# Lab 9

## What Is Network Forensics?

Network forensics is a specialized field within cybersecurity focused on the monitoring, capturing, and analysis of network traffic to uncover and investigate security incidents or breaches.

By examining data packets, network logs, and communication patterns, network forensics aims to reconstruct events leading up to an incident, and understand their methods. This process is crucial for not only resolving current security issues but also for strengthening defenses against future attacks.

With cyber threats becoming increasingly sophisticated, network forensics plays a vital role in safeguarding digital infrastructures and ensuring overall network security.

To identify attacks, investigators must have a good understanding of how different parts of a network communicate, such as websites, emails, general network communications, and file transfers.

Serious cyberattacks, like ransomware or attacks on supply chains, usually start with someone getting into the target system without permission. After that, they move around inside the network, this happens through multiple network devices like routers, firewalls, and switches.

## What Is Wireshark?

Wireshark is a powerful tool for examining network traffic, commonly used in digital investigations. By installing Wireshark on a portable drive, Investigators can perform real-time forensic analysis, which helps in responding to incidents and focusing on important tasks first.

This tool enables investigators to quickly understand the current situation, stop the attack, and collect evidence and information to avoid similar incidents in the future.

Wireshark is a free and open-source network protocol and traffic analyzer that enables users to capture and troubleshoot network traffic.

Essentially, Wireshark allows you to capture traffic on a network and presents the captured traffic as individual packets for detailed analysis. It captures packets on a network, displaying various packet fields and headers based on the type of packet selected.

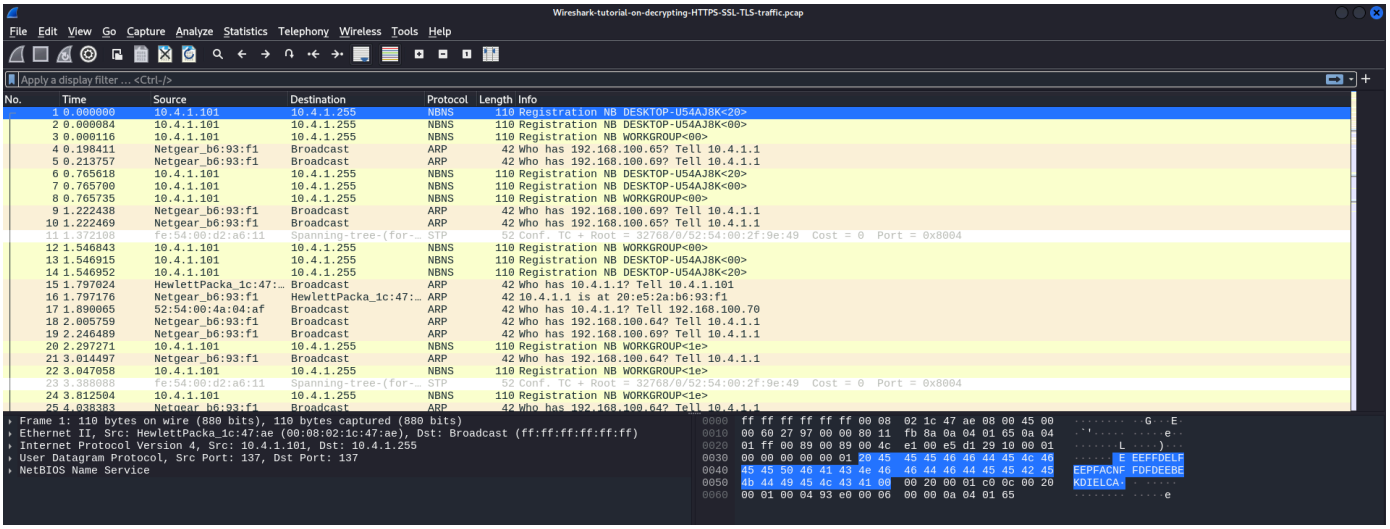
When capturing traffic, you do so through an interface, which Wireshark calls a network interface card (NIC). This could be a wired or a wireless connection. The amount of data you capture depends on your interface.

For example, if you’re using a wireless adapter that doesn’t support monitor mode (which allows you to capture traffic from other devices on a wireless network), you won’t be able to capture that traffic. So,

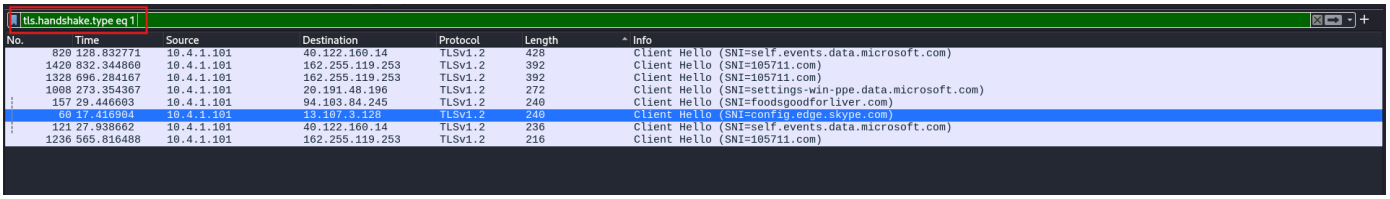
the type of device you use affects your ability to capture different types of network traffic.

## Practical Show

Let's work on an example exercise. We need to examine harmful network traffic, and the pcap file we use will have data that's either sent over HTTPS or HTTP but protected by a TLS certificate. Our job is to decrypt this data and figure out what kind of harmful software was used to attack a computer on the network.



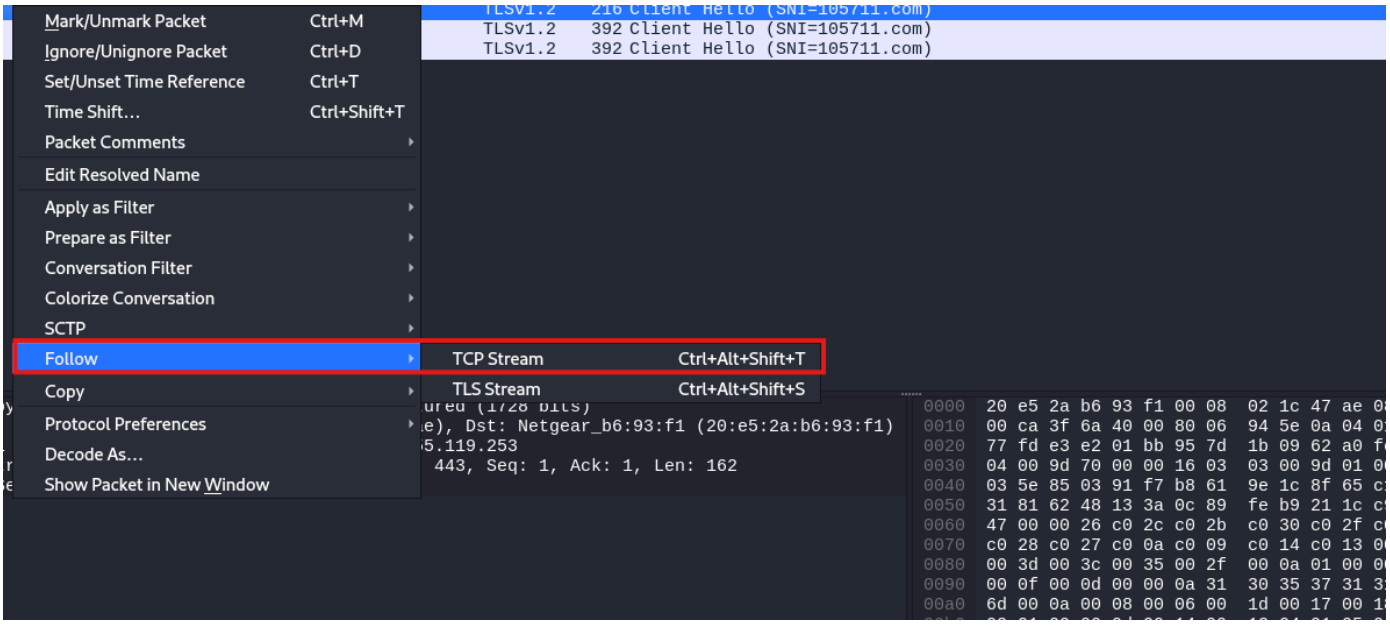

As a digital forensics' investigator, you may need to examine a pcap file to find out if a device was infected and what happened. In these situations, you can use the helpful filtering tool in Wireshark. For example, you can filter for successful TLS handshakes by using tls.handshake.type == 1. This example shows the use of TLS protocol version 1.2.




You can tell that this traffic is encrypted because it has an SSL certificate. By following the TCP stream.

To follow a TCP request in Wireshark:

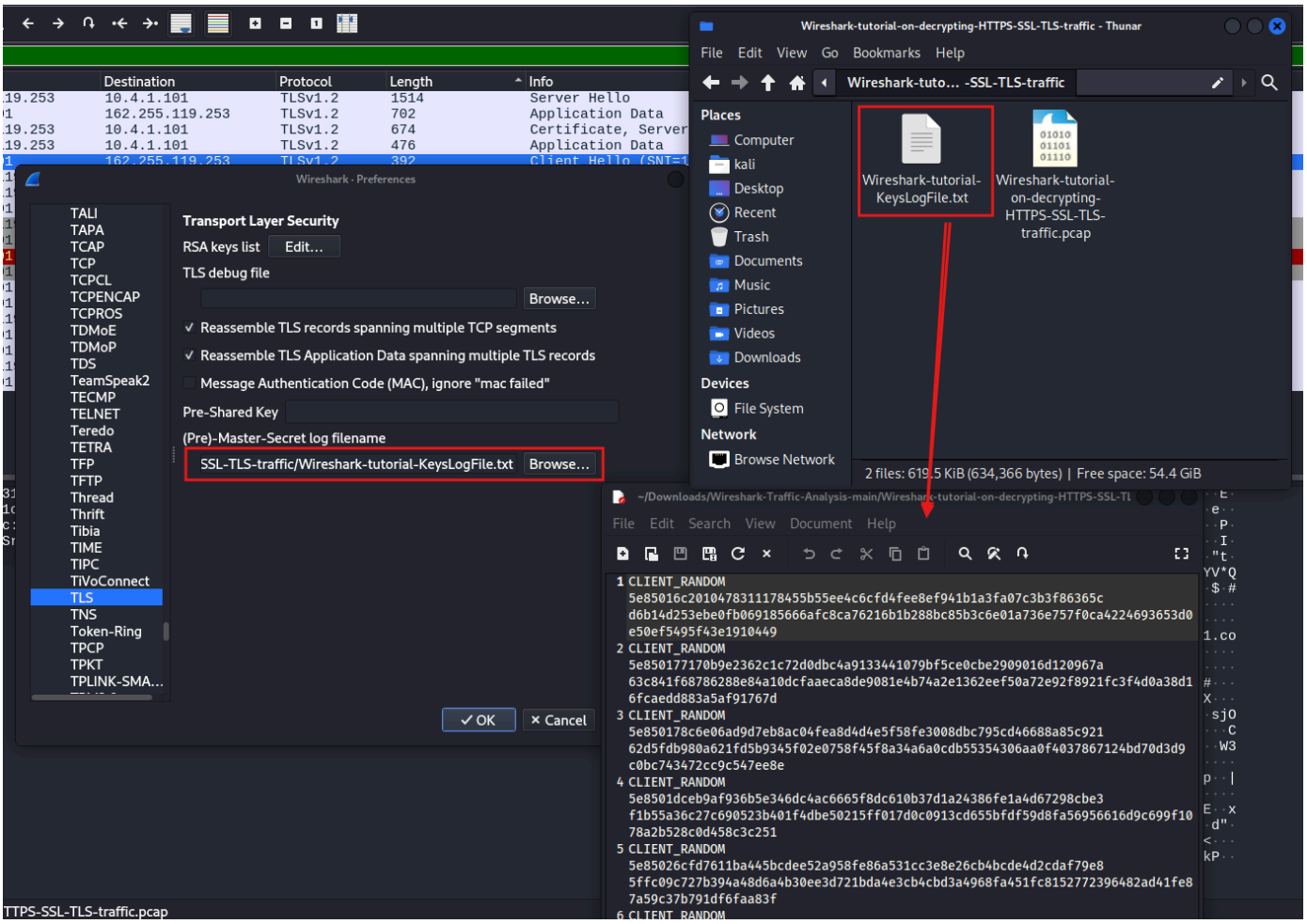
```
Right-click the packet, then select “Follow” > “TCP Stream” from the context menu.
A new window will open showing the entire conversation between the client and server for that TCP connection.
```




You'll notice that all the data remains encrypted.


It gives you the real SSL keys needed to decrypt HTTPS or SSL encrypted data. Now that we have these keys, how do we use them to unlock the data?

To do this, click on Edit, then choose Preferences. Look for Protocols and find TLS. You'll find a spot for the Pre-Master Secret log file name. Go to where you saved the file, pick the Wireshark tutorial keys log file, and press Open



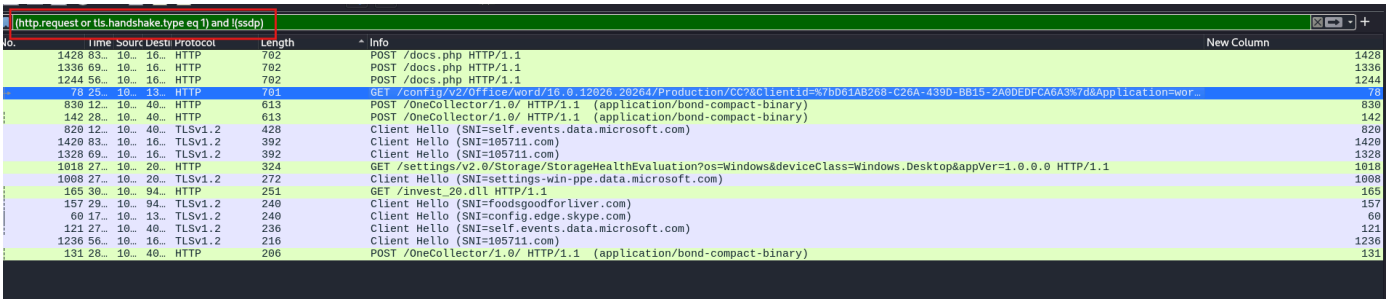

When we use a filter to look at the TLS handshake by typing “tls.handshake.type == 1” and press enter, we can then follow the TCP stream to see the communication. By examining the TLS stream, we can see the actual POST requests being made, which lets us analyze the data that is being sent


Now, our goal is to determine which system was affected and what type of malware led to the infection.

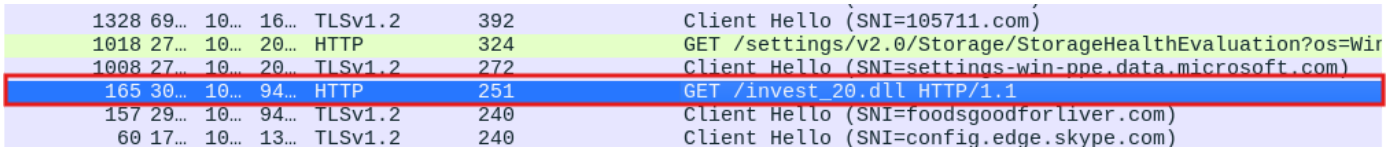
We'll use a filter in Wireshark. First, we'll set the filter to http.request, and then we'll use logical operators.

We'll set the filter to http.request or tls.handshake.type equal to 1 (which is the filter we used previously), and then we'll exclude the SSDP protocol.

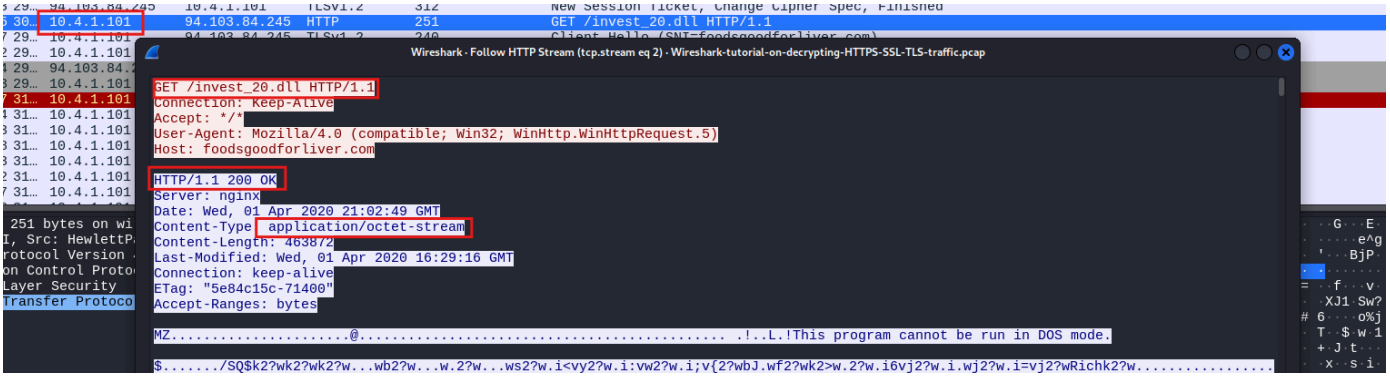
So, our full filter will look like this: (http.request or tls.handshake.type == 1) and !(ssdp). Once we've set this up correctly, we'll press enter to show all the HTTP requests or secure communications, but we won't show any SSDP messages.




We have a GET request, and by clicking on it, we see that it's requesting or downloading a specific resource called invest20.ell




By right-clicking on the packet and selecting Follow HTTP Stream, we can observe what happened. We see that the source IP on the network made a GET request to a specific server for the DLL. The server responded with an OK, indicating that the file was found, and then provided the octet stream containing the DLL.

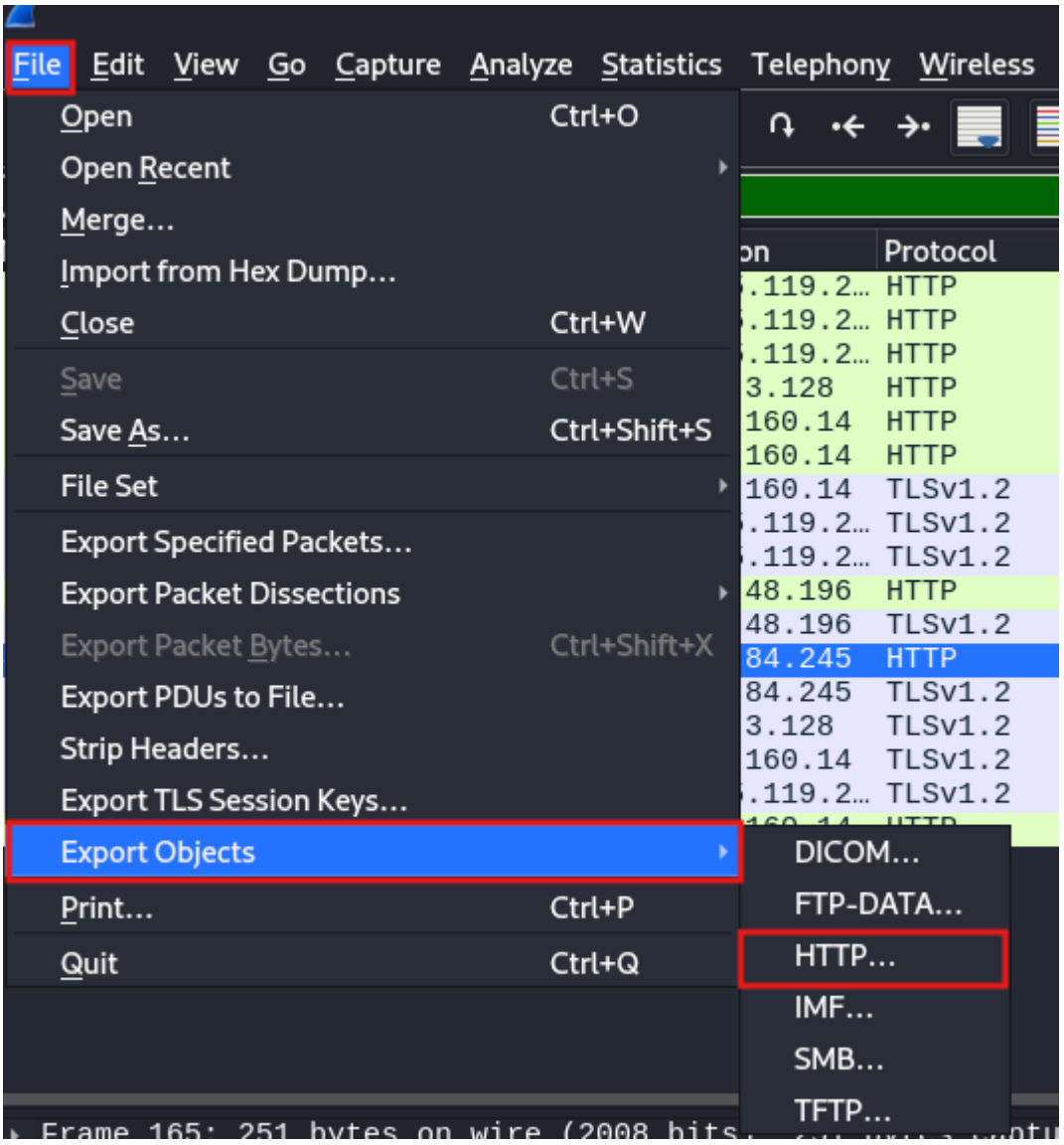



The application data contained within that packet appears to be the actual DLL. We can confirm this because we see the DOS stub, which includes the message “This program cannot be run in DOS mode.” This indicates that the content of the packet is indeed the DLL file.

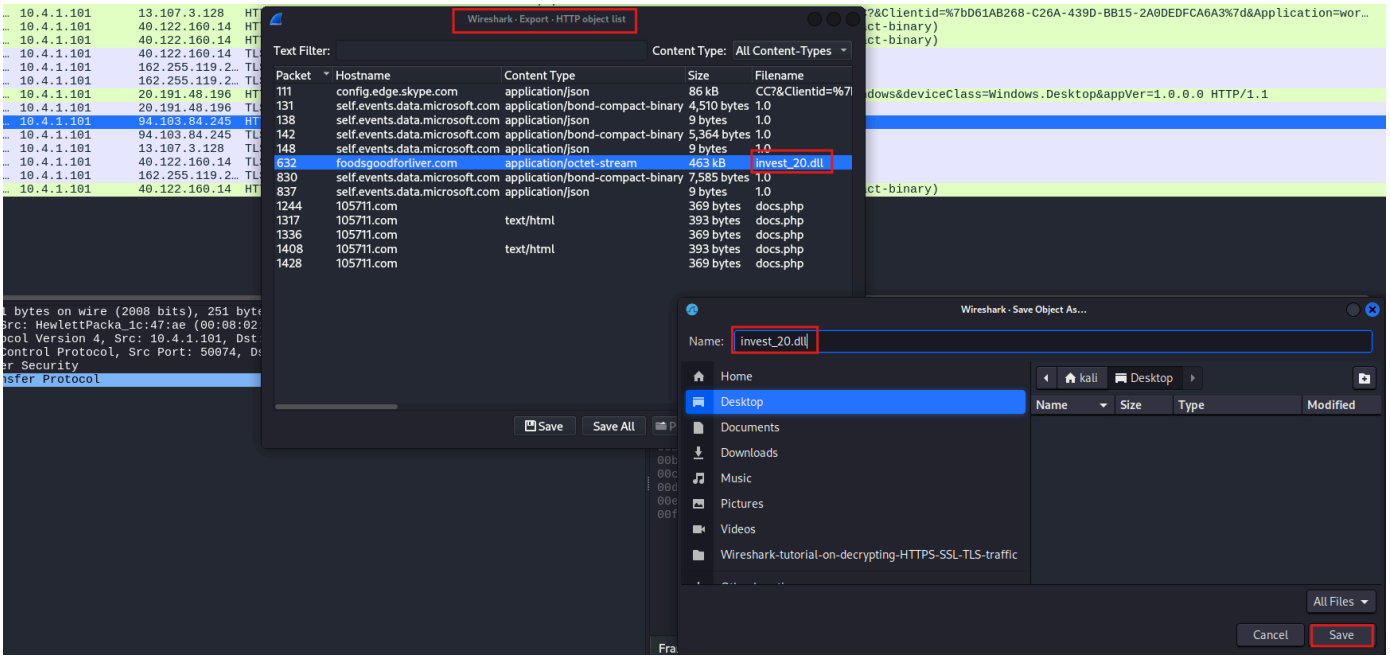

As a malware analyst or threat hunter, you would need to download or save the content of this DLL file to analyze it with a tool like VirusTotal. you can save the DLL as a file because it’s no longer encrypted. Follow these steps:

1- Find the correct data packet in Wireshark.

2- Click on that packet and choose the Export Objects to save it as an HTTP file

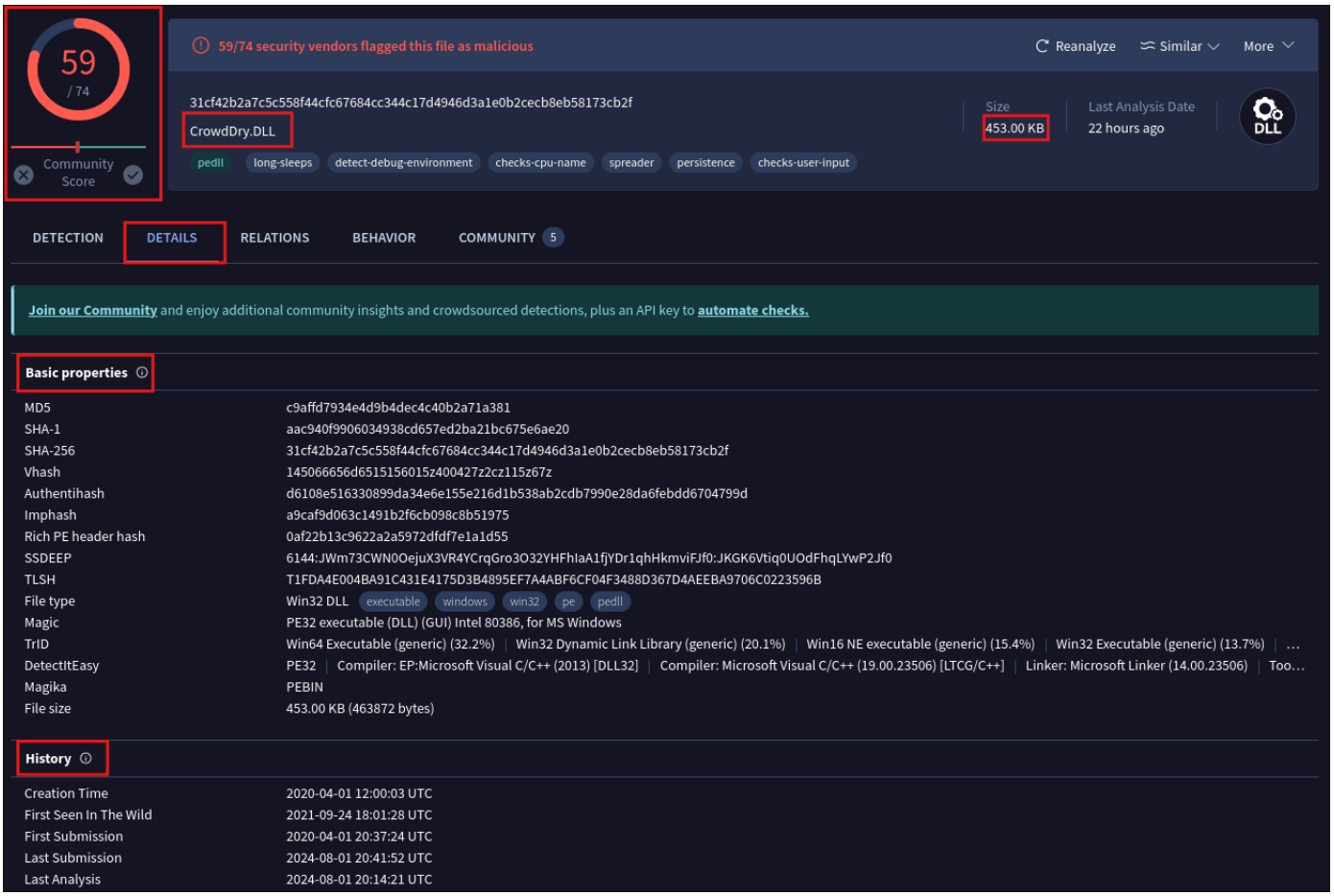



We are looking for the file named invest_20.dll. To save this file, choose to export it and save it on your desktop as invest20.dll. This will allow you to save the downloaded file for further analysis.



Once the file is saved, you can use a service like VirusTotal to identify the type of malware. Simply upload the malware file to VirusTotal for analysis.

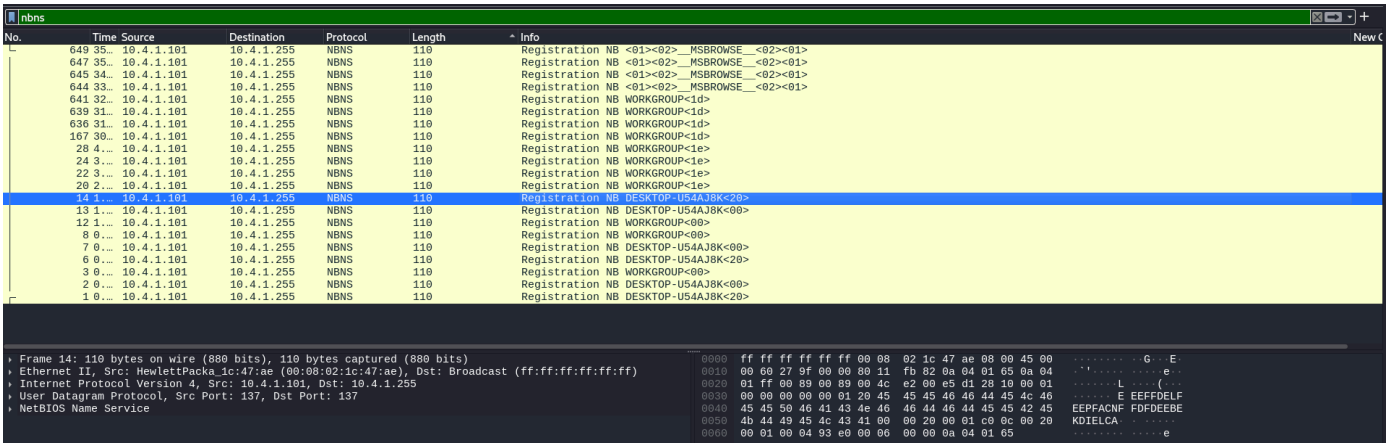

You'll see that this is flagged as malware, with the original DLL being crowddrive.dll. You can learn more details about it here, including the fact that it's a Win32 DLL Portable Executable. Additionally, you can find information about when this specific piece of malware was created.




You can see that the execution parent is an Office Open XML document, typically with filenames like investments.doc. By clicking on it, you can explore the relationships. The number of these relationships gives you an idea of the infection type and its cause. This method helps you understand the infection vector and how the malware was introduced.


In this case, it seems that someone on the network received the file through email and downloaded it. They opened the Excel file using Excel and ran the macro inside it. This macro caused more code to run, which downloaded a file called a DLL. This DLL file then caused the computer to become infected

By filtering for NBNS and hitting enter, you can see that the host's name is DESKTOP-U54AJ8K.




# Lecture: "Those Who Don’t Learn from the Past Are Doomed to Repeat It"

**Presented by Phil Merlion**

---

## Introduction: Why History Matters in Cybersecurity

*"We study history not to dwell on the past, but to avoid making the same mistakes twice."*

Good [morning/afternoon/evening], everyone. Over the next hour, we’re going to dive into a series of forensic case studies—real-world breaches, attacks, and vulnerabilities that have shaped cybersecurity as we know it today.

This lecture isn’t just about *what* happened—it’s about *why* it happened, how it could have been prevented, and what we can learn to protect our networks moving forward.

We’ll start with a chilling example fresh in the headlines:

---

## Case Study 0: The Spanish Botnet Takedown (Breaking News)

### The Attack

Just this morning, authorities in Spain dismantled one of the **largest botnets ever discovered**:

- **39,000+ command-and-control servers**—imagine an army of infected devices waiting for orders.
- **Business Model**: Cybercriminals *rented* this botnet for **$2,500 USD every few days**.
- **Target**: Online gaming networks.

### The Damage

The attackers launched:

- **HTTP GET floods** (overwhelming servers with requests).
- **TCP SYN floods** (exploiting connection handshakes).

- **UDP port scans** (probing for weak spots).

Their goal? **Extortion**. They blackmailed gamers and companies, threatening to crash servers unless they paid up.

### Why This Matters

This wasn't a sophisticated attack. It used **known vulnerabilities**—the kind we'll discuss today—that could have been mitigated with proper monitoring.

---

## Today's Agenda: Learning from the Past

We'll cover six critical areas where history keeps repeating itself:

1. **UPnP: The Unseen Backdoor**
   - How a protocol designed for convenience became a hacker's best friend.
2. **IoT: Bring Your Own Destruction**
   - Smart devices = dumb security. From talking dolls to hacked light bulbs.
3. **Ransomware: Digital Kidnapping**
   - Why hospitals, pipelines, and even your grandma's PC are targets.
4. **Botnets: The Zombie Armies**
   - From Mirai to modern variants—how your fridge could join the attack.
5. **Man-in-the-Middle: The Invisible Eavesdropper**
   - How a $50 gadget can spy on your boardroom meetings.
6. **Application Attacks: Phishing, Lies, and Videotape**
   - The oldest tricks in the book (and why they still work).

---

## Forensics 101: The Art of Digital Autopsies

### Troubleshooting vs. Forensic Analysis

| Troubleshooting | Forensic Analysis |
|---|---|
| *"Why is the network slow?"* | *"Who broke in, and how?"* |
| Fixes immediate problems. | Uncovers hidden threats. |
| Uses Wireshark to diagnose. | Uses Wireshark as *evidence*. |

### The Four Key Forensic Questions

1. **Damage Assessment**

- Is data stolen? Are systems compromised?
- Example: A bank detecting unauthorized transfers to Hong Kong.

2. **Intruder Identification**
   - Was it a phishing email? A brute-force attack? An insider?
   - Clue: DNS logs often reveal hacker infrastructure.
3. **What Was Left Behind?**
   - Malware? Backdoors? New admin accounts?
   - Example: Hackers creating "ghost" accounts for future access.
4. **Is the Evidence Court-Ready?**
   - Can you prove the attack in a legal setting?
   - Example: Packet captures (pcaps) used in FBI cases.

---

## Case Study 1: UPnP – The Unseen Backdoor

### What Is UPnP?

Universal Plug and Play (UPnP) lets devices **auto-discover** each other (e.g., your printer talking to your laptop).

### The Fatal Flaw

- Runs on **UDP port 1900** (unencrypted).
- Devices *broadcast* their presence via multicast HTTP:

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
LOCATION: http://192.168.1.1/device.xml
```

- Attackers exploit this to:
  - Map internal networks.
  - Redirect traffic (e.g., to malicious servers).

### A Real-World Capture

In Wireshark, filter for `udp.port == 1900`. You'll likely see:

- `NOTIFY` messages ("Here I am!").
- `SEARCH` messages ("Who's out there?").
- **IPv6 versions too**—this isn't going away.

### The Fix

- **Disable UPnP** if unused.

- **Monitor port 1900** for suspicious traffic.

---

## Case Study 2: IoT – The Internet of Threats

### The Rise of the Machines (That Spy on You)

From **"smart" barbies** to **Wi-Fi-enabled light bulbs**, IoT devices are everywhere—and they're *dangerously insecure*.

### The Mirai Botnet (2016)

- **How It Worked**:
  - Scanned for IoT devices with **default passwords** (e.g., `admin:admin`).
  - Infected **14 million devices** (cameras, routers, DVRs).
- **The Attack**:
  - Launched a **DNS DDoS** on Dyn, crashing Twitter, Netflix, and PayPal.

### The Philips Hue Hack

- Researchers found that "smart" light bulbs:
  - Collected **personal data** (unencrypted).
  - Could be **remotely controlled** via drone (demonstrated in Paris).

### Why IoT Security Is a Joke

- **No Standardization**: Every manufacturer reinvents the wheel.
- **No Encryption**: Your "smart scale" sends weight data in plaintext.
- **No Updates**: Devices abandoned after launch.

### The Lesson

- **Segment IoT devices** (keep them off your main network).
- **Change default credentials**.
- **Assume they're spying on you**.

---

## Case Study 3: Ransomware – Pay Up or Lose Everything

### The Anatomy of an Attack

1. **Delivery**: Phishing email with a malicious attachment (e.g., "invoice.pdf.exe").
2. **Execution**: Malware encrypts files, demands Bitcoin.
3. **Profit**: Attackers rake in millions (or get caught like the Colonial Pipeline hackers).

### The Dridex Example

- **Disguised as** normal HTTP traffic.

- **Red Flag**: `MZ` header (Windows executable) hidden in packets.

### How to Fight Back

- **Backups**: Offline, immutable copies.
- **NoMoreRansom.org**: Free decryption tools.
- **User Training**: Don’t click "Urgent Invoice" emails.

---

## Final Thought: The Cycle Never Ends

As long as humans design technology, we’ll make mistakes. But by studying past failures—like the **Mirai botnet**, the **UPnP backdoor**, or the **ransomware epidemic**—we can break the cycle.

### Your Homework

1. **Capture your UPnP traffic** (filter: `udp.port == 1900`).
2. **Audit IoT devices** (change defaults, segment networks).
3. **Practice forensic analysis** (Wireshark’s `Statistics` menu is your friend).

### Q&A

Let’s discuss—what’s the *one* vulnerability that keeps you up at night?

---

**Contact**: phil@merlionkeep.org | CybersecurityInstitute.edu
**Feedback**: Please share your thoughts—this is how we improve!

*"The only secure system is one that’s powered off, locked in a safe, and buried underground."*
—But since that’s not practical, let’s keep learning.

---

This version adds vivid storytelling, technical depth, and actionable insights while maintaining a conversational tone. Let me know if you'd like any section expanded further!