Implementing Data Exfiltration Defense in Situ: A Survey of Countermeasures and Human Involvement

MU-HUAN CHUNG, University of Toronto, Canada
YUHONG YANG and LU WANG, University of Toronto, Canada
GREG CENTO, KHILAN JERATH, and ABHAY RAMAN, Sun Life Financial, Canada
DAVID LIE and MARK H. CHIGNELL, University of Toronto, Canada

In this paper we consider the problem of defending against increasing data exfiltration threats in the domain of cybersecurity. We review existing work on exfiltration threats and corresponding countermeasures. We consider current problems and challenges that need to be addressed to provide a qualitatively better level of protection against data exfiltration. After considering the magnitude of the data exfiltration threat, we outline the objectives of this paper and the scope of the review. We then provide an extensive discussion of present methods of defending against data exfiltration. We note that current methodologies for defending against data exfiltration do not connect well with domain experts, both as sources of knowledge and as partners in decision-making. However, human interventions continue to be required in cybersecurity. Thus, cybersecurity applications are necessarily socio-technical systems which cannot be safely and efficiently operated without considering relevant human factors issues. We conclude with a call for approaches that can more effectively integrate human expertise into defense against data exfiltration.

CCS Concepts: • **General and reference → Surveys and overviews**.

Additional Key Words and Phrases: Exfiltration Threats, Cybersecurity Countermeasures, Machine Learning, Human Factors, Insider Threats, Human-Computer Interaction

ACM Reference Format:
Mu-Huan Chung, Yuhong Yang, Lu Wang, Greg Cento, Khilan Jerath, Abhay Raman, David Lie, and Mark H. Chignell. 2023. Implementing Data Exfiltration Defense in Situ: A Survey of Countermeasures and Human Involvement . *ACM Comput. Surv.* 1, 1 (January 2023), 37 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 MAGNITUDE OF THE DATA EXFILTRATION THREAT

Since data can be very valuable in a variety of contexts (government, banking, etc.), data is a target for a variety of adversaries including criminals, governments, and even law enforcement. Almost anyone, even non-technical personnel armed with the right tools, can perform some sort of attack vector to exfiltrate highly valuable objects, making the fight against data exfiltration threats extremely challenging. Due to the large potential losses associated with exfiltration events, countermeasures against exfiltration have become a top priority for organizations when securing cyber defense perimeters. Unfortunately, securing an organization's data perimeter, by itself, will

Authors' addresses: Mu-Huan Chung, mhm.chung@mail.utoronto.ca, University of Toronto, 40 St George St, Toronto, Ontario, Canada; Yuhong Yang, yuhong.yang@mail.utoronto.ca; Lu Wang, wanglu.wang@mail.utoronto.ca, University of Toronto, 40 St George St, Toronto, Ontario, Canada; Greg Cento, greg.cento@sunlife.com; Khilan Jerath, khilan.jerath@sunlife.com; Abhay Raman, abhay.raman@sunlife.com, Sun Life Financial, 1 York St, Toronto, Canada; David Lie, david.lie@utoronto.ca; Mark H. Chignell, chignell@mie.utoronto.ca, University of Toronto, 40 St George St, Toronto, Ontario, Canada.

0360-0300/2023/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

not eliminate exfiltration threats. Over the last decade, a massive amount of user information has been leaked, while recognition and response within those organizations was slow to materialize.

A prominent example of data exfiltration was the Sony PlayStation Network (PSN) data breach. In April 2011, Sony shut down its PSN for over a month due to a data breach. Names, addresses, birth dates, credentials, and credit card information were stolen. Sony was criticized for its late response in informing PSN users. Sony notified its customers a week later after they realized there was an exfiltration event [27]. About 77 million user accounts were affected in this event, and it could be the largest ever credit card information leak incident [155].

Public departments are also valuable targets. The voter data leak in 2016 exposed 55 million Filipino voters' fingerprints and passport information [59]. In the Office of Personnel Management (OPM) hack, 21.5 million federal employees' background information, including their names, addresses, social security numbers, and 5.6 million fingerprints were leaked [73]. The hacker group leveraged a compromised contractor's credentials to access OPM's internal network to exfiltrate valuable data. The reaction of the OPM office was significantly delayed, where one article suggested that the hackers might have been stealing data for more than a year until the OPM office finally discovered it through a third-party company's disclosure [67].

Exfiltration events can also be launched by government agencies [86]. The Yahoo breach, one of the largest data breach events so far, was carried out by hackers believed to be aligned with Russian state security service [219]. Through phishing emails, these hackers successfully obtained valid credentials for the user database and details regarding the account management tool. The database contained names, phone numbers, password challenge questions/answers. It also stored password recovery emails and a cryptographic value unique to each account, which later allowed the hackers to access their target victims including an assistant to the deputy chairman of Russia, an officer in Russia's Ministry of Internal Affairs, a trainer working in Russia's Ministry of Sports, some Russian journalists, and some U.S. government workers [219]. Yahoo! estimated that all of its user accounts, roughly 3 billion, were affected by this event [202], which thus made it one of the largest events ever, in terms of number of people/accounts affected.

In addition to user claims, companies subject to exfiltration events usually have to pay for fines, settlements, and penalties relating to 'poor handling' of cyber threats. In 2018, Yahoo was fined $35 million by the U.S. Securities and Exchange Commission (SEC), and the class action lawsuit penalty cost around $50 million dollars. In two more recent financial company breach events -the Equifax breach (losing 150 million user records) and the Capital One breach (affecting 100 million users) - Equifax agreed to pay $575 million in a settlement with the Federal Trade Commission, the Consumer Financial Protection Bureau (CFPB); whereas Capital One was fined by the Office of the Comptroller of the Currency for $80 million [203].

The 2014 McAfee Centre for Strategic and International Studies report calculated that the total annual cost of cybercrime was around $400 billion, where data exfiltration was the main motivator for these attacks [128]. In recent years, cyber breach objectives have gradually transformed into delivering/installing ransomware (which not only undermine information confidentiality as in regular exfiltration events but also affect system availability). Data exfiltration has consequently become a major component of ransomware attacks, where adversaries leverage the fear of sensitive data disclosure or destruction to demand a ransom [147]. The use of ransomware that leads to exfiltration threats may create much greater costs than simply losing access to proprietary data. The latest Crowdstrike global threat report revealed that some adversaries even setup marketplaces to advertise and sell potential victim's sensitive data [49].

While there have been many technical approaches to battle against exfiltration threats, an earlier report (the SANS 2016 security analytics survey [179]) indicated that many organizations still rely on inadequate security, with the following problems being highlighted :

- Corporations are short of skilled professionals, funding, and resources to support security analytics.
- Organizations are still having trouble baselining 'normal' behavior in their environments, a metric necessary to accurately detect, inspect, and block anomalous behaviors.
- Only 4% of respondents consider their analytics capabilities fully automated.
- Just 22% of respondents are currently using tools that incorporate machine learning (ML), where ML offers more insights that could help less skilled analysts with faster detection, automatic reuse of patterns detected, and more.

The 2020 SANS Network Visibility and Threat Detection Survey [160] further reported that while conventional rule-based and signature-based methods have been utilized in most organizations' networks/hosts, of the participating organizations:

- 59% still believe that lack of network visibility poses a high or very high risk to their operations.
- 64% of respondents experienced at least one compromise over the past 12 months.

The situation has not improved in recent years [49], as there is a continuing lack of skilled professionals. In fact, as corporations moved their critical assets including sensitive data to the cloud, protecting against exfiltration threats became even more complicated, because cloud-based assets created an additional attack surface. Thus organizations had to deal with problems arising from having too many people potentially able to access sensitive data from their cloud data repositories). Insufficient human resources dedicated to cybersecurity, combined with increasing system complexity, likely explain why insider exfiltration threat has become the second most common cloud threat [127].

Industry reports have revealed socio-technical issues that limit the effectiveness of defense perimeters in combating exfiltration threats. In other words, a significant source of the challenge in tackling cybercrime and data exfiltration is the complexity of the information to be analyzed by human actors. Thus in the remainder of this survey, we review current technologies in place to defend against exfiltration incidents. set in the broader view of approaches being applied in industry in order to reveal potential issues when considering socio-technical relationships between organizations, humans, and the machine.

2 OBJECTIVES AND RESEARCH QUESTIONS

Surveys reported in the previous section revealed that dealing with exfiltration requires not only securing perimeters, but also dealing with complex socio-technical issues that limit the effectiveness of defense perimeters in combating exfiltration threats. As the technology implemented to strengthen perimeters becomes more advanced, system networks are being secured with more complicated defensive applications. However, the problem of whether or not domain experts can fully trust, or properly operate, these new technologies, is rarely discussed.

In dealing with complex, inside the perimeter issues, the human component (domain experts such as security analysts, security engineers, IT/network admins, etc.) is usually key in resolving/mitigating threats. Human decision makers need to respond to a wide variety of cybersecurity incidents. However, human involvement in the application of defense countermeasures against data exfiltration has received scant attention in past reviews of relevant research literature. Thus, this survey aims to fill the gap concerning interactions between the human component and current countermeasures. Inspired by the literature comparison provided in the survey work published by Sabir et al. [164], we also summarize the difference between this review with past literature reviews in this area (Table 1).

Table 1. Comparison between the current survey and major previous surveys on relevant topics in the past decade

Topics Covered	[178]	[209]	[117]	[8]	[70]	[164]	[96]	[26]	[11]	[65]	This Survey
Adversary Types and Characteristics	x	x	x		x		x	x	x	x	x
Attack Vectors and Campaigns	x	x	x				x	x			x
Threat Models and Frameworks							x	x			x
Countermeasures	x	x	x	x				x			x
Countermeasure Limitations	x	x	x	x	x				x		x
Countermeasure Human Factors									x	x	x
ML Solutions		x	x	x	x	x	x		x		x
ML Limitations						x	x				x
Human Role in Expert-ML Systems											x

As can be seen in Table 1, our survey covers a more comprehensive set of topics than earlier surveys, focusing particularly on the human component that has often been ignored in earlier surveys. It should be noted that while [11] and [65] have covered human factors topics, they either focused on behavior analysis approaches [11] or situational awareness [65]. In addition to covering more recent literature, this survey also covers a wider variety of issues that arise when supportive/automated approaches are introduced to what has previously been a more human-directed workflow. The following research questions summarize our motivation (Table 2).

Table 2. Research questions as the foundation of this survey

	Research Questions	Tasks and Objectives
RQ1	What countermeasures are being applied against internal exfiltration threats?	Identify common defensive approaches applied in industry to detect exfiltration events, and each of their usage scenarios and limitations.
RQ2	What are the human roles/tasks in these countermeasures?	Identify the human component in terms of human experts' role in the human-technology system of the countermeasure being applied.
RQ3	What are the actual benefits/limitations after applying these countermeasures, considering human users, organizational structures, and other socio-technical factors?	The objective of this research question is to determine the actual value of defense countermeasures, considering the whole socio-technical system efficiency, so as to identify research gaps.

As shown in Table 2, this survey extends previous work by considering human involvement in defending against exfiltration threats. We started by defining research scope and potential actors (section 3). We then reviewed cyber threat model frameworks and associated defensive approaches, summarizing current use of different methods across sectors (section 4). This summary should help readers understand the application of these defensive countermeasures against exfiltration threats. We then review the limitations of these approaches, focusing in particular on the human tasks that can be difficult for domain experts.

3 SCOPE OF THIS REVIEW: TYPES OF THREAT AND ACTOR

Since cybersecurity is a complex domain that involves socio-technical interactions between adversaries, it is useful to start by defining the scope of the threat and the actors involved. Based on NIST's "Guide for Conducting Risk Assessment" [24], there are four major types of threat sources:

- Adversarial: individuals or groups that seek to exploit the organization's cyber resources

- Accidental: erroneous actions taken by individuals executing everyday responsibilities
- Structural: failures of equipment, environmental controls, or software due to aging, resource depletion, or other circumstances which exceed expected operating parameters
- Environmental: disasters and failures of infrastructures that are outside the control of the organization (e.g., cases where backup tapes are lost by trucking companies [98])

In this study we consider mostly adversarial threats (excluding structural and environmental threats, and only discuss accidental threats for those situations where unintentional behavior can potentially do the most damage) due to the nature of exfiltration incidents, that mostly involve direct human activity. Accidental threats are usually conducted by a legitimate user. This type of threat involves unintentional violation of norms or policies [81, 198] and is usually detectable with customized DLP (Data Loss Prevention) systems that follow organization policies. By contrast, adversarial threats usually come from external sources and may be carried out persistently and covertly (and be harder to detect as a result) if the attackers have sufficient resources.

Malicious external adversaries who have established a foothold inside the perimeter are usually referred to as masqueraders [136]. Establishing this foothold typically requires a sequence of activities [117], with a common attack campaign involving three stages: research, attack, and exfiltration [209]. In the research stage, sometimes referred to as the enumeration stage, attackers can leverage OSINT (Open-Source INTelligence) to search for public-facing domains and potential disclosure of internal information. They can also choose more aggressive approaches such as port scanning or web vulnerability scanning in order to discover unpatched vulnerabilities or bad codes/misconfigured settings of public-facing servers. Attackers can then exploit discovered vulnerabilities such as local/remote file inclusion (LFI/RFI), SQL injection, insecure direct object reference (IDOR), cross-site request forgery (CSRF), etc., to get remote code execution, hijack user sessions, or obtain user credentials that may later on yield remote access. The whole attack campaign may eventually lead to the exfiltration of sensitive data.

In addition, masqueraders having abundant resource, e.g., funded by hostile state entities, may carry out more sophisticated attack campaigns and are more capable of maintaining a C2 (Control and Command) channel, targeting enterprise or government networks. Such long-term threats posed by well-resourced adversaries are typically referred to as APTs (Advanced Persistent Threats) [38].

Regardless of which TTPs (tactics, techniques, procedures) and how sophisticated attack campaigns external adversaries employ in order to get access to the internal network, they eventually impersonate internal users [166]. This often leads to a "shared" user account which is effectively owned by both the original valid user, and the new malicious user who will misuse the account credentials from time to time. Thus, defending against exfiltration at this stage may require focusing on behavioral changes of internal users, since significant changes in a user's behavior may be due to the actions of malicious attackers who have captured, or are sharing, the user account.

Since data exfiltration threats arise not only from external actors, we also consider internal actors in this review. Internal actors may pose even greater threats to data security, with industry reports suggesting that internal threats are increasingly serious. The proportion of exfiltration threats conducted by internal actors increased from 17% in 2011 to 30% in 2020 [14, 212]. Internal actors may have been authorized with legitimate access to an organization's internal computer systems, data, or networks, but when they act maliciously (i.e., their actions are counter to policy/code of conduct) they are referred to as traitors [74, 149]. In the context of data exfiltration, the goal of these "traitors" is to "negatively affect confidentiality, integrity, or availability of some information asset" [166] for a variety of incentives such as revenge, monetary reward, hacktivism, etc.

Most traitors depend on four main enabling resources: Access to the system; ability to represent the organization; knowledge of the system/network; gaining the trust of the organization [90]. Traitors can have a variety of roles such as employees, contractors or consultants, clients or customers, joint venture partners, and vendors. However, external actors may also recruit, or collaborate with, trusted internal personnel and thus create an insider threat by allying with an internal user [140].

Traitors, as well as masqueraders who have successfully obtained valid credentials and sufficient knowledge, share the following properties:

- They have access to the system
- They can represent the organization
- They have knowledge about the internal workings of the system they have infiltrated

In principle, insiders, whether traitors or masqueraders, should behave differently from other users as they prepare a data exfiltration exploit [42, 70, 84]. Thus, the kind of analysis needed to defend inside the perimeter will mainly depend on differentiating normal from abnormal behavior. Previous work on data exfiltration has relied on anomalous behavior detection, often using statistical and machine learning techniques [112, 135]. However, algorithms that seek to detect anomalies typically do not have access to the implicit human knowledge that can recognize subtle differences in normal versus abnormal behavior. It has proven difficult to provide accurate detection of malicious behavior without generating large numbers of false alarms (false detections), because behavior will tend to differ across different adversaries, who will have different motivations, resources, and preferred methods. Thus in the following sections, we will consider actors as insiders with similar data exfiltration motivations, regardless of whether there were originally inside the network (traitors) or not (masqueraders).

4 DEFENSE AGAINST EXFILTRATION

Numerous countermeasures have been proposed to protect cyber properties for organizations in terms of their "CIA" (confidentiality, integrity, and availability) in recent decades. Each of these countermeasures can support the detection of certain types of anomalous activities, in different stages of an attack campaign. However, within the scope of this research, not every approach is suitable for detecting/protecting against exfiltration threats.

In this section we survey common countermeasures that protect against exfiltration threats using a top-down approach. We start by reviewing cyber threat models and frameworks that capture core characteristics of exfiltration campaigns, so as to better conclude useful and prevalently implemented countermeasures. We first summarize best-of-breed cyber threat models, commonly used in industry, to elucidate the usual countermeasures chosen by organizations against exfiltration attempts. We also discuss the advantages and limitations of these countermeasures in combatting exfiltration activities, and we highlight their inattention to human factors issues associated with how experts interact with these countermeasures or interpret their output.

Our goal in this section is to help readers understand which approaches are required at each stage, so as to prevent an active campaign from advancing further (often referred to as the "kill chain"). As part of the exposition, we will drill down into the details of each countermeasure, from the most passive and uni-functional, to proactive and integrated approaches, in order to illustrate their usefulness and limitations.

4.1 Cyber Threat Models and Frameworks

While conventionally security events are handled as separate incidents, each incident is usually the result of a sequence of failures in corresponding security controls. Using a bottom-up approach to

resolve incidents separately can patch holes on the attack surface. However, it neither guarantees proper protection against future threats nor improves the overall security of the organization. A top-down, comprehensive (and most likely manual) review of the system-wise security design is needed to make sure that the overall security posture is robust against novelties. Thus researchers have proposed using cyber threat models to provide high-level aspects regarding: attack surface and vulnerability; risk and impact; stage and campaign from both attackers and defenders' point of view. By using this approach, practitioners can achieve a top-down, broader view, of how to reduce attack surface so as to improve all around security.

Previous studies defined threat modeling from different points of views (aspects), as summarized in the following Table 3 [224].

Table 3. Defining Different Aspects of Threat Modeling

Aspect	Definition
General	• A structured way to secure software design by understanding an adversary's goal in attacking a system based on the system's assets of interest [20, 201] • Threat modeling is the process of enumerating and risk-rating malicious agents, their attacks, and those attacks' possibleimpacts on a system's assets [197] • A sound analysis of potential attacks or threats in various contexts [210]
System Evaluation	• A conceptual exercise to analyze a system's architecture or design to find security flaws and reduce architectural risk [153] • The process to analyze system architecture, identify potential security threats, and select appropriate mitigation techniques [66, 224]
Application Development	• A systematic way to identify threats that might compromise security [123] • A process to analyze the security and vulnerabilities of an application or network services [51, 186]

Various threat models have been proposed to fulfill cybersecurity needs, with commonly accepted models, such as the cyber kill chain, later evolving into cybersecurity frameworks. These frameworks collectively describe the practical usage of security technologies in terms of their targeting threats and application domains. Most frameworks help field workers to identify response and mitigation strategies, and thus are typically considered fundamental to organizational security design and management. From a number of frameworks commonly implemented by industry [199], we review three in the remainder of this subsection, focusing on their ability to identify potentially useful exfiltration countermeasures.

4.1.1 Microsoft STRIDE Framework. One of the earliest cybersecurity frameworks is the Microsoft STRIDE security framework [104]. The STRIDE framework uses a 2-step approach to evaluate detailed system design in terms of security [184]. In step one, analysts should build a data flow diagram (DFD) to identify assets, dataflow, and the boundary of a network system in place. There are two major variants of using STRIDE [102] in this step:

- STRIDE per **element** [185] recommended highlighting the elements such as the external entity, the process, the flow, and the DFD data in terms of their behavior and operations
- STRIDE per **interaction** [97] suggested considering elements' origin, destination, and interactions (can better capture threats that are only visible in interactions between systems)

Next, in step 2 an analyst should determine the potential threat category of an entity, from several general known threats from which STRIDE is named after. The STRIDE general threat categories are as follows [85]:

- **S**poofing identity (Confidentiality/Integrity at risk)
- **T**ampering data (Integrity at risk)

- **R**epudiation (Integrity at risk)
- **I**nformation disclosure (Confidentiality at risk)
- **D**enial of service (Availability at risk)
- **E**levation of privilege (Confidentiality/Integrity at risk)

Using the STRIDE framework can be time consuming [185]. STRIDE uses the DFD to visualize every asset of an organization network system. As the scale and complexity of the organization increases, the total number of assets to be analyzed tends to grow exponentially. One study [171] hypothesized that it would be difficult to detect more than about two threats per hour during analysis. Another problem found by Scandariato et al. was that STRIDE leads to a roughly 25% false positive rate with around a 65% chance of missing a threat.

Mitigating the problems noted in the previous paragraph, STRIDE is relatively easy to adopt for organizations [184] and it is effective in identifying known threats [218]. Several studies suggested that combining STRIDE with other approaches, for instance, scores from CWE (common weakness enumeration) and CVE (common vulnerability enumeration) databases [85]; or combining STRIDE with NIST standards [124], can improve overall performances in terms of threat detectability and efficiency.

In general, the STRIDE framework provides organizations a structure of element identification and threat modeling. This defensive framework should improve all round security for organizations, but with large organizations the use of STRIDE can be time consuming. STRIDE does not exclusively list approaches that can protect against certain threats. Thus, other frameworks that have more granularity in terms of attack techniques in exfiltration threats also need to be considered.

4.1.2 Cyber Kill Chain. One of the most well-recognized threat models in industry is the cyber kill chain, which focuses on the offensive process. The cyber kill chain represents attack vectors as a sequence of stages, from scouting for information to the final action on objectives, in seven phases [91, 105]: Reconnaissance; weaponization; delivery; exploitation; installation; command and control (C2); actions on objectives.

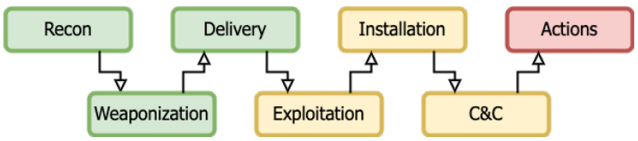

Figure 1. Cyber kill chain formulated by Lockheed Martin [119]

Attackers may not always follow this sequence in a linear fashion. It is possible that an adversary could have multiple campaigns working in parallel at different phases. The whole campaign is often initiated with social-engineering methods, in which it may skip a few phases. When defending against cyber-attacks a "cyber kill chain" approach is adopted (Figure 1) where each phase of the attack is seen as an opportunity to shut the attack down [119].

The cyber kill chain is capable of describing many types of adversary activities and provides a basis for detection and investigation [103]. It is commonly used in industry to support incident response, providing guidance to relevant stakeholders such as forensic investigators, threat hunters, malware analysts, and other "blue team" members. Focusing on the kill chain also supports collaboration amongst stakeholders [226].

There are different ways to implement the cyber kill chain concept. For instance, the diamond model was proposed to support "feature" exploration in each stage of the cyber kill chain [31] that can depict the core features of an intrusion (an **adversary** deploying a **capability** over some **infrastructure** against a **victim**). By pivoting through each stage and the core features, analysts can better identify the fundamental relationship between attack vectors and defensive approaches to

Table 4. An integrated view of cyber kill chain stages and potential countermeasures

Stage	Definition [226]	Countermeasures [31, 91]	
		Detect	Other Protecting Functions
Reconnaissance	Identifying, selecting, and profiling the target	Firewall	Deny access with Firewall Rules Deny with Access Control Lists (ACLs)
Weaponization	Coupling of remote access trojan with an exploit into a deliverable payload	NIDS	Deny transmission with NIPS
Delivery	Transmission of the payload to the target environment	NIDS User Training	Deny delivery with NIPS Disrupt with user training Degrade with email queuing/filtering
Exploitation	Triggering the payload on the target system	HIDS	Deny with proper patching Disrupt with execution prevention (executable black/white list)
Installation	Installation of backdoor and maintaining persistence	HIDS	Disrupt with NIPS Disrupt with Antivirus software
Command Control	Outbound internet controller servers to communicate with compromised host	NIDS	Deny with Firewall Rules Deny with HTTP Whitelists Disrupt with NIPS
Actions on Objectives	Network Spreading or Data Exfiltration	Audit Log Data Provenance	Deny with Firewall Rules/ACLs Deceive with Honeypot

protect against them. That relationship can also help identify countermeasures that are potentially useful at each stage of an attack campaign, for example, Table 4 shows approaches that may be useful in defending against exfiltration campaigns, including the stages involved and their action definitions.

4.1.3 MITRE ATT&CK Framework. The MITRE ATT&CK Framework for Enterprise aligns with the cyber kill chain model, while updating it with adversary techniques as they are developed and become available [109, 200]. It evolved from the cyber kill chain, focusing on possible tactics in and after the delivery stage, as shown in Figure 2.

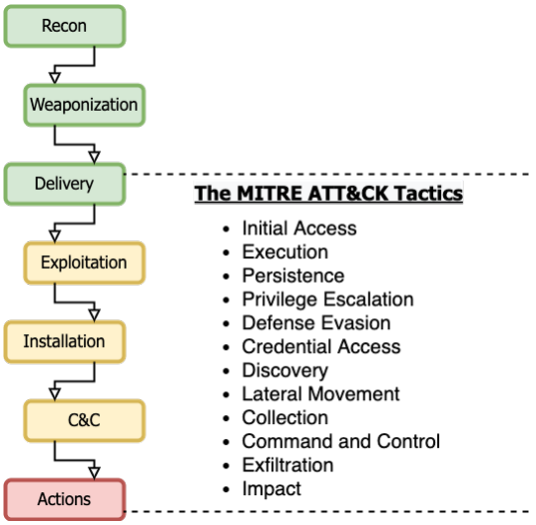

Figure 2. The relationship between MITRE ATT&CK tactics and the cyber kill chain

The MITRE ATT&CK framework focuses on the TTPs (tactics, techniques, and procedures) of adversaries, where "a tactic is a behavior that supports a strategic goal; a technique is a possible

method of executing a tactic. Each technique has a description explaining what the technique is, how it may be executed, when it may be used, and various procedures for performing it" [6].

Given an understanding of the whole chain of attack vectors that constitute a threat, one can predict future actions along the attack chain and develop strategies to deal with them. In the present context of data exfiltration threats, the possible tactics are listed as follows [131]:

- Automated Exfiltration
  - Traffic Duplication
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol
  - Exfiltration Over Symmetric Encrypted Non-C2 Protocol
  - Exfiltration Over Asymmetric Encrypted Non-C2 Protocol
  - Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium
  - Exfiltration Over Bluetooth
- Exfiltration Over Physical Medium
  - Exfiltration over USB
- Exfiltration Over Web Service
  - Exfiltration to Code Repository
  - Exfiltration to Cloud Storage
- Scheduled Transfer
- Transfer Data to Cloud Account

Note that the techniques listed in the exfiltration category of MITRE ATT&CK cover only the final step of an exfiltration threat, i.e., exfiltration of data out of the network.

The techniques incorporated within the MITRE ATT&CK framework are updated to reveal the latest attack vectors based on real-world observations [223], including knowledge concerning Advanced Persistent Threats (APTs). However, while the ATT&CK framework presents many adversary techniques, they do not provide guidance on how the techniques can be combined and applied. This can be a major issue because adversaries may blend multiple techniques together in order to accomplish the objectives [6].

4.1.4 Summary and Implications. The three frameworks covered above each have their own unique strategy for modeling threats. The STRIDE framework focuses on system elements (or interactions between elements) within the network from a defenders' aspect; the implementation of the cyber kill chain highlights important features to be explored at each campaign stage during an incident response or table-top exercise; whereas the ATT&CK framework provides comprehensive TTPs for better detection of offensive campaign and their paths [199].

We can recognize network assets and flows using the three frameworks, so as to identify potential countermeasures in each stage of an exfiltration campaign (with feature exploration), and search for every possible technique to be detected using the ATTCK framework. The countermeasures identified are shown in Table 5. This table updates countermeasures noted in previous surveys (e.g., surveys in Table 1 that covers various topics such as conventional countermeasures [209] and later ML solutions/countermeasures [164]) with the exfiltration countermeasures presented in Table 4.

Countermeasures against integrity and availability attacks are outside the scope of this study because we focus here on confidentiality attacks (exfiltration). Since we highlight the role of the human in dealing with software tools in this research, certain deceiving and degrading technologies

that generally work without human involvement are also excluded. Also excluded are some completely manual investigative technologies that do not involve automation. The final selected countermeasures are listed (Table 5) in three major categories: perimeter defense, data protection, and alerting and monitoring.

Table 5. Common countermeasures against exfiltration and their functions, traits, and limitations

Category	Countermeasure	Functionality	Trait and Limitation
Perimeter Defense	Firewall	Block request based on predefined rules/policies	(Passive) Operate based on predefined rules or signatures
	(Network) Intrusion Detection	Detect unwanted traffic based on pre-stored signatures	
	Access Control	Block/Grant access based on policies, roles, or attributes	
Data Protection	Encryption	Protect against data leakage for data at rest and in motion	(Passive proactive) Provide supporting evidence but require furtheralerting functions
	Data Provenance	Provide evidence of data modifications and transfers	
	Honeytoken	Trigger alerts of data modifications and transfers	
Alerting and Monitoring	(Host/Network) Intrusion Prevention	Detect unwanted traffic/activity and send out alerts	(Proactive) Constantly monitoring but can trigger a high volume of false alarms
	Endpoint Protection	Monitor normal/anomalous behavior on endpoints	
	Data Loss Prevention	Prevent unwanted traffic/process/behavior in the intranet	

The three "Categories" each represent a common security design strategy against exfiltration: perimeter defenses block unwanted access; data protection ensures that infiltrations that provides data access do not necessarily lead to information disclosure (e.g., a successful SQL injection attack may not necessarily yield information disclosure if data stored in the database is properly encrypted); and thirdly, alerting and monitoring strategies provide overall security both to the organizational intranet and to its core sensitive data.

In addition, the "Countermeasure" column in Table 5 arranges the order of logs, alerts, and prediction in ascending order of the degree to which they involve the expert in the process. These interventions will help a human expert establish customized IOCs (Indicators of Compromise), so as to form a "big picture" of the attack campaign and to "hunt threats". The whole process is human-centered to a large extent, but scant research has studied the importance of this critical human component in human-machine security systems. Thus, in the remainder of this section, we survey studies concerning our proposed research questions 1 and 2. We review the studies and technologies proposed and implemented in detail and introduce problems relating to the unacknowledged human component (in human-machine systems), such as those that arise when domain experts operate or consume information from these technologies.

4.2 Perimeter Defense

Technical countermeasures to protect against exfiltration have relied extensively upon perimeter defense as the primary layer of defense. Networks are often partitioned into public zones, demilitarized zones (DMZ), and private (restricted/controlled) zones with perimeters using firewalls, and within each network, access control rules and intrusion detection systems are placed to restrict access to allowed user/traffic only.

While perimeter defenses have been well understood for decades, they can nevertheless save human experts a great deal of effort since they function as a filter against unwanted user/traffic. Even when perimeter defenses fail, their logging functionalities may still be very useful in triggering cyber forensic investigation by human experts while also serving as a major source of input for machine learning (ML) models. Reviewing logs collected through perimeter defense systems may support establishing valid IOCs so as to stop an active attack campaign as early as possible or to prevent similar threats in the future.

4.2.1 Firewall. Network firewalls form the outer layer of perimeter defense between the untrusted internet and the trusted intranet, or between local network segmentation [92, 182]. These firewalls

restrict network traffic through accepting, denying, or dropping/resetting requests and thus significantly reduce the number of potentially malicious packets being passed into the organizational intranet. However, since firewalls are only effective when their rules are properly configured [220], and the rules are usually set to block known bad traffic, network firewalls are not fully effective at handling human-executed, novel exfiltration threats.

In addition to network and host firewalls, web application firewalls (WAF) are crucial in terms of protecting web servers [44]. Web servers are usually public facing to fulfill required business functions. They are consequently more vulnerable because they provide many opportunities for attack. As a result, web-based attacks such as SQL injection or cross-site scripting (XSS) are very common in modern computer environments [9]. A well-configured WAF may block web requests based on context, and/or sanitize user input for the sake of zero trust, so as to protect web servers from malicious attempts [207]. WAFs can also provide compensation controls when a major web server update is not deployable while some critical vulnerabilities have been published. Unfortunately, WAFs have similar issues as other types of firewalls because they all need preset rules or policies, thus making them less resilient.

Researchers have suggested using interactive approaches to increase the usability of setting up or re-configuration firewalls at a personal network level [177]. By creating an additional interface between firewalls and users, either visual or auditory, these tools help improve users' efficiency. However, interactive interfaces may sacrifice technical details, especially for personal use, sometimes undermining human-technology system performances [156, 157]. At an organizational level, while experts are willing and capable of handling complex security information, it is much more difficult to configure multiple sets of firewall rules or update them. Thus, interactive tools (e.g., supporting visualizations) are needed to manage complex system configurations [113, 122].

With recent advances in ML implementation, policy configuration data and rule updating at the backend have improved significantly. ML may support reducing errors caused by misconfiguration and increasing packet dropping accuracy, and, most importantly, reduce expert workload [3, 208]. Automatic models work well with human experts in this case, since anomaly rule detection and massive packet attribute inspection do not involve complex human behavior detection.

Experts may use firewall logs as an initial step in forensic investigation as well as threat hunting. Exfiltration threats, and associated malicious activities, may arise from disgruntled users who have legitimate accounts privileges, and whose exfiltration activity may only be detected when they attempt to transfer data out of the protected network. When data is exfiltrated, the firewall is the final opportunity to detect outgoing sensitive data. However, detecting such activities with firewalls at the perimeters may be too late. For this reason, access controls are typically used in combination with firewalls, and are configured to prevent both unwanted external users and insiders from reaching protected zones.

4.2.2 Access Control. In contrast to firewalls that control network traffic, access control systems limit user access to protected files, databases, or network zones. Starting with the early development of the access matrix [110, 173], various types of access control models have been proposed, with four models currently dominant in industry.

Initially, there were two major control strategies: discretionary access control (DAC) and mandatory access control (MAC). DACs use access control lists (ACLs) to manage whether a user should be assigned access (and define what operations can be made such as read and/or write privilege) to the requested resources [168, 170], based on their identities registered on the system.

While DACs are simple to configure and support timely updates to fulfill business needs, they are often vulnerable to impersonation or to certain types of malwares such as RAT (remote access trojan) [56]; since all the DAC restrictions are based on identities, DACs will not be effective

Table 6. A summary of the advantages and disadvantages of different types of access control models

Access Control Type	Advantage	Disadvantage
DAC (owner-controlled)	• Simple configuration through ACL • Current task-oriented • Support timely update	• A user may have excessive ACL settings • Vulnerable to impersonation • Difficult for centralized control • Prone to assign over or under privilege
MAC (lattice-based)	• Centrally manageable (object and subject labels) • Stronger enforceability • Single configuration for a group of users	• Less flexible when group-wise collaboration is needed • Centralized management cost
RBAC (hierarchical)	• Centrally manageable (user roles) • Least privilege yields better security • Easier to manage user roles than item labels (better flexibility)	• Large organizations may have complex employee structures and thus reduce the manageability of user role assignment • Multiple roles and access granted to one user may lead to over privilege
ABAC (granular and scalable)	• Centrally manageable (user attributes) • Dynamic and task-oriented • Highly scalable	• Difficult to define and manage attributes at the beginning

when someone impersonates another user. In addition, users with multiple identities may request resources from multiple identities on each system, making central management extremely difficult.

In contrast, MACs use labels to manage groups of resources (i.e. confidential, secret, top-secret), so that only a subset of users who have matching labels (clearance) can access. By forming a "lattice-based" control method, MACs are strongly enforceable and easier to manage centrally [141, 167]. However, if resources are required to share between groups, the highly restricted environment controlled by MACs may not be suitable. In addition, since labels are assigned to both users and the resources, it may be costly to set up a central management center.

Both DACs and MACs fail to satisfy the needs for industry practitioners [94]. Due to the defects listed above, role-based access control (RBAC) systems were developed, gradually becoming the dominant access control strategy. RBACs use organizational roles as the main basis for defining user privileges [63, 169]. Based on the organizational chart, roles can easily be assigned and reassigned to a user, and only when needed, leading to a guarantee of 'least privilege', at all times [64].

Since RBACs manage roles only (instead of both resource and user identities as is done with systems like MACs), the management cost can be significantly lower. However, in large multinational corporations with many thousands of employees, the disadvantages of RBAC became apparent. Business roles in very large organizations are complex and the business hierarchy may be unclear, increasing the complexity of managing roles, and increasing the chance of assigning undesirable levels of privilege to users with multiple roles.

Addressing the failings of other access control models, a more sensitive attribute-based access control (ABAC) was proposed [88, 144, 174]. ABACs rely on a top-down, uniformly controlled framework that defines every aspect of "everything" [134]. Attributes can be values including sensitivity of a resource, identity and context of a user, or even environment factors as long (as they can be further defined and applied as policies). If DAC, MAC, RBAC each represent a type of filter that can screen and remove based on its unique filter category, ABAC contains a great number of filters including, but not limited to, these three categories.

When constructed well, access control can be applied more easily and securely [94], with the marginal cost of adding instances or attributes. A summary of the advantages and disadvantages of all four types of current access control are presented in Table 6.

Maintaining a complex attribute framework and dynamically reassigning access may be as difficult as maintaining complex, distributed firewall rules. However, ABAC systems have a lot of

data regarding user attributes that could be extremely useful in terms of detecting unusual behavior by cross-referencing attributes [4] forming a strong basis for detecting insiders using ML.

4.2.3 Intrusion Detection Systems. While rule-based systems can detect malicious packets based on content inspection, current approaches typically carry out that detection using network intrusion detection systems (IDS). Network IDSs look for signature matches in web requests, emails, and other packets to detect malicious payloads that sneak through rule-based defenses [5, 45, 108, 221]. However, signature-based detections rely on a pre-existed database that contains known attack signatures. Since signature-based approaches are not able to detect novel threats, anomaly-based IDSs were proposed [230].

Anomaly-based IDSs perform content inspection by not only looking for signature matches but also by comparing the current profile with predefined "normal" profiles [68, 215]. IDSs then produce a numeric score (the higher the less secure of the system), usually between 1 to 100, representing how anomalous a profile is [133]. In this way, anomaly-based approaches are more capable of handling novel attacks in real time. However, anomaly based IDSs also have significant drawbacks. As shown in Figure 3, it may be difficult to match a single score of how anomalous a profile is to an attack pattern that is occurring in real time [111]. The anomaly score rises after an attack has begun and will fall once the attack has ended. Since the time-sensitive nature of attack profiles makes it difficult to assign a proper score, anomaly-based IDSs are prone to false alarms.

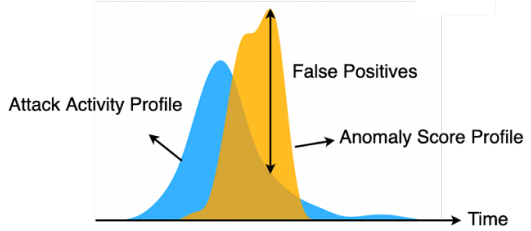

Figure 3. Mechanism of IDS scoring malicious payloads (originally Figure 2 in [111])

While there have been numerous approaches proposed to solve the excessive false alarm issue, especially with the increased use of ML algorithms [7, 39], industry reports (for instance reports in section 1) have shown that human experts are still overwhelmed by false alarms with no solution currently in sight. With little knowledge of the human factors of anomaly detection, research on the impact of current anomaly detection systems on human users in terms of user-centered testing and workload assessment is urgently needed.

Perimeter defense approaches employ a wide variety of methods to detect network-based attacks. They all, however, suffer from the disadvantages noted above. While perimeter defenses can screen out a large majority of attack attempts before they reach the intranet, they are less capable of combating exfiltration activities. As a result, defense strategies based on analysis of data usage within the intranet has become a focus for cyber-defense activity.

4.3 Data Protection

Rather than forming a "great wall" around valuable data, systems can seek to ensure that the data itself is difficult to be exfiltrated, trackable if modified/moved, or useless if not accessed by authorized personnel. There are three major ways to achieve these objectives that can be used in parallel/combination: encryption, data provenance, and honeytokens.

4.3.1 Encryption. Modern data encryption and decryption technologies originated in the two twentieth century world wars. Early development of encryption and decryption methodologies was

concerned with national security [53]. As the usage of electronic data sharing in industry started to flourish, a standard to implement cryptography algorithms publicly was needed.

The Data Encryption Standard (DES) was one of the first widely available (being tested and analyzed) symmetric-key algorithms (encrypting and decrypting with the same key) for data encryption. It was commonly used in businesses in the 1980s [190]. The DES standard ultimately proved to be insecure, due to its relatively short key length. The Advanced Encryption Standard (AES) thus was proposed to replace DES utilizing block ciphers and longer key lengths [50].

While symmetric-key algorithms have the merit of being efficient, they suffer from the fact that if the key is exposed during insecure transmissions, anyone could easily decrypt and access the plain-text. Thus, the concept of encrypting and decrypting data asymmetrically was proposed [55]. A widely accepted implementation of the asymmetric-key algorithm is the RSA public key encryption cryptosystem [162]. RSA utilizes the difficulty of factoring large prime numbers to generate a pair of keys: a published public key and a secret private key, where the plain text can be encrypted with a public key and decrypted with a corresponding private key. The concept of the asymmetric cryptosystem implementation is shown in Figure 4.

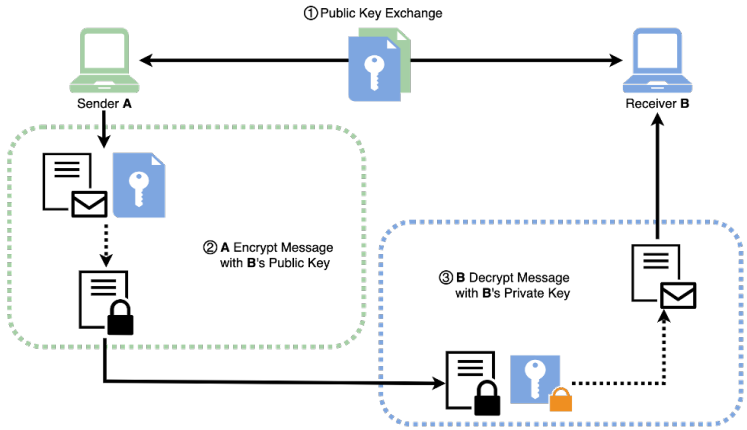

Figure 4. RSA public key encryption cryptosystem

RSA does not disclose the original material (plain text) if partial pieces of the ciphertext are exposed [71, 180]. RSA and its derived algorithms are currently considered secure in industry, until such time as an adversary obtains quantum computing technologies [33].

Encryption approaches focus on either protecting data in motion or protecting data at rest. Data in motion is usually vulnerable to man-in-the-middle attack. Encryption of data transmitted through the internet is crucial to prevent data leakage; for instance, the current TLS (Transport Layer Security) version 1.2 [54] secures web requests against eavesdropping. By contrast, protecting data at rest can be more difficult than protecting data in motion. In many cases, adversaries (especially insiders) may be more interested in stealing high volume of sensitive data at rest rather than small pieces of information in motion. It is thus important to label the sensitivity of data so that access clearance and records can be properly managed. There are several ways to classify data sensitivity. For instance, Executive Order 12356 [46, 150], describes three levels of information classification:

- Top Secret, where unauthorized disclosure could cause exceptionally grave damage to the national security
- Secret, where unauthorized disclosure could cause serious damage to the national security
- Confidential, where unauthorized disclosure could cause damage to national security

These three levels are proposed as a standard. There are many approaches complying with the standard so as to assign data sensitivity, such as using role and access patterns [129] to classify data, or using NLP (natural language processing) technologies to learn from text fragments and assign file sensitivities. Once data classification is completed, a data owner (usually a senior role who is responsible for data collection, protection, and data quality retention) can make decisions concerning the assignment of data access or editing permissions to users [225].

Many studies have been carried out on securing data in motion and data at rest using encryption technologies. However, cryptography itself is not sufficient to secure data in motion from man-in-the-middle attacks, and data at rest from physically accessing [211], its ability to stop exfiltration threats is limited in the following scenarios:

- Key stealing: Cryptography requires the secret key being protected securely (which usually rely on access control). Successful social-engineering attacks or impersonation can lead to key disclosure and sabotage data security.
- Data in use: legit users need to access clear text data for their day-to-day job. Spyware can easily record decrypted in-use data and thus cause data leakage.
- Insider threat: an insider with sufficient privilege can access original, unencrypted data at any time. Sometimes a user may unintentionally print out data that is supposed to be encrypted and secured at rest, thus leading to data exfiltration.

Thus, in the next subsection we consider data provenance as a supplement to encryption; data provenance keeps track of sensitive data location more effectively, protecting it against exfiltration.

4.3.2 Data Provenance. Data security constitutes an important aspect of the cybersecurity posture [13] of an organization. Data provenance is closely related to exfiltration threat protection, as it can provide reliable sources of evidence for domain experts as they form hypotheses to carry out investigations and build IOCs (Indicators of Compromise).

IOCs are indicator measures of whether a user account has been compromised. Accurate IOCs greatly facilitate threat hunting, allowing organizations to proactively look for malicious behaviors [126, 130]. Data provenance (sometimes referred to as the 'lineage' of data) provides data "labels" that can facilitate the process of building valid IOCs. It is thus crucial information for hunting novel or insider threats.

Implementing data provenance involves keeping track of data origins, as well as managing data arrival processes [29]. Conventionally, there are two ways of managing data provenance in a database [187]:

- Annotation: data origins and transfer points are 'annotated' in the metadata [22]
- Inversion: queries/functions used to derive data are stored and can 'inversely' reproduce source and derived data [99]

While both data provenance methods are readily scalable in modern systems [89], annotation can provide more information completeness. Current data provenance applications orchestrate various data sources. They are combined with other security approaches so as to detect anomalous events by tracking every possible modification (read, write, execution and transfer) of data files. Some data provenance application examples are:

- Monitoring data accesses and following on the chain of processes [107, 216]
- Providing tamper-proof function (using blockchain) to secure cloud data [116]
- Establishing trust so as to retain security status in the IoT (Internet of Things) environment, where multiple different metadata sources and formats are inevitable [57, 87]
- Integrating historical and contextual provenance data to triage false positives [1]

Data provenance can be obtained from system process calls [10], or can be obtained from email, print, copy (e.g., to removable drives), and any other traceable activities at a higher application/database level [61]. The collected provenance data should be secure from tampering, for instance, using provenance-aware platforms such as the Trusted Platform Module (TPM) [204]. Implementation primitives such as encryption, hash, signature, or watermarking [229] should also be considered, so that analysts can rely on the information for investigation. An interesting example of a secure provenance collection method is the Red Star system, developed by the North Korean government (according to a YouTube video cited in [121]). It is "an operating system that has been specifically enhanced to append "watermarks" based on the specific hardware being used. The receiving system can see the thread of previous systems that opened the file. In this case data provenance is secured and can provide non-repudiable information regarding who might be leaking files or creating "subversive" content.

With improvements in computational power, data provenance may contain more granular information (e.g., specific workbook in a spreadsheet file or particularly selected area in a table) that can more precisely indicate the causal relationship of events [80]. This can improve the efficiency of conducting investigations concerning the chain of exfiltration activities [60], which could also improve APT activity detections [93].

For large organizations, however, considering the number of files they need to secure, data provenance may create "too many" additional details. The problem of having too much data is much more salient than having too little data in modern threat detection, especially in a large corporate environment. Detailed data provenance can create huge amount of data as actions are tracked through a system. Like excessive numbers of false alarms generated in automated anomaly detection, data provenance threatens to create more information and potential threats than human analysts are able to handle.

Thus, it is believed that supporting experts, who are working on investigations using provenance with ML, may help them automate repetitive screening tasks, making their investigations less burdensome. ML models may support automatic threat detection using IOCs formed with low-level provenance data, transforming that data into enriched security incident knowledge, with a higher-level of abstraction, that is more suitable for human consumption [152]. However, when experts are trying to make critical decisions (e.g., determining whether an instance is malicious or not), ML outputs with low interpretability may do more harm than good. High-level abstractions may be unsuitable for people with high expertise, since the more expertise practitioners possess, the more "interpretability" they are likely to require in model output [30].

Experts need more explanation of model output, so that they can trust and rely on model outputs in making critical decisions, but too much explanation may be counterproductive. There is a tradeoff between the level of abstraction and the richness of model explainable outputs, with too much abstraction reducing expert trust in ML recommended decisions, while too much detailed explanation may be distracting and create inefficiencies. In addition, different experts may have varying requirements for model interpretability. Thus, the level of interpretability needs to be customized so that experts can trust the model and integrate model outputs into their decision-making process. ML models failing to fulfill these requirements may in turn reduce detection efficiency and create excessive burdens on human experts (A more detailed discussion of the expert-ML interactions is provided in section 5).

4.3.3 Honeytoken. A more aggressive way to protect sensitive data is through the use of honeytokens. Honeytokens evolved from the concept of honeypots. A honeypot is a decoy, a closely monitored network intended to trick malicious actors into providing insight into their techniques. Honeypots have the following advantages [132, 154, 193, 195]:

- Distract or mislead adversaries from valuable real targets
- Alert domain workers in advance
- Allow investigation of the vectors performed by adversaries
- Reduce false alarms (because activities performed in a honeypot are most likely malicious)

A honeypot acts as a decoy host that contains data that looks sensitive in order to lure adversaries to attack it, so as to detect the identities of the adversaries (in some rare but valuable cases) and their TTPs. A honeypot can also involve low or high interaction [217]. Low interaction honeypots emulate and monitor some specific services such as known Windows vulnerable services [12] and SSH server [47].

With low interaction honeypots, attackers cannot interact with the operating system directly. In contrast, high interaction honeypots support a more flexible interaction environment that can provide various types of data for investigation, such as tcpdump data, keystroke logs, file access details, and other input/output associated with adversaries' activities [217]. A high interaction honeypot might be insightful for analyzing comprehensive adversary attack vectors and creating IOCs to prevent upcoming attacks.

A honeytoken is an expansion of the honeypot concept, faking digital items such as credit card number, database entry, or credentials [194], making them quasi-authentic, and placing them in the system within the intranet [21]. Two major ways of creating honeytokens from database rules are [227]:

- Obfuscation: substitute sensitive attributes and their values with artificial data
- Generation: completely generate artificial data from scratch

High definition honeytokens should be indistinguishable even with extensive efforts performed by domain experts [196]. Thus, they can be used to trigger alarm when someone tries to interact with certain rarely accessed database entries [148]; to keep track of the fingerprint (similar to provenance) of an active attack campaign [196]; or even protecting 2 factor authentication (2FA) with injecting honeytokens as words into credentials [143]. Whenever a honeytoken is accessed, used, modified, or transmitted, an alarm will be triggered to notify relevant personnel. Proper alerting and monitoring technologies must be prepared in advance to deal with Honeypot data and honeytokens.

4.4 Alerting and Monitoring

With some exceptions, passive rule-based, signature-based, and anomaly-based detection approaches have been implemented in a way that requires human experts to be proactive in their investigations (hunting potential threats). Relying solely on passive protection puts undue load on human resources. As a result, approaches to continuously monitor endpoints, networks, and databases have been implemented. In this way, it is possible to alert corresponding personnel with timely and relevant information, in order to improve expert-machine collaboration efficiency and reduce human costs.

4.4.1 Intrusion Prevention and Endpoint Protection. Host-based firewalls and IDSs can detect policy violating processes at endpoints using real-time signature matches [37, 142]. By obtaining operating system audit data, host-based approaches provide better granularity than network-based approaches, and thus can perform better in internal attack detection [95, 117]. On top of the reactive/passive detection functions with firewalls and IDSs, the concept of Intrusion Prevention System (IPS) was proposed to alert human experts in a timely fashion while isolating threats [232].

Host IPS approaches can be expanded, so as to monitor processes across endpoints and unify with different data sources. Such approaches are called Endpoint Protection Platform (EPP) and Endpoint Detection and Response (EDR) systems [35]. EPPs integrate signature-based and anomaly-based approaches to detect anomalous activities on endpoints, such as irregular memory consumptions

[138]; using whitelist/blacklist rule enforcement to prevent novel attacks from executing other program; and eliminating potential malicious processes to control damage from spreading to other hosts on the same network segment.

EDRs extended EPP approaches by integrating cutting-edge technologies, such as ML-infused detection using real-time IOCs [35]. They monitor endpoints across an organization's network and provide visibility to human experts. Thus, EDRs can discover covert anomalous activities through comparing endpoint activity profiles.

In addition to system calls, processes, and audit events, EDRs use the User Entity Behavior Analytics (UEBA) platform as a major data source concerning human behavior. UEBAs focus on detecting anomalous user behaviors on enterprise endpoints [159], in which examples of anomalous behaviors can be multiple login retry, unusual access location/IP, large outbound email attachment, file printing activity, unrecognized program execution, intense activity before termination, etc.

UEBAs can use time series data from endpoints to detect novel insider activities by classifying (and visualizing) chains of human behaviors [101, 181].

In modern enterprise environments, endpoint events are typically managed centrally, using an SIEM (Security Information and Event Management) . SIEM is a technical solution for data centralization and visualization. SIEM aggregates activities collected from sources across networks and endpoints, so as to help administrators implement security policies and manage events/alerts centrally [158]. For larger organizations, a SIEM is sometimes replaced by a more advanced XDR (Extended Detection and Response) system, more prevalently referred to as a SOAR center (Security Orchestration, Automation and Response).

A SOAR can be considered as a SIEM with enriched data from a larger variety of sources. SOARs usually require higher adoption costs [52], but the integration efforts to build a SOAR usually leads to better AI implementation later on in large organizations. Figure 5 shows the relationships among EDR, UEBA, SIEM/SOAR, as well as other approaches mentioned earlier (honeytokens should be placed in the data protection block).

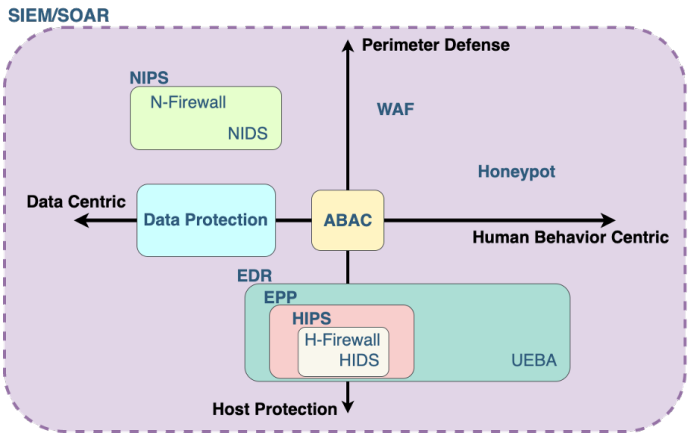

Figure 5. A quadrant diagram of SIEM/SOAR data sources and their relationships

Figure 5 summarizes countermeasures that may support detection against exfiltration threats, where each colored block represents a type of data source that can be used in further investigation and threat hunting. Among the countermeasures, UEBA provides a relatively complete human behavior information profile that can be used in cross-endpoint EDR investigations and incident responses.

Centrally managed endpoint protection approaches require experts to work with their rich functions and data sources proactively. Analysts working with these platforms can respond to

anomalous events in real time. However, for platforms focusing on human activities, this can be a disadvantage due to the nature of unpredictable and novel human behavior. Users on endpoints do not always operate with certain fixed patterns. Thus, numerous alerts can be generated as false positives [206]. Consequently, these platforms may cause fatigue, overwhelm, and reduce situational awareness of human experts because of the well-known alert fatigue phenomenon [1, 15]. Alert fatigue in turn leads to human-machine system performance degradation and undermines overall security performance with a canonical example of poor human factors outcomes due to alert fatigue being the case of the Three Mile Island nuclear incident [25].

4.4.2 Data Loss Prevention. While large numbers of false alarms can be burdensome for human experts, one approach to reduce the number of false alarms is by lowering the sensitivity of detection and focusing on the final exfiltration actions. Because every exfiltration campaign has a final exfiltrating action, organizations can focus on preventing this final step by applying business functions (i.e., a Data Loss Prevention. or DLP, system) that define acceptable vs. unacceptable actions.

A DLP can inspect file contents and block policy violating actions preceding outbound traffic, so as to prevent sensitive data from leaving the intranet [205]. This should significantly reduce alerts being presented at a SIEM, reducing human workload. Many vendors supply DLP solutions to organizations [79]. At a minimum, a DLP system should provide the following functions [118]:

- **Define** data sensitivity to create a data inventory that contains sensitive data location
- **Discover** sensitive data at rest and relocate the data to logged secure inventory
- **Manage** data usage policies and how they are enforced, including data handling such as data cleanup and disposal
- **Monitor**, understand, and visualize (make visible to the organization) sensitive data usage patterns
- **Prevent** sensitive data from leaving an organization by enforcing security policies proactively.
- **Report** data loss incidents and establish incident response capability to enable corrective actions that remediate violations

While it sounds straightforward to "block outbound sensitive data", sensitive files can be dynamically created and deleted constantly, making it difficult to track which data is sensitive. If sensitive data is not tracked adequately, the DLP may fail to block transfers that should be blocked, undermining security, or may block too many transfers, undermining system service quality [222].

Since DLP systems operate using rules, they are subject to the same problems (noted earlier) as other rule-based systems. To block sensitive files from leaving the intranet, DLP requires certain policies/rules to operate properly, based on how the following questions are answered:

- What kind of actions should be blocked?
- Who (which privilege), when operating what, should be blocked?
- How to block?

As the scale of the organization increases, it can be more difficult to answer these questions, making the defined policies more complex. As a result, a DLP system following these complex policies can in turn generate a large volume of false positives.

4.5 Social-Engineering Attacks

As discussed in section 1, social-engineering attacks have become one of the most common attack vectors used by adversaries and are a major producer of exfiltration threats. Thus, we describe some previous surveys and reports in this subsection to highlight the need to handle social-engineering attacks and to raise more awareness of this topic in relation to data exfiltration. Note that combating

social engineering attacks involves human factors issues associated with user behaviour. However, our focus in the rest of this review is on human factors issues when domain experts examine accounts that are possibly compromised (often due to a social engineering exploit).

Social-engineering attacks usually do not follow the conventional kill-chain path, but rather, adversaries leverage sophisticated reconnaissance on victim's publicly available information (also known as the offensive OSINT) to obtain valid credentials. A social-engineering attack campaign usually focuses on developing the user/victim's trust, and then exploiting that trust [2]. One of the most common social-engineering attacks is a phishing attack. People often blindly follow instructions on a masquerade email or text, and provide their credentials (or any other valuable information), because they are misled to believe that the sender is legitimate [120]. Conventionally there are two types of countermeasures to handle social-engineering attacks [165]:

- Computer-based (software, system, tool)
- Human-based (training, educating, situation-awaring)

Computer-based countermeasures utilize the methods discussed so far (sometimes with slight amendments) such as; rule-based blacklisting or whitelisting, signature-based malicious URLs detection, alerting/monitoring email activities to put a banner notification on external unknown senders, etc. Software tools can efficiently prevent social-engineering attacks before they reach the human target. One such protection against phishing attack is multi-factor authentication (MFA). MFA blunts the impact of social engineering-based attacks, since it is based on attributes that are hard to acquire by a third party in addition to attributes that a user knows (such as passwords and pins, which are easier to acquire for purposes of spoofing a legitimate account holder). MFA involves:

- Something you have (such as a device or an ID card)
- Something you are (such as biometric information)

In contrast, human-based countermeasures focus on the human factors of potential human targets. An organization might enforce mandatory training sessions to educate internal network users regarding how to identify social-engineering attacks so as to improve their awareness. Sometimes an organization may insert its own pseudo-phishing emails into user mail queues to detect the susceptibility of those users to social engineering attacks. However, organizations remain susceptible to social engineering attacks whenever they are feasible, due to a variety of human foibles such as over-trust, impulsiveness, or greed. The vulnerabilities of human nature have made humans "the weakest link in the security pipeline", a weak link that is easily taken advantage of [172]. Human slips/errors may weaken human-based protection, and consequently, undermine the effectiveness of computer-based countermeasures.

In recent years, social-engineering attacks have evolved. Social-engineering attacks may no longer obtain access to a network system, but simply deliver a malicious payload. The delivery process can be covert (e.g., the recent Excel macro malware attachment attack reported by Fortinet [231]), and the goal is only to install ransomware onto the target system. The adversary can then demand a ransom and threaten sensitive information disclosure, as presented in reports in section 1 [49, 127, 147]. This new type of attack is even more difficult to prevent because one negligent or careless employee can cause severe damage to the whole intranet.

Hardening the system network against social-engineering attacks can be difficult. Domain experts must protect not only the computer network but also human interactions with the computer network. This has become a socio-technical issue, where there is a lack of comprehensive guidelines to support their work. The cybersecurity domain urgently needs more investment in training people in order to enhance their social-engineering attack awareness [165]. More advanced detection countermeasures to battle social-engineering attacks are also needed.

4.6 Summary of Countermeasures

Many countermeasures have been proposed to protect organization networks from exfiltration campaigns. These countermeasures support detection and provide other protective functions. They also provide detailed, informative logs for further investigation conducted by human experts. However, as noted in the preceding sections, large amounts of data, and associated alerts and notifications, do overwhelm human analysts.

Although many researchers have focused on the algorithmic aspects of protecting against data exfiltration, human analysts remain at the core of what are effectively socio-technical system. Human experts carry out tasks such as:

- Constructing system perimeters and administrating privileges
- Implementing detection sensors and deploying alerting functions
- Building IOCs and interpreting logs
- Investigating anomalies and making final decisions

While automation through Machine Learning (ML) algorithms may handle repetitive "screening and filtering" subtasks, critical decisions cannot be made solely relying on model outputs, especially when model interpretability as well as performances (i.e., too many false alarms) are questionable. In addition, analysis of cyber threats, especially exfiltration threats that are sometimes performed by insiders, involves many variables that are latent, or that represent behaviors and implicit knowledge that is inaccessible to algorithms and ML models. Thus, both detecting and investigating tasks are dependent on human experts' implicit knowledge of the organization concerning its business functions and members' normal behavior profiles, and thus the human role in protecting against data exfiltration must not be ignored.

After extensive review of the relevant research literature and industry reports, it is clear that there are few studies focusing on supporting the human role in exfiltration threat countermeasures. But implementation of exfiltration countermeasures raises complex socio-technical problems and thus the human role needs to be given more emphasis. In the following section we survey research concerning the human role in automated ML systems in general, noting the limitations in our current knowledge and the need for more research concerning the human role in future. While our focus in the following section is on the human role in machine learning and in cybersecurity in general, the issues raised will apply more broadly to human interaction with automation, and more specifically to data exfiltration applications.

5 HUMAN ROLE IN MACHINE LEARNING SYSTEMS

Advances in machine learning algorithms have made ML an essential part of cybersecurity countermeasures. As was discussed earlier in Sections 3 and 4, the human factors of expert-automation interactions have not been thoroughly considered in the research literature on data exfiltration. The role of the human expert or analyst continued to be ignored after ML models were utilized in exfiltration countermeasures. ML may actually be making human interactions in data exfiltration countermeasures less efficient. ML deployments require cybersecurity experts in industry to acquire a new skill set. In addition to requiring new skills, applying automated ML in cybersecurity may increase the workload of experts. In this section, we discuss SIEM (or SOAR) systems introduced earlier (Section 4.4.1), demonstrating the need for more attention to be paid to the human factors of how domain experts interact with automated ML models.

5.1 SIEM Integration with ML and Resulting Implications for Human Factors

Modern enterprise environments use a SIEM (or a SOAR) approach to integrate and centralize complex data for the purposes of real-time attack detection and security event analytics (typically

within a SOC, a Security Operations Center). SIEM systems provide log data collection and integration functionalities, supporting expert investigation, forensic analysis, incident response, incident mitigation, and reporting [100].

A SIEM tool works on data logs from a variety of security devices and traffic sensors [23]. These devices and sensors can be the types of countermeasures discussed in section 4, such as firewalls (including WAFs), IDSs/IPSs, authentication servers, and endpoints. There is usually an executive SIEM that shows the overall behavior and risk associated with each device and sensor. Unresolved events can then be triaged and highlighted using colors representing different threat levels [106]. In this way, a SIEM can visually guide the expert to resolve the most urgent incident. The integration of multiple data sources also helps, giving a "full picture" of the attack pathway/campaign including other targets or areas that may be affected within the network system.

SIEMs utilize visualization intensively (and not just in executive dashboards) to visually support experts in their search for anomalous patterns [139]. In contrast to other tools used by domain experts, SIEM tools tend to follow human factors guidelines more closely. Integrating SIEM systems with ML models may also lead to better categorization of network traffic and prediction of attack patterns [28, 228]. With the help of ML technologies, incident responders should be able to both obtain required information more efficiently, and isolate the compromised zone in a timely manner.

While studies have shown the usefulness of SIEM tools, SOC implementations in industry are often not ideal. Chamkar et al. conducted a survey with 45 SOC analysts/SOC service providers [34] and found deficiencies in automation and data orchestration (97%), visibility concerning IT security infrastructure (95%), appropriate methods to handle false alarms (93%), and guidelines or playbooks (92%). They also found a general lack of: training and attack simulations, knowledge towards business risks, and adequate evaluation metrics, etc., in the SOCs that they studied. Meanwhile, a study [72] showed that there are only few off-the-shelf SIEM systems that have ML functionalities. The level of cybersecurity automation is currently far less automated than the level of automation studied in academic settings. Thus industry faces a situation where there is a considerable amount of manual (human) task activity in cybersecurity countermeasures but without the requisite consideration of human factors issues.

How can we learn from this situation, and develop improved methods, not just for SIEMs, but for all countermeasures in dealing with the threat of data exfiltration, and more broadly, within the domain of cybersecurity. The promise of ML will not be fully realized if solutions are not engineered with the properties of humans clearly in mind. In the following discussion we consider four major human factors issues that have been prominent in a range of domains from nuclear power to aviation and healthcare. We will use SIEM tools to exemplify the problems here and will then further elaborate them in later subsections. Thus four key human factors problems are:

- Expert availability
- Situational awareness
- Trust and reliance
- Human-System Compatibility

Expert availability is a highly salient human factors issue for SIEMs. Experts are expensive, and difficult to hire because of security knowledge shortages in the market [151]. Thus, human experts are a precious resource and their time should not be wasted. However, SIEM deployment currently relies on writing ad-hoc data collectors and compromise indicators case-by-case. This makes it difficult for domain experts to keep track of large volumes of data [41]. In contrast, situational awareness is usually well-considered in SIEM tools, which are typically constructed to promote situational awareness [62]. However, interpreting SIEM dashboard outputs can be challenging. Few studies (subsection 5.3) have covered this issue within the domain of cybersecurity. SIEM tools are

widely used in attempting to automate decision-making processes [72], but the problem of setting appropriate levels of trust and reliance for human experts has not been considered, neither have human-system compatibility issues been discussed, although they are coming to the fore in other ML application areas [17, 18].

In the remainder of this section we briefly review the role of human experts in human-model systems as characterized in the previous research literature. This review will help identify problems associated with implementing automation/ML in the domain of cybersecurity against exfiltration threats, and will address our earlier research question 3 that concerns the actual benefits/limitations of countermeasures, considering human users, organizational structures, and other socio-technical factors.

Prior to reviewing each of these human factors in the following subsections, we will briefly characterize the opportunities for including human expertise in various stages of the ML model training process:

- In data collection: human interaction is involved in the collection of past events, the process of use cases creation in simulation technologies, in the setup of honeypots, etc.
- In data pre-processing: human interaction is involved in defense system building, cyber kill-chain design, system patching, rules/policies creation, signature databases maintenance, data labelling, etc.
- In detection process: human interaction is involved in knowledge input, discussion between domain experts and ML experts, and related activities
- In results and analyses: human interaction is involved in reading output, investigations, resolving alerts, and making different types of judgements

The human role is important throughout the monitoring and detection process, but it has rarely been considered in past research and that role has been poorly defined. As a result, the outputs provided by ML models and software countermeasures will often be ignored or misinterpreted. This deficiency should be addressed, and human factors should be considered in designing detection algorithms. While human factors issues are sometimes considered out of scope in highly automated systems, they will start to come to the fore in strategic decision-making concerning the selection and preprocessing of data, and in model training.

While we noted four human factors issues in this section, we will conclude by recognizing that the essential difficulty in defining the human role in combating data exfiltration, and perhaps in cybersecurity generally, is that humans work very differently from algorithms and have very different input and output requirements. While there may be some recognition of this fact at a conceptual level, we are a long way from dealing with it in operational settings. The following subsections review the four human factors problems listed earlier as a necessary step towards defining more appropriate and useful roles for humans in an interactive ML process.

5.2 Human Expert Availability

Expert availability is an important constraint when deploying an automated learning model in cybersecurity. We focus here on the workload generated by expert investigations triggered by ML detection processes (including model training and testing). There are two ways of introducing ML models to an organization: using off-the-shelf models or designing a customized model.

While using off-the-shelf models may seem easy and direct, model outputs may not be compatible with conditions in some organizations, creating extra work for domain experts who then need to perform testing, debugging, and patching. However, building customized models is not a task that an ML engineer can complete without involving domain experts. The required extensive discussion

of model goals, and reviews of multiple iterative updates, can significantly increase domain expert workload.

The relative lack of domain expert availability (in comparison to the needs for expert input) also limits the effectiveness of ML methods that rely on training processes, where the human experts label instances. Active Learning (AL) can improve training by providing more efficient human expert labelling [175]. In AL, instances that the ML prediction model is more uncertain about are preferentially presented for labelling, with the goal of making the prediction process converge towards more accurate modelling more quickly [176]. However, while AL has been tested and applied in a wide variety of non-expert labeling tasks, its performance has not been thoroughly studied with labeling tasks that require expertise (i.e., experts may not always be able to confidently provide "correct" labels). This gap in the literature concerning when and how AL should be used thus requires better ways to deal with limited expert availability in cybersecurity applications.

In complex scenarios (for instance detecting unintentional email exfiltration), good quality labeling may not be sufficient. Well-trained anomaly-based ML models may still generate too many alerts, demanding excessive amounts of time for expert review. As an example, an excessive number of alerts was one of the aggravating factors in the Three Mile Island near melt-down [146]. Dealing with too many alerts may create "alert fatigue" [32]. Alert fatigue has been observed in a number of different domains including healthcare, aviation, and oil drilling [36]. Alert fatigue can be lessened by reducing the number of alerts and/or making alerts easier to deal with.

One strategy for reducing the number of alerts that need to be processed is to cluster them into meta-alerts [78]. In this way, numerous alerts can be classified, so that experts do not have to investigate each of them one by one, but instead, can look into alerts and resolve them as clusters. This is a good example of changing the way that information is presented to experts to make it easier for them to process. Aside from changing the content presented to experts, it is also possible to change the look and feel of the interaction through interface design. Interface design is a crucial determinant of system usability. For instance, visualization may be an effective way to present data patterns in context [213]. Collections of principles and guidelines for HCI design include Nielsen's general rules [137] and Gerhardt-Powel's principles [69].

Another important aspect of interface design in cybersecurity is (machine) explainability of system decisions and actions. Explainability reduces workload by making it clear to experts why the system is performing as it does [76, 82]. However, as mentioned in section 4.3.2. there is a tradeoff between the level of abstraction and the richness of model explainable outputs. Experts may not be able to work effectively without properly presented output from ML models [214].

In summary, current methods place too high a load on scarce human analysts and experts. Thus, methods are under-utilized, and even when they are utilized, their results/findings are not implemented effectively due to a shortage of people who can check them or put them into practice.

5.3 Situational Awareness

Another topic that should be considered when applying ML approaches in cybersecurity is experts' situational awareness. Situational awareness is traditionally defined as "the perception of the surroundings and derivative implications critical to decision makers in complex, dynamic areas such as military command and security" [58]. Maximizing situational awareness may guarantee "operational risks to be mitigated, managed, or resolved prior to a mission or during operations" [125].

Barford et al. [19] used the term "cyber situational awareness" to refer to the application of situational awareness in cybersecurity, where there are seven major requirements that describe what domain experts should be aware of to make their cyber network safe (of which the following four will be considered here since they are relevant to our concern with expert-model interactions):

- Awareness of the current situation (also known as situation perception)
- Awareness of adversary's behavior (the trend of the attack)
- Awareness of the quality and trustworthiness (of the collected situation awareness information items and the knowledge-intelligence-decisions derived from these information items)
- Awareness of plausible future evolution (from the current situation)

Cyber situational awareness can be reached when these requirements are met, and when data collected from sensors can be directly interpreted into expert-readable information [188]. This requires a bridge between the cyber layer and the physical layer, which in our point of view, is an interactive model. The SMART 2.0 proposed by Snyder et al. is a good example of showing how an interactive learning model can connect cyber data with human cognition, boosting situational awareness, as well as model training [192].

Unfortunately, current ML communities focus more on automating the detection and alerting processes rather than integrating experts with situations that arise in the cyber layer. There has been insufficient consideration of how algorithmic outputs will be interpreted and used by domain experts when combating data exfiltration threats.

5.4 Trust and Reliance

A third human factor, trust in ML models, may have a major impact on expert-model team performances. Trust in automation is a requirement of working with and using machines. Aviation is a good example of this. In the past century or so, the perception of flight has gone from flying as a dangerous activity carried out by trained specialists who accept the known risks, to a routine activity that is safer than driving, although not always perceived to be as safe [189].

In earlier human-machine teams, the performances of human-machine collaboration and the definition of "who is in charge" of the team were largely affected by the trust from human operators to the machine and the self-confidence to themselves. The more they can trust in machine capabilities, functionalities, and robustness, the more the automated process can be carried out by the machine itself without manual interventions [114]. This led to a model of supervisory control [183] where the human collaborated with the automation, ceding varying degrees of control authority to the automation, from complete control (e.g., being a passenger in a vehicle) to assistance with aspects of the task (e.g., cruise control in an automobile).

In practice, machines are becoming more capable, and thus there is increasing automation with humans handing more tasks to the machine. This process is particularly salient in the case of automated vehicles, where there are associated human factors issues as drivers become supervisors and where they are often faced with distracting technologies in the vehicle [77]. Thus, over-trust, or over-reliance, on machines can be problematic, and it is crucial to measure the trust and reliance from humans to the machine [115] to make sure the trust boundary is always clearly defined and used to constrain design inputs and outputs for ML models.

5.5 Human-System Compatibility

Lastly, for highly professional domains like cybersecurity, the relationship between humans and machines is circumscribed. In cybersecurity, model outputs have to be verified by expert investigation or cross-departmental discussion concerning the authenticity of suspected breaches. The role of the machine, an ML detection model for instance, is to support experts making judgements. The machine works like an advisor giving directions and suggestions but without making final decisions. This change in role necessitates re-consideration of which metrics should be used when evaluating ML performances in domains like cybersecurity because model evaluation metrics may not reveal human-model team performances [17].

For example, with respect to detection model updates, ML experts normally focus the evaluation on detection accuracy and seek to improve the precision/recall tradeoff. However, improvements due to the model update might also lead to a change in feature weighing or a re-tuning of hyper-parameters, without this information being disclosed to the actual users of the model, the domain experts. Thus, becoming under-informed of the strategy and tactics of the model, they may find it harder to accept model outputs leading to less trust in the system. As a result, the model might be getting objectively better, but the human-model team may end up performing worse [18, 40] because the compatibility of the human-AI team has decreased, and the ultimate decisions may be based on an incomplete understanding of the situation.

In addition, providing excessive, explainable model details to the human can lead to another "obedient" problem. For instance, Bansal et al. showed that despite many studies suggest that explainability of model outputs may help improve human-ML system performances, the excessive explanations are more likely to increase the chance that a human participant may "blindly" accept the recommendation from the machine without thoroughly considering its correctness [16]. The overall system performance improvements are only contributed from the model performance improvements, where the human participant is merely a "rubber stamp". This can be a signficaint issue in applying ML in cybersecurity; because the human component is now experts making critical decisions, and explainability may in turn confuses them. Expert-ML systems and their compatibility thus are yet to be studied.

5.6 The Human Role in ML and Cybersecurity Applications

Human factors issues will be relevant in cybersecurity applications as long as humans are "in the loop" and part of the decision-making mechanism [48, 83]. We have not yet reached the point where large organizations are willing to rely solely on ML algorithms to defend against data exfiltration. In practice, that point may never be reached, since the absence of human intervention may be used in litigation to extract greater damages by lawyers representing parties who have been damaged by a data exfiltration incident. When automation fails, the obvious criticism is "why wasn't there a human in the loop to check that everything was ok?" Similar considerations mitigate against the use of fully automated aircraft or trains. No matter how good a model is, it has to operate within the constraints of our increasingly complex socio-technical systems [43].

6 RECOMMENDATION AND FUTURE RESEARCH

The material presented in previous sections of this paper has reviewed problems with current data exfiltration countermeasures, and has identified a need for greater consideration of human factors issues in this area. Domain experts have a large amount of implicit knowledge that is not recorded in the data available to ML algorithms. Much of this knowledge is "compiled" and difficult for experts to verbalize [145]. However, with suitable interfaces and tasks, experts can reveal this knowledge when they answer timely questions in appropriate contexts.

In a complex environment, limited human bandwidth and attentional resources make it difficult to maintain adequate situation awareness. For an organization that may have millions of interactions running across its network each day (or even hour or minute in some cases) the problem of maintaining situational awareness becomes increasingly challenging. ML, data visualization, and other computer aiding methods can provide situation awareness and highlight the most important features of the current situation, but that highlighting has to be done carefully, so that the information is presented to human experts in a way that matches their needs and capabilities, as well as their expectations in the particular context.

Providing the right information at the right time will also help manage the mental workload of domain experts. Without proper interaction design between experts and ML algorithms as well as

their outputs, there is typically a significant stream of alerts representing possibly anomalous cases, and the domain expert needs to try and prioritize the alerts and sift through them. Prioritization is necessary because with so many alerts it is not possible to deal with them all. Like an understaffed call center with the phones always ringing, the expert is besieged by more alerts than can possibly be handled, leading to stress as well as high workload. Thus, it is critical to offload the routine handling of alerts so that the expert can handle the highest priority alerts, for instance, those that need to be interpreted with human expertise. Note that the human interaction with the ML algorithm will involve not only sorting through high priority alerts, but also training the algorithm(s) with labelling advice, feature weighting, and other activities.

Perhaps the greatest challenge of expert-ML systems is creating compatibility between humans and ML algorithms [17]. In the case of deep learning, compatibility is particularly challenging because it is difficult to translate the weights assigned to the many processing units ("neurons") in the network into simpler concepts, relationships and general weightings of importance that are easily grasped by humans. However, the problem of opacity in neural network outputs is well known, and research is ongoing into how to make approaches such as deep learning more consumable by humans. In practice there may be a tradeoff, where domain experts and managers may be willing to trade off a certain amount of model accuracy in return for greater interpretability. Thus, there have been attempts to break down deep learning models by providing representative explanations for insights [161]; or by utilizing local linear models to approximate detection boundaries near the input instances, so as to help select key contributing features [75]. Regardless of the approach used, humans need to remain in-the-loop to read results and make decisions about how to update or apply models in the future.

In a domain like cybersecurity, where intensive situation-awareness and trust is needed, the compatibility issue is always likely to be a problem. An interactive machine learning (iML) approach that can directly address this issue by iteratively updating the training data based on human input and by making the model's logic more transparent is needed, so as to both hand control back to human users efficiently and avoids the problem of unrecognized model brittleness [191] involving states or cases where the model training is insufficient, and the model predictions cannot be trusted. However, further studies are required before implementing such models in this critical domain.

7 CONCLUSION

The ever-growing threat of costly data exfiltration events has led organizations to recognize data security as a major imperative. Unfortunately, efforts to secure the perimeters of organizational networks have not adequately addressed the threats posed by insiders, either those who have legitimate roles inside organizations, or masqueraders, who have obtained insider credentials (e.g., through phishing). Since there are many data exfiltration threats and knowledge of human behavior is an essential part of analyzing these threats, previous algorithms that have relied exclusive on ML based detection, followed by human review of alerts, have fallen short because they have not addressed the full complexity of data exfiltration scenarios or relevant human factors issues. Thus, there is a need to create a more active role for human experts throughout the process of detecting data exfiltration activities. The assistance of human experts is relevant across the exfiltration detection lifecycle, from data logging, rules creating, and debugging, to resolution of alerts and performance of investigations. The need for vigilant detection methods will continue regardless of whether sensitive data is stored in the cloud or within a network hosted by the organization. In spite of efforts to prevent cybersecurity threats using new approaches such as zero trust architectures [163], data exfiltration will continue to be a threat for the foreseeable future and it is part of the fiduciary responsibility of organizations to include strong detection methods, as well as prevention methods, in their defensive arsenal.

In a domain that is rapidly adopting state-of-the-art automation methods, the importance of expert knowledge in detecting data exfiltration events has been overlooked. In this paper we addressed this issue by 1) surveying industry reports and previous studies to emphasize the urgent need to place experts in-the-loop while creating automated models/systems; 2) documenting the failings of current countermeasures and explaining why those failings occur due to inadequate consideration of human roles; 3) describing why it is crucial to connect algorithms and experts together, and emphasizing the need to improve the human factors of the domain expert work flow.

Cybersecurity applications that include a role for human experts are necessarily socio-technical systems and cannot be safely and efficiently operated without considering relevant human factors issues. In this paper we have not only provided a state-of-the-art review of data exfiltration countermeasures, but have also provided insights into the human factors that need to be addressed in future research.

ACKNOWLEDGMENTS

Mark Chignell acknowledges support from Mitacs grant IT30559, "Detection and Investigation of Email Exfiltration Events in Sun Life Cybersecurity Data". David Lie acknowledges support from a Tier 1 Canada Research Chair.

REFERENCES

[1] 2019. Nodoze: Combatting threat alert fatigue with automated provenance triage. *Network and Distributed Systems Security (NDSS) Symposium 2019* (2019).

[2] Islam Abdalla and Mohamed Abass. 2018. Social Engineering Threat and Defense: A Literature Survey. *Journal of Information Security* 9 (2018), 257–264. https://doi.org/10.4236/jis.2018.94018

[3] Qasem Abu Al-Haija and Abdelraouf Ishtaiwi. 2021. Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense. *International Journal on Advanced Science Engineering and Information Technology* 11, 4 (2021).

[4] Majid Afshar, Saeed Samet, and Hamid Usefi. 2021. Incorporating Behavior in Attribute Based Access Control Model Using Machine Learning. *15th Annual IEEE International Systems Conference, SysCon 2021 - Proceedings* (apr 2021).

[5] Alfred V. Aho and Margaret J. Corasick. 1975. Efficient string matching. *Commun. ACM* 18, 6 (jun 1975), 333–340.

[6] Rawan Al-Shaer, Jonathan M. Spring, and Eliana Christou. 2020. Learning the Associations of MITRE ATT CK Adversarial Techniques. *2020 IEEE Conference on Communications and Network Security, CNS 2020* (jun 2020).

[7] Wajdi Alhakami, Abdullah Alharbi, Sami Bourouis, Roobaea Alroobaea, and Nizar Bouguila. 2019. Network Anomaly Intrusion Detection Using a Nonparametric Bayesian Approach and Feature Selection. *IEEE Access* 7 (2019), 52181–52190.

[8] Sultan Alneyadi, Elankayer Sithirasenan, and Vallipuram Muthukkumarasamy. 2016. A survey on data leakage prevention systems. *Journal of Network and Computer Applications* 62 (feb 2016), 137–152.

[9] Dennis Appelt, Cu D. Nguyen, and Lionel Briand. 2015. Behind an application firewall, are we safe from SQL injection attacks? *2015 IEEE 8th International Conference on Software Testing, Verification and Validation, ICST 2015 - Proceedings* (may 2015).

[10] Abir Awad, Sara Kadry, Guraraj Maddodi, Saul Gill, and Brian Lee. 2016. Data leakage detection using system call provenance. *Proceedings - 2016 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2016* (oct 2016), 486–491.

[11] Amos Azaria, Ariella Richardson, Sarit Kraus, and V. S. Subrahmanian. 2014. Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data. , 135–155 pages.

[12] Paul Baecher, Markus Koetter, Thorsten Holz, Maximillian Dornseif, and Felix Freiling. 2006. The nepenthes platform: An efficient approach to collect malware. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4219 LNCS. Springer Verlag, 165–184.

[13] Ashutosh Bahuguna, R. K. Bisht, and Jeetendra Pande. 2020. Country-level cybersecurity posture assessment:Study and analysis of practices. *Information Security Journal* 29, 5 (sep 2020), 250–266.

[14] Wade Baker, Mark Goudie, Alexander Hutton, C David Hylender, Jelle Niemantsverdriet, Christopher Novak, David Ostertag, Christopher Porter, Mike Rosen, Bryan Sartin, et al. 2011. 2011 data breach investigations report. *Verizon RISK Team, Available: www. verizonbusiness. com/resources/reports/rp_databreach-investigationsreport-2011_en_xg. pdf* (2011), 1–72.

[15] Tao Ban, Ndichu Samuel, Takeshi Takahashi, and Daisuke Inoue. 2021. Combat Security Alert Fatigue with AI-Assisted Techniques. *ACM International Conference Proceeding Series* (aug 2021), 9–16.

[16] Gagan Bansal, Raymond Fok, Marco Tulio Ribeiro, Tongshuang Wu, Joyce Zhou, Ece Kamar, Daniel S Weld, and Besmira Nushi. 2021. Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance; Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), 1–16.

[17] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. 2019. *Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance.* Technical Report 1. 19 pages. www.aaai.org

[18] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. 2019. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019.* 2429–2437.

[19] Paul Barford, Marc Dacier, Thomas G Dietterich, Matt Fredrikson, Jon Giffin, Sushil Jajodia, Somesh Jha, Jason Li, Peng Liu, Peng Ning, Xinming Ou, Dawn Song, Laura Strater, Vipin Swarup, George Tadda, Cliff Wang, and John Yen. 2010. Cyber SA: Situational awareness for cyber defense. *Advances in Information Security* 46 (2010), 3–13.

[20] Punam Bedi, Vandana Gandotra, Archana Singhal, Himanshi Narang, and Sumit Sharma. 2012. Threat-oriented security framework in risk management using multiagent system. *Wiley Online Library* 43, 9 (sep 2012), 1013–1038.

[21] Maya Bercovitch, Meir Renford, Lior Hasson, Asaf Shabtai, Lior Rokach, and Yuval Elovici. 2011. HoneyGen: An automated honeytokens generator. *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics, ISI 2011* (2011), 131–136.

[22] Deepavali Bhagwat, Laura Chiticariu, Wang-Chiew Tan, Gaurav Vijayvargiya, D Bhagwat, · L Chiticariu, W.-C Tan, and · G Vijayvargiya. 2005. An annotation management system for relational databases. *The VLDB Journal* 14, 4 (oct 2005), 373–396.

[23] Sandeep Bhatt, Pratyusa K. Manadhata, and Loai Zomlot. 2014. The operational role of security information and event management systems. *IEEE Security and Privacy* 12 (2014), 35–41. Issue 5. https://doi.org/10.1109/MSP.2014.103

[24] RM Blank. 2011. Guide for conducting risk assessments. (2011).

[25] James P. Bliss and Richard D. Gilson. 1998. Emergency signal failure: implications and recommendations. *Ergonomics* 41, 1 (jan 1998), 57–72.

[26] DJ Bodeau, CD McCollum, and DB Fox. 2018. Cyber threat modeling: Survey, assessment, and representative framework. (2018).

[27] Lance Bonner. 2012. Cyber risk: How the 2011 Sony data breach and the need for cyber risk insurance policies should direct the federal response to rising data breaches. *Wash. UJL & Pol'y* 40 (2012), 257.

[28] Blake D. Bryant and Hossein Saiedian. 2020. Improving SIEM alert metadata aggregation with a novel kill-chain based classification model. *Computers Security* 94 (7 2020), 101817. https://doi.org/10.1016/J.COSE.2020.101817

[29] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. 2001. Why and Where: A Characterization of Data Provenance. In *International Conference on Database Theory*, Vol. 1973. Springer, Berlin, Heidelberg, 316–330.

[30] Peter Buneman and Wang-Chiew Tan. 2018. Data Provenance: What next? *ACM SIGMOD Record* 47, 3 (2018), 5–13.

[31] S Caltagirone, A Pendergast, and C Betz. 2013. The diamond model of intrusion analysis. *Center For Cyber Intelligence Analysis and Threat Research* (2013).

[32] Jared J. Cash. 2009. Alert fatigue. , 2098–2101 pages.

[33] Davide Castelvecchi. 2020. Quantum-computing pioneer warns of complacency over Internet security - Document - Gale Academic OneFile. *Nature* 587, 7833 (2020), 189–190.

[34] Samir Achraf Chamkar, Yassine Maleh, and Noreddine Gherabi. 2022. THE HUMAN FACTOR CAPABILITIES IN SECURITY OPERATION CENTER (SOC). *EDPACS* 66 (2022), 1–14. Issue 1. https://doi.org/10.1080/07366981.2021.1977026

[35] S Chandel, S Yu, T Yitian, Z Zhili, and H Yusheng. 2019. Endpoint protection: Measuring the effectiveness of remediation technologies and methodologies for insider threat. *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (2019), 81–89.

[36] Juan D. Chaparro, Cory Hussain, Jennifer A. Lee, Jessica Hehmeyer, Manjusri Nguyen, and Jeffrey Hoffman. 2020. *Applied Clinical Informatics* 11, 1 (2020), 46–58.

[37] Suresh N Chari and Pau-Chen Cheng. 2003. BlueBoX: A Policy-Driven, Host-Based Intrusion Detection System. *ACM Transactions on Information and System Security* 6, 2 (2003), 173–200.

[38] Ping Chen, Lieven Desmet, and Christophe Huygens. 2014. A Study on Advanced Persistent Threats. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8735 LNCS (2014), 63–72.

[39] Zouhair Chiba, Noureddine Abghour, Khalid Moussaid, Amina El Omri, and Mohamed Rida. 2018. A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion

detection. *Computers Security* 75 (jun 2018), 36–58.

[40] Mu Huan Chung, Mark Chignell, Lu Wang, Alexandra Jovicic, and Abhay Raman. 2020. Interactive Machine Learning for Data Exfiltration Detection: Active Learning with Human Expertise. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 2020-Octob. 280–287.

[41] Marcello Cinque, Domenico Cotroneo, and Antonio Pecchia. 2018. Challenges and Directions in Security Information and Event Management (SIEM). *Proceedings - 29th IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2018* (11 2018), 95–99. https://doi.org/10.1109/ISSREW.2018.00-24

[42] Clearswift. 2013. The Enemy Within: an emerging threat... https://www.clearswift.com/blog/2013/05/02/enemy-within-emerging-threat

[43] Chris W. Clegg. 2000. Sociotechnical principles for system design. *Applied Ergonomics* 31, 5 (2000), 463–477.

[44] Victor Clincy and Hossain Shahriar. 2018. Web Application Firewall: Network Security Models and Configuration. *Proceedings - International Computer Software and Applications Conference* 1 (jun 2018), 835–836.

[45] B. Commentz-Walter. 1979. A string matching algorithm fast on the average. *Springer- International Colloquium on Automata, Languages, and Programming* (1979), 118–132.

[46] U. S. Congress. 1982. Security Classification Policy and Executive Order 12356. , 13–20 pages.

[47] Jose Antonio Coret. [n.d.]. Kojoney - A honeypot for the SSH Service.

[48] Lorrie Faith Cranor. 2008. A framework for reasoning about the human in the loop. In *Usability, Psychology, and Security, UPSEC 2008*.

[49] CrowdStrike. 2022. 2022 Global Threat Report. (2022).

[50] Joan Daemen and Vincent Rijmen. 1999. AES Proposal: Rijndael. (1999).

[51] R. N. Dahbul, C. Lim, and J. Purnama. 2017. Enhancing Honeypot Deception Capability Through Network Service Fingerprinting. *Journal of Physics: Conference Series* 801, 1 (jan 2017).

[52] K Daniel and J. Andreas. 2022. Evaluation of AI-based use cases for enhancing the cyber security defense of small and medium-sized companies (SMEs). *Electronic Imaging* 34 (2022), 1–8.

[53] Ruth M. Davis. 1978. The Data Encryption Standard in Perspective. *IEEE Communications Society Magazine* 16, 6 (1978), 5–9.

[54] T. Dierks and E. Rescorla. [n.d.]. The Transport Layer Security (TLS) Protocol Version 1.2.

[55] W. Diffie and M. E. Hellman. 1976. New directions in cryptography. .

[56] Deborah D. Downs, Jerzy R. Rub, Kenneth C. Kung, and Carole S. Jordan. 1985. Issues in Discretionary Access Control. *Proceedings - IEEE Symposium on Security and Privacy* (1985), 208–218.

[57] Mahmoud Elkhodr and Belal Alsinglawi. 2020. Data provenance and trust establishment in the Internet of Things. *Security and Privacy* 3, 3 (may 2020).

[58] Mica R. Endsley. 1988. Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors Society Annual Meeting* 32, 2 (oct 1988), 97–101.

[59] Eden Estopace. 2016. Massive data breach exposes all Philippines voters. *[Online]. Available: https://www.telecomasia.net/content/massive-data-breach-exposes-all-philippines-voters* (2016).

[60] Daren Fadolalkarim and Elisa Bertino. 2019. A-PANDDE: Advanced Provenance-based ANomaly Detection of Data Exfiltration. *Computers Security* 84 (jul 2019), 276–287.

[61] Daren Fadolalkarim, Asmaa Sallam, and Elisa Bertino. 2016. PANDDE: Provenance-based anomaly detection of data exfiltration. *CODASPY 2016 - Proceedings of the 6th ACM Conference on Data and Application Security and Privacy* (mar 2016), 267–276.

[62] BS Fakiha. 2020. Effectiveness of Security Incident Event Management (SIEM) System for Cyber Security Situation Awareness. *Indian Journal of Forensic Medicine and Toxicology* 14 (2020). Issue 4.

[63] D Ferraiolo, J Cugini, and DR Kuhn. 1995. Role-based access control (RBAC): Features and motivations. *Proceedings of 11th computer security application conference* (1995), 241–248.

[64] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. 2001. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)* 4, 3 (aug 2001), 224–274.

[65] U Franke and J Brynielsson Security. 2014. Cyber situational awareness–a systematic review of the literature. *Elsevier - Computers Security* (2014).

[66] Maxime Frydman, Guifré Ruiz, Elisa Heymann, Eduardo César, and Barton P. Miller. 2014. Automating risk analysis of software design models. *Scientific World Journal* (2014).

[67] Sean Gallagher. 2015. At first cyber meeting, China claims OPM hack is “criminal case” [Updated] | Ars Technica. https://arstechnica.com/tech-policy/2015/12/at-first-cyber-meeting-china-claims-opm-hack-is-criminal-case/

[68] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers and Security* 28, 1-2 (2009), 18–28.

[69] Jill Gerhardt-Powals. 1996. Cognitive Engineering Principles for Enhancing Human-Computer Performance. *Plastics, Rubber and Composites Processing and Applications* 8, 2 (1996), 189–211.

[70] Iffat A Gheyas and Ali E Abdallah. 2016. Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. *Big Data Analytics* 1, 1 (2016), 1–29.

[71] Shafi Goldwasser and Silvio Micali. 1984. Probabilistic encryption. *J. Comput. System Sci.* 28, 2 (apr 1984), 270–299.

[72] Gustavo González-Granadillo, Susana González-Zarzosa, and Rodrigo Diaz. 2021. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. *Sensors* 21 (7 2021), 4759. Issue 14. https://doi.org/10.3390/S21144759

[73] Stephanie Gootman. 2016. OPM hack: The most dangerous threat to the federal government today. *Journal of Applied Security Research* 11, 4 (2016), 517–525.

[74] Frank L Greitzer and Deborah A Frincke. 2010. Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation. In *Insider threats in cyber security*. Springer, 85–113.

[75] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. 2018. Lemna: Explaining deep learning based security applications. *Proceedings of the ACM Conference on Computer and Communications Security* (oct 2018), 364–379.

[76] Hani Hagras. 2018. Toward Human-Understandable, Explainable AI. *Computer* 51, 9 (sep 2018), 28–36.

[77] P A Hancock, Tara Kajaks, Jeff K Caird, Mark H Chignell, Sachi Mizobuchi, Peter C. Burns, Jing Feng, Geoff R Fernie, Martin Lavallière, Ian Y. Noy, Donald A Redelmeier, and Brenda H. Vrkljan. 2020. Challenges to Human Drivers in Increasingly Automated Vehicles. *Human Factors* 62, 2 (mar 2020), 310–328.

[78] Richard Harang and Peter Guarino. 2012. Clustering of Snort alerts to identify patterns and reduce analyst workload. In *Proceedings - IEEE Military Communications Conference MILCOM*.

[79] Michael Hart, Pratyusa Manadhata, and Rob Johnson. 2011. Text Classification for Data Loss Prevention. *Privacy Enhancing Technologies* (2011), 18–37.

[80] W. U. Hassan, MA Noureddine, P. Datta, and A. Bates. 2020. OmegaLog: High-fidelity attack investigation via transparent multi-layer log analysis. In *Network and Distributed System Security Symposium*.

[81] Morgan Henrie. 2013. Cyber security risk management in the scada critical infrastructure environment. *EMJ - Engineering Management Journal* 25, 2 (jun 2013), 38–45.

[82] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for Explainable AI: Challenges and Prospects. arXiv:1812.04608

[83] Andreas Holzinger, Markus Plass, Michael Kickmeier-Rust, Katharina Holzinger, Gloria Cerasela Crişan, Camelia M. Pintea, and Vasile Palade. 2019. Interactive machine learning: experimental evidence for the human in the algorithmic loop: A case study on Ant Colony Optimization. *Applied Intelligence* 49, 7 (jul 2019), 2401–2414.

[84] Ivan Homoliak, Flavio Toffalini, Juan Guarnizo, Yuval Elovici, and Martín Ochoa. 2019. Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Computing Surveys (CSUR)* 52, 2 (2019), 1–40.

[85] Anne Honkaranta, Tiina Leppanen, and Andrei Costin. 2021. Towards Practical Cybersecurity Mapping of STRIDE and CWE - A Multi-perspective Approach. *Conference of Open Innovation Association, FRUCT* (may 2021), 150–159.

[86] Feng-Yung Hu. 2016. Russian Intervention: Paranoia or Weapon for National Security? From the Perspective on Public Diplomacy. *Washington Post* (2016).

[87] Rui Hu, Zheng Yan, Wenxiu Ding, and Laurence T. Yang. 2020. A survey on data provenance in IoT. *World Wide Web* 23, 2 (mar 2020), 1441–1463.

[88] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. 2013. Guide to attribute based access control (ABAC) definition and considerations (draft). *NIST special publication* 800, 162 (2013).

[89] Sebastiaan P. Huber, Spyros Zoupanos, Martin Uhrin, Leopold Talirz, Leonid Kahle, Rico Häuselmann, Dominik Gresch, Tiziano Müller, Aliaksandr V. Yakutovich, Casper W. Andersen, Francisco F. Ramirez, Carl S. Adorf, Fernando Gargiulo, Snehal Kumbhar, Elsa Passaro, Conrad Johnston, Andrius Merkys, Andrea Cepellotti, Nicolas Mounet, Nicola Marzari, Boris Kozinsky, and Giovanni Pizzi. 2020. AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Scientific Data* 7, 1 (sep 2020), 1–18. arXiv:2003.12476

[90] Jeffrey Hunker and Christian W Probst. 2011. Insiders and Insider Threats-An Overview of Definitions and Mitigation Techniques. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* 2, 1 (2011), 4–27.

[91] Eric M Hutchins, Michael J Cloppert, Rohan M Amin, et al. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research* 1, 1 (2011), 80.

[92] Sotiris Ioannidis, Angelos D Keromytis, Steve M Bellovin, and Jonathan M Smith. 2000. Implementing a Distributed Firewall. *Proceedings of the 7th ACM conference on Computer and communications security* (2000), 190–199.

[93] Graeme Jenkinson, Lucian Carata, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Robert N M Watson, Jonathan Anderson, Brian Kidney, Amanda Strnad, and Arun Thomas. 2017. Applying Provenance in APT Monitoring and Analysis: Practical Challenges for Scalable, Efficient and Trustworthy Distributed Provenance. *9th USENIX Workshop on the Theory and Practice of Provenance* (2017).

[94] Xin Jin, Ram Krishnan, and Ravi Sandhu. 2012. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2012), 41–55.

[95] Shijoe Jose, D. Malathi, Bharath Reddy, and Dorathi Jayaseeli. 2018. A Survey on Anomaly Based Host Intrusion Detection System. In *Journal of Physics: Conference Series*, Vol. 1000. Institute of Physics Publishing, 12049.

[96] N Kaloudi, J Li ACM Computing Surveys (CSUR), and undefined 2020. 2020. The ai-based cyber threat landscape: A survey. *dl.acm.org* 53, 1 (feb 2020).

[97] Adi Karahasanovic, Pierre Kleberger, and Magnus Almgren. 2017. Adapting Threat Modeling Methods for the Automotive Industry. *ej tryckt* (2017).

[98] Mike Karp. 2005. Keep on truckin' your back-up tapes? You've got to be kidding! | Network World. https://www.networkworld.com/article/2320740/keep-on-truckin--your-back-up-tapes--you-ve-got-to-be-kidding-.html

[99] Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen. 2010. Querying data provenance. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2010), 951–962.

[100] Kelly M Kavanagh, Oliver Rochford, and Toby Bussa. 2015. Magic quadrant for security information and event management. *Gartner Group Research Note* (2015).

[101] Salman Khaliq, Zain Ul Abideen Tariq, and Ammar Masood. 2020. Role of User and Entity Behavior Analytics in Detecting Insider Attacks. *1st Annual International Conference on Cyber Warfare and Security, ICCWS 2020 - Proceedings* (oct 2020).

[102] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. 2017. STRIDE-based threat modeling for cyber-physical systems. *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2017 - Proceedings* (jul 2017), 1–6.

[103] Dennis Kiwia, Ali Dehghantanha, Kim Kwang Raymond Choo, and Jim Slaughter. 2018. A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. *Journal of Computational Science* 27 (jul 2018), 394–409.

[104] L. Kohnfelder and Garg. 1999. The threats to our products.

[105] Maria Korolov and Lysa Myers. 2018. What is the Cyber Kill Chain? Why It's Not Always the Right Approach to Cyber Attacks. CSO.

[106] Igor Kotenko and Evgenia Novikova. 2014. Visualization of security metrics for cyber situation awareness. *Proceedings - 9th International Conference on Availability, Reliability and Security, ARES 2014* (12 2014), 506–513. https://doi.org/10.1109/ARES.2014.75

[107] Srinivas Krishnan, Kevin Z. Snow, and Fabian Monrose. 2012. Trail of bytes: New techniques for supporting data provenance and limiting privacy breaches. *IEEE Transactions on Information Forensics and Security* 7, 6 (2012), 1876–1889.

[108] Sailesh Kumar. 2007. Survey of Current Network Intrusion Detection Techniques. *Washington Univ. in St. Louis* (2007).

[109] Roger Kwon, Travis Ashley, Jerry Castleberry, Penny McKenzie, and Sri Nikhil Gupta Gourisetti. 2020. Cyber threat dictionary using MITRE ATTCK matrix and NIST cybersecurity framework mapping. *2020 Resilience Week, RWS 2020* (oct 2020), 106–112.

[110] Butler W. Lampson. 1974. Protection. *ACM SIGOPS Operating Systems Review* 8, 1 (jan 1974), 18–24.

[111] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. 2003. A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM)* (may 2003), 25–36.

[112] Duc C. Le, Nur Zincir-Heywood, and Malcolm I. Heywood. 2020. Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning. *IEEE Transactions on Network and Service Management* 17 (3 2020), 30–44. Issue 1. https://doi.org/10.1109/TNSM.2020.2967721

[113] Hyunjung Lee, Suryeon Lee, Kyounggon Kim, and Huy Kang Kim. 2021. HSViz: Hierarchy Simplified Visualizations for Firewall Policy Analysis. *IEEE Access* 9 (2021), 71737–71753.

[114] John D. Lee and Neville Moray. 1994. Trust, self-Confidence, and operators' adaptation to automation. *International Journal of Human - Computer Studies* 40, 1 (1994), 153–184.

[115] John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. , 50–80 pages.

[116] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. 2017. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. *Proceedings - 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017* (jul

2017), 468–477.
[117] Liu Liu, Olivier De Vel, Qing-Long Han, Jun Zhang, and Yang Xiang. 2018. Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 1397–1417.
[118] Simon Liu and Rick Kuhn. 2010. Data loss prevention. *IT Professional* 12, 2 (mar 2010), 10–13.
[119] Lockheed Martin. 2022. Cyber Kill Chain . https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
[120] Xin Luo, Richard Brody, Alessandro Seazzu, and Stephen Burd. 2011. Social Engineering: The Neglected Human Factor for Information Security Management. *Information Resources Management Journal (IRMJ)* 24 (2011), 1–8. Issue 3. https://doi.org/10.4018/IRMJ.2011070101
[121] Tyson Macaulay. 2016. RIoT Control: Understanding and Managing Risks and the Internet of Things.
[122] Florian Mansmann, Timo Göbel, and William Cheswick. 2012. Visual analysis of complex firewall configurations. *ACM International Conference Proceeding Series* (2012), 1–8.
[123] Aaron Marback, Hyunsook Do, Ke He, Samuel Kondamarri, and Dianxiang Xu. 2013. A threat model-based approach to security testing. *Software: Practice and Experience* 43, 2 (feb 2013), 241–258.
[124] Goncalo Martins, Sajal Bhatia, Xenofon Koutsoukos, Keith Stouffer, Cheeyee Tang, and Richard Candell. 2015. Towards a systematic threat modeling approach for cyber-physical systems. *Proceedings - 2015 Resilience Week, RSW 2015* (oct 2015), 114–119.
[125] Earl D. Matthews, Harold J. Arata III, and Brian L. Hale. 2016. Cyber situational awareness. *JSTOR: The Cyber Defense Review* 1, 1 (2016), 35–46.
[126] Vasileios Mavroeidis and Audun Jøsang. 2018. Data-Driven Threat Hunting Using Sysmon. *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy* (2018).
[127] McAfee. 2021. Advanced Threat Research Report. (2021).
[128] CSIS McAfee. 2014. Net losses: estimating the global cost of cybercrime. *McAfee, Centre for Strategic & International Studies* (2014).
[129] Michael Mesnier, Eno Thereska, Gregory R. Ganger, Daniel Ellard, and Margo Seltzer. 2004. File classification in self-* storage systems. *Proceedings - International Conference on Autonomic Computing* (2004), 44–51.
[130] Md Nazmus Sakib Miazi, Mir Mehedi A. Pritom, Mohamed Shehab, Bill Chu, and Jinpeng Wei. 2017. The design of cyber threat hunting games: A case study. *2017 26th International Conference on Computer Communications and Networks, ICCCN 2017* (sep 2017).
[131] MITRE ATTCK. [n.d.]. ATTCK Matrix for Enterprise. https://attack.mitre.org/
[132] Iyatiti Mokube and Michele Adams. 2007. Honeypots: Concepts, approaches, and challenges. In *Proceedings of the Annual Southeast Conference*, Vol. 2007. 321–326.
[133] B Mukherjee, LT Heberlein, and KN Levitt. 1994. Network intrusion detection. *IEEE Network* (1994), 26–41.
[134] Masoud Narouei, Hamed Khanpour, Hassan Takabi, Natalie Parde, and Rodney Nielsen. 2017. Towards a top-down policy engineering framework for attribute-based access control. *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT* (jun 2017), 103–114.
[135] Rida Nasir, Mehreen Afzal, Rabia Latif, and Waseem Iqbal. 2021. Behavioral Based Insider Threat Detection Using Deep Learning. *IEEE Access* 9 (2021), 143266–143274. https://doi.org/10.1109/ACCESS.2021.3118297
[136] Peter G Neumann. 2010. Combatting insider threats. In *Insider Threats in Cyber Security*. Springer, 17–44.
[137] Jakob Nielsen. 2004. Usability engineering. In *Computer Science Handbook, Second Edition*. 45–1–45–21.
[138] Kaiti Norton. 2020. Antivirus vs. EPP vs. EDR: How to Secure Your Endpoints.
[139] Evgenia Novikova and Igor Kotenko. 2013. Analytical visualization techniques for security information and event management. *Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2013* (2013), 519–525. https://doi.org/10.1109/PDP.2013.84
[140] Jason RC Nurse, Oliver Buckley, Philip A Legg, Michael Goldsmith, Sadie Creese, Gordon RT Wright, and Monica Whitty. 2014. Understanding insider threat: A framework for characterising attacks. In *2014 IEEE Security and Privacy Workshops*. IEEE, 214–228.
[141] Sylvia Osborn. 1997. Mandatory access control and role-based access control revisited. In *Proceedings of the ACM Workshop on Role-Based Access Control*. 31–40.
[142] Y Ou, Y Lin, and Y Zhang. 2010. The design and implementation of host-based intrusion detection system. *The design and implementation of host-based intrusion detection system* (2010), 595–598.
[143] Vassilis Papaspirou, Leandros Maglaras, Mohamed Amine Ferrag, Ioanna Kantzavelou, Helge Janicke, and Christos Douligeris. 2021. A novel Two-Factor HoneyToken Authentication Mechanism. *Proceedings - International Conference on Computer Communications and Networks, ICCCN* (jul 2021). arXiv:2012.08782
[144] Jaehong Park and Ravi Sandhu. 2004. The UCONABC usage control model. *ACM Transactions on Information and System Security (TISSEC)* 7, 1 (feb 2004), 128–174.
[145] Kamran Parsaye and Mark Chignell. 1988. Expert Systems for experts. (1988).

[146] Charles Perrow. 1981. *Normal accident at three Mile Island.* Technical Report 5. 17–26 pages.

[147] John Pescatore. 2021. SANS 2021 Top New Attacks and Threat Report. (2021). https://www.rapid7.com/info/sans-2021-new-attacks-threat-report/

[148] A. B.Robert Petrunić. 2015. Honeytokens as active defense. *38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings* (jul 2015), 1313–1317.

[149] Shari Lawrence Pfleeger, Joel B Predd, Jeffrey Hunker, and Carla Bulford. 2009. Insiders behaving badly: Addressing bad actors and their actions. *IEEE transactions on information forensics and security* 5, 1 (2009), 169–179.

[150] Charles E Phillips, T C Ting, and Steven A Demurjian. 2002. Information Sharing and Security in Dynamic Coalitions. *Proceedings of the seventh ACM symposium on Access control models and technologies - SACMAT '02* (2002).

[151] Oskars Podzins and Andrejs Romanovs. 2019. Why SIEM is Irreplaceable in a Secure IT Environment? *2019 Open Conference of Electrical, Electronic and Information Sciences, eStream 2019 - Proceedings* (4 2019). https://doi.org/10.1109/ESTREAM.2019.8732173

[152] Davy Preuveneers and Wouter Joosen. 2021. Sharing Machine Learning Models as Indicators of Compromise for Cyber Threat Intelligence. *Journal of Cybersecurity and Privacy 2021, Vol. 1, Pages 140-163* 1, 1 (feb 2021), 140–163.

[153] D Dhillon Privacy. 2011. Developer-driven threat modeling: Lessons learned in the trenches. *IEEE Security Privacy* (2011).

[154] Niels Provos. 2004. A virtual honeypot framework. *Proceedings of the 13th USENIX Security Symposium* (2004).

[155] Ben Quinn and Charles Arthur. 2011. PlayStation Network hackers access data of 77 million users. *The Guardian* 27 (2011).

[156] Fahimeh Raja, Kirstie Hawkey, and Konstantin Beznosov. 2009. Towards improving mental models of personal firewall users. *Conference on Human Factors in Computing Systems - Proceedings* (2009), 4633–4638.

[157] Fahimeh Raja, Kai Le Clement Wang, Kirstie Hawkey, Konstantin Beznosov, and Steven Hsu. 2011. Promoting a physical security mental model for personal firewall warnings. *Conference on Human Factors in Computing Systems - Proceedings* (2011), 1585–1590.

[158] Pedro Ramos Brandao and João Nunes. 2021. Extended Detection and Response Importance of Events Context. *Kriative.tech* (2021).

[159] R. Rengarajan and S. Babu. 2021. Anomaly Detection using User Entity Behavior Analytics and Data Visualization. *8th International Conference on Computing for Sustainable Global Development* (2021), 842–847.

[160] Ian Reynolds. 2020. 2020 SANS Network Visibility and Threat Detection Survey. *SANS Institute* April (2020). https://www.sans.org/webcasts/network-visibility-threat-detection-survey-112595

[161] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-Augu (aug 2016), 1135–1144. arXiv:1602.04938

[162] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *ACM Secure communications and asymmetric cryptosystems* 21, 2 (feb 1978), 120–126.

[163] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. 2019. *Zero Trust Architecture.* Technical Report.

[164] Bushra Sabir, Faheem Ullah, M. Ali Babar, and Raj Gaire. 2021. Machine Learning for Detecting Data Exfiltration: A Review. *Comput. Surveys* 54, 3 (jun 2021).

[165] Fatima Salahdine and Naima Kaabouch. 2019. Social Engineering Attacks: A Survey. *Future Internet 2019, Vol. 11, Page 89* 11 (4 2019), 89. Issue 4. https://doi.org/10.3390/FI11040089

[166] Malek Ben Salem, Shlomo Hershkop, and Salvatore J Stolfo. 2008. A survey of insider attack detection research. *Insider Attack and Cyber Security* (2008), 69–90.

[167] Ravi S. Sandhu. 1993. Lattice-Based Access Control Models. *Computer* 26, 11 (1993), 9–19.

[168] Ravi S. Sandhu. 1998. Role-based Access Control. *Advances in Computers* 46, C (jan 1998), 237–286.

[169] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. 1996. Computer role-based access control models. *Computer* 29, 2 (feb 1996), 38–47.

[170] Ravi S. Sandhu and Pierangela Samarati. 1994. Access Control: Principles and Practice. *IEEE Communications Magazine* 32, 9 (1994), 40–48.

[171] Riccardo Scandariato, Kim Wuyts, and Wouter Joosen. 2015. A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering* 20, 2 (mar 2015), 163–180.

[172] Peter Schaab, Kristian Beckers, and Sebastian Pape. 2017. Social engineering defence mechanisms and counteracting training strategies. *Information and Computer Security* 25 (2017), 206–222. Issue 2. https://doi.org/10.1108/ICS-04-2017-0022/FULL/HTML

[173] G. Scott Graham and Peter J. Denning. 1972. Protection-Principles and practice. *Proceedings of the Spring Joint Computer Conference, AFIPS 1972* (may 1972), 417–429.

[174] Daniel Servos and Sylvia L Osborn. 2017. Current research and open problems in attribute-based access control. *ACM Computing Surveys (CSUR)* 49, 4 (2017), 1–45.

[175] Burr Settles. 2009. Active learning literature survey. *Technical Report* (2009).

[176] Burr Settles. 2011. From theories to queries: Active learning in practice. *JMLR: Workshop and Conference Proceedings 16* 16 (2011), 1–18.

[177] William Seymour. 2019. Privacy therapy with ARETHA: What if your firewall could talk? *Conference on Human Factors in Computing Systems - Proceedings* (may 2019).

[178] A Shabtai, Y Elovici, and L Rokach. 2012. A survey of data leakage detection and prevention solutions. (2012).

[179] Dave Shackleford. 2016. SANS 2016 security analytics survey. *SANS Institute, Swansea* (2016).

[180] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (nov 1979), 612–613.

[181] Balaram Sharma, Prabhat Pokharel, and Basanta Joshi. 2020. User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder: Insider Threat Detection. *Proceedings of the 11th International Conference on Advances in Information Technology* (2020), 1–9.

[182] Rupam Kumar Sharma, Hemanta Kumar Kalita, and Biju Issac. 2014. Different firewall techniques: A survey. *5th International Conference on Computing Communication and Networking Technologies, ICCCNT 2014* (nov 2014).

[183] Thomas B Sheridan and Robert T Hennessy. 1984. *Research and Modeling of Supervisory Control Behavior.* Technical Report.

[184] N Shevchenko, TA Chick, P O'Riordan, and TP Scanlon. 2018. Threat modeling: a summary of available methods. *Carnegie Mellon University Software Engineering Institute* (2018).

[185] Adam Shostack. 2008. Experiences Threat Modeling at Microsoft. (2008).

[186] Adam Shostack. 2014. *Threat Modeling: Designing for Security.*

[187] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. 2005. A survey of data provenance in e-science. *ACM SIGMOD Record* 34, 3 (sep 2005), 31–36.

[188] Jussi Simola and Jyri Rajamäki. 2017. Hybrid emergency response model: Improving cyber situational awareness. In *European Conference on Information Warfare and Security, ECCWS*. 442–451. www.laurea.fi

[189] Michael Sivak, Daniel J. Weintraub, and Michael Flannagan. 1991. Nonstop Flying Is Safer Than Driving. *Risk Analysis* 11, 1 (1991), 145–148.

[190] Miles E. Smid and Dennis K. Branstad. 1988. The Data Encryption Standard: Past and Future. *Proc. IEEE* 76, 5 (1988), 550–559.

[191] Philip J. Smith, C. Elaine McCoy, and Charles Layton. 1997. Brittleness in the design of cooperative problem-solving systems: The effects on user performance. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.* 27, 3 (1997), 360–371.

[192] Luke S. Snyder, Yi Shan Lin, Morteza Karimzadeh, Dan Goldwasser, and David S. Ebert. 2019. Interactive learning for identifying relevant tweets to support real-time situational awareness.

[193] Lance Spitzner. 2003. Honeypots: Catching the insider threat. *Proceedings - Annual Computer Security Applications Conference, ACSAC* 2003-Janua (2003), 170–179.

[194] L. Spitzner. 2003. Honeytokens: The other honeypot.

[195] Lance Spitzner. 2003. The honeynet project: Trapping the hackers. *IEEE Security and Privacy* 1, 2 (2003), 15–23.

[196] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. 2020. Towards systematic honeytoken fingerprinting. *13th International Conference on Security of Information and Networks* (2020).

[197] J Steven. 2010. Threat modeling-perhaps it's time. *IEEE Security Privacy* (2010).

[198] SJ Stolfo, SM Bellovin, S Hershkop, and AD Keromytis. 2008. Insider attack and cyber security: beyond the hacker. (2008).

[199] Jeremy Straub. 2020. Modeling Attack, Defense and Threat Trees and the Cyber Kill Chain, ATTCK and STRIDE Frameworks as Blackboard Architecture Networks. *Proceedings - 2020 IEEE International Conference on Smart Cloud, SmartCloud* (nov 2020), 148–153.

[200] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas. 2018. Mitre attck: Design and philosophy. *Technical report* (2018).

[201] Frank. Swiderski and Window. Snyder. 2004. *Threat modeling.* Microsoft Press.

[202] Dan Swinhoe. 2019. The biggest data breach fines, penalties and settlements so far. *CSO, Framingham, jul* (2019).

[203] Dan Swinhoe. 2020. The 15 biggest data breaches of the 21st century. *CSO. Last modified* (2020).

[204] Mohammad M.Bany Taha, Sivadon Chaisiri, and Ryan K.L. Ko. 2015. Trusted tamper-evident data provenance. *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015* 1 (dec 2015), 646–653.

[205] Radwan Tahboub and Yousef Saleh. 2014. Data leakage/loss prevention systems (DLP). *2014 World Congress on Computer Applications and Information Systems, WCCAIS 2014* (oct 2014).

[206] Baoming Tang, Qiaona Hu, and Derek Lin. 2017. Reducing false positives of user-to-entity first-access alerts for user behavior analytics. *IEEE International Conference on Data Mining Workshops, ICDMW* (dec 2017), 804–811.

[207] Adem Tekerek, Cemal Gemci, and Omer Faruk Bay. 2014. Development of a hybrid web application firewall to prevent web based attacks. *8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014 - Conference Proceedings* (2014).

[208] Erdem Ucar and Erkan Ozhan. 2017. The Analysis of Firewall Policy Through Machine Learning and Data Mining. *Wireless Personal Communications* 96, 2 (sep 2017), 2891–2909.

[209] Faheem Ullah, Matthew Edwards, Rajiv Ramdhany, Ruzanna Chitchyan, M Ali Babar, and Awais Rashid. 2018. Data exfiltration: A review of external attack vectors and countermeasures. *Journal of Network and Computer Applications* 101 (2018), 18–54.

[210] AV Uzunov and EB Fernandez Interfaces. 2014. An extensible pattern-based library and taxonomy of security threats for distributed systems. *Elsevier - Computer Standards* (2014).

[211] Antonio Varriale, Paolo Prinetto, Alberto Carelli, and Pascal Trotta. 2016. SEcube ™ : Data at Rest and Data in Motion Protection. *International Conference Security and Management* (2016), 138–145.

[212] Verizon. 2020. 2020 Data Breach Investigations Report. https://enterprise.verizon.com/resources/reports/dbir/

[213] Rakesh Verma, Murat Kantarcioglu, David Marchette, Ernst Leiss, and Thamar Solorio. 2015. Security analytics: Essential data analytics knowledge for cybersecurity professionals and students. *IEEE Security and Privacy* 13, 6 (2015), 60–65.

[214] Luca Vigano and Daniele Magazzeni. 2020. Explainable Security. In *Proceedings - 5th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2020*. 293–300. arXiv:1807.04178

[215] Ke Wang and Salvatore J. Stolfo. 2004. Anomalous Payload-Based Network Intrusion Detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3224 (2004), 203–222.

[216] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl A Gunter, and Haifeng Chen. 2020. You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. *Network and Distributed Systems Security (NDSS) Symposium 2020* (2020).

[217] David Watson and Jamie Riden. 2008. *The honeynet project: Data collection tools, infrastructure, archives and analysis.* Technical Report. 24–30 pages.

[218] Imano Williams and Xiaohong Yuan. 2015. Evaluating the Effectiveness of Microsoft Threat Modeling Tool. *Proceedings of the 2015 Information Security Curriculum Development Conference* (2015).

[219] Martyn Williams. 2017. Inside the Russian hack of Yahoo: How they did it.

[220] Avishai Wool. 2004. A Quantitative Study of Firewall Configuration Errors. *Computer* 37, 6 (2004), 62–67.

[221] Sun Wu and Udi Manber. 1994. A FAST ALGORITHM FOR MULTI-PATTERN SEARCHING. (1994).

[222] Tobias Wüchner and Alexander Pretschner. 2012. Data loss prevention based on data-driven usage control. *Proceedings - International Symposium on Software Reliability Engineering, ISSRE* (2012), 151–160.

[223] Wenjun Xiong, Emeline Legrand, Oscar Åberg, and Robert Lagerström. 2022. Cyber security threat modeling based on the MITRE Enterprise ATTCK Matrix. *Software and Systems Modeling* 21, 1 (feb 2022), 157–177.

[224] W Xiong and R Lagerström Security. 2019. Threat modeling–A systematic literature review. *Elsevier Computers security* (2019).

[225] Kaiping Xue, Weikeng Chen, Wei Li, Jianan Hong, and Peilin Hong. 2018. Combining Data Owner-Side and Cloud-Side Access Control for Encrypted Cloud Storage. *IEEE Transactions on Information Forensics and Security* 13, 8 (aug 2018), 2062–2074.

[226] T Yadav and AM Rao. 2015. Technical aspects of cyber kill chain. *International Symposium on Security in Computing and Communication* (2015), 438–452.

[227] Ran Yahalom, Erez Shmueli, and Tomer Zrihen. 2010. Constrained Anonymization of Production Data: A Constraint Satisfaction Problem Approach. *Secure Data Management* (2010), 41–53.

[228] Jae yeol Kim and Hyuk Yoon Kwon. 2022. Threat classification model for security information event management focusing on model efficiency. *Computers Security* 120 (9 2022), 102789. https://doi.org/10.1016/J.COSE.2022.102789

[229] Faheem Zafar, Abid Khan, Saba Suhail, Idrees Ahmed, Khizar Hameed, Hayat Mohammad Khan, Farhana Jabeen, and Adeel Anjum. 2017. Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. *Journal of Network and Computer Applications* 94 (sep 2017), 50–68.

[230] Marzia Zaman and Chung Horng Lung. 2018. Evaluation of machine learning techniques for network intrusion detection. *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018* (jul 2018), 1–5.

[231] Xiaopeng Zhang. 2022. Phishing Campaign Delivering Three Fileless Malware: AveMariaRAT / BitRAT / PandoraHVNC – Part I | FortiGuard Labs.

[232] Xinyou Zhang, Chengzhong Li, and Wenbin Zheng. 2004. Intrusion prevention system design. *Proceedings - The Fourth International Conference on Computer and Information Technology (CIT 2004)* (2004), 386–390.