Network Forensics

in Industrial Networks

Tilbe Ugurel
Tim Nebel

Outline

1. Definition: Network Forensics
2. Selection of Forensic tools & live demo
    - 2.1. tcpdump
    - 2.2. wireshark
    - 2.3. snort
    - 2.4. pcapxray
    - 2.5. scapy
    - 2.6. ARMORE
3. Conclusion: Comparison of tools

Digital forensics [hide]
Computer exams · Data analysis · Database study · Malware analysis · Mobile devices · **Network analysis** · Photography · Video analysis · Audio analysis

Network forensics

From Wikipedia, the free encyclopedia

Network forensics is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence, or intrusion detection.[1] Unlike other areas of digital forensics, network investigations deal with volatile and dynamic information. Network traffic is transmitted and then lost, so network forensics is often a pro-active investigation.[2]

https://en.wikipedia.org/wiki/Network_forensics

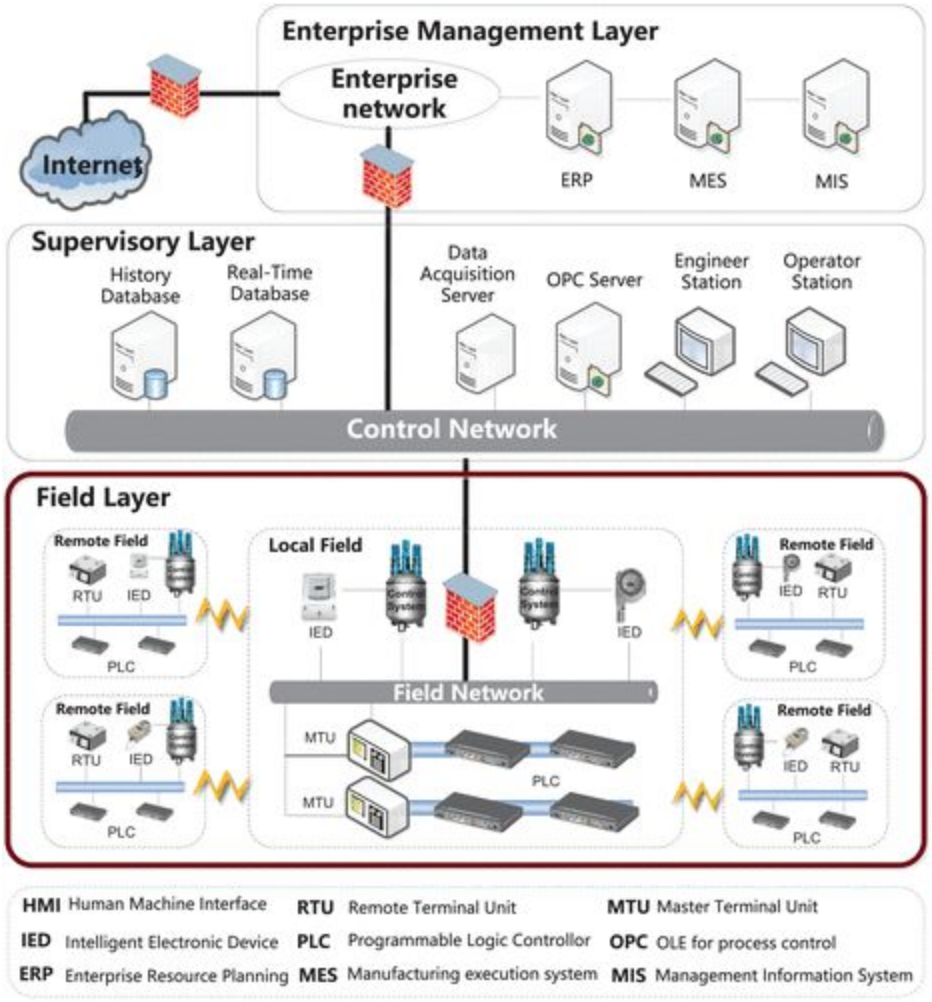
Industrial Networks

Characteristics

- Machine-to-machine communication
- Predictable system behaviour
- Protocols unencrypted
- Usually concise traffic

Challenges

- Unintuitive protocol structure
- Limited computing resources
- Different hardware, software and network protocols



https://journals.sagepub.com/doi/full/10.1177/1550147718794615

ISO/OSI Model (Recap)

Layer			Protocol data unit (PDU)
Host layers	7	Application	Data
	6	Presentation	
	5	Session	
	4	Transport	Segment, Datagram
Media layers	3	Network	Packet
	2	Data link	Frame
	1	Physical	Bit, Symbol

https://en.wikipedia.org/wiki/OSI_model

Selection of Forensic tools

Live Demo

TCPDUMP

What it does:

+ Packet sniffing
+ Packet analysis
+ Filter network traffic
+ passive

What it doesn't:

- IDS
- GUI for packet inspection
- Interfere in network traffic

How it works:

- sniff packets matching boolean expressions
- save captured packets to a file

→ preprocessing network dump for further analysis

https://www.tcpdump.org

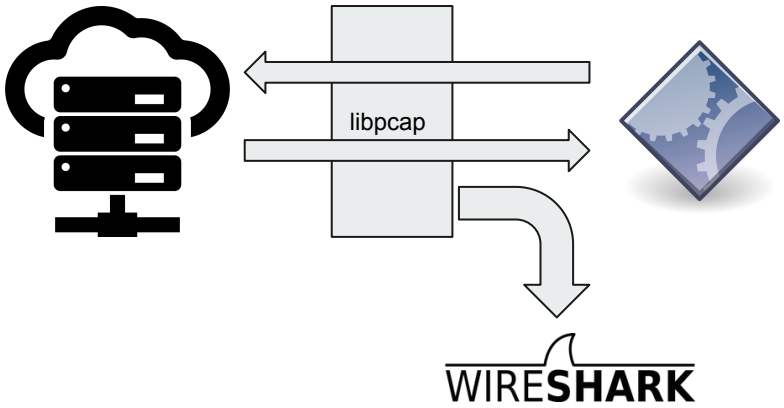
What it does:

+ Packet capture
  + information gathering
  + libpcap / npcap → pcap / pcapng
+ Analysis Tool
+ GUI to inspect captured traffic
+ Visually prepares ISO/OSI model
+ passive

What it doesn't:

- Intrusion Detection System
- send packets
- check legality of operation

➔ copies incoming / outgoing traffic
➔ may conflict with company policy (!)
➔ not "safe" https://www.opencve.io/cve?vendor=wireshark
➔ CLI: tshark (terminal wireshark)
➔ API-daemon: sharkd
➔ DEMO



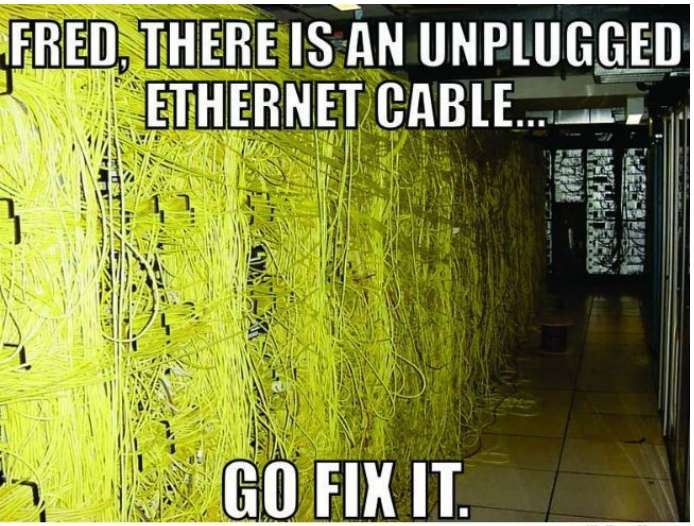
https://www.wireshark.org

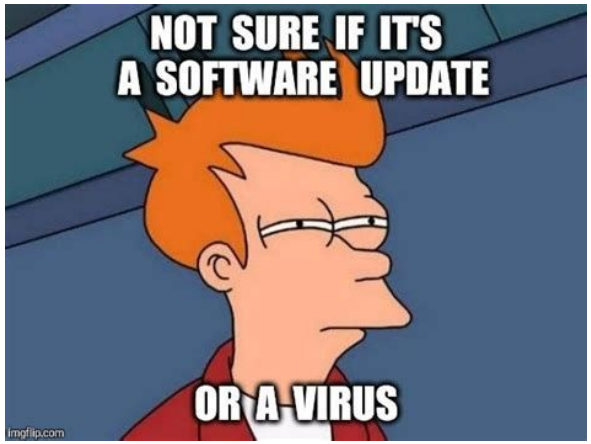
tshark

- ➔ CLI for wireshark
- ➔ Remote capture (like tcpdump)
- ➔ more complex pcap-editing

sharkd

- ➔ Wireshark JSON-API
- ➔ send/receive wireshark information





What it does:

+ IDS & IPS
+ Packet sniffing & capture
+ Analyse captured traffic
+ Block network traffic
+ Can be passive and active

What it doesn't:

- GUI for packet inspection (possibility to visualize alerts in external tools)

How it works:

- Rule-based
- Compare packets with rules/signatures
- Write known attacks in rules

https://snort.org/

Snort: Modbus extension

- Morris et al.
- Monitor and analyze Modbus traffic
- Uses Snort rules
- Accuracy depends on definition of rules
- Proposed 50 Modbus-specific rules

https://ieeexplore.ieee.org/document/6149298

What it does:

+ Plot network diagram using pcap file
+ Display network topologies
+ Highlight important traffic

What it doesn't:

- Packet sniffing
- Capture packets

https://github.com/Srinivas11789/PcapXray

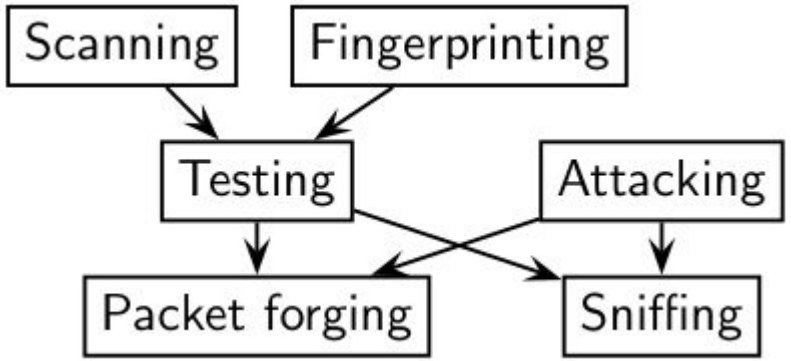
Scapy

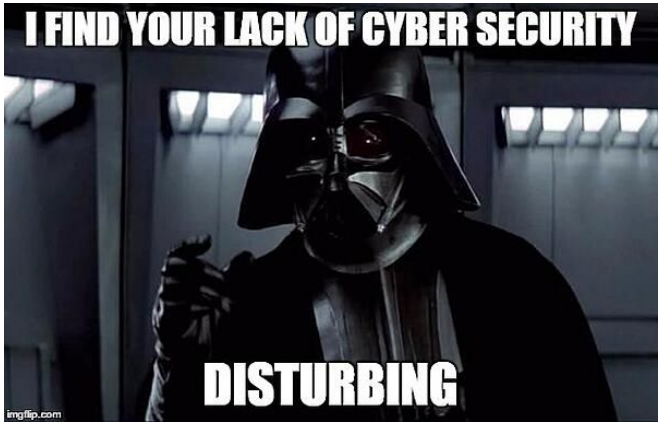
What it does:

+ CLI / Python Library
+ Packet manipulation
  + craft and monitor
  + scanning
  + tracerouting
  + probing
  + unit tests
  + attacks
  + network discovery

What it doesn't:

- Intrusion Detection
- provide ready to execute exploits





https://scapy.net/
https://scapy.readthedocs.io/en/latest/introduction

What it does:

+ IDS/IPS for ICS
+ Detects suspicious communication
+ Enforce defined policies
+ Communication frequencies
+ Function details
+ Collect and visualize statistics
+ Encrypt communication

What it doesn't:

+ Analyze network dumps
+ Send packets
+ Filter network traffic

https://github.com/ITI/ARMORE

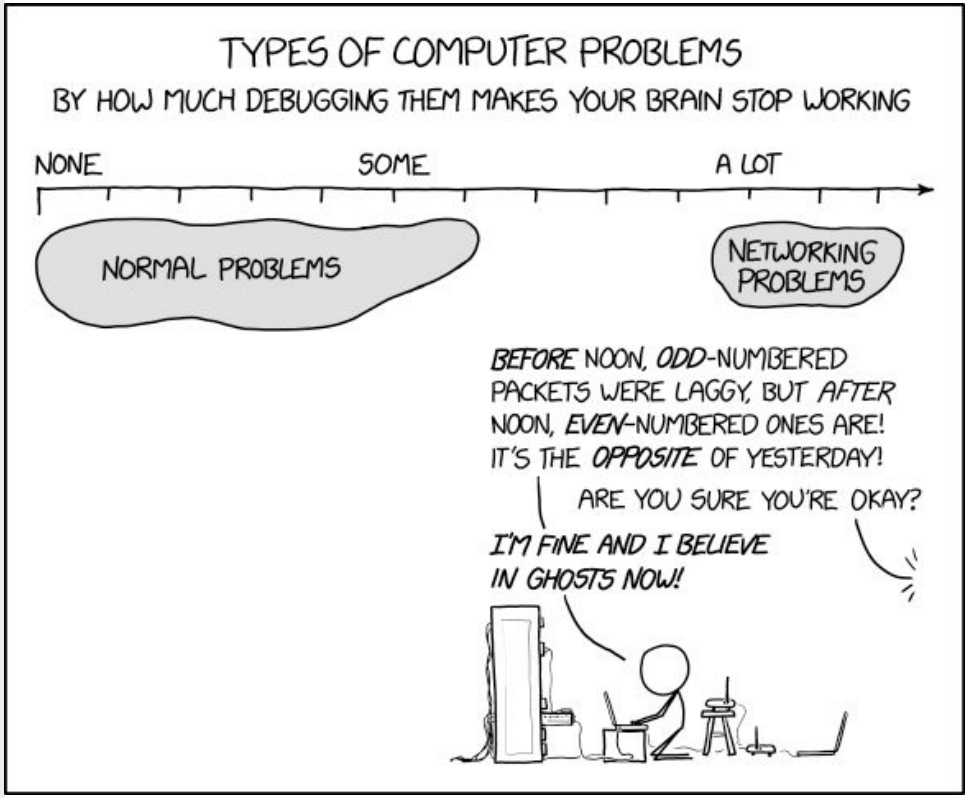
Model-based IDS for SCADA networks

- Cheung et al.
- Process control systems have regular traffic patterns
- Construct models for expected behaviour of the system
- Potential for detecting unknown attacks

- Construct protocol specification model
- Higher false alarm rate
- Difficult/expensive to construct models

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.2076&rep=rep1&type=pdf

Conclusion


TYPES OF COMPUTER PROBLEMS
BY HOW MUCH DEBUGGING THEM MAKES YOUR BRAIN STOP WORKING
NONE
SOME
A LOT
NORMAL PROBLEMS
NETWORKING PROBLEMS
BEFORE NOON, ODD-NUMBERED PACKETS WERE LAGGY, BUT AFTER NOON, EVEN-NUMBERED ONES ARE! IT'S THE OPPOSITE OF YESTERDAY!
ARE YOU SURE YOU'RE OKAY?
I'M FINE AND I BELIEVE IN GHOSTS NOW!

Comparison

Summary & Conclusion

- ➔ lots of capable tools
- ➔ GUI and CLI versions
- ➔ .pcap-files are used as a foundation
- ➔ tools can be interchangeable

- ➔ Using the right tool makes the job easier
- ➔ CLI versions can be leveraged to automate monitoring
- ➔ use visualisation tools for a first “feel”
- ➔ we can’t rely on identifying intrusion with available IDSs
- ➔ start with wireshark!

References & further reading

Demo: https://github.com/loeschzwerg/IIDL-Network-Forensics

Clipart: https://openclipart.org/

Industrial control protocols in the internet core:
https://onlinelibrary.wiley.com/doi/full/10.1002/nem.2158

https://journals.sagepub.com/doi/full/10.1177/1550147718794615

https://github.com/ITI/ICS-Security-Tools/tree/master/tools/analysis

T. Morris, R. Vaughn and Y. Dandass, "A Retrofit Network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems," 2012 45th Hawaii International Conference on System Sciences, 2012, pp. 2338-2345, doi: 10.1109/HICSS.2012.78.