33rd USENIX Security Symposium · August 16, 2024 · Philadelphia, PA, USA

You Cannot Escape Me: Detecting Evasions of SIEM Rules in Enterprise Networks

Rafael Uetz[1]
Marco Herzog[1]
Louis Hackländer-Jansen[1]
Simon Schwarz[2]
Martin Henze[1,3]

fkie-cad/amides

[1] Fraunhofer FKIE

[2] GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

[3] RWTH AACHEN UNIVERSITY

Threat Detection in Enterprise Networks

Adversaries frequently attack and intrude enterprise networks

→ Data theft, sabotage, extortion
→ Timely detection of malicious activity vital to limit damage
→ CSOCs perform centralized security monitoring using SIEM systems

Expert-written rules are (still) the primary means for threat detection

- Also called Misuse Detection
- Outperforms anomaly detection (more true and less false alerts)
- Alerts are easy to understand and can be tuned to the environment

Many organizations rely on community-driven, open-source rules

- Go nicely with open-source security monitoring stacks (OpenSearch etc.)
- Several projects exist; most notable: Sigma (https://github.com/SigmaHQ/sigma)
- Sigma has ~3000 rules, ~500 contributors, converters for various SIEM products

Can Adversaries Deliberately Evade Rules?

Example: Evading a Sigma rule for process creation events

- `wmic.exe /node:mailserver … SetAllowTSConnections 1` ✓
- `wmic.exe -node:mailserver … SetAllowTSConnections 1` ✗

We analyzed 292 Sigma process creation rules w.r.t. potential evasions

- Evasion = Command does the same, but rule does not trigger
- We discovered five evasion types:

Insertion	* /create *	schtasks.exe /create …	schtasks.exe /"create“ …
Substitution	… -O …	curl -O http://…	curl --remote-name http://…
Omission	*cscript.exe *.vbs	cscript.exe evil.vbs	cscript evil.vbs
Reordering	* -ma ls*	procdump.exe -ma ls	procdump.exe ls -ma
Recoding	*address=127.0.0.1*	…address=127.0.0.1,…	…address=2130706433,…

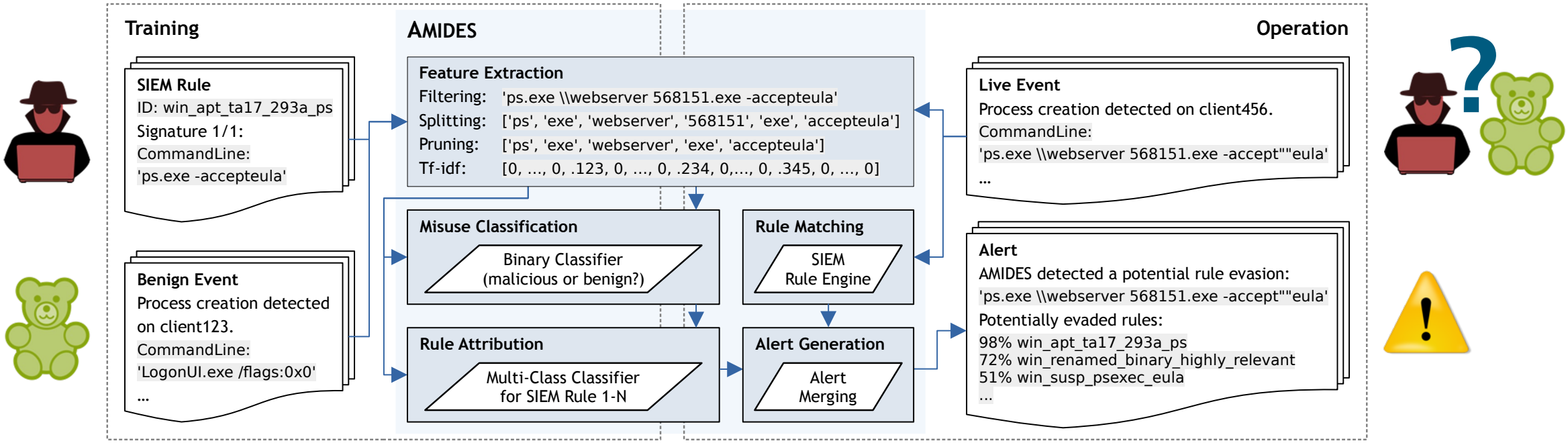
Result: 38% fully, 7% partially evaded → major detection blind spots!

```
title: Webshell Detection With Command
    Line Keywords
description: Detects certain command
    line parameters often used during
    reconnaissance activity via web shells
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage:
            - '*\apache*'
            - '*\tomcat*'
            - '*\nginx.exe'
            …
        CommandLine:
            - '*wmic.exe /node:*'
            - '*whoami*'
            - '*net user *'
            …
    condition: selection
…
```

How Can We Detect Such Evasions?

Basic idea: Detect events that are *similar* to those triggering rules

- Approach: Supervised learning from rules versus benign events!
- Also allows to estimate which detection rules were evaded (*Rule Attribution*)
- We call this idea "*Adaptive* Misuse Detection" and its implementation "Adaptive Misuse Detection System (AMIDES)"



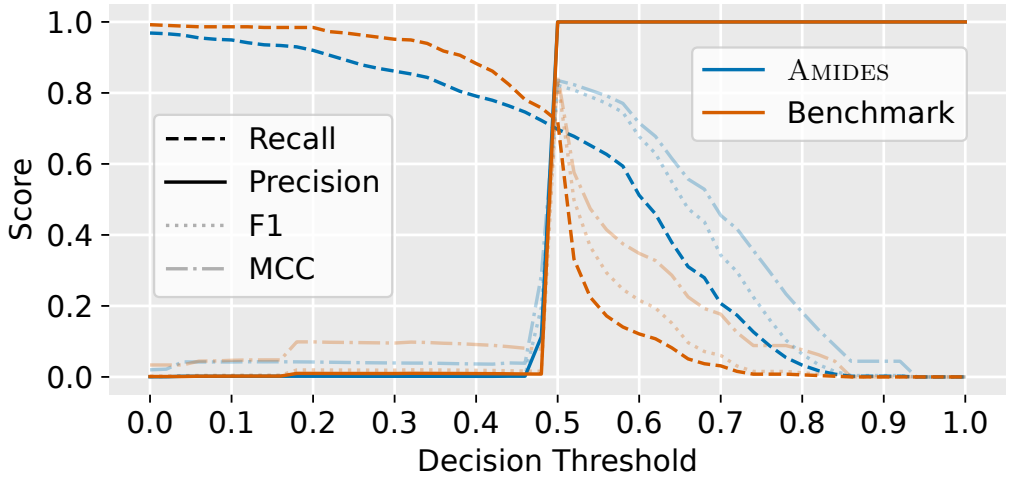
Evaluation

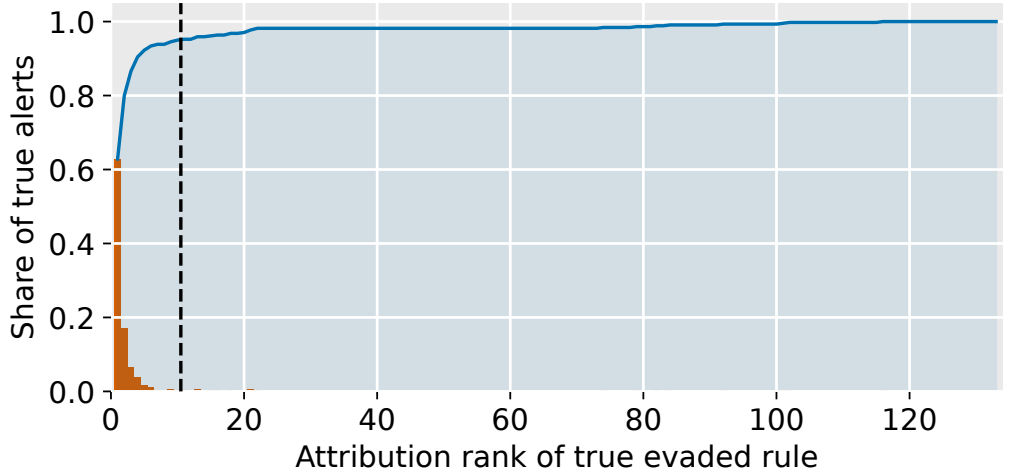
We evaluated AMIDES in a large enterprise network

- More than 50,000 users
- Four weeks of Windows process creation events (~155 million)
- 266 unique executable filenames

Research questions and answers

- RQ1: How well does AMIDES detect rule evasions?
  - ➔ AMIDES detects 70% of our evasions at zero false alerts
  - ➔ Keeps up with learning from malicious events instead of rules
- RQ2: How accurate is the rule attribution?
  - ➔ The evaded rule is within the top 10 for 95% of evasions
- RQ3: Is AMIDES suited for real-world operation?
  - ➔ Yes. 156k EPS, 42 minutes training, thus by far fast enough
  - ➔ Evasions in benign training data cause graceful degradation
  - ➔ Other rule & event types work as well (web, registry, PowerShell)





Recap

- Many organizations rely on misuse detection rules such as Sigma to discover intrusions
- We showed that almost half of the ~300 analyzed rules can be evaded easily
- We introduced Adaptive Misuse Detection and AMIDES to detect such evasions
- AMIDES is fit for purpose and freely available

 fkie-cad/amides  rafael.uetz@fkie.fraunhofer.de  ru37z