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.

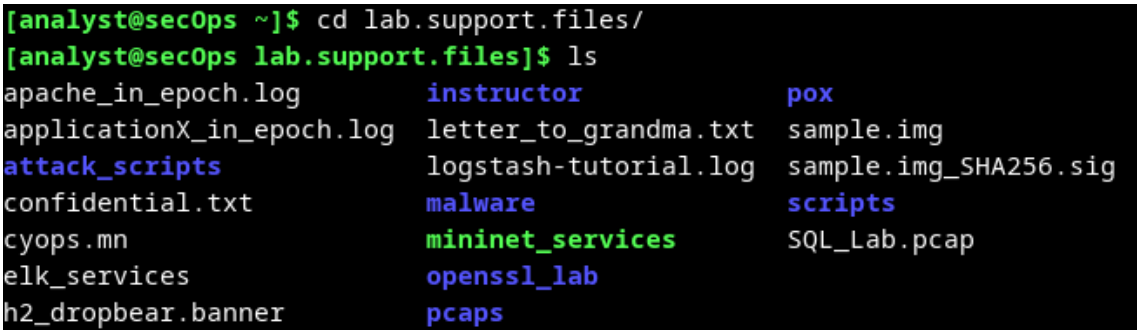
PARTE 2:

- Estrarre File Scaricati dal `PCAP`

Svolgimento Parte 1

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

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

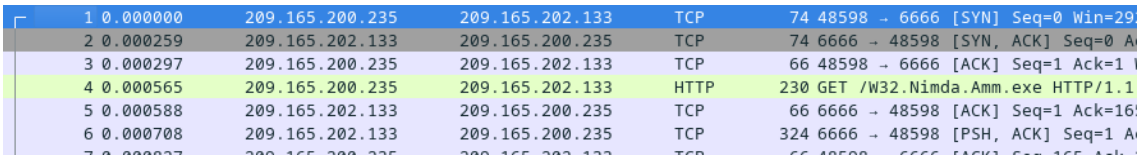


La directory che ci interessa è chiamata `pcaps`.

```
[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.

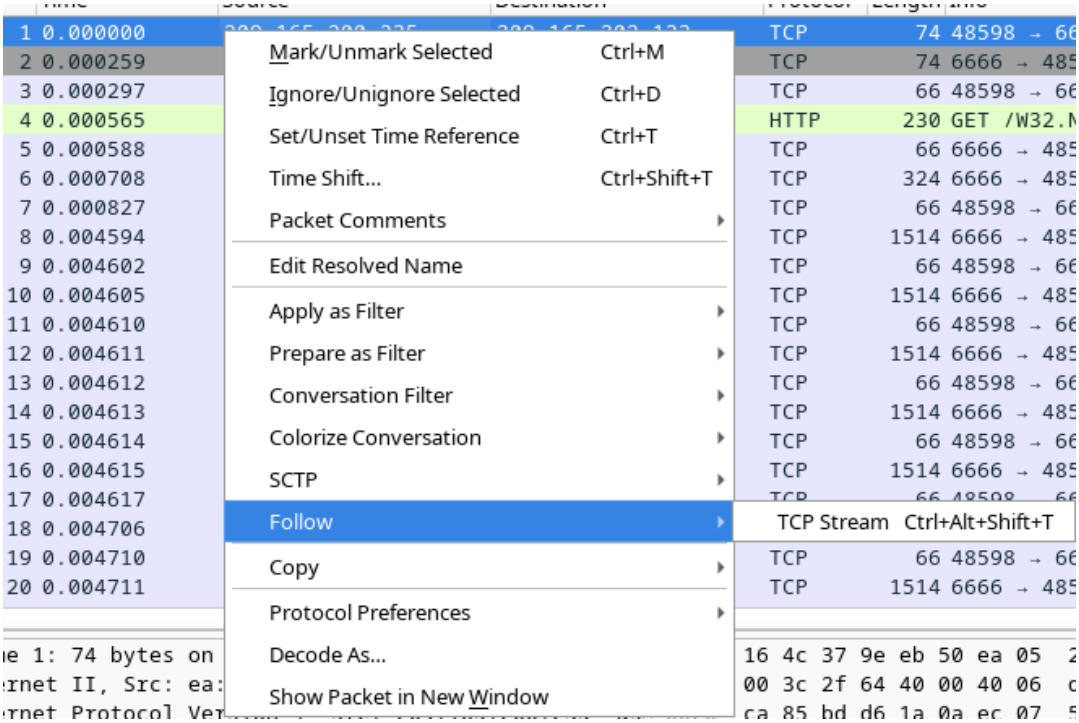
```
[analyst@secOps pcaps]$ wireshark nimda.download.pcap &
```



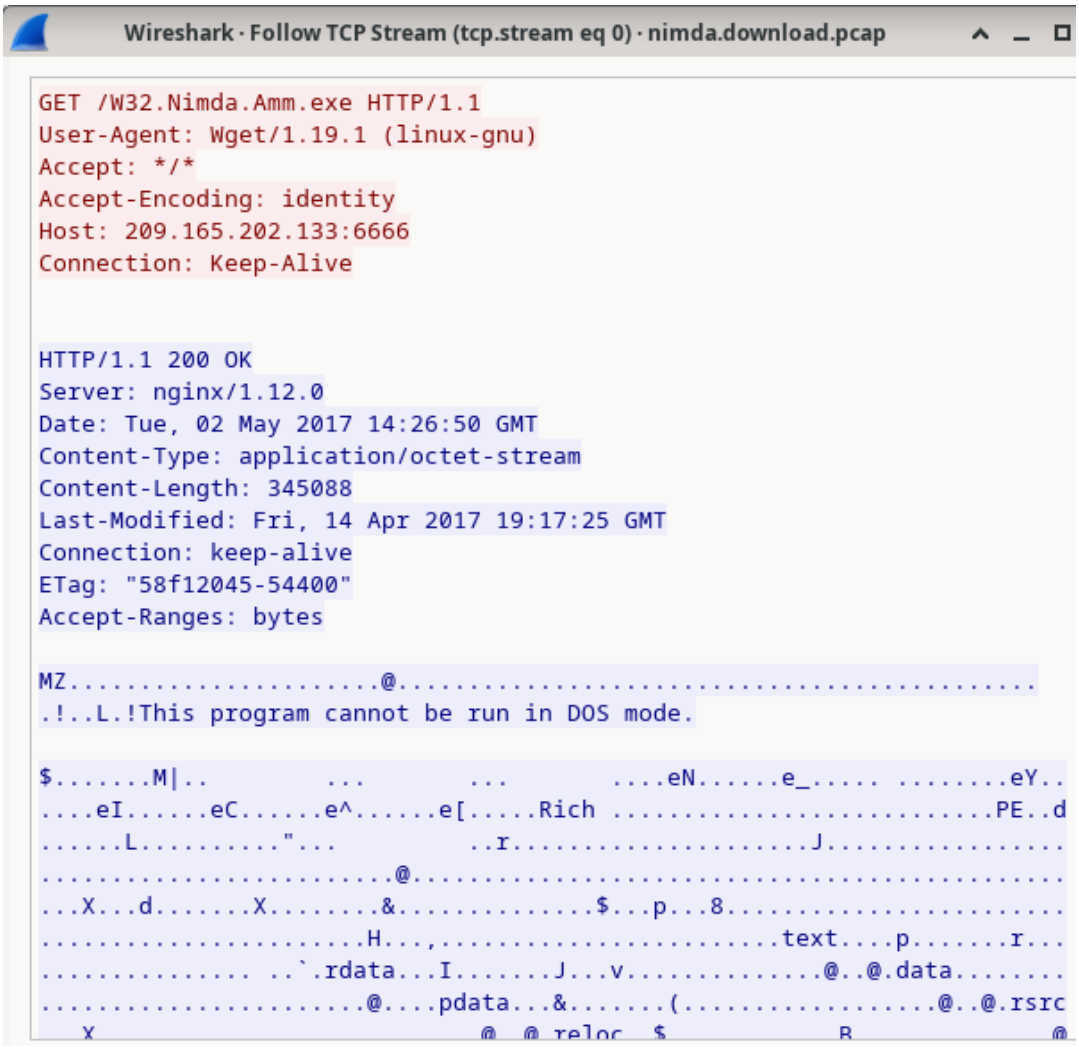
Questa cattura di Wireshark contiene l'inizio della comunicazione, in dettaglio:

- 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.

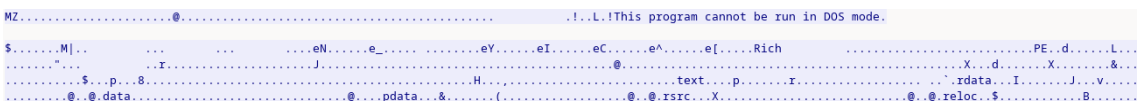
Selezionando il primo pacchetto TCP nella cattura, un pacchetto `SYN`. Fai clic con il pulsante destro del mouse su di esso e scegli `Follow > TCP Stream`.



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



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.



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 programmatori 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-processthreads-l1-1-0.DLL.WINBRAND.dl
```

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
```

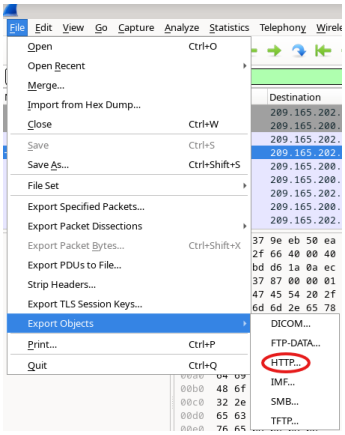
`/W32.Nimda.Amm.exe` è il percorso e il nome esatto del file eseguibile richiesto e scaricato.
Invece scorrendo più giù troviamo questa stringa:

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

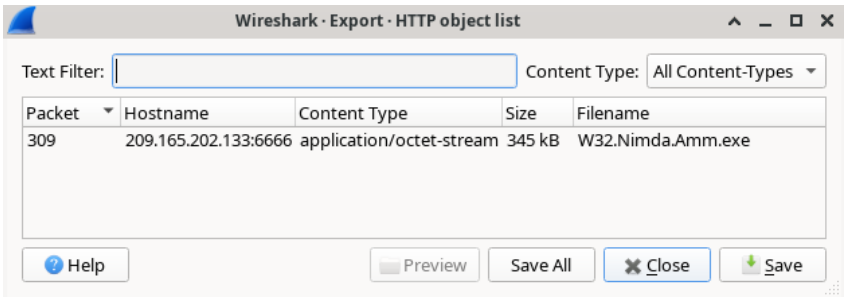
Dove le prime due lettere del payload, ossia MZ, identificano un file eseguibile Portable Executable (PE) di Windows. Mentre “this program cannot be run in DOS mode.” è parte integrante dell'intestazione di un eseguibile Windows, confermando che i dati binari seguenti sono effettivamente il programma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

Svolgimento Parte 2

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



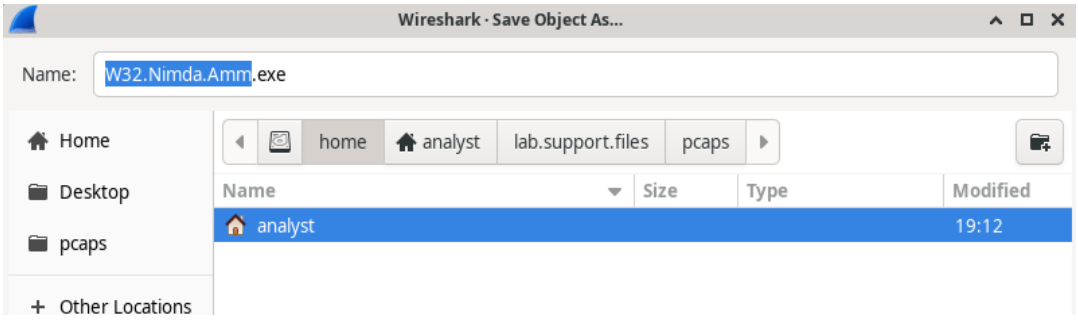
Vedremo questa schermata:

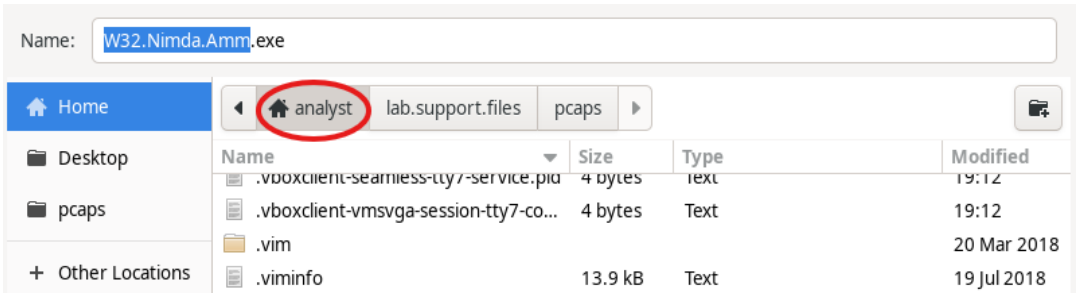


DOMANDA: Perché `W32.Nimda.Amm.exe` è l'unico file nella cattura?
RISPOSTA: `W32.Nimda.Amm.exe` è l'unico file visualizzato nella lista "HTTP object list" di Wireshark perché la lista degli oggetti HTTP non mostra automaticamente tutto il traffico della cattura, ma solo gli oggetti identificati come 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`.

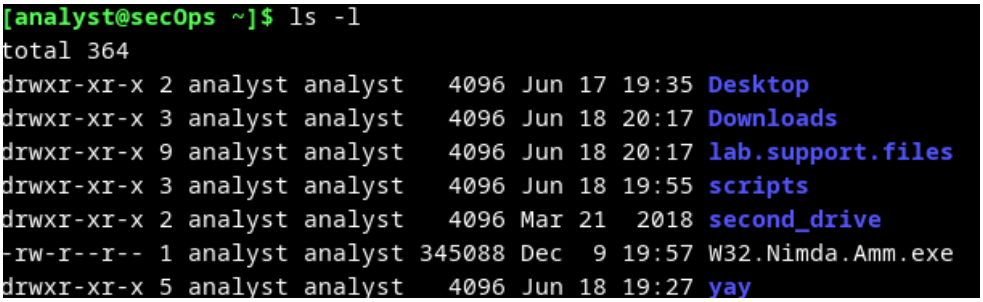
```
Content-Type: application/octet-stream
```

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

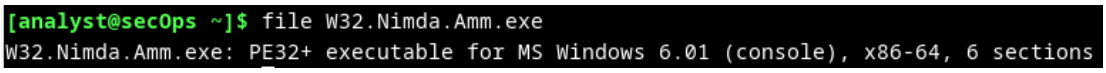




Assicuriamoci di aver fatto tutto nel modo corretto andando a vedere da terminale:



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



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.