CIRCL
Computer Incident
Response Center
Luxembourg

Introduction to endpoint visibility

Quentin JEROME (quentin.jerome@circl.lu)

2025/05/26

University of Luxembourg - Luxembourg

Question?

Do you have an idea
how antivirus software works?

How Antivirus Works

- Antivirus software primarily uses signature-based detection:
  - Scans files and processes for known malware signatures
  - Relies on databases regularly updated with new threat signatures
- Some AVs use heuristic or behavior-based detection to catch unknown threats
- However, advanced threats often evade detection by:
  - Using custom or polymorphic malware
  - Employing stealth techniques and zero-day exploits

Antivirus Management in Large Organizations

- Antivirus software is just another IT tool to deploy and maintain
- Often managed by software or IT teams, not specialized security teams
- Limited involvement of security experts in configuring or responding to alerts
- AV alerts often handled as routine IT issues rather than security incidents

Question?

Is the traditional antivirus model effective at detecting advanced targeted threats?

Not Really!

- Traditional antivirus relies mostly on known signatures (static and dynamic)
- Advanced targeted threats use custom malware, polymorphic code, unknown techniques and sometimes zero-days
- Such threats often evade signature-based detection (because they have tested their malware against the AV the targeted entity is running)
- Heuristic and behavior-based methods improve detection but have limits

Limitations of AV Software

(Ideally) they have to **detect** all malware and they have **not to detect** all benign software, this in a single product distributed to all customers!

- What is a malware?
- Does everyone have the same definition?

> AV impose to their definition of what is a malware and what is not. While providing no context about their detections.

Question?

What is the main objective when doing incident response?

What Are We Trying to Achieve in Incident Response?

- The main objective: **gather context around an alert** to understand the incident
- Context includes:
  - What triggered the alert?
  - What happened before and after?
  - Which users, processes, and systems were involved?
- This contextual information is essential to:
  - Assess the impact and scope of the incident
  - Build and update the **incident timeline**
  - Guide response actions and recovery

Incident Response Mantra

💡 Without context, alerts are just noise.

- An alert alone tells you something happened — but not what, why, or how.
- Context transforms a raw signal into actionable intelligence:
  - Who did it?
  - What else happened around the same time?
  - Is it part of a larger pattern?
- Effective incident response is not just reacting to alerts — it's **investigating with context**.

Question?

Knowing that, how do you think incident response goes with AV alerts only?

Bad

- Limited visibility into what actually happened during an attack
- Difficulty in tracing attack vectors and lateral movements
- Delays in detection reduce chances for quick containment
- Lack of rich context makes investigations slow and inefficient

> Security teams need transparent, flexible, and extensible tools to monitor endpoints → **endpoint visibility software**

What is an Endpoint Visibility Software?

A tool that monitors and collects information from endpoints (servers, workstations, network devices)

- Provides real-time insights into what is happening on a system:
    - Process executions
    - File accesses and modifications
    - User activities
    - Network connections
- Helps detect suspicious or malicious behaviors
- Essential to implement **customized / tailored** threat detection scenarios

Beyond Detection: Incident Response Value

Endpoint visibility tools are not only useful for threat detection

- They provide **invaluable data** during incident response:
  - Timeline reconstruction of attacker activity
  - Identification of persistence mechanisms
  - Tracing lateral movement across systems
  - Understanding the scope and impact of the breach
- Without visibility, responders are often blind, forced to make assumptions while relying on time consuming analysis methods
  - Quick artifact acquisition (`osquery`, `velociraptor`)
  - Disk acquisition (need someone with a physical access to the device)
  - Artifact / disk acquisition take time and any time saved in incident response is good to take

Specific Terminology

- **Security event**: an event happening on a system which may indicate a potential security incident

  Example: A log entry that indicates the execution of a given binary

- **Security monitoring tool**: a tool monitoring a system or an infrastructure and generating security events for analysis.

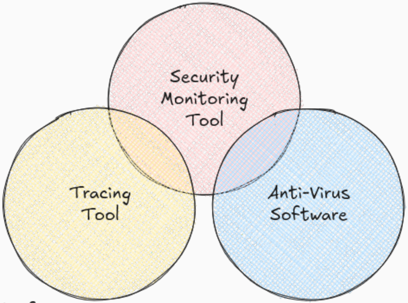
  Example: Intrusion Detection Systems (IDS), Endpoint Detection and Response (EDR)

- **Security visibility**: the ability to monitor,detect and analyze security events

  Example: A company using a security monitoring software increases its security visibility

Different Software for Different Purposes

Goal: be a great source of events to improve security visibility



Goal: be good at detecting known threats

Non-Goal: provide many security events
- not good for security analysts
- hard to understand why the software took this or that decision

Goal: provide events for performance diagnosis / debugging
- some event can be used as security event

Non-Goal: provide security sound data
- information is limited

Question

Do you know any endpoint visibility software?

Current Landscape of Endpoint Visibility Software

- Real-time visibility is essential for detecting and responding to threats
- Few open-source tools provide comprehensive real-time event collection
- **Linux:**
  - `Auditd` – Kernel-level auditing, powerful but complex
  - `Kunai` – Real-time system event collection and threat hunting
  - `Falco` – Real-time detection focused on containers and syscall activity
- **Windows:**
  - `Sysmon` – Free tool from Microsoft for detailed event logging (not open-source)
  - `Wazuh` – Some real-time monitoring via event/log analysis

Key Steps for Effective Endpoint Visibility (1/2)

1. **Monitor Endpoints**
   - Deploy lightweight agents or kernel features to collect real-time events
2. **Define Logging Policies**
   - Determine what events are relevant (processes, file changes, network activity)
   - Balance between data volume and usefulness
3. **Forward and Aggregate Logs**
   - Securely send logs to central servers or SIEM platforms
   - Ensure reliability and scalability

Key Steps for Effective Endpoint Visibility (2/2)

4. **Analyze and Correlate**
    - Use rules, heuristics, and behavioral analytics to detect anomalies
    - Enrich events with contextual data for better investigations
5. **Respond and Hunt**
    - Trigger alerts, perform threat hunting, and conduct incident response
    - Continuously refine detection and response strategies

With Great Power Comes Great Responsibilities (1/2)

- **1. Choosing the right data:**
  - Select logs that are useful for detection
  - Select logs that support incident response and forensics
- **2. Performance considerations:**
  - Minimize CPU, memory, and disk impact on monitored endpoints
  - Ensure tools do not degrade system reliability
- **3. Backend pressure:**
  - Limit log volume to avoid overloading storage and processing pipelines
  - Enable long-term retention with reasonable storage costs

With Great Power Comes Great Responsibilities (2/2)

- **4. Signal-to-noise ratio:**
  - Avoid collecting redundant or low-value events
  - Prioritize actionable and context-rich logs
- **5. Security and privacy:**
  - Protect collected data (it may contain sensitive information)
  - Comply with legal and organizational data retention policies

> **Analyst / detection engineers** will want all the logs possible but you will have to do **trade-offs**

Theory vs. Practice in Endpoint Visibility

In theory, comprehensive endpoint monitoring covers all systems and activities

- In practice, blind spots remain due to:
  - Legacy systems running critical but unmodifiable applications
  - Network appliances and proprietary hardware with limited visibility
  - Constraints on deploying agents or updating software
- These gaps require tailored solutions and additional network-level monitoring

> Endpoint visibility is necessary but not always sufficient for full coverage

Question

Do you believe we should throw antivirus away?

Of Course Not

- Antivirus can still serve as a first layer of defense
- Antivirus have very solid signature database detecting many commodity malware you don't want to bother with
- It might detect some threats or serve as a trigger for deeper investigations
- AV should be used **in conjunction with** a solid endpoint visibility strategy

> Combining AV software with a good endpoint visibility strategy improves detection rates and shortens incident response time

Key Takeaways

- **Traditional antivirus** mainly detects known threats using signatures — often ineffective against advanced attacks.
- **AV alerts lack context**, making incident response difficult and slow.
- **Context is king**: effective incident response requires understanding what happened before, during, and after an alert.
- **Endpoint visibility tools** provide real-time, high-context data critical for detection and response.
- **With great power comes responsibility**: collecting the right logs, minimizing system impact, and balancing retention and privacy are essential.
- **AV is not obsolete** — it still plays a role as part of a layered security approach.

Linux Endpoint Visibility with Kunai

What is Kunai?

Kunai is a free and open-source Linux endpoint monitoring and hunting tool

- Designed to provide deep visibility into what happens on a Linux system
- Features:
  - Lightweight agent for collecting real-time system events
  - Extensible rule engine for detecting suspicious or malicious activity
  - Outputs structured `JSON` events for easy integration and analysis
  - Correlates with tools like **Suricata** and **Zeek** for enriched detection
- Aimed at individuals, companies, and public sector entities looking to improve visibility and response

But, why Kunai?

> Kunai is **open-source** and is developed here in **Luxembourg**

Many concepts you will learn with Kunai will be transposable into other tools

Q&A

Any question?

Exercises

https://hdoc.csirt-tooling.org/5VEw-vd_ShmRs0T0307lSQ#