Security Operations

Threat Monitoring and Detection

Security Operations, often referred to as SecOps, involve the continuous monitoring and management of an organization's security posture. This function combines processes, tools, and skilled personnel to detect, investigate, and respond to security threats and incidents. The goal of SecOps is to protect an organization's digital assets, ensure compliance with regulatory requirements, and minimize the impact of security incidents.

Lecture at Technical University of Berlin

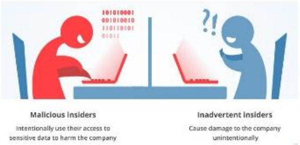
Jorge Cardoso
Chief Engineer for Hyperscale AIOps
Munich Research Center

2023.04.12

Security Operations
Threat Monitoring and Detection

PAIN POINT

Insider Threats can cause IT disruptions affecting productivity, profitability, and reputation



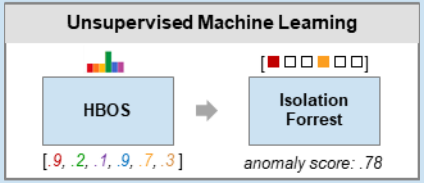
CLOUD PROVEIDERS

Problem

- Hyperscalers undergoes 1k+ changes per day
- Rapid detection of abnormal behaviors/operations is required to ensure live network security

INNOVATION

Combine rule-based and Machine Learning algorithms



Multi-dimensional behavior modeling

- 5WH model: 5W → What, Where, When, Why, Who; H → How
- Construct vectors for user activities and profile information at different granularity level

DESCRIPTION

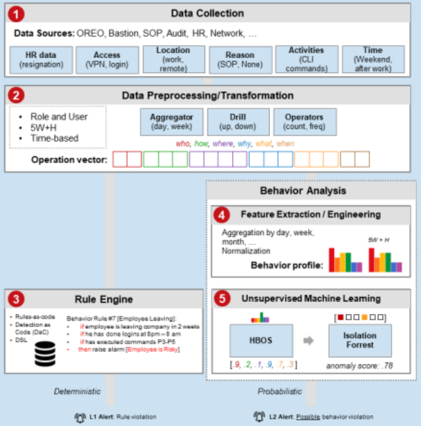
MAIN ACHIEVEMENT

System Design

- Literature Review of key approaches for Insider Thread detection
- Competitor Analysis
- System and Algorithm Design

HOW IT WORKS

1) **Input data:** SOP, Audit, HR, Network, IAM, logging
2) **Pre-processing:** Apply transformations to incoming data
3) **Rule engine:** Apply known expert rules
4) **Feature Extraction**
5) **Machine Learning:** Automatically detect abnormal behavior



ASSUMPTIONS & LIMITATIONS

- Proposal can generate too many false positives
- Filtering/noise reduction techniques need to be explored

TRL 2: *Technology formulation. Principles have been studied and practical applications can be developed based on initial findings*

IMPACT

Reduce the number of incidents

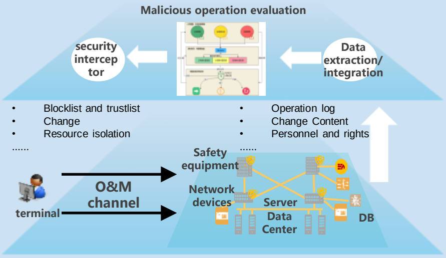
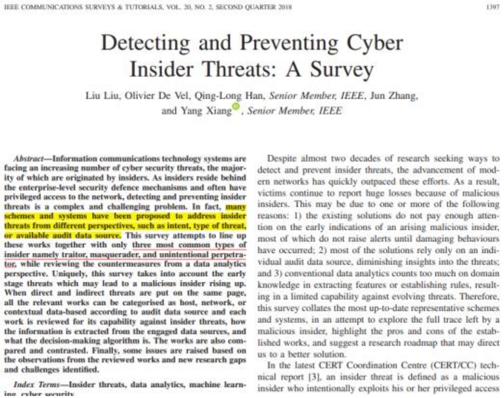


Fig. Automatically identify malicious operations so they can be intercepted



IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 20, NO. 2, SECOND QUARTER 2018 1397

Detecting and Preventing Cyber Insider Threats: A Survey

Liu Liu, Olivier De Vel, Qing-Long Han, *Senior Member, IEEE*, Jun Zhang, and Yang Xiang, *Senior Member, IEEE*

Abstract—Information communications technology systems are facing an increasing number of cyber security threats, the majority of which are originated by insiders. As insiders reside behind the enterprise-level security defence mechanisms and often have privileged access to the network, detecting and preventing insider threats is a complex and challenging problem. In fact, many schemes and systems have been proposed to address insider threats from different perspectives, such as intent, type of threat, or available audit data source. This survey attempts to line up these works together with only three most common types of insider namely traitor, masquerader, and unintentional perpetrator, while reviewing the countermeasures from a data analytics perspective. Uniquely, this survey takes into account the early stage threats which may lead to a malicious insider rising up. When direct and indirect threats are put on the same page, all the relevant works can be categorised as host, network, or contextual data-based according to audit data source and each work is reviewed for its capability against insider threats, how the information is extracted from the engaged data sources, and what the decision-making algorithm is. The works are also compared and contrasted. Finally, some issues are raised based on the observations from the reviewed works and new research gaps and challenges identified.

Index Terms—Insider threats, data analytics, machine learning, cyber security.

Despite almost two decades of research seeking ways to detect and prevent insider threats, the advancement of modern networks has quickly outpaced these efforts. As a result, victims continue to report huge losses because of malicious insiders. This may be due to one or more of the following reasons: 1) the existing solutions do not pay enough attention on the early indications of an arising malicious insider, most of which do not raise alerts until damaging behaviours have occurred; 2) most of the solutions rely only on an individual audit data source, diminishing insights into the threats; and 3) conventional data analytics counts too much on domain knowledge in extracting features or establishing rules, resulting in a limited capability against evolving threats. Therefore, this survey collates the most up-to-date representative schemes and systems, in an attempt to explore the full trace left by a malicious insider, highlight the pros and cons of the established works, and suggest a research roadmap that may direct us to a better solution.

In the latest CERT Coordination Centre (CERT/CC) technical report [3], an insider threat is defined as a malicious insider who intentionally exploits his or her privileged access

*Liu, Liu, et al. "**Detecting and preventing cyber insider threats: A survey**." IEEE Communications Surveys & Tutorials 20.2 (2018): 1397-1417.*

Background
Scope

What Is an Insider Threat?
Threats come from a trusted individual or privileged user who is authorized to access your organization's IT assets. It includes former employees; or outsiders, such as contractors and can damage IT in many ways:
- IT disruptions affecting productivity and profitability
- Reputational damage

Insider Types
- Malicious insiders: purposefully damage IT systems. They may be ex-employees or unhappy employees seeking revenge (unhappy employees, personal motive, espionage).
- ~~Negligent insiders: are those inside the organization who unintentionally misuse or abuse computer systems~~

Insider methods
- Execute authorized or unauthorized commands to damage IT systems
- ~~Exploitation of physical security vulnerabilities~~: where the attacker is physically inside the building or the data center
- ~~Script or program~~: development of a program or script, like a logic bomb, and then creating a backdoor account to be used later to initiate the script or include a time for activation in the script

Keywords
- User and Entity Behavioral Analytics (UEBA) solutions
- User Activity Monitoring (UAM) solutions
- Digital forensics, incident response, internal threat, insider threat, investigations

Challenges
- **Extremely Unbalanced Data**. Activities from insiders are extremely rare in real-world scenarios
- **Temporal Information in Attacks**. Many approaches focus on activity type, e.g., copying files to USB or browsing a Web page, incorporating temporal information is important and challenging. Copying files in working hours looks normal, but copying files at mid-night is suspicious
- **Heterogeneous Data Fusion**. temporal information, layoff, user profile (i.e., psychometric score), etc.
- **Subtle Attacks**. To evade detection, insider threats are subtle and hard to notice, which means that insiders and benign users are close in the feature space
- **Adaptive Threats**. Insiders always improve attacking strategies to evade detection. However, the learning-based models are unable to detect new types of attacks after training.
- **Fine-grained Detection**. Existing approaches usually detect malicious sessions that contain malicious activities. However, users usually conduct a large number of activities in a session. Hence, how to identify the fine-grained malicious subsequence or the exact malicious activity is important
- **Early Detection**. Approaches focus on insider threat detection, which means malicious activities already occur and the significant loss is already caused to organizations, Prediction is also important
- **Lack of Testbed**. No real-world dataset that is publicly-available.

Requirements
Examples Provided by HQ

1

On February 23, Weimob, a leading domestic SaaS service provider, suffered a vicious deletion incident, and it was not until March 1 that the data was fully retrieved, and various businesses gradually returned to normal.

This incident exposed the problem that Chinese enterprises have long emphasized business support and implementation in IT construction, while paying attention to and investing in support systems such as security operation and maintenance, disaster preparedness and disaster recovery, which has caused a great shock to the industry.

微盟事件时间线

2. 23晚7-8点	• 微盟系统崩溃，基于微盟的商家小程序都宕机，无法打开。商家的线上生意基本停摆。
2月24日晚间	• 微盟官方发布公告，表示正在紧急修复中，服务恢复预计还需要24-48小时 • 腾讯云也给出紧急回复：微盟运维事故发生后，腾讯云的技术团队已经在第一时间与微盟对齐，研究制定修复方案。工程师们正在日夜赶工，将尽最大努力协助微盟降低损失。
2月25日	• 截至2月25日早上7点，我们的生产环境和数据修复在有序推进，我们预计2月25日晚上24点前微盟集团的生产环境将修复完成，微盟所有新用户将可恢复服务；老用户由于数据修复时间问题，微盟集团将提供临时过渡方案，预计老用户数据修复将可在2月28日晚上24点前完成。
2月26日	• 服务恢复。微盟新用户，可以直接注册开通使用；微盟老用户，重新注册一个账户，待数据恢复后合并数据。
2月28日	微盟2020.2.28公告：微盟所有业务恢复服务 数据恢复进展顺利。.jpg
3月01日	微盟数据已经全面找回 并公布商家赔付计划

https://cloud.tencent.com/developer/article/1862390?from=15425

2

PRESS RELEASE

San Jose Man Pleads Guilty To Damaging Cisco's Network

Wednesday, August 26, 2020

Share

For Immediate Release

U.S. Attorney's Office, Northern District of California

Unauthorized Access Led to Deletion of 16,000 WebEx Teams Accounts in the Fall of 2018

SAN JOSE – Sudhish Kasaba Ramesh pleaded guilty in federal court in San Jose today to intentionally accessing a protected computer without authorization and recklessly causing damage, announced United States Attorney David L. Anderson and Federal Bureau of Investigation Special Agent in Charge John L. Bennett.

According to the plea agreement, Ramesh admitted to intentionally accessing Cisco Systems' cloud infrastructure that was hosted by Amazon Web Services without Cisco's permission on September 24, 2018. Ramesh worked for Cisco and resigned in approximately April 2018. During his unauthorized access, Ramesh admitted that he deployed a code from his Google Cloud Project account that resulted in the deletion of 456 virtual machines for Cisco's WebEx Teams application, which provided video meetings, video messaging, file sharing, and other collaboration tools. He further admitted that he acted recklessly in deploying the code, and consciously disregarded the substantial risk that his conduct could harm to Cisco. As a result of Ramesh's conduct, over 16,000 WebEx Teams accounts were shut down for up to two weeks, and caused Cisco to spend approximately $1,400,000 in employee time to restore the damage to the application and refund over $1,000,000 to affected customers. No customer data was compromised as a result of the defendant's conduct.

https://www.justice.gov/usao-ndca/pr/san-jose-man-pleads-guilty-damaging-cisco-s-network

3

… the former database administrator of Lianjia Network Technology Co., Ltd., was sentenced […] for the crime of damaging computer information systems […].

…Han Bing, the database administrator of Lianjia.com (Beijing) Technology Co., Ltd., used his "root" permission of the company's financial system to log in to the company's financial system and delete the financial data and related applications in the system, making the company's financial system completely inaccessible […]

链家程序员“删库”9TB数据被判7年

鞭牛士Bianews 关注
2022年05月15日 07:37:13 来自北京 2人参与 1评论

鞭牛士 5月15日消息，据中国裁判文书网消息，原链家网（北京）科技有限公司数据库管理员韩冰，因犯破坏计算机信息系统罪一审被判处有期徒刑七年，二审维持原判。

据悉，在2018年6月4日，链家网（北京）科技有限公司数据库管理员韩冰利用其担任并掌握该公司财务系统“root”权限的便利，登录该公司财务系统，并将系统内的财务数据及相关应用程序删除，致使该公司财务系统彻底无法访问。

被破坏的服务器是公司专门用于EBS系统的2台数据库服务器和2台应用服务器，存放着公司成立以来所有的财务数据，直接影响公司人员的工资发放等，对公司整个运行有非常重要的意义。

该公司恢复数据及重新构建该系统共计花费人民币18万元。

https://tech.ifeng.com/c/8G1klaVX7pY

Source https://clouddevops.huawei.com/domains/29680/wiki/1/WIKI2022092000894

Literature Review
Frequency-based methods

Main idea
- Use the frequencies of system calls to characterize user behavior

TABLE II
TAXONOMY OF SYSTEM CALL BASED ANALYTICS

Threat type	Model	Tech category	Algorithm	Data source
Intrusion & malware	n-gram	statistical	sequence match [51] [52]	System call sequence
	n-gram	statistical	Markov [59] [60]	
	n-gram	machine learning	feedforward neural network [61] [62]	
	n-gram	deep learning	recurrent neural network (RNN) [63]	
	frequency	statistical	LLRT, LR [64]	
	frequency	machine learning	kNN [65] [66]	
		machine learning	kMC [67]	
		machine learning	SVM [68] [64]	
Insider threats		rule	signature match [30]	System call parameter
		graph	minimum description length (MDL) [53]	

Technique
- Sequences of system calls are transformed into a fixed-length frequency vector according to the occurrence number of system calls
- Apply, e.g., k-nearest neighbour (kNN), k-means clustering or support vector machine (SVM) to identify anomalous frequency vectors
- Use tf-idf weighting to encode system calls $$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{l=1}^{M} f_{lj}^2}} \times log\left(\frac{N}{n_i}\right)$$
- 1) Use only normal vector: if unknown frequency vector has a high similarity using cosine distance to a normal vector, mark the sequence as normal
- 2) Use normal and abnormal vectors: use the same distance function as with 1)

TF-IDF Technique: Text Processing Metaphor [1]
- Each system call is treated as a "word" of a document
- Set of system calls generated by a process is treated as a "document"
- Use text processing methods for intrusion detection problem, e.g., k-nearest neighbor classification method

Table 1: Analogy between text categorization and intrusion detection when applying the kNN classifier.

Terms	Text categorization	Intrusion Detection
N	total number of documents	total number of processes
M	total number of distinct words	total number of distinct system calls
n_i	number of times ith word occurs	number of times ith system call was issued
f_{ij}	frequency of ith word in document j	frequency of ith system call in process j
D_j	jth training document	jth training process
X	test document	test process

The DARPA data was labeled with session numbers. Each session corresponds to a TCP/IPconnection between two computers. Individual sessions can be programmatically extracted fromthe BSM audit data. Each session consists of one or more processes. A complete ordered list ofsystem calls is generated for every process. A sample system call list is shown below. The firstsystem call issued by Process 994 was *close*, *execve* was the next, then *open*, *mmap*, *open* and soon. The process ended with the system call *exit*.

Sample system call list: The first system call issued by Process 994 was close, execve was the next, then open, mmap, open and soon. The process ended with the system call exit.

Process ID: 994

close	*execve*	*open*	*mmap*	*open*	*mmap*	*mmap*	*munmap*	*mmap*
mmap	*close*	*open*	*mmap*	*close*	*open*	*mmap*	*mmap*	*munmap*
mmap	*close*	*close*	*munmap*	*open*	*ioctl*	*access*	*chown*	*ioctl*
access	*chmod*	*close*	*close*	*close*	*close*	*close*	*exit*	

[1] Use of K-Nearest Neighbor classifier for intrusion detection

Literature Review
Frequency-based methods: Practical Example

Data, Events

DataLake, databases, tables

HR

Empl_Id	Qualification
J00760260	Class-2
X00123456	Class-1

Change Orders

Order_Id	Class	Empl_Id
OID111	Class-B	J00760260
OID112	Class-A	X00123456

SOP

Order_Id	Command
OID111	ps -ef
OID111	rm /tmp/*.yaml

OREO

Order_Id	Command
OID111	ps –ef
OID111	kill -9 123

1

Data

Rules

Business Rule

Use Case 1. Authorized Insider

Internal operator executes commands with a change order

MITRE ATT&CK Framework:

- https://attack.mitre.org/techniques/T1059/004/
- https://attack.mitre.org/datasources/DS0017/

Description

- Huawei employees with Class-2 operation qualification (the highest qualification) uses Class-B change orders to escalate rights to the bastion host, and then exploit the bastion host vulnerabilities to execute Commands (e.g., rm -rf XX) that are <u>not described in the change guide</u>.

2

Technical Rule (e.g., KQL)

```
HR
| where Qualification == "Class-2"
| project Empl_Id
| join (Change_Orders
        | where Class == "Class-B"
        | project Order_Id)
        on Empl_id
| join (SOP
        | project Cmd)
        on Order_Id
| join (OREO
        | project Order_Id, Cmd)
        kind = rightanti on Cmd, Order_Id
| summarize NumCmd=count() by Order_Id
```

Results

3

Results

Order_Id	NumCmd
OID111	1

[1] MITRE ATT&CK Framework: https://attack.mitre.org
https://clouddevops.huawei.com/domains/29680/wiki/1/WIKI2022112101100

Microsoft Excel Worksheet

Literature Review
Sequence-based methods

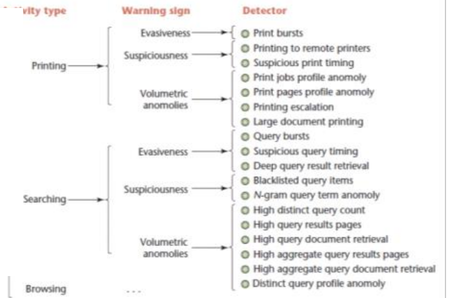
Main idea
- Host-based analytics / behavioral analysis: model the sequences of user actions and employ them to detect unusual sequences
- Analyze system calls (OS), shell command lines (application-level), keystroke/mouse dynamics, *nix syslog, Windows logs, etc.
- Data captures how a host behaves and the human interactive behavior with the host

Techniques
- Hidden Markov Model (HMM). Rashid et. al. [1]
  - 7 event types; Training: 5 weeks
  - Logs: login/logoff, web access, USB connection, and email
- RNN autoencoder model. Ha et. al. (same method as Rashid et. al.) [1]
  - Split sequences into small fixed-size sequences
  - Model clearly outperforms HMM-based models
- Stacked LSTM models. Tuor et. al. [1]
  - 408 event types (usage time, email attachments, file operations (e.g., reading, writing, copying, and deletion)
- LSTM models and CNN models. Yuan et. al. [1]
  - 16 events types
  - LSTM with CNN model: AUC = 0.9449 (best)
- Seq2seq learning model. Jang, et al. [1]
  - 7 event types; Training: 60 days
  - Use attention mechanism
  - Introduce standard deviation factors to estimate each event's global characteristics
- Bayesian Networks. Caputo [3]
  - Monitors user activities and indicating malicious activities

TABLE II
TAXONOMY OF SYSTEM CALL BASED ANALYTICS

Threat type	Model	Tech category	Algorithm	Data source
Intrusion & malware	n-gram	statistical	sequence match [51] [52]	System call sequence
	n-gram	statistical	Markov [59] [60]	
	n-gram	machine learning	feedforward neural network [61] [62]	
	n-gram	deep learning	recurrent neural network (RNN) [63]	
	frequency	statistical	LLRT, LR [64]	
	frequency	machine learning	kNN [65] [66]	
		machine learning	kMC [67]	
		machine learning	SVM [68] [64]	
Insider threats		rule	signature match [30]	System call parameter
		graph	minimum description length (MDL) [53]	

Elicit detector [3]
- Examine how users manipulate Web information
- Use electronic records: location, job title, projects
- Collected **284 days of data**, from **3,900 users**, and produced **91 million events**
- Research team consulted with three subject-matter experts to develop **76 detectors**
- Implemented detectors using rules and statistical methods. E.g., a rule issues an alert if an individual used a printer other than the one closest to his or her office



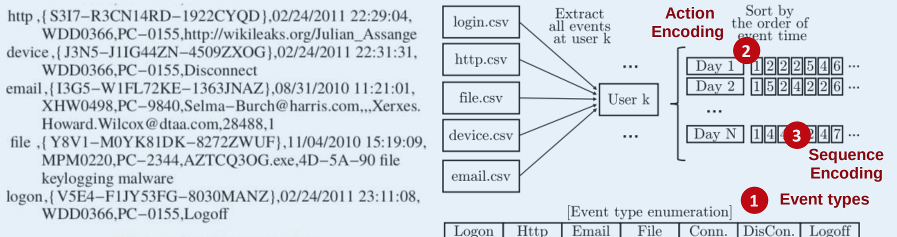


Fig. 1 Examples on input logs

[1]

[1] Against Insider Threats with Hybrid Anomaly Detection with Local-Feature Autoencoder and Global Statistics (LAGS)
[2] Detecting and Preventing Cyber Insider Threats: A Survey
[3] Detecting insider theft of trade secrets, D. Caputo, et. al.

Literature Review
Graph-based methods

HR Data
- Contextual information regarding a human user such as HR and psychological data
- Available from an employee directory or a specific ERP
- e.g., type of employment, remaining years of contract, remaining days of leave, job title, salary range, participated projects, business travel records, performance review, etc.

TABLE VII
TAXONOMY OF CONTEXTUAL DATA BASED ANALYTICS

Threat Type	Tech category	Algorithm	Data source
Insider threats	rule-based	signature match [122]	HR, network
	statistical	KDE [122]	
	graph-based	bipartite graph [123]	HR, host
	conceptual framework	NA [124]	HR, psychological, host, network
	factor analysis	scoring [10]	psychological, host
	graph-based	SAD [11]	psychological, network
	machine learning	Bayesian [11]	

Bipartite graph analysis [1] / Peer Analysis
- Establish a pattern of acceptable actions based on workgroup role classifications
- Provides a means to identify and determine normal or expected behavior for workgroups

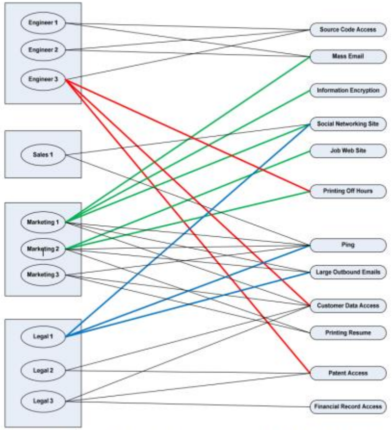


Figure 2: Visualization of individuals, workgroups and precursors showing normal and precursor behavior.

Identifying Threats Using Graph-based Anomaly Detection [2]
- 1) Find normative patterns in the data using graph-based data mining and
- 2) Searching for small, unexpected deviations to normative patterns
- 3) Illicit behavior tries to mimic legitimate, normative behavior

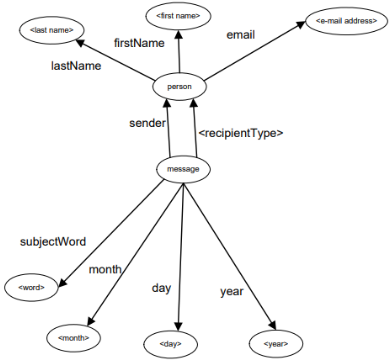


Figure 5. Graphical representation of Enron e-mail.

[1] Detecting and Preventing Cyber Insider Threats: A Survey
[2] Identifying threats using graph-based anomaly detection, W. Eberle

Literature Review
Rule-based methods

Main idea
- Automatically extract rules from instances
- Generalize system calls sequence using a set of rules

Techniques
- Use a sliding window (length 11) to create sequences of consecutive system calls
- Create two sets of system call data: normal and abnormal
- Scan intrusion traces and look it up in the “normal” list
- If a match can be found then the sequence is labeled as “normal”. Otherwise it is labeled as “abnormal”
- Applied RIPPER (Cohen 1995), a rule learning program, to generate concise rule sets from training data
- RIPPER rules can be used to predict whether a sequence is “abnormal“ or “normal “
- Use a sliding window of length to scan the predictions made by RIPPER
- If the percentage of “abnormal” predictions is above a threshold value, then the trace is an intrusion

1

Example of mapping between pid and sequences of system calls

pids	282 282 …	291 291…
system calls	4 2 66 66 4 138 66 5 23 45 4 27 …	155 104 106 105 104 104 106 56 19 155 83 155 …

Table 1. System Call Data. Each file has two columns, the pids and the system call numbers.

2

Example labelled sequences of system calls

System Call Sequences (length 11)	Class Labels
4 2 66 66 4 138 66 5 23 45 4	“normal”
…	…
104 106 105 104 104 106 56 19 155 83 155	“abnormal”
…	…

Table 2. Pre-processed System Call Data. System call sequences of length 11 are labeled as “normal” or “abnormal”.

3

Example of a rule extracted

[covers 84 positive and 0 negative examples; here positive=”abnormal” and negative=”normal”]
abnormal:- p_5=’112’, p1=’112’, p3=’128’.
[meaning: if p_5 and p1 are 112 (*vtrace*) and p3 is 128 (*flock*) then the sequence is “abnormal”]

[covers 75 positive and 0 negative examples]
abnormal:- p0=’128’, p2=’112’.
[meaning: if p0 is 128 and p2 is 112 then the sequence is “abnormal”]
…
[covers 4188 positive and 0 negative examples; here positive=”normal” and negative=”abnormal”]
normal:- true.
[meaning: if none of the above, the sequence is “normal”]

[1] Learning Patterns from Unix Process Execution Traces for Intrusion Detection, Wenke Lee, et. al.

Technology Review

Web Articles

1

Google

Cyber Threats | Insider Threat

Insider Threat to Google as it fires 36 employees in 2020

By Naveen Goud 1565

How do you detect insider threats? “here are some of the things Google detection teams typically look for” [1]:

- Users from one department or job role accessing or attempting to access data from another in a suspicious way
- Large file transfers by employees around the time they're **leaving a company**
- **Accesses to services like Dropbox** that the company doesn't usually use
- Employees with privileged access using that access **much more often than their peers**
- Unusual login activity that might indicate someone using someone else's password or computer
- Data leaving the company that shouldn't be, detected by packet inspection (DLP)
- Employees reporting that their coworker is acting suspiciously

How do big tech companies minimize the risk of cyber insider threats?

- Elevated privileges need to be specifically requested for each use. Ideally each request should have to be approved by another person.
- Elevated actions are monitored for suspicious behavior, like someone requesting access to hundreds of accounts when usually they access one.
- Making it very visible when a person uses elevated permissions, like sending an email to the team.
- Reducing the number of people who need elevated permissions and reducing the permissions they need. For example, if someone needs to view the customer's address, give them access only to the address and not the rest of the customer's information.
- Locking down document permissions

2

aws

Software Development Engineer, Security Analytics and AI Research

Amazon Web Services (AWS) · New York, NY 2 weeks ago · 9 applicants

- $124,000/yr - $181,000/yr + Sign-on bonus, Stock (LinkedIn est.) · Full-time · Mid-Senior level
- 10,001+ employees · IT Services and IT Consulting
- 12 connections · 2 company alumni · 122 school alumni
- See how you compare to 9 applicants. Retry Premium Free
- Actively recruiting

Apply | Save

About the job

Job Summary

DESCRIPTION

Amazon Web Services is looking for experienced software developers to join the Security Analytics and AI Research group within AWS Security Services. This team is entrusted with researching and developing core data mining and machine learning algorithms for various AWS security services like GuardDuty (https://aws.amazon.com/guardduty/) and Macie (https://aws.amazon.com/macie/). On this team, you will invent and implement innovative solutions for never-before-solved problems. If you have a passion for security and experience with large scale machine learning problems, this will be an exciting opportunity.

The AWS Security Services team builds technologies that help customers strengthen their security posture and better meet security requirements in the AWS Cloud. The team interacts with security researchers to codify our own learnings and best practices and make them available for customers. We are building massively scalable and globally distributed security systems to po

Key Responsibilities

- Collaborate with scientists to inte services.
- Build complex systems that turn m customers.
- Rapidly design and conduct large s quantitative and business judgmen

Amazon GuardDuty

Protect your AWS accounts with intelligent threat detection

Create an AWS Account

Start your 30-day free trial

with the AWS Free Tier

Continuously monitor your AWS accounts, instances, container workloads, users, and storage for potential threats.

Expose threats quickly using anomaly detection, machine learning, behavioral modeling, and threat intelligence feeds from AWS and leading third-parties.

Mitigate threats early by initiating automated responses.

How it works

Amazon GuardDuty is a threat detection service that continuously monitors your AWS accounts and workloads for malicious activity and delivers detailed security findings for visibility and remediation.

3

Insider threats
Mitigation Guidelines

Carnegie Mellon University | Enter

Software Engineering Institute

About | Our Work | Publications | News and Events

SEI › Publications › Digital Library › CERT Insider Threat Center

CERT Insider Threat Center

NOVEMBER 2017 • BROCHURE

By CERT Insider Threat Center

CERT Insider Threat Center and the U.S. National Insider Threat Task Force have issued common guidelines to help prevent and mitigate insider threats in organizational environments [4], [6]. The guidelines in [4] describes 20 practices that organizations should implement across the enterprise to prevent and detect insider threats, as well as case studies of organizations that failed to do so.

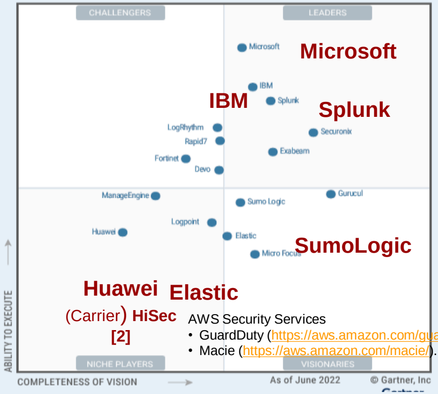
- [4] M. L. Collins et al., “Common sense guide to mitigating insider threats, fifth edition,” CERT Insider Threat Center, Carnegie Mellon Univ., Pittsburgh, PA, USA, Rep. CMU/SEI-2015-TR-010, 2016.
- [6] “Combating the insider threat,” Nat. Cybersecurity Commun. Integr. Center, U.S. Dept. Homeland Security, Washington, DC, USA, Rep., 2014. Accessed: Jun. 15, 2019. [Online]. Available: https://www.uscert.gov/sites/default/files/publications/Combating%20the%20Insider%20Threat_0.pdf

[1] https://www.cybersecurity-insiders.com/insider-threat-to-google-as-it-fires-36-employees-in-2020/
[2] David Seidman, Sr Director of Detection @ Salesforce, Head of Detection and Response @ Robinhood, Security Engineering Manager @ Google

Technology Review
Systems and Trends

Approaches

- **SIEM** (Security Information and Event Management): Collects security events. Analyze, investigate, correlate events often using logical rules. Create reports and dashboards
- **UEBA** (User and Entity Behavior Analytics): Uses large datasets to model typical and atypical behaviors of humans and machines by means of machine learning algorithms
- **SOAR** (Security Orchestration, Automation and Response): acts as the remediation and response engine to those alerts



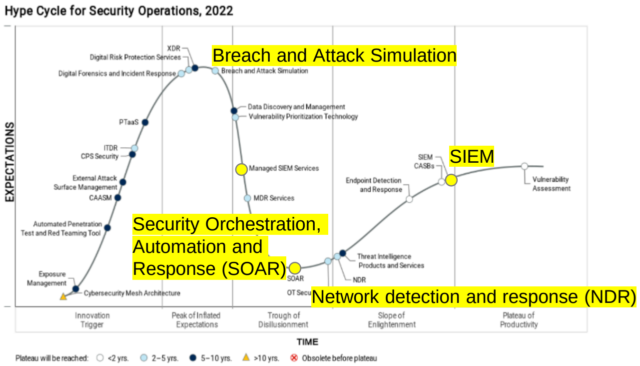
(Carrier) HiSec [2]

AWS Security Services
- GuardDuty (https://aws.amazon.com/guardduty/)
- Macie (https://aws.amazon.com/macie/)

Trends

Trends related to man-malicious operations

SIEM → UEBA → SOAR



Systems

Splunk splunk>

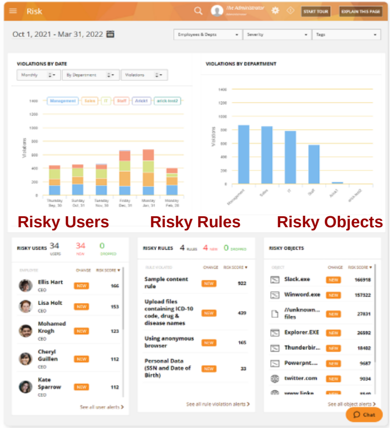
Splunk is not a cloud-native SaaS solution. It was designed as an on-prem solution that the company later moved to the cloud. While its SaaS solution—Splunk Cloud—is growing, it's a lifted-and-shifted architecture.

Microsoft Sentinel Azure

Sentinel is a cloud-native SaaS solution. Built on Azure, Sentinel was designed to live in the cloud and overcome many of the challenges of on-prem solutions. But Sentinel can only be deployed in Azure.

Google Chronicle Google

Chronicle is also a cloud-native SaaS solution, built on top of Google's public cloud. It's the least mature offering in this list, but does scale well. It can only be deployed in GCP.



Limitations

SIEM History
- SIEMs have been deployed in security operations centers (SOC) for 15 years
- Originally designed as log collectors, SIEMs expanded capabilities to include detection methods (e.g., UEBA), compliance monitoring, …

SIEM/UEBA Challenges
- Complex and expensive to deploy
  - Use simpler rule languages (c.f., SQL is too complex)
- Generates large amounts of false positives
  - Statistical and ML-based filtering/noise reduction is needed
- Requires technical expertise
  - Enable communities to create/manage their own rules
  - Use ML to automatically identify potential malicious attacks

[1] https://luminaanalytics.com/insider-threat/
[2] https://carrier.huawei.com/en/products/fixed-network/sub-solution-data-communication/HiSec

Technology Review

User and Entity Behavior Analytics (UEBA)

Definition: UEBA

- Cybersecurity solution that uses algorithms and machine learning to detect anomalies in the behavior
- Recognize unusual or suspicious behavior which diverges from normal everyday patterns or usage
- For example, if a user regularly downloads files of 20 MB every day but starts downloading 4 GB of files, the UEBA system would consider this an anomaly and either alert an IT administrator

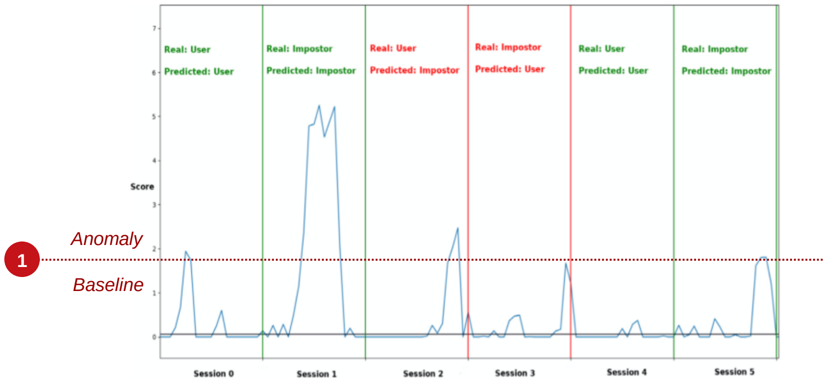
UEBA vs SIEM

- UEBA use machine learning while SIEM uses rule-cased detection
- UEBA creates models of normal behavior for individual users or components such as IP addresses, servers, applications by means of statistical analysis or learning methods in order to detect deviations.
- The model is used to detect abnormalities
- e.g. IBM QRadar UBA App, LogRhythm UEBA, ArcSight UBA, DarkTrace Enterprise

SIEM Limitations

First SIEM generation uses fix rules and regulations → too inflexible/costly

- Static rules maintained by the provider that compare against dynamic lists of suspicious objects (e.g. IP addresses, URLs, hashes of binary code)
- Rules and dynamic lists are renewed during updates, e.g. monthly
- e.g. IBM QRadar SIEM, Tenable LCE and McAfee Enterprise Security SIEM



[1] **Fig. 4 – Excerpt of the buffer values for a specific user (keystroke dynamics).**

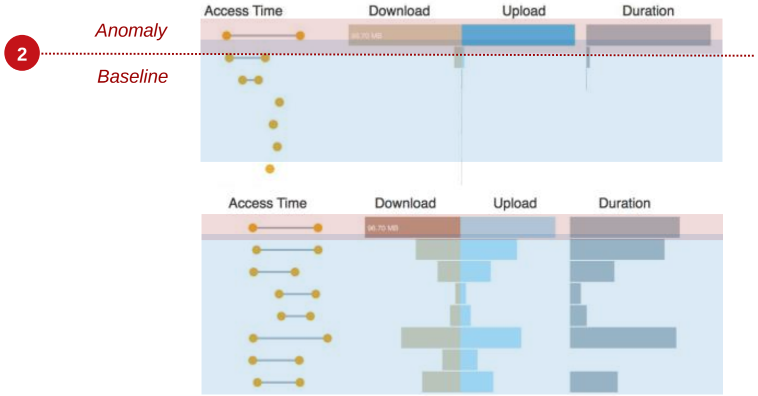


Fig. Examples of baselines and anomalous instances

[1] An approach to detect user behaviour anomalies within identity federations

Technology Review

UEBA: Microsoft & others

Machine learning (ML) behavioral analytics (1)

- Templates based on proprietary Microsoft machine learning algorithms
- Not possible to see the internal logic of how they work

Rule type	Description
Microsoft security	Microsoft security templates automatically create Microsoft Sentinel incidents from the alerts generated in other Microsoft security solutions, in real time. You can use Microsoft security rules as a template to create new rules with similar logic. For more information about security rules, see Automatically create incidents from Microsoft security alerts.
Fusion (some detections in Preview)	Microsoft Sentinel uses the Fusion correlation engine, with its scalable machine learning algorithms, to detect advanced multistage attacks by correlating many low-fidelity alerts and events across multiple products into high-fidelity and actionable incidents. Fusion is enabled by default. Because the logic is hidden and therefore not customizable, you can only create one rule with this template. The Fusion engine can also correlate alerts produced by scheduled analytics rules with those from other systems, producing high-fidelity incidents as a result.
Machine learning (ML) behavioral analytics	ML behavioral analytics templates are based on proprietary Microsoft machine learning algorithms, so you cannot see the internal logic of how they work and when they run. Because the logic is hidden and therefore not customizable, you can only create one rule with each template of this type.
Threat Intelligence	Take advantage of threat intelligence produced by Microsoft to generate high fidelity alerts and incidents with the **Microsoft Threat Intelligence Analytics** rule. This unique rule is not customizable, but when enabled, will automatically match Common Event Format (CEF) logs, Syslog data or Windows DNS events with domain, IP and URL threat indicators from Microsoft Threat Intelligence. Certain indicators will contain additional context information through MDTI (**Microsoft Defender Threat Intelligence**). For more information on how to enable this rule, see Use matching analytics to detect threats. For more details on MDTI, see What is Microsoft Defender Threat Intelligence
Anomaly	Anomaly rule templates use machine learning to detect specific types of anomalous behavior. Each rule has its own unique parameters and thresholds, appropriate to the behavior being analyzed. While the configurations of out-of-the-box rules can't be changed or fine-tuned, you can duplicate a rule and then change and fine-tune the duplicate. In such cases, run the duplicate in **Flighting** mode and the original concurrently in **Production** mode. Then compare results, and switch the duplicate to **Production** if and when its fine-tuning is to your liking.

Microsoft

1

https://learn.microsoft.com/en-us/azure/sentinel/detect-threats-built-in

Top UEBA vendors main features (2)

- *Multi-dimensional behavior baseline*
- *Predictive threat models*
- *Predictive and adaptive learning*
- *Behavioral groups*
- *Daily consolidated risk scores for individuals risk prioritization*
- *Risk scoring, risk-ranked timelines*
- *Risky user behavior analysis*
- *Alert scoring and prioritization*

Top UEBA vendors

2

UEBA Vendor	Use Cases	Special Features	Delivery
Aruba	High-risk and regulated industries	Integrated network traffic analysis	Appliance and software
Dtex	Security operations teams	Forensic audit trail	On-premises software
Exabeam	Large organizations, federal agencies	Ransomware detection and prevention	Physical appliance or cloud-ready virtual machine
Forcepoint	Security operations teams	Consolidated risk scores for individuals; video replays of users' screens	On-premises software
Fortinet	Banks, manufacturers and game developers	Monitors endpoints even when off network	Hosted solution
Fortscale	Organizations of all sizes; security vendors	Darknet analysis; DLP integration	On-premises software or embedded in other security solutions
Gurucul	Corporate security operations	Large library of machine learning algorithms; fuzzy logic-based link analysis	Appliance, virtual machine, cloud or bare metal
Haystax	Federal government, financial industry, corporate IT security, public safety	Integrated view of insider trustworthiness; low rate of false positives	Software or cloud-based
Interset	Security operations teams	Used by multiple U.S. intelligence agencies; more than 200 machine learning models	On-premises or cloud
LogRhythm	High-risk and highly regulated industries	Embedded orchestration, automation and response	Appliance, software and cloud
Microsoft	Small businesses	Mobility support; deep packet inspection	On-premises software
One Identity	Aimed at high-risk privileged accounts	Real-time threat detection, behavioral biometrics	Appliance
Palo Alto	Security operations teams seeking broad protections	The automated alert investigation, impact analysis, threat hunting	Cloud
Preempt	Security operations teams	User risk scoring; forensics; reduced alerts	On-premises software
RSA	Security operations teams seeking automation	Unsupervised anomaly detection and machine learning	Appliance and virtual formats
Securonix	Security operations teams, especially in very large enterprises	Fraud reporting; trade surveillance; patient data analytics	On-premises software or cloud-based
Splunk	Security operations teams	Multi-dimensional behavior baseline; anomaly exploration	On-premises software or AWS service
Varonis	Security operations teams	"Security Time Machine" analyzes past data; ransomware detection	On-premises software
Veriato	Security operations teams and HR departments	Psycholinguistic analysis; screen snapshots; keystroke recording	On-premises software
VMware	Security operations teams seeking broader app and device management	Integrates access control, application management and endpoint management	

https://media.licdn.com/dms/image/C5612AQGIdRyltZM4ZQ/article-cover_image-shrink_720_1280/0/1651446475114?e=1686182400&v=beta&t=MKT1KPXUxWg4y3rDoQ6YtpoDBezzcKPcjxfNUxny7qM

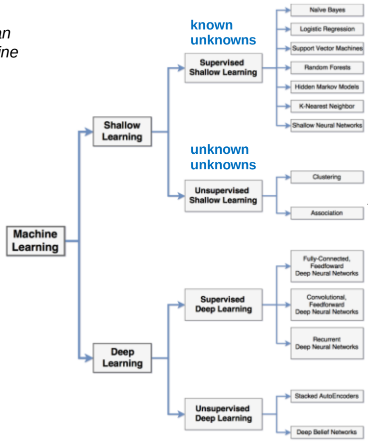
Technology Review

UEBA: Algorithms and System Design

Machine Learning Algorithms

- Two classes of ML algorithms can be used: unsupervised and supervised
- There is no “best” algorithm. It depends on the data and goal

Fig. Algorithms which can be used to build a baseline (i.e. model) compare instances against it [2]



Unsupervised Learning
- K-means
- Hierarchical Clustering
- DBSCAN (Density-based Spatial Clustering of Application with Noise)
- Local Outlier Factor
- One Class SVM

[3] Insider Threat Detection using Deep Learning: A Review

TABLE III. SUMMARY OF DEEP LEARNING MODELS FOR INSIDER THREAT DETECTION

Deep Learning Model	Granularity	Advantages	Disadvantages	Evaluation Metric (Value)
Deep Belief Network [22]	Sessions	Suitable for primitive multi-domain feature processing solutions.	Cannot handle temporal data and generates a large number of false alarms.	AR (87.79%), DR (81.04%), FPR (12.18%)
Autoencoder [23]	Sessions	Autonomous organization-level solutions giving minimum false alarms.	Cannot handle temporal data.	TPR (100%), FPR (best case: 1%)
Recurrent Neural Network [18, 19, 20, 21]	Sessions	Most suitable approach for Insider Threat Detection as it can efficiently handle temporal data.	Can find suspicious events but cannot confirm if they amount to insider threats.	AUC (best case: 0.9449)
Convolutional Neural Network [24]	Sessions	Enables faster processing of feature matrices when combined with LSTM.	Cannot handle temporal data if not combined with LSTM.	AR (best case: 100%)

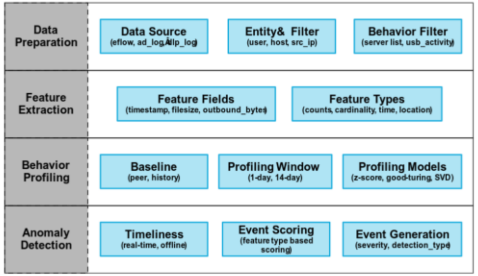
How it Works

- ML algorithms are used to create baseline models to capture the normal behavior of users and entities, e.g.
  - *Create a model for user A which logs into the same group of machines at approximately the same times every day*
  - *if user A suddenly logs into a different machine, the ML algorithm marks this new machine as an outlier since the behavior is far from the normal baseline*
- Challenge: how to identify how risky this abnormal behavior is?
- An approach is to look at context: e.g., to analyze the behavior of peers

System Design (1)

- *Data Preparation*. Obtain data from all relevant data sources and apply filters, groupby, transformation, and integration
- *Feature Extraction*. Generate new features based on custom made functions
- *Behavior Profiling*. For each user/entity, the extracted features are used to generate model baselines using machine learning algorithms
- *Anomaly Detection*. Instances (i.e., feature vectors) are scored against behavior profiles with an associated confidence score

Fig. General UEBA Architecture [1] 1



[1] User and Entity Behavior Analytics for Enterprise Security
[2] On the effectiveness of machine and deep learning for cyber security

Technology Review

Azure Sentinel

Key features: Query language, connectors and rules build be the community, and support for investigations via jupyter notebook

Collect Data 1

- Common Event Format (CEF), Syslog, REST-API, Fluentd, LogStash
- Azure Active Directory, Azure DDoS Protection, Azure Firewall, Azure Web Firewall, Office 365, AWS CloudTrail, DNS, Azure Stack VMs, …

Community 2

- GitHub that contains several data sources for threat: hunting queries, playbooks, workbooks..

Query Language 3 4

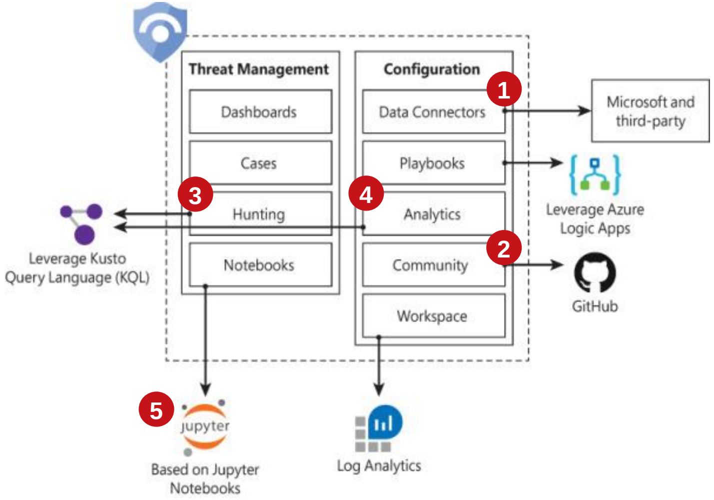
- Kusto Query Language (KQL) enable to manipulate data in Sentinel: simple and efficient

Query for retrieving the failed logons (event id 4625) for the last 24 hours

```
SecurityEvent
| where EventID == 4625
| where AccountType == 'User'
| where TimeGenerated > now() - 24hrs
| project TimeGenerated, Account, DomainController=Computer, IpAddress,
LogonType, ClientHostName=WorkstationName, Activity, SubStatus
```

Query When a domain is flagged by Defender for Cloud (Azure Security Center) as suspicious then find any other clients that have queried that domain in DNS events

```
let suspiciousurl=
SecurityAlert
| where AlertName startswith "Communication with suspicious random domain name"
| mv-expand todynamic(Entities)
| project Entities
| extend SuspiciousURL = tostring(Entities.DomainName)
| where isnotempty(SuspiciousURL)
| distinct SuspiciousURL;
    DnsEvents
    | where QueryType == "A"
    | project Name, ClientIP
    | where Name in (suspiciousurl)
| summarize ['Client IPs']=make_set(ClientIP) by Name
```



3 4 **SQL to KQL mapping**

- https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/sqlcheatsheet

2 **Query wrapper for (complex) query**

- https://github.com/Azure/Azure-Sentinel/blob/master/Solutions/Azure%20Active%20Directory/Analytic%20Rules/FailedLogonToAzurePortal.yaml

5 **Investigation via notebooks**

- https://github.com/Azure/Azure-Sentinel-Notebooks

CI/CD

- https://learn.microsoft.com/en-us/azure/sentinel/ci-cd?tabs=github

https://github.com/reprise99/Sentinel-Queries

Technology Review
Matano (open source)

1 Data Ingestion
- S3, SQS, AWS CloudTrail (governance, compliance, operational auditing, and risk auditing), Zeek (network security monitor), Okta (identity management service), and SaaS sources

2 ETL
- Transformation pipeline via Vector Remap Language (VRL)
  - VRL was bought by DataDog. It replaces, e.g., Logstash
- Uses Elastic Common Schema (from ELK/ES)
- Dozens of transformations, pre-built parsers and integrations exist to ingest security logs from popular cloud, host, and SaaS tools using

Transform HTTP log events

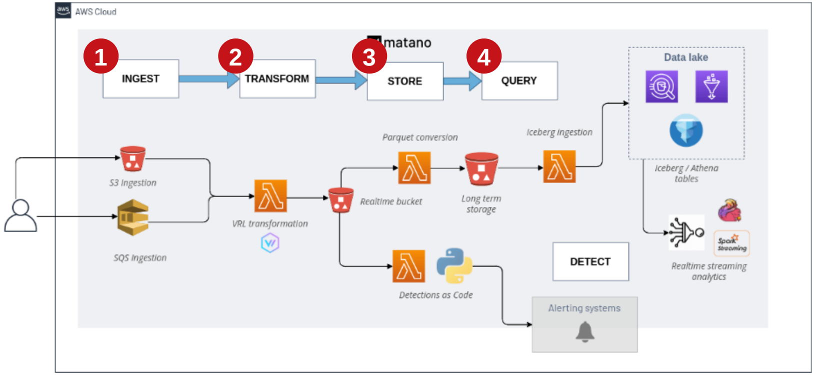
- *Parse the raw line string into JSON, and explode the fields to the top level*
- *Rename srcIpAddress to the source.ip ECS field*
- *Remove the username field*
- *Convert the message to lowercase*

```
transform: |
  . = object!(parse_json!(string!(.json.line)))
  .source.ip = del(.srcIpAddress)
  del(.username)
  .message = downcase(string!(.message))

schema:
  ecs_field_names:
    - source.ip
    - http.status
```

3 Storage/Query Engines
- Log stored in Parquet files in S3 object storage
- Data lake with Big Data technologies: Apache Arrow (columnar memory format for flat/hierarchical data); Apache Iceberg (high-performance format for huge analytic tables; SQL; engines: Spark, Flink, Snowflake, Hive)

4 Detections as Code
- Detection rules are coded in Python (detection-as-Code)
- Rules are managed in Git (test, code review, audit for hardening)

Matano Architecture



Detection-as-Code

Detect Brute Force Logins by IP across all configured log sources (e.g. Okta, AWS, GWorkspace)

```
---
tables:
  - aws_cloudtrail
  - okta_system
  - o365_audit
alert:
  severity: medium
  threshold: 5
  deduplication_window_minutes: 15
  destinations:
    - slack_my_team
```

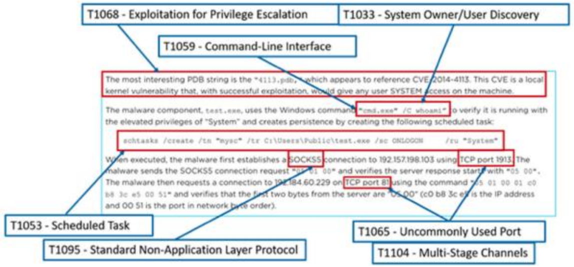
```
def detect(r):
    return (
        "authentication" in r.deepget("event.category", [])
        and r.deepget("event.outcome") == "failure"
    )


def title(r):
    return f"Multiple failed logins from {r.deepget('user.full_name')} - {r.deepget('source.ip')}"


def dedupe(r):
    return r.deepget("source.ip")
```

System Design
Workflow Description [1]

1 **Identify Behavior**
- Brainstorming and understanding how a malicious attack works
- Describe the scenarios to model

2 **Map scenario to ATT&CK**
- Map tactics: e.g., TA0001: Initial Access
  - https://attack.mitre.org/tactics/enterprise/
- Map Techniques: e.g., T1078: Valid Accounts
  - https://attack.mitre.org/tactics/TA0001/
- Map Procedures: e.g., S0362: Linux Rabbit
  - Linux Rabbit acquires valid SSH accounts through brute force

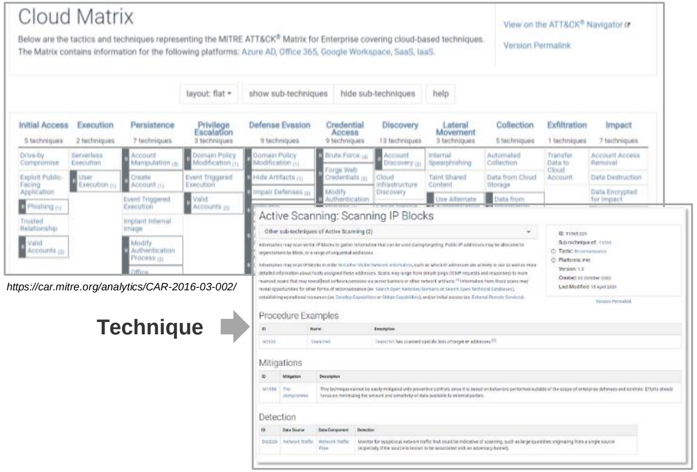
Workflow can also start at step #2, and use the MITRE ATT&CK framework as a scenario generator



https://medium.com/mitre-attack/getting-started-with-attack-cti-4eb205be4b2f

The Enterprise Matrix is more complete and can also be used

Tactics



https://car.mitre.org/analytics/CAR-2016-03-002/

Technique

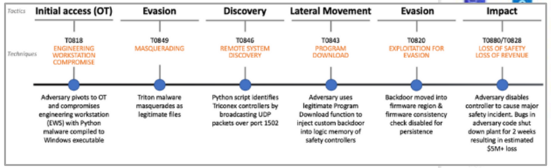
https://attack.mitre.org/techniques/T1595/001/

3 **Identify Data sources**
- Several technique lists relevant data sources that can help with the attack detection
- E.g., process and process command line monitoring, often collected by Sysmon, file and registry monitoring, authentication logs, packet capture, …

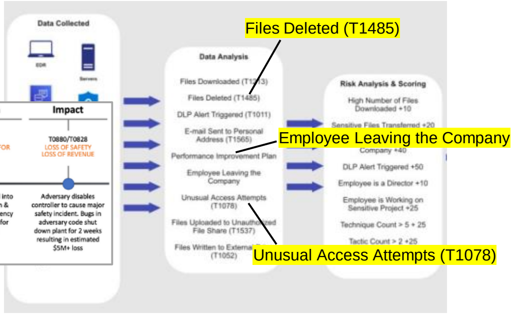
4 **Collect Data**
- Collect key data into some kind of search platform (DataLake, SIEM, DB, …)
- E.g., Sysmon data → ELK

5 **Update Knowledge Base of Analytics**
- Implement rules (pseudocode and native) and select machine learning models
- See https://car.mitre.org/analytics/CAR-2016-03-002/
- See https://github.com/redcanaryco/atomic-red-team/tree/master/atomics
- See https://github.com/reprise99/Sentinel-Queries

Using Kill-Chain and ATT&CK

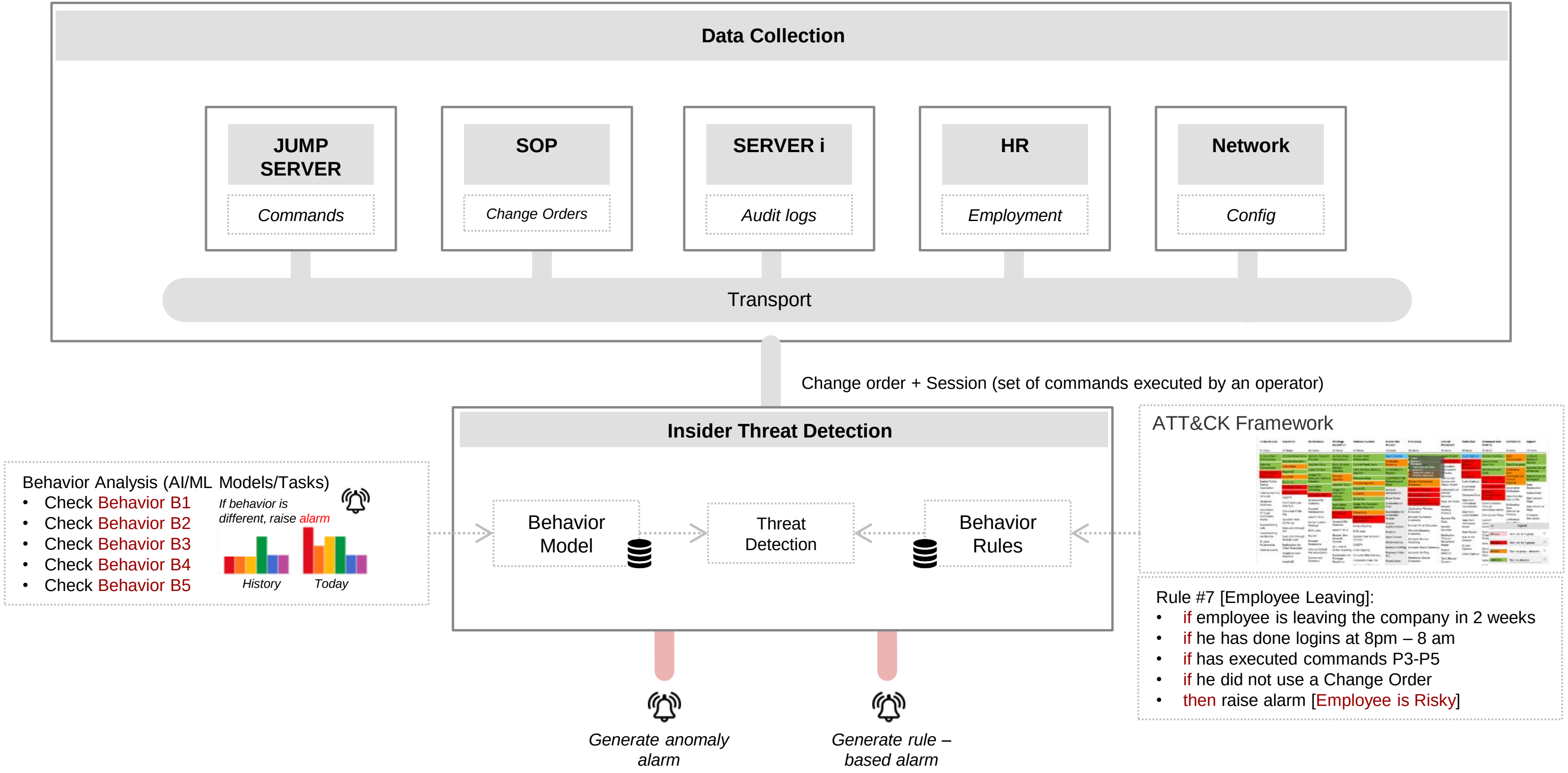


How to use Techniques and Rules to calculate an overall risk score



[1] https://medium.com/mitre-attack/getting-started-with-attack-detection-a8e49e4960d0

System Design
Context



(*) https://ctid.mitre-engenuity.org/our-work/insider-threat-ttp-knowledge-base/
https://github.com/center-for-threat-informed-defense/insider-threat-ttp-kb

System Design
High Level View

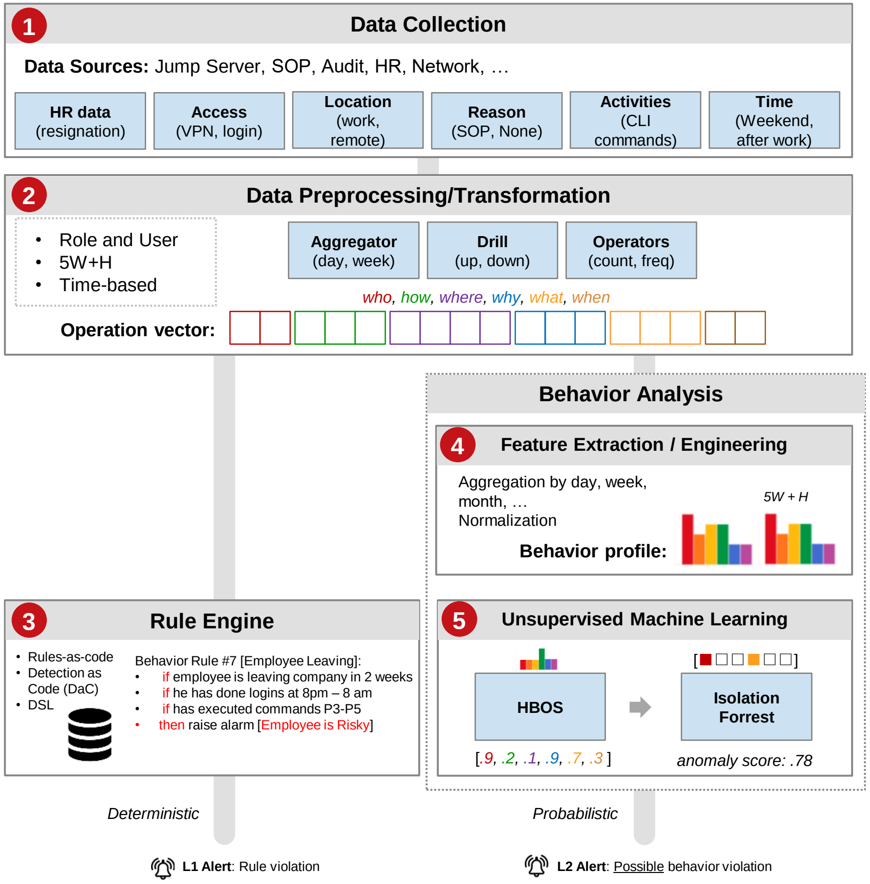
1 Input Data
- Data comes from different logging systems such as OREO, Bastion, SOP, Audit, HR, Network

2 Data Pre-processing/Transformation
- Apply transformations to incoming data
- Filter out irrelevant data and remove sensitive information
- Enrich existing data with analytics or external data

3 Rule Engine
- SecOps: Detection as Code (DaC)
- DaC is a modern approach to building threat detection
- The simplicity of rule definitions is key for adoption and management
  - Don't make the query any more complex than absolutely needed
  - Contrast: Google YARA-L vs. Azure KQL
- Sigma simplicity is an option to consider
  - https://github.com/SigmaHQ/sigma
- Rule development follows CI/CD workflows
  - Structured process that lets teams participate in building and improving rules for SIEM

4 Feature Engineering
- Aggregated data is processed to construct feature vectors representing user activities and profile information at different granularity levels
- Frequency features. Count of different types of actions the user performed in the aggregation period, e.g., number of commands executed, number of logins after work hour, …
- Statistical features. Descriptive statistics, such as mean, median, standard deviation, of data. Examples of data that are summarized in statistical features are file sizes and session duration

5 Unsupervised Learning
- ML algorithms are employed to data analytics based on the constructed feature vectors.
- Supervised learning algorithms cannot be employed to learn the ground truth on malicious/normal user behaviors
- System operates without the participation/supervision of security analysts
- Automatically learn patterns characterizing normal activities and behaviors



(*) https://ctid.mitre-engenuity.org/our-work/insider-threat-ttp-knowledge-base/
https://github.com/center-for-threat-informed-defense/insider-threat-ttp-kb

Data Collection/Preprocessing
Concepts and Technologies

Data Sources
- System records each user operation/action. The information comes from different sensors within the organization.
- Information is classified using the 5W1H method
- In our initial work, we used the data sets provided by CMU-CERT which provide activity logs with different activities: login, usb device, e-mail, web, file access.
- For Huawei Cloud, the complete set of sensors still need to be identified.

5W1H Method
- We use 5W1H as a questioning and problem-solving method to analyze users' operations from different perspectives.
- It helps to understand the context of users' operations and find the root cause of abnormal/malicious actions.
- 5W → What, Where, When, Why, Who; H → How
  - *Who executed the operation?*
  - *What was executed?*
  - *When was it executed?*
  - *Where was the operation executed?*
  - *Why was the operation executed?*
  - *How was it executed?*

Examples of data sensors for *Who*

Security	Frequent or unusual security incidents, compliance violations
Performance	Declining or poor performance, HR complains or demotion
Personality	Past lies to the employer, psychological disorders

Data Sources: OREO, Bastion, SOP, Audit, HR, Network, …

WHO			*HOW*			*WHERE*			*WHY*			*WHAT*			*WHEN*		
Role	**User**	**Termination**	**Protocol**	**Session**	**Status**	**Office**	**Remote**	**…**	**SOP**	**Change Ticket**	**…**	**Command**	**OREO Label**	**…**	**Date / Time**	**Weekend**	**After hours**
SRE	J00123	TRUE	bastion	SE123	SUCCESS	TRUE	FALSE		S92	CT03		ls –al	P5		03.04.23	TRUE	TRUE

Semantic classification of data sources

Data Preprocessing
- Machine Learning algorithms understand only numbers, thus, conversion is needed

Table 2. Encoded features at pre-processing phase.

Feature	Possible Values
Day	0, 1, 2, 3, 4, 5, 6
Time	1, 2, 3, 4, … , 24
User	String Type
PC	String Type
Activity (1)	1, 2, 3, 4, 5, 6, 7

(1) The values for the activity feature correspond to the user's activities, such as logon, logoff, connect, disconnect, HTTP, file and e-mail.

An example dataset comprised of the aforementioned features is shown in Table 3.

Table 3. Encoded features at pre-processing phase.

Day	Time	User	PC	Activity (1)
5	2	4512	4512	3

(1) We encoded categorical features using One-Hot Encoding scheme at a later stage since Machine learning algorithms are more effective in prediction when working with datasets encoded this scheme.

Technologies
- Technologies: Vector, Vector Remap Language (VRL), Filebeat, FluentBit, FluentD, Logstash

Examples of VRL language for .json transformation

```
. |= parse_regex!(.message, r'^(?P<timestamp>\d+/\d+/\d+ \d+:\d+:\d+ \+\d+) \[(?P<severity>\w+)\] (?P<pid>\d

# Coerce parsed fields
.timestamp = parse_timestamp(.timestamp, "%Y/%m/%d %H:%M:%S %z") ?? now()
.pid = to_int!(.pid)
.tid = to_int!(.tid)

# Extract structured data
message_parts = split(.message, ", ", limit: 2)
structured = parse_key_value(message_parts[1], key_value_delimiter: ":", field_delimiter: ",") ?? {}
.message = message_parts[0]
. = merge(., structured)
```

[1] Metadata and examples from CERT r5.2 (CERT Datasets: https://doi.org/10.1184/R1/12841247.v1.) which simulates an organization with 2000 employees over the period of 18 months. Dataset consists of user activity logs, categorized as follows: login/logoff, email, Web, file and thumb drive connect, as well as organizational structure and user information

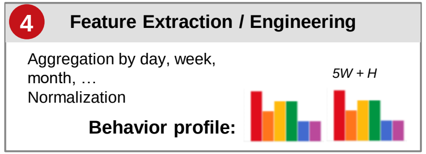
readme.txt

Behavior Analysis
Machine Feature Extraction / Engineering

Behavior Profile

- For each user and role (who), and for each 4W+H model of the vector, we build a behavior profile that denotes the operations for a particular user based on the observed records
- E.g., in [1], user-week, user-day, and user-session profiles were adopted

Data type	Notation	Aggregation criterion c
User-Week	$\mathbf{x}_w$	Week of user actions on all PCs
User-Day	$\mathbf{x}_d$	Day of user actions on all PCs
User-Session	$\mathbf{x}_s$	Session of user actions, from login to logoff on a PC
User-Subsession Ti	$\mathbf{x}_{t=i}$	i hours of user actions in each session
User-Subsession Nj	$\mathbf{x}_{n=j}$	j user actions in each session



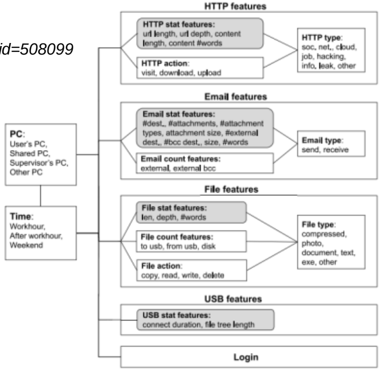
Graphical representation of the profiles created. (1) For each 4W+H profiles, we compute deviation values using, e.g., standards deviation or covariance. (2) The deviation value is used to identify anomalous behavior.

Feature Extraction

For each behavior profile, perform feature extraction

- This enables the ML algorithm to compare between users, roles, and derived characteristics of the 5W+H model which are relevant
- Features enable an assessment of 3 key areas:
  - User's hourly/daily/week/weekend operations
  - Comparison (user's operations vs previous operations)
  - Comparison (user's operations vs other users with same role)

Examples of features extracted for the CERT R5.1 dataset
https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099



Features

Several types of features need to be evaluated, e.g.:

- New observations across *where*, *when*, *what*, *how*, *who*
  - e.g., new remote login on holiday
  - New computer access for user/role
  - New command execution for user/role
- Count. Assess the hourly and daily usage counts for *where*, *when*, *what*, *how*, *who*
  - e.g., number of logins during the night
  - Daily number of commands executed
- Time. Time-based features for each particular activities (e.g., earliest login)

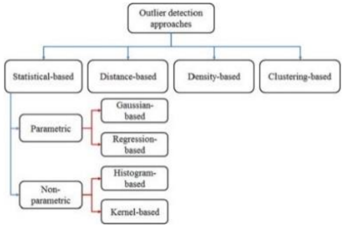
The final feature matrix will contain tens extracted features

[1] Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning

Behavior Analysis
Unsupervised Machine Learning

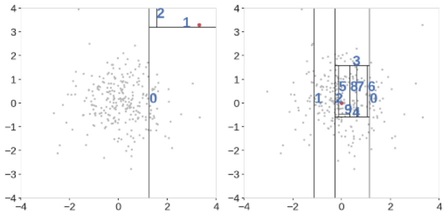
Single Abnormal Behavior Detection
- Each behavior profile is compared using, e.g., HBOS [1]
- For each selected feature, we prepare a set of training histograms.
- We then map the vectors of training histograms into a metric space so that: 1) two similar histograms are close in space, whereas 2) two dissimilar histograms are far away.
- A number of different approaches can be used to quantify how similar two histograms are.

Several techniques for histogram-based outlier detection need to be evaluated.
- *Statistical techniques*
- *Density-based*
- *Clustering-based*

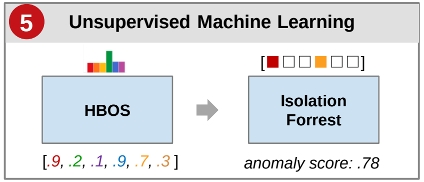


Full Abnormal Behavior Detection
- Isolation Forests (IF) can be used for anomaly detection since they are efficient for large datasets
- The results of single abnormal behavior detection are placed in a vector
  - E.g., [.9, .2, .1, .9, .7, .3]
- All samples are used to build a model, IF will classify vectors which can be quickly isolated within the tree constructed

Example of the use of Isolation Forests to isolate vectors which are outliers



(a) Anomaly point (b) Nominal point



HBOS is used to detect profiles which are outliers. In some cases, all individual profiles are normal, but their combination is abnormal. Thus, we use Isolation Forests as a second level of insider threat detection.

[1] Goldstein, M. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. 2012

Thank you.

Bring digital to every person, home and organization for a fully connected, intelligent world.