES-CFB8 de forma segura. Su seguridad únicamente se mantiene cuando los IVs son aleatorios, por lo que en caso contrario permitirá al atacante jugar con una probabilidad de predecir (aunque baja) y obtener un bloque de todo ceros con una clave aleatoria. Para 1 de las 256 claves (8 bits, 1 byte), con una entrada de todo ceros en texto plano la salida de la encriptación *AES-CFB8* será de todo ceros para el texto cifrado. De hecho, generalizando esta propiedad: cuando un IV consista en solo ceros habrá un entero $0 \leq X \leq 255$ para el cual se mantenga que para un texto plano que comience con n bytes de valor X el texto cifrado comenzará por n bytes

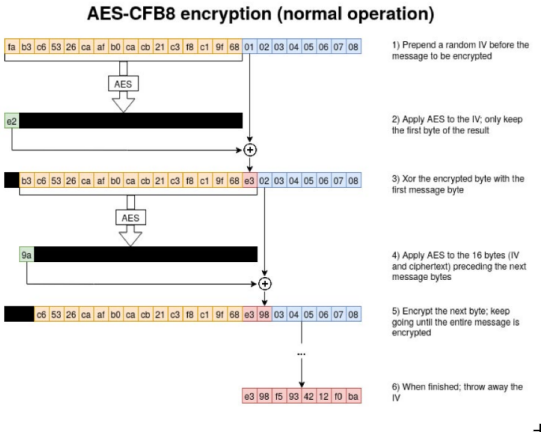


Figura 2: Encriptación de mensaje empleando el protocolo AES-CFB8 [1]

de valor 0 [1].

Teniendo en cuenta todo esto se puede construir un *exploit* a partir del simple hecho de que teniendo un entrada de todo ceros en algún momento podemos obtener una salida de todo ceros.

Este ataque consistiría en jugar con lo que espera el servidor a nivel de parámetros, en primer lugar organizando un *spoofing* sobre las credenciales del cliente tras el intercambio de desafíos y aprovechando que las cuentas de equipo no se bloquean tras X intentos para lanzar una ataque de fuerza de bruta con un promedio de 256 intentos (256 claves) y con una duración aproximada de 3 segundos, es decir, en este intervalo de tiempo tan corto podríamos iniciar sesión como cualquier equipo del dominio. Además es necesario señalar que se permite este bypass debido a que la lógica RPC de firma y sellado se puede deshabilitar, ya que en el spoofing de la primera iteración no conocemos (ni conoceremos) la clave de sesión que se emplea en el mecanismo de cifrado de transporte de Netlogon (lógica RPC de firma y sellado).

Finalmente, a través de la llamada *NetrServerPasswordSet2* se puede organizar un cambio de password al equipo del AD (active directory) con el que hemos iniciado conexión. La contraseña de texto está conformada por 516 bytes de los cuáles los últimos 4 indican la longitud de la contraseña y está cifrada por la clave de sesión (que desconocemos). Si envían 516 ceros, se descifrarán 516 ceros y con ello una contraseña de longitud cero (debido a los últimos 4 bytes), de modo que también conocemos el hash de cero y podremos *loguearnos* en la máquina. A partir de aquí podemos hacer login con la nueva contraseña (0) y hacer

un cambio de contraseña tanto en máquinas del dominio como en el propio controlador de dominio, si se da en este último tipo la escalada de privilegios será total [7].

Pasamos a la parte práctica, en primer lugar he creado una *tool dockerizada* con todas las dependencias necesarias y varias herramientas/repos de terceros instaladas:

- SecuraBV/CVE-2020-1472
- bb00/zer0dump
- dirkjanm/CVE-2020-1472
- winrm

He subido el *dockerfile* y el fichero de dependencias a un repositorio propio que utilizaré a lo largo de la asignatura, cedelasen/ofensivaPoc.

```
FROM ubuntu:18.04

RUN apt update -y
RUN apt install python3-pip python3-venv iputils-ping git ruby-full -y

RUN python3 -m pip install virtualenv

RUN gem install winrm winrm-fs stringio

WORKDIR /root

RUN git clone https://github.com/SecuraBV/CVE-2020-1472.git SecuraBV && \
    git clone https://github.com/dirkjanm/CVE-2020-1472.git dirkjanm && \
    git clone https://github.com/bb00/zer0dump.git zer0dump && \
    git clone https://github.com/Hackplayers/evil-winrm.git evil-winrm

RUN chmod +x ./*/*.py

ENV VIRTUAL_ENV=/opt/venv
RUN python3 -m venv $VIRTUAL_ENV
ENV PATH="$VIRTUAL_ENV/bin:$PATH"

COPY requirements.txt .
RUN pip3 install -r requirements.txt

ENTRYPOINT  ["/bin/bash"]
```

Figura 3: *Tool's dockerfile*

En segundo lugar, he utilizado la utilidad (SecuraBV/CVE-2020-1472) de la empresa Secura para checkear si el *domain controller* del entorno *DetectionLab* es vulnerable a *Zerologon*. He lanzado el propio *script* del repo contra el servidor *domain controller* "*dc*" alojado en *192.168.38.102*, como podemos ver en la figura 4.

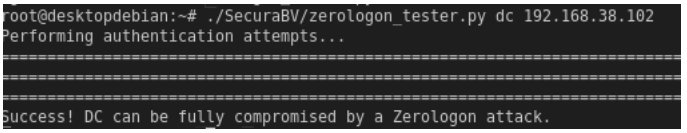


Figura 4: *Script* para *checkear* si el *domain controller* es vulnerable a *Zerologon*

En tercer lugar, podemos emplear la utilidad del repo bb00/zer0dump para extraer el *hash NTLM* de la cuenta administradora *Administrator*.

El *hash NTLM* sería *e02bc503339d51f71d913c245d35b50b*.

Figura 5: Extracción del hash *NTLM* de la cuenta *Administrator*

En cuarto lugar podemos incluso emplear este *hash* para abrir una consola a través de la utilidad Evil-WinRM como se puede apreciar en la figura 6.

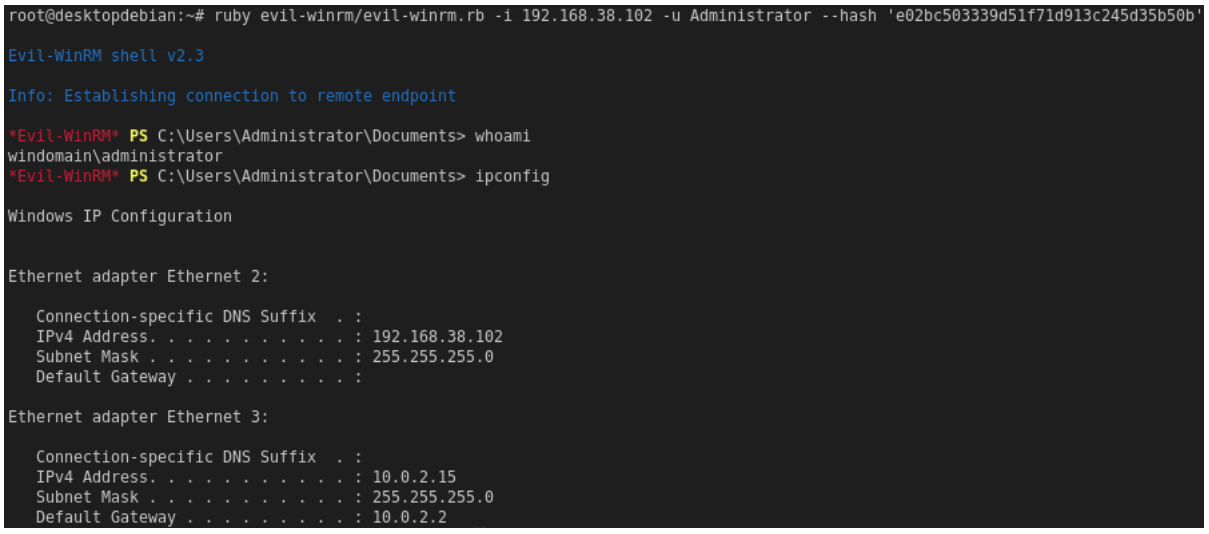


Figura 6: Apertura de consola con *Evil-WinRm*

Finalmente podemos emplear *exploit* provisto en el repo dirkjanm/CVE-2020-1472 para cambiar la contraseña del *domain controller* a 0. (Una vez cambiada la contraseña de la cuenta de máquina de un controlador de dominio, este pierde toda la confianza que tenía con los servidores a su alrededor, esto implicaría que los servidores no puedan autenticarse de nuevo contra el dominio.)

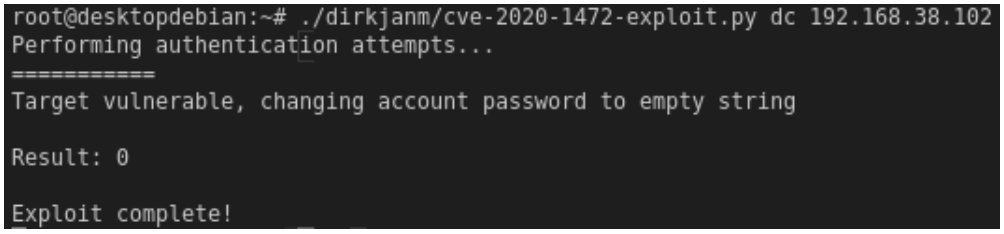


Figura 7: *Exploit* de *Zerologon*

Como se puede ver no he utilizado la herramienta *Metasploit*, me apetecía más trastear con los repos, y además ya he utilizado esta herramienta cuando he estado jugando con *HackTheBox* por mi cuenta. En las siguientes iteraciones del estudio ofensivo se verán técnicas para ganar persistencia y a la vez se tratará emplear alguna de las herramientas de DetectionLab destinadas a la visibilidad y estudio del sistema Windows.

Bibliografía

[1] T. Tervoort. (2020) Zerologon: Unauthenticated domain controller compromise by subverting netlogon cryptography (cve-2020-1472). [Online]. Available: https://www.secura.com/pathtoimg.php?id=2055

[2] C. Long. (2021, 1) Accelerating the Analysis of Offensive Security Techniques Using DetectionLab . [Online]. Available: https://jsac.jpcert.or.jp/archive/2021/pdf/JSAC2021_102_chris_en.pdf

[3] O. Hartong. Endpoint Detection Superpowers with Sysmon + Splunk . [Online]. Available: https://conf.splunk.com/files/2019/slides/SEC1620.pdf

[4] ——. Endpoint detection superpowers on the cheap, Threat Hunting app . [Online]. Available: https://conf.splunk.com/files/2019/slides/SEC1620.pdf

[5] ——. ThreatHunting wiki . [Online]. Available: https://github.com/olafhartong/ThreatHunting/wiki

[6] Microsoft. Sysmon . [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

[7] P. G. Pérez. (2020, 09) Hacking windows con zerologon: Vulnerabilidad crítica que puede comprometer tu domain controller. [Online]. Available: https://www.elladodelmal.com/2020/09/hacking-windows-con-zerologon.html

[8] infosecn1nja. Active Directory Kill Chain Attack & Defense . [Online]. Available: https://github.com/infosecn1nja/AD-Attack-Defense

[9] (2020) Cve-2020-1472. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1472

[10] (2020) Cve-2020-1472 — netlogon elevation of privilege vulnerability. [Online]. Available: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1472

[11] CVE-2020-1472 - Zerologon: Por qué debes priorizar el parcheo antes que la detección. [Online]. Available: https://www.flu-project.com/2020/09/zerologon.html