"T.I.S.P. Community Meeting 2022"

Berlin, 09.-10.11.2022

Cloud Security is a data Problem

Sascha Dubbel, T.I.S.P.
Tech. Partner Manager CEUR, Lacework

Schützt klassische Security auch meine Cloud?

TRADITIONAL SECURITY
WATERFALL
NETWORK
SOC
AGILE AND THE CLOUD CAUSED IT

Die Vorteile haben Ihren Preis: RISIKO

- Komplexität
- Mangelnde Sichtbarkeit
- GRC Anforderungen
- Klassische (endpoint & on-premises) Lösungen oft unpassend

Cloud security is a data problem

And it requires a fundamentally different approach

Enormous scale

Evolving technologies

Constant changes

Adaptive infrastructure

Talent shortage

Massive amounts of data, hard to interpret at scale

Traditional way

- Rules-based
- Watching the data
- Static
- Too many alerts

Cloud provider security services & tools

This is where cloud security becomes a data problem

Data sources

CloudTrail Logs

VPC Flow Logs

Vulnerability findings

Resources

Data types

Millions lines of logs

Millions of network connections

Tens of thousands of findings

Thousands of resources

Cloud provider security services & tools

This is where cloud security becomes a data problem

Data sources	Data types
CloudTrail Logs	Millions lines of logs
VPC Flow Logs	Millions of network connections
Vulnerability findings	Tens of thousands of findings
Resources	Thousands of resources
Workloads (processes, containers, etc.)	Billions of behaviours

Cloud security posture, compliance and vulnerability management

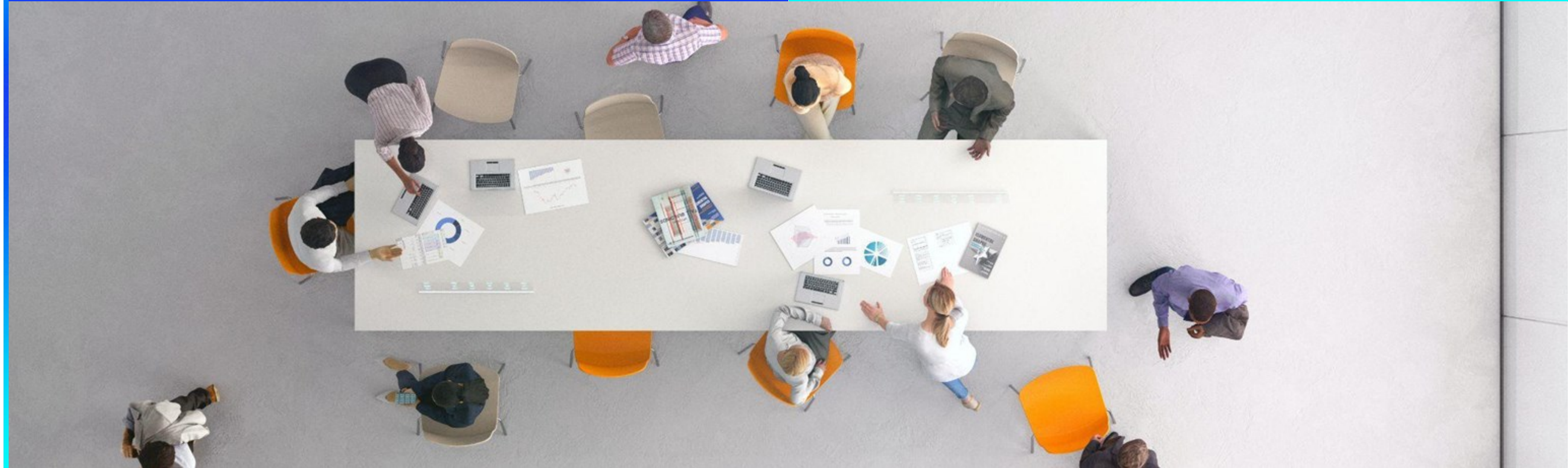
Rules-based

Identify known security risks

Anomaly and threat detection

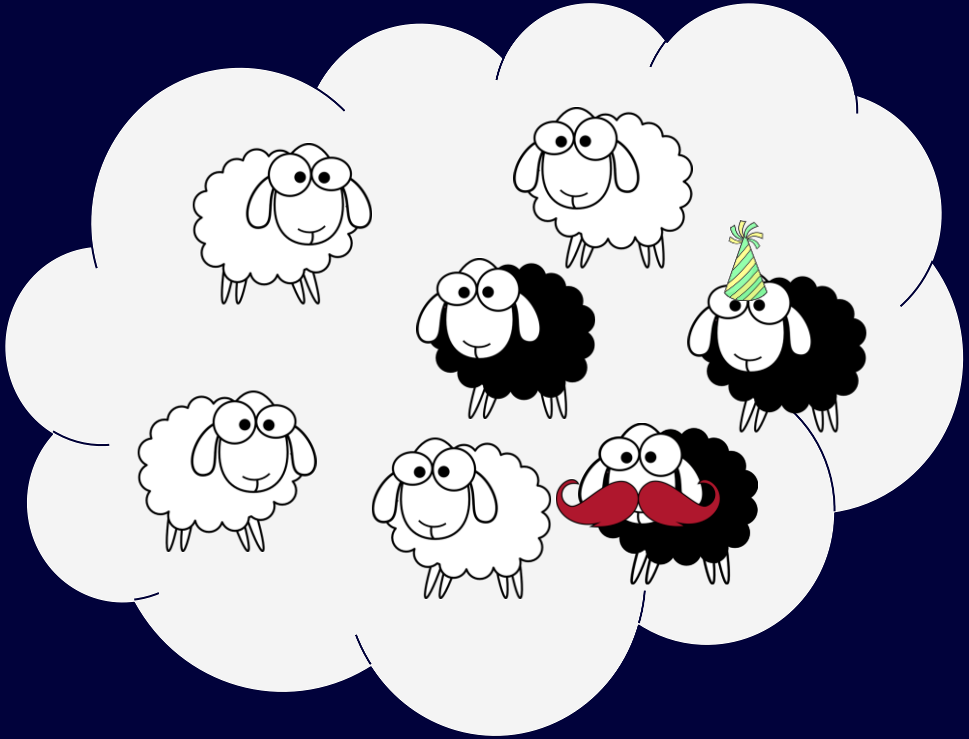
Rules-based

Identify known security threats

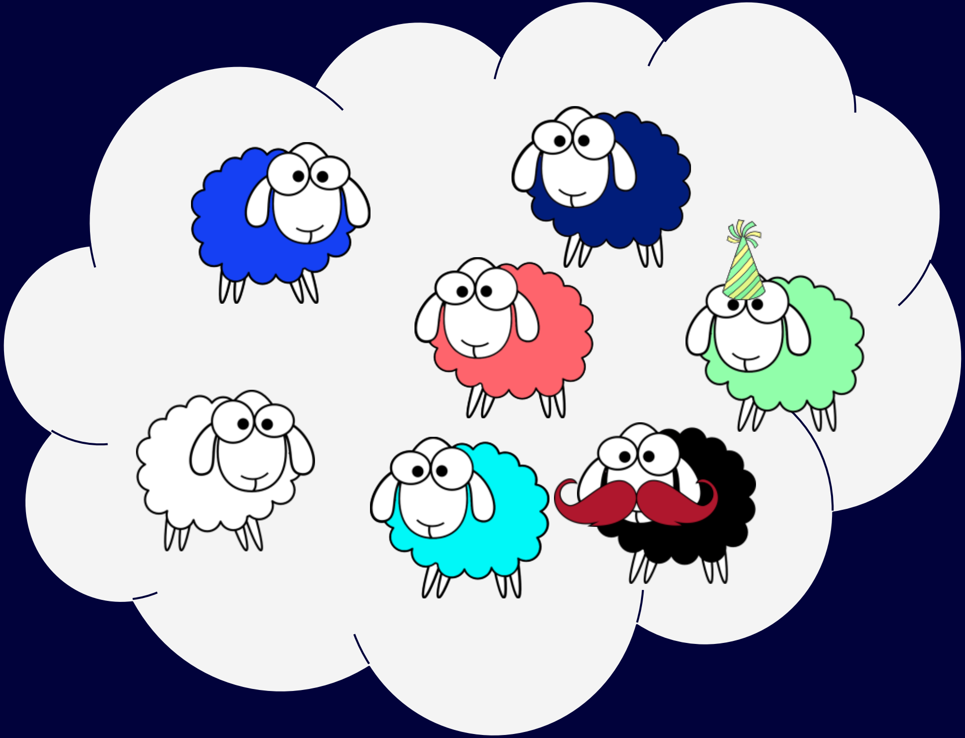


What about the unknown security threats?

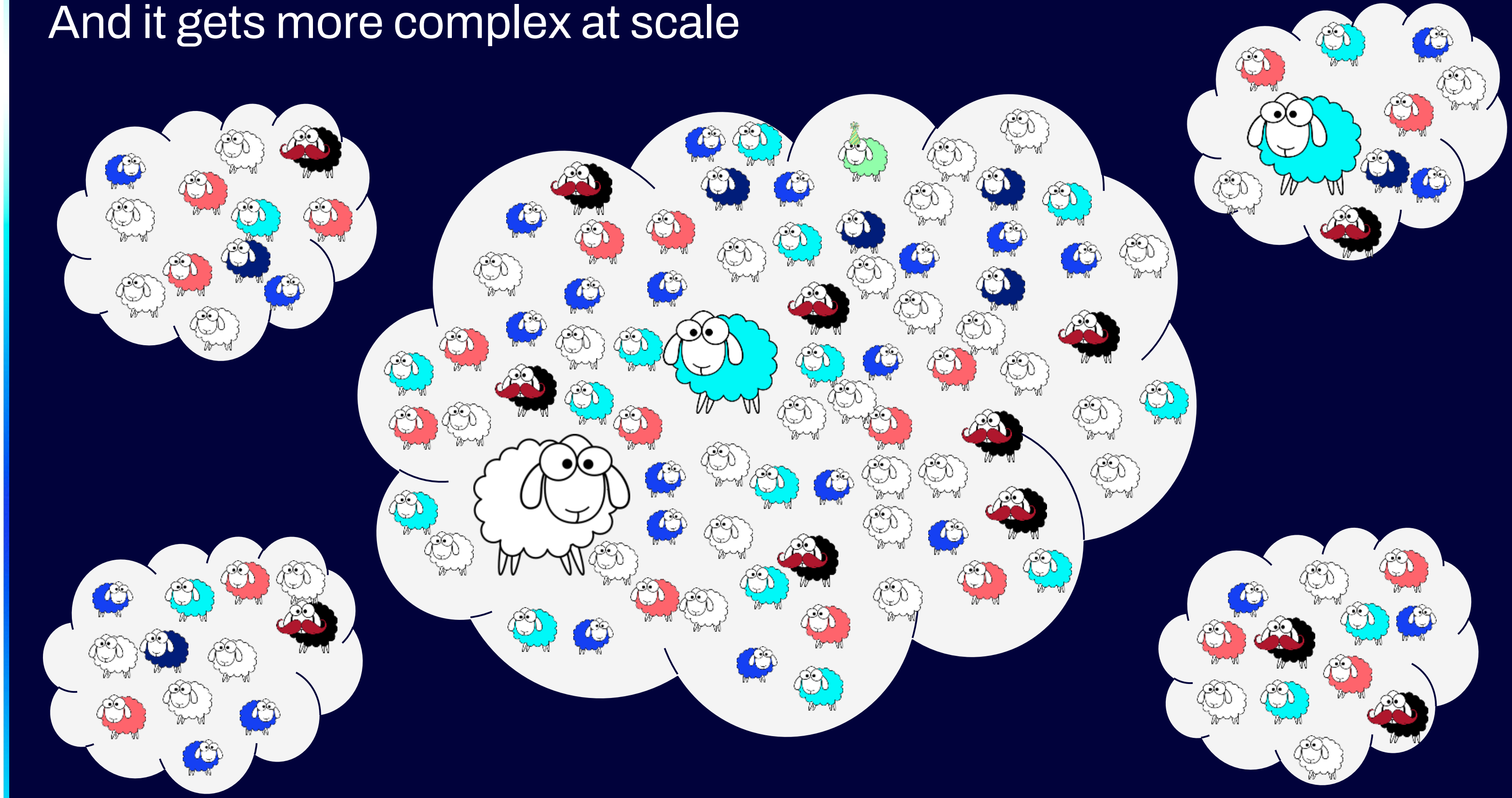
Rule-based threat or anomaly detection



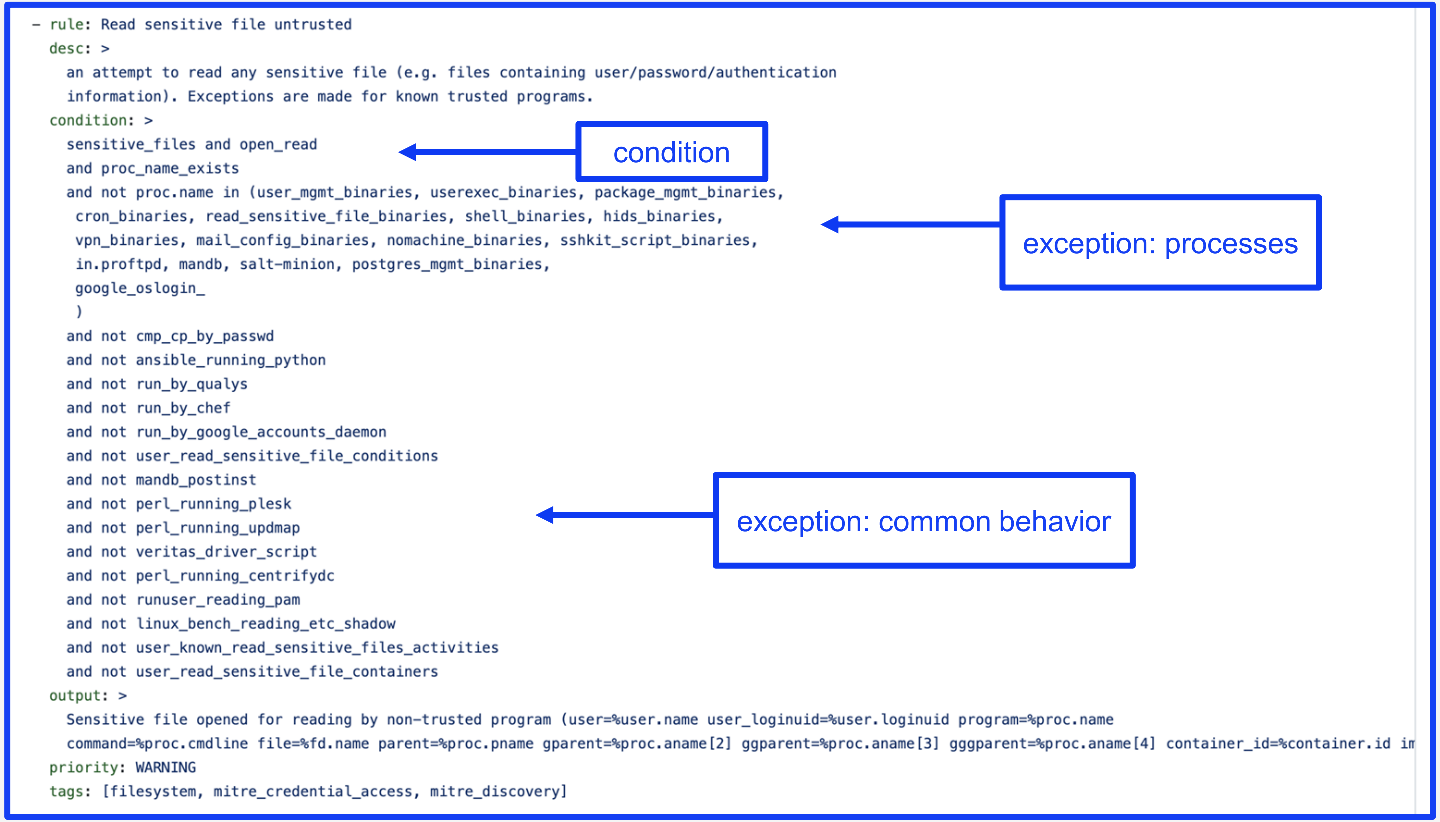
Behavioral Anomaly Detection



And it gets more complex at scale



Rule Based Security



```
- rule: Read sensitive file untrusted
  desc: >
    an attempt to read any sensitive file (e.g. files containing user/password/authentication
    information). Exceptions are made for known trusted programs.
  condition: >
    sensitive_files and open_read
    and proc_name_exists
    and not proc.name in (user_mgmt_binaries, userexec_binaries, package_mgmt_binaries,
     cron_binaries, read_sensitive_file_binaries, shell_binaries, hids_binaries,
     vpn_binaries, mail_config_binaries, nomachine_binaries, sshkit_script_binaries,
     in.proftpd, mandb, salt-minion, postgres_mgmt_binaries,
     google_oslogin_
     )
    and not cmp_cp_by_passwd
    and not ansible_running_python
    and not run_by_qualys
    and not run_by_chef
    and not run_by_google_accounts_daemon
    and not user_read_sensitive_file_conditions
    and not mandb_postinst
    and not perl_running_plesk
    and not perl_running_updmap
    and not veritas_driver_script
    and not perl_running_centrifydc
    and not runuser_reading_pam
    and not linux_bench_reading_etc_shadow
    and not user_known_read_sensitive_files_activities
    and not user_read_sensitive_file_containers
  output: >
    Sensitive file opened for reading by non-trusted program (user=%user.name user_loginuid=%user.loginuid program=%proc.name
    command=%proc.cmdline file=%fd.name parent=%proc.pname gparent=%proc.aname[2] ggparent=%proc.aname[3] gggparent=%proc.aname[4] container_id=%container.id im
  priority: WARNING
  tags: [filesystem, mitre_credential_access, mitre_discovery]
```

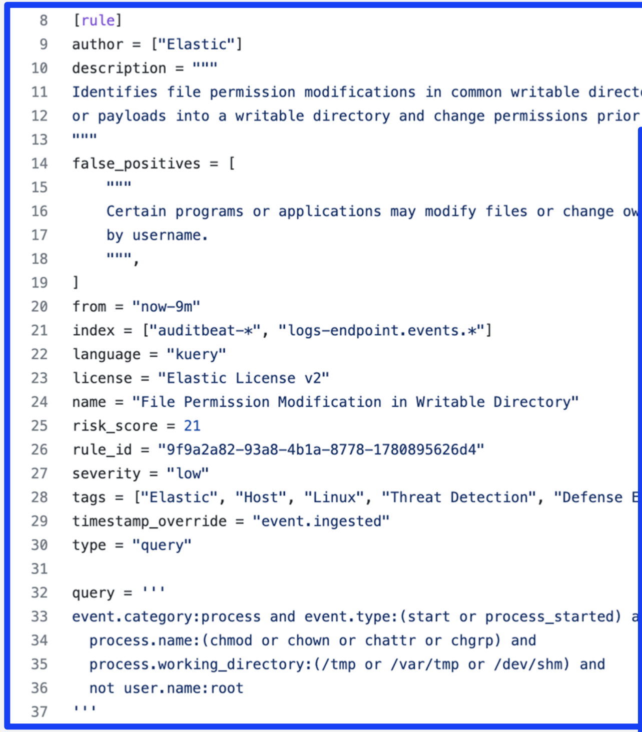
condition

exception: processes

exception: common behavior

LACEWORK

Risk Scoring



```
[rule]
author = ["Elastic"]
description = """
Identifies file permission modifications in common writable directories by a non-root user. Adversaries often drop files
or payloads into a writable directory and change permissions prior to execution.
"""
false_positives = [
    """
    Certain programs or applications may modify files or change ow
    by username.
    """,
]
from = "now-9m"
index = ["auditbeat-*", "logs-endpoint.events.*"]
language = "kuery"
license = "Elastic License v2"
name = "File Permission Modification in Writable Directory"
risk_score = 21
rule_id = "9f9a2a82-93a8-4b1a-8778-1780895626d4"
severity = "low"
tags = ["Elastic", "Host", "Linux", "Threat Detection", "Defense E
timestamp_override = "event.ingested"
type = "query"

query = '''
event.category:process and event.type:(start or process_started) a
  process.name:(chmod or chown or chattr or chgrp) and
  process.working_directory:(/tmp or /var/tmp or /dev/shm) and
  not user.name:root
'''
```

Unsupervised Machine Learning

Anomaly Detection:
Does **a car** park often at this space?
Does **this car** park often at this space?
Does **this car** park anywhere next to the other bank branches?
Do **cars from that state or city** park at this space or next to the other bank branches?

Suppression:
Is this **an emergency** (police / fire department / ambulance) vehicle?

IoC enrichment:
Has this car been **stolen?**
Has this car been used in a **robbery?**

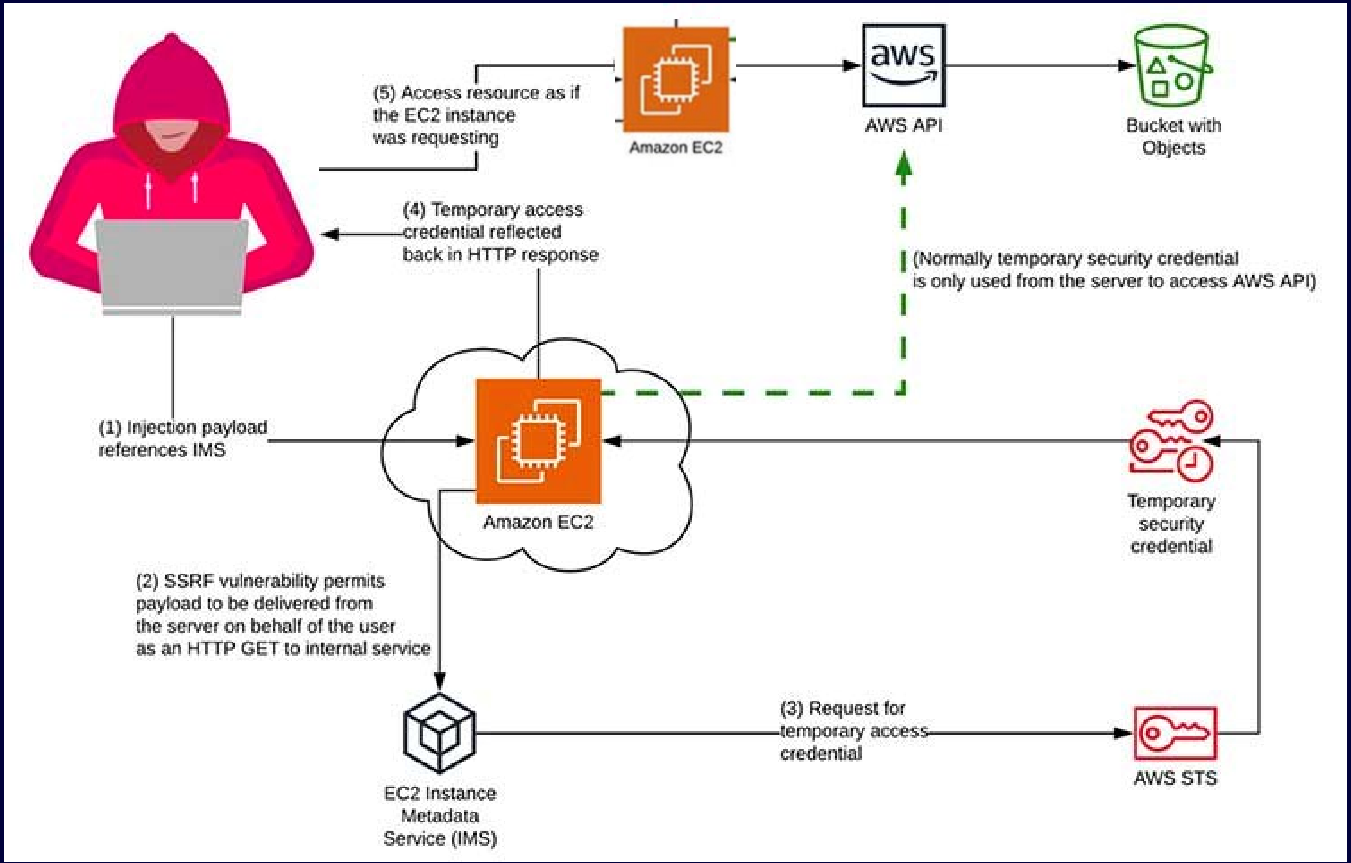
8 years Log4Shell vulnerability existed unnoticed

12 years before PwnKit was found and disclosed

52 days to fix security vulnerabilities

207 days on average to identify a breach

What does a breach look like?


Amazon EC2
AWS API
Bucket with Objects
Amazon EC2
Temporary security credential
EC2 Instance Metadata Service (IMS)
AWS STS

(1) Injection payload references IMS

(2) SSRF vulnerability permits payload to be delivered from the server on behalf of the user as an HTTP GET to internal service

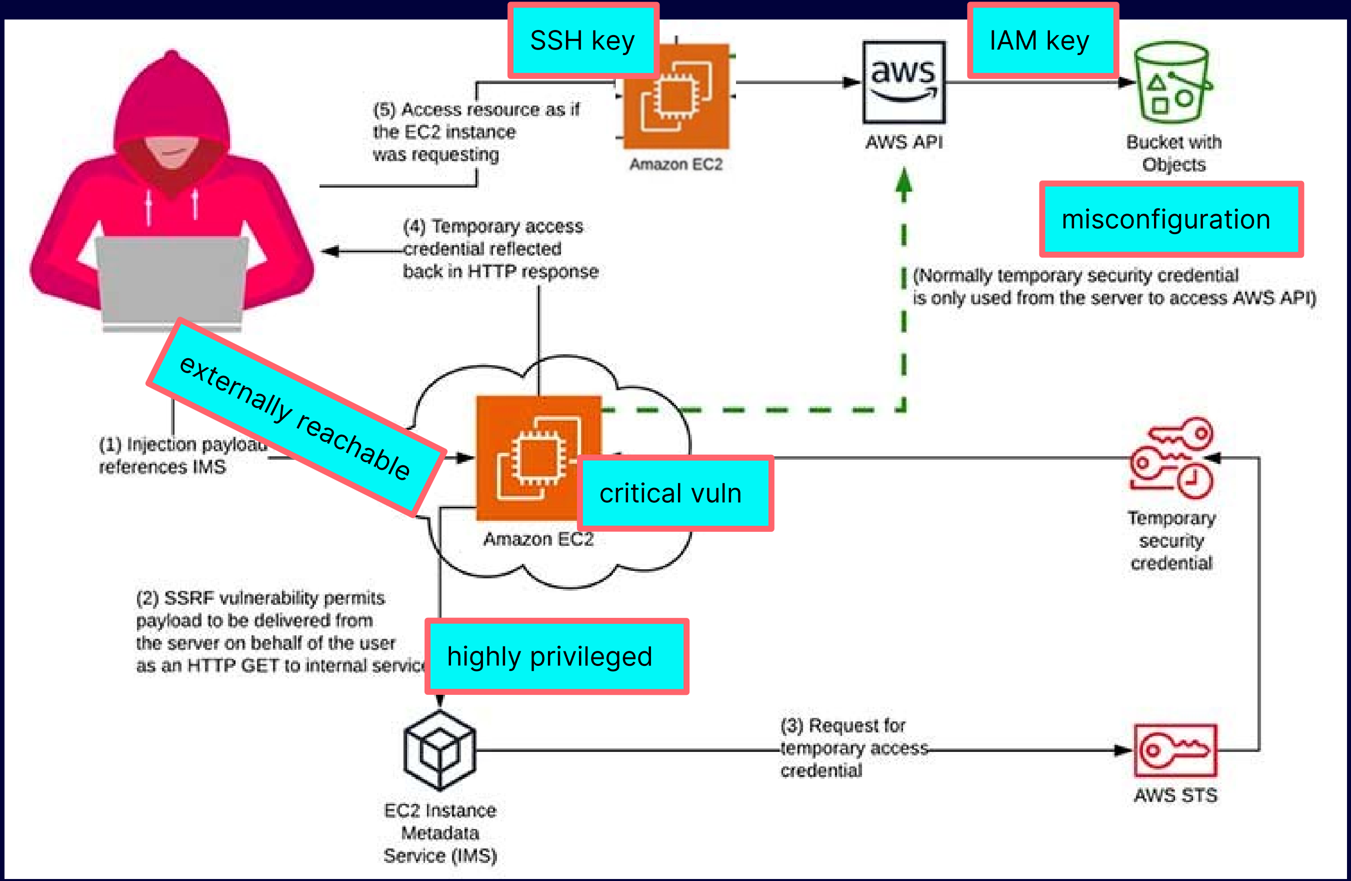
(3) Request for temporary access credential

(4) Temporary access credential reflected back in HTTP response

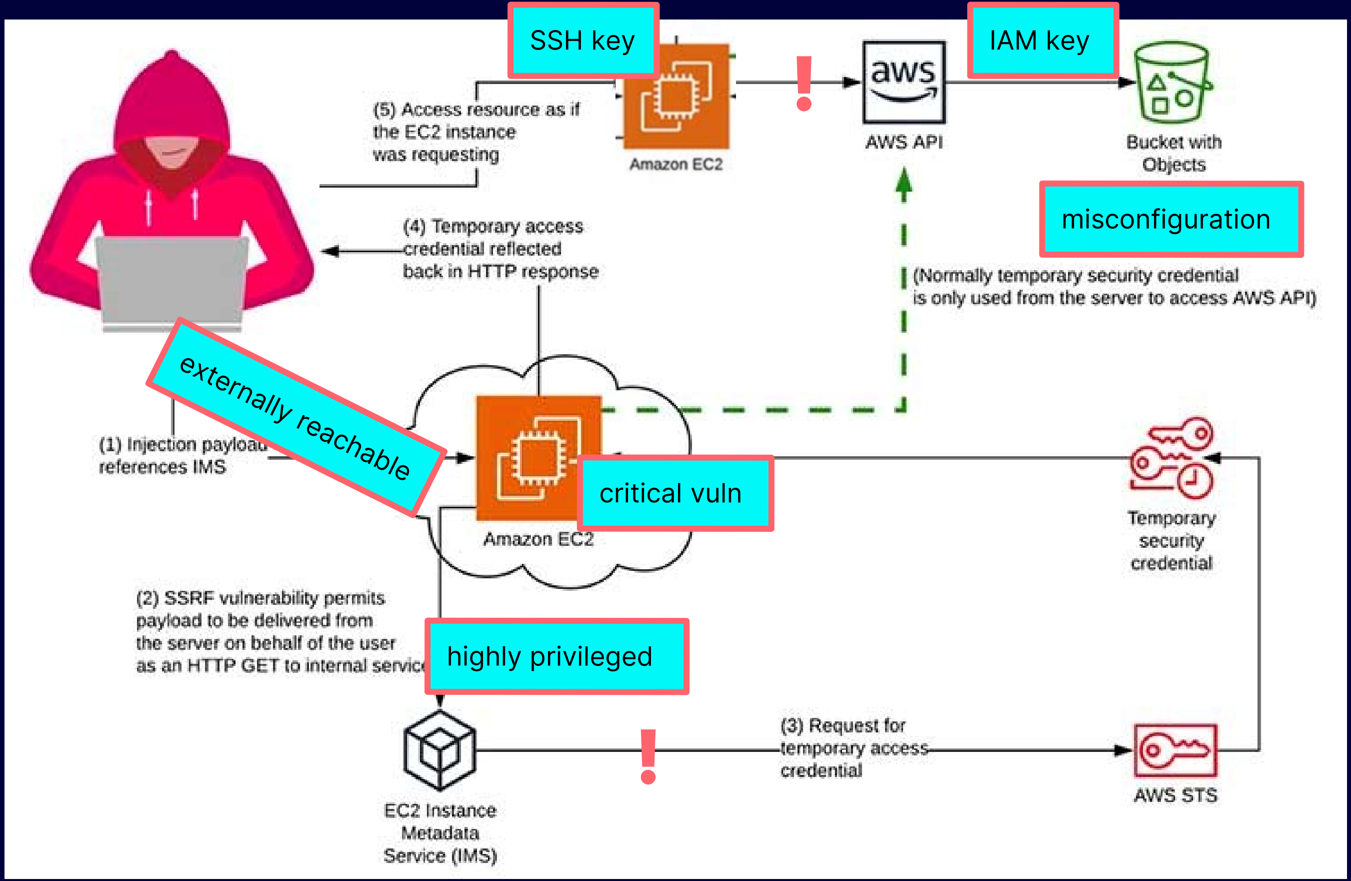
(5) Access resource as if the EC2 instance was requesting

(Normally temporary security credential is only used from the server to access AWS API)

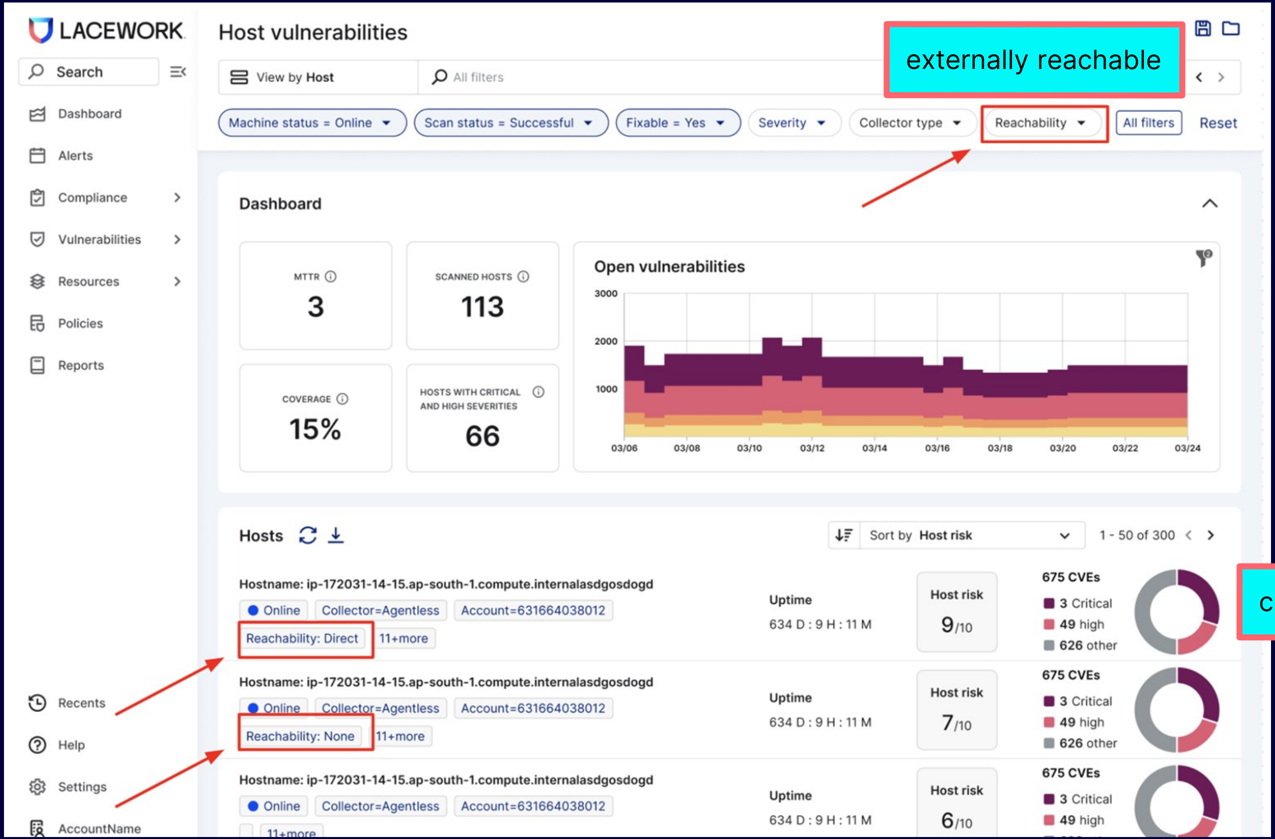
What does a breach look like?


SSH key
IAM key
Amazon EC2
aws
AWS API
Bucket with Objects
misconfiguration
(5) Access resource as if the EC2 instance was requesting
(4) Temporary access credential reflected back in HTTP response
(Normally temporary security credential is only used from the server to access AWS API)
externally reachable
(1) Injection payload references IMS
Amazon EC2
critical vuln
Temporary security credential
(2) SSRF vulnerability permits payload to be delivered from the server on behalf of the user as an HTTP GET to internal service
highly privileged
(3) Request for temporary access credential
AWS STS
EC2 Instance Metadata Service (IMS)

What does a breach look like?


SSH key
IAM key
(5) Access resource as if the EC2 instance was requesting
Amazon EC2
!
AWS API
Bucket with Objects
misconfiguration
(4) Temporary access credential reflected back in HTTP response
(Normally temporary security credential is only used from the server to access AWS API)
externally reachable
(1) Injection payload references IMS
Amazon EC2
critical vuln
Temporary security credential
(2) SSRF vulnerability permits payload to be delivered from the server on behalf of the user as an HTTP GET to internal service
highly privileged
!
(3) Request for temporary access credential
EC2 Instance Metadata Service (IMS)
AWS STS

Riskogestützes Vulnerability Management



LACEWORK

Search

- Dashboard
- Alerts
- Compliance
- Vulnerabilities
- Resources
- Policies
- Reports
- Recents
- Help
- Settings
- AccountName

Host vulnerabilities

View by Host | All filters

Machine status = Online ▾ | Scan status = Successful ▾ | Fixable = Yes ▾ | Severity ▾ | Collector type ▾ | Reachability ▾ | All filters | Reset

Dashboard

MTTR	SCANNED HOSTS
3	113

COVERAGE	HOSTS WITH CRITICAL AND HIGH SEVERITIES
15%	66

Open vulnerabilities

Hosts

Sort by Host risk ▾ | 1 - 50 of 300

Hostname: ip-172031-14-15.ap-south-1.compute.internalasdgosdogd
● Online | Collector=Agentless | Account=631664038012
Reachability: Direct | 11+more
Uptime 634 D : 9 H : 11 M
Host risk 9/10
675 CVEs
- 3 Critical
- 49 high
- 626 other

Hostname: ip-172031-14-15.ap-south-1.compute.internalasdgosdogd
● Online | Collector=Agentless | Account=631664038012
Reachability: None | 11+more
Uptime 634 D : 9 H : 11 M
Host risk 7/10
675 CVEs
- 3 Critical
- 49 high
- 626 other

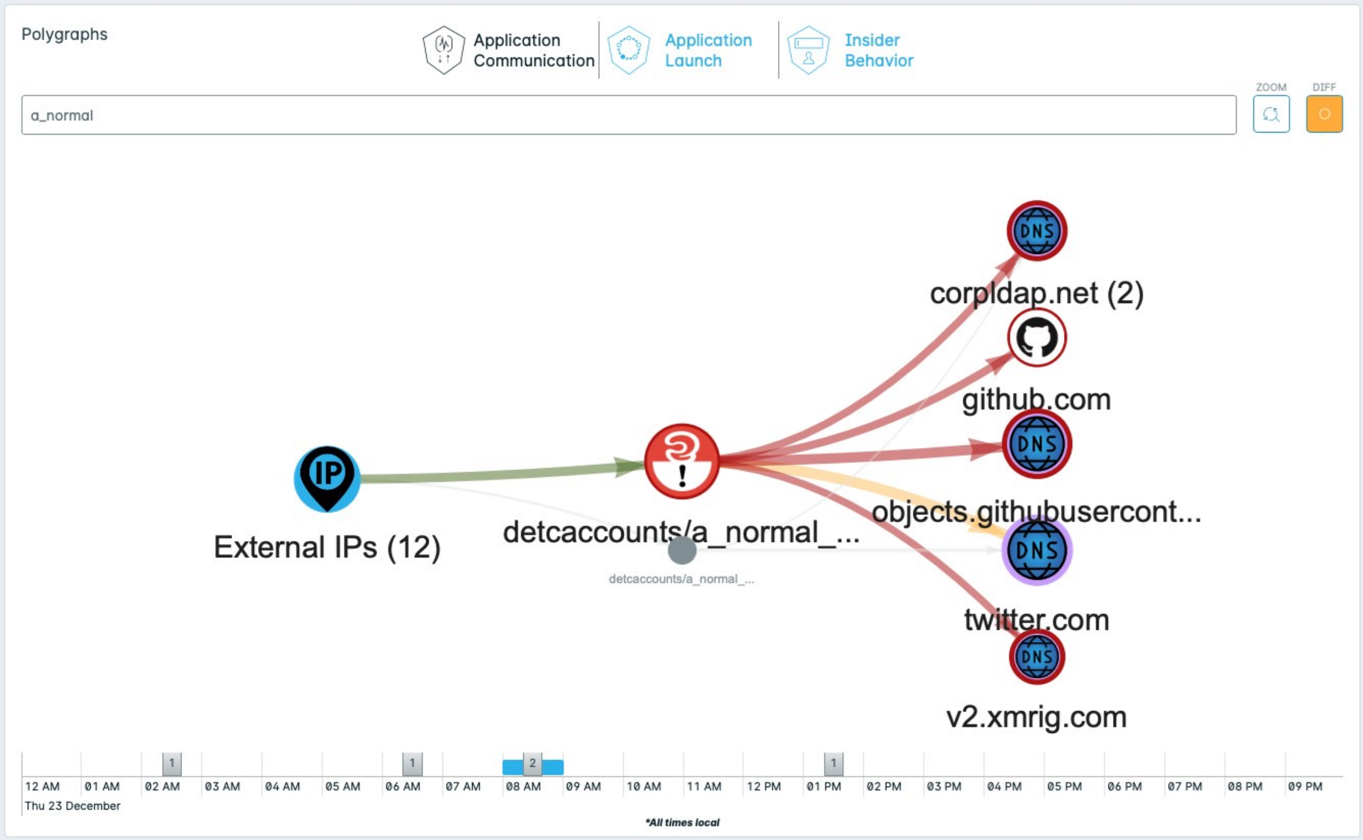
Hostname: ip-172031-14-15.ap-south-1.compute.internalasdgosdogd
● Online | Collector=Agentless | Account=631664038012
11+more
Uptime 634 D : 9 H : 11 M
Host risk 6/10
675 CVEs
- 3 Critical
- 49 high

externally reachable

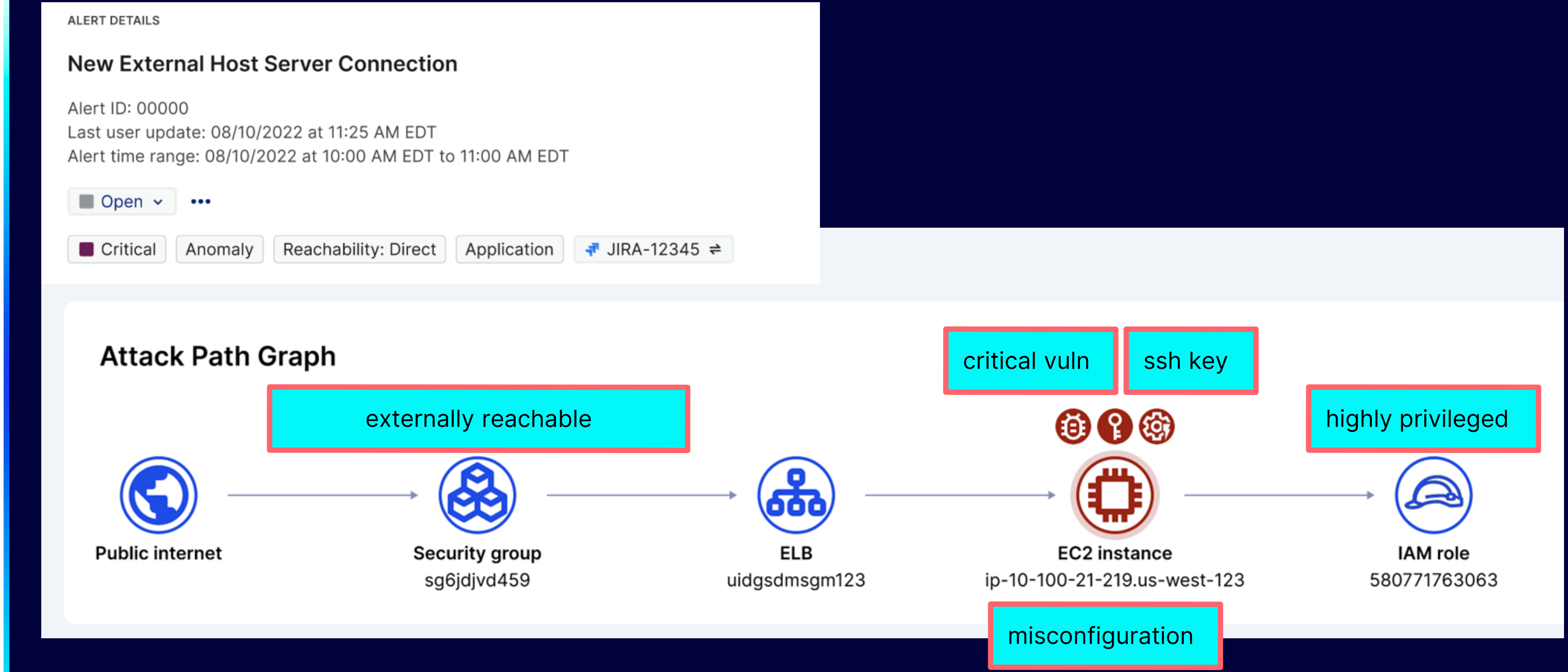
critical vuln

Regel-/Signaturunabhängige Anomalie-Erkennung(unsupervised ML)

Detect known and unknown threats with ease



Root Cause Analyse:
Anomalieerkennung + Angriffsoberflächenmanagement (CASM)



ALERT DETAILS

New External Host Server Connection

Alert ID: 00000
Last user update: 08/10/2022 at 11:25 AM EDT
Alert time range: 08/10/2022 at 10:00 AM EDT to 11:00 AM EDT

Open ⌄ •••

Critical | Anomaly | Reachability: Direct | Application | JIRA-12345 ⇄

Attack Path Graph

Cloud security is a data problem

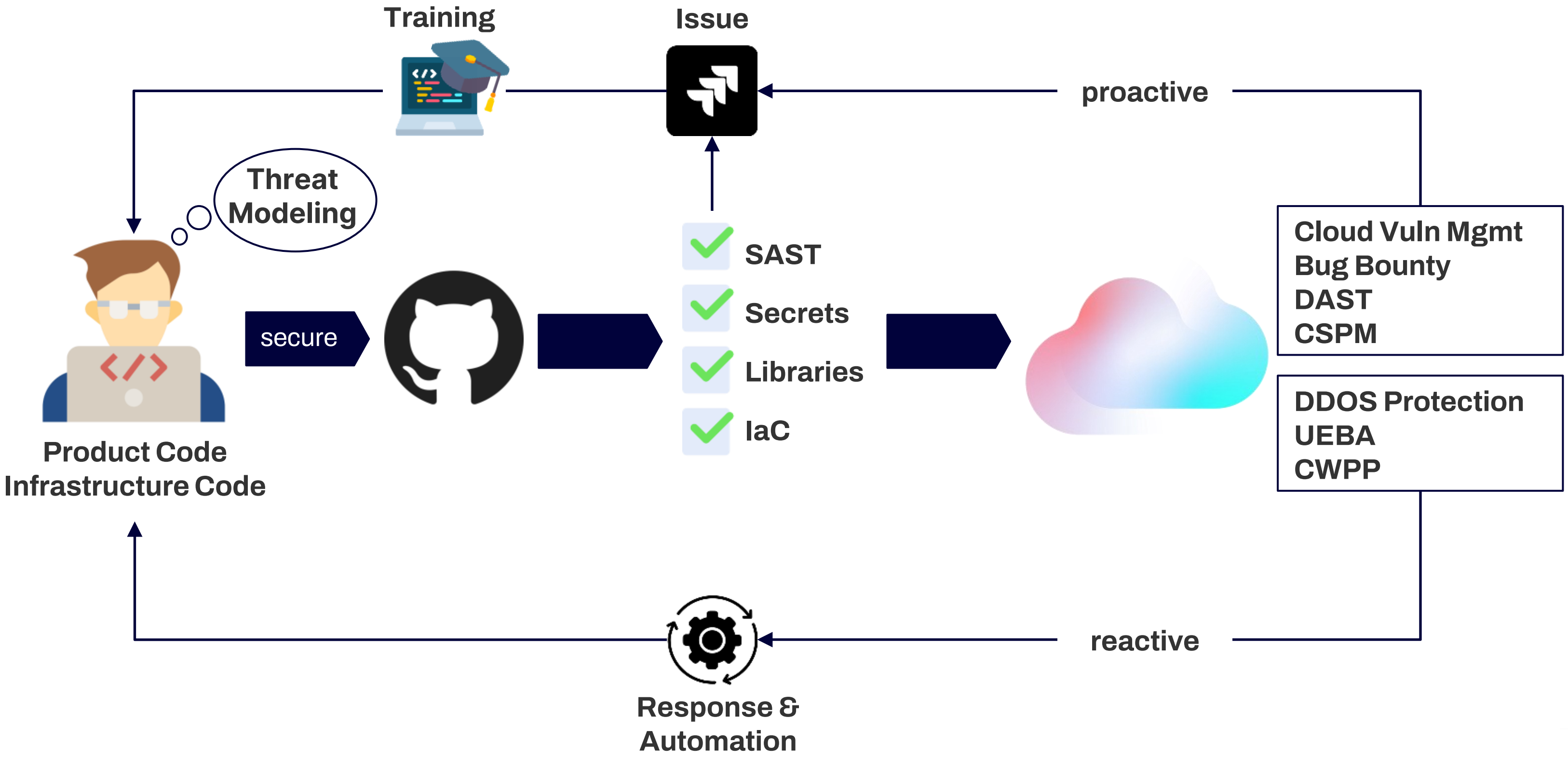
And it requires a fundamentally different approach

- Enormous scale
- Evolving technologies
- Constant changes
- Adaptive infrastructure
- Talent shortage

Massive amounts of data, hard to interpret at scale

Traditional way	New way
Rules-based	Behavioral-based
Watching the data	Using the data
Static	Dynamic, exponential scale
Too many alerts	Right alert, right time

Product & Cloud Security



To summarize

Lacework learns what’s normal and alerts on anomalies — leaving rules optional

Cloud security is a data problem

Massive amounts of data, hard to interpret at scale

Rule-based is for known security threats, but need context to prioritize

Use cases:

- Cloud Security Posture Management
- Risk-based Vulnerability Management

Behavioral analytics and anomaly detection

Detect unknown threats without writing a single rule

Vielen Dank!

https://www.linkedin.com/in/saschadubbel/