THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

EXPANDING AUTHENTICATION LOG UTILITY THROUGH NOVEL EVENT AGGREGATION: A GENERALIZABLE APPROACH

by
Seth G. Hastings

A dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
in the Discipline of Computer Science

The Graduate School

The University of Tulsa

2025

THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

EXPANDING AUTHENTICATION LOG UTILITY THROUGH NOVEL EVENT AGGREGATION: A GENERALIZABLE APPROACH

by
Seth G. Hastings

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

COMPUTER SCIENCE

By Dissertation Committee

Tyler Moore, Chair
Sal Aurigemma
Bradley Brummel
Mauricio Papa
John Hale
Chris Fennel

COPYRIGHT STATEMENT

ABSTRACT

Seth G. Hastings (Doctor of Philosophy in Computer Science)

Novel Event Aggregation

Directed by Tyler Moore

104 pp., Chapter 6: SOC Dashboard Utility

(206 words)

Modern organizations increasingly mandate multifactor authentication (MFA) to bolster security, yet the raw logs these systems generate remain underutilized due to noise and complexity. This dissertation presents a novel event-based framework that aggregates and labels authentication logs into coherent "events," offering a clearer lens on real-world user behavior. First, we detail a methodology for transforming raw sign-in data into streamlined records of user-centric authentication attempts, covering both technical states (e.g., success, partial interruption, errors) and human inputs (e.g., password entries, MFA responses). Through this process, the dissertation quantifies the user costs of enterprise MFA policies, showing how changes to MFA methods can elevate login failures and prolong lockouts. Next, we link psychometric constructs, such as security-related stress and self-efficacy, to observed authentication performance, underscoring the interplay between human factors and security outcomes. Finally, we deploy a real-time SOC dashboard that leverages aggregated events to identify struggling users, neglected applications, and suspicious behavior patterns. Analyst feedback indicates that the dashboard significantly streamlines incident investigation and heightens situational awareness. This comprehensive exploration, spanning event methodology, user-cost analysis, psychometric correlations, and SOC integration, demonstrates that recasting noisy authentication logs as interpretable events not only ad-

vances academic understanding of MFA usage but also drives operational impact through improved security workflows.

ACKNOWLEDGEMENTS

Firstly, I extend my gratitude to my friends and family for their unwavering support throughout my academic journey. I would like to thank Dr. Tyler Moore for his mentorship, patience, and guidance throughout this graduate program. Despite his demanding schedule, he ensured that I always had someone to turn to during critical moments, and I aspire to become a fraction of the researcher he represents. I would like to thank my committee members not only for their role in this dissertation, but also in their roles as professors and co-authors. A huge thank you to the IT and IT Security staff at the University of Tulsa for their assistance throughout my program at TU, specifically Reid VanDeventer and Tyler Burroughs. Without a department that values research and innovation, this research would have been impossible. I also want to thank my alma-mater John Brown University; without the challenging education I received there, and the support and encouragement of Dr. Song, Dr. Gilmour, and many others, I would not have been prepared to succeed in this work. Lastly, I would like to thank the George Kaiser Family Foundation and the University of Tulsa for my position in the Cyber Fellows program. Without this program I likely would not be in graduate school, and would have missed out on an invaluable opportunity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Multifactor Authentication (MFA) has become a cornerstone in modern cybersecurity measures. Critical to any authentication system are the logs generated when that system is used. This dissertation seeks to expand and enhance the use of authentication logs, as they are ubiquitous and widely available to organizations. These logs often serve as a front line of defense in detection and diagnosis of everything from systemic issues to targeted account compromises. The importance of authentication logs to the broader MFA ecosystem is multi-faceted. MFA is still a relatively immature technology in terms of its adoption, despite clear indications of its positive impact on security. Prior work on MFA has largely focused on factors impacting adoption rates and reactions, though more recently focus has broadened to Multi-Factor Authentication (MFA) performance, as with [38] and [39]. End users are still split, with companies such as Microsoft reporting low adoption rates seen wherever adoption is optional, including enterprise consumers [30].

Organizations increasingly mandate the use of MFA by their users, rather than providing an option for opt-in participation [1]. In these environments, the burden imposed is often not explicitly accounted for. MFA deployments typically exhibit a multi-modal configuration, allowing users the flexibility to select from various technologies to fulfill the second factor of authentication. However, it should be noted that users may lack the necessary expertise to identify an option that is both user-friendly and efficient for their needs [22]. Authentication logs may be an under-utilized resource in measuring the costs and benefits associated with these systems.

In this dissertation we explore the utility of authentication logs and their derivatives for usability research, measurement of authentication burdens or “costs”, and practical applications in security operation centers (SOCs). We believe a derivative form of authentication logs which focus on the user interaction would benefit usability research through more precise and contextual capture of user experience, enabling more complex analysis. Similarly, such a dataset would provide a straightforward measure of time costs and authentication frictions as experienced by the user. This may allow organizations to more reliably estimate and measure the costs and benefits of various MFA implementations and configurations. Given the ability to measure user MFA performance, we can explore the connection between user psychology and MFA outcomes. Understanding between-person differences in MFA usage and outcomes may enable user-specific interventions or configurations to maximize user success and security.

A more accurate and sensitive measure of user activity naturally lends itself to security-related purposes, such as proactive monitoring and alert investigations. In pursuit of this goal, we scrutinize metrics derived from authentication events to evaluate their pertinence to key security tasks. We assess the utility of these metrics within a Security Operations Center (SOC) framework by deploying a security dashboard that correlates security alerts with the derived metrics.

1.1 Prior Work

Existing work investigating authentication use has focused on adoption rates, usability, and user attitudes towards the technology. While qualitative feedback on user experience is valuable, it should be complemented by quantitative empirical measures where possible to more precisely localize delays, failures, and other pain points that contribute to a negative experience with authentication. In this section, we review the pertinent literature pertaining to each application of authentication logs: usability, measurement, psychological factors, and security analyst task relevancy.

1.1.1 Authentication and Usability/Cost

In 2018 Reese et al. examined the usability of multifactor authentication in the context of a university [38], and proposed a four-phase model of user behavior that describes the adoption and use cycle. They specifically examined the usability of Yubikeys, surveying the participants after they had setup the devices, and following up after four weeks for a semi-structured interview. They found that while most users recognized the potential security benefit, some did not find the additional trouble worthwhile when used for non-critical accounts. Other pain points included instances of authentication where their second factor was not immediately available to them, and struggles to input verification codes within the designated time frame.

Concurrently in 2018, Colnage et al. published a study from Carnegie Mellon University observing the deployment of a 2FA system utilizing Duo [10]. They explored user behaviors and opinions around mandatory adoption and analyzed usage data including over one million authentication attempts and many help-desk tickets. This was combined with two online surveys; the first sent prior to the MFA deadline, and the second three months after. In survey one they focused on perception of Duo and 2FA, likelihood of adoption, perceived advantages/disadvantages, past and present 2FA usage, and six constructs: Security, Tranquility, Fun, Ease, Difficulty, and Annoyance. Survey two asked about Duo activation, credential sharing, patterns of Duo and computer use, 2FA use on other accounts, and the same six constructs from survey 1. They found 40% of participants had prior 2FA experience, and more than half were using CMU Duo on a weekly or greater basis. This was compared to the log data, which showed users were under reporting their use, with over half of users averaging more than one Duo use per day. They found that while making adoption mandatory increased negative perception over those who adopted voluntarily, attitudes towards adoption improved across all six constructs between pre- and post-activation of Duo.

In 2020 Reynolds et al. published a study tracking users through the first 90 days of MFA use in a university setting, and reported on most commonly failed modes of MFA, as well as average times to authenticate and the time between a failed attempt and subsequent successful login, dubbed “recovery time” [39]. They utilized sign-in logs and performed a few data cleaning procedures, removing duplicate records and malformed logs. Their results showed a large divide in several measures of success, such as recovery time. While most errors led to a recovery time of under a minute, 20% of users failed to authenticate after such an error until the next day, which represents a significant loss of functionality. They also report on 2nd factor usage rates, which showed a large preference for push notification. Help desk tickets were also collected and examined qualitatively.

These works set the stage for our current research enhancing the interpretability of authentication logs and assessing their utility in both research and SOC implementations. This example of capturing authentication experience through authentication logs showed utility but had limitations. Comparison to survey data showed that user’s self-reported MFA usage was under-estimated, highlighting the importance of an objective measure. Metrics for authentication friction and success were derived from simple counts of errors and couldn’t precisely quantify the time to authenticate or categorize the various errors encountered. We extend this methodology with a new filtration and aggregation process in Section 2.2 to produce more sensitive and informative measures of authentication experience.

1.1.2 Authentication and Security

The body of work researching the identification of security issues or compromises in security logs generally is large and wide ranging in scope. In this dissertation, we focus on those works which specifically examine use analysis of authentication logs. A recent study by Sonneveld et al. touches on this idea with the goal of extracting security relevant information from non-intrusive data available to a Security Operations Center (SOC) [41]. They analyzed data reflecting the resource users accessed, the time they accessed it, and in what manner. Through what they call “access behavior profiling”, researchers were able to identify all users with the “ITAdmin” role. They also investigated deviation from baseline cluster location as an indicator of insider threats. Using the “Insider Threat” data set, they were able to detect 80% of insider threats within the ITAdmin group [26]. When applying the methodology to real-world data, cluster consistency dropped by 50%, partially attributed to the differences in granularity in the most relevant features from each data set. For more work on clustering users, see [18] and [17].

Similarly, Liu et al. [16] used a private dataset of 4 million logs to demonstrate a behavior-based authentication compromise detection model. They used only two features to model users: consecutive failures and login time. This paper is particularly interesting due to their similar method of event construction: raw log rows are aggregated as series of 0-n failures leading up to a success. This method accomplishes the same goal of capturing a complete user interaction as its atomic event, though there are some notable differences. The gap between failed attempts has no time cap, so a single event may span a large time frame, even with long periods of non-interaction, which is not true to the user experience. In addition, if events are not completed successfully by EOD, the event is considered "incomplete", and dropped from the dataset. The probabilistic model developed by Liu et al. demonstrated a good true positive false positive trade off with high prediction accuracy and low FPR at a low computational cost. See Bian et al [7] for similar work focusing on identifying lateral movement.

Finally, we note the work of Alahmadi [3], who conducted a survey of Security Operations Center (SOC) practitioners to investigate analysts' perspectives on security alerts. The survey reveals an excessive volume of alerts encountered across organizations, leading to increased analyst fatigue and a higher incidence of human error. This issue is further compounded by the low interpretability of the generated alerts. These findings, in conjunction with those of Zhao et al. [46], who identified that log data is utilized in over 30% of incident diagnoses, emphasize poor interpretability as a limiting factor in both the accuracy and actionability of generated alerts, and advocate for systems that combine a level of domain knowledge with the raw data to produce logs and alerts than are more easily interpreted.

1.1.3 Psychology and Authentication

Security-related stress (SRS) has been a focal point in understanding the relationship between psychological factors and information security compliance. D'Arcy et al. (2014) conceptualized SRS as a second-order construct, encapsulating dimensions of security-related overload, uncertainty, and complexity. This construct effectively delineates the relationship between various stressors and the intention to violate information security policies (ISP). Their investigation illuminated key stressors, including security demands, overload, complexity, and uncertainty, which collectively contribute to the phenomenon of SRS [14].

Moody and Galletta (2015) expanded this research by exploring the impact of stress on online information retrieval performance. They proposed an "inverted-U" relationship, where moderate stress levels could potentially enhance performance, while both low and high stress levels negatively affect it. Their findings highlighted the significance of time constraints and information scent in influencing user stress and performance [31].

Ament and Haag (2016) provided an empirical test of a multidimensional construct of security-related stress, revealing mixed effects on ISP compliance intentions. They introduced different stressors, including invasion of privacy and job insecurity, showing how these factors collectively contribute to overall security-related stress [4]. In the same year, Lee et al. (2016) investigated the impact of work overload and privacy invasion as stressors in information security stress (ISS). They found that work overload significantly influences ISS, particularly in technical, security-oriented organizations. Attitudes toward ISP compliance, prior security knowledge, and perceived security threats were identified as mitigating factors [25].

Belk et al. (2017) examined the difference in authentication performance across authentication devices for users classified as field-dependent or field-independent. They highlighted that visual perceptiveness, or the ability to interpret the surrounding environment by processing information in visible light, plays a significant role in authentication

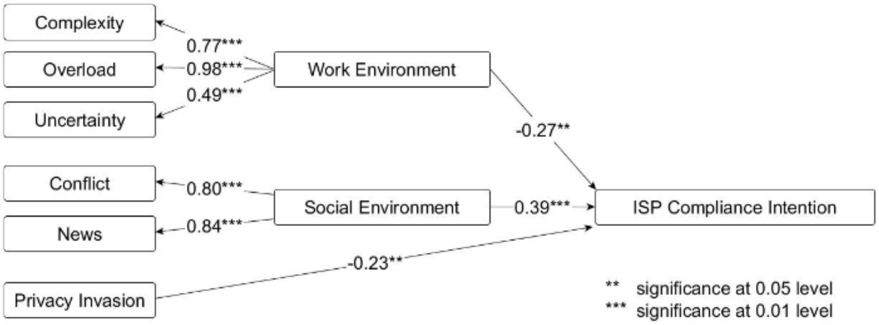


Figure 1.1: Ament and Haag Construct

performance [6]. Hwang and Cha (2018) focused on the role of technostress creators and role stress in compliance with information security for employees. Their results indicated that technostress negatively impacts compliance by decreasing organizational commitment, with promotion focus moderating the relationship between technostress and role stress [20].

Furthering this, D'Arcy and Teh (2019) examined the daily variability of security-related stress and its impact on ISP compliance, emphasizing the role of emotions in the coping process. They highlighted how certain stressors, categorized as hindrance stressors, can deplete employee resources leading to negative outcomes such as psychological strain and reduced compliance [15]. Maier et al. (2019) explored how personality traits influence the perception of technostress. They identified that IT mindfulness, a dynamic trait, significantly moderates technostress perceptions, suggesting that some users are more susceptible than others to stress induced by security demands [28].

Nasirpouri and Biros (2020) provided a comprehensive overview of technostress, identifying five key stressors: techno-overload, techno-invasion, techno-complexity, techno-uncertainty, and techno-insecurity. They found that these stressors lead to adverse outcomes such as reduced productivity, ISP non-compliance, and discontinued IT use [32]. Cram et al. (2021) introduced the concept of security fatigue, identifying its antecedents and consequences on ISP compliance. They described security fatigue through symptoms such as frustration, tiredness, and hopelessness, which significantly impact employee compliance behaviors [12].

Kim et al. (2022) used eye-tracking technology to study the impact of technostress on cognitive load. Their study differentiated between low-stress and high-stress individuals, showing that high-stress participants exhibited more distractions and slower task completion times [23]. Jeon et al. (2023) focused on the emotional responses of employees to security policy compliance, particularly the role of frustration. They found that frustration negatively impacts compliance, but this effect can be mitigated by providing autonomy to employees [21].

Finally, there are three recent meta-analyses on this topic. Yuan et al. (2023) investigated the effects of specific technostressors on strain and job performance, including techno-complexity, techno-insecurity, and techno-uncertainty. The results revealed that techno-complexity and techno-insecurity were significant predictors of both strain and job performance. Employees facing high levels of these stressors experienced increased strain and decreased job performance. Interestingly, the study found that techno-uncertainty did not have a significant impact on job performance, suggesting that not all technostressors equally affect employees. Their paper also highlighted the moderating roles of demographic factors, such as age and job experience, in shaping the relationship between technostressors and job outcomes. They concluded that tailored interventions considering these demographic factors could help mitigate the negative effects of technostress on employees.[45]

Concurrently, Aggarwal and Dhurkari conducted a comprehensive meta-analysis to investigate the association between stress and non-compliance intention towards information security policy (ISP). Their findings included a weak positive correlation between stress and ISP non-compliance, indicating that higher stress levels are associated with a slight increase in non-compliance. Notably, the study emphasizes the importance of demographic characteristics—such as age, geographic location, and employment status—as moderating factors in this relationship.[2]

Lastly, Singh et al. (2023) [40] provided a systematic review of the literature on stress

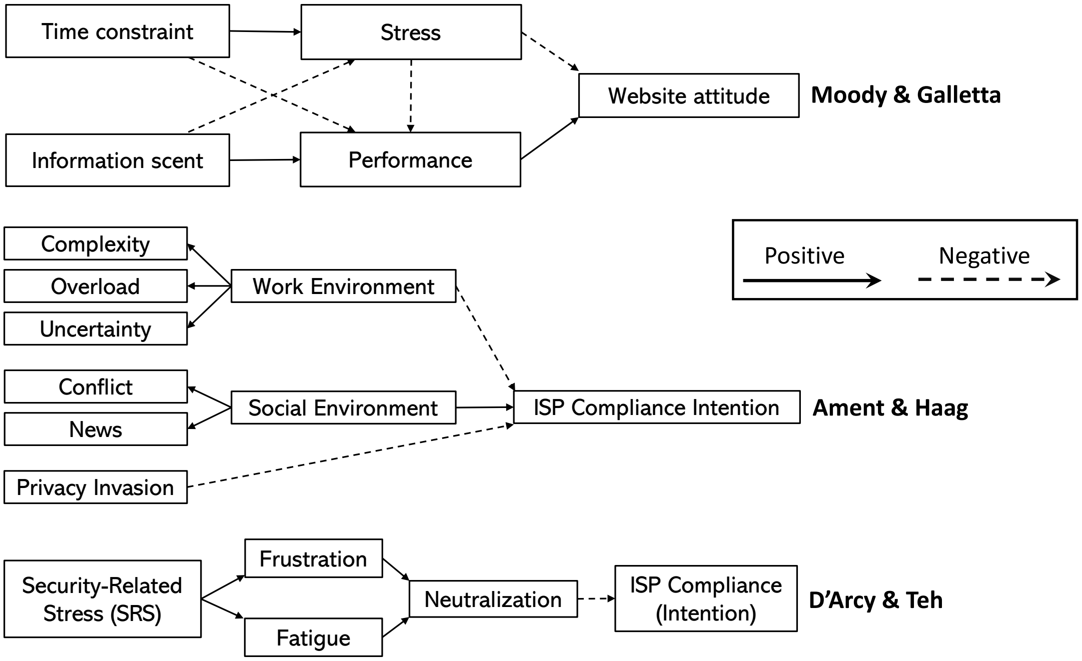


Figure 1.2: SRS-related behavioral models

in the cybersecurity profession, focusing on the appraisal process of security demands and the outcomes of stress beyond mere compliance. Their review identified unique stressors faced by cybersecurity professionals, such as constant exposure to high-stakes security threats and the pressure to maintain vigilance against potential breaches. Their call to action highlighted the need to go beyond compliance metrics and consider the holistic impact of stress on cybersecurity professionals.

Figure 1.2 shows the models used in this research by Ament and Haag, D'arcy and Teh, and Moody and Galletta. Across these different models of the impact of stress on cyber security, the main commonality is that they predict attitudes of intentions with the link from intention to future behavior left to be assumed beyond the model.

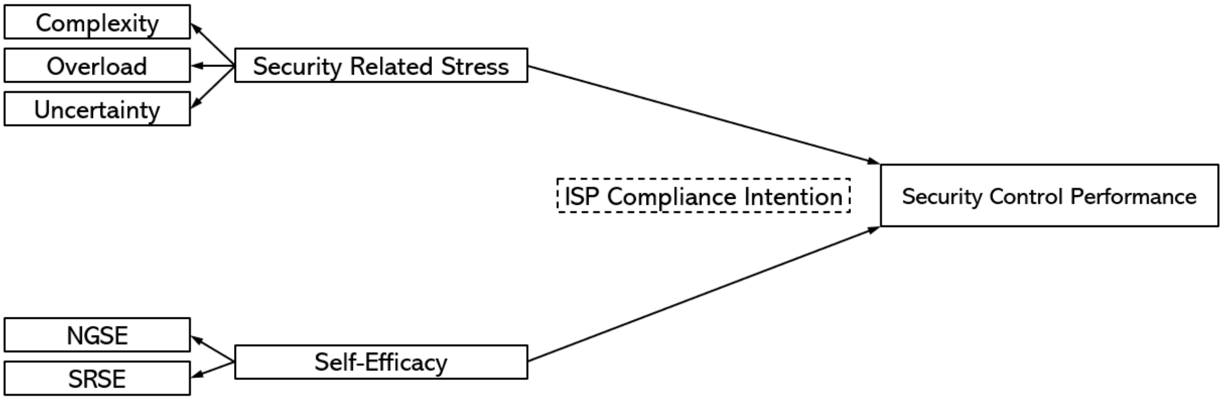


Figure 1.3: Behavioral model adopted by this study

My work builds on this growing body of research through a longitudinal study of actual security control performance derived from Azure sign-in logs at a University, as shown in Figure 1.3. In Section 4.3 we compare self-reported stress and efficacy measures to observed security control performance without relying on self-reported compliance intentions.[1] This enables a more direct measure of any links between stress, self-efficacy, and security control performance.

1.2 Structure and Contribution of this Thesis

1.2.1 Thesis Statement

This dissertation introduces a new approach to distill user experience and authentication costs from unrefined log data, presents empirical findings on the heightened burdens introduced by a change in two-factor authentication methods, investigates how security-related stress correlates with authentication performance, and evaluates the practical benefits of applying this event approach within a university Security Operations Center (SOC).

[1] The constructs used in this study were selected and collected prior to the formation of the security control performance measure.

1.2.2 Structure

In chapter 2 the event methodology is presented, this is the foundation for all following research. Raw logs are labeled, filtered, and aggregated into user-centric single row representations of an authentication experience. Chapter 3 presents an analysis of the user costs of enterprise MFA policies in a university context. A policy change increasing the security of MFA response options was shown to increase the time users spent unauthenticated after failure. Chapter 4 combines our unique, event-level dataset with a survey of users to analyze the connection between users psychometric profiles and their MFA performance. Psychometric constructs Security Related Self-Efficacy, New General Self-Efficacy, Security Related Overload, Security Related Stress, and Security Related Complexity were measured and correlated with authentication outcomes using event data. Chapter 5 explores the user-level and application-level performance metrics and utilities most relevant to a SOC. Over 50% of applications at the University of Tulsa were found to be lapsed. Error composition was found to be a primary driver of failure, with non-user errors resulting in failure in over 80% of represented events. User errors were found to be more common and less severe. Chapter 5 also describes the construction and deployment of a dashboard tool utilizing the "event" methodology into a University SOC. The dashboard offers access to processed event logs, flags struggling users and applications, and assists analysts with alert investigation through log correlation and data visualizations. SOC analyst feedback and usage of provided tools is discussed. Chapter 6 concludes with a discussion of the observed utility of authentication events across applications.

1.2.3 Contributions

Research contributions include both the methodology of event construction with its derivatives and its novel application in a variety of analyses. Chapter 2 presents the novel event dataset; this presents a unique form of user-centric authentication dataset which removes noise and focuses on interactive user experience. Chapter 3 presents a first of its kind analysis of the impact of an MFA policy change on the user-experience. Results confirm the measurable impact on the user experience. In Chapter 4, we present novel analysis on the relationship between Security Related Stress and other constructs on user multifactor authentication performance. This analysis confirms the presence of meaningful between-person differences in authentication outcomes that are related to psychological traits. In Chapter 5, we demonstrate the utility of the event log to a SOC through example metrics and analyses. Further, an implementation of the event methodology is deployed in a University SOC and analyst feedback is collected and discussed.

1.3 Authorship Statement

Chapters 2 through 5 of this dissertation have been adapted from manuscripts that have been published or are currently under review in academic journals and conferences. The research described was conducted collaboratively with co-authors who contributed significantly to the conceptualization, methodology, analysis, and interpretation of the studies.

Chapters 2 (Event Methodology) includes content from the paper titled "Transforming Raw Authentication Logs into Interpretable Events" co-authored with Dr. Tyler Moore, Philip Shumway, and Corey Bolger.

Chapter 3 (User Costs of Enterprise Multifactor Authentication Policies) includes material published as "Quantifying opportunity costs of enhanced security in multifactor authentication" by Seth G. Hastings, Dr. Tyler Moore, Dr. Neil Gandal, and Noa Barnir.

Chapter 4 (Psychometric Constructs and User Multifactor Authentication) contains data and analysis from the paper "Psychometric Determinants of Multifactor Authentication Outcomes," co-authored Dr. Tyler Moore, Dr. Sal Aurigemma, and Dr. Bradley Brummel.

Chapter 5 (SOC Dashboard Utility) contains data and analysis from the paper "Authentication-Event Processing for Enhanced SOC Investigations," was co-authored by Dr. Tyler Moore.

The contributions of each co-author are gratefully acknowledged. I confirm that I am the primary researcher and author of all chapters of this dissertation, responsible for data collection, analysis, and writing, under the guidance and supervision of my dissertation committee members.

CHAPTER 2
EVENT METHODOLOGY

2.1 Introduction

Currently, authentication logs are utilized to investigate user and application issues, as well as serve as sources for systems that generate alerts regarding suspicious activities [41, 35]. For instance, Security Operations Center (SOC) analysts can identify potential account takeovers when logs indicate login attempts from an unexpected country or exhibit a high frequency of failures. There is a growing trend of employing artificial intelligence (AI) and machine learning (ML) models to flag anomalies, which promises to considerably reduce the time to detection. However, Zhao et al. (2021) [46] have identified several limitations, including difficulties in handling complex abnormal log patterns, poor interpretability of alerts, and an insufficient level of domain knowledge.

Traditional monitoring involves engineers examining logs and writing keyword and regular expression-based rules for detection. This methodology is becoming increasingly complex due to the proliferation of components and the diversity of logs, which leads to noisy datasets that necessitate considerable domain knowledge for interpretation. New and updated service components generate an ever-expanding variety of log messages. While AI and ML systems can offer sensitivity to abnormality, they present challenges with interpretation. Engineers may be notified that a particular state is anomalous; however, the reasons behind the anomaly and the characteristics of a “normal” pattern often remain unclear.

Raw authentication logs exhibit significant levels of noise. These logs were not designed with the objective of facilitating straightforward interpretability. A single login attempt frequently generates numerous log entries, each seemingly unrelated to the others. Navigating through this complexity, whether manually or through automated systems, presents significant challenges. In this paper, we delineate a methodology for constructing interpretable, user-centric "event logs" derived from raw authentication logs. This process aims to mitigate noise, eliminate redundant entries, and amalgamate entries into distinct user experiences with enhanced data attributes.

2.2 Methodology for Constructing Authentication Events

Using data obtained through the University IT department, and approved for analysis by the Institutional Review Board (IRB), we developed a process to capture user authentication events from raw authentication logs. We define an event as:

> The events captured in log data that are directly experienced by a user commence with the initiation of authentication to a specific application and conclude with the eventual success, failure, or abandonment.[1]

By reducing sign-in logs to atomic events directly experienced by the user, we can construct event-based metrics of usage and performance while reducing noise and increasing interpretability. In this section, we provide an overview of the process for translating authentication logs into distinct events.

2.2.1 Process Overview

Before we dive into details of the process, we first give a high level example in Figure 2.1. The steps are:

1. **De-Identify**: These logs are first stripped of four direct identifiers which are replaced by the "Participant ID" attribute[2].

2. **Row Code**: Each row is assigned one of 46 "Row Codes" which captures both the overall success or failure result and detail about the action performed. This row code is the backbone of the encoding system, and will be explained in greater detail in 2.2.3.

[1] The results are robust to changing the definition of abandonment, i.e., the length of the lapse in activity. Periods up to 10 minutes were examined, and later chapters use 10 minutes.

[2] This step is only necessary in a research context where the users remain anonymous to the researcher.

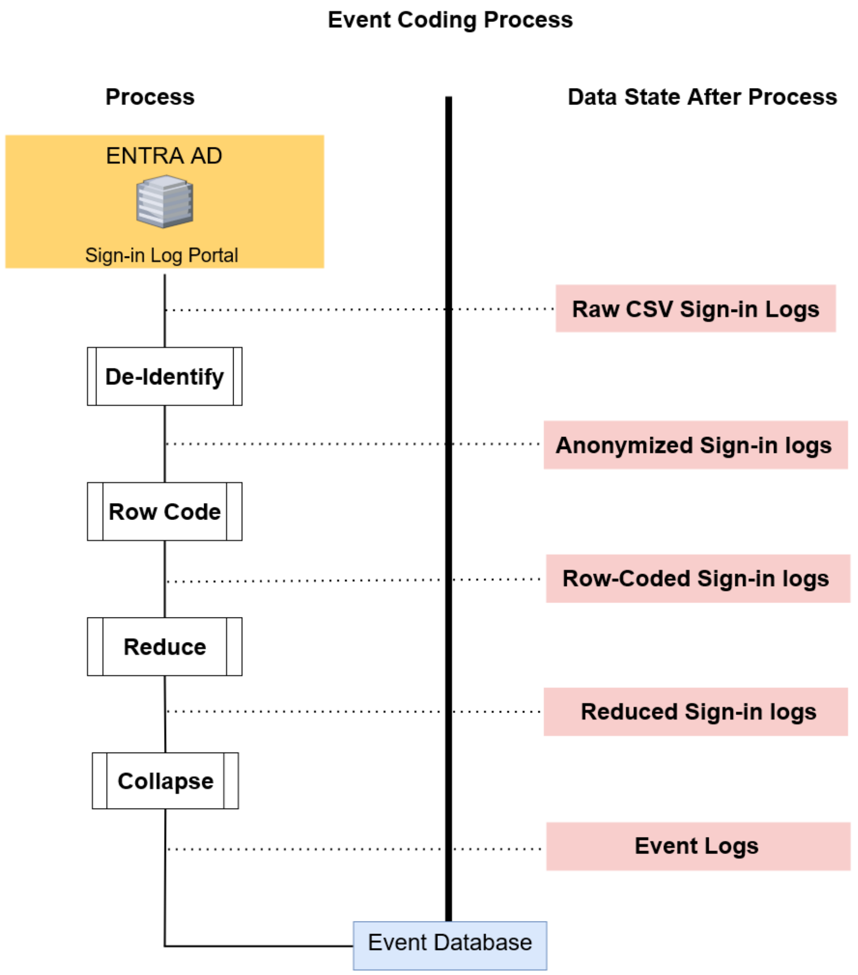


Figure 2.1: Event coding overview diagram.

3. **Reduce**: Several helper attributes are added, such as "event number" to indicate which event a particular authentication entry is associated with. An attribute tracking if a password is entered is added by cross referencing an entry's "RequestID" with its entry(s) in the "authDetails" files. These attributes are used in combination with the row code to produce the "interactive" attribute. Duplicates and known or suspected malicious entries are removed.

4. **Collapse**: Finally, we create a derivative data set by aggregating the key attributes from all rows for a given event, tracking the number and type of errors encountered, the form of MFA used, etc. This yields a smaller data set comprised of rows with

20 attributes, each row describing a complete authentication attempt to a particular application. The final set of attributes is easily adjusted based on the attributes available in the raw data.

2.2.2 Raw Log Data Description

The University of Tulsa utilizes Microsoft Entra AD for authentication. Data is first collected through the Entra AD portal, which presents the sign-in logs broken down into six categories. The first four categories are interactive and non-interactive sign-in logs and their corresponding interactive and non-interactive "auth details" files. The final two categories are called "Application logs" and "MSISignins"; these refer to authentications by service principals, and authentications by an Azure Managed Identity, respectively. Interactive logs are defined by Microsoft as those sign-ins where "a user provides an authentication factor, such as a password, a response through an MFA app, a bio-metric factor, or a QR code".

To investigate user experience, we exclude application and managed identity logs, as they do not reflect human interaction. For example, visibility into the errors that occur prior to or following the presentation of an authentication factor is essential. Thus, both files and their associated authentication details are downloaded. A single log entry comprises 44 attributes and delineates a singular system interaction. A brief period of user interaction can generate numerous log entries per minute, many of which may represent back-end processes that are not directly experienced by users during an authentication attempt. The process described herein is implemented on Entra AD logs, but is designed to be generalized to other sources of authentication logs.

Table 2.1 summarizes the attributes, which we have split into 7 broad categories. *Direct ID* attributes identify the specific user, which, for this work, are immediately removed and replaced with a unique user number. *Device* and *Connection* attributes detail the network connection and device characteristics. *Session Info* attributes comprise the bulk of the data, including the name and ID of the application and resource being used, token infor-

Table 2.1: Raw Azure AD sign-in log attributes

Category	Attributes
Direct ID	User, User ID, Username, Sign-in Identifier
Device	Device ID, **Operating System**, Browser
Connection	IP Address, Location, Latency
Connection	IP (seen by resource)
Session Info	**Date (UTC)**, Application, Application ID
Session Info	Resource, Resource ID, Resource Tenant ID
Session Info	Home Tenant ID, Home Tenant Name, Request ID
Session Info	Correlation ID, Cross Tenant Access Type
Session Info	Incoming Token Type, Unique Token Identifier
Session Info	**Client app**, Client Credential Type
Session Info	Autonomous System Number, Token Issuer Type
Session Info	Incoming Token Type, Token Issuer Name
User Provenance	User Type, Compliant, Managed, Join Type
Authentication Info	Authentication Protocol, Conditional Access
Authentication Results	**Status**, **Sign-in Error Code**, Failure Reason
Authentication Results	**MFA Result**, **MFA Auth Method**, MFA Auth Detail

mation, client application, and so on. Redundant attributes will be dropped in processing, such as alpha-numeric "ID" fields like "Resource ID"; as "Resource" is retained, which is the name of the Resource. The "Request ID" field is always retained, as it is the unique key linking a particular log item with other associated data in the Azure AD system. The *User Provenance* category includes information about a particular user's account, such as their user type (member or guest) and join type (Azure AD Registered, Azure AD Joined, Hybrid Azure AD Joined). The smallest category is *Authentication Info.* Relevant attributes include "Authentication Requirement", which indicates if the authentication requires single or multi-factor authentication, and "Conditional Access", which indicates any conditional access policies that were applied and the result. Finally, the *Result Info* category includes details about the authentication attempt and result.

The "Status" attribute has one of three values: Failure, Interrupted, and Success. Note that many "Failure" results are not caused by improper user action, and "Interrupted" results often do not tangibly disturb the user experience. The "Sign-in error code" attribute contains a numerical error code when an error is present, which is true for any entry that

Table 2.2: Row codes - success (all)

Code	Item	Authentication
0	Token Success	Multi-Factor
1	App Password	Multi-Factor
2	Remembered Device	Multi-Factor
3	Registered Device	Multi-Factor
4	App Notification	Multi-Factor
5	SMS Verification	Multi-Factor
6	Phone Call	Multi-Factor
7	OATH	Multi-Factor
8	Token Authentication	Single-Factor
9	Password Authentication	Single-Factor
10	Password Authentication	Multi-Factor

Table 2.3: Row codes - errors (sample)

Code	Item	Type
9	Token Failure	Interrupt
10	Needs to Complete MFA	Interrupt
16	Device Code Expired	Interrupt
14	User has no Role in Application	Configuration
19	Error Issuing Token	Configuration
27	Failed to Complete MFA	User
30	Limit on MFA Calls	User
34	Blocked for Malicious IP	Hacker
45	Uncategorized Single Factor Error	Unknown
46	Uncategorized Multi-Factor Error	Unknown

is not labeled “Sucesss”. This error code is the key attribute used to assign row codes for non-pass rows. The “Failure reason” attribute contains a description of the error code result when an error is present, and detailed descriptions of errors and remediation are available from Microsoft on their website [29]. There are three MFA-related fields: “MFA result” provides a text description of the authentication result; “MFA auth method” contains the type of MFA used when applicable, and “MFA auth detail”, which may contain a phone number associated with the MFA with the last two digits revealed. The last field is a Boolean “Flagged for review”, which is only true when an admin flags a user account.

2.2.3 Row Coding

Adding a row code enables us to distill the 44 attributes included in raw log instances to a minimal expression. Thus, a set of 46 *row codes* were created to capture critical information about an authentication attempt's result. There are two broad results that a single entry can indicate: Pass (Success), or Fail, indicated by the attempt concluding in an entry marked "Failure" or "Interrupted" in the "Result" field of raw sign-in logs.

Nine categories of logs were identified that indicate successful authentication, as presented in Table 2.2. These 9 categories are variations of 3 basic results: Token Successes, Remembered Device Successes, and MFA Successes. Token Successes are split between single and multi-factor authentications, and all multi-factor authentications that are not token-related are either a primary form of MFA such as Text message, OATH, etc. or fulfilled through remembered device. Six row codes capture the various forms of MFA Successes, and two capture the remaining single factor successes.

The remaining row codes are utilized for entries that do not indicate a successful authentication pass. We categorize these 38 row codes into four principal types of errors: Interrupts, User Errors, Configuration Errors, and Hacking Errors.

Interrupts occur when the "Failure" (or Interruption) reported is not an actual failure; rather, it is a redirection or part of the intended authentication flow. In our user-centric paradigm, this indicates that the user is not confronted with an error message; they do not perceive a failure. One specific instance is Row Code #9: Token Failure. This scenario is not an error in the sense that the user or application encountered an issue; instead, it represents an expected occurrence within a token's life-cycle. When this Token Failure error arises, a user has entered their password, and their device asserts a token that would otherwise meet the second factor requirement. However, that token is deemed invalid for various reasons. The user experiences this situation as being redirected to their multifactor authentication (MFA) prompt screen following the input of their password. This represents a typical use case and is not perceived as a failure or an additional delay. "Interrupts" do not detract from the typical user experience.

The key difference between user and configuration errors is the agency of the user to resolve the error. 8 row codes are used for the user errors. For example, row code #27 indicates a user initiated a multi-factor sign in but never provided the second factor, and row code #26 indicates a user input an incorrect password. An additional 8 row codes are used for configuration errors, which includes transient errors. Row Code #18 is a good example, wherein a user tries to authenticate to an application, but is denied because their account has no associated role in the application. The error message presented indicates that an administrator must give the user access, it can not be dynamically requested, making this an error outside direct control of the user. Finally, we have codes that capture behavior identified by Azure AD as malicious, and a catchall for uncategorized errors.

We now describe the process of creating these row codes, beginning with non-pass entries. Three of the co-authors, two with high domain knowledge and one with low domain knowledge, independently inspected log samples encompassing each unique "Sign-in Error Code" present in the dataset. Co-authors labeled each error code with one of four categories: Interrupt, User Error, Configuration Error, or Hacking Error. Each error was considered alongside all available documentation and examples of the error appearing in the data. Krippendorff's alpha was 0.73 considering all three raters, and 0.86 for the two raters with high domain knowledge. Majority opinion was sufficient for all but one of 127 unique error codes labeled, and each labeling was reviewed and confirmed by the authors. Labeled errors were then grouped into row codes by similar themes within each category of error. These processes yielded the final set of 36 error groupings, which were then given integer representations beginning after the 10 "Pass" row codes. Labeled errors were then grouped into row codes by similar themes within each category of error.

1. **Row Code 11**: There are 2 error codes that indicate MFA Completion is required. They redirect the user to use their second factor for the authentication, "Sign-in Error

Table 2.4: Event log attributes.

Attribute	Category	Comments
Direct ID	User	Participant ID
Device	OS	String
Device	Browser	String
Connection	IP Address	Alpha-numeric
Session Info	Event#	Int
Session Info	Application	String
Session Info	Service	String
Session Info	ClientApp	String
Session Info	Start	DateTime
Session Info	End	DateTime
Auth Info	MFA Type	String
Auth Info	AuthReq.	Single/Multi-Factor
Result Info	Result	Success/Failure
Result Info	Detail	Result Details
Result Info	Password Entries	Int
Result Info	Elapsed	Elapsed Time in Seconds
Result Info	TA	Time Away in Minutes
Result Info	UEs	User Errors Count
Result Info	IEs	Int. Errors Count
Result Info	CEs	Config Errors Count
Result Info	Error Codes	Int List of Errors

Code" 50074 and 50076.

2. **Row Code 18**: error codes 50105 and 50177 both describe a user who has not been granted specific access to an application, and is classified as a configuration error. This is distinct from a user who is dynamically requesting access to an application, which is classified as an interrupt, as it is an intended step in the authentication cycle, not the result of incorrect permissions or any failure.

2.2.4 Reduction

There are four steps taken to reduce the authentication logs after row coding. Here, we note that the focus of this paper and the authors' related research has been on measuring and characterizing legitimate use. As such, we discard known and suspected malicious authentication attempts when constructing events. First, we discard logs from non-standard user agents, including POP and IMAP, as well as logs categorized as "Hacking Errors", such as those with row code #34: "Blocked for Malicious IP", since these attempts are unlikely to originate from legitimate users interacting with our applications.[3] Second, we exclude logs from authentication attempts made to "API" resources, which do not represent interactive user authentication. These are authentications conducted by client-side applications to access third-party resources.[4] Third, we discard duplicate logs, defined as logs with identical attributes occurring within one second of each other, retaining only the later of the duplicates. Finally, we also discard any logs whose row codes are not labeled as interactive, which is a sub-attribute of our row codes.

2.2.5 Collapse into Events

Returning to our definition, we define an *event* as:

> The events captured in log data that are directly experienced by a user commence with the initiation of authentication to a specific application and conclude with the eventual success, failure, or abandonment.

Each event captures the number of errors encountered prior to achieving either success or failure, alongside the types of errors involved, the time allocated for an attempted authentication, and the authentication methods utilized. Given that these characteristics are encapsulated in the row codes specified above, monitoring their occurrence across events is a straightforward process.

After the raw data has been de-identifed, row coded, and reduced to interactive, non-duplicate entries, events are constructed by aggregating rows with the same "Event Number". This number is created by first sorting entries by user and datetime, and setting a boolean "New Event" to TRUE if the gap between the current entry and prior entry exceeds 90 seconds for a given user. A cumulative sum is run on the "New Event" attribute to assign

[3] These "Hacking Errors" and non standard user agent authentication logs were not discarded when conducting security-focused analysis in later chapters

[4] These non-user API authentications are retained and marked as non-interactive in the SOC dashboard in Chapter 5

an event number to each log. In an enterprise environment without SSO implementation, a second condition is introduced: the successful completion of an authentication. In our SSO environment, once an authentication succeeds, any subsequent authentications to related sites will be non-interactive and fulfilled by the token presented by the user, resulting in no authentication interaction.

By defining events in this manner, we maintain the flexibility necessary to accommodate scenarios in which the user initiates multiple applications concurrently.

For example, a user might first be prompted for MFA on their desktop Outlook client. If that fails, a user could authenticate using a web-based interface instead. For our purposes, this is treated as a single event when occurring in close temporal proximity, which is effective for our enterprise environment in which there are many different applications which can be satisfied by completing authentication in any one service. The resulting event provides a clear indication of overall success, the application used, MFA Type, time spent, count and classification of errors, and provides the error codes associated with the errors to enable user and population metrics.

2.2.6 Event Examples

Events are comprised of the 21 attributes listed in Table 2.4.[5] The first attributes tell us who authenticated, the system they used to do so, and total time elapsed. We also retain authentication information (MFA type and whether one or two factors were required). The final 9 attributes capture relevant details about the authentication experience by aggregating the observed row codes for log entries in the event. Note that a user can experience one or more errors, from mis-configurations to failed passwords or MFA prompts, before ultimately succeeding in the authentication. Such impediments are reflected in the other fields, such as the “Password Entries” attribute that tracks the number of times the user input their password during the authentication event. The “Elapsed” attribute is calculated by the

[5] Retained attributes are easily adjusted to suite a given organizations system or topic of research

Table 2.5: Sample authentication events.

Ev #	Result	MFA Type	PW #	Time (s)	TTR (min)	OS	Application	UE #	CE #
3	Failure		0	2	4	Windows	Office 365	0	1
4	Success	App	2	16	NA	Windows 10	Azure Portal	1	0
6	Success	App	1	0	NA	Windows 10	Azure Portal	0	0
12	Failure		1	2	4	Windows	Office 365	0	1
13	Success	App	1	0	NA	Windows 10	Teams	0	0

difference between the first and last rows in a sequence that collapses into an event. Because there is no indicator in the raw sign-in logs when a Multi-factor prompt is initiated, this measure captures the extra time spent due to errors and interruptions in the authentication process. [6] Time Away (TA) measures the gap in time between a failed authentication event and the next attempted login.[7] The final attributes tally the number of User, Interrupt, and Configuration Errors experienced during the authentication event.

Table 2.5 illustrates the "event" log with example events. Event #3 shows a simple failure with a single "Configuration Error" (CE). A "Time Away" of 4 minutes is listed, indicating that 4 minutes elapsed before the next successful authentication, event #4. We interpret event #4 is as follows: App-based MFA was used to successfully sign into the Azure Portal on a Windows device after a single "User Error" (UE), an invalid password entry. The authentication process took 16 seconds after initiation, significantly longer than that observed by [37], which is likely a consequence of the failed password. Event #6 offers another example of a simple success with no errors that takes 0 seconds after initiation to complete. This zero second time reflects the complete lack of friction in the event, as we do not know when the user started to input their password, use MFA, etc; we only know when the user hit ENTER and attempted the authentication. By breaking down authentication

[6] This limitation is overcome when using Graph API data in Chapter 5. Elapsed Time in Chapter 5 captures the moment the first factor is entered, in addition to the moment the second factor is initiatied or completed.

[7] TA is similar to "recovery time" reported by [39], which captures the time between a failure and the next success.

logs into discrete user-centric events, we can provide meaningful insights into user experience and application health, as we demonstrate next.

2.3 Discussion

In this chapter, we presented a systematic methodology for transforming raw authentication logs into user-centric event logs. This transformation was achieved by introducing a set of row codes that capture both successful authentications and a spectrum of error types, which includes user errors, configuration issues, instances of malicious or hacking attempts, and intermediate interruptions. These row codes afford a concise yet comprehensive summary of the underlying raw log entries.

We further introduced an event-based approach, wherein logically grouped sequences of user actions are collapsed into single records, each reflecting a complete authentication attempt. This restructuring yields a simpler, smaller dataset with attributes that capture key aspects of each sign-in experience, including the number and type of errors, elapsed time, and the authentication methods used. By focusing on user-driven activity rather than system-oriented log entries, we gain a more interpretable view of authentication behaviors, allowing for straightforward aggregation and analysis of metrics such as user error rates and service configuration stability.

Taken together, these steps constitute an extensible framework that can be adapted to various authentication environments. By emphasizing interpretability, we aim to equip analysts, security practitioners, and researchers with enhanced insights into real-world authentication processes. This focus ultimately facilitates more effective security monitoring and improvements in user experience. Now that our core methodology has been presented, we proceed to use this process as a tool for analyzing multiple aspects of authentication log utility. We begin with an economic analysis of the user burden or “costs” associated with a change to MFA configuration that offered higher security while introducing a more cumbersome user experience.

CHAPTER 3
USER COSTS OF ENTERPRISE MULTIFACTOR AUTHENTICATION POLICIES

3.1 Introduction

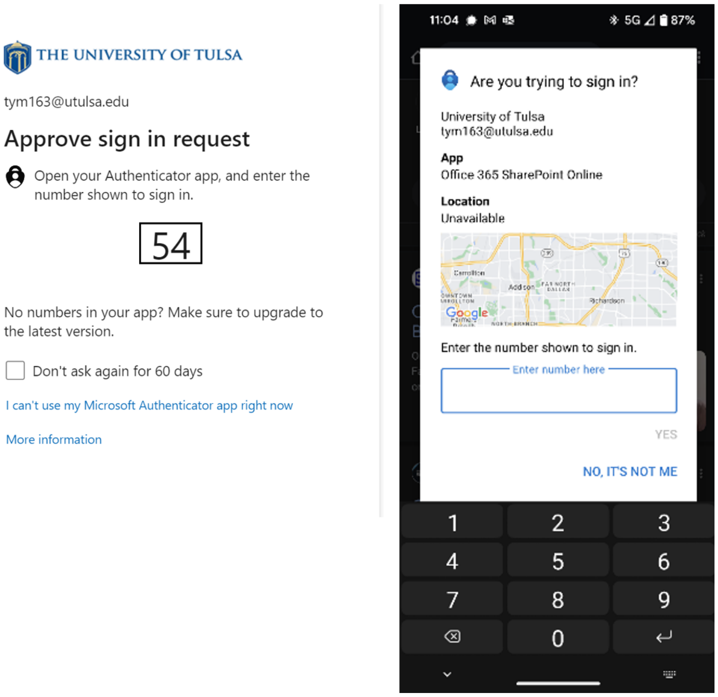
In response to growing threats and increased regulatory pressures, organizations have sought to strengthen their cybersecurity posture. They are allocating more resources towards cybersecurity initiatives, and adopting new security controls to mitigate elevated risks. Such investments have undoubtedly brought benefits in terms of reduced exposure to attacks. However, increased security can also introduce opportunity costs. Some legitimate tasks may now be blocked, from emails mistakenly caught in a spam filter to accounts being locked out following mandatory password changes. Additionally, even when working properly, security controls introduce friction that can slow task performance and frustrate users. Such opportunity costs are often overlooked, but they are critically important because they may add up to substantial losses and can even alter behavior to be less productive or secure.

As organizations seek to strengthen their cybersecurity posture, changes often come first to how authentication works. The Cybersecurity and Infrastructure Security Agency (CISA) recommends four critical steps individuals and organizations can take to strengthen security [13]. The first item of the four is to “turn on multifactor authentication” (MFA). For individuals, the process can be as simple as tweaking a configuration setting. For firms, the process can be a bit more involved, as it requires changes to how enterprise IT infrastructure is configured and operated. Nonetheless, organizations are increasingly supporting MFA. Most often, they are actually mandating its use throughout the enterprise [1].

MFA provides an excellent opportunity to study the opportunity costs of cybersecurity controls. That is because authentication affects everyone and is highly visible to users. Moreover, MFA significantly alters the steps users must take to use an enterprise IT system. When MFA works well, it can be seamless. Enrolled users provide a second factor (often a mobile device) and carry on with their tasks as before. However, when users fail to authenticate, they cannot complete their intended task. This can happen because they forgot their second factor, got a new phone, or for a variety of other reasons. Correcting the problem can be time consuming and costly, often requiring manual assistance from IT staff.

While we fully expect that the benefits of MFA to outweigh the costs, the burden imposed is often not explicitly accounted for. In this paper, we empirically analyze the opportunity costs of MFA in a deployed setting. Once opportunity costs are identified, it becomes possible to take steps that minimize them. As we will show, choices in how the technology is deployed can greatly impact how users respond and the resulting magnitude of the costs imposed.

Increasingly, the technologies deployed by enterprises generate large amounts of "data exhaust" that could be mined for insights into user behavior [19]. We leverage a very large dataset of Microsoft Azure Active Directory sign-in logs (now known as Microsoft Entra ID) from a University between 2021–2023. Using these data, we examine the opportunity costs associated with the adoption of a more onerous multifactor authentication process. Critically, at the end of the 2021–2022 academic year, the University changed the MFA procedure for mobile use with the authenticator app from a deny/approve "push" notification to a more cumbersome two-digit code which needs to be entered into the authenticator app when prompted on the login screen. This was especially cumbersome for users using Mobile MFA who attempted to login from a mobile device. This is because both the authentication app and the login window had to be open at the same time and users had to switch between them. Figure 3.1 provides screenshots. This exogenous change allows us to examine the added costs associated with a more secure mobile MFA method. In the case of text messages, no change was made.



THE UNIVERSITY OF TULSA

tym163@utulsa.edu

Approve sign in request

Open your Authenticator app, and enter the number shown to sign in.

54

No numbers in your app? Make sure to upgrade to the latest version.

□ Don't ask again for 60 days

I can't use my Microsoft Authenticator app right now

More information

11:04 5G 87%

Are you trying to sign in?

University of Tulsa
tym163@utulsa.edu

App
Office 365 SharePoint Online

Location
Unavailable

Enter the number shown to sign in.

Enter number here

Figure 3.1: Changed MFA procedure. Interstitial prompt with two-digit number (left) and phone-based application for entering the code (right).

We focus on two measures that serve as proxies for increased opportunity costs associated with the change in MFA policy. (1) The first measure is the number of login failures users experience. (2) The university employs a single-sign-on system and tracks all authentication attempts to any university service. Hence, the second measure we employ is how long a user remains without access to IT resources following a failure. In particular, we measure the *time away* following a failed login until the user attempts to login again. If users become frustrated, distracted or disengaged after failing to authenticate, then they may take longer to reengage. Hence, both failed logins and time away are promising measures of the opportunity cost from onerous security measures.

We first report descriptive statistics. This section clearly shows that there were significant increases in the number of log-in failures and in time spent away following failures when using mobile MFA following the exogenous change. We then employ "fixed effects" econometric models to analyze how these costs changed over time.[1] The econometric results confirm the descriptive data "results" and provide us with estimates of the effect of changes in the MFA procedure on the number of failures and time away. Although we have very limited data on user characteristics, we do know the time of day for each attempted login. We find that users who were primarily active from 8:00 am to 5:00 pm during the week had the greatest difficulty adjusting to the new mobile MFA procedure. These users likely contain more staff members than faculty or students.

The chapter is organized as follows. Section 3.3 describes our derived event logs, defining the most relevant attributes. Section 3.3 discusses our data and provides descriptive statistics. In section 3.4.1, we conduct the econometric analysis and provide our results. Section 3.5 discusses the analysis.

3.2 Methodology for Constructing Authentication Events

Whereas most prior work studying authentication usage has surveyed users about their experience, we seek to go straight to the source: authentication logs. Through a partnership with the IT department at the authors' university we obtained access to anonymized Entra ID authentication logs for analysis, approved by the Institutional Review Board (IRB) under protocol 24-02.

Interpretable user experience is buried in raw security logs, with combinations of values for different attributes indicating meaningful states. A sign-in log entry contains around 36 attributes representing a single system interaction. Hence, a small period of user interaction can generate many log entries, sometimes dozens per minute. Critically, many of these entries represent back-end processes that users do not directly experience. By inspecting these logs carefully, we constructed a set of 38 *row codes* that capture critical information about an authentication attempt, and are used to characterize an attempted individual login. This allows us to discard irrelevant entries and consolidate significant

[1] These models explicitly take into account that there are repeated observations on users. This enables us to examine how user costs increased from enhanced security changes to how MFA was deployed.

interactions as we construct events. Recalling the methodology from chapter 2, we define an *event* as follows:

> The occurrences reflected in log data that are directly experienced by a user, beginning when an authentication to a particular application is initiated, and terminated upon the eventual success or failure of the authentication attempt.[2]

Each event captures the number of errors encountered before eventual success or failure, as well as the type of errors involved, the type of authentication used, as well as whether the attempted login was from a desktop/laptop or a mobile device

During a login attempt, a user can experience one or more errors, from misconfigurations to failed passwords or MFA prompts, before ultimately succeeding in the authentication. Errors are assigned to three primary categories: User and Configuration Errors which are split by attribution, and Interrupts. User Errors are those error codes generated by invalid or missing user input, such as failure to answer an MFA prompt or incorrect password entry. Configuration Errors encompass errors that are not due to user error, such as developer errors or issues with the user's account status. Interrupts occur when the system needs to take further action during an authentication flow, such as when the token presented has expired, and the user must be redirected to use their second factor. These "Interrupt Errors" do not indicate adverse events or impediment to normal usage flows, and instead serve as flags for various operations. We also track the number of times the user input their password during the authentication event. Time Away measures the gap in time between a failed authentication to a service and the next attempted login[3].

[2] If there is a lapse of activity great than 90 seconds, we also define this as a failure. The results are robust to changing the length of the lapse in activity.

[3] Time away is similar to "recovery time" reported by [39], except our measure does not discriminate between successful and failed follow ups; it simply captures the gap between interactions after a failed login.

3.3 Data and Descriptive Statistics

3.3.1 Time Periods for Analysis

Our data is from November 15 2021 to May 31 2023. We divided the data into four periods:

- Academic year 2021–22: from (November 15 2021 — May 31 2022)
- Summer 2022 (June 1 2022 — August 15 2022)
- Early Academic year 2022–23: (August 16 2022 — November 14 2022)
- Academic year 2022–23: from (November 15 2022 — May 31 2023)

In the analysis, we employ data from the two "partial" academic years covering the November 15 to May 31 period to keep the dates consistent across samples. Our results are qualitatively unchanged if we include the data from August 16 2022 — November 14 2023 in the 2022-2023 academic year.

We examine what happened to the number of login failures and "Time Away" (TA) following the 2021–2022 academic year. This made authentication more secure, but with a "cost" in that authentication became more complicated. From our standpoint, this yields a natural experiment and enables us to compare the before and after periods and the effect of an (exogenous) increase in mobile MFA authentication procedures on Time Away and the number of failures.

We are particularly interested in how this change affected time away and the failure rate. The explanatory variables (factors) we employ in the analysis are discussed when we present our models.

3.3.2 Descriptive Statistics

The first step when analyzing a large data set is to cut the data in many ways and look for patterns. When we examined the data by academic year at the event level, we were struck by the significant increase in TA and the number of failures during the 2022-23 academic year for attempted logins using mobile MFA relative to the 2021-2022 academic year.

Following a discussion with the University IT department, we learned that following the 2021-22 academic year, the mobile MFA authentication process was changed. It was changed from a (1) push notification, where users simply had to approve or deny that they were trying to login to a (2) two-digit approval system requiring the user to enter a number shown in the login process into a mobile authenticator. The effect of this change is well illustrated by the descriptive statistics for failures and time way at the event level. Below we show comparisons on these measures when(i) mobile MFA was employed and (ii) when Text MFA was employed.

In Table 3.1, we report the descriptive data at the event level when mobile or text MFA is used.

Descriptive statistics at the event level for the mobile MFA login procedure show a very significant absolute and percentage increase in mean TA from approximately 35 minutes per event in the 2021-22 academic year to approximately 81 minutes per event in the 2022-23 academic year. More importantly, the table shows that the 90^{th} percentile of the distribution of TA increased dramatically from 0 minutes in the 2021-22 academic year to approximately 170 minutes per event in the 2022-23 academic year. Thus, a non-trivial percent of users have struggled with the enhanced MFA procedure for Mobile MFA.[4]

Table 3.1 shows that the failure rate (the percent of times an authentication attempt was not successful) increased significantly in the second academic year when the mobile MFA procedure changed: The failure rate with mobile MFA rose from 10.2 percent in the 2021-22 academic year to 17.9 percent in the 2022-23 academic year. This is a very large absolute increase.

[4] The increased mean authentication delay (denoted elapsed), on the other hand, is virtually unchanged: From approximately 3.5 seconds per event in the 2021-22 academic year to approximately 4.2 seconds per event in the 2022-23 academic year. Hence, we do not focus on this variable as we noted in the introduction.

	21-22 academic year		22-23 academic year	
	Mean		Mean	
Time Away (minutes) - Mobile MFA	34.9	0	81.3	170.0
Time Away (minutes) - Text MFA	10.3	0	24.3	0
Failure Rate (Mobile MFA)	10.2%		17.9%	
Failure Rate (Text MFA)	2.6%		4.8%	

Table 3.1: Descriptive statistics: event level data

Table 3.1 also shows that MFA using text messages is much less problematic for users and there was a much smaller change from the 2021-22 Academic year to the 2022-23 Academic year. The mean time away was approximately 10 minutes when using Text MFA login procedure in the 2021-22 academic year and approximately 24 minutes per event in the 2022-23 academic year.

Importantly, the 90^{th} percentile of the distribution of time away for text messages was zero in both the 2021-22 academic year and the 2022-23 academic year. Additionally, the differences between these two methods in mean time away was 25 minutes (35-10) in the first academic year and 57 minutes in the second academic year.

While the failure rate was higher for Text MFA in the second period (4.8% in the second period vs. 2.6% in the first period), it was much lower than when Mobile MFA was used. Further, the differences between these two methods in the failure rate (by academic year) was 7.6% (10.2-2.6) in the first academic year and 13.1% (17.9.-4.8) in the second academic year. Hence, the difference nearly doubled in the second year.[5]

3.4 Econometric Analysis

We now turn to the formal analysis, in which we use (i) Time Away and (ii) log-in failures as the dependent (or response) variables. To ensure that our results are not due to new users, as discussed, we only include users that were active in both academic years.

[5] Once we control (in the regressions) for whether the login attempt was from a mobile or desktop/laptop device, there is virtually no change in the failure rate between the periods when using Text MFA.

Since there is little change in faculty and staff users from year to year and since most undergraduate students are at University for four years, most of the users (around 90%) are repeat users.

3.4.1 Fixed Effect Models

We have panel data, that is, repeated observations on each individual. Having a panel rather than cross-sectional data (one data point on each individual) is advantageous, since a cross-section cannot control for time-invariant individual characteristics, like user attitudes towards risk. Such unobservable factors are included in the error term in cross-sectional analysis. If these unobserved effects are correlated with the right-hand-side variables of the estimation equation, the estimates from the cross-sectional analysis will be biased. However, we eliminate this problem by using fixed effect models. We now describe the fixed effect model.

The equation we start with is the following:

$$Y_{it} = \alpha_i + X_{it}\beta + \delta_t + \epsilon_{it}. \tag{3.1}$$

The dependent variable Y_{it} is (say) the sum of TA for user i at time t, where time is at the aggregated weekly level.[6]

The explanatory variables in X_{it} are observable time-varying factors that likely affect Time Away and β are coefficients to be estimated. The vector $\alpha_i = \alpha + A_i\eta$ is such that α is a constant and A_i is the vector of unobserved time-invariant user characteristics. An example is user attitudes towards risk. The key is that the user characteristics in the vector A_i is do not change over time. As we show below, we do not need to know the value of these characteristics in order to estimate the model. δ_t is the week effect. Finally, ϵ_{it} is an error term.

The following equation expresses the mean values at the level of the user, where the

[6] In fixed-effect analysis, the data must be in time periods (say a day or week).

mean is computed over time from equation (3.1).

$$\bar{Y}_i = \alpha_i + \bar{X}_i\beta + \bar{\delta} + \bar{\epsilon}_i \tag{3.2}$$

Subtracting (2) from (1) yields:

$$Y_{it} - \bar{Y}_i = (X_{it} - \bar{X}_i)\beta + (\delta_t - \bar{\delta}) + (\epsilon_{it} - \bar{\epsilon}_i) \tag{3.3}$$

Since the vector $\alpha_i = \alpha + A_i\eta$ does not depend on time, it drops out in equation (3) which are the deviations from the mean. Equation (3) is the fixed effects model we will estimate.[7]

We employ a variable (denoted "Post") in X_{it} that takes on the value zero if the data are in the first academic year and one if the data are in the second academic year. We interact "Post" with all of the other explanatory variables. In this way, we analyze both years together, which is preferred to estimating both years separately, since we can easily see the differences between the first and second year. Our results are robust to running separate regressions for each year.

3.4.2 Variables in the Analysis

The variables we employ (and their definitions) in the analysis (at the weekly level) are as follows:

- Dependent Variables:
  - Time Away (TA): The sum of the Time Away in minutes for that user during the period, which is a week in our analysis.[8]

[7] See Angrist (2009) for more a detailed discussion of fixed effects models [5].

[8] When calculating the mean TA for descriptive statistics, we limited TA to 1000 minutes. We do this so not to "distort" the means, as several values reach 14,000 minutes. In the regressions, we do not restrict TA. Because we have so many observations, and because we are running a log/log model, nothing in the results changes if we restrict Time Away to 1000 minutes in the econometric analysis.

  - The sum of the number of failures for that user during the week.

Independent Variables:

- IEs: The number of Interrupt Errors during the period.
- CEs: The number of Configuration Errors during the period.[9]
- Text-MFA: The Number of Logins when a text message MFA procedure was used during the period.
- Mobile-MFA: The Number of Logins when a mobile app MFA procedure used during the period.
- Pw-uses: The number of Password Entries (whether correct or incorrect) during the period.
- Mobile entries is the number of attempted logins from a Mobile device.[10]
- Period: The week number
- Post is a binary variable that takes on the value zero if the data are in the first academic year and one if the data are in the second academic year. We interact Post with all of the independent variables.

We are mainly interested in how (i) the different MFA uses (Text message, Mobile app) and (ii) whether the user attempted to login from a mobile or desktop device affected (I) the number of failures and (II) "Time Away", the time in between a failed authentication attempt and the next attempt to authenticate. The other variables are primarily controls.

Overall in both academic years, 13 percent of attempted logins used text message MFA procedures, while 17 percent of attempted logins used mobile MFA procedures. The

[9] Nothing changes in the analysis if we combine the interrupt and configuration errors into one variable of "non-user" errors.

[10] This is regardless of whether text MFA or Mobile MFA was employed.

remainder of the attempted logins were primarily from a Remembered device. In many cases, when using a remembered device,the user did not have to use MFA.[11] The breakdown among these categories did not change from year to year.

The formal analysis is at the weekly level. The dependent variables are failures and TA, which is defined as the time in minutes between a failed login and the subsequent attempt to login for that user for each event in the week. We add the TA and number of failures as well as all independent variables for each event to get the totals at the weekly level for each user.

We employ a log/log functional form (which employs the natural logarithm (ln) of each variable). This functional form typically gives better results in terms of the explanatory power of the model when the variables employed have skewed distributions. This is true in our case as well, since the raw data is quite skewed.

The overall R-squared, which measures the explanatory power of the model (and ranges from 0 to 1) is 0.504 for the log/log model with Time Away as the dependent variable and 0.509 when using the number of failures as the dependent variable. [12]

In (natural) logarithm form, the variables are as follows:

- ln-TA is the natural logarithm of the sum of ”Time Away” (ln(TA + .001)[13]
- In-Failures = ln(Failures + .001)
- ln-Ies = ln(Ies + .001)
- ln-Ces = ln(Ces + .001)
- ln-Rem-Device = ln(Rem-Device +.001)

[11] In these case there are virtually no login failures. Less than two percent of total attempted logins used either Phone Call or OATH MFA procedures.)

[12] Unsurprisingly, the overall R-squared is much lower when estimating linear/linear models. This was what we expected given the skewed distribution of the data.

[13] Since these variables can take on the value zero, we add a very small number (.001) in order to create the logarithms.Nothing changes if we add a slightly larger value than 0.001.

- ln-Text-MFA = ln(Text MFA + .001)

- ln-Mobile-MFA = ln(Mobile MFA + .001)

- ln-Pwuses = ln(Pw-uses + .001)

- ln-Mobile-entries = ln(Mobile-entries + .001)

3.4.3 Regression Results: Time Away as the Dependent Variable

The results in the first column of Table A.1 (in the appendix) show that in the case of Time Away, other things being equal, the estimated coefficient associated with the number of mobile MFA uses per week is positive (0.16). The Table shows that this result is statistically significant at the 99 percent level of confidence for the first academic year. That is, more mobile MFA login attempts per week, other things being equal, leads to significantly more Time Away in that week.

Strikingly, in the case of the 2022-23 academic year, the estimated coefficient associated with the number of mobile MFA uses is (0.25=0.16+0.09). The difference in the coefficient estimates between the two years (0.09) is statistically and economically significant as shown in the Table.

Since we are estimating a log-log model, this means that a 100 percent increase in the number of Mobile MFA uses leads to a 25 percent increase in Time Away in the 2022-23 Academic year vs. 16 percent in the 2021-2022 academic year. This means that, conditional on the same number of mobile MFA uses, there is significantly more Time Away in 2022-23 than in 2021-2022. Thus, controlling for other factors, the change in mobile MFA policy (which made it more secure) greatly increases the weekly Time Away when Mobile MFA is used, relative to the effect in the academic year 2021-22.

Importantly, the results show that, other things being equal, the estimated coefficient associated with the number of text messages MFA uses is much smaller in both academic years (0.027 in year one and 0.042 in year two). Other things being equal, there is a very small change in Time Away in the second period (relatively to the first period) when text MFA is employed.

In the case of attempting to login in from a mobile device (whether it is using text MFA or Mobile MFA), the number of attempted logins from a mobile device had virtually no effect on Time Away in the first period. (The estimated coefficient is 0.005.) However, this coefficient is much larger (0.094=0.005+0.089) in the second period reflecting the fact that logins from a mobile device became more cumbersome in the second period.

3.4.4 Regression Results: Number of Failures as the Dependent Variable

In the case of the number of failures as the dependent variable, the results are qualitatively the same. The results in the second column of Table A.1 show that in the case when the dependent variable is the number of failures, other things being equal, the estimated coefficient associated with the number of mobile MFA uses per week is positive (0.10) and is statistically significant for the first academic year. That is, more mobile MFA login attempts per week, other things being equal, leads to significantly more Time Away in that week. In the case of the 2022-23 academic year, the estimated coefficient associated with the number of mobile MFA uses (0.14=0.10+0.04) is 40 percent larger than the coefficient associated for the 2021-2022 academic year. Again, the difference in the coefficient estimates between the two years is both statistically and economically significant. Thus, controlling for other factors, the change in mobile MFA policy (which made it more secure) greatly increases the number of failures when Mobile MFA is used, relative to the effect in the academic year 2021-22.

Similarly to the case when time away is the dependent variable, the results show that, other things being equal, the estimated coefficient associated with the number of text message MFA uses is much smaller in both academic years (0.022 in year one and 0.027 in year two) and that there is virtually no change in the second year.

In the case of attempting to login in from a mobile device (whether it is using text

MFA or Mobile MFA), the number of attempted logins from a mobile device had virtually no effect on the number of failures in year one. (The estimated coefficient is 0.006.) The estimated coefficient associated with the number of attempted logins from a mobile device is much larger (0.051=0.006+0.045) in the second period. This again reflects the fact that logins from a mobile device became more cumbersome in the second period.

3.4.5 Different Types of Users

In this section we examine how different types of users were affected by the change in Mobile MFA policy. We do not know the identity of the users and do not know if they are faculty, staff or students. However, we can proxy for these groups. It is probably likely that many of the University staff primarily use the online system during work hours, which we defined to be 8:00am - 5:00m pm. Hence, we divided the users as follows:

- Group 1 - less than 1/3 of their logins in 2021-22 academic year occurred during “work hours”.
- Group 2 - Between 1/3 and 2/3 of their logins in 2021-22 academic year occurred during “work hours”.
- Group 3 - More than 2/3 of their logins in 2021-22 academic year occurred during “work hours”.

It is likely that Group 3 consists includes much of the University staff, while Group 1 has a greater percentage of students and faculty members.

In the case of Time Away, other things being equal, the estimated coefficient associated with the number of mobile MFA uses per week is 0.14 for Group 1 and 0.19 for Group 3 in the first academic year. In the case of the 2022-23 academic year, the estimated coefficient associated with the number of mobile MFA uses is 0.22. (0.22=0.14+0.08) In the case of Group 3, the estimated coefficient associated with the number of mobile MFA uses in the second academic year is 0.30 (0.30=0.19+0.11). Thus the difference between the groups essentially nearly doubles in the second academic year from 0.05 (0.19-0.14) to 0.08 (0.30-0.22). Group 3 users had much greater difficulty adjusting to the new mobile MFA policy.

In the case of the Number of failures, other things being equal, the estimated coefficient associated with the number of mobile MFA uses per week is 0.09 for Group 1 and 0.12 for Group 3 in the first academic year. In the case of the 2022-23 academic year, the estimated coefficient associated with the number of mobile MFA uses is 0.12. (0.12=0.09+0.03) In the case of Group 3, however, the estimated coefficient associated with the number of mobile MFA uses in the second academic year is 0.17 (0.17=0.12 +0.05). Thus the difference (0.03 vs. 0.05) nearly doubles in the second academic year. See Table A.2. Again, this shows that group 3 users had much greater difficulty adjusting to the new mobile MFA policy.

3.5 Discussion

Multifactor authentication is widely touted as one of the most important security controls organizations can deploy to improve cybersecurity. While the benefits of MFA are well understood, the burdens they impose are not. We seek to fix this discrepancy by providing a method for measuring MFA performance and outcomes.

Using a large dataset gathered from a University with mandatory multifactor authentication requirements, we studied login failures the time users spend away from IT systems and services following a failed authentication attempt. In particular, we investigated the impact of a change in policy to a more secure and onerous configuration requiring users to input codes to app-based (Mobile) MFA. We find that the number of login failures and time away increases substantially for Mobile MFA following this policy change. This suggests that the opportunity costs imposed by the more secure configuration are high for a non-trivial number of users. This is especially the case when the attempted login is from a mobile device.

Significantly, the increased opportunity costs from the more secure MFA configuration were statistically significant. Thus, when contemplating new policies and procedures, the impact on user outcomes and performance should be assessed. Using this methodology, users with high degree of difficulty authenticating can be identified easily for personalized interventions, reducing the overall burden on an IT team, while increasing user productivity.

CHAPTER 4
PSYCHOMETRIC CONSTRUCTS AND USER MULTI-FACTOR AUTHENTICATION

4.1 Introduction

A growing body of research has investigated the human factors associated with Information Security Policy (ISP) compliance and performance by measuring compliance intention, as real compliance performance data is hard to acquire. One body of such research focuses on Security Related Stress, as developed by D'arcy [14]. This and subsequent work explores security related psychological constructs and how these individual differences influence self-reported ISP Compliance intention. While we discussed related work in greater detail in Section 1.1, this study directly answers the call issued by several specific papers.

- "...the research would be strengthened by a longitudinal design with a lag between the collection of the dependent and independent variables or through measures of actual ISP violations obtained from independent sources" [14, p. 307].
- "Even though it is plausible to assume that behavioral intention (i.e., compliance intention) can predict actual behavior, future research should consider measuring actual behaviors to clearly establish the relationship between information security-related technostress and information security compliance" [20, p. 290].
- "Opportunity 4: Cybersecurity scholars should seek to study novel stress outcomes [40, p. 115]".

Despite the perceived value of measuring actual security behaviors, most research thus far has not gathered such data. Warketin and Mutchler (2014) [44] refer to this as the "holy grail" of behavioral research in their chapter on behavior information security management, a helpful reference that surveys the theories and methods applied to behavioral information security research. A recent meta-analysis analyzed studies that measure the relationship between self-efficacy and security behavior [8]. Out of 52 peer-reviewed studies exploring behavior, only one instance effectively measured actual performance in security-related tasks. Kwak et al. (2020) [24] tasked participants with identifying phishing emails within a laboratory setting, rather than in real-world conditions.

In this chapter, we critically assess the security behaviors manifested during multi-factor authentication activities in an enterprise context. We investigate the correlation between Security Related Stress, New General Self-Efficacy, Security Related Self-Efficacy, and the observed performance of users in multifactor authentication (MFA) over a duration of seventeen months. We developed a dataset of organic user authentication events to serve as ground truth, rather than relying on users' self-reported experiences with multifactor authentication (MFA). Gaining insights into these patterns and relationships could inform targeted interventions, enhance usability, and assist in the identification of compromised accounts. This work, while preliminary, offers a novel analysis of the relationships between psychology and multi-factor authentication performance, which have not yet been assessed to our knowledge.

4.2 Methodology

4.2.1 Psychological measures

Measures: We take constructs measuring security-related Overload, Complexity, and Uncertainty from [14]. New General Self-Efficacy (NGSE) and Security Related Self-Efficacy (SRSE) were adapted from Chen et al. (2001) [9] and Compeau and Higgins. (1995) [11] respectively. The Overload, Complexity, and Uncertainty constructs are relatively new (circa 2014), and the authors were unable to find a study where they were used in analysis of authentication performance.

The two efficacy measures are designed to capture an individual's belief in their own abilities. Computer Self-Efficacy was adapted as Security-Related Self-Efficacy, for example SRSE Item 1 reads:

> "Regarding the use of 2FA for my [EDU] accounts, we could configure and use 2FA. . . if there was no one around to tell me what steps to follow."

And the Computer Self-Efficacy item it was adapted from reads:

> "I could complete my job using the technology if...there was no one around to tell me what to do."

We similarly adapt the other items without making substantive changes to the wording. The NGSE items and Security Related Stress construct items are used verbatim. In related work we reference studies that look at both state, or situational stress and trait-like representations of stress; our study assumes that the SRS variables have trait-like properties attributable to the individual, which is necessary to predict over time.

Data source: Between October 12, 2020, and January 18, 2021, 167 people at the author's university completed an IRB approved survey of psychological variables and attitudes towards MFA, 162 of the participants chose to participate in the study. The survey was collected via Qualtrics, and was composed of items for the five referenced psychological constructs, several additional items on security policy at The University of Tulsa, and user sentiment about MFA recently after its mandatory roll-out. The ability to compare user differences from survey data to observed network behavior affords a unique opportunity to draw connections between psychology and organic security control performance.

Constructs are examined as superscores, averages across all items within a user and construct. The Security Related Stress constructs are 1-7 Likert scale responses, and NGSE is Likert 1-5. SRSE questions were posed as binary response with a follow up confidence range of 1-10 for affirmative answers. "False" responses were coded as "0".

4.2.2 Authentication performance metrics

Data source: We collect authentication events data from the author's university between November 8, 2021, and March 1, 2023. Follow the methodology developed in Chapter 2, we define events as the occurrences reflected in log data that users directly experience, beginning when an authentication to a particular application is initiated, and terminated upon the eventual success, failure, or abandonment ($>$ 600 second lapse of activity) of the authentication attempt. By extracting these events from raw authentication logs, we can measure the interactive components of authentication while reducing the noise in the raw data, such as applications accessing resources or non-interactive authentications occurring in the background.

In total, this dataset includes 24,326 complete authentication events spanning 4 semesters from 115 network users who participated in the survey, with an average of 53 events per user per semester. After filtering for users who were active across all semesters, the study is left with 19,515 events from 111 users with an average of 44 events per user each semester. Events are single row representations of complete interactions. Attributes are fairly intuitive, including the time spent authenticating, result, application being authenticated to, form of authentication used, types of errors encountered, and more. In the dataset used for analysis, we summarize these events over monthly time periods for each user, and describe the specific metrics next.

Measures: We developed several performance metrics to capture not only the success users have with authentication, but also the amount of errors they encounter, and the associated time costs to a user or organization.

- Success Rate: The number of successful events divided by the number of total events for a particular user within a Period.

- Success Rank: Success Rate over a given Period relative to peers (least successful user ranked 1)

- Elapsed Time: The mean time per event in seconds across a given Period and user[1].

- Days Locked Out: The number of days within each month that a user could not successfully authenticate to any service. We require two or more consecutive, separate, failed authentication events resulting in over 6 hours unauthenticated to consider a user locked out.

- Time Away (TA): The time in minutes between a failed authentication event and the next attempted authentication; summed over the full month Period within a user. This is another measure of time cost to the user and organization.

- Friction: An error rate; the number of errors for a user in a given Period divided by the number of events they had in that Period.

- Period: An integer index variable tracking which monthly time period a given user observation is associated with.

Descriptive statistics for variables are shown and discussed in Section 4.3.1.

4.2.3 Research Hypotheses

As prior work examines relationships with compliance intention, rather than actual behavior, we hypothesized with fresh eyes; hypotheses may diverge from the expectations of prior work. The anticipated relationships between psychological constructs and response variables are shown in Table 4.1. New General Self-Efficacy (NGSE) measures the confidence someone has in their ability to be successful in their daily lives and overcome challenges. Given this, we expect those with higher NGSE will overcome errors more often, and have a

[1] Note that this captures the time between the first row of data associated with an event and the last row of the event.

Table 4.1: Hypothesized relationships

	Success	**Success**	**Elapsed**	**Timey**	**Days Locked**	
	Rate	**Rank**	**Time**	**Away**	**Out**	**Friction**
NGSE	H1a: +	H1b: +		H1c: -	H1d: -	
SRSE	H2a: +	H2b: +		H2c: -		H2d: -
Overload	H3a: -	H3b: -		H3c: +	H3d: +	H3d: +
Complexity	H4a: -	H4b: -		H4c: +	H4d: +	
Uncertainty	H5a: -	H5b: -	H5c: +	H5d: +		

higher Success Rate and Success Rank, relative to their peers. Similarly, those with greater confidence are more likely to seek help when they can't log in, resulting in lower Time Away and fewer Days Locked Out.

Security-Related Self-Efficacy (SRSE) measures the confidence someone has to succeed with technical security controls. We expect someone with greater SRSE to use security controls more proficiently, leading to the same positive relationships as NGSE. Since SRSE is specific to security controls, and not a general efficacy measure, we don't necessarily expect someone with higher SRSE to be more likely to seek help when locked out.[2] Similarly to NGSE, we expect someone with high SRSE to have lower Time Away, as they are more confident overcoming security related challenges. Unlike NGSE, we expect those with higher SRSE to be relatively lower in Friction, a measure of the frequency of errors encountered. This reduction in Friction is expected to come from a reduction in user errors relative to a low SRSE individual, and prior work indicates that the vast majority of authentication errors are user errors.

Overload, a measure of the user's perception of excessive demands placed upon them by security controls, is expected to have negative impacts on performance. We hypothesize that higher levels of Overload is associated with lower Success Rates due to the increased cognitive burden leading to more frequent mistakes and reduced perseverance in resolving errors. Consequently, users experiencing high Overload are expected to exhibit higher Time

[2] A review of the events causing lock-outs shows a vast majority of errors are configuration errors. Thus, we expect someone's proficiency to have little bearing on their chances of getting locked out.

Away. Additionally, Overload is likely to result in more Days Locked Out and higher Friction rates, as the strain from excessive security demands leads to more frequent errors and failures.

Complexity captures contexts in which security requirements require significant time or effort to learn and understand. While multi-factor authentication may be a new experience for some users, its usage is relatively static; consequently, we don't expect a great difference in raw performance for users who have higher security related complexity. As perceived complexity may drive the level to which a user engages with the security control, high complexity users may also be more prone to seeking compensatory tools, such as a password manager, to offload some of the burden. With those considerations, no hypotheses were made about the relationship with success rate or fortitude. Instead, we hypothesize that users with high complexity will also have longer Time Away, as they may expend more time or effort to address a failure. Similarly, we hypothesize a positive relationship between complexity and how long or often a user is locked out of their account.[3]

Uncertainty measures the user's perception of the unpredictability and lack of easy understanding related to security controls, policies, and procedures. We expect higher levels of Uncertainty is associated with lower Success Rates and higher mean Elapsed Time, as users may be less confident in their ability to navigate the authentication process, leading to mistakes and longer time spent authenticating.

As this is a first test between these constructs and the observed authentication metrics, the hypotheses function as a vehicle for our analysis. Should the analysis be confirmatory, all we can say is that we have observed that relationships do exist. Further studies and theory building are required to properly explain any such relationships and quantify their strength.

[3] Complexity was not observed to have significant relationships throughout analysis, so we omit it from discussion for brevity

4.3 Analysis

4.3.1 Summary Statistics

As we transition into the analysis phase, we delineate our independent variables in Table 4.2 and response variables in Table 4.3. Examining our independent variables, we note large correlations between SRSE and NGSE, and similarly sized inverse correlation with Overload and Complexity. NGSE shows inverse correlations with all three SRS constructs, and Overload has large positive correlations with Uncertainty and Complexity.

	Mean	SD	Correlations				
			1.	2.	3.	4.	5.
1. SRSE	7.82	2.20	**.94**				
2. NGSE	4.17	.57	.32	**.86**			
3. Overload	2.90	1.29	-.32	-.19	**.87**		
4. Uncertainty	3.93	1.15	-.06	-.08	.30	**.83**	
5. Complexity	3.54	1.06	-.30	-.25	.58	.11	**.73**

Table 4.2: Means, standard deviations, correlations, Cronbach's Alpha for independent variables.

	Mean	SD	Correlations							
			6.	7.	8.	9.	10.	11.	12.	13.
6. Success Rate	.94	.18	**.76**							
7. Success Rank	49.3	13.9	.74	**.63**						
8. Elapsed Time	37.2	151.1	-.11	-.10	**.56**					
9. Days Locked Out	17.7	19.4	.27	.54	-.03	NA				
10. Time Away(hrs)	64.88	238.6	-.16	-.18	.03	-.19	**.31**			
11. Friction	.08	.19	-.71	-.54	.16	-.20	.29	**.73**		
12. Period	7.04	4.06	-.27	-.28	.08	-.21	.11	.21	NA	

Table 4.3: Means, standard deviations, correlations for monthly period response variables, Cronbach's Alpha on the diagonal[5]

Moving to our response variables, we first notice the large correlation we anticipated between Success Rate and Success Rank. Success Rate has a similarly large inverse cor-

[5] calculated using the first 10 months of data to limit missing-ness and avoid the influence of the observed natural experiment in SP23

relation with Friction, as failures driven by the errors experienced during authentication, and both relationships are echoed by the Success Rank variable. Next, we see a positive correlation between Elapsed Time and Friction, and an inverse relationship between both Days Locked Out and Time Away with Friction. These results are largely intuitive, but the positive relationships between Success Rate and Rank with the negative performance metric Days Locked Out are puzzling.

	1. SRSE	**2. NGSE**	**3. Overload**	**4. Uncertainty**	**5. Complexity**
6. Success Rate	-0.03	**-0.11**	**-0.11**	-0.02	**-0.06**
7. Success Rank	-0.03	-0.00	-0.03	-0.05	-0.03
8. Elapsed Time	0.03	0.01	0.00	0.01	-0.01
9. Days Locked Out	0.02	0.05	0.02	-0.00	0.05
10. Time Away(hrs)	**-0.12**	-0.08	0.02	-0.00	0.00
11. Friction	-0.03	0.01	**0.08**	-0.05	**0.08**

Table 4.4: Correlation Results between independent and response variables, correlations significant at the 0.05 level in bold font

Finally, we examine the correlations between Independent and Response variables in Table 4.4.[6] We observe significant inverse correlation between Success Rate and NGSE, Overload, and Complexity. The relationships with Overload and Complexity are intuitive, as those stressors increase, authentication success would naturally decrease. The inverse relationship with NGSE is counter-intuitive, as we expect those with higher generally self-efficacy to perform in line with their elevated confidence. We explore this result more in later sections.

Time Away has a significant inverse correlation with SRSE; users with higher Security Related Self-Efficacy are correlated with less Time Away after authentication failure, which matches our intuition. Friction had significant correlations with both Overload and Complexity. Friction is a simple measure of errors per event; this suggests as a user has increasing Security Related Overload or Complexity, they experience more errors.

[6] We omit the Period variable from these correlations, as the various self-reported construct superscores were collected at a single point

4.3.2 Single Predictor Regressions

A series of single predictor regressions were conducted to evaluate our hypotheses against within-user averages across construct items we call construct superscores. Single item regressions were performed using authentication event data aggregated within users across a monthly time period.[7] Natural log transforms were used for both the construct averages and response variables, enabling an intuitive reading of each beta values as an elasticity.[8] Using hypothesis **H3c** in Table 4.5 as an example: a .88 beta value means a 10% increase in Security Related Overload is associated with in a 8.8% increase in mean Time Away after failure per month. We set .05 as our threshhold significance for hypotheses support, and bold results that reach significance when listing hypotheses.

Simple regressions supported five of our twenty-one hypotheses. Two additional hypotheses were inversely related but significant: NGSE shows a negative relationship with Success Rate and positive relationship with Days Locked Out. Users with higher NGSE had lower success rates and had more days in which they were locked out of digital systems. Three Overload relationships were supported: Success Rate, Success Rank, and Time Away. Highly overloaded users were less successful, and spent more time away from their accounts after a failed event. A 10% increase in Uncertainty was associated with a 10.9% increase in Time Away after a failed event, and a 0.4% decrease in Success Rate.[9][10]

4.3.3 Multiple Regression Analysis

Next, we move beyond single regression first through incorporating two control variables, then moving to multi-construct regressions. The control variable Period has a monthly

[7] Regressions revealed that weekly periods capitalized on chance and found significant (but small) relationships where none existed on the semester or monthly time scales.

[8] When both the dependent Y and independent X are log-transformed, the coefficient β in the regression model can be interpreted as an elasticity, which represents the percentage change in Y for a one percent change in X

[9] These results are qualitatively unchanged when using the bi-weekly or per semester datasets

[10] Analysis was replicated on datasets including the summer months, anticipating this data would be less reliable due to reduced student activity. Results confirmed this intuition, yielding less significant relationships across the board.

Table 4.5: Hypotheses, support indicators, and regression statistics

Hypothesis	**Construct**	**Metric**	**Supported**	**Beta**
H1a	**NGSE**	**Success Rate (+)**	**No**	**-0.16**
H1b	NGSE	Success Rank (+)	No	−0.24
H1c	NGSE	Time Away (-)	No	0.63
H1d	**NGSE**	**Days Locked Out (-)**	**No**	**0.51**
H2a	SRSE	Success Rate (+)	No	−0.03
H2b	SRSE	Success Rank (+)	No	−0.05
H2c	SRSE	Time Away (-)	No	−0.55
H2d	SRSE	Friction (-)	No	0.03
H3a	**Overload**	**Success Rate (-)**	**Yes**	**-0.05**
H3b	**Overload**	**Success Rank (-)**	**Yes**	**-0.12**
H3c	**Overload**	**Time Away (+)**	**Yes**	**0.88**
H3d	Overload	Days Locked Out (+)	No	−0.06
H3e	Overload	Friction (+)	No	−0.15
H4a	Complexity	Success Rate (-)	No	−0.03
H4b	Complexity	Success Rank (-)	No	−0.07
H4c	Complexity	Time Away (+)	No	0.27
H4d	Complexity	Days Locked Out (+)	No	−0.11
H5a	**Uncertainty**	**Success Rate (-)**	**Yes**	**-0.04**
H5b	Uncertainty	Success Rank (-)	No	−0.12
H5c	Uncertainty	Mean Elapsed (+)	No	−0.52
H5d	**Uncertainty**	**Time Away (+)**	**Yes**	**1.09**

Note: Bold font indicates significance at the 0.05 level

frequency, inclusion of this variable into our initial regressions allows us to observe if users' performance changed over time. When re-running our regressions adding Period, all previously significant relationships from Table 4.1 remained, with no qualitative changes to effect sizes or significance.

Our second control variable is PrimaryMFA, which is an important moderator to the 2FA experience. Users may experience different issues depending on the type of second factor used. In our dataset, PrimaryMFA includes three second factor types: SMS, App Notification, and OATH code.[11] These forms of 2FA events do not include instances where no second factor presentation is required due to fulfillment by session token, or similar temporary credential, which don't require interaction by the user. One common MFA

[11] Phone Call MFA was also present, but removed due to having only 19 associated observations

feature, where the user can choose to “Remember my Device”, enables the user’s device to serve as the second factor confirmation. This type of authentication is included when the authentications are interactive through password entry or similar. Finally, we add NumEvents, the number of events in a given period as a third control variable.[12]

Overload, Uncertainty, and Stress are sub constructs of the Security Related Stress (SRS) second-order construct; we expect them to only increase the significance of our observed relationships when included, as they are designed to capture orthogonal variance. Of our two efficacy constructs, only NGSE has significant relationships using single regression, but controlling for users’ reported SRS may help clarify these relationships. All construct superscores are included in the regressions with control variables Period, NumEvents and PrimaryMFA. We evaluate these regressions for each response variable, and present the results in Table 4.6.

Overload was significantly related to Success Rate with an effect size of -0.07. The relationship with Success Rank was insignificant after controlling for other constructs, with the p-value dropping to 0.14. Overload was also related to Time Away and Friction with considerable effect size, but did not meet our standard for significance. A negative relationship between Overload and Friction runs counter to our intuition, as overloaded users may be expected to make more mistakes, not less. Similarly, we expected overloaded users to be more likely to stay away longer following an authentication failure, contrary to the direction of the relationship implied by these regressions.

After controlling for the other constructs, neither Uncertainty nor Complexity have statistically significant relationships with any of the response variables. Moving on to our two self-efficacy constructs, we see no significant relationships with SRSE. NGSE had a highly significant inverse relationship to Success Rate with an effect size of -0.20. This negative relationship is puzzling and bears further investigation. If we assume that those with higher self-efficacy are more competent or capable, this suggests some users may be

[12] Single regressions were recomputed with control variables added; results were consistent with Table 4.5.

Table 4.6: Regression results

	Dependent variable (larger or smaller values “better” indicated below):					
	ln(Success Rate) *Larger*	ln(Success Rank) *Larger*	ln(Elapsed Time)	ln(Time Away) *Smaller*	ln(Days Locked Out) *Smaller*	ln(Friction) *Smaller*
ln(Overload)	-0.07^{***}	-0.10	0.01	-0.07	0.03	-0.21
ln(Complexity)	-0.003	-0.08	0.31	-0.16	-0.13	0.28
ln(Uncertainty)	0.01	-0.03	-0.73	0.83	-0.08	-0.49
ln(NGSE)	-0.19^{***}	-0.28	-0.08	-0.01	0.12	-0.97
ln(SRSE)	-0.03	-0.12	-0.24	0.49	-0.07	-0.13
Period	-0.004^{*}	-0.06^{***}	-0.08^{*}	0.04	-0.03^{***}	0.03
App Notification	-0.003	0.03	-0.24	0.03	-0.003	0.33
OATH Code	-0.15^{***}	-0.26^{*}	-1.02^{*}	0.59	-0.12	0.76^{*}
Remembered Device	-0.02	-0.03	-4.07^{***}	-2.82^{***}	-0.55^{***}	-1.35^{***}
numEvents	-0.001^{**}	-0.001	0.09^{***}	0.09^{***}	0.04^{***}	0.03^{***}
Constant	0.38^{***}	4.88^{***}	1.37	-6.85^{*}	2.08^{***}	-4.83^{***}
Observations	1,132	1,132	1,132	1,132	1,132	1,132
Adjusted R^2	0.06	0.07	0.21	0.07	0.70	0.06

Note: Primary MFA uses Text Message MFA as reference level; $^{*}p<0.05$; $^{**}p<0.01$; $^{***}p<0.001$

over confident in NGSE responses.

Finally, we look at the relationships with our control variables: Period, MFA Type, and NumEvents. The control variables are not natural log transformed; for these relationships, we exponentiate the beta value to find the percentage change in our response variable relative to the reference category. Period was negatively associated with Success Rate and Rank, Elapsed Time, and Days Locked Out. These relationships indicate that over time, users' spent fewer days locked out, and less time authenticating; conversely, they failed more often, though the effect size is very small. The type of second factor used in authentication was also significant in our analysis. Mobile App MFA had no significant relationships relative to the reference method Text Message second factor. OATH Code MFA was significant and inversely related to Success Rate and Rank, and Elapsed Time. This suggests that users relying on OATH Code as a second factor experienced a higher failure rate than those using Text as a second factor, with a much larger effect size. Additionally, the relationship with Time Away indicates they spent nearly three times as much time authenticating. The positive correlation with friction reveals that OATH Code users encountered more than twice as many errors compared to their Text MFA peers, which likely contributed to the lower success rate. Use of the “Remember My Device” option, resulting in MFA fulfilled by a “Remembered Device” was associated with significant reductions in Elapsed Time, Time Away, Days Locked Out, and Friction, as expected. These results underscore the benefits of adopting this feature for the user experience and emphasize the significant influence of the chosen MFA type. Finally, we look at the relationship with the numEvents control variable. The number of events had a small negative relationship with Success Rate and Rank, which we expect, as users who fail to authenticate and attempt to reauthenticate will have more events in the same period of time. Similarly, it had a significant positive relationship with Elapsed Time, Time Away, Days Locked Out, and Friction.

Regressions with Self-Efficacy Categorical Variables: Throughout the analysis we observe a negative relationship between NGSE and users' Success Rate, and Success Rank, which is their performance relative to their peers for a given period. This, along with other unexpected results, suggest that we may have an uncontrolled effect confounding our results. We posit this may be due to poorly informed users rating their NGSE too highly; as we close out our analysis, we briefly investigate this result. The left plot in Figure 4.1 shows the distribution of the NGSE measure, an average of NGSE item responses.

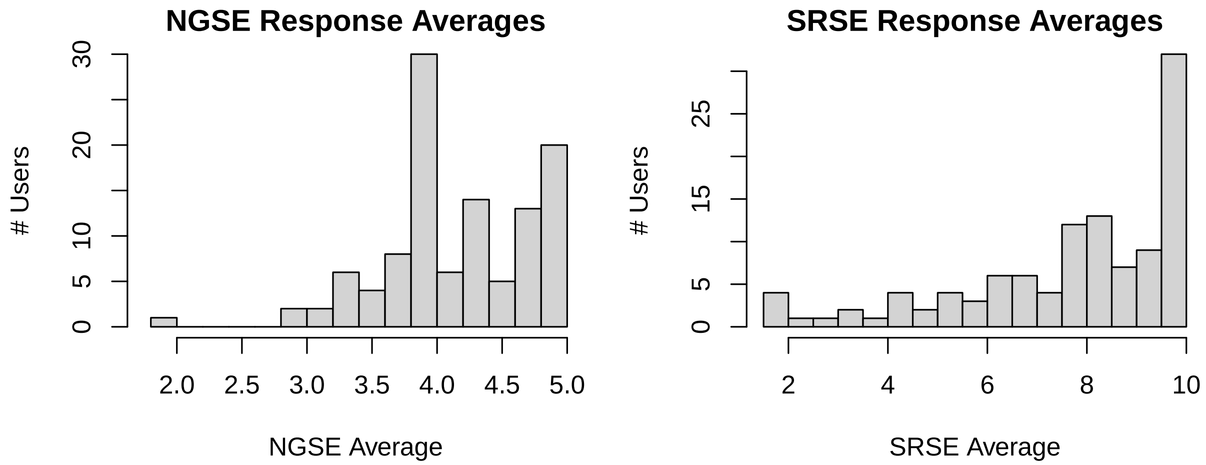


Figure 4.1: Self-Efficacy constructs superscore distribution

We break NGSE scores into three roughly equal sized categories based on the his-

Table 4.7: Multiple regression with categorical efficacy variables

	Dependent variable:					
	ln(Success Rate)	ln(Success Rank)	ln(Elapsed Time)	ln(Time Away)	ln(Days Locked-Out)	ln(Friction)
ln(Overload)	−0.06***	−0.08	0.02	0.89**	0.04	−0.25
ln(Complexity)	−0.03	−0.13	0.33	−0.26	−0.16*	0.23
ln(Uncertainty)	−0.001	−0.03	−0.72	0.81*	−0.08	−0.59*
LOW NGSE (n = 308)	−0.08***	−0.20**	0.21	0.02	0.01	0.51*
HIGH NGSE (n = 384)	0.03	0.005	0.29	−0.62	0.01	0.90***
LOW SRSE (n = 345)	0.02	0.07	−0.07	0.31	−0.02	−0.16
HIGH SRSE (n = 409)	0.06**	0.17*	−0.11	0.16	−0.002	−0.07
Period	−0.004*	−0.06***	−0.08*	0.07*	−0.03***	0.04
App Notification	0.01	0.08	−0.35	0.39	−0.02	0.18
OATH Code	−0.14***	−0.22*	−1.06*	0.98*	−0.12	0.63
Remembered Device	−0.01	−0.002	−4.11***	−1.66**	−0.55***	−1.50***
numEvents	−0.001	−0.0002	0.09***	−0.01	0.04***	0.03***
Constant	0.06	4.25***	0.70	4.10***	2.16***	−6.49***
Observations	1,151	1,151	1,151	442	1,151	1,151
Adjusted R^2	0.07	0.07	0.21	0.08	0.70	0.07

*p<0.05; **p<0.01; ***p<0.001

togram, using breaks at the values 4.0 and 4.4.[13] As SRSE and NGSE were correlated in our analysis at 0.31 and are thematically similar, we repeat this process on SRSE, which has a similarly large rise in the distribution of response averages near the ceiling. We split the SRSE superscores at 7.5 and 9 after consulting the second distribution plotted in Figure 4.1, yielding balanced groups. Replacing the NGSE and SRSE superscores with response categories allows us to control for this potential non-linear correlation with performance metrics. In the regressions, we use the medium score ranges as baseline, so we can look at how low and high-scoring individuals perform relative to those in between.

The regression results controlling for both SRSE and NGSE response levels are shown in Table 4.7 using the natural log transforms on each variable except for our new categorical variables. Starting with the Success Rate response variable and our new categorical variables, we find Low NGSE is negatively related to both Success Rate and Success Rank, relative to their Medium NGSE peers. This contradicts the counter-intuitive results of earlier regressions, indicating that moderate NGSE responses are associated with higher success rates than the lowest NGSE users. However, we note that High NGSE users did not have significantly higher success than their Medium NGSE peers, suggesting some users

[13] Note that the graph shows user response average frequencies for 111 users, and the number of observations associated with each user depends on presence in the authentication dataset.

may be over-confident in their responses. More work is needed to investigate this result. Low NGSE is also correlated with higher friction, with those in the low category having 66% higher incidence of errors than their medium NGSE peers. Interestingly, high NGSE users, while not having significantly different Success Rates, also have much higher error incidence, at 145% higher Friction. We hypothesize that this result suggest high NGSE users engage in more complex or diverse authentication behaviors, which result in higher error rates, but balance that behavior with greater remediation skills, resulting in equivalent Success Rates. As we may expect, high NGSE users were associated with 33% higher time spent authenticating, which may capture the time spent remediating those errors, though this result did not reach significance.

In summary, moderate NGSE responses were associated with the highest Success Rates and Ranks, fewest Days Locked Out, and least errors per authentication event, while high NGSE responses correlated with equivalent success, but higher error rates and more time spent authenticating. Low NGSE responders had the lowest authentication success, and elevated error rates.

SRSE, for which no significant relationship was found in the prior regressions, is significantly related to Success Rate and Rank. High SRSE users show elevated Success Rate and Rank, with a 6% increase in absolute Success Rate over medium SRSE users and a 19% increase in ranking compared to medium SRSE peers. Both High and Low SRSE users are associated with a reduction in Days Locked Out compared to Medium SRSE peers, at 17% and 36% fewer respectively, with corresponding decreases in Friction, though these results did not reach significance.

Overload gained a new significant relationship in this analysis, with a 10% increase correlated with an 8.9% increase in Time Away. This realigns the results with our expectations, as the prior analysis showed no significant change to Time Away for overloaded users. Complexity gained its first significant relationship, with high Complexity correlating with fewer Days Locked Out. Taken in combination with insignificant but large effects showing an increase in Friction and Elapsed Time, this suggests that users who see the authentication experience as more complex may spend additional time authenticated, working through their higher rates of errors for equivalent results. This persistence to eventual success could explain the reduction in Days Locked Out, as users spent additional time to successfully authenticate, and thus end up with fewer sessions ending in failure, a precursor to lockout status.

Uncertainty showed its first significant relationships in this analysis. The large preexisting effect sizes with Time Away and Friction became significant after controlling for self-efficacy response levels. A 10% increase in Uncertainty was associated with an 8.1% increase in Time Away and a 5.9% reduction in Friction. Unlike the Complexity measure, this relationship was accompanied by a large but statistically insignificant decrease in Elapsed Time. Taken together, these findings suggest that users with higher Uncertainty around the authentication experience may be more risk-averse, encountering fewer errors but demonstrating less resilience, as reflected in shorter authentication times. If users achieve similar Success Rates despite spending less time authenticating, encountering higher error rates, and exhibiting more Time Away, it may indicate that they experience a higher cost for each error. These users might abandon the authentication process without attempting to resolve errors and spend more Time Away following a failed attempt. This makes intuitive sense and agrees with our hypothesis H5d.

In summary, controlling for NGSE and SRSE response levels clarified the relationships between self-efficacy measures and performance metrics. Moderate NGSE users demonstrated the best overall performance, with the highest Success Rates, Success Ranks, and lowest error rates, while high NGSE users achieved equivalent Success Rates despite higher error rates and increased time spent authenticating. Low NGSE users, however, exhibited the poorest performance metrics, including the lowest Success Rates and elevated Friction. SRSE, previously insignificant, became a key predictor, with high SRSE users achieving higher Success Rates and Ranks and fewer Days Locked Out. Overload and Complexity measures also gained significance, aligning with hypotheses regarding their influence on Time Away and Days Locked Out, respectively, suggesting that users adapt their behavior to manage these challenges. Uncertainty, while reducing Friction, was positively associated with Time Away, highlighting a potential risk-avoidance behavior among users with higher uncertainty levels. Importantly, these results reaffirm the significant role of second-factor authentication types across all metrics and reveal that raw Success Rate and Friction were notably unaffected by time, underscoring the complexity of authentication behaviors.

4.4 Discussion

We now summarize the main findings of the analysis after controlling for both time and type of second factor used, and breaking NGSE and SRSE supserscores into response level categorical variables to control for response biases. Table B.1 in the appendix lists our original twenty-one hypotheses with additional sub-hypotheses to account for NGSE response categories, self-efficacy hypotheses are only broken into their constituent categories where necessary to discuss significant hypotheses. Significant results are in bold font, and labeled supported when both significant and the effect size is in the predicted direction. In this exploratory study, the hypotheses results are secondary to the relationships discovered in the analysis.

This study examined how self-efficacy, stress, and psychological factors influence multi-factor authentication (MFA) performance, using authentication logs collected over 17 months. By categorizing NGSE and SRSE responses, we identified nuanced relationships between self-efficacy levels and security performance.

Moderate NGSE responses were associated with the best overall performance, suggesting that balanced self-efficacy fosters effective authentication behaviors. Low NGSE users faced the greatest challenges, including lower success rates and higher lockout occurrences, highlighting the impact of insufficient confidence. Interestingly, high NGSE users achieved comparable success rates to moderate users but encountered significantly more errors, pointing to potential overconfidence leading to riskier or more complex authentication behaviors.

SRSE emerged as a robust predictor of success, with high SRSE users demonstrating increased success rates and fewer lockouts compared to their medium and low SRSE peers. These findings emphasize the importance of confidence in security-related skills, particularly in high-stakes environments.

Stress-related constructs also played significant roles. Overload correlated with longer Time Away after failed authentications, reflecting the cognitive burden placed on users. Uncertainty, while reducing error rates, increased Time Away, suggesting that risk-averse behaviors may trade efficiency for cautious decision-making. These results provide insights into how psychological factors shape user interactions with authentication systems and inform design considerations for reducing user friction.

Control variables highlighted key trends, such as improved lockout rates over time and variations in performance by MFA type. The “Remember My Device” token option stood out as an effective feature, reducing errors, lockouts, and time costs, underscoring its value in improving user experience.

CHAPTER 5

SOC DASHBOARD UTILITY

5.1 Introduction

A Security Operations Center (SOC) serves as the “nerve center” of an organization’s cybersecurity efforts. It should receive inputs from multiple sources, be sensitive to stimuli that may signal danger, and present the organization with a comprehensive representation of its environment. The primary functions range from monitoring, assessing, and defending against cyber threats, to surveillance of networks, servers, applications and users. This enables the SOC to identify pain points, potential vulnerabilities, and areas for improvement.

The number of digitally connected systems has rapidly increased as the transition to cloud services continues. This expansion has resulted in larger attack surfaces and a growing number of users and systems subjected to active monitoring, consequently leading to a significant increase in the volume of alerts that organizations encounter on a daily basis. The average SOC team now experiences thousands of alerts per day on average, which leads to 67% being ignored due to alert fatigue and a high volume of false positives [43].

As such, a SOC is heavily limited by the quality of its inputs, i.e., its data sources. Many tools are utilized to develop and leverage data sources, such as Security Information and Event Management (SIEM) systems, Intrusion Detection Systems (IDS), vulnerability management tools, and other analytical tools. These systems work together to enable SOCs to detect, investigate, and respond to issues at speed, with accuracy.

One primary data source for SOCs is authentication logs. Controlling who uses a given service or application, and in what capacity, is key to both proper security and functionality. Many organizations have deployed single sign-on (SSO) services such as Microsoft Azure AD (now Azure Entra AD) to streamline their users’ authentication experience.

5.2 Exploratory Anlaysis

In this chapter, we discuss examples of the event log and its derivatives that may be of interest to a SOC or IT team. The event data utilized spans around three semesters: Spring and Fall of 2022 and Spring of 2023. These slices include one week prior to the first day of class and end one week after the semester concludes; January 8th through May 17th for the spring semesters, and between August 13th and December 19th for the Fall. Filtering for users that had at least one successful authentication left 1.7m events across 7,419 users, an average of 77 authentication events per user, per semester.

5.2.1 Basic Outcome Measures

The examples discussed in this section demonstrate the utility of user-focused event aggregates and their derivatives. A proactive SOC may directly utilize some of these capabilities beyond the standard authentication log use cases of alert diagnosis and incident response. For example, the detection of lapsed applications discussed in section 5.2.2 could be used to reduce threat surfaces by retiring unused applications. As we consider the utility of an event-based approach to authentication logs in a SOC, we begin by examining the basic unit of analysis, the event, before moving on to derivative measures. Each event reports success or failure, the time elapsed, the form of MFA used, types of errors encountered, and application being authenticated to. The most straightforward measure then, is failure rate, the complement of success rate.

An intuitive way to examine failure rates is by error content: does error type impact the user experience differently? We anticipate that errors caused by users are both more common and more easily resolved; passwords can be re-entered, MFA can be properly completed, etc. We find that over 80% of users who encounter a configuration error will never succeed when they experience a configuration error, and 94% of events containing a configuration error end in failure. Conversely, we find that only 7% of users who encounter user errors will never succeed when they experience a user error, and only 56% of events

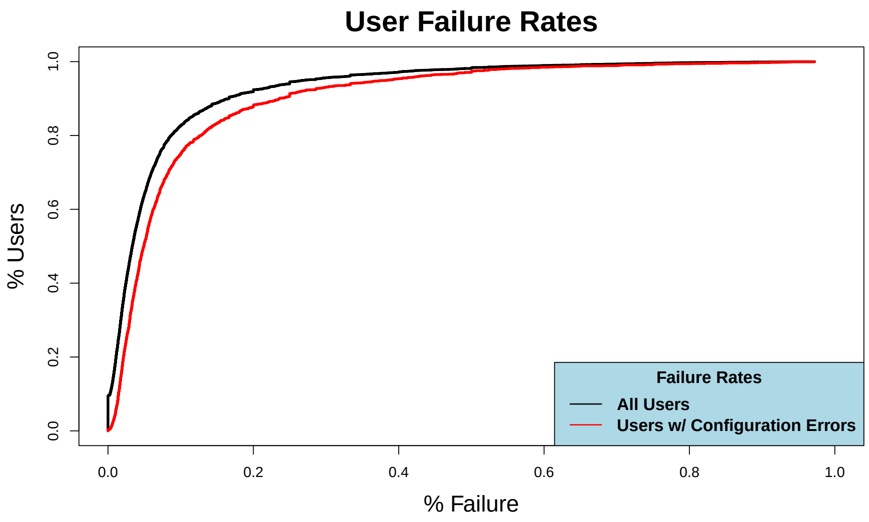


Figure 5.1: Failure rate per user, considering errors.

with user errors conclude in failure. While configuration errors are clearly more difficult to resolve, they are also less common. 93% of users experience user errors, while only 27% of users experience configuration errors. This confirms our expectation that user errors are both more common and more easily resolved.

Examining the cumulative distribution function (CDF) plots in Figure 5.1, the majority of our 7,305 valid users experience a very low failure rate. Mean failure rate is 8%, with the 10% worst users failing over 20% of authentications, and the 10% best users fail only 0.4%. The failure rate increases substantially for our worst users when we examine those who ever experience configuration errors, plotted here in red. The 10% best users fail only 1.5% of authentication events, whereas the 10% worst fail over 30% of authentications. The bottom 5% fail an astounding 47% of authentication attempts.

We can take away a few lessons from these distributions for utilizing event data in a SOC. First, configuration errors may be worth investigating, as they reliably trigger failures through no fault of the user. Second, relatively few users fail frequently, and it may be beneficial to target efforts at assisting these struggling users.

5.2.2 Identification of Locked-Out Users

Creating derivative metrics lends greater utility, such as the ability to identify locked-out users. An alert prompted by a lockout metric might trigger automated assistance, which in turn could forestall help tickets and issue early alerts for developer issues that cause service interruptions.

To construct this measure, we first add helper variables to our event dataset: we add a "consecutive failures" and "hours away" attribute to each event. Next, we set a variable Lockout to true when consecutive failures is greater than one and Time Away exceeds twelve hours. Each week is summarized by the longest lockout experienced for each user. Figure 5.2 shows the number of users locked out for each week of the semesters. The average number of users locked out for more than 12 hours, each week in the semester, was approximately 2.5% (152 of 6017) of the total. If we filter this for lockouts over 24 hours in duration, this shifts to a 105 users per week, or 1.7% of our users.

Next we plot another set of CDFs, this time examining the duration of lockouts. As configuration errors affect failure rates more than user errors, we plot Figure 5.3 with a series of mixed errors in black, and a series with only configuration errors in red. Across three semester, we observe 8350 lockouts for 2656 unique users, which is 36% of our total user base. We note that nearly 93% of lockouts were associated with both user and configuration errors, and the mean ratio of CEs to UEs for those lockouts was 3.7. Over 6% of lockouts only had configuration errors, and less than 1% only had user errors. Lockouts commonly persist beyond 12 hours, with a median lockout duration of 43 hours, and the 90th percentile being locked out for over 193 hours, or 8 days. Lockout times begin to diverge based on error composition after the 24 hour mark and are longer when caused by configuration errors.

Lockouts happen often enough to benefit from proactive investigation and resolution, but they are uncommon enough to not overwhelm analysts. Moreover, since lockouts can persist for a long time, steps to eliminate them sooner would bring substantial value.

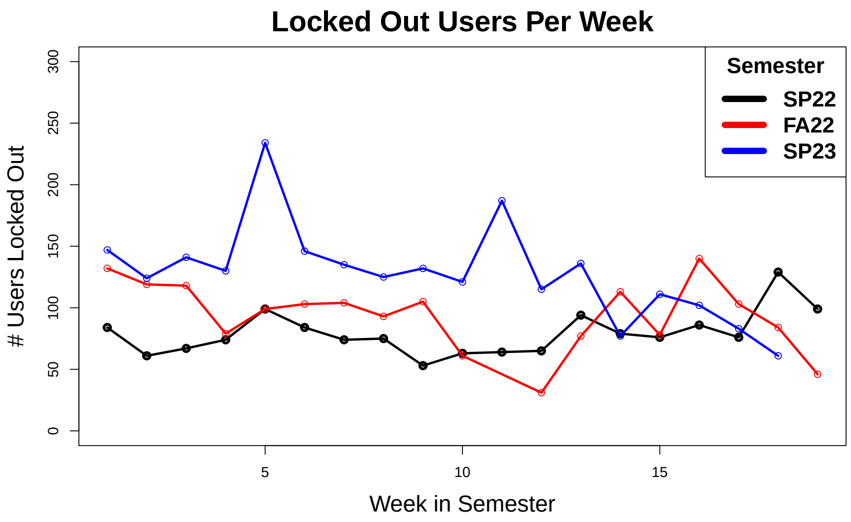


Figure 5.2: Number of locked out users per week in each semester.

Identification of Lapsed and Struggling Applications: Maintaining the security and performance of enterprise applications is a key function of a SOC. Applications that are unused and/or not associated with any successful authentications present a security risk; these applications are more likely to lapse into unsafe states, and misuse may be harder to detect. In our data, we observe 689 unique applications across three semesters, 348 of which never show a successful authentication. In our organization, over 50% of applications can be easily identified and classified as lapsed, and may be de-commissioned to increase security.[1] These lapsed applications may otherwise persist for long periods of time, as we observe in the bar chart in Figure 5.5, which shows the number of valid and invalid applications per semester.

The next utility is early identification of struggling applications. Using the most recent semester, SP23, we first filter out the lapsed applications with no record of successful authentications. This results in a median success rate of 95% percent, closely matching our median user success rate for that semester of 94%. The mean success rate per application is somewhat lower, at 76%, indicating that some of our highly used applications have lower

[1] We do not currently posses a master list of applications for our organization, and can only detect applications with at least one authentication attempt made. In a SOC setting, this is easily remedied to be exhaustive and complete.

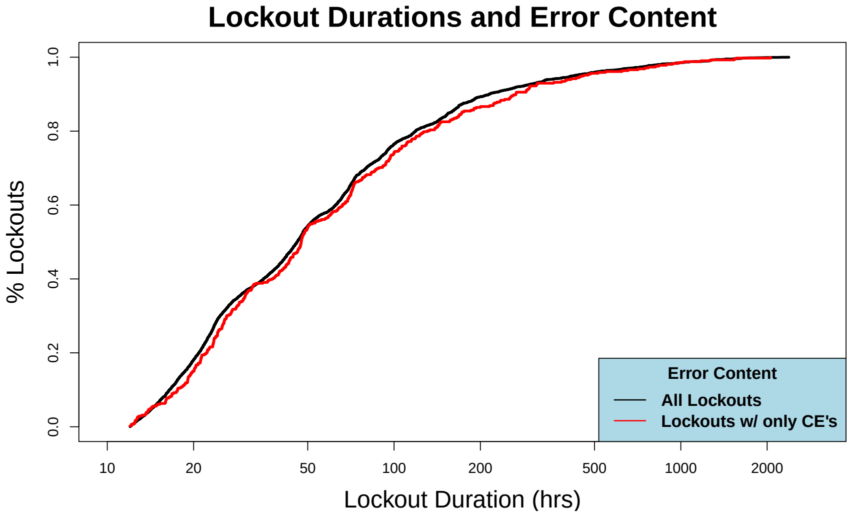


Figure 5.3: Cumulative distribution function of lockout duration

success rates. Examining the 20 most used applications, which in our data incur an average of 140 unique users per week, we plot the per application success rate over time to observe struggling applications. We define a struggling application as an application experiencing a success rate 50% below its mean success rate across the semester. In Figure 5.4 we report the lagging top 20 applications per week in the SP22 semester.

As one might expect, the top applications usually perform well, but it is not uncommon for one or a few to be lagging. Taking a specific application as an example, Microsoft Teams had a fairly low mean success rate of 53% in the SP23 semester, compared to 78% in the SP22 semester. Our "Lagging" metric flags a per-day success rate of under 10% on the last day of week 9, pointing to acute issues with the application. The graph in Figure 5.6 shows the downward trend of weekly success rate and its impact on the success rate across all applications in the following weeks. Early identification of such issues is key in reducing the impact of lagging applications on an organization.

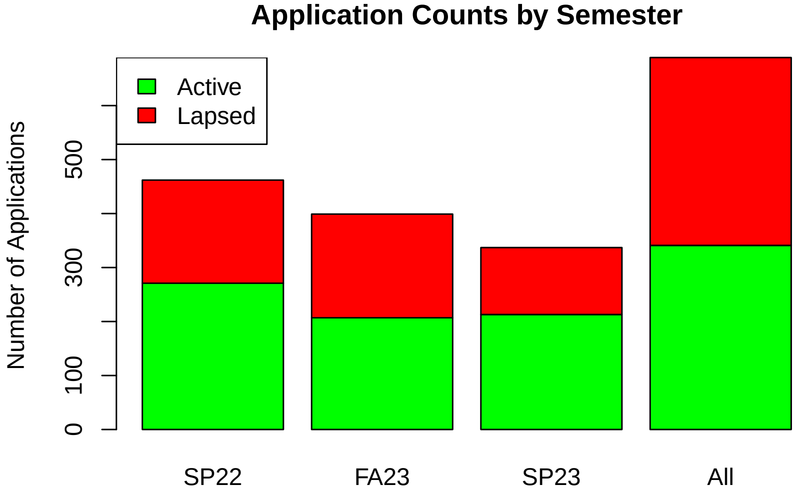


Figure 5.4: Lapsed and active applications per period

5.3 SOC Tool Development

5.3.1 Motivation

Our work on authentication logs has produced promising results, offering insights into user and application health, and demonstrating value in econometric analysis. The next logical step is to test the usefulness of our methodology and derivative metrics in a real SOC environment. Testing in a production SOC will give us access to valuable feedback from security analysts in the real world, and allow us the opportunity to refine features and implementation.

As we seek to provide value to the SOC, we began by meeting with our university SOC to present our findings and seek feedback. In the Spring of 2024, the authors held a meeting with the University of Tulsa's IT Security team to review our research findings, which was made possible by the departments ongoing support. SOC leadership expressed interest in our results, and agreed to allow the authors to develop a tool to provide the event log and derivatives to the SOC as a supplemental tool. The event log itself is to be made available in near real-time, with less frequent updates to derivatives. Deployment of

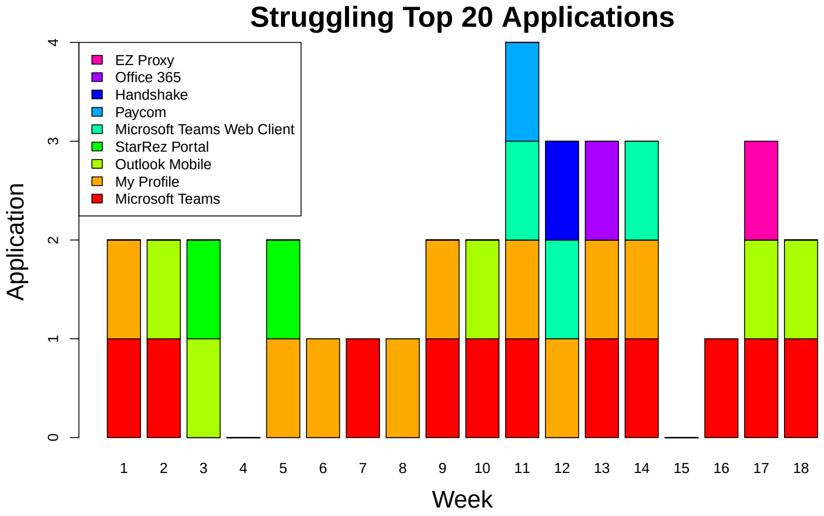


Figure 5.5: Lagging top applications per week

the log and its derivatives in the university SOC will allow us to study how authentication logs are used in a SOC, the perceived value of event logs as a replacement or enhancement for traditional alert diagnostics, and more.

5.3.2 Development

The SOC dashboard, still in the 0.x phase under active development, was developed and tested using Qt Creator, a cross-platform integrated development environment (IDE), written in C++ [36]. The libraries utilized were primarily native QT libraries, with an open source json parser from nlohmann[27]. Threading was implemented using the QThread class, and API calls are performed using the curl c++ library [42]. The dashboard ingests raw sign-in logs and security alerts from the Microsoft Graph API to generate authentication events. These events subsequently facilitate the production of contextual derivatives and aggregates, which can be visually represented and associated with security alerts.

5.3.3 Implementation

The dashboard is designed as a single-window application, incorporating tabs or “panels” for each category of data utilized, and presenting this information through visual-

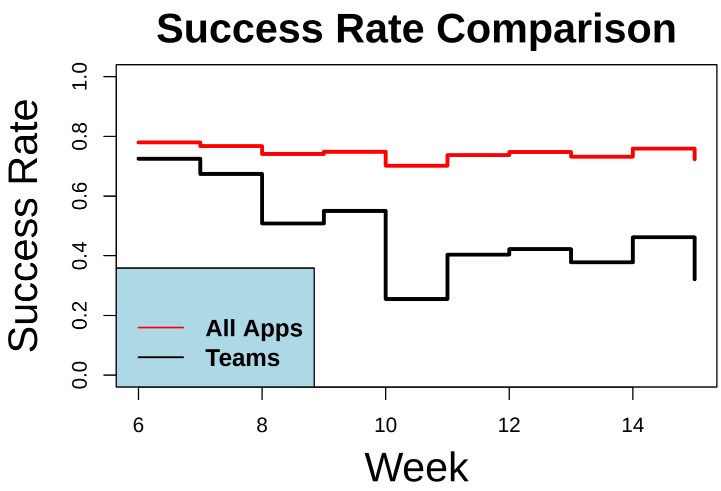


Figure 5.6: Success rate of Microsoft Teams vs all applications

izations such as charts and graphs. The layout is methodically organized to provide access to three primary categories of data: derivative reports that utilize higher-level metrics related to both individuals and groups, raw and aggregated event authentication logs, and security alerts. Lastly, an "alert details" tab links a given alert to relevant authentication data and reports. Table 5.1 lists the relevant panels, their purpose, data types and selected examples.

User Logs: User logs are event-derived data tables summarizing key event attributes for specific users over a given time period. In the research setting, periodic log aggregation has been used to transform longitudinal event logs into panel data, offering opportunities to perform various kinds of analysis as we reviewed in Chapters 3 and 4. These logs include the time spent authenticating, number of failures, type and frequency of error encounters, time spent unauthenticated after failure, forms of second-factor authentication used, the number of applications being authenticated to, and more.

Reporting artifacts generated from this enhanced data aggregation include graphs of population success and error rates, and the number of users currently locked out of their

Table 5.1: Dashboard panels

Panel	Purpose	Data Types	Example
1. Data Tables	Viewing raw data	Raw sign-in logs, coded logs, event logs, user/application/device logs	Tables described in text
2. User Health	At a glance view of global user performance	Graphs: time to authenticate, locked-out users, success rate; Charts: error counts by type, portion of events containing errors	Figure 5.7
3. App Health	At a glance view of global application performance	Tables: lowest success rate, highest hacking errors, highest configuration errors; Charts: top 10 application event counts, top 10 applications total time authenticating	Figure 5.8
4. Device Health	At a glance view of global device statistics	Graphs: unique devices per day, devices with hacking errors	Not fully implemented
5. Reports	Reports for users, apps, and devices	Equivalent to the "Health" panels, but specific to a subset of users/applications/devices	NA
6. Alert Summary	Summary of an alert with relevant data	Graphs: success rate, errors, app history; Charts: MFA history, login time of day, location history; Tables: event logs, raw logs	Figure 5.9

accounts.[2] Figure 5.7 provides an example.

Application Logs: Application logs are derived from event logs and present aggregations over a particular application or set of applications. These logs provide several strategic opportunities, including visibility into the unique applications to which users attempted to authenticate. The typical organization introduces over 300 new services every month; in some industries, such as telecommunications and insurance, this number increases to 1,000 new services per month [34]. The application-specific log summary allows organizations to quickly identify lapsed or unused authentication targets that can be decommissioned or otherwise protected to reduce their attack surfaces. In a 30-day sample of authentication data, 22% of the targeted applications had zero successful authentications, and can be quickly identified using the Application Health panel of the dashboard as discussed in section 5.3.3.

[2] We say a user is locked out when they have two or more consecutive failed authentication events with over 12 hours unauthenticated

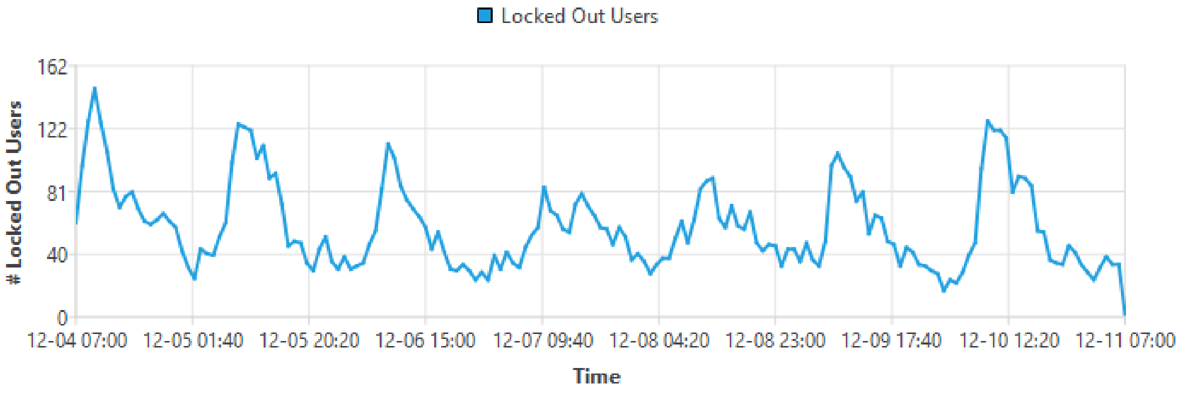


Figure 5.7: Example locked-out graph

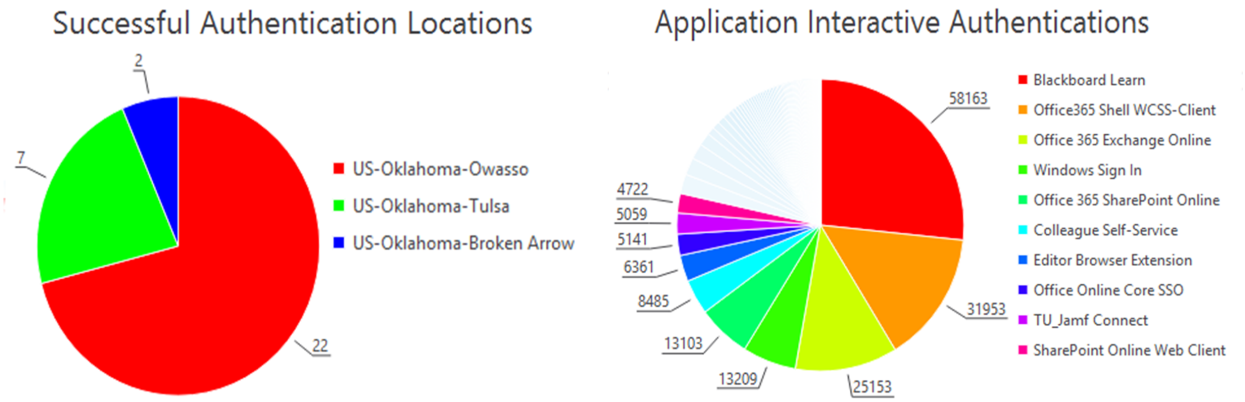


Figure 5.8: Example charts: A. Successful auth locations B. Top applications

Device Logs: Device logs share the same basic design as application and user logs, aggregating authentication data specific to devices over specified time periods. Their primary use case is investigating device-specific alerts, such as indications of unwanted software or unusual network activity. Charts and graphs of a device's recent authentication history including number of users, average session time, authentication failures, and forms of second factor authentication enable an analyst to quickly search for behavior indicating compromise.

Alert Logs: The final type of data used in the SOC dashboard is the alert logs, which analysts review when clearing alerts in web portals such as Microsoft Defender.

Reports and Health Summaries: In addition to panels where the above data types can be generated as needed, the dashboard includes “health” and “report” panels for the application, user, and device logs. A health panel includes several tables and charts, constructed from the prior log types, to enable a snapshot assessment of the state of authentication for users, devices, or applications, as described in Table 5.1. See Figure 5.8 for an example application chart.[3]

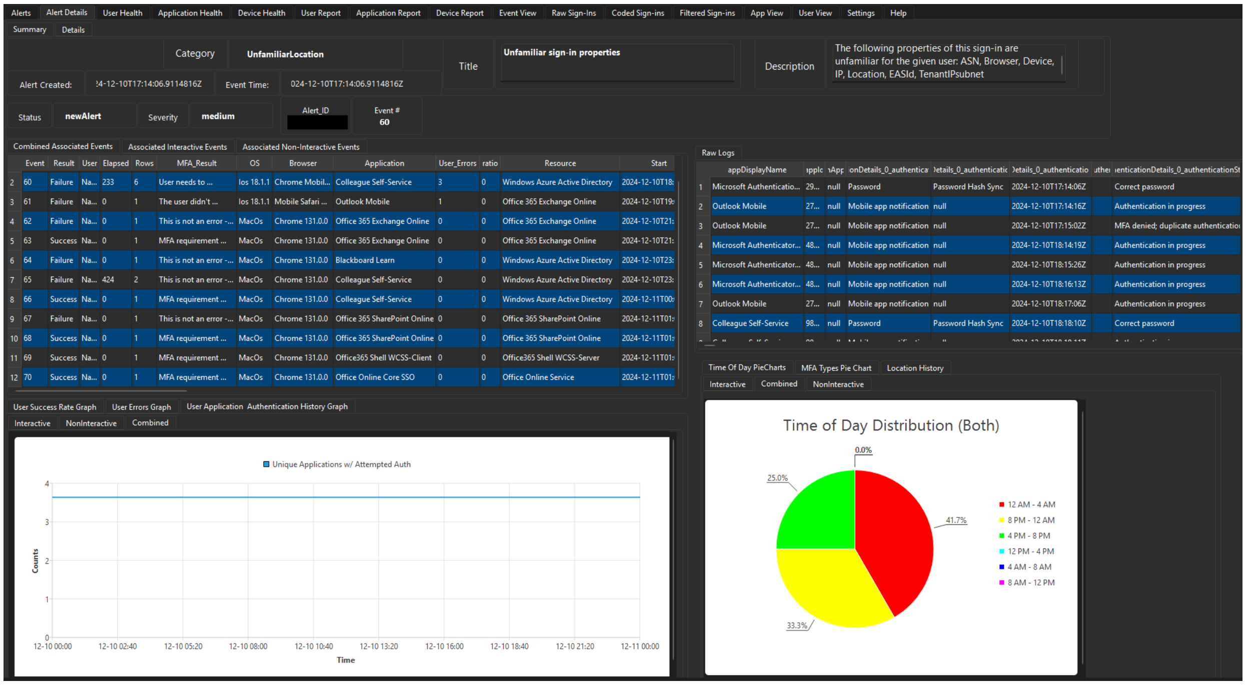
The report panels are similar to the health panels, but default to longer timelines and allow the analyst to select a subset 1–n of users, applications, or devices on which to generate the charts and graphs. These panels are designed to quickly provide a historical view of a given asset that can be shared or archived for investigation or audit purposes. The primary difference between the report and health panels is that the health panels are global, while the report panels are specific to one or more users, applications, or devices.

Alert Summaries: The alert summary panel leverages available data to consolidate the most relevant information for an alert into a single space, enabling quick and comprehensive investigations. It includes basic details of the alert, tables of event and raw logs, and various charts and graphs displaying success rates, error rates, application usage, authentication times, and more, depending on the alert type. An example of this can be seen in Figure 5.9 above.

5.4 Implementation and Usage

This section presents example investigations complete with the graphs, charts, and data utilized. Investigations are then generalized into two core workflows that can address the majority of alerts.

[3] Charts and tables are generated using a default time range that can be adjusted as needed.



Category	UnfamiliarLocation	Title	Unfamiliar sign-in properties
Alert Created:	:4-12-10T17:14:06.9114816Z	Event Time:	024-12-10T17:14:06.9114816Z
Status	newAlert	Severity	medium
Alert_ID	[redacted]	Event #	60
Description	The following properties of this sign-in are unfamiliar for the given user: ASN, Browser, Device, IP, Location, EASId, TenantIPsubnet		

Combined Associated Events
Associated Interactive Events
Associated Non-Interactive Events

	Event	Result	User	Elapsed	Rows	MFA_Result	OS	Browser	Application	User_Errors	ratio	Resource	Start
2	60	Failure	Na...	233	6	User needs to ...	Ios 18.1.1	Chrome Mobil...	Colleague Self-Service	3	0	Windows Azure Active Directory	2024-12-10T18:
3	61	Failure	Na...	0	1	The user didn't ...	Ios 18.1.1	Mobile Safari ...	Outlook Mobile	1	0	Office 365 Exchange Online	2024-12-10T19:
4	62	Failure	Na...	0	1	This is not an error -...	MacOs	Chrome 131.0.0	Office 365 Exchange Online	0	0	Office 365 Exchange Online	2024-12-10T21:
5	63	Success	Na...	0	1	MFA requirement ...	MacOs	Chrome 131.0.0	Office 365 Exchange Online	0	0	Office 365 Exchange Online	2024-12-10T21:
6	64	Failure	Na...	0	1	This is not an error -...	MacOs	Chrome 131.0.0	Blackboard Learn	0	0	Windows Azure Active Directory	2024-12-10T23:
7	65	Failure	Na...	424	2	This is not an error -...	MacOs	Chrome 131.0.0	Colleague Self-Service	0	0	Windows Azure Active Directory	2024-12-10T23:
8	66	Success	Na...	0	1	MFA requirement ...	MacOs	Chrome 131.0.0	Colleague Self-Service	0	0	Windows Azure Active Directory	2024-12-11T00:
9	67	Failure	Na...	0	1	This is not an error -...	MacOs	Chrome 131.0.0	Office 365 SharePoint Online	0	0	Office 365 SharePoint Online	2024-12-11T01:
10	68	Success	Na...	0	1	MFA requirement ...	MacOs	Chrome 131.0.0	Office 365 SharePoint Online	0	0	Office 365 SharePoint Online	2024-12-11T01:
11	69	Success	Na...	0	1	MFA requirement ...	MacOs	Chrome 131.0.0	Office365 Shell WCSS-Client	0	0	Office365 Shell WCSS-Server	2024-12-11T01:
12	70	Success	Na...	0	1	MFA requirement ...	MacOs	Chrome 131.0.0	Office Online Core SSO	0	0	Office Online Service	2024-12-11T01:

	appDisplayName	applc	ıApp	ionDetails_0_authentica	Details_0_authenticatic	Details_0_authenticatio	uther	ıenticationDetails_0_authenticationSt
1	Microsoft Authenticatio...	29...	null	Password	Password Hash Sync	2024-12-10T17:14:06Z		Correct password
2	Outlook Mobile	27...	null	Mobile app notification	null	2024-12-10T17:14:16Z		Authentication in progress
3	Outlook Mobile	27...	null	Mobile app notification	null	2024-12-10T17:15:02Z		MFA denied; duplicate authenticatio
4	Microsoft Authenticator...	48...	null	Mobile app notification	null	2024-12-10T18:14:19Z		Authentication in progress
5	Microsoft Authenticator...	48...	null	Mobile app notification	null	2024-12-10T18:15:26Z		Authentication in progress
6	Microsoft Authenticator...	48...	null	Mobile app notification	null	2024-12-10T18:16:13Z		Authentication in progress
7	Outlook Mobile	27...	null	Mobile app notification	null	2024-12-10T18:17:06Z		Authentication in progress
8	Colleague Self-Service	98...	null	Password	Password Hash Sync	2024-12-10T18:18:10Z		Correct password

Figure 5.9: Dashboard prototype example

5.4.1 Case 1: Unfamiliar Location

The severity of the "Unfamiliar Location" alerts range from low to high, as captured by Table 5.4. We chose a high-severity instance as an example of an alert that would receive investigation priority. The "Unfamiliar Location" category is selected from the filter options and a high severity example is selected by double-click. The alert summary is populated similarly to Figure 5.9, and we began evaluating the inciting event. We define an inciting event as:

> The derived single-row event summary representing 1-n rows of data that includes the single raw authentication log the alert is associated with.

The event attributes were examined, revealing a disparity between the locations of the inciting event and the surrounding events—specifically, an authentication attempt from "AU", Australia, when the user was based in the central United States. Observing this discrepancy, we proceeded with a more comprehensive evaluation of the location relative to the user's history and consulted the location chart. The left chart in Figure 5.8 displays the distribu-

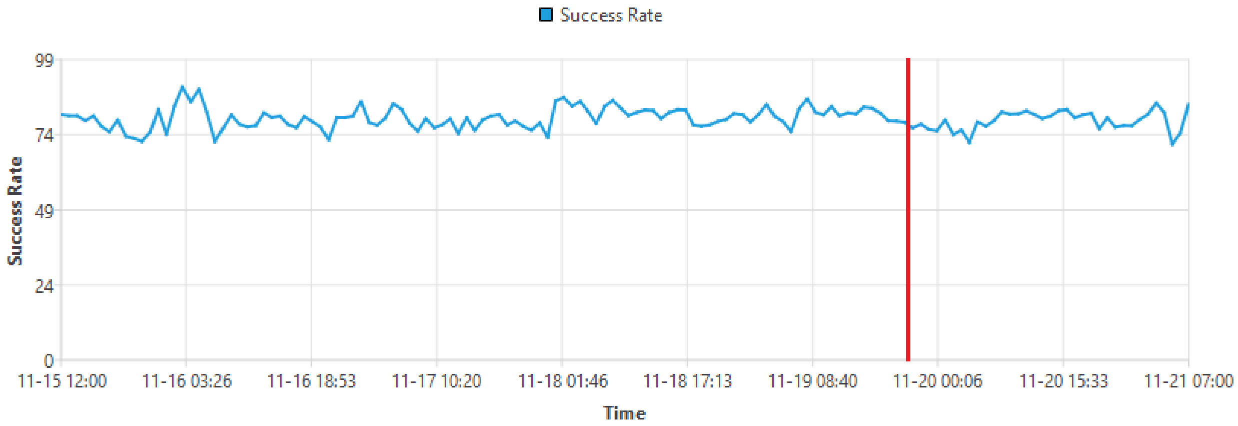


Figure 5.10: Global Success Rate around Alert

tion of locations for successful authentications, showing that the user had never successfully authenticated from the location of the sign-in attempt that triggered the alert.[4] Observing a significant deviation from the user's successful authentication history, we concluded that the alert was a true positive. The entire investigation was conducted within a single tab of the dashboard tool, without requiring additional inputs to generate the referenced context.

Case 2: Password Spray Attack: OWASP defines password spray attacks as "a type of brute-force attack in which the attacker uses one password (e.g. Secure@123) against many different accounts to avoid account lockouts that would normally occur when brute-forcing a single account with many passwords" [33]. Microsoft assigned high severity ratings to both instances of the password spray alert in the sample. The two alerts, triggered by authentication attempts less than 10 minutes apart, likely indicate the same password spray attack.

The investigation of the alerts begins by clicking on either password spray alert to generate a summary. The inciting event was reviewed, revealing a failed authentication due to a configuration error, which ruled out erroneous user input. The event summary also indicated that a password was used in the authentication attempt, warranting further

[4] The same chart is available for failed authentications.

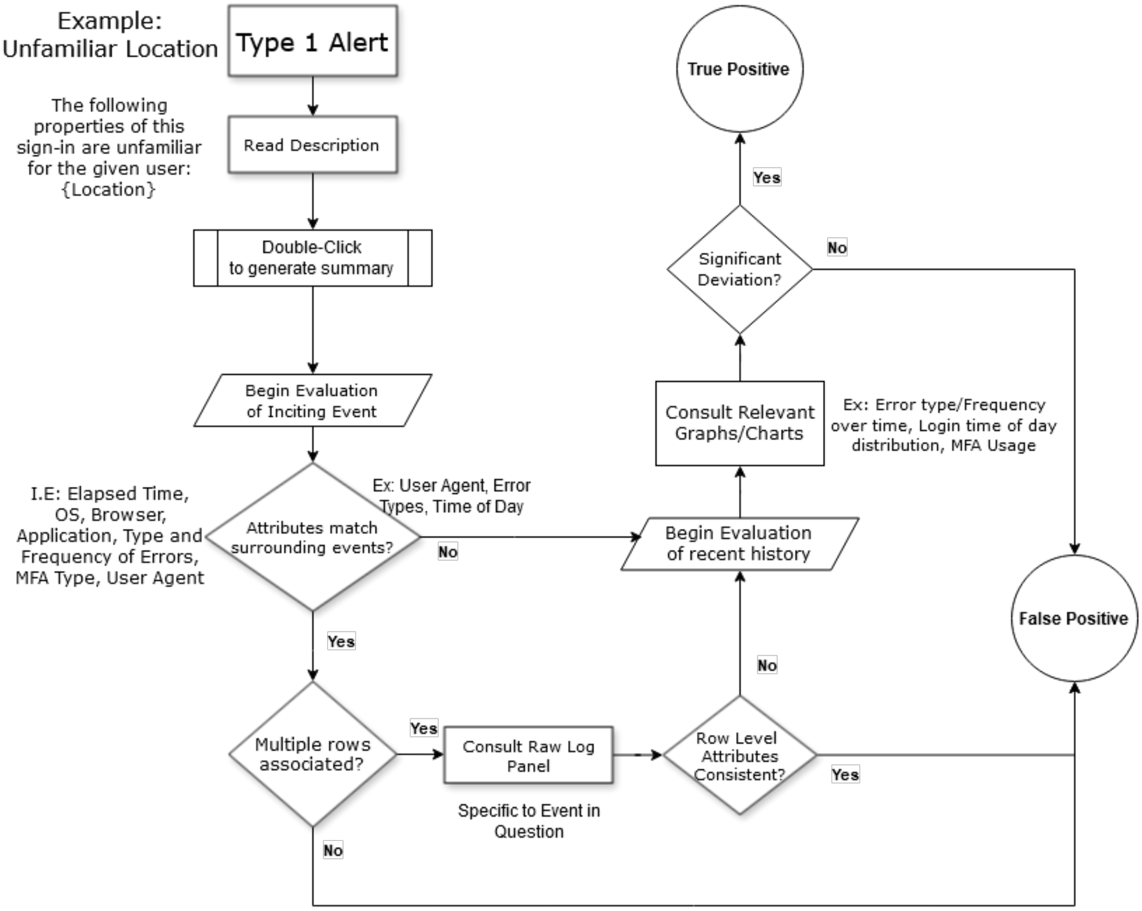

Example:
Unfamiliar Location
Type 1 Alert
The following properties of this sign-in are unfamiliar for the given user: {Location}
Read Description
Double-Click to generate summary
Begin Evaluation of Inciting Event
I.E: Elapsed Time, OS, Browser, Application, Type and Frequency of Errors, MFA Type, User Agent
Attributes match surrounding events?
Ex: User Agent, Error Types, Time of Day
No
Yes
Multiple rows associated?
Yes
Consult Raw Log Panel
Specific to Event in Question
No
Row Level Attributes Consistent?
Yes
No
Begin Evaluation of recent history
Consult Relevant Graphs/Charts
Ex: Error type/Frequency over time, Login time of day distribution, MFA Usage
Significant Deviation?
Yes
No
True Positive
False Positive

Figure 5.11: Workflow 1

evaluation. While the event partially matched the characteristics of a password spray attack, it did not align with the expected behavior; such an attack would not typically result in a configuration error, but rather show predominantly incorrect password attempts.

Next, because this type of attack targets multiple users, the global success rate and error rate graphs were consulted, as shown in Figure 5.10. The graph displays the global success rate, with a red line marking the time of the alert. Observing no drop in success rates or increases in password-related errors at the time of the alert (not shown), we concluded that the alert was a false positive. Additional due diligence can be performed by referencing the application-specific history, which includes a success rate graph providing greater granularity to ensure the attack was not localized and undetectable at the global level.

5.4.2 Workflows

We now present generalized workflows inspired by the two examples. Figure 5.11 shows the generalized workflow of the "unfamiliar location" example alert. This workflow is suitable for alerts triggered by a single action and not indicative of a pattern or multiple actions. The second workflow, generalized from Example 2, is designed for alerts involving or implying the relevance of multiple actions leading to a trigger.

Workflow 1: Upon loading the alert summary, the analyst first compares the inciting event to those immediately proceeding and following it.

The analyst evaluates the primary attributes of the inciting event, such as the applications being authenticated to, the types of second factor being used, and more. The exact attributes of interest vary depending on the alert under investigation. The analyst then determines whether the attributes are consistent with the user's observed authentication history. If they match, the alert is typically identified as a false positive. If the inciting event encompasses multiple rows of raw data, the analyst consults the "Raw" tab which is pre-populated with the original data rows associated with the event. The raw data available here has a greater number of attributes than the equivalent raw data normally consulted in the authentication log portal, giving the analyst enhanced granularity when greater scrutiny is required. This can help the analyst confirm that each individual action surrounding the alert is legitimate. If the inciting event does not represent multiple rows of raw data, analysts may conclude the alert is a false positive.

If the attributes of the inciting event do not match those in the user's surrounding event history, the analyst evaluates the inciting event in relation to the user's broader history. The exact charts and graphs consulted will vary with the specific alert being investigated. When an alert is triggered by a suspicious failure, the graphs of error types and success

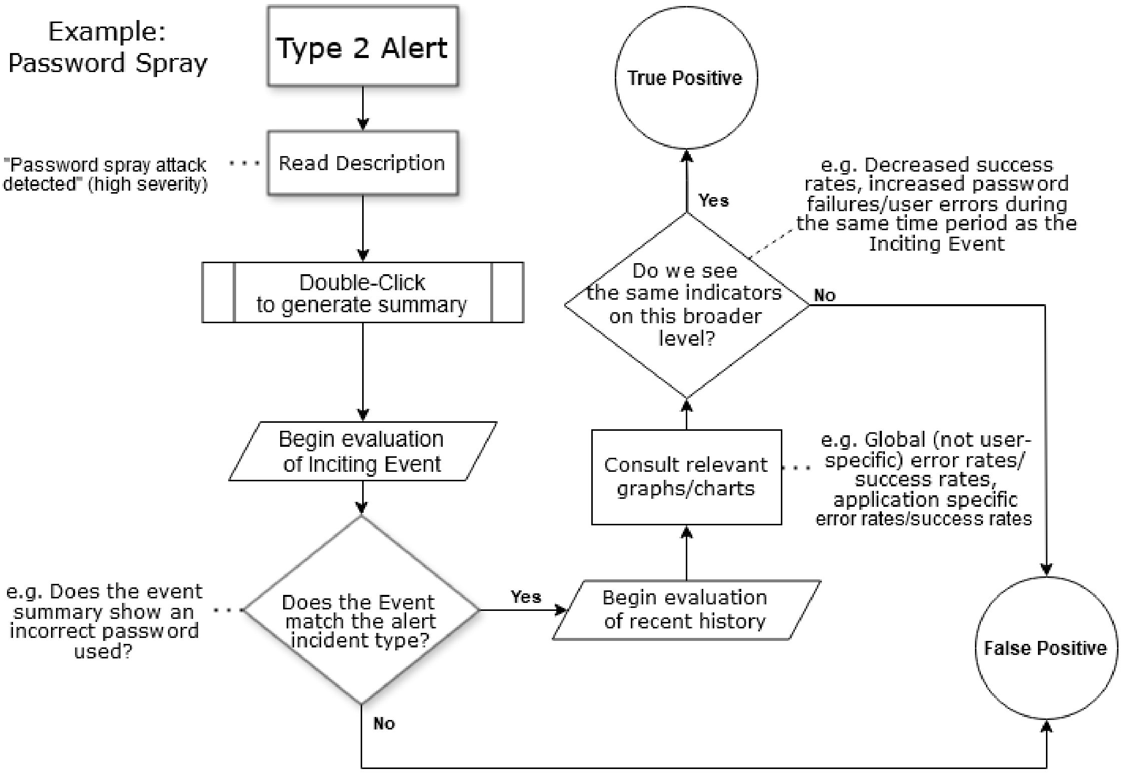

Example:
Password Spray
Type 2 Alert
"Password spray attack detected" (high severity)
Read Description
Double-Click to generate summary
Begin evaluation of Inciting Event
e.g. Does the event summary show an incorrect password used?
Does the Event match the alert incident type?
Yes
No
Begin evaluation of recent history
Consult relevant graphs/charts
e.g. Global (not user-specific) error rates/ success rates, application specific error rates/success rates
Do we see the same indicators on this broader level?
Yes
No
e.g. Decreased success rates, increased password failures/user errors during the same time period as the Inciting Event
True Positive
False Positive

Figure 5.12: Type 2 alert workflow

rates over time can be consulted. This enables the analyst to evaluate whether the type of error(s) encountered were novel in that user's history or if they had previous encounters. Changes in the error composition of failed events can be indicative of malicious or fraudulent activity. Changes in the user's success rate, number of applications being authenticated to, or number of devices used can be similar indicators.

Workflow 2: In this workflow, the alert being investigated implies risk based on multiple actions by a single user or interactions from multiple users. For a single user, this can involve repeated authentication attempts or authentications coming from different locations within a timeframe too short for the user to have traveled the distance between them. For an alert with a multi-user trigger, this could be something like a password spray attack.

The investigation begins by double-clicking the alert on the main display and reviewing the inciting event on the generated alert summary page. In this workflow, the evaluation focuses less on specific raw data attributes and more on a holistic view of the interaction. For the example in Figure 5.12, which involves a password spray attack, the event is first checked for incorrect password usage. If the inciting event does not match the expected behavior for the alert, a false positive determination may be made.

If the event closely matches the alerted behavior, the analyst proceeds to the next step: evaluating recent history. At this stage, the analyst may expand the scope beyond a single user and consults graphs and charts showing metrics relevant to the alert category; in the previous example, the success rates and error rates across the global set of users. If global metrics unchanged in the time around the alert, a more specific set of charts and graphs can be generated for the application authenticated to in the inciting event. If the charts and graphs show the effects that the inciting incident suggested, the analyst may make a true positive determination. If the expected behavior for the alert under investigation is limited to the inciting event and does not persist within that same user, as in a brute force attack, or across the broader user-base, as in a password spray attack, the analyst may determine that the alert is a false-positive.

5.5 Event Data Coverage of Alerts

In this section we review the types of alerts present in the Microsoft Defender alert data queried from the Graph API and describe the relevant raw and derived data that the dashboard utilizes for enhanced alert review. Alerts are grouped by the Microsoft assigned, organizationally managed alert category.

5.5.1 Review of Alert Types

Table 5.2 lists the primary categories of alerts present in a 90 day sample of security alerts from a single alert source, Microsoft Defender, at the author's university. The table provides the abbreviated name of the alert category, the number of subtypes of alert in

Table 5.2: Alert categories

Category	**#Sub-Types**	**Severity Range**	**#Alerts**
1. AnomalousToken	1	Medium	10
2. AnonymousLogin	1	Low to Medium	159
3. CredentialAccess	1	High	9
4. DefenseEvasion	1	Informational	12
5. Discovery	1	Low to Medium	50
6. Execution	1	Informational	7
7. ImpossibleTravel	1	Low to Medium	549
8. InitialAccess	1	Low	9
9. LateralMovement	1	High	2
10. LdapSearchRecon.	1	Medium	4
11. Malware	3	Info. to Medium	74
12. MCASALERT	6	Informational	222
13. PassTheTicket	1	Medium	74
14. PasswordSpray	1	High	2
15. Priveledge Escalation	2	Medium	3
16. Ransomware	2	Medium to High	13
17. SuspiciousActivity	1	Medium	17
18. ThreatManagement	5	Info. to High	466
19. UnfamiliarLocation	1	Low to High	1436
20. UnwantedSoftware	1	Low to High	10
Total Alert Count: 3067			

Alert categories with 1 or less alerts are omitted

that category, the range of severity associated, and the number of alerts in that category for the sample. Subtypes represent minor differences within a category, which we omit from this table as they represent no substantive difference in associated data. Note that the category names may differ between organizations, and that a category like “Execution” may represent different alerts in another organizations context. Alerts range in severity from “Informational”, a notification that does not require investigation, to “Low”, “Medium”, and “High” severity, which can include indicators of account compromise or active presence of malware.

A quick examination of Table 5.2 reveals a wide variety of alerts with varying severity and frequency. With a total alert count of 3,067, this averages out to 34.1 per day. Noting that many of these may be ”informational”, which do not require investigation, we

Table 5.3: Dashboard data presence for alert types

Data Type	**Alert Category**																	
	Anomalous Token	Anonymous Login	Credential Access	Defense Evasion	Discovery	Execution	Impossible Travel	Initial Access	Lateral Movement	LDAP Alerts	Malware	MCAS Alerts	Pass The Ticket	Password Spray	Ransomware	Suspicious Activity	Threat Management	UnfamiliarLocation
Raw Logs	x	x					x							x				x
Row Codes	x	x					x							x				x
Event Logs							x							x				x
User Charts	x	x	x	x										x				x
App Charts		x	x	x	x	x	x	x	x	x	x	x	x			x	x	x
Investigation State	C	C	P	P	P	P	C	P	N	N	N	P	N	C	P	N	N	C

C = Complete, P = Partial, N = None; Categories limited to those with multiple instances

tally the number of alerts for each category of each severity. The total number of alerts requiring investigation is 2,569, including 80 high-severity, 366 medium-severity, and 2,123 low-severity alerts. On a daily basis, this averages to ≤ 1 high, 4.1 medium, and 23.6 low severity alerts, for a total of 28.5 per day.

5.5.2 Dashboard Investigation Completeness

Table 5.3 presents a matrix indicating each type of data utilized by the dashboard (non-exhaustive) that is associated with each alert category. Each row is a different data type and each column is an alert category, with the final row "Investigation Relevant" an indicator that the head of the SOC confirmed the available data can be used to completely or partially investigate the alert as specified. A "C" in this row indicates complete investigation coverage, an "N" indicates none, and a "P" indicates partial coverage that would need to be supplemented with another tool for a thorough investigation. The raw data category covers the raw attributes present in the sign-in logs queried from the Graph API. The derived categories include the attributes added through coding and aggregation, and the rest of

Table 5.4: Dashboard alert severity coverage

Category	**High**	**Medium**	**Low**	**Info.**	**Total**
Anomalous Token	0	10	0	0	10
Anonymous Login	0	10	149	0	159
Impossible Travel	0	14	535	0	549
Password Spray	2	0	0	0	2
Unfamiliar Location	40	272	1124	0	1436
Complete Investigation	42	306	1808	0	2156
Partial Investigation	35	43	273	64	415
No Investigation	3	16	48	438	505
Total	80	365	2129	502	3076
Complete Investigation %	53%	84%	85%	0%	70%
Partial Investigation %	44%	12%	13%	13%	13%
No Investigation %	4%	4%	2%	87%	16%

the data types are aggregations of those derived attributes. After identifying the data of interest for each type of alert, we design an alert summary page that can be used for relevant categories.

5.5.3 Alert Coverage

Table 5.4 presents a breakdown of the number of alerts at each severity level, noting which can be investigated by the dashboard. The alerts are summed to calculate the dashboard coverage by category, totaled by level of investigation, and the percentage calculated. In our 90-day sample of 3,067 alerts, we found that 2,156 (70%) could be fully investigated using dashboard data, while 13% could be partially investigated. Excluding ’informational’ severity alerts, which do not require investigation, 84% of alerts can be fully investigated using the dashboard, and 13% can be partially investigated.

5.6 Deployment

After incorporating feedback into the dashboard implementation, the tool was deployed to the university SOC with approval from CISO Michael Epperson. Following the deployment, live training sessions were conducted for the analysts to ensure a comprehensive understanding of the dashboard's functionalities. The tool was made readily accessible in conjunction with other tools utilized on a daily basis within the university SOC. Analysts utilized the dashboard proactively to identify disruptions or problematic users, and reactively to investigate flagged alerts. Subsequent to and throughout deployment, informal interviews were conducted with the analyst, which we discuss next.

5.6.1 Feedback

8 analysts and 1 manager tested and gave feedback on the dashboard. Feedback was solicited focusing on the following items using open-ended prompts.

- Interpretability: Did the analyst find the provided data set and metrics to be readily interpretable?
- Frequency of Use: How often did the analyst use the tool?
- Learning Curve: Did the analyst experience a learning curve with the tool? How long did it take to feel comfortable using it, if ever?
- Utility: What utility did the analyst find in the provided data and metrics?
- Form: How did the analyst feel about the form the data was provided in?
- Functions: Are there additional related functions the analyst feels could be added to the tool?

Finally, a formal interview with the SOC leadership was conducted to collect administrative feedback.

5.6.2 Feedback and Changes

Feedback from SOC analysts was rapidly implemented throughout the deployment period. Oral feedback centered on initial usability, leading to deployment of several quality of life patches during testing. Analysts requested multiple updates ranging from small bug fixes (display cut off from viewport), to larger feature changes (inclusion of specific variables in various data forms). One such change came from an analyst who had difficulty visually searching the alert logs for an alert on a specific user. In the source data, the user's name and email address are not included by default, instead including an alpha-numeric azure user id. In this example, a new subroutine was created to add the corresponding username and email address to the alert logs upon download, and the new attributes were added to the output.

Second, as VPN's play an expanding role in alert investigations, the same user reported the importance of knowing both the primary IP address and the range of IP addresses associated with a given authentication event. A new event attribute was created to track each unique IP address associated with the authentication, and the attribute was added to the output as suggested.

Several future improvements were suggested by analysts. Improvements focused on presenting additional context around a user's past behavior, such as average login times and typical IP ranges. Simple heatmaps or graphs showing login volume over time and IP geo-distribution were proposed. Other suggestions included advanced flagging of suspicious event sequences, such as repeated lockouts followed by success, and integrating a threat intel feed to flag bad IPs and domains.

5.7 Discussion

Feedback from analysts indicates strong alignment between the tool's intended design and its perceived utility among domain users. Analysts acknowledged that the system meets a growing need for proactive security monitoring in environments characterized by high user mobility, heterogeneous applications, and evolving policy constraints. Reviewers responded favorably to the system's core emphasis on alert triage and contextualized decision-making. Specifically, the ability to pivot from high-level anomalies to detailed user or application views was seen as critical for maintaining operational awareness during both routine monitoring and emergent security investigations. The integration of temporal context, dynamic reports, and user-centric alert histories were highlighted as particularly useful in supporting forensic analysis in alert investigations.

CHAPTER 6
CONCLUSION

This dissertation aims to enhance the interpretability and utility of authentication logs through a novel event-based aggregation methodology, and to demonstrate how these enriched logs can yield deeper insights into multifactor authentication (MFA) performance, user experience, and Security Operations Center (SOC) workflows. Traditional raw authentication logs are notoriously noisy and difficult to interpret, as numerous backend entries can obscure the true, user-facing steps of each login attempt. By proposing a process to systematically collapse raw entries into coherent events, this work provides a clearer view of user authentication behavior, one that captures not only successes or failures but also the kinds of errors and delays that occur along the way.

6.1 Summary

In Chapter 2 we described how raw Azure AD (now Entra ID) logs were transformed into user-centric events. Through row coding and aggregation, single rows in the final dataset represent complete attempts: when a user first tries to authenticate (entering credentials, responding to MFA prompts) through either success, failure, or abandonment. This helps eliminate redundant or backend-only data, allowing researchers and practitioners to focus on the genuine user experience.

Chapter 3 presented an empirical study quantifying the added burden,or “costs”, that certain MFA policy changes can impose. Drawing on event-level metrics such as the number of authentication failures and the length of time users remain “locked out” (Time Away) after failing a login, the analysis revealed that a more secure (but more cumbersome) mobile-based MFA method measurably increased these costs. Fixed-effects regression modeling confirmed that after the policy change, users experienced statistically higher failure counts and longer Time Away, underscoring the need for careful balancing of security with usability.

Chapter 4 led an investigation into how users' psychological traits can influence MFA outcomes. By linking Security-Related Stress (SRS), self-efficacy, and related constructs to real authentication logs, this dissertation sheds light on which psychometric dimensions correlate most strongly with MFA performance. This chapter demonstrated that even technical processes like login events can be meaningfully impacted by individual attitudes and stressors, opening the door for more human-centered interventions in cybersecurity.

Finally, Chapter 5 surveyed security-relevant metrics and demonstrated how the event-based logs can improve SOC workflows and alert triage. After developing a real-time dashboard that highlights user lockouts, failing applications, and suspicious login attempts, the system was deployed in a university SOC. Feedback from SOC analysts suggested that aggregated event logs helped them quickly identify urgent user issues, reduce investigative time, and spot "lapsed" applications that might go unmonitored with conventional logs. This chapter illustrated the operational impact of turning raw logs into interpretable event data and showcased the ways that even smaller security teams can integrate such analytics into daily processes.

Through these four major chapters of original analysis, this dissertation highlights both research contributions (a generalizable event methodology, new empirical evidence on policy burdens, quantification of psychometric correlates) and practical contributions (a working SOC dashboard prototype, immediate gains in alert interpretability).

6.2 Key Findings and Contributions

Chapter 2 delineated the process by which raw logs, frequently amounting to dozens for each user login, can be consolidated into succinct events. By categorizing each log row with "row codes" (user error, configuration error, standard interrupts, etc.) and then combining them, the final event format is both much smaller (fewer rows) and more meaningful (each row now represents a complete user-centric authentication attempt). This methodology is readily adaptable to other authentication systems beyond Azure AD, provided there are enough log-level attributes to track key user-facing steps.

The analysis in Chapter 3 highlighted that corporate or institutional moves toward stricter MFA protocols can inadvertently increase friction for legitimate users. The dissertation introduced and validated Time Away (the period between a failed login and the user's next attempt) as a novel metric that captures user disengagement or lockout after errors. Results confirmed that when the university upgraded its MFA procedure (from simple push notifications to a two-digit confirmation code), users' failure rates and Time Away intervals rose significantly. These findings suggest that how MFA is implemented (e.g., which second factor method is enforced) can measurably affect user productivity and frustration, in addition to improving security.

Chapter 4 established that user performance in MFA is not solely a technical matter but can be influenced by psychological constructs. In particular, users with high Security-Related Stress (SRS) tended to experience more authentication failures and longer Time Away than less-stressed peers, amplifying the challenge of adopting additional security measures. By combining surveys on stress and self-efficacy with event logs, this research bridges the gap between human factors and technical performance, suggesting that addressing user stress (through training, better user support, or more intuitive MFA flows) can tangibly improve security outcomes.

In Chapter 5, the dissertation demonstrated how the aggregated event log can facilitate rapid investigations. The newly developed SOC dashboard highlights failing applications, locked-out users, and unusual login patterns, helping analysts prioritize critical alerts while reducing "noise." Feedback from the pilot deployment produced several material improvements to the prototype design while confirming percieved benefit by analysts. Aggregation and contextualization of alert-related logs was the primary utility voiced in feedback, the location and error history data were specifically noted as useful.

Together, these findings confirm that derived authentication events, properly aggregated from the raw logs, yield valuable user-level metrics (Time Away, number of errors, etc.), can be linked to user-level psychological data, and hold strong operational benefits for security staff. In essence, the dissertation's main contribution is demonstrating that we can systematically transform raw, noisy authentication logs into aggregated records of user experiences—and that this translation drives both novel academic research and practical tools in cybersecurity.

6.3 Limitations

While the event-based approach and its applications showed promise, several limitations should be noted:

All data collections and dashboard deployment were conducted at a single university with its own unique IT environment, user base, and security policies. While the broad approach is generalizable, some usage patterns (e.g., academic staff vs. corporate workforce, frequent remote logins, varied devices) may differ in other organizations. Additional case studies could confirm the replicability of these findings. The dissertation captured a snapshot in time around a specific policy change (i.e., the introduction of more secure mobile MFA). If subsequent MFA evolutions or enterprise shifts occur, user costs might fluctuate. Longitudinal studies could reveal more nuanced adoption cycles or "learning effects," such as whether user frustration tapers off with consistent usage of the new method.

The relationships uncovered between constructs like SRS and MFA performance are correlational. Although strong correlations exist, more work is needed to tease out causal pathways. A large-scale, multi-institution design that randomizes or controls for interventions (e.g., stress-management training) could better clarify the directionality of these relationships.

While Chapter 5 offered a dashboard prototype and initial feedback from analysts, a broader set of user studies, perhaps measuring time saved or comparing the efficacy of event-based dashboards to legacy log-based workflows, would strengthen conclusions on the exact degree of operational benefit. Despite these constraints, the research provides robust evidence that aggregated events improve both academic insight (quantifying user costs, linking performance to psychometrics) and security operations (enabling more actionable dashboards).

6.4 Generalizability

In Chapter 2 we presented a method to distill raw authentication logs into user-centric authentication events. While this implementation is specific to the University of Tulsa and its enterprise environment, the process was designed to be adaptable for use in different environments. Specifically, the event methodology relies on the ability to do a few basic things:

- Capture sign-in logs: Have access to user authentication logs for your organizations identity and access management (IAM) platform, in this case, Microsoft's Entra ID. Logs must be atomic in nature and include authentication details indicating password use, second factor use, any associated error, and the authentication result.
- Classify errors: Capture and analyze each error in a large sample of authentication logs to categorize errors. This is essential to capture the relevancy of a given error to a user session. Errors must be grouped into unique error row codes.
- Capture authentication details: Authentication details are similarly mapped to unique row codes and grouped by authentication forms used, such as token, password, or second factor.

These attributes may be reasonably expected to be present in any industry Identity and Access Management (IAM) platform, as they are fundamental to the functionality of the access management system itself. An organization without a SSO implementation would need to modify the aggregation step in order to ensure that authentication attempts to different services in a similar time-frame are separated, but that would be a trivial modification.

In Chapter 3 we measured the cost of a change to MFA implementation at our university. The metrics we use, such as Elapsed Time and Time Away, are directly transferable to other organizations; however, we do not expect our specific findings to translate directly to a more corporate enterprise environment, as user behavior itself is likely different when it comes to authenticating promptly. University users may be able to "afford" to spend more time unauthenticated than their corporate peers, so the measure of time lost due to authentication issues may differ between populations. Indeed, as this is a novel measurement, we encourage other universities and corporations to pursue research to evaluate the cost associated with MFA in their own organizations.

Chapter 4 explored the relationship between user's self-reported psychological traits and their resulting MFA performance and outcomes. It is important to remind the reader that this analysis compares longitudinal authentication event data derived from Azure authentication logs starting in November 2021 with survey data collected in late 2020. Users' Security Related Stress, New General Self-Efficacy, and Security Related Self-Efficacy may have changed in that time, and due to the time lag we don't capture any state-like effects. We also note that the Uncertainty and Complexity constructs may capture more state-like situational information than Overload or the efficacy measures, contributing to their lack of significance in the analysis. Next, this study uses a convenience sample of 111 students and faculty associated with several business and technology courses at the authors' university. In this academic context, effect sizes for measures of time cost to the user and organization such as Time Away and Days Locked Out may be inflated compared to a population with more rigid time restrictions on work. With these considerations, we emphasize this study was a first test of the relationship between users reports about their own emotion states and their measured performance later in the future. We find the presence of statistically significant relationships between these measures to be an exciting discovery that opens the door for future work. Hence, this study is generalizable in its findings of a significant relationship, but does not represent a thorough assessment of the effects themselves.

Finally, in Chapter 5, we embodied the potential of the event methodology in a dashboard for SOC analysts designed to enhance proactive monitoring and alert investigations. The generalizability of this chapter can be broken into two halves: the ability to generalize the event methodology to other log sources, as discussed previously in this section, and the ability to generalize the utility of the dashboard to an arbitrary SOC. For the first half, I defer to the subsection above, and for the second, we assess it as follows. The stated primary purpose is to assist in alert investigations. It can be reasonably inferred that all SOC's will have some non-trivial portion of their alerts be triggered around authentication abnormalities, therefor, we can reasonably expect that any organization with a SOC that can create the event dataset will be able to utilize the SOC dashboard similarly. The exact displays and metrics may be customized, but the framework and the value proposition are unchanged.

6.5 Future Work

In light of the preceding discussion regarding the generalizability of this work, we propose several avenues for future research and practical applications:

Although the methodology in Chapter 2 was demonstrated using Azure AD logs, many organizations rely on other identity providers (Okta, Duo, Ping Identity, etc.). While we expect the method to be generalizable, validating whether row coding and event aggregation translate seamlessly to these environments would both broaden the approach's applicability and highlight any platform-specific nuances.

Chapter 3 demonstrated the cost of a policy change within a single academic organization. A next logical step would be to partner with multiple universities or companies to compare the impact of similar policy changes, as well as examine the variance in costs between organization. This could further validate (or refine) the relationships discovered around user stress, error frequency, and time away. Future work could extend this research through replication in a corporate population, where the variables capturing time cost are more meaningful.

Chapter 4 suggests that psychological factors influence MFA outcomes, but no interventions were tested to modulate that relationship. Future experiments could provide users with targeted training to see if measured performance improves, or if certain user groups (e.g., older adults, new employees) benefit disproportionately from such interventions. Similarly, capturing state-like affects through repeated measurements could further elucidate the relationship between psychology and MFA outcomes.

Future work should explore additional constructs for inclusion; we recommend considering field dependency vs. independence as a promising addition for predicting performance with digital interfaces often used for security controls[6]. Lastly, future work should consider recruiting a more diverse or representative group of participants, and specifically investigate the puzzling non-linear relationships between self-efficacy and performance metrics as observed in our analysis.

From Chapter 5, while focusing on authentication logs captures a critical part of user interaction, relevance of other logs such as firewall data, endpoint detection logs, and application usage records remain largely unexplored. Combining the event approach with these logs may yield a more holistic picture of user activity and anomalies (e.g., correlating suspicious file access with repeated MFA failures).

The initial dashboard demonstration in Chapter 5 can be extended by implementing more advanced analytics (e.g., anomaly detection, machine learning predictions of future failures) and user-friendly features such as recommended mitigations and investigation decisions. Testing these enhancements in controlled field experiments would further quantify improvements in analyst efficiency.

6.6 Closing Remarks

This dissertation has demonstrated that something as routine as authentication logs. Often overlooked as a mere byproduct of daily user logins—can, with the right methodology, they can be transformed into a valuable lens on both security and human factors. By consolidating raw data into meaningful events, we uncovered patterns that speak to the effectiveness of multifactor authentication policies, the challenges faced by stressed or uncertain users, and the operational needs of SOC analysts. This bridging of technical log analysis with user-centric research is crucial in modern cybersecurity, where human behavior remains a pivotal link in the security chain.

Looking ahead, the confluence of large-scale authentication data, evolving user behaviors, and dynamic threats will only intensify. Harnessing data-driven methods—like the event-based approach—and understanding the role of user psychology in security performance offer a path toward more resilient, user-friendly systems. With continued refinement, these insights and tools can help organizations move beyond purely reactive measures and toward proactive, empathetic strategies that protect systems while acknowledging the legitimate frictions of robust security controls.

BIBLIOGRAPHY

[1] Jacob Abbott and Sameer Patil. How mandatory second factor affects the authentication user experience. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, April 2020. https://dl.acm.org/doi/10.1145/3313831.3376457.

[2] Akshay Aggarwal and Ram Kumar Dhurkari. Association between stress and information security policy non-compliance behavior: A meta-analysis. *Computers & Security*, 124:102991, January 2023.

[3] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *Proceedings of the 31st USENIX Security Symposium*, pages 2783–2800, 2022. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi.

[4] Clara Ament and Steffi Haag. How information security requirements stress employees. *Thirty Seventh International Conference on Information Systems*, 2016.

[5] Joshua Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 2009.

[6] Marios Belk, Christos Fidas, Panagiotis Germanakos, and George Samaras. The interplay between humans, technology and user authentication: A cognitive processing perspective. *Computers in Human Behavior*, 76:184–200, November 2017.

[7] Haibo Bian, Tim Bai, Mohammad A. Salahuddin, Noura Limam, Abbas Abou Daya, and Raouf Boutaba. Uncovering lateral movement using authentication logs. *IEEE*

Transactions on Network and Service Management, 18(1):1049–1063, March 2021. Conference Name: IEEE Transactions on Network and Service Management.

[8] Nele Borgert, Luisa Jansen, Imke Böse, Jennifer Friedauer, M. Angela Sasse, and Malte Elson. Self-efficacy and security behavior: Results from a systematic review of research methods. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–32, New York, NY, USA, May 2024. Association for Computing Machinery.

[9] Gilad Chen, Stan Gully, and Dov Eden. Validation of a new general self-efficacy scale. *Organizational Research Methods - ORGAN RES METHODS*, 4, January 2001.

[10] Jessica Colnago, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Lorrie Cranor, and Nicolas Christin. It's not actually that horrible: Exploring adoption of two-factor authentication at a university. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 04 2018.

[11] Deborah R. Compeau and Christopher A. Higgins. Computer self-efficacy: development of a measure and initial test. *MIS Quarterly*, 19(2):189–211, June 1995.

[12] W. Alec Cram, Jeffrey G. Proudfoot, and John D'Arcy. When enough is enough: Investigating the antecedents and consequences of information security fatigue. *Information Systems Journal*, 31(4):521–549, July 2021.

[13] Cybersecurity and Infrastructure Security Agency. 4 things you can do to keep yourself cyber safe. Technical report, Cybersecurity and Infrastructure Security Agency, 2022. `https://www.cisa.gov/news-events/news/4-things-you-can-do-keep-yourself-cyber-safe`.

[14] John D'Arcy, Tejaswini Herath, and Mindy Shoss. Understanding employee responses to stressful information security requirements: A coping perspective. *Journal of Management Information Systems*, 31:285–318, 10 2014.

[15] John D'Arcy and Pei-Lee Teh. Predicting employee information security policy compliance on a daily basis: The interplay of security-related stress, emotions, and neutralization. *Information & Management*, 56(7):103151, November 2019.

[16] Liu et al. LOG-OFF: A novel behavior based authentication compromise detection approach. In *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, 8 2022. https://doi.org/10.1109/PST55820.2022.9851969.

[17] David Freeman, Sakshi Jain, Markus Duermuth, Battista Biggio, and Giorgio Giacinto. Who are you? a statistical approach to measuring user authenticity. In *Proceedings 2016 Network and Distributed System Security Symposium*, San Diego, CA, 2016. Internet Society. https://www.ndss-symposium.org/wp-content/uploads/2017/09/who-are-you-statistical-approach-measuring-user-authenticity.pdf.

[18] Mathieu Garchery and Michael Granitzer. Identifying and clustering users for unsupervised intrusion detection in corporate audit sessions. In *2019 IEEE International Conference on Cognitive Computing(ICCC)*, pages 19–27, Milan, Italy, July 2019. IEEE. https://ieeexplore.ieee.org/document/8816990/.

[19] Gerard George, Martine R Haas, and Alex Pentland. Big data and management, 2014.

[20] Inho Hwang and Oona Cha. Examining technostress creators and role stress as potential threats to employees' information security compliance. *Computers in Human Behavior*, 81:282–293, April 2018.

[21] Soohyun Jeon, Insoo Son, and Jinyoung Han. Understanding employee's emotional reactions to ISSP compliance: focus on frustration from security requirements. *Behaviour & Information Technology*, 42(13):2093–2110, October 2023.

[22] Nishant Kaushik. Designing MFA for humans. Technical report, IDPro Body of Knowledge 1(3), 2020. `https://bok.idpro.org/article/id/49/`.

[23] Se Young Kim, Hahyeon Park, Hongbum Kim, Joon Kim, and Kyoungwon Seo. Technostress causes cognitive overload in high-stress people: Eye tracking analysis in a virtual kiosk test. *Information Processing & Management*, 59(6):103093, November 2022.

[24] Youngsun Kwak, Seyoung Lee, Amanda Damiano, and Arun Vishwanath. Why do users not report spear phishing emails? *Telematics and Informatics*, 48:101343, May 2020.

[25] Chunghun Lee, Choong C. Lee, and Suhyun Kim. Understanding information security stress: Focusing on the type of information security compliance activity. *Computers & Security*, 59:60–70, 2016.

[26] Brian Lindauer. Insider Threat Test Dataset, 9 2020. `https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247`.

[27] Niels Lohmann. json, 11 2023. `https://json.nlohmann.me/`.

[28] Christian Maier, Sven Laumer, Jakob Wirth, and Tim Weitzel. Technostress and the hierarchical levels of personality: a two-wave study with multiple data samples. *European Journal of Information Systems*, 28(5):496–522, September 2019.

[29] Microsoft. Error documentation, 2024. `https://login.microsoftonline.com/error`.

[30] Microsoft Corporation. Cyber Signals. Technical report, Microsoft Corporation, May 2023. `https://news.microsoft.com/cyber-signals/`.

[31] Gregory D. Moody and Dennis F. Galletta. Lost in cyberspace: The impact of information scent and time constraints on stress, performance, and attitudes online. *Journal of Management Information Systems*, 32(1):192–224, November 2015.

[32] Forough Nasirpouri Shadbad and David Biros. Technostress and its influence on employee information security policy compliance. *Information Technology & People*, 35(1):119–141, January 2020.

[33] OWASP Foundation. Password spraying attack, 2024. `https://owasp.org/www-community/attacks/Password_Spraying_Attack`, Last accessed on 2024-12-16.

[34] Palo Alto Networks. Unit 42 attack surface threat report. Technical report, Palo Alto Networks, January 2023. `https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/reports/unit42-attack-surface-threat-report.pdf`.

[35] Grant Pannell and Helen Ashman. Anomaly detection over user profiles for intrusion detection. *Proceedings of the 8th Australian Information Security Mangement Conference*, Edith Cowan University, 2010. `http://ro.ecu.edu.au/ism/94`.

[36] Qt Company Ltd. Qt creator, 9 2024. `https://www.qt.io/download-open-source`.

[37] Ken Reese. Evaluating the usability of two-factor authentication. Master's thesis, Brigham Young University, 2018.

[38] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. A usability study of five two-factor authentication methods. In *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*, SOUPS'19, page 357–370, USA, 2019. USENIX Association.

[39] Joshua Reynolds, Nikita Samarin, Joseph D. Barnes, Taylor Judd, Joshua Mason, Michael Bailey, and Serge Egelman. Empirical measurement of systemic 2FA usability. In *USENIX Security Symposium*, 2020.

[40] Tripti Singh, Allen C. Johnston, John D'Arcy, and Peter D. Harms. Stress in the cybersecurity profession: a systematic review of related literature and opportunities for future research. *Organizational Cybersecurity Journal: Practice, Process and People*, 3(2):100–126, January 2023.

[41] J. J. Sonneveld. Profiling users by access behaviour using data available to a security operations center. info:eu-repo/semantics/masterThesis, University of Twente, January 2023. `https://essay.utwente.nl/94221/`.

[42] Daniel Stenberg. libcurl, 11 2024. `https://curl.se/`.

[43] Vectra AI. 2023 state of threat detection. Technical report, Vectra AI, Inc., 2023.

[44] Merrill Warkentin and Leigh Mutchler. *Behavioral Information Security Management*, chapter 54, pages 54.1–54.20. Taylor & Francis Group, January 2014.

[45] Qin Yuan, Jun Kong, Chun Liu, and Yushi Jiang. Understanding the effects of specific techno-stressors on strain and job performance: a meta-analysis of the empirical evidence. *Information Technology & People*, 38(2), January 2025. `https://doi.org/10.1108/ITP-08-2022-0639`.

[46] Nengwen Zhao, Honglin Wang, Zeyan Li, Xiao Peng, Gang Wang, Zhu Pan, Yong Wu, Zhen Feng, Xidao Wen, Wenchi Zhang, Kaixin Sui, and Dan Pei. An empirical investigation of practical log anomaly detection for online service systems. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1404–1415, August 2021. `https://dl.acm.org/doi/10.1145/3468264.3473933`.

APPENDIX A

REGRESSION TABLES

Table A.1: User Fixed Effects Regressions

Dependent Variables	**ln Time Away**	**ln Failures**
Explanatory Variables		
ln Mobile app MFAs	0.161***	0.103***
	(0.006)	(0.003)
ln Mobile app MFAs*Post	0.092***	0.040***
	(0.008)	(0.005)
ln Text message MFAs	0.027***	0.022***
	(0.005)	(0.003)
ln Text message MFAs*Post	0.015**	0.005
	(0.007)	(0.004)
ln Interrupt Errors	1.102***	0.636***
	(0.011)	(0.006)
ln Interrupt Errors*Post	0.121***	0.041***
	(0.015)	(0.008)
ln Configuration Errors	1.231***	0.682***
	(0.027)	(0.013)
ln Configuration Errors*Post	-0.268***	-0.124***
	(0.036)	(0.018)
ln Password Entries	0.116***	0.084***
	(0.010)	(0.006)
ln Password Entries*Post	0.221***	0.185***
	(0.026)	(0.015)
ln Mobile Entries	0.005	0.006***
	(0.004)	(0.002)
ln Mobile Entries*Post	0.089***	0.045***
	(0.007)	(0.004)
Adjusted R^2	0.504	0.522
Observations	210,167	210,167

Notes: Robust Standard errors are clustered at user level, (ln=natural log). ** (**) significant at 99% (95%) level.

Table A.2: Fixed Effects Regressions for different types of users

Dependent Variables	**ln Time Away**	**ln Failures**
Explanatory Variables		
ln Mobile app MFAs*Group 1	0.138***	0.089***
	(0.009)	(0.005)
ln Mobile app MFAs*Post*Group 1	0.081***	0.033***
	(0.010)	(0.006)
ln Mobile app MFAs*Group 2	0.151***	0.098***
	(0.010)	(0.006)
ln Mobile app MFAs*Post*Group 2	0.087***	0.038***
	(0.011)	(0.006)
ln Mobile app MFAs*Group 3	0.189***	0.121***
	(0.010)	(0.006)
ln Mobile app MFAs*Post*Group 3	0.107***	0.049***
	(0.011)	(0.006)
ln Text message MFAs*Group 1	0.018***	0.014***
	(0.007)	(0.004)
ln Text message MFAs*Post*Group 1	0.028***	0.014***
	(0.09)	(0.005)
ln Text message MFAs*Group 2	0.010	0.013***
	(0.07)	(0.004)
ln Text message MFAs*Post*Group 2	0.027**	0.010*
	(0.010)	(0.006)
ln Text message MFAs*Group 3	0.054***	0.039***
	(0.008)	(0.005)
ln Text message MFAs*Post*Group 3	-0.013	-0.009
	(0.011)	(0.006)
Adjusted R^2	0.504	0.509
Observations	210,167	210,167

Notes: The regressions also include controls for the natural logarithms of Interrupt Errors, Configuration Errors, Password Entries, Mobile Entries, and their interactions with the post variable.

Robust Standard errors are clustered at user level, (ln=natural log).*** (**) significant at 99% (95%) level.

APPENDIX B

HYPOTHESES RESULTS

Table B.1: Chapter 4 Hypotheses, Support Indicators, and Regression Statistics

Hypothesis	Construct	Metric	Supported	Beta	P-Value
H1a1	NGSE Low	Success Rate (-)	Yes	−0.08	0.203
H1a2	**NGSE High**	**Success Rate (+)**	**Yes**	**0.03**	**1.7e-05**
H1b1	NGSE Low	Success Rank (-)	Yes	−0.20	0.984
H1b2	**NGSE High**	**Success Rank (+)**	**Yes**	**0.00**	**3.8e-02**
H1c1	NGSE Low	Time Away (+)	Yes	0.02	0.244
H1c2	**NGSE High**	**Time Away (-)**	**Yes**	**-0.62**	**0.041**
H1d1	NGSE Low	Days Locked Out (+)	Yes	0.01	0.845
H1d2	NGSE High	Days Locked Out (-)	No	0.01	0.845
H2a1	**SRSE Low**	**Success Rate (-)**	**No**	**0.02**	**0.022**
H2a2	SRSE High	Success Rate (+)	Yes	0.06	0.303
H2b1	**SRSE Low**	**Success Rank (-)**	**No**	**0.07**	**0.020**
H2b2	SRSE High	Success Rank (+)	Yes	0.17	0.473
H2c	SRSE Low	Time Away (+)	Yes	0.31	0.391
H2c	SRSE High	Time Away (-)	No	0.16	0.483
H2d	SRSE Low	Friction (+)	No	−0.16	0.816
H2d	SRSE High	Friction (-)	Yes	−0.07	0.710
H3a	**Overload**	**Success Rate (-)**	**Yes**	**-0.06**	**8.1e-04**
H3b	Overload	Success Rank (-)	No	−0.08	0.260
H3c	Overload	Time Away (+)	Yes	0.87	0.825
H3d	Overload	Days Locked Out (+)	Yes	0.04	0.377
H3e	Overload	Friction (+)	No	−0.25	0.371
H4a	Complexity	Success Rate (+)	No	−0.03	0.328
H4b	Complexity	Success Rank (-)	Yes	−0.13	0.194
H4c	Complexity	Time Away (+)	No	−0.26	0.626
H4d	Complexity	Days Locked Out (+)	No	−0.16	0.052
H5a	Uncertainty	Success Rate (-)	Yes	−0.001	0.981
H5b	Uncertainty	Success Rank (-)	Yes	−0.03	0.603
H5c	Uncertainty	Elapsed Time (+)	No	−0.72	0.057
H5d	**Uncertainty**	**Time Away (+)**	**Yes**	**0.81**	**1.3e-02**

Note: Bold font indicates significance ($P < 0.05$)