Cyber Deception: Using Honey Accounts to Deceive Attackers in a Windows Active Directory Environment: Cost-Effective Password Spray Defense

Benjamin Rader
IUPUI
Purdue School of Engineering & Technology

Abstract—This paper delves into the realm of cybersecurity, particularly focusing on the innovative use of cyber deception in the form of honey (decoy) user accounts within a Windows Active Directory (AD) environment for the detection of password spray attacks. Grounded in the principle of asymmetry that characterizes cybersecurity, the research explores the defender's dilemma and proposes cyber deception as a proactive defense strategy by utilizing the asymmetric knowledge advantage that defenders have in their environments. Employing a sophisticated purple teaming approach, the study contrasts the effectiveness of decoy user accounts against traditional univariate time series anomaly detection methods like STL decomposition in Azure Cloud. The research demonstrates that while time series analysis has its merits, the simplicity, directness, and versatility of decoy user accounts make them a superior method in detecting password sprays, especially in the face of Advanced Persistent Threats (APTs). This paper contributes to the cybersecurity discourse by highlighting the strategic value of cyber deception in modern cyber defense strategies and its role in shifting the balance of power in cybersecurity operations.

1 Introduction

1.1 The Defender's Dilemma

Cybersecurity landscapes are fraught with asymmetry—defenders must fortify every conceivable entry point, while attackers need but a single overlooked vulnerability to compromise a system. This paradigm, known as the defender's dilemma, encapsulates the systemic complexity and inherent power imbalances that skew in favor of the aggressor. In the intricate weave of a Windows Active Directory (AD) environment, the defender's mission is often futile; the system's intricacies provide a multitude of shadows for adversaries to hide and an array of potential entry points to exploit. Given infinite time and persistence, an attacker's chances of breaching defenses and infiltrating the system rise inexorably [1]. Consequently, cybersecurity strategies often caution against putting all of one's eggs in a single basket and instead for a multi-layered defense strategy that encompasses perimeter defenses, Endpoint Detection and Response (EDR) systems, and sophisticated analytics to unearth the oft-veiled malicious activities within the network. As the digital realm evolves, so too must the strategies employed by those entrusted with its defense. The traditional reactive posture—waiting for an attack to occur before responding is proving inadequate. In this dynamic battleground, innovative thinkers in security operations are pioneering unconventional methods to address the defender's dilemma. Cyber deception emerges as one such avant-garde approach, offering a proactive defense mechanism that not only detects but actively engages would-be attackers.

1.2 Cyber Deception

The essence of cyber deception lies in exploiting the knowledge asymmetry between attackers and defenders, creating a mirage of vulnerabilities and opportunities that lead adversaries astray. Reminiscent of the cunning ploys of Kevin McAlister in "Home Alone," where the young protagonist conjures the illusion of a bustling household to deter burglars, cyber deception weaves a digital tapestry of false realities to ensnare attackers. Honey accounts in a Windows AD environment are emblematic of such tactics—artificially constructed user profiles, replete with credentials and privileges, designed to mimic legitimate users. Yet, their sole purpose is to serve as digital tripwires, alerting defenders to authentication failures or unauthorized interactions with AD objects, thereby betraying the presence of an adversary within the system.

1.3 Cyber Deception Examples in Literature

The academic discourse on cyber deception is rich and evolving, offering concrete examples and novel insights that underscore its significance in modern cyber defense strategies. The following are examples of various use cases and goals of cyber deception technology.

1.3.1 Psychological Warfare

n the realm of cyber defense, the psychological impact of deception plays a crucial role in the efficacy of decoy systems. One research project called the "Tularosa Study"

demonstrates this by revealing how the explicit mention of deception, combined with the presence of decoys, profoundly influences attacker behavior. This approach leads to an altered perception in attackers, inducing cognitive biases and decision-making disruptions. Particularly noteworthy is the influence on attackers' cognitive state, where the awareness of potential deception not only causes confusion but also affects their perception of success and failure, potentially leading to a self-serving bias. These psychological dimensions underscore the strategic importance of managing information in cyber deception, as over-sharing could diminish the effectiveness of decoys. This psychological warfare in cyberspace, which involves exploiting attackers' cognitive biases, represents a sophisticated layer in cyber defense strategies, suggesting the necessity of integrating cyber deception into a broader, behavior-focused security framework [2].

1.3.2 Hard-To-Understand Systems

In the realm of cyber security, the synergy between Moving Target Defense (MTD) and cyber deception also represents a sophisticated stratagem in mitigating the inherent asymmetry that favors attackers. MTD, as a proactive defense mechanism, dynamically and randomly alters system attributes, creating a moving target that is difficult for attackers to pin down. This fluidity in system attributes inherently increases the complexity and uncertainty for the attacker. Cyber deception complements this by adding layers of plausible, carefully crafted misinformation. This dual approach serves to mislead attackers, drawing them into a labyrinth of false realities. The deception elements like honeypots, honey baits, honey tokens, breadcrumbs, and well-constructed deception scenes are not mere traps; they are intricate components of a larger illusion, increasing the entropy and disorientation for the attacker. Quantifying the effectiveness of deception within the network remains an elusive task, as traditional evaluation methods fall short of capturing the dynamic and interactive complexity inherent in MTD systems. Moreover, many existing MTD evaluation methods are specific to certain technologies and scenarios, lack broad applicability. Despite these challenges, the strategic integration of MTD and cyber deception marks a significant leap in cyber defense, offering a more resilient and confusing landscape for attackers [3].

1.3.3 Early Warning Systems

Cyber deception emerges not only as a means to obscure the true identity of cyber assets but also as an efficient early warning system in the face of cyber-attacks. By introducing controlled misinformation and decoys into the network, cyber deception creates a layer of confusion that disrupts the attackers' ability to accurately identify and target genuine assets. This obfuscation plays a crucial role in blurring the attacker's perception, thereby delaying and potentially derailing their progress within the cyber kill chain. Importantly, this approach allows for the early detection of adversarial activities, a key advantage considering that most cyber incidents often remain undetected for extended periods, sometimes up to a year. The implementation of cyber deception strategies, therefore, not only acts as a shield for critical assets but also serves as an efficient, cost-effective method for early intrusion detection and response, fundamentally altering the traditional dynamics of cyber warfare. In this light, cyber deception can be viewed as a minimal investment with significant returns in the realm of cybersecurity. It effectively shifts the balance of power in cyber warfare, turning what is traditionally a reactive defense into a proactive engagement. The early warning system aspect of cyber deception, derived from its ability to cause confusion and misdirection, provides invaluable time for defenders to identify and respond to threats. This proactive stance in identifying threats early in the kill chain not only conserves resources but also provides a strategic edge, allowing for a more robust and prepared response to cyber threats. Thus, cyber deception stands as a pivotal element in contemporary cyber defense strategies, offering a cost-effective and efficient solution to the ever-evolving challenges of cyber warfare [4].

1.3.4 Threat Intelligence & Attack Research

Active Cyber Deception (ACD) is another approach to cyber deception that transcends traditional defense mechanisms by not only misleading adversaries with falsified data but also facilitating deep engagement to dissect and comprehend novel attack techniques. This strategic interaction with malware and Advanced Persistent Threats (APTs) allows for a detailed analysis of their behaviors at both technical and tactical levels. SODA's approach, involving the analysis of real-world malware and the creation of tailored Deception Playbooks, utilizes this insight to construct deceptive environments that are specifically engineered to mislead and manipulate malware. By extracting Malicious Sub-graphs (MSGs) from malware and mapping these to MITRE ATT&CK techniques, SODA effectively deciphers how attack techniques are manifested in malware, guiding the development of sophisticated deception actions. This methodology not only demonstrates high efficacy in deception, evidenced by impressive accuracy and recall rates, but also significantly enhances threat intelligence and security research by providing a deeper understanding of the threat landscape and adversary tactics, thereby refining the overall threat model [5].

In conclusion, the literature on cyber deception paints a vivid picture of its efficacy and evolution. From decoy systems to the integration with MTD, cyber deception has emerged as a formidable tool in the cybersecurity arsenal. Its ability to manipulate attacker perception, decision-making, and ultimately, behavior, establishes it as a crucial element in the quest for a resilient and proactive cyber defense strategy.

2 The Controversy of Deception Technology in Security Operations

The discourse surrounding cyber deception technology in the cybersecurity community is marked by a palpable contradiction, largely rooted in varying levels of understanding and experience. On one hand, proponents of cyber deception confidently advocate for its efficacy, underscoring its role as an early warning system, its ability to unveil adversary techniques, and notably, its low false positive rate. This

supportive stance likely stems from a nuanced comprehension of cyber deception's potential when applied judiciously, such as in targeted traps or in monitoring for "hard-to-fake" behaviors, which align with the more straightforward applications like early detection mechanisms.

In contrast, the skeptics of cyber deception technology often voice concerns about its appropriateness only in mature organizations and point to higher false positive rates. This skepticism may be less about the fundamental concept of cyber deception and more about the apprehensions surrounding its complex applications, like "honey networks" or sophisticated threat intelligence systems. These more intricate implementations require a deeper understanding of network dynamics and attacker behavior, increasing the risk of false positives due to normal system interactions being misinterpreted as malicious. This viewpoint possibly reflects a psychological tendency towards caution in the face of complex, less predictable systems, and a preference for traditional, well-established security measures.

The divergence in opinions is, therefore, not just a reflection of differing attitudes towards cyber deception itself but also a manifestation of the varying levels of experience and understanding of its potential applications. While experienced practitioners might see cyber deception as a nuanced tool adaptable to specific contexts, less experienced individuals might view it as an overly complex strategy fraught with potential pitfalls. The essence of this debate lies in the realization that cyber deception, like any other tool in cybersecurity, is highly dependent on its use case and the skill with which it is implemented [6].

3 Purpose & Motivation

In the evolving domain of cybersecurity, the deployment of cyber deception techniques, particularly in creating fake user accounts for detecting password sprays and advanced threats, has garnered considerable attention for its cost-effectiveness and strategic value. My journey into the intricate world of cyber deception began through anecdotal experiences shared by colleagues, who were successfully employing canary tokens as part of their defensive arsenal. These tokens, ingeniously woven into seemingly innocuous documents, would trigger alerts upon unauthorized viewing, serving as an unobtrusive yet highly effective early warning system. The simplicity and efficiency of these tokens, requiring minimal effort for deployment yet offering substantial returns in terms of threat detection, underscored their potential as a pivotal component in a layered defense strategy.

Further exploration led me to the Active Defense Harbinger Distribution (ADHD), a curated collection of open-source tools designed for active cyber defense. This innovative platform, standing out from traditional security distributions like Kali Linux or Parrot Security, offers a unique blend of tools categorized into Annoyance, Attribution, and Attack. The philosophy behind ADHD resonates with my pursuit of cost-effective, proactive defense mechanisms. It's not just about the array of tools available but the underlying principle of making an attacker's endeavors as challenging as possible. ADHD, with its functionalities like honeypots and fake file systems, does not just complement existing security solutions; it enhances them by introducing an element of unpredictability and complexity for attackers. This approach aligns perfectly with the psychological aspect of cyber defense, where the objective is not just to block or detect threats, but to actively engage and confuse adversaries, turning their tactics against them. The incorporation of ADHD in my security repertoire represents a significant step in advancing the effectiveness of cyber deception, not just as a technical measure, but as a psychological stratagem designed to outmaneuver and demoralize potential attackers.

4 Credentials: Targeted Attacks vs Wide-Nozzle Attacks

Understanding the landscape of password spray attacks and the threat model necessitates a differentiation between targeted and wide-nozzle credential attacks.

4.1 Wide-Nozzle Profitability with Password Sprays

Wide-nozzle attacks, especially password sprays, are favored for their high profitability and lower resource expenditure [7]. Their success is often attributed to the commonality of weak passwords and the general lack of robust password policies, as evidenced by a study examining password policies across over 20,000 sites [8]. This research found pervasive weaknesses, with many sites permitting easily guessable passwords and lacking proactive measures against known compromised passwords like something you would find at "HaveIBeenPwned." Such vulnerabilities make wide-nozzle attacks particularly effective in scenarios involving advanced persistent threats (APTs), where attackers target extensive networks with the hope of stumbling upon systems with privileged access. The prevalence of simplistic passwords in large, complex IT operations, coupled with inadequate password policy enforcement, starkly highlights the susceptibility of numerous systems to wide-nozzle password spray attacks.

4.2 The Ineffectiveness of Targeted Attacks & Wordlist Generation: Comparing Threat Models

Targeted attacks, while potentially effective in certain scenarios, often hinge on specific conditions for profitability. Primarily, they are more likely to succeed when an attacker can persistently focus on a single point of entry, particularly in an offline attack context. This approach necessitates a belief in the feasibility of entry through brute force, access to relevant data for generating effective wordlists, and the capability for numerous trial attempts without immediate detection or countermeasures. However, in online environments, especially during privilege escalation or lateral movement, targeted attacks can become conspicuously risky. Repeatedly attempting to brute force a single account is a telltale sign for security systems, leading to quick detection and potential lockdown, thereby undermining the attack's stealth and efficacy.

When it comes to wide-nozzle attacks, the concept of generating optimized wordlists, as exemplified by the

use of GPT-3 models, presents a theoretically sound approach. These AI-driven models, as explored in the ACM-Research's targeted-password-guesses project, can tailor wordlists based on specific data related to the entity under attack, ostensibly increasing the chances of a successful breach [9]. However, the reality of such approaches often falls short of expectations. The primary limitation lies in the scalability of this strategy: each guess generated by the sophisticated wordlist must be tested across a potentially vast array of accounts, resulting in an exponentially growing set of combinations to try. This not only demands significant computational resources but also raises the likelihood of detection due to the volume of attempts. Therefore, while the intellectual exercise of using advanced AI models like GPT-3 for wordlist generation in password spray attacks is intriguing, the practical application of such methods is usually inefficient and less effective compared to other attack vectors.

5 Traditional Password Spray Detection Methods

Mitre ATT&CK, a globally-accessible knowledge base of adversary tactics and techniques, provides a structured approach to understanding and countering cyber threats. In the context of password spraying, this framework underscores the importance of vigilance across common management services and ports, including SSH (22/TCP), LDAP, Kerberos, RDP, and others [10].

A critical aspect of these traditional methods is the application of univariate time series analysis. This analytical technique involves examining single-variable data points over time to identify anomalies or trends. In the case of password spraying detection, univariate time series analysis is employed to spot spikes in failed authentication attempts, which may signal an ongoing attack. Such analysis can be effective in identifying large-scale, coordinated attempts to breach systems through commonly used passwords against multiple accounts, thereby sidestepping the account lockout mechanisms typically triggered by brute-force attacks on single accounts.

However, these traditional detection methods are not without their shortcomings. The reliance on log and event analysis can lead to complications, especially in environments where not all authentication attempts generate consistent or comprehensive log entries. For instance, in default settings, LDAP and Kerberos connection attempts might not trigger the same level of logging as SMB, leading to potential gaps in monitoring. This inconsistency can create blind spots in detection strategies, allowing sophisticated attackers to exploit less-monitored protocols and services. Furthermore, password spray attacks are often slow and methodical, designed to fly under the radar of typical detection mechanisms that look for rapid, high-volume attack patterns. Attackers may use low-frequency attempts spread across many accounts and services, making it challenging to distinguish malicious activity from routine authentication failures.

6 Modern Univariate Time Series Anomaly Detection

6.1 Efficacy and Evolution of Time Series Analysis in Authentication Data

The adoption of time series analysis in authentication data has proven to be a cost-effective and insightful approach for uncovering user behavior and potential security threats. This method has evolved significantly, transitioning from basic statistical techniques to more advanced ARIMA and SARIMA models, as highlighted in [11] and [12]. This shift is driven by the increasing complexity of datasets and the enhanced computational capabilities available, allowing for more accurate predictions and the efficient identification of anomalous behaviors.

6.2 Challenges in Univariate Time Series Analysis

Authentication data is often fraught with inherent patterns of seasonality and trends. For instance, login failures may spike during specific holidays or show a downward trend when users are typically asleep. Traditional analysis methods like simple moving averages or threshold-based detections often fall short in accurately capturing these nuances, as noted in [13]. The challenge lies in differentiating these natural variations from genuine anomalies, a task that requires more sophisticated analytical models.

6.3 STL Decomposition in Anomaly Detection

STL Decomposition, as explored in [11], provides a robust framework for dissecting authentication data into seasonal, trend, and residual components. This technique is particularly effective in isolating outliers like password sprays by filtering out the regular patterns of seasonality and trends. This method allows for a more precise identification of anomalies that could indicate malicious activities.

The implementation of SARIMAX, SARIMA, and ARIMA models, detailed in [11] and [13], marks a notable progression in time series analysis. These models enhance the understanding of time-dependent data patterns by integrating seasonality, trends, and even external factors, thereby offering a more refined approach for detecting anomalies like password sprays.

6.4 Transition to Machine Learning and Deep Learning Techniques

Looking ahead, the future of anomaly detection in time series analysis is veering towards the integration of machine learning and deep learning methods, as suggested in [11] and [14]. These cutting-edge techniques, capable of processing large-scale data sets and adapting to the complex dynamics of cybersecurity threats, are set to revolutionize anomaly detection. Temporal approaches like LSTM networks and autoencoders, categorized in [15], are at the forefront of this new wave of anomaly detection methodologies.

6.5 Real-World Application and the Move Towards Multivariate Analysis

The practical application of these advanced models, exemplified in [14], demonstrates their adaptability and non-intrusive deployment in various cybersecurity settings. Yet,

as pointed out in [16], there is a noticeable trend moving from univariate to multivariate analysis. This shift towards considering multiple variables, such as IP reputation and login patterns, provides a more comprehensive perspective on security threats. However, this complexity may pose accessibility challenges for security operations engineers who may not have the resources or expertise to implement sophisticated ML and AI models.

6.6 Password Sprays in the MITRE D3FEND Knowledge Graph

In the ever-evolving domain of cybersecurity, a multitude of variables play critical roles in both univariate and multivariate contexts. The MITRE DEFEND knowledge graph [17], serving as a structured compendium of cyber defensive tactics, elucidates the relationships between various types of data and their corresponding analysis techniques about attacks. This is an incredibly elegant way to threat model. For instance, methods such as session duration analysis, resource access pattern analysis, and authentication event thresholding are pivotal in univariate anomaly detection (Figure 2.) On the other hand, multivariate analysis techniques could integrate variables like user geolocation logon patterns, credential compromise scope, and network traffic community deviation to construct a more comprehensive defense posture. These methods analyze the interplay between different data types, such as authentication logs, intranet administrative network traffic, and client-server payload profiles, to detect complex anomalies like password spraying.

The application of such analytical methods maps a graph-based approach to security, where each node represents a unique variable or analysis type, and the edges signify the potential relationships or effects between them. This network of analytical methods, when effectively leveraged, can unearth subtle correlations and causalities, which might otherwise go unnoticed in a univariate approach. By employing this taxonomy, SecOps teams can adapt their strategies to encompass a wide array of attack vectors and defense mechanisms, ranging from the straightforward detection of decoy user credentials to the more nuanced analysis of protocol metadata anomalies. This adaptability not only heightens security measures but also equips personnel with a holistic view of the cyber ecosystem, enhancing their ability to preemptively counteract sophisticated cyber threats.

6.7 Balancing Sophistication and Accessibility

The integration of sophisticated anomaly detection methods within cybersecurity, particularly for teams with limited resources, necessitates a strategic approach that balances advanced analytical capabilities with practical implementation. Utilizing existing unsupervised models alongside STL (Seasonal and Trend decomposition using Loess) decomposition presents an effective gateway for security operations (SecOps) personnel to delve into more advanced analytics without overextending resources. This approach, as highlighted in [14], allows SecOps teams to leverage the power of machine learning and AI in a more accessible and manageable manner. Unsupervised models, which do not require extensive training datasets, can be deployed to identify patterns and anomalies in authentication data. When combined with STL decomposition, which effectively isolates seasonal and trend components from the actual data, SecOps teams can gain a clearer understanding of anomalies, such as password sprays, without the need for deep technical expertise in advanced AI methodologies.

The pathway to adopting more complex multivariate approaches, as discussed in [16], should be viewed as a progressive journey rather than an immediate overhaul. By starting with existing unsupervised models and STL decomposition, SecOps personnel can gradually build their analytical skills and understanding of the underlying patterns in their data. This gradual progression allows for the development of more sophisticated anomaly detection capabilities over time, aligning with the evolving nature of cyber threats. Moreover, it enables SecOps teams to efficiently allocate their limited resources, focusing on tools and methods that offer immediate value while laying the groundwork for more advanced future implementations. In essence, this approach fosters a learning environment where SecOps teams can incrementally enhance their analytical prowess, ensuring that the advancements in cybersecurity capabilities are both sustainable and aligned with the operational realities of their organizations.

7 Dilemmas in Other Credential Attack Detection Methods

7.1 The Role and Challenges of UEBA in Password Spray Detection

User and Entity Behavior Analytics (UEBA) leverages machine learning and artificial intelligence to monitor user behavior, offering a sophisticated approach to detect anomalous activities that might indicate a password spray attack. By analyzing patterns and deviations from normal behavior, UEBA can flag potential threats, effectively operating under the principle of ”you’re not supposed to be here.” However, this technology is not without its challenges. Implementing UEBA often requires divulging some level of identity information or connection to identity, which raises concerns about privacy and data security.

Previous research [18] reveals a significant gap in employee awareness regarding the extent of data collection by enterprise security systems. This lack of understanding can lead to privacy concerns and a breakdown in trust, especially when behavior analytics are used. The findings from this paper are indirectly relevant to UEBA systems, as they too collect and analyze detailed user behavior data, potentially raising similar concerns among employees.

7.2 The Double-Edged Sword of Revealing Password Hashes for Detecting Password Sprays

On the other hand, a more direct approach to detecting password sprays involves analyzing password hashes. If systems could securely analyze password hashes or use some cryptographic-related scheme, detecting password spray attacks could become more straightforward. For instance, a system could flag instances where the same password or hash is used across multiple accounts, even if attackers

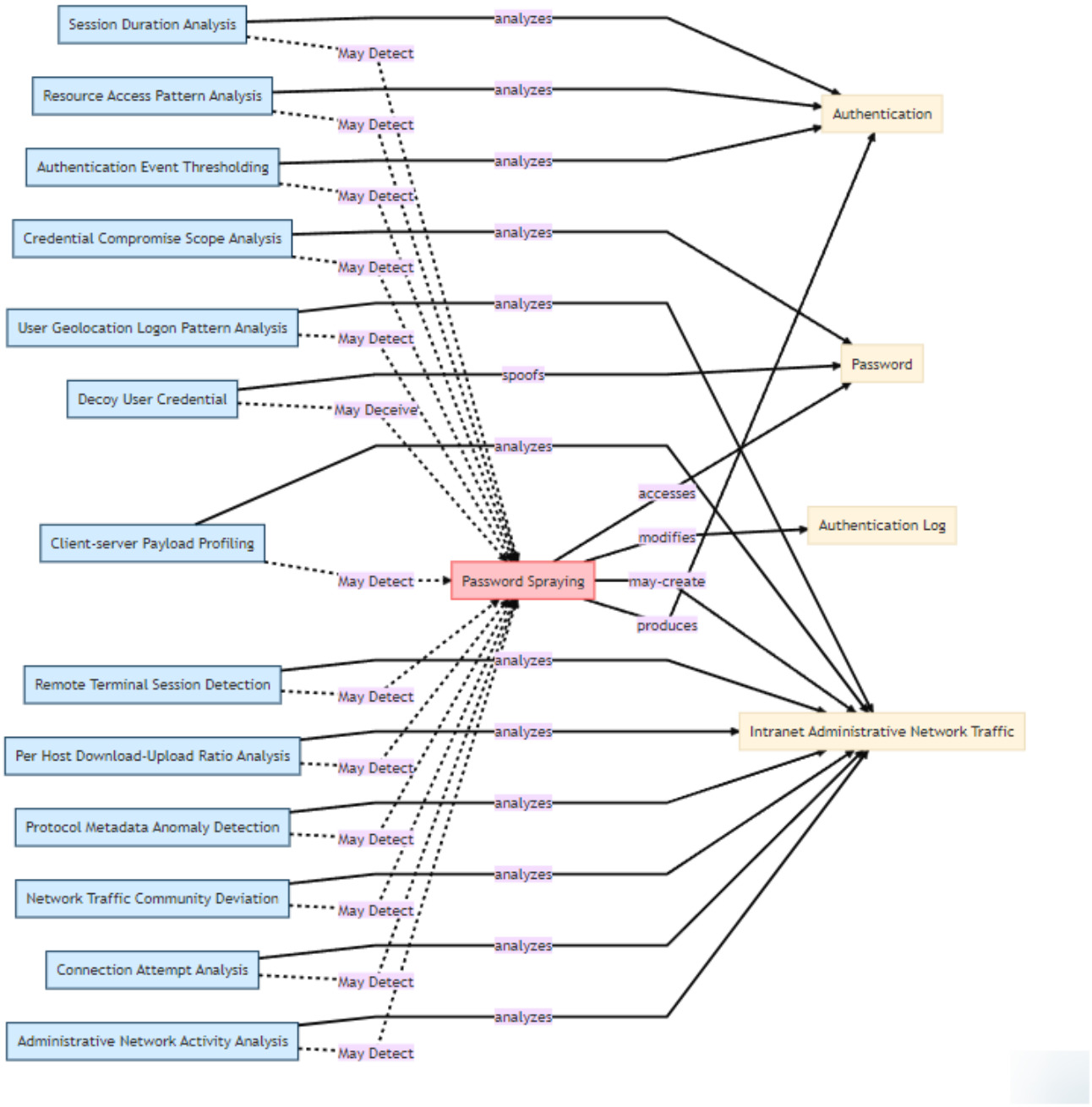

Fig. 1. The D3FEND Knowledge Graph from Mitre showing all of the related detections and data points related to Password Spraying - T1110.003.

rotate their IP addresses. However, this approach presents a significant dilemma. While providing valuable data for detection, the exposure of password hashes or cryptographic schemes could subvert security objectives [19].

Moreover, the challenges associated with this approach are further compounded by the difficulty in obtaining ground truth labels for logins [20]. The lack of definitive evidence of what constitutes an attack in the data makes it difficult to use supervised machine learning effectively for attack detection. This dilemma underscores the need for a balanced approach that considers both the effectiveness of detection methods and the potential risks to user privacy and data security.

8 Honey User Accounts for Password Spray Detection

8.1 Honey Users in Windows AD

Active Directory (AD), a directory service developed by Microsoft, is pivotal in managing permissions and access to networked resources within a Windows environment. AD stores information about objects on the network and makes this information easy for administrators and users to find and use. Many organizations employ a hybrid model

using Azure AD and AD Connect, integrating cloud and on-premise resources seamlessly. In such complex environments, detecting unauthorized access attempts like password spraying becomes challenging.

Honey user accounts, also known as decoy accounts, are strategically crafted credentials designed to deceive attackers. These accounts act as digital tripwires, alerting administrators when interacted with. Unlike genuine user accounts, they have no legitimate purpose other than serving as a security measure. Their deployment is simple: they are created within AD and monitored for any interaction. If an attacker, having infiltrated the network, attempts to use these credentials, their actions can be immediately flagged.

8.2 The Strategic Value of Honey User Accounts in Various Threat Models

In the realm of cyber threats, attackers range from "script kiddies" – inexperienced hackers using pre-written scripts to attack systems – to Advanced Persistent Threats (APTs) – highly skilled, well-resourced groups often backed by nation-states. Script kiddies typically engage in rapid and conspicuous attacks, while APTs execute stealthy, slow, and sporadic operations to avoid detection, as exemplified in various cases documented by MITRE [10].

Regardless of the attacker's sophistication, honey user accounts prove to be a highly effective early warning system. For script kiddies, their quick and noisy approach will likely trigger these decoys, instantly alerting administrators. In contrast, APTs' low and slow tactics, aiming to blend into normal network activity, can also be unveiled when they inadvertently interact with these decoy accounts.

The Honey User Hypothesis posits that employing honey user accounts in a Windows AD environment is not only cost-effective but also simpler than other detection methods, like time-series analysis where APTs can still avoid detection with slower attempt rates. The detection criterion is straightforward for honey users: any interaction with these decoy accounts is a potential unauthorized access attempt. Moreover, these accounts can be configured to permit authentication without granting actual access, further misleading attackers. This setup is crucial as it prevents attackers from using the account while allowing administrators to monitor their actions.

Another significant advantage is the maintenance of these accounts. While integrating decoy credentials within a larger decoy environment enhances their effectiveness, as suggested by MITRE's D3FEND framework [17], it is not always necessary. Simple decoy accounts with minimal privileges can still offer substantial security benefits with relatively low upkeep. The key is continuous monitoring and occasional updates to maintain their authenticity and effectiveness.

In conclusion, honey user accounts in a Windows AD environment present a versatile and low-effort security strategy suitable for various threat models. Their ease of implementation, coupled with their ability to provide early warnings of intrusion attempts with low false positives, makes them a valuable tool in the arsenal of cybersecurity defenses and potentially better than deep-learning time series-based models.

9 Implementation of Decoy User Accounts in a Windows AD Environment

9.1 Threat Model & Architecture Setup

In this purple team exercise, the primary focus is on establishing a secure yet realistic Windows Active Directory (AD) environment for Identity and Access Management (IAM), particularly targeting lateral movement and privilege escalation threat models. The assumption here is that an attacker has already gained access to the network and is aiming to perform internal reconnaissance to identify domain controllers (DCs) and admin accounts for password spraying.

The deployment infrastructure is set up in Azure, utilizing an Azure Resource Manager (ARM) template, with a daily operational cost of approximately 10 dollars. This setup includes:

- Logging infrastructure
- Separation of logical networks for attackers and defenders
- A dedicated domain controller

To simulate a realistic testing environment, the AD domain was created and "polluted" using *Badblood* [21]. This tool generates a complex environment with numerous users, organizational units, and randomized group structures, ideal for testing both attack strategies and detection mechanisms.

Honey accounts were then created within the AD using Remote Administration Tools in PowerShell. These accounts are designed to appear legitimate to attackers but are closely monitored for any unauthorized access attempts.

9.2 Attack Strategies and Tools

9.2.1 Bloodhound and Plumhound

For detecting relationships in AD, Bloodhound, which utilizes graph theory and Neo4J, was employed. Its intuitive interface and Cypher queries helped process AD relationships to identify high-value targets. Complementing Bloodhound, Plumhound [22] was used for simplifying the data analysis, catering to the needs of both blue and purple teams.

9.2.2 Bruteloops and BFG

For the actual password spray attack, two tools were utilized: Bruteloops and DomainPasswordSpray.ps1. Bruteloops, a credential attack framework, provides a nuanced approach with features such as guess scheduling, attack resumption, multiprocessing, and detailed logging. This tool's design is more aligned with APT-style attacks, focusing on a 'low and slow' strategy. On the other hand, DomainPasswordSpray.ps1 offers a more direct and "loud" approach to password spraying across the domain.

The entire exercise was conducted under a purple teaming framework, focusing on both attack execution and detection strategy development.

9.3 Detection Setup

To detect these simulated attacks, Azure Sentinel was set up alongside Azure Monitor logs. The use of Kusto Query Language (KQL) allowed for complex and efficient querying of logs to identify potential security breaches or unauthorized access attempts.

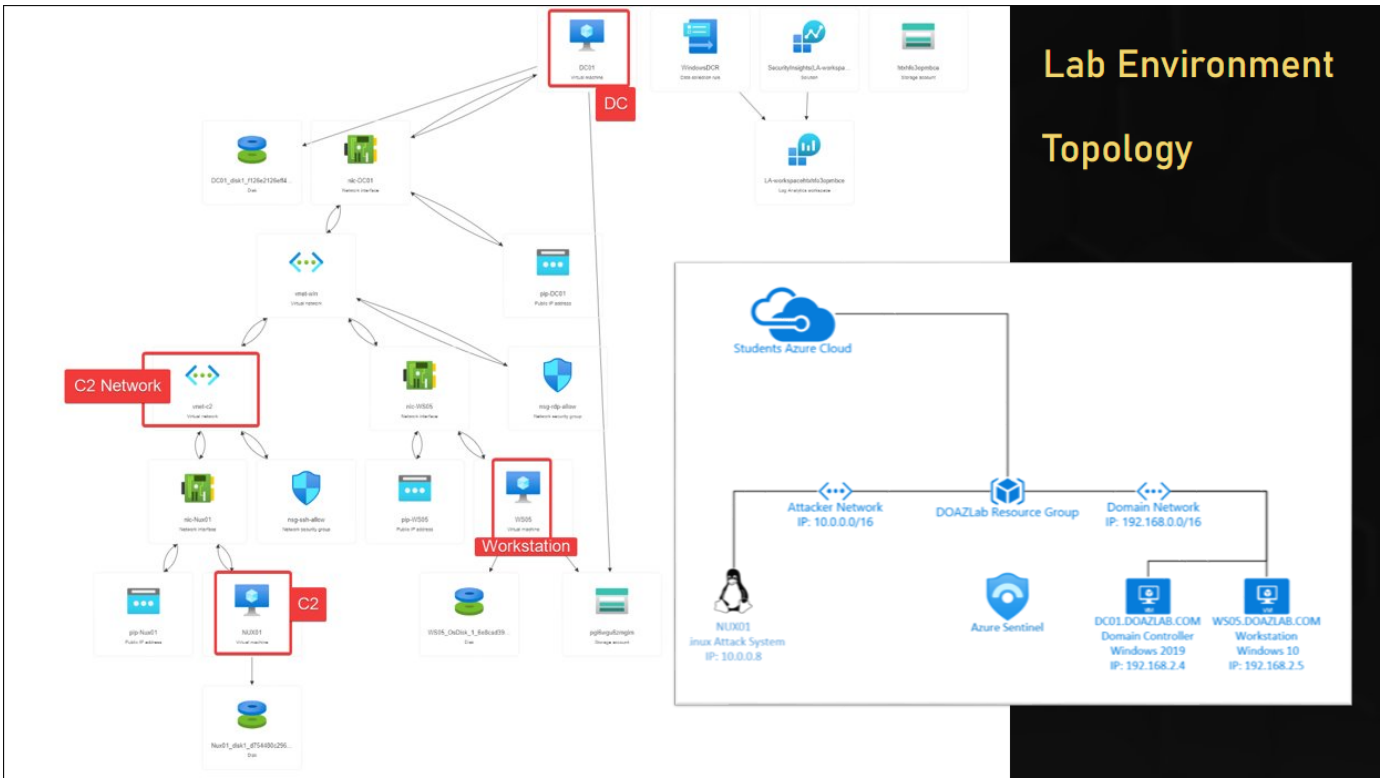

Fig. 2. Diagram of architecture and infrastructure setup in Azure Cloud. Includes: attacker/c2 network, a domain controller (DC), a workstation, and logging with Azure Sentinel.

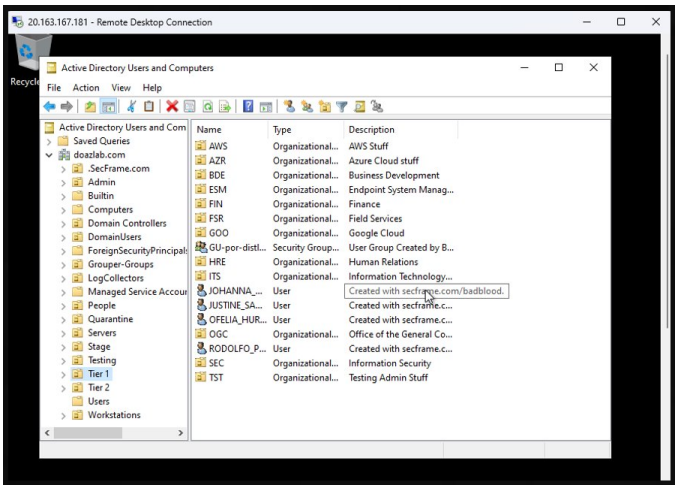

Fig. 3. Results of using Badblood - view from AD User and Computers while remotely accessing a workstation.

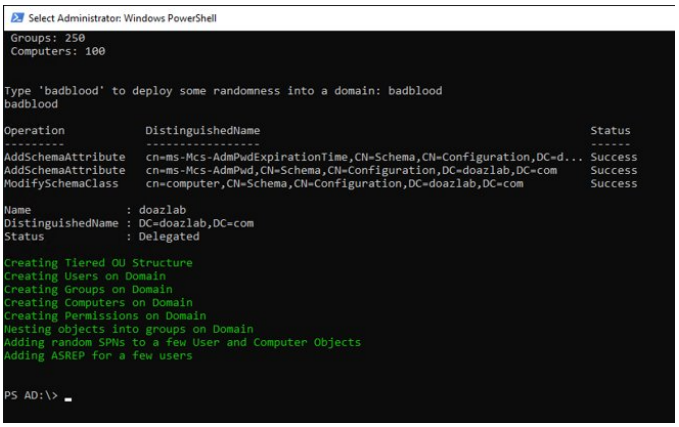

Fig. 4. CLI results of using Badblood - notice 250 groups and 100 computers

9.4 Fake User Account Generation

Generating fake user accounts involved several steps, starting with installing PowerShell dependencies on a Windows 10 system. New user objects were then added, followed by the deployment of decoy users and computers. The goal was to make these accounts appear attractive to attackers, while ensuring they had no actual system privileges. Although these accounts permitted authentication, any further access attempts were denied, and detailed logs of these attempts were maintained.

9.5 Password Spray Methodology

9.5.1 DomainPasswordSpray.ps1

This tool was used to execute a straightforward password spray across the domain. The simplicity of this tool made it ideal for a quick and broad attack, testing the overall resilience of the network against common password attacks.

9.5.2 Bruteloops Implementation

Implementing Bruteloops required additional steps, including using impacket [23] alongside Linux CLI transformations to prepare and ingest a list of accounts into Bruteloops [24]. This method's complexity mirrors more sophisticated

```
SecurityEvent
| where EventID == 4624 or EventID == 4625 or EventID == 4776
| where TimeGenerated > ago(4h)
| where Account contains "Heloise.Brinn" or Account contains "luis.graves"
| project Activity, Account, Computer, IpAddress
```

Results Chart

Activity	Account	Computer	IpAddress
4625 - An account failed to log on.	\Luis.Graves	DC01.doazlab.com	192.168.2.5
4625 - An account failed to log on.	\Heloise.Brinn	DC01.doazlab.com	192.168.2.5

Fig. 5. Results of KQL query finding login attempts to decoy accounts. Shows results from a password spray successfully (query in Listing 1, pg. 10)

APT attacks, offering a contrasting approach to Domain-PasswordSpray.ps1 [25] with a more low and slow approach. However, it was out of scope to properly do the low and slow approach as APTs tend to do this over months rather than days.

10 Results & Password Spray Detection Method Comparison

In this section, we analyze and compare the effectiveness of two distinct password spray detection methods: the use of decoy user accounts and STL (Seasonal-Trend decomposition using Loess) decomposition method, inspired by univariate time series anomaly detection research. Both methods were executed using Kusto Query Language (KQL) queries in Azure.

10.1 Fake User Account Alerts

10.1.1 Decoy Account Detection

Two decoy user accounts, Heloisse Brinn and Luis Graves, were created to test password spray detection. The results from these accounts were remarkably straightforward. By monitoring specific event ID logs associated with account login attempts, we could easily identify unauthorized access attempts. This method provided clear-cut results indicating which accounts attempted to access the decoy accounts.

10.1.2 Advantages of Decoy Accounts

The primary advantage of using decoy user accounts for detection is the directness and simplicity of the approach. Unlike other methods that require complex analysis or interpretation, decoy account alerts are binary — any access attempt is a clear indication of malicious activity. Furthermore, expanding this method is relatively easy by incorporating additional event IDs linked with these accounts.

10.2 Time Series Anomaly Detection

10.2.1 STL Decomposition Analysis

The STL decomposition method, utilizing Azure KQL's native function "series_decompose_anomalies", was effective in identifying the password spray attack (shown in Figure 6.) However, it's important to note that the effectiveness of this method in this instance may be attributed to the nature of the password spray attack — which was not low and slow but rather targeted hundreds of accounts simultaneously, creating a significant anomaly.

Fig. 6. Decomposed authentication data over time detecting the password spray as an anomaly. KQL query is shown in Listing 2 - pg. 10.

10.2.2 Limitations of STL in Subtle Attacks

In scenarios where an attacker employs a more subtle, APT-like approach (low and slow), the STL method might not be as effective in detecting password sprays. Such attacks are designed to blend in with normal traffic, making them less likely to create noticeable anomalies in time series data.

10.3 Comparing Methods

10.3.1 STL Method Versus Decoy User Accounts

While the STL method can provide insights into domain activities and identify large-scale anomalies, it falls short in several areas compared to the decoy user account method. The STL method indicates that an attack is happening but requires additional queries and analysis to delve deeper.

10.3.2 Superiority of Decoy User Accounts

Conversely, the decoy user account method provides immediate, actionable data. Not only does it show that an attack is happening, but it also identifies the accounts involved in the attempt and other metadata that can be useful for immediate triage. This feature is particularly beneficial for integrating with security automation tools, enabling immediate and efficient response without further computing resources by redirecting fields from the query output into other tooling.

Additionally, the versatility of the decoy user account method makes it suitable for various analytics platforms, whereas implementing STL decomposition and models like SARIMA can be time-consuming and complex on some platforms.

11 Conclusion

The research presented in this paper underscores the significance and superiority of employing decoy user accounts in

```
SecurityEvent
| where EventID == 4624 or EventID == 4625 or EventID == 4776
| where TimeGenerated > ago(4h)
| where Account contains "Heloise.Brinn" or Account contains "luis.graves"
| project Activity, Account, Computer, IpAddress
```

Listing 1: KQL query that looks for login attempt event IDs related to the decoy user accounts. Password attempts are shown easily as failed logins as shown in Figure 5. - pg. 9

```
SecurityEvent
| where TimeGenerated >= ago(14d)
// | where FailureReason!="%%2313"
| where EventID == 4625
| summarize count_=count() by bin(TimeGenerated, 1h)
| make-series num=avg(count_) on TimeGenerated from ago(14d) to ago(0d) step 24h
| extend (anomalies, score, baseline) = series_decompose_anomalies(num, 1.5, -1, 'linefit')
| render timechart with(title='Failed Authentication, decomposition', ysplit=panels)
```

Listing 2: KQL query that looks for login failure event IDs, uses STL decomposition to find the residual despite the "ups" and "downs" and trend, then looks for anomalies based on the residual/score spiking upwards. Results shown in Figure 6 - pg. 9.

a Windows AD environment as a cost-effective and efficient method for detecting password spray attacks. The analysis of the decoy user accounts, juxtaposed against the STL decomposition method for time series anomaly detection, revealed clear advantages in terms of ease of implementation, directness of attack detection, and adaptability across different analytics platforms.

Decoy user accounts, serving as digital tripwires, offer a straightforward approach to identifying unauthorized access attempts. Their ability to provide immediate, actionable data without necessitating complex analyses or interpretation positions them as an invaluable tool in the cybersecurity arsenal. This method's adaptability to various threat models, from inexperienced attackers to sophisticated APTs, further reinforces its strategic value.

Conversely, while time series analysis using STL decomposition can identify large-scale anomalies and offer insights into domain activities, it falls short in effectively detecting more subtle, low-and-slow attack strategies characteristic of APTs. The need for additional queries and analysis to interpret STL decomposition results highlights the inherent complexity and limitations of this approach.

In summary, this paper illustrates the evolving landscape of cybersecurity defense mechanisms, emphasizing the importance of innovative strategies like cyber deception to counteract sophisticated cyber threats. The successful implementation and analysis of decoy user accounts in this study provide a blueprint for cybersecurity practitioners seeking efficient, cost-effective, and reliable methods for early detection and response to password spray attacks. The findings advocate for a paradigm shift in cybersecurity strategies, moving away from traditional reactive postures to proactive engagement, thereby equipping defenders with tools and techniques to outmaneuver and neutralize potential attackers.

REFERENCES

[1] M. C. Libicki, L. Ablon, and T. Webb, *The defender's dilemma: Charting a course toward cybersecurity*. RAND Corporation, 2015.

[2] *Examining the efficacy of decoy-based and psychological cyber deception*, 30th USENIX Security Symposium. USENIX Association, 08 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/ferguson-walter

[3] *Game theory approaches for evaluating the deception-based moving target defense*. Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3560828.3563995

[4] *CONCEAL: A strategy composition for resilient cyber deception-framework, metrics and deployment*, 2018.

[5] *SODA: A system for cyber deception orchestration and automation*. Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3485832.3485918

[6] M. J. Carey and J. Jin, *Tribe of hackers blue team*. John Wiley Sons, 08 2020.

[7] S. Wiefling, P. R. Jørgensen, S. Thunem, and L. L. Iacono, "Pump up password security! evaluating and enhancing risk-based authentication on a real-world large-scale online service," *ACM Trans. Priv. Secur.*, vol. 26, no. 1, 11 2022. [Online]. Available: https://doi.org/10.1145/3546069

[8] *Measuring website password creation policies at scale*, 2023 ACM SIGSAC Conference on Computer and Communications Security. ACM, 11 2023. [Online]. Available: http://dx.doi.org/10.1145/3576915.3623156

[9] R. Hauksson and B. Johnson, "Automating targeted password guessing," GitHub, 03 2022. [Online]. Available: https://github.com/ACM-Research/targeted-password-guesses

[10] T. M. Corporation, "Brute force: Password spraying, sub-technique t1110.003 - enterprise — mitre attck®," attack.mitre.org, 2023. [Online]. Available: https://attack.mitre.org/techniques/T1110/003/

[11] M. Braei and S. Wagner, "Anomaly detection in univariate time-series: A survey on the state-of-the-art," 2020.

[12] A. Blazquez-Garcia, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, 04 2021. [Online]. Available: https://doi.org/10.1145/3444690

[13] J. Kohlrausch and E. A. Brin, "Arima supplemented security metrics for quality assurance and situational awareness," *Digital Threats*, vol. 1, no. 1, 03 2020. [Online]. Available: https://doi.org/10.1145/3376926

[14] G. Apruzzese, P. Laskov, E. , W. Mallouli, B. Rapa, A. V. Grammatopoulos, and D. Franco, "The role of machine learning in cybersecurity," *Digital Threats*, vol. 4, no. 1, 03 2023. [Online]. Available: https://doi.org/10.1145/3545574

[15] A. Almeida, S. Bras, S. Sargento, and F. Pinto, "Time series big data: a survey on data stream frameworks, analysis and algorithms," *Journal of Big Data*, vol. 10, 05 2023.

[16] A. Weinart, "Advancing password spray attack detection," TECHCOMMUNITY.MICROSOFT.COM, 10 2020. [Online]. Available:

https://techcommunity.microsoft.com/t5/microsoft-entra-blog/advancing-password-spray-attack-detection/ba-p/1276936
[17] T. M. Corporation, "Offensive technique details — mitre d3fend™," d3fend.mitre.org, 2023. [Online]. Available: https://d3fend.mitre.org/offensive-technique/attack/T1110.003/
[18] *"My Privacy for their Security": Employees privacy perspectives and expectations when using enterprise security software*, 32nd USENIX Security Symposium. USENIX Association, 08 2023. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/stegman
[19] *Gossamer: Securely measuring password-based logins*, 31st USENIX Security Symposium. USENIX Association, 08 2022. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/sanusi-bohuk
[20] *Araña: Discovering and characterizing password guessing attacks in practice*, 32nd USENIX Security Symposium. USENIX Association, 08 2023. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/islam
[21] D. Rowe, "Badblood," GitHub, 07 2023. [Online]. Available: https://github.com/davidprowe/BadBlood
[22] K. Ickler, "Plumhound - bloodhoundad report engine for security teams," GitHub, 12 2022. [Online]. Available: https://github.com/PlumHound/PlumHound
[23] coresecurity, "Impacket," GitHub, 01 2023. [Online]. Available: https://github.com/fortra/impacket
[24] J. Angel, "Bruteloops," GitHub, 12 2023. [Online]. Available: https://github.com/ImpostorKeanu/BruteLoops
[25] dafthack, "dafthack/domainpasswordspray," GitHub, 06 2020. [Online]. Available: https://github.com/dafthack/DomainPasswordSpray