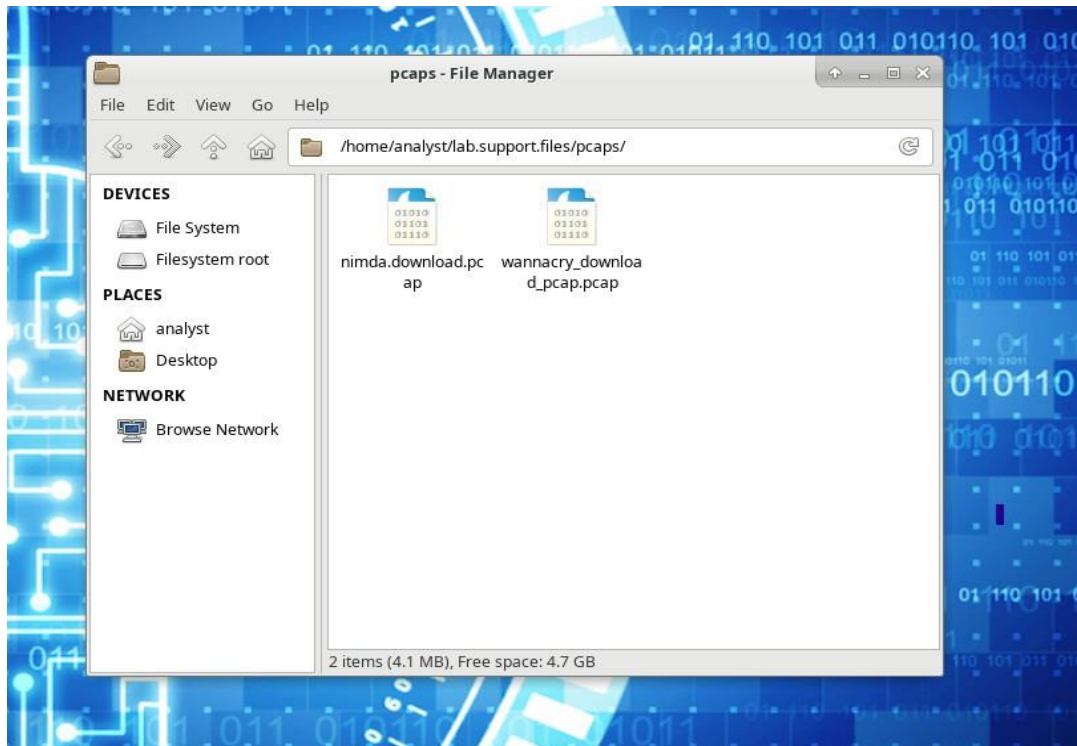


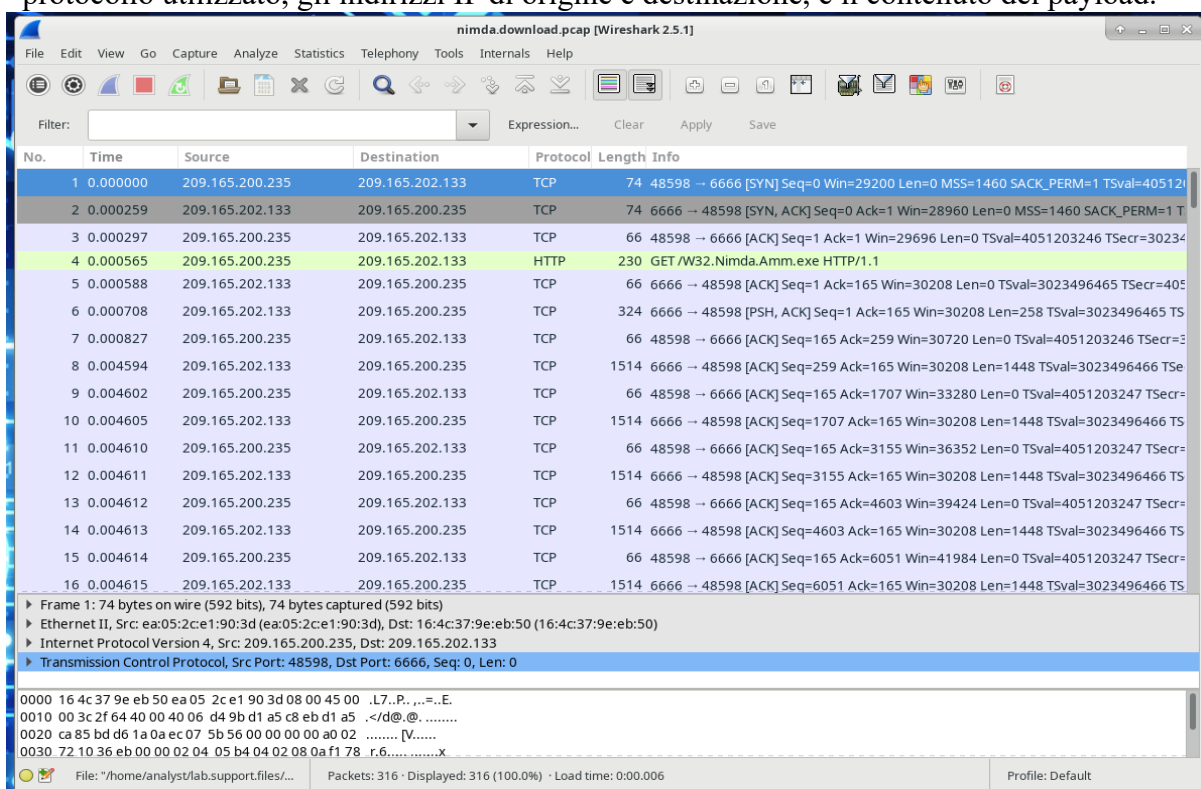
1. Individuazione del file PCAP

- Il file nimda.download.pcap, contenente i dati catturati durante l'attacco, si trova nella directory /home/analyst/lab.support.files/pcaps.
- I file PCAP sono fondamentali per le indagini forensi poiché registrano dettagliatamente ogni pacchetto scambiato in rete, inclusi dati di livello applicativo come richieste HTTP o contenuti binari.



2. Apertura del file PCAP in Wireshark

- Avviare Wireshark e aprire il file nimda.download.pcap.
- Wireshark consente di visualizzare ogni pacchetto trasmesso, mostrando dettagli come il protocollo utilizzato, gli indirizzi IP di origine e destinazione, e il contenuto del payload.



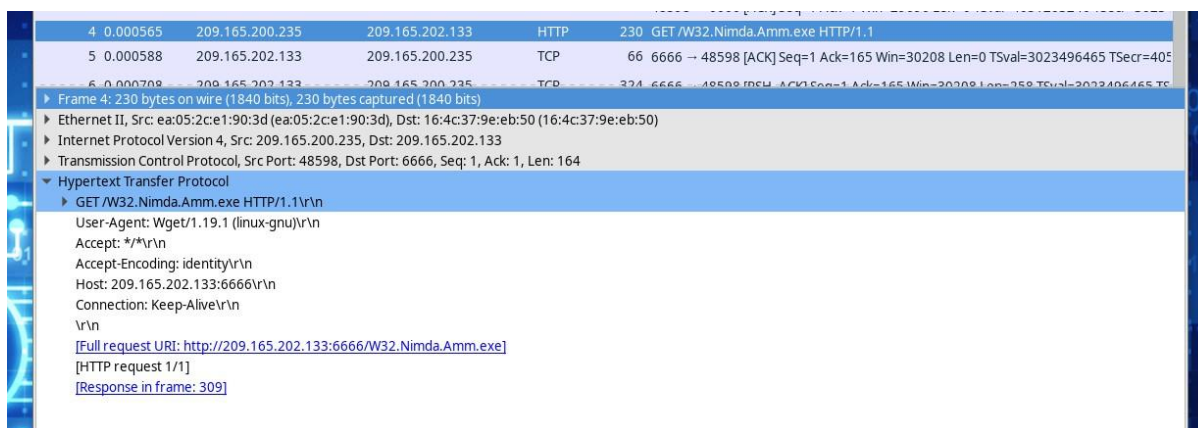
3. Identificazione della stretta di mano a tre vie

- I primi tre pacchetti catturati rappresentano la stretta di mano a tre vie del protocollo TCP, essenziale per stabilire connessioni affidabili:
 - Il client invia un pacchetto SYN (Synchronize) al server.
 - Il server risponde con un pacchetto SYN/ACK (Synchronize-Acknowledge).
 - Il client completa il processo con un pacchetto ACK (Acknowledge).

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. Analisi della richiesta HTTP GET

- Nel quarto pacchetto, si osserva una richiesta HTTP di tipo GET dal client all'indirizzo IP del server, finalizzata al download del file W32.Nimda.Amm.exe.
- Questa richiesta indica che il file è stato scaricato tramite una connessione web non crittografata, utilizzando il protocollo HTTP.



The image shows a Wireshark packet capture. The top pane displays a list of packets. Packet 4 is an HTTP GET request from 209.165.200.235 to 209.165.202.133. The middle pane shows the packet details, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The bottom pane shows the raw packet data in hexadecimal and ASCII.

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=165 Win=30208 Len=0 TSval=3023496465 TSecr=4051203246
6	0.000708	209.165.202.133	209.165.200.235	TCP	324	6666 → 48598 [ACK] Seq=1 Ack=165 Win=30208 Len=0 TSval=3023496465 TSecr=4051203246

Frame 4: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits) on interface eth0

Ethernet II, Src: ea:05:2c:e1:90:3d (ea:05:2c:e1:90:3d), Dst: 16:4c:37:9e:eb:50 (16:4c:37:9e:eb:50)

Internet Protocol Version 4, Src: 209.165.200.235, Dst: 209.165.202.133

Transmission Control Protocol, Src Port: 48598, Dst Port: 6666, Seq: 1, Ack: 1, Len: 164

Hypertext Transfer Protocol

GET /W32.Nimda.Amm.exe HTTP/1.1\r\n

User-Agent: Wget/1.19.1 (linux-gnu)\r\n

Accept: */*\r\n

Accept-Encoding: identity\r\n

Host: 209.165.202.133:6666\r\n

Connection: Keep-Alive\r\n

\r\n

[Full request URI: http://209.165.202.133:6666/W32.Nimda.Amm.exe]

[HTTP request 1/1]

[Response in frame: 309]

5. Esame del flusso TCP

- Utilizzare la funzione "Follow TCP Stream" di Wireshark per visualizzare l'intera comunicazione tra client e server come un flusso continuo.
- Nel flusso, notiamo:
 - Caratteri binari:** rappresentano il contenuto del file eseguibile scaricato.
 - Stringhe leggibili:** frammenti di testo che possono fornire indizi sul comportamento del file.

Follow TCP Stream (tcp.stream eq 0)

Stream Content:

```

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.....f.....rdata...I.....j...v.....@...@.dat
a.....@....pdata...&.....(.....@....rsrc...X.....@....@.reloc.
$.B.....@..B7..L@....LK.....LK.....LU.....LK.....Lb.....L.....msvcrt.dll.NTDLL.DLL.KERNEL32.dll.api-
ms-win-core-processthreads-
11-1-0.DLL.WINBRAND.dll.....H;
...$Q..H...f...Q.....%.....H...teSH..H.....H..tO....LA.H...t>H.L$0!;.....H;H.C.....H.....
7...L..3.H...1....
...H..[.....%.....H..$..H.t$.WH.....H.....H...%...=.....$...>...$...=.....t$D.F..H.L$ 3.....H.T$
E3.H...P1...Uf;.....;3.....;.....<.....;.....Hc.H.
.b..H...H...H.....H..t
H9X...Z...=*...t.H..t.H.T$A..H...0.....L..$.I.[I.S.I.....H..
(.....H...j.....H.....H...G...t...y.....<...!..3.....H..a..H.....t1.
N.....<.....H..a...H..t.H.
H..(.....H..(H.....H.....3...H..(.....>...;.....H.Fp....F.;.....H.Fp....*...D.X.3.H...H..-pf..H..H.....C..H...Y....H..

```

Entire conversation (345510 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Filter Out This Stream Close

File: "/home/analyst/lab.support.files/... Packets: 316 · Displayed: 316 (100.0%) · Load time: 0:00.006

```

sh....wcsstr...j.iswalph...wcstoul..._errno....printf....rand...o..._job..
3.fprintf...wcschr...realloc...tolower....setlocale..._wcsupr.a.iswdigit.y._wcsicmp...f.iswspace....wcschr....memmov
e.*.fgets..._pclose.&.ferror..
%.feof...._wpopen...._wcsncmp.X._vsnwprintf...wcstol...D._get_osfhandle..O._getch....towupper....wcssp...._tell.s.lo
ngjmp..._local_unwind.
{.RtlCaptureContext...RtlLookupFunctionEntry....RtlVirtualUnwind..K.RtlFreeHeap.*.NtFsControlFile..I.NtOpenThrea
dToken...NtClose.d.NtOpenProcessToken.....NtQueryInformationToken...RtlDosPathNameToNtPathName_U..
9.RtlFindLeastSignificantBit....NtSetInformationProcess...NtQueryInformationProcess...RtlNtStatusToDosError...G
etTimeFormatW....GetTickCount....QueryPerformanceCounter...SetUnhandledExceptionFilter...Sleep...DelayLoadF
ailureHook...?.LoadLibraryExA...g.FreeLibrary...CreateHardLinkW...CreateSymbolicLinkW...GetVolumePathNameW...
.GetThreadLocale...ResumeThread....SetProcessAffinityMask..
0.GetNumaNodeProcessorMaskEx....GetThreadGroupAffinity..
9.FindFirstFileExW...GetDiskFreeSpaceExW.K.FindNextStreamW.@.FindFirstStreamW...DeviceIoControl...Compare
FileTime...RemoveDirectoryW...GetCurrentDirectoryW...GetExitCodeProcess....WaitForSingleObject...TerminateP
rocess..X.SetCurrentDirectoryW..t.SetFileTime...DeleteFileW.^..SetEndOfFile...k.SetFileAttributesW...u.CopyFileW...C
reateDirectoryW..Q.SetConsoleTextAttribute.
+.FillConsoleOutputAttribute...&.ScrollConsoleScreenBufferW..m.GetACP..c.FormatMessageW..
\..FlushFileBuffers....DuplicateHandle...HeapSize....HeapReAlloc...VirtualAlloc....VirtualFree...HeapSetInformation...
.GetCurrentThreadId....OpenThread....GetFileAttributesExW....GetDriveTypeW...GetVersion...;.LeaveCriticalSection
....EnterCriticalSection...GetModuleFileNameW...GetWindowsDirectoryW..
8.SetConsoleCtrlHandler...InitializeCriticalSection...ExpandEnvironmentStringsW.D.CancelSynchronousIo...GetVolu
meInformationW...GlobalFree....GlobalAlloc.q.SetFilePointerEx...1.WriteFile.
(.SearchPathW.J.LocalFree.S.SetConsoleTitleW..a.MoveFileExW.d.MoveFileW...QueryFullProcessImageNameW....Rea
dProcessMemory.A.LoadLibraryW....RegSetValueExW....RegCreateKeyExW...UnhandledExceptionFilter....GetCurr
entProcess...~.GetSystemTimeAsFileTime...VirtualQuery..
\..CmdBatNotification...w.GetCPIInfo...GetConsoleOutputCP...SetThreadLocale.I.GetProcAddress....GetModuleHandl
eW..R.CloseHandle...GetLastError...p.SetFilePointer....GetFullPathNameW...>.FindFirstFileW..J.FindNextFileW.
3.FindClose...CreateFileW...ReadFile...h.MultiByteToWideChar...GetFileSize...WideCharToMultiByte.U.IstrcmpiW.R.Ist
rcmpW.i.GetStdHandle..
[.FlushConsoleInputBuffer...HeapAlloc.N.GetProcessHeap....HeapFree....GetConsoleScreenBufferInfo....ReadCon
soleW...<.SetConsoleCursorPosition...-.FillConsoleOutputCharacterW.
0.WriteConsoleW...GetFileType...GetUserDefaultLCID....GetLocaleInfoW....SetLocalTime...|.GetSystemTime...Syste
mTimeToFileTime...).FileTimeToLocalFileTime.*.FileTimeToSystemTime....GetDateFormatW....RegDeleteValueW...Ge
tLocalTime....GetConsoleMode..H.SetConsoleMode....GetEnvironmentVariableW...GetCommandLineW..-GetNuma
HighestNodeNumber....GetEnvironmentStringsW..f.FreeEnvironmentStringsW.b.SetEnvironmentVariableW'.SetE

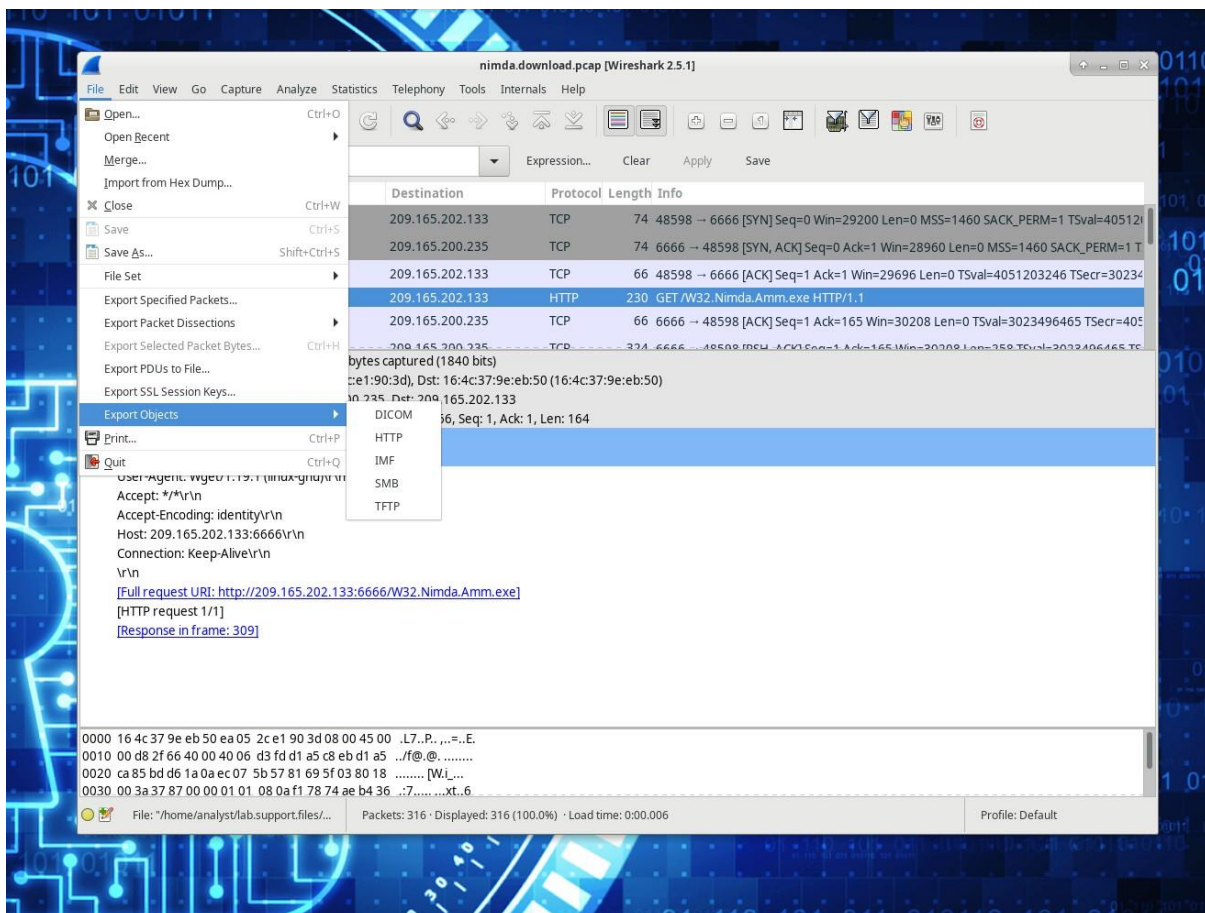
```

Entire conversation (345510 bytes)

6. Identificazione del file scaricato

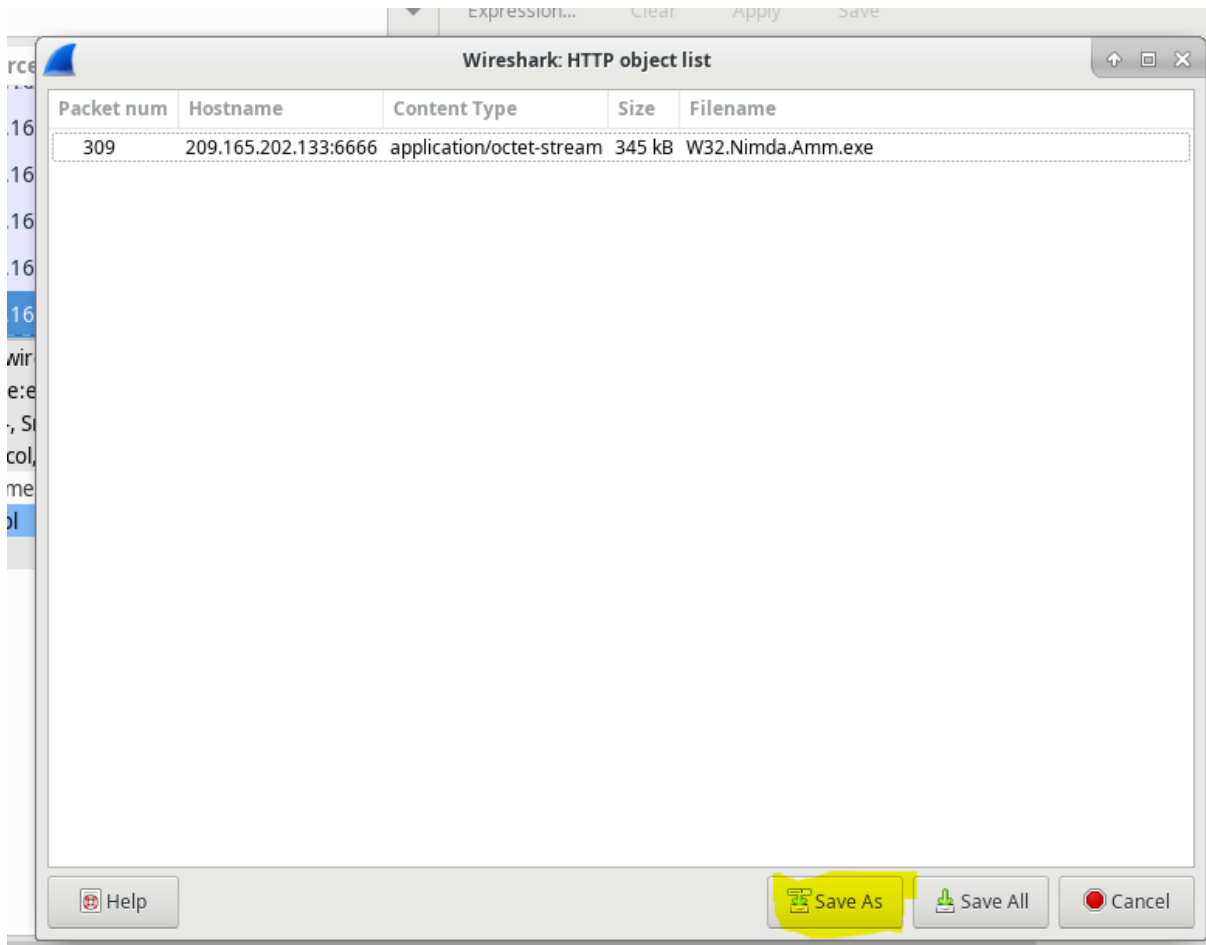
- Analizzando le stringhe presenti nel flusso TCP, si scopre che il file W32.Nimda.Amm.exe è in realtà una copia del file di sistema cmd.exe di Windows.
- Questo è preoccupante, poiché cmd.exe può essere utilizzato per eseguire comandi dannosi o script malevoli.

```
.....00.....h.....(.....00.....h.....00.....%.....h.....  
.....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.....C.o.m.p.a.n.y.N.a.m.e.....M.i.c.r.o.s.o.f.t..C.o.r.p.o.r.a.t.i.o.n...  
\\.....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..r)...F.i.l.e.V.e.r.s.i.o.n.....  
6...1...7.6.0.1...1.7.5.1.4..(w.in.7.s.p.1._r.t.m...1.0.1.1.1.9.-1.8.5.0.)....  
(.....I.n.t.e.r.n.a.l.N.a.m.e...c.m.d.....L.e.g.a.l.C.o.p.y.r.i.g.h.t.....M.i.c.r.o.s.o.f.t..C.o.r.p.o.r.a.t.i.o.n...A.l.l.r.i.g.h  
t.s.r.e.s.e.r.v.e.d.....8.....O.r.i.g.i.n.a.l.F.i.l.e.n.a.m.e...C.m.d..E.x.e...j  
%...P.r.o.d.u.c.t.N.a.m.e.....M.i.c.r.o.s.o.f.t...W.i.n.d.o.w.s...O.p.e.r.a.t.i.n.g..S.y.s.t.e.m.....B.....P.r.o.d.u.c.t.V.e.r.s.i  
o.n...6...1...7.6.0.1...1.7.5.1.4....D....V.a.r.F.i.l.e.I.n.f.o.....$.....T.r.a.n.s.l.a.t.i.o.n.....J..  
7.....0...@.../...!  
8...d.....M.U.I.....M.U.I.....e.n.-U.S.....  
.....  
.....0.....(.0.8.@.H.P.X.h.x.....p.....  
(.@.H.`h.....(.@.H.`h.....(.@.H.`h.....(.@.H.`h.....  
(.@.H.`h.....H.h.....  
(.@.H.`h.....
```



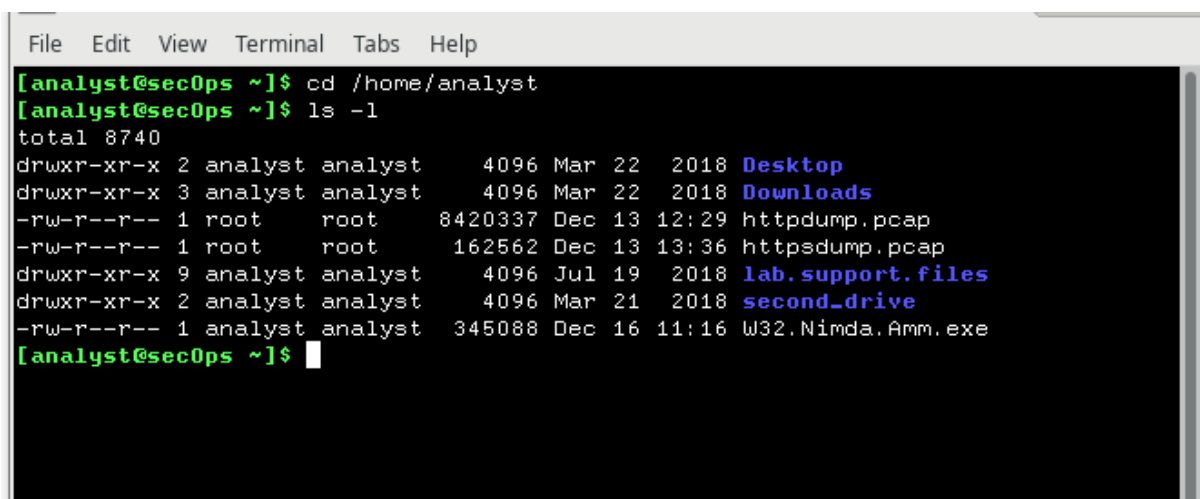
7. Esportazione dell'oggetto HTTP

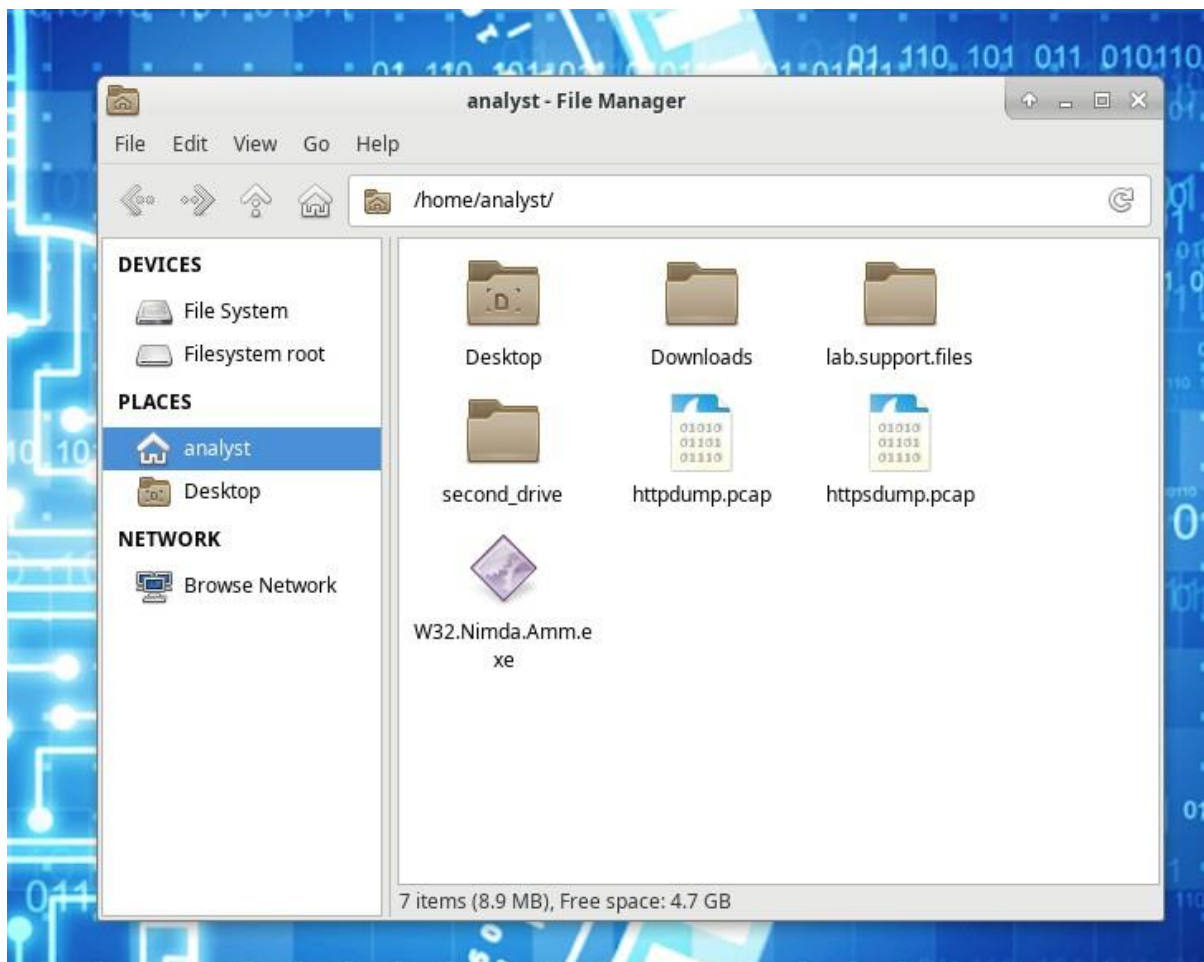
- In Wireshark, cliccare: File > Export Objects > HTTP.
- Nella finestra che appare, seleziona il file W32.Nimda.Amm.exe e clicca su Save As per salvarlo nella directory desiderata.



8. Verifica del file recuperato

- Dopo aver esportato il file, aprire un terminale e navigare fino alla directory in cui è stato salvato.
- Utilizzare il comando `ls -l` per verificare la presenza del file.





9. Analisi del file recuperato

- Eseguire il comando `file W32.Nimda.Amm.exe` per determinare il tipo di file.
- Il risultato confermerà che si tratta di un eseguibile per Windows.

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

Conclusioni

L'analisi dei file PCAP con Wireshark è uno strumento fondamentale per individuare attività sospette in rete. In questo laboratorio, abbiamo ricostruito il download di un file eseguibile malevolo, dimostrando come un attaccante possa sfruttare protocolli comuni come HTTP per distribuire malware.

Abbiamo visto come:

- ✓ Identificare un file sospetto all'interno di una cattura di rete.
- ✓ Esaminare le comunicazioni TCP per individuare richieste di download.
- ✓ Estrarre e analizzare un file eseguibile per capire la sua natura.

Questa procedura è essenziale per gli analisti di sicurezza, che devono saper riconoscere e bloccare minacce prima che possano compromettere un sistema. Monitorare il traffico di rete e analizzare i file sospetti aiuta a proteggere infrastrutture aziendali e dati sensibili da potenziali attacchi informatici.