

Esercizio 6:

Estrarre un Eseguibile da un PCAP

Obiettivi

- **Parte 1** Analizzare Log e Catture di Traffico Pre-catturati
- **Parte 2** Estrarre File Scaricati dal PCAP

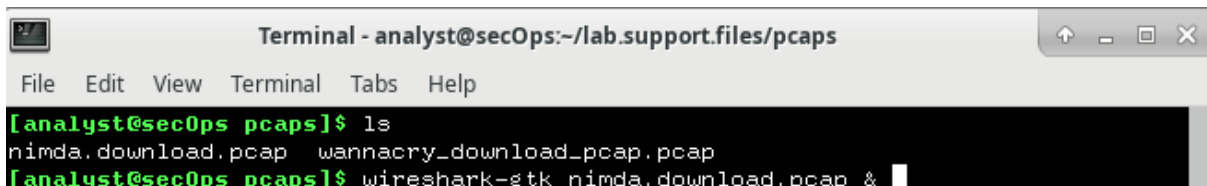
Contesto / scenario:

Verra' analizzato il traffico relativo al download di un malware in un file **PCAP** ed estratto un file eseguibile.

Obiettivo: Comprendere come avvengono le transazioni di rete a livello di pacchetto.

PARTE 1: Analizzare Log e Catture di Traffico Pre-catturati

Apertura del file PCAP:



```
Terminal - analyst@secOps:~/lab.support.files/pcaps
File Edit View Terminal Tabs Help
[analyst@secOps pcaps]$ ls
nimda.download.pcap  wannacry_download_pcap.pcap
[analyst@secOps pcaps]$ wireshark-gtk nimda.download.pcap &
```

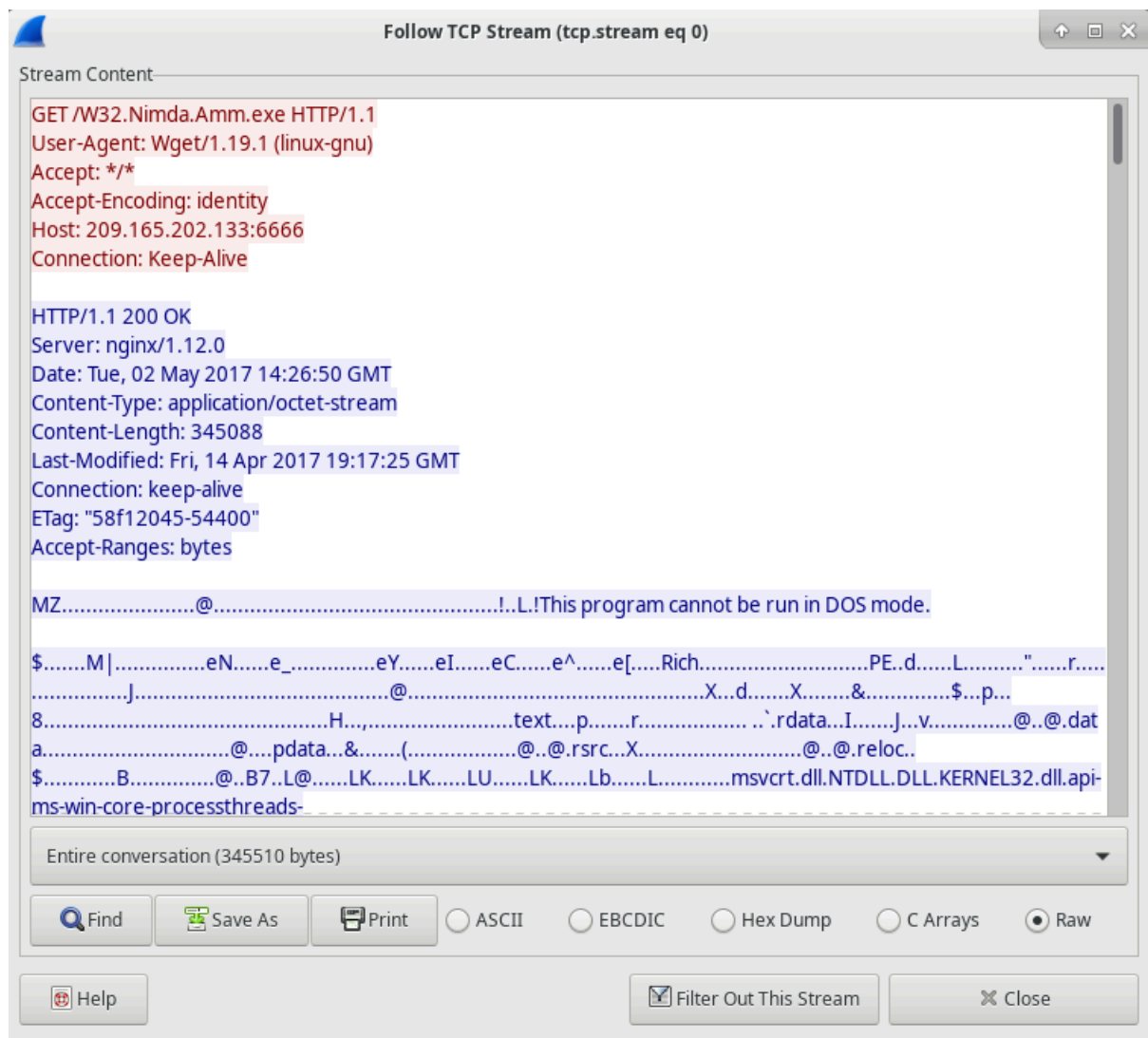
Primi 4 pacchetti della comunicazione:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	209.165.200.235	209.165.202.133	TCP	74	48598 → 6666 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4051203246 TSecr=0 WS=512
2	0.000259	209.165.202.133	209.165.200.235	TCP	74	6666 → 48598 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3023496465 TSecr=4051203246 WS=512
3	0.000297	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=4051203246 TSecr=3023496465
4	0.000565	209.165.200.235	209.165.202.133	HTTP	230	GET /W32.Nimda.Amm.exe HTTP/1.1

Questa cattura contiene l'inizio della comunicazione, in dettaglio:

- **Primi 3 pacchetti:** Questi rappresentano l'handshake a 3 fasi del TCP (**SYN**, **SYN-ACK**, **ACK**) e sono responsabili di ogni inizio di connessioni di tipo TCP.
- **Pacchetto n. 4:** Il quarto pacchetto rappresenta la richiesta fatta tramite **HTTP** usando il verbo **GET** per scaricare il malware.

Follow del TCP stream:



E' stata ricostruita la transazione TCP scegliendo con il tasto destro del mouse **follow > TCP stream**.

Wireshark visualizza un'altra finestra contenente i dettagli per l'intero flusso TCP selezionato.

Ciò che viene mostrato nella finestra e' **il contenuto integrale della comunicazione TCP intercettata**, in questo caso tra un client che richiede un file **.exe** e un server che lo fornisce.

Domanda: Cosa sono tutti quei simboli mostrati nella finestra Follow TCP Stream? Sono rumore di connessione? Dati? Spiega. Ci sono alcune parole leggibili sparse tra i simboli. Perché sono lì?

R:

```
MZ.....@.....!..L!This program cannot be run in DOS mode.
```

- La prima riga dopo la risposta del server, e' l'intestazione del file `.exe`, dove "MZ" è la **firma dei file PE (Portable Executable)** di Windows.
- Sono presenti **simboli e caratteri "casuali"** perché Wireshark mostra **tutti i byte**, inclusi i binari non stampabili, e tenta di rappresentarli come caratteri **ASCII** o **UTF-8**, ma spesso questi byte **non corrispondono a simboli leggibili**, quindi appaiono anche in questo modo. Ecco un esempio:

```
$.....M|.....eN.....e_.....eY.....eI.....eC.....e^.....e[.....Rich.....PE..d.....L.....".....r.....  
.....J.....@.....X..d.....X.....&.....$..p.....  
8.....H.....text....p.....r.....`..rdata...I.....]..v.....@..@.dat
```

- Oltre ai "simboli" e' possibile ritrovare alcune stringhe comprensibili. Alcune **stringhe ASCII** sono effettivamente contenute nel file `.exe`, come nomi di API o DLL (`KERNEL32.dll`, `msvcrt.dll`, ecc.).

I file binari spesso includono **metadati** o **stringhe di codice** visibili (messaggi di errore, nomi di funzione, percorsi). Eccone un esempio:

```
...APerformArithmeticOperation: '%c'  
..=...%0.1C.....S.o.f.t.w.a.r.e.\C.l.a.s.s.e.s.....N.T.D.L.L...D.L.L.....NtQueryInformationProcess...
```

Nonostante il nome `W32.Nimda.Amm.exe`, questo eseguibile non è il famoso worm. Per motivi di sicurezza, questo è un altro file eseguibile che è stato rinominato come `W32.Nimda.Amm.exe`.

Domanda: Usando i frammenti di parole visualizzati nella finestra Follow TCP Stream di Wireshark, puoi dire quale eseguibile sia realmente?

R: All'interno della finestra Follow TCP Stream sono presenti varie stringhe leggibili, ognuna di esse ci da informazioni diverse utili a individuare il vero contenuto. Ognuna di esse e' riportata e spiegata di seguito:

```
.....4...V.S._V.E.R.S.I.O.N_...I.N.F.O.....jD.....jD.?......String.File.Info.....0.4.0.9.0.4.B.  
0...L...Company.Name...Microsoft.Corporation...L...File.Description...Windows.Command.Processor...File.Version...  
6...1...7.6.0.1...1.7.5.1.4...(.win7.sp1.rtm.1.0.1.1.19~1.8.5.0).....  
(...Internal.Name...cmd.....Legal.Copy.righ.t....Microsoft.Corporation...All.rights.reserved....  
8...Original.File.name...Cmd.Exe.j%...Product.Name...Microsoft...Windows...Operating.System...B...Product.Version...  
6...1...7.6.0.1...1.7.5.1.4...D...Var.File.Info...$...Translation.....l.7...0...@.../..!
```

Metadati PE: Contiene dati dalla struttura del file eseguibile, tra cui:

- **CompanyName:** Microsoft Corporation
- **FileDescription:** Windows Command Processor
- **InternalName:** cmd
- **OriginalFilename:** Cmd.exe
- **ProductName:** Microsoft Windows Operating System

Queste informazioni potrebbero bastare a affermare con adeguata sicurezza che non si tratta di un malware ma invece del file eseguibile **cmd.exe di windows**. Ma sono presenti altre informazioni leggibili che meritano un'analisi:

```

<!-- Copyright (c) Microsoft Corporation -->
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity
  version="5.1.0.0"
  processorArchitecture="amd64"
  name="Microsoft.Windows.FileSystem.CMD"
  type="win32"
/>
<description>Windows Command Processor</description>

```

Questo è il **manifesto integrato** in un eseguibile Windows, usato per definire:

- **Livello privilegi**
- **Nome del programma** (`Microsoft.Windows.FileSystem.CMD`)

Infine, è presente anche una sezione che contiene svariati comandi del CMD, togliendo qualsiasi dubbio.

```

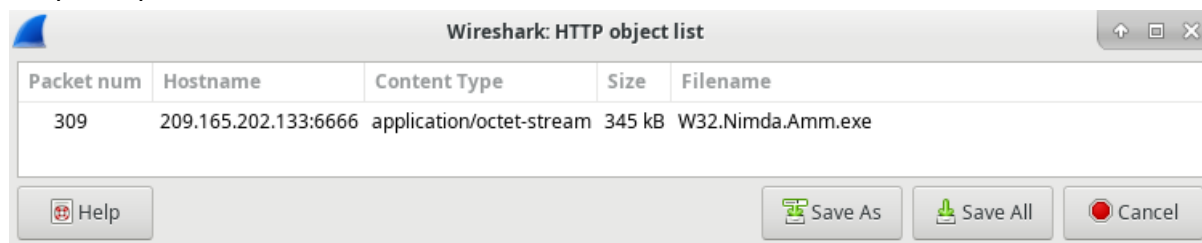
.....CLS...DEL...DIR...FOR...NOT...REM...REN...SET...VER...VOL...D.PATH...CALL...CD...COLOR...TITLE...CHD
.IR...PU.SHD...A.S.SOC...F.TY.PE...E.R.A.S.E...I.F...M.K.D.I.R...M.D...PA.U.S.E...R.D...D.E.F.I.N.E.D...C.O.P.Y...P.A.T.H...P.R.O.M.P.T...P.O.P.D...D.A
.T.E...E.C.H.O...E.X.I.T...E.X.I.S.T...B.R.E.A.K...G.O.T.O...K.E.Y.S...M.O.V.E...R.E.N.A.M.E...R.M.D.I.R...S.H.I.F.T...S.T.A.R.T...T.I.M.E...T.Y.P.E...
V.E.R.T.I.F.Y...M.K.I.T.N.K...E.N.D.I.L.O.C.A.L...F.R.R.O.R.I.E.V.F.I...S.E.T.I.L.O.C.A.L...C.M.D.E.X.T.V.E.R.S.I.O.N

```

Svolgimento parte 2: Estrarre File Scaricati dal PCAP

Con il pacchetto della richiesta **GET** selezionato, si naviga su **File > Export Objects > HTTP**, dal menu di **Wireshark**.

Si aprirà questa schermata:

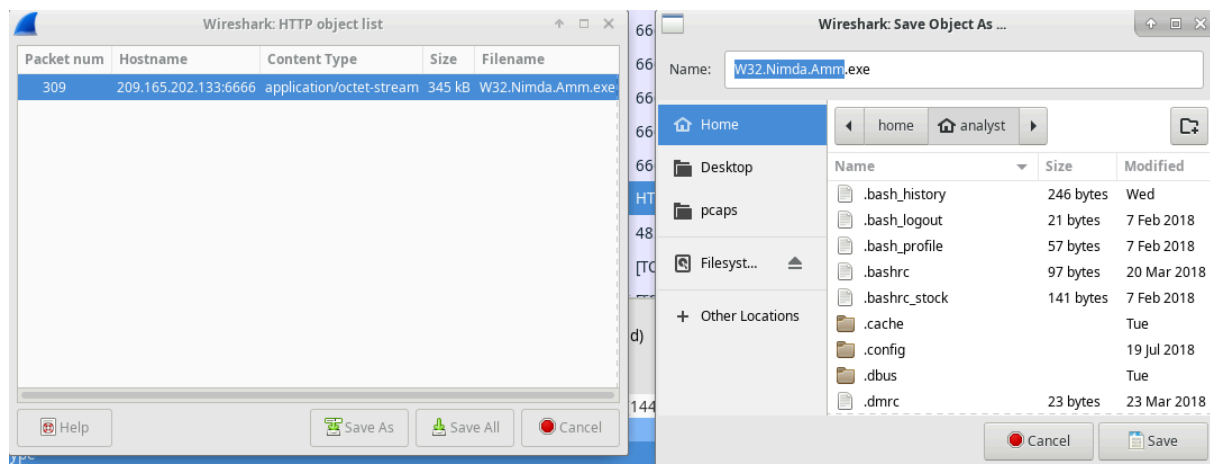


La finestra mostrata da Wireshark contiene tutti gli oggetti **HTTP** presenti nel flusso **TCP** che contiene la richiesta **GET**.

Domanda: Perché W32.Nimda.Amm.exe è l'unico file nella cattura?

W32.Nimda.Amm.exe è l'unico file presente nella cattura perché è l'unico oggetto scaricato tramite una richiesta HTTP GET in quel particolare flusso TCP. Wireshark mostra solo i file effettivamente trasferiti tramite HTTP nei pacchetti catturati. Se nessun altro file è stato scaricato via HTTP durante la sessione monitorata, questo risulterà l'unico esportabile.

Salvataggio del file:



Controllo dell'effettivo salvataggio del file:

```
[analyst@secOps ~]$ ls -l
total 368
-rw-r--r-- 1 root root 6869 Jun 10 10:44 capture.pcap
drwxr-xr-x 2 analyst analyst 4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst analyst 4096 Mar 22 2018 Downloads
drwxr-xr-x 9 analyst analyst 4096 Jul 19 2018 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 315 Jun 12 10:15 space.txt
-rw-r--r-- 1 analyst analyst 345088 Jun 16 13:57 W32.Nimda.Amm.exe
[analyst@secOps ~]$
```

Il file e' stato salvato correttamente e nella giusta directory.

Verifica del tipo di file:

Tramite il comando di linux "`file ./[nome-file]`":

```
[analyst@secOps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
```

Come visto sopra, `W32.Nimda.Amm.exe` è di fatto un file eseguibile di Windows.

Domanda: Nel processo di analisi del malware, quale sarebbe un probabile passo successivo per un analista di sicurezza?

Un probabile passo successivo per un analista di sicurezza sarebbe eseguire un'**analisi statica** del file salvato.

Ad esempio:

- **Verifica dell'hash del file (MD5, SHA256):**
Utile per confrontarlo con database di minacce note (come **VirusTotal**).
- **Usare strumenti come `strings`, `objdump`, o `die`:**
Utile per identificare sezioni, compressioni o packer.

In seguito, si potrebbe procedere a un'**analisi dinamica** in ambiente controllato (sandbox o VM isolata) per osservare il comportamento del file in esecuzione: modifiche al file system, traffico di rete, creazione di processi, ecc.

Conclusione:

L'analisi del traffico di rete tramite file PCAP consente di comprendere in modo dettagliato le dinamiche dietro il download di un file eseguibile. Attraverso tecniche di ricostruzione dei flussi e ispezione dei dati grezzi, è possibile identificare con precisione la natura dei file trasferiti e valutarne la legittimità. Questa pratica rappresenta una competenza fondamentale per chi si occupa di sicurezza informatica.