Build Week 3

Esercizio 6

LANDA

TRACKER SPA

Guardare i log è molto importante, ma è anche importante capire come avvengono le transazioni di rete a livello di pacchetto. In questo laboratorio, analizzerai il traffico in un file pcap catturato in precedenza ed estrarrai un eseguibile dal file.

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

Cosa sono tutti quei simboli mostrati nella finestra Follow TCP Stream? Sono rumore di connessione? Dati? Spiega.

Non è rumore di connessione.
Quei simboli sono la rappresentazione testuale dei dati binari del file eseguibile (W32.Nimda.Amm.exe) che è stato scaricato.
Un file .exe è composto da codice macchina, che non è testo leggibile.
Quando uno strumento di analisi di rete prova a mostrare questi dati binari come testo, la maggior parte dei byte non corrisponde a caratteri stampabili e viene quindi visualizzata come simboli, caratteri accentati o spazi vuoti.

Ci sono alcune parole leggibili sparse tra i simboli. Perché sono lì?

Le parole leggibili sono **stringhe di testo** che sono intenzionalmente incluse all'interno del file eseguibile. Servono per diverse funzioni del programma, tra cui:

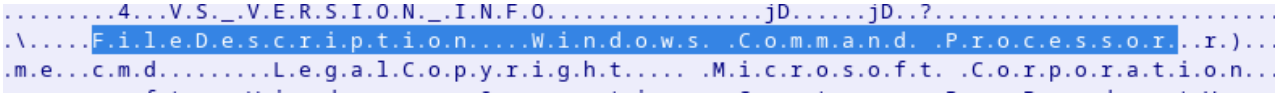
- **Firme di File**: MZ e PE sono "numeri magici" che identificano il file come un eseguibile per Windows.
- **Messaggi di Compatibilità**: "This program cannot be run in DOS mode." è un messaggio standard per i vecchi sistemi operativi.
- **Nomi di Sezioni**: .text, .rdata, .data sono i nomi delle sezioni interne del programma che contengono rispettivamente il codice e i dati.
- **Nomi di Librerie**: KERNEL32.dll, NTDLL.DLL sono librerie di sistema di Windows che il programma deve utilizzare per funzionare.
- **Manifesto dell'Applicazione**: Il blocco di testo XML alla fine del file è un manifesto che fornisce al sistema operativo informazioni sul programma.

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?

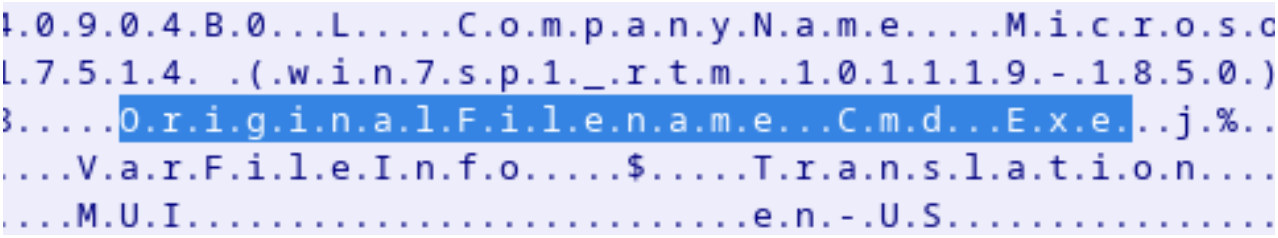
È possibile identificarlo basandosi sulle stringhe di testo leggibili presenti nel flusso, il file eseguibile è cmd.exe, l'interprete dei comandi di Windows (Prompt dei comandi/cmd).

Ecco le prove principali visibili nel dump dei dati:

- **Descrizione del File**: Una delle stringhe più chiare è *F.i.l.e.D.e.s.c.r.i.p.t.i.o.n.....W.i.n.d.o.w.s. .C.o.m.m.a.n.d. .P.r.o.c.e.s.s.o.r.*



- **Nome Originale del File**: Un'altra stringa indica esplicitamente il nome originale: *O.r.i.g.i.n.a.l.F.i.l.e.n.a.m.e...C.m.d...E.x.e.*



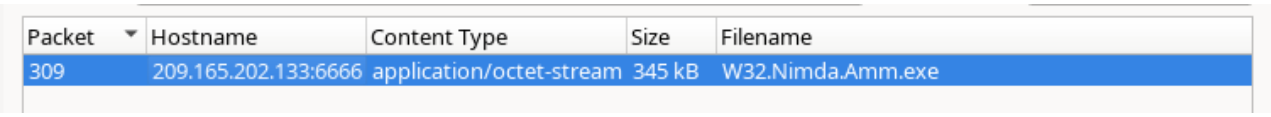
- **Manifesto XML:** Verso la fine, c'è un blocco XML che include *<description>Windows Command Processor</description>* e *name="Microsoft.Windows.FileSystem.CMD"*.

- **Comandi Interni:** Sono visibili numerosi comandi tipici del prompt di Windows, come *C.L.S, D.E.L, D.I.R, C.O.P.Y, E.C.H.O, E.X.I.T*, ecc.

- **File di Debug:** La stringa cmd.pdb si riferisce al file di database del programma, utilizzato per il debugging, che corrisponde al nome dell'eseguibile.

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

Il file W32.Nimda.Amm.exe è l'unico oggetto mostrato perché la cattura dei pacchetti (nimda.download.pcap) ha registrato unicamente la transazione HTTP relativa al download di quel singolo file.
La funzione Esporta Oggetti > HTTP di Wireshark analizza tutto il traffico catturato e isola tutti i file (immagini, eseguibili, documenti, ecc...ecc) che sono stati trasferiti tramite il protocollo HTTP.
In questo caso specifico, l'unica attività registrata è stata una singola richiesta GET per il file W32.Nimda.Amm.exe e la successiva risposta del server che lo ha inviato. Se durante la cattura fossero stati scaricati altri file o immagini tramite HTTP, anch'essi sarebbero apparsi in quella lista.



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

Il file è stato salvato?

Si, il file è stato salvato correttamente come mostrato nello screen.

```
[analyst@secOps pcaps]$ cd /home/analyst/
[analyst@secOps ~]$ ls -l
total 376
-rw-r--r-- 1 root    root      6603 Sep 23 08:14 capture.pcap
drwxr-xr-x 2 analyst analyst   4096 Sep 30 08:59 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    264 Sep 25 07:56 space.txt
-rw-r--r-- 1 analyst analyst 345088 Sep 30 09:52 W32.Nimda.Amm.exe
drwxr-xr-x 5 analyst analyst   4096 Jun 18 19:27 yay
```

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

Un probabile e comune passo successivo per un analista di sicurezza è eseguire un'analisi statica di base per raccogliere più informazioni senza eseguire il file malevolo.
Lo strumento più immediato per questo scopo è il comando strings:

```
[analyst@secOps ~]$ strings W32.Nimda.Amm.exe
```

Questo comando estrae e visualizza tutte le sequenze di caratteri leggibili (testo) presenti all'interno del file binario. Questo permette all'analista di cercare rapidamente indizi importanti, come:

- **Indirizzi IP o nomi di dominio**: A cui il malware potrebbe tentare di connettersi.
- **Nomi di file**: Che potrebbe creare, cercare o modificare sul sistema infetto.
- **Chiavi di registro**: Che potrebbe leggere o scrivere.
- **Messaggi di errore o comandi**: Che possono rivelare la sua funzionalità.

Dopo aver usato strings, i passi successivi potrebbero includere un'analisi statica più avanzata (usando un disassembler come Ghidra o IDA Pro) o un'analisi dinamica (eseguendo il malware in un ambiente controllato e isolato per osservarne il comportamento).