# Malware Analysis

## FUNDAMENTALS

LUKE GROVER

2023-05-16

# $whoami

- Currently a Security Engineer
- Former Software Developer
- 10 years prior in healthcare
- Unquenchable curiosity

# Scope/Disclaimers

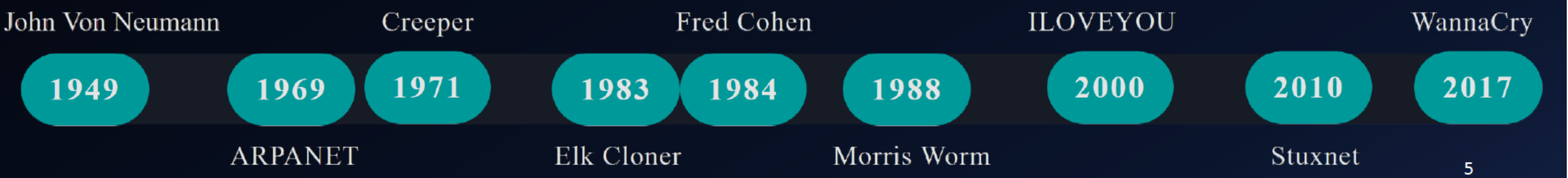
- Keeping presentation under an hour
  - We will not discuss every element of malware analysis
  - Coverage depth will be surface level
- Not technical, focusing on concepts
- Opinions are my own, do not reflect the positions of my employer
- **DO NOT** interact with malware on production connected systems/networks

# Agenda

- Brief History
- Definitions/Classifications
- Use Cases
- Analysis Strategies
- Malware Handling
- Lab Demo
- Q&A

# Malware Origins – Some Notable Events

- John von Neumann – *Theory and Organization of Complicated Automata*
- ARPANET – Early packet switched network using TCP/IP protocol
- Creeper – “I'm the creeper, catch me if you can!”
- Elk Cloner – “But…Macs don’t get viruses!?”
- Frederick Cohen – *Computer Viruses - Theory and Experiments*
- Morris Worm – “Grandfather of Worms”, *DOJ has entered the chat*
- ILOVEYOU – Catching the “Love Bug”
- Stuxnet – Nation states joining in on the fun
- WannaCry – hoarding exploits…what could go wrong?



John Von Neumann
1949
1969
ARPANET
Creeper
1971
1983
Elk Cloner
Fred Cohen
1984
1988
Morris Worm
ILOVEYOU
2000
2010
Stuxnet
WannaCry
2017

# What is Malware?

- Any software that does malicious things
  - Simple enough, right?
- Depends, who’s asking
  - What about legitimate tools…PsExec?
  - Hello, wtfbins.wtf

# Defining our Terms

- Trojan
  - Dropper (downloader/loader)
  - Backdoor (RAT)
  - Ransomware
  - Infostealer
  - Spyware
  - Banker
  - DoS
  - DDos
  - Wiper
  - Clicker
  - Miner
  - Spammer
  - Keylogger
  - hacktools

- Virus
- Worm
- Rootkit
- Exploit
- Potentially Unwanted Application (PUA)
- Adware
- ...and many more

# Classifying Malware

- Commodity
  - Readily available, used by multiple threat groups
  - **Usually automated** – commonly email phishing/malspam
  - Examples
    - Redline Stealer
    - Qbot
- Bespoke
  - Customized, specific objective (targeted industry/company/person)
  - **Human-operated** - “hands on keyboard”
  - Advanced Persistent Threat (APT)
  - Examples
    - Stuxnet
    - SUNBURST/TEARDROP

# Use Cases

- Security Operations Center (SOC) Alerting and Triage
- Digital Forensics and Incident Response (DFIR)
- Cyber Threat Intelligence
- Threat Hunting/Detection Engineering
- Malware Research

# Different Strategies for Different Audiences

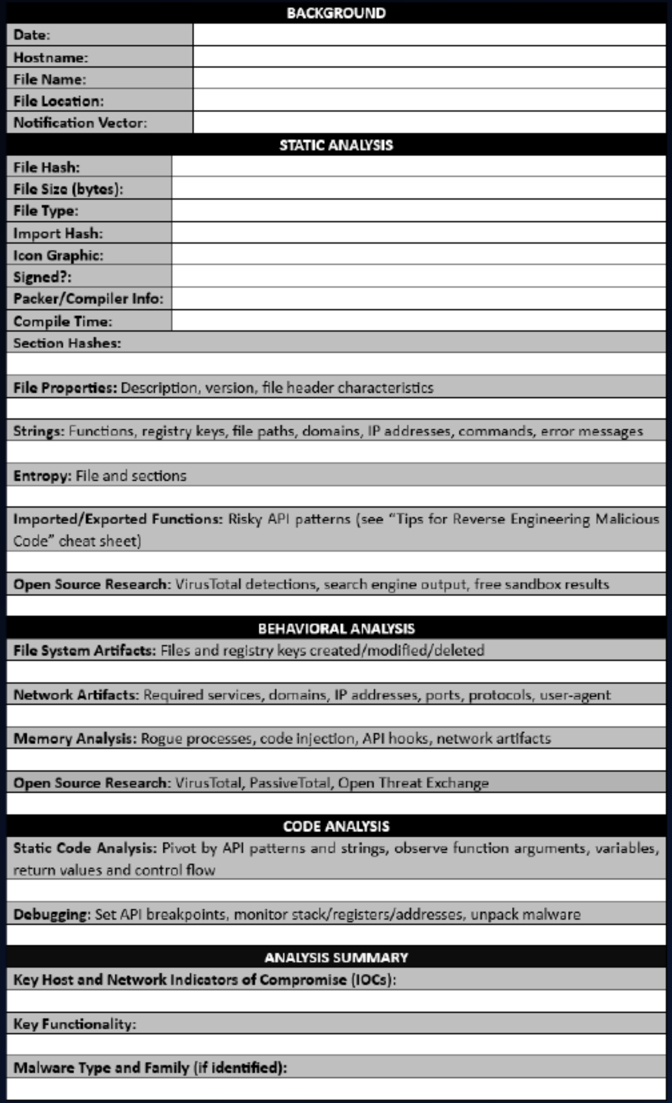
- Understand what brings value to your audience
- Technical Report vs Executive Summary
- What Intelligence Requirements need to be met?
  - Strategic
  - Operational
  - Tactical
- Available resources (time/expertise) determines goal prioritization

# Answering Stakeholder Questions

- Executives (CISO)
  - Impact
  - Blast radius
  - Containment/Recovery plan
- SOC
  - Indicators of Compromise (IoCs)
  - Detection rules
- Incident Response Team(IR)
  - Fill in context gaps, what happened
  - What were the attackers’ objectives/targets
  - Assist in containment/recovery
- Vulnerability Management
  - Zero-Day involved?
  - Did the attack leverage vulnerabilities that need to be patched

# Authoring an Analysis Report

- Background/Identifiers
  - File name, type, size, hash
  - Location Found
  - Discovery/Notification vector
- Static Analysis Findings
  - Passive
  - Strings
  - Imported/Exported Functions
- Dynamic/Behavioral Findings
  - Active
  - Registry, filesystem, process, network, and memory activities
- Reverse Engineer/Code Findings
  - Disassembly/Debugging
  - Identifying functionality not seen in prior analysis
- Analysis Summary
  - Capabilities
  - IoCs
  - Detection Rules
  - Remediation Steps



| | |
|---|---|
| **BACKGROUND** | |
| **Date:** | |
| **Hostname:** | |
| **File Name:** | |
| **File Location:** | |
| **Notification Vector:** | |
| **STATIC ANALYSIS** | |
| **File Hash:** | |
| **File Size (bytes):** | |
| **File Type:** | |
| **Import Hash:** | |
| **Icon Graphic:** | |
| **Signed?:** | |
| **Packer/Compiler Info:** | |
| **Compile Time:** | |
| **Section Hashes:** | |
| | |
| **File Properties:** Description, version, file header characteristics | |
| | |
| **Strings:** Functions, registry keys, file paths, domains, IP addresses, commands, error messages | |
| | |
| **Entropy:** File and sections | |
| | |
| **Imported/Exported Functions:** Risky API patterns (see "Tips for Reverse Engineering Malicious Code" cheat sheet) | |
| | |
| **Open Source Research:** VirusTotal detections, search engine output, free sandbox results | |
| | |
| **BEHAVIORAL ANALYSIS** | |
| **File System Artifacts:** Files and registry keys created/modified/deleted | |
| | |
| **Network Artifacts:** Required services, domains, IP addresses, ports, protocols, user-agent | |
| | |
| **Memory Analysis:** Rogue processes, code injection, API hooks, network artifacts | |
| | |
| **Open Source Research:** VirusTotal, PassiveTotal, Open Threat Exchange | |
| | |
| **CODE ANALYSIS** | |
| **Static Code Analysis:** Pivot by API patterns and strings, observe function arguments, variables, return values and control flow | |
| | |
| **Debugging:** Set API breakpoints, monitor stack/registers/addresses, unpack malware | |
| | |
| **ANALYSIS SUMMARY** | |
| **Key Host and Network Indicators of Compromise (IOCs):** | |
| | |
| **Key Functionality:** | |
| | |
| **Malware Type and Family (if identified):** | |
| | |

https://github.com/as0ni/templates/blob/master/Malware_Analysis_Template.docx

# Tips When Handling Malware

- Do not allow malicious samples to touch any system you are not willing to destroy
  - Analysis should occur isolated from production systems/environments
  - Use dedicated bare metal/analysis VMs
- Disarm samples when not in use
  - Defang IPs, URLs
  - Remove/change file extensions
  - Compress (zip/7zip) sample with password
- Manage network connectivity
  - Disable network access when not needed
  - Segment/Firewall network if network access needed
- Protect analysis VMs with proper snapshot practices
- Regularly patch and backup host environment

# Closing Thoughts

- Understand the fundamentals
- Effective Malware Analysis is a force multiplier
- Know your audience
- Safety first

- Slides available for download:
  - https://github.com/LukeGrover-Public/presentations

# References

- Cohen, F. (1987). Computer viruses: Theory and experiments. *Computers & Security, 6*(1), 22-35. https://doi.org/10.1016/0167-4048(87)90122-2
- Cucci, K. (2024). *Evasive Malware (Early Access Edition 5/2/23)*. O'Reilly.
- Heuer, R. (2021). *Psychology of Intelligence Analysis*. Martino Fine Books.
- Kleymenov, A., & Thabet, A. (2022). *Mastering Malware Analysis: A malware analyst's practical guide to combatting malicious software, APT, cybercrime, and IoT attacks* (Second). Packt Publishing Ltd.
- Mohanta, A., & Saldanha, A. (2020). *Malware Analysis and Detection Engineering: A comprehensive approach to detect and analyze modern malware*. Apress.
- Roberts, S., & Brown, R. (2017). *Intelligence-Driven Incident Response: Outwitting the Adversary*. O'Reilly.
- Soni, A. (2021, June 12). Malware Analysis Template. https://github.com/as0ni/templates/blob/master/Malware_Analysis_Template.docx
- Von Neumann, J., & Burks, A. W. (1966). *Theory of self-reproducing automata.* University of Illinois Press.
- Zeltser, L. (2023, February). *Malware Analysis: Tips & Tricks Poster*. Sans Institute. https://sansorg.egnyte.com/dl/JLJariGJOZ