

# Report Esercizio 6: Estrarre un Eseguibile da un PCAP

A cura di Iris Canole, Federico Giannini, Daniele Castello, Luca Pani, Rosario Papa, Yari Olmi, Alessandro Salerno

## Obiettivi

### PARTE 1:

- Cercare all'interno della CyberOps WorkStation il file `pcap nimda.download.pcap`.
- Rispondere alle domande.

### PARTE 2:

- Estrarre File Scaricati dal `PCAP`

## Svolgimento Parte 1

Come prima cosa cercare la directory `lab.support.files`:

```
[analyst@secOps ~]$ ls
capture.pcap  Desktop  Downloads  lab.support.files  scripts  second_drive  yay
```

Spostarsi all'interno della directory con il comando “cd” e vedere cosa si trova all'interno di quest'ultima.

```
[analyst@secOps ~]$ cd lab.support.files/
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log      instructor          pox
applicationX_in_epoch.log letter_to_grandma.txt  sample.img
attack_scripts          logstash-tutorial.log sample.img_SHA256.sig
confidential.txt          malware            scripts
cyops.mn                 mininet_services    SQL_Lab.pcap
elk_services              openssl_lab
h2_dropbear.banner        pcaps
```

La directory che ci interessa è chiamata `pcaps`.

```
[analyst@secOps lab.support.files]$ cd pcaps/
[analyst@secOps pcaps]$ ls
nimda.download.pcap  wannacry_download_pcap.pcap
```

All'interno della directory troviamo il file che stavamo cercando e con il comando qui sotto lo apriamo tramite Wireshark.

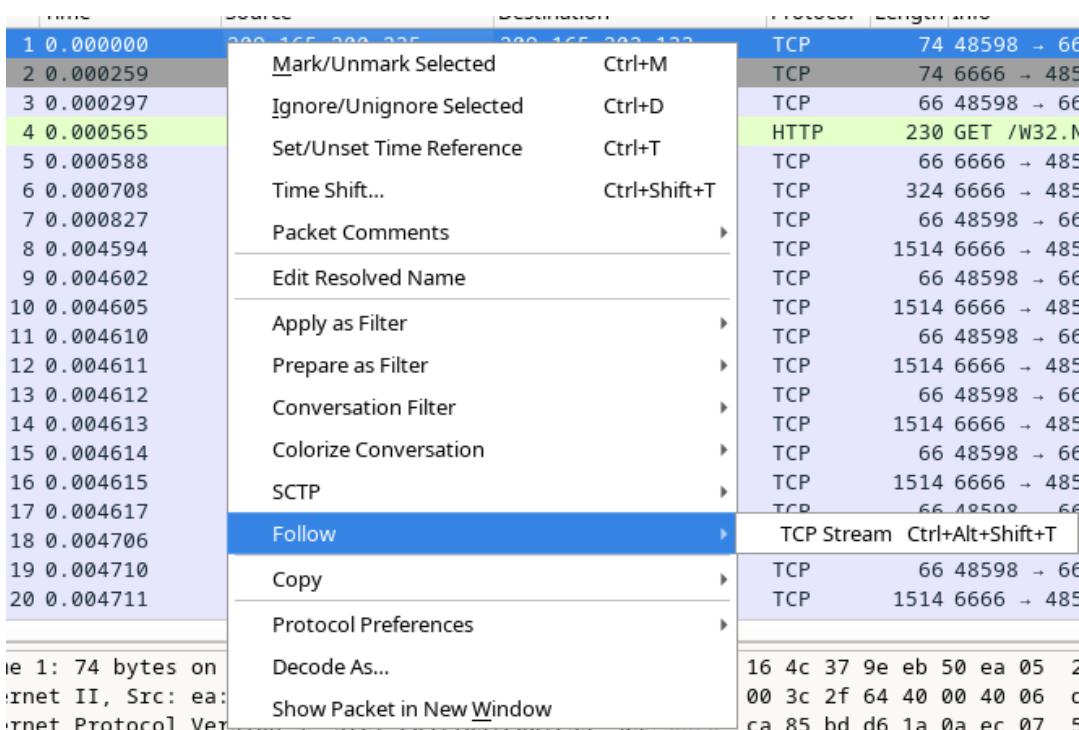
```
[analyst@secOps pcaps]$ wireshark nimda.download.pcap &
```

1 0.000000	209.165.200.235	209.165.202.133	TCP	74 48598 → 6666 [SYN] Seq=0 Win=29
2 0.000259	209.165.202.133	209.165.200.235	TCP	74 6666 → 48598 [SYN, ACK] Seq=0 Ack=1
3 0.000297	209.165.200.235	209.165.202.133	TCP	66 48598 → 6666 [ACK] Seq=1 Ack=1
4 0.000565	209.165.200.235	209.165.202.133	HTTP	230 GET /W32.Nimda.Amm.exe HTTP/1.1
5 0.000588	209.165.202.133	209.165.200.235	TCP	66 6666 → 48598 [ACK] Seq=1 Ack=16
6 0.000708	209.165.202.133	209.165.200.235	TCP	324 6666 → 48598 [PSH, ACK] Seq=1 Ack=16
7 0.000827	209.165.200.235	209.165.202.133	TCP	66 48598 → 6666 [ACK] Seq=16 Ack=16

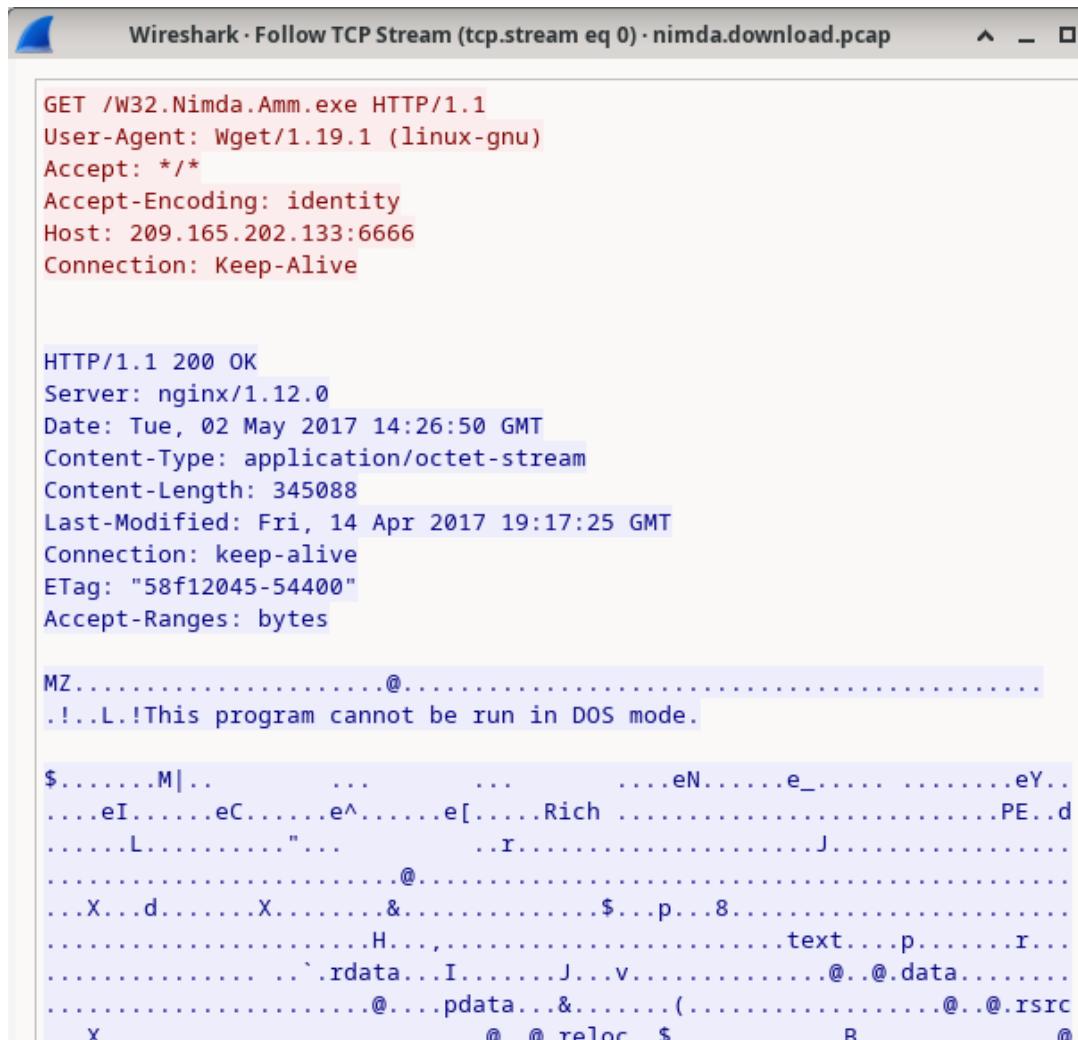
Questa cattura di Wireshark contiene l'inizio della comunicazione, in dettaglio:

- 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.

Selezionando il primo pacchetto TCP nella cattura, un pacchetto SYN. Fai clic con il pulsante destro del mouse su di esso e scegli Follow > TCP Stream.



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



```
GET /W32.Nimda.Amm.exe HTTP/1.1
User-Agent: Wget/1.19.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 209.165.202.133:6666
Connection: Keep-Alive

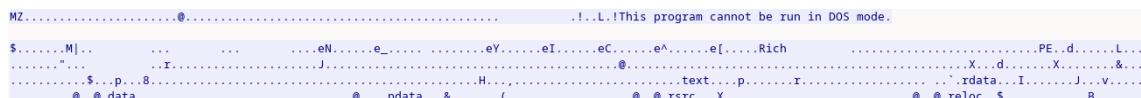
HTTP/1.1 200 OK
Server: nginx/1.12.0
Date: Tue, 02 May 2017 14:26:50 GMT
Content-Type: application/octet-stream
Content-Length: 345088
Last-Modified: Fri, 14 Apr 2017 19:17:25 GMT
Connection: keep-alive
ETag: "58f12045-54400"
Accept-Ranges: bytes

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

$.....M|...      ...      ...      ...eN.....e_.....eY...
....eI.....eC.....e^.....e[.....Rich .....PE..d
.....L....."....      .r.....J.....
.....@.....
...X..d.....X.....&.....$...p...8...
.....H.....text...p.....r...
.....`rdata...I.....J...v.....@...@.data...
.....@...pdata...&.....(.....@...@.rsrc
.....X.....@...@.reloc...$.....R.....@...
```

**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?

**RISPOSTA:** I simboli che si vedono nella finestra Follow TCP Stream non sono rumore o dati casuali ma sono la rappresentazione grezza (raw data) del file che è stato scaricato.



```
MZ.....@.....
.!.L.!This program cannot be run in DOS mode.
$.....M|...      ...      ...      ...eN.....e_.....eY...
....eI.....eC.....e^.....e[.....Rich .....PE..d.....L...
....."....      .r.....J.....
.....@.....
...X..d.....X.....&.....$...p...8...
.....H.....text...p.....r...
.....`rdata...I.....J...v.....@...@.data...
.....@...pdata...&.....(.....@...@.rsrc
.....X.....@...@.reloc...$.....R.....@...
```

La maggior parte dei simboli illeggibili, come la chiocciola, o i punti e i caratteri fuori standard, sono la rappresentazione ASCII di dati binari non stampabili. Un file eseguibile come l' `.exe` che viene scaricato è composto principalmente da codice macchina.

Mentre le parole leggibili sono lì perché i programmati includono queste stringhe leggibili all'interno

del codice binario. Wireshark le estrae e le rende visibili poiché converte la sequenza di bit e byte che viaggiano sui cavi di rete in informazioni comprensibili, suddividendole per protocollo e mostrandone il contenuto dettagliato.

.msvcrt.dll.NTDLL.DLL.KERNEL32.dll.api-ms-win-core-processsthreads-l1-1-0.DLL.WINBRAND.dll

**DOMANDA SFIDA:** 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`.

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

**RISPOSTA:** Nonostante non sia il vero worm possiamo capire che il punto più chiaro e diretto per vedere il nome del file è nella richiesta HTTP inviata dal client al server ossia:

`GET /W32.Nimda.Amm.exe HTTP/1.1`

`/W32.Nimda.Amm.exe` è il percorso e il nome esatto del file eseguibile richiesto e scaricato.

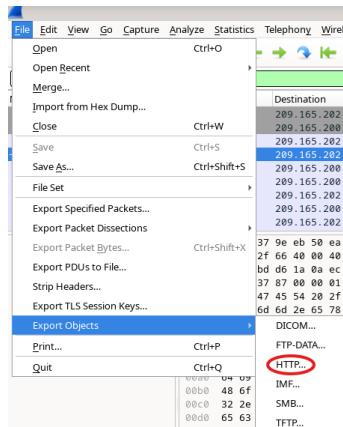
Invece scorrendo più giù troviamo questa stringa:

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

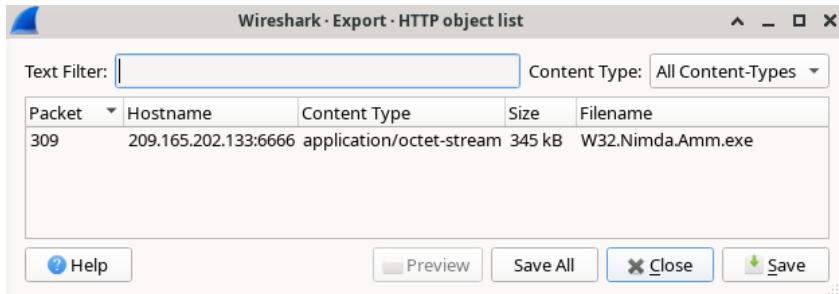
Dove le prime due lettere del payload, ossia MZ, identificano un file eseguibile Portable Executable (PE) di Windows. Mentre “this program cannot be run in DOS mode.” è parte integrante dell'intestazione di un eseguibile Windows, confermando che i dati binari seguenti sono effettivamente il programma `.exe` che corrisponde al nome `W32.Nimda.Amm.exe`.

## Svolgimento Parte 2

Andando su `File > Export Objects > HTTP`, dal menu di Wireshark



Vedremo questa schermata:



Packet	Hostname	Content Type	Size	Filename
309	209.165.202.133:6666	application/octet-stream	345 kB	W32.Nimda.Amm.exe

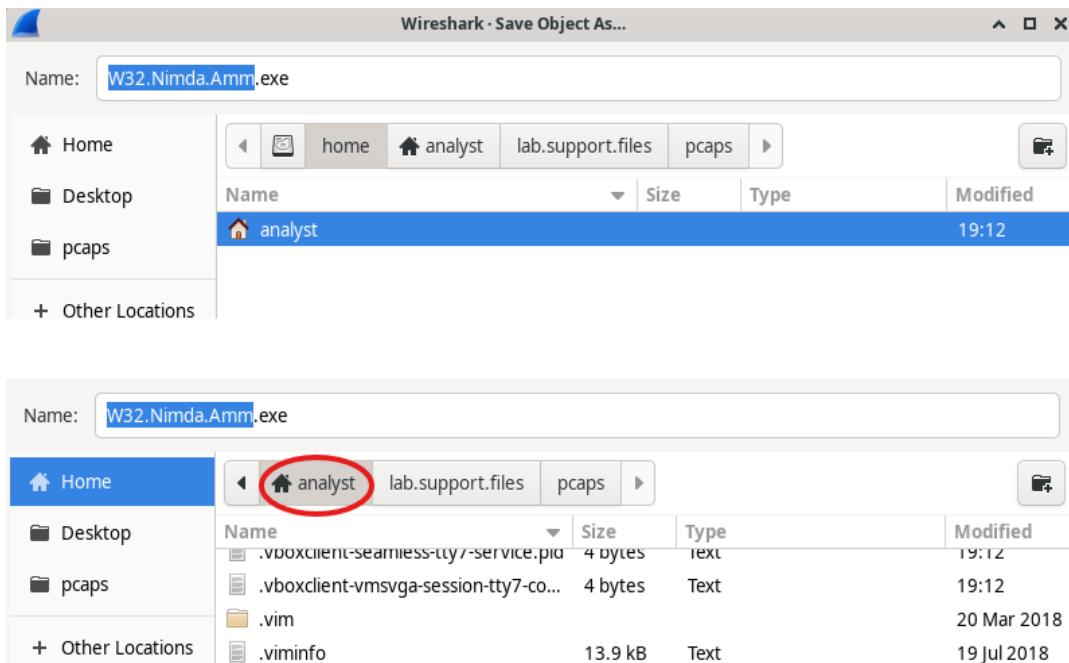
**DOMANDA:** Perché `W32.Nimda.Amm.exe` è l'unico file nella cattura?

**RISPOSTA:** `W32.Nimda.Amm.exe` è l'unico file visualizzato nella lista "HTTP object list" di Wireshark perché la lista degli oggetti HTTP non mostra automaticamente tutto il traffico della cattura, ma solo gli oggetti identificati come trasferimenti HTTP completi. L'oggetto `W32.Nimda.Amm.exe` è stato scaricato tramite una richiesta `HTTP/1.1` ed è stato identificato dal server con il Content-Type: `application/octet-stream`.

**Content-Type: application/octet-stream**

### Salvataggio file:

Salviamo il file `W32.Nimda.Amm.exe` nella cartella `analyst`.



Wireshark · Save Object As...

Name: `W32.Nimda.Amm.exe`

Home 

Desktop pcaps

+ Other Locations

analyst 

Name: `W32.Nimda.Amm.exe`

Home  lab.support.files pcaps

Desktop pcaps

+ Other Locations

Name	Size	Type	Modified
<code>.vboxclient-seamless-tty7-service.pna</code>	4 bytes	Text	19:12
<code>.vboxclient-vmsvga-session-tty7-co...</code>	4 bytes	Text	19:12
<code>.vim</code>			20 Mar 2018
<code>.viminfo</code>	13.9 kB	Text	19 Jul 2018

Assicuriamoci di aver fatto tutto nel modo corretto andando a vedere da terminale:

```
[analyst@secOps ~]$ ls -l
total 364
drwxr-xr-x 2 analyst analyst 4096 Jun 17 19:35 Desktop
drwxr-xr-x 3 analyst analyst 4096 Jun 18 20:17 Downloads
drwxr-xr-x 9 analyst analyst 4096 Jun 18 20:17 lab.support.files
drwxr-xr-x 3 analyst analyst 4096 Jun 18 19:55 scripts
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 345088 Dec 9 19:57 W32.Nimda.Amm.exe
drwxr-xr-x 5 analyst analyst 4096 Jun 18 19:27 yay
```

Tramite il comando “file” possiamo vedere più informazioni sul tipo di file:

```
[analyst@secOps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable for MS Windows 6.01 (console), x86-64, 6 sections
```

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

**RISPOSTA:** Il probabile passo successivo, dopo aver identificato il download di un eseguibile malevolo come `W32.Nimda.Amm.exe` e aver confermato il suo tipo di contenuto, sarebbe:

- Analisi statica
  - **Hashing e Database di Minacce:** Calcolare l'hash del file (es. SHA256) e confrontarlo con database di minacce noti (come VirusTotal). Questo confermerebbe immediatamente che si tratta del worm Nimda e fornirebbe un riepilogo delle sue capacità note.
  - **Identificazione di Stringhe:** Estrarre tutte le stringhe leggibili rimanenti dal codice binario. Queste possono rivelare:
    - ◆ Nomi di file o chiavi di registro che il malware crea.
    - ◆ URL o indirizzi IP a cui tenta di connettersi.
    - ◆ Messaggi di errore o nomi di funzioni API di Windows che intende chiamare.
- Analisi dinamica
  - **Ambiente Isolato:** Eseguire l'eseguibile su un sistema virtuale appositamente configurato (sandbox).
  - **Monitoraggio del Comportamento:** Monitorare e registrare ogni azione del malware:
    - ◆ **Modifiche al File System:** Quali file crea, elimina o modifica.
    - ◆ **Modifiche al Registro di Sistema:** Quali chiavi crea o modifica per ottenere la persistenza (avvio automatico).
    - ◆ **Traffico di Rete:** Se tenta di connettersi ad altri host (come un server C2 o tentativi di scansione/propagazione, tipici di Nimda).
    - ◆ **Iniezioni di Codice:** Se cerca di iniettare codice in altri processi.

