Next Gen SOC: Automating Alert Overload

GIAC (GCDA) Gold Certification

Author: Jon-Michael Lacek, jmlacek@gmail.com
Advisor: *Russell Eubanks*

Accepted: July 6, 2021

Abstract

In every Security Operations Center (SOC) analysts continue to be flooded with alerts. As the adversary continues to develop and enhance their attack methodologies, security vendors continue to produce new and innovative ways of detecting alerts. These technologies/solutions leverage machine learning algorithms to build a baseline profile on user behavior and network traffic to alert when activity falls outside that established pattern. Unfortunately, the alerts generated from the machine learning solutions add to an already overwhelmed SOC. In addition to the growth in toolset usage, the amount of data coming in from those tools continue to grow, all while the headcount within a SOC typically does not. While traditional SOCs focus on tuning alerts to meet their organizational behaviors, this research focuses on combining detection mechanisms from various tools or cross-referencing data from the different sources in an automated fashion. By modifying the fidelity of these alerts, analysts are left with more context and actionable alerts to investigate.

1. Introduction

Security Operations Center (SOC) analysts continue to try to balance the continually overwhelming number of alerts that need to be analyzed versus configuring the tools available to them to tune the logic within those detection mechanisms to fit their organization's traffic patterns. While the vendors in the security space continue to develop robust and innovative solutions to keep up with adversaries' new attack methods, the tools used to detect these methods continue to exacerbate the problem within the SOC. In Gartner's Top 9 Security and Risk Trends article for 2020, "Extended detection and response (XDR) solutions are emerging that automatically collect and correlate data from multiple security products to improve threat detection and provide an incident response capability" (Petty, Gartner Top 9 Security and Risk Trends for 2020).

The use of Machine Learning (ML) technologies is not new to the Information Security world. Tools classified as User Behavior Analytics (UBA) or Network Traffic Analysis (NTA) utilize the Big Data Lake concepts to aggregate and correlate as much data as possible. As Dias, L.F. and Correia state, "ML allows the implementation of advanced algorithms to extract information from data using behavioral analysis or to find hidden correlations" (2020). Rather than merely implementing these technologies on endpoints or key collection points throughout the network, XDR SIEM technologies centralize all that data to correlate the logs across all platforms.

While these technologies continue to provide great value in the detection capabilities and raising awareness for the SOC, the continuation of a siloed toolset approach will continue to overwhelm SOC teams with too many tools, generating too many alerts, ultimately leaving them in a similar position as Target during their well-documented breach in 2013. They further explain: "The next-generation security information and event management (SIEM) systems should provide security monitoring with the means for automation, orchestration, and real-time contextual threat awareness. However, recent research shows that further work is needed to fulfill these requirements" (Dias, L. F., & Correia, 2020).

Author Name, email@address

The ability to quickly correlate additional data sources using XDR machine learning algorithms will provide beneficial information for SOC analysts. These algorithms increase/decrease the fidelity of those alerts, ultimately reducing the noise generated for SOC analyst review. To do this, analysts can leverage the robust alert data coming out of a machine learning platform and combine or cross-reference that data at that specific time to increase or decrease the fidelity of those alerts, effectively reducing the false positives that SOC analysts must investigate. Based on annual surveys, such as Ponemon's SOC Performance Report, successful implementation of these concepts will continue to ease the burden or stress that SOC teams find themselves under, ultimately making them happier, more productive, and committed to continue securing the environment they are protecting.

1.1. Outlining the Need

Reports such as The State of Security Operations 2020 and the 2020 Devo SOC Performance Report identify a gap between technology and process regarding alert handling. With the average number of alerts per day hovering around 11,000 (Brzezinska, 2020), SOC analysts are yearning for a way to reduce this amount to a manageable level. However, SOC analysts need to tread carefully with the amount of tuning that takes place within the log sources or alert-generating platforms. Finding the right balance between exclusion creation and alerting is a constantly evolving problem that faces security practitioners. What if there was a way to reprioritize the alerts generated from individual sources in an automated or more efficient manner without having to ignore the alerts generated from the powerful tools that our organizations have invested in? What if SOC analysts were able to implement cross-technology queries to correlate alert data, ultimately enriching that data to help the analysts focus on the alerts that have the most significant chance of being a true positive? Of the respondents in the Devo SOC Performance survey, 56% stated that they had an inability to prioritize threats properly (2020). With most security tools leveraging Application Programmable Interfaces (API), SOC analysts can pull data from multiple sources into a platform to correlate and enrich that data using technologies classified as Security Orchestration Automation and

Author Name, email@address

Response (SOAR). After processing, the enriched data is then pushed back into a SIEM or centralized alert dashboard for analyst review.

2. SOC Technologies

Traditional Security Operations Centers utilize a SIEM as their main point of reference. However, with the emergence of technologies such as Endpoint Detection and Response (EDR), Cloud log availability, and Network Traffic Analysis (NTA) technologies, the SOC analysts are forced to monitor multiple platforms. As always, technology providers have taken note of this and have developed tools that fall into the XDR classification, or Extended Detection and Response, which aim to aggregate these logs and concepts into what is being called the "Next Generation SIEM."

2.1. Extended Detection and Response

The Extended Detection and Response technologies show a promising future for several reasons. As the rise of cloud computing started to take off, so did the concepts of Big Data. The ability to store large amounts of data in a single repository is much more feasible for data scientists who could now leverage that data in various ways. In the more recent past, technologies such as User Behavior Analytics (UBA) and Network Traffic Analysis (NTA) toolsets leverage the capabilities in data lakes to provide advanced detection mechanisms within the IT Security space. The purpose of XDR platforms is to bring information generated from individual platforms and log sources together, much like a traditional SIEM would do, into a data lake, allowing for security analysts interact with that data at a large scale.

2.2. User Behavior Analytics

Sometimes referred to as User and Entity Behavior Analytics (UEBA), the technology focuses on indexing data relative to servers, endpoints, account names, and applications. This data is used to build and continually update a profile on unique users and accounts. After establishing baselines, the technology continues to monitor and compare new logs against that baseline to look for abnormalities such as impossible travel, login time, user asset access, user application usage, user login location, and user

process usage. These detection capabilities focus on providing insight into insider threats, account misuse, or compromised accounts or assets within an organization.

The major SIEM providers started adding UBA into their solutions around 2015. Gartner has now incorporated this capability into the evaluation criteria of their Security Information and Event Management Magic Quadrants. Gartner first introduced this concept in their evaluation criteria in the 2016 SIEM Magic Quadrant report and stated that they believe that at least 60 percent of the SIEM providers in the quadrant will have incorporated UBA functionality into their products by the end of 2017 (Gartner, 2016). Proving to be true, in their 2018 SIEM Magic Quadrant report, all seven companies in the leader's quadrant and 12 of 17 (71%) overall had incorporated UBA detections into their technology.

The ability to leverage the powerful computing readily available in today's landscape is a must. The baselines created for the unique users aggregate categorical, numerical, and contextual information to alert on risky behavior. The analyzed data revolves around individuals, devices, and applications, which is traditionally done manually by overwhelmed SOC analysts. The manual effort is extremely inefficient and highly error-prone, making UBA functionality a must-have for any organization committed to keeping their organization secure and retaining the talent within their security operations teams (Richards, 2017).

2.3. Network Traffic Analysis

Another popular use of Big Data is in the Network Traffic Analysis (NTA) space. Like UBA detections, endpoints, servers, network devices, and user data is being indexed for correlation. The detection capabilities in NTA technologies focus on user data volume (inbound and outbound), firewall accept or deny policy hits, long sessions, and IP/port scanning anomalies. Like the recent addition to the evaluation of UBA technologies in the SIEM Magic Quadrant, the Network Traffic Analysis technologies have made it on Gartner's radar. The difference with this technology is that it received its own Magic Quadrant in 2019 highlighting the fact that there are significant enough differences in how the technology operates to warrant separate classification.

In a traditional means of searching through network-based traffic logs, network or security engineers relied heavily on interface counter or bandwidth utilization metrics. These technologies evolved into tools that could classify traffic based on flow, or session information, and eventually by attributes. When these concepts were combined with the power of data lakes and machine learning, analysts could receive alerts when traffic patterns fell outside of the baseline established by the algorithms. This is once again a very powerful upgrade for the security community since it is another means of detection at our fingertips. However, like the UBA detection capabilities, NTA detections add another set of alerts for SOC analysts to evaluate, ultimately adding to the overall problem facing the SOC today, which is managing too many alerts.

This research aims to prove that combining the wealth of knowledge from products in the NTA realm with data from UBA or single sources in an automated fashion can add context and enrich the alert presented to the SOC analysts.

3. Alert Fatigue

With the introduction of User Behavior Analytics and Network Traffic Analysis comes additional detection mechanisms, which ultimately lead to additional alerts that Security Operation Center Analysts must evaluate. These alerts often warrant immediate attention due to the enhanced means of detection; however, analysts still must remain attentive to and respond to traditional SIEM alerts such as brute force login attempts, intrusion detection/prevention system (IDS/IPS) signature alerts, anti-virus (AV) detections, etc.

In an annual report compiled by Ponemon, 68% of the respondents stated too many alerts to chase when asked what makes working in a SOC so painful (2020). Additionally, respondents indicated information overload, inability to prioritize threats, and lack of tool integration as other factors contributing to fatigue. Participants were also asked to identify the most time-consuming tasks that their security analysts face. Responses in the top 10 answers included: gathering evidence for incidents, alert management, correlating data, and configuring automation (Ponemon, 2020). While this

data comes as no surprise, there has been no significant progress to address these sources of stress that cause burnout within a SOC.

Finding ways to improve the responses to these annual surveys is a must, and the reduction of tools is not a feasible solution. This research aims to identify simple, reusable concepts that can be easily modified to fit the tools within an organization to start improving the efficiencies of the way data is presented within the SOC.

3.1. Experiments and Testing

When it comes to User Behavior Analytics, some of the most triggered alerts revolve around account login activity. Detection mechanisms such as Impossible Travel and location or time-based anomalies are at the top of the list of most frequently triggered. Ironically, they often lack information to confidently classify the alert as a false-positive or true-positive without reaching out to the user to inquire if the end-user generated the logs presented in the tools.

The Microsoft O365 API provides information related to some of the common UBA detections and will be utilized as the basis of data within this research. The following sections will outline how the research will be conducted for the detection capabilities mentioned above.

3.1.1. Impossible Travel

Microsoft defines Impossible Travel: “The impossible travel detection identifies unusual and impossible user activity between two locations. The activity should be unusual enough to be considered an indicator of compromise and worthy of an alert” (Sagir et al.). One of the biggest causes of this alert type to be classified as a false-positive is virtual private network (VPN) solutions. This false-positive classification has seen a significant increase with the immediate shift to completely remote work in 2020 due to the COVID-19 pandemic. Additionally, the increased use of company-integrated applications on mobile devices and their authentication methods have triggered Impossible Travel detections. This research will leverage additional data sources such as threat intelligence feeds, multi-factor authentication logs, and specific application

interaction logs to help increase or decrease the fidelity of the Impossible Travel alerts that are generated from the O365 cloud.

3.1.2. Login Time Anomaly

Like the Impossible Travel detection capabilities within Microsoft's cloud, the login time anomaly detection utilizes machine learning algorithms to build a baseline for the user activity. When an individual typically works an 8 AM to 5 PM, Monday through Friday schedule, most authentication logs fall into that baseline. If authentication activity is observed outside of that window, anomaly detection is triggered, forcing the SOC to investigate. However, with the ease of access built into the mobile devices that employees carry today, it is not uncommon to authenticate outside of those windows, whether it be to quickly read and respond to email or check some of the collaboration applications to see if any teammates or peers are attempting to reach out to one another.

4. Findings and Discussion

The data collection utilized in this research was centralized into a SIEM based on the ELK stack. For the focus of this research, collecting data or Impossible Travel alerts is done through the Microsoft O365 API log collector. For the login time anomaly alerts, in addition to the O365 API, the Microsoft Azure Active Directory API and Windows event logs from Active Directory Domain Controllers are leveraged to capture authentication logs. The following sections describe how the testing was completed for each alert detection.

4.1. Impossible Travel Findings

The Microsoft O365 cloud offering continues to expand upon its functionalities along with its security detections. One of the security detections based on Machine Learning (ML) that has been around the Microsoft ecosystem for quite some time but continues to produce a high number of false positives is the Impossible Travel Anomaly. With a remote workforce becoming more commonplace, successful authentications from different IP addresses located in different geographic regions have become more of a legitimate reality within unrealistic travel periods.

Drastic and sudden changes to the workforce behavior warrant the analysis of evaluating alerts through a different lens to alert analysts to investigate Impossible Travel-related incidents when the fidelity is more indicative of malicious behavior. The data used in this research leveraged a 30-day interval of time and focused on the following data points within different log sources: username, time, event_id, userAgent, login_type, source_ip, and operation. The baseline for comparison for the Impossible Travel detection type is a total of 1,103 alerts. This alert type indicates the Microsoft infrastructure, either O365 or Azure Active Directory logs, captured two event IDs of 4624. That event ID indicates a successful account login. When logs with event ID 4624 are generated from two IP addresses in different geographic locations, within an unrealistic time period, an Impossible Travel alert is generated.

Since the Impossible Travel detection capability has been present within the Microsoft ecosystem for many years now, several methodologies have been published on how to help tune or reduce the noise generated from this detection. Instead of evaluating different ways to enhance or alter those findings, this research will raise the fidelity of alerts with additional indicators of compromise related to the original alert. The percentage of alerts presented to SOC analysts will be far fewer. The alerts that remain, will possess a greater probability of user account compromise that is also originating from a foreign location to that organization.

4.1.1. Testing

With the increase of mobile device functionality through enterprise-focused apps, users can quickly pick up their phone or tablet to check their email or any one of many different collaboration applications. These applications generate authentication messages when pulling down the most recent data, thus creating a 4624 successful login event. Additionally, with VPN technologies, that same user's laptop may be generating successful events with a different source IP address through the VPN tunnel, ultimately generating the Impossible Travel alert. Instead of having a SOC analyst investigate each one of these alerts, leveraging key variables within the payload of those alert to query other sources of data flowing into the SIEM provides powerful data enrichment. Key data points such as Password Spraying Activity, Multi-factor authentication success on the

user account, and bad-reputation detection from the source IP addresses involved in the login activity were utilized as variables within this testing. These three data points were leveraged to see what percentage of alerts remain in need of investigation from the original 1,013 we started with as our baseline.

Many, if not all, of the most popular SIEM technologies in place today have detections for brute force password attempts, and attackers know this. One of the password-based attacks that have emerged is password spraying, where an attacker attempts to log into each of the known accounts within a target organization with a common password on a low and slow cadence. While specific detections are being developed and implemented within a SIEM to combat this activity, these failed logins make for an excellent opportunity for cross-correlation to increase the fidelity of the Impossible Travel alerts being detected. The goal of the logic utilized in this testing was to take the username from the Impossible Travel alert and run a query to see if that username had a failed authentication attempt within the data set to Microsoft-based services. Since many of the application in use on a user's mobile device are cached, this correlation script returns data worthy of investigation. Out of the original 1,013 alerts, 799 had authentication failures associated with the offending username, resulting in a 21% reduction in alerts in need of investigation by the SOC.

The second set of correlations tested were the presence of multi-factor authentication logs. Using the username and IP address present in the Impossible Travel alert, the multi-factor authentication logs can be queried to determine whether successful authentications were made from the same IP addresses for that user. For the events where only successful multi-factor authentication logs were present, the Impossible Travel alert could be closed as a false-positive. This would indicate that the user provided the second form of authentication properly. However, immediate escalation of the alert for analysis is needed if the username is also present in a failed multi-factor authentication log. This escalation is necessary because the original Impossible Travel alert indicates successful authentication. For example, suppose an attacker successfully authenticated using stolen credentials and the compromised user instinctively provided the steps required for the second stage of the multi-factor authentication process after rejecting it one of the initial

times. In that case, this detection could provide valuable correlation to the Impossible Travel alert.

With the growing use of multi-factor authentication applications or integrations, users become numb to the steps needed for some forms of multi-factor authentication. They are beginning to blindly accept these notifications if they become repetitive to "silence" them. Out of the original 1,013 alerts, 34 had authentication failures associated with the offending username, resulting in a 97% reduction in alerts in need of investigation by the SOC.

The final correlation tested evaluated the source IP address in each Impossible Travel alert against Cyber Threat Intel feeds. The threat intel feeds utilized in this testing include the Department of Homeland Security (Automated Indicator Sharing), ProofPoint's ETPro (Emerging Threats Intelligence: Proofpoint US 2021), and Alien Vault's OTX (AlienVault - Open Threat Exchange). While the testing done on this data set did not find any IP addresses associated with a user in an Impossible Travel alert, this detection capability should be treated with urgency and immediately investigated when triggered. An expansion of this testing could include evaluating all public IP addresses associated with the offending user from the original alert and cross-correlate each of those IP addresses against the threat intel feeds to determine if any malicious activity could be related in a slightly removed manner.

4.2. User Login Time Findings

Login time anomaly detection is becoming harder and harder to utilize based on the high false-positive rate; however, if correlated with additional data points, this detection method can become extremely useful in detecting initial reconnaissance activity early in the Cyber Security Kill Chain model (Cyber Kill Chain®). Similar to the Impossible Travel analysis, the data points used in this alert correlation are username, time, event_id, userAgent, login_type, source_ip, and operation.

To establish the baseline on how many alerts were generated from the user login anomaly detection mechanism, a query was executed within the SIEM to specifically detect that alert over the same 30-day period the Impossible Travel alerts evaluated. This

resulted in 3,005 alerts generated from successful authentication logs within the Microsoft O365 or Azure Active Directory logs outside of the established user baseline after the machine learning algorithms have had ample time to baseline each user. As described in detail above, this is far too many alerts for a single SOC team to be able to handle effectively.

4.2.1. Testing

To demonstrate the ease of query reuse, the same query logic will be leveraged to promote the fidelity of the alerts generated from Microsoft's Login Time Anomaly detection. The username found in these alerts was first evaluated for the presence of external authentication failures. Then, like the Impossible Travel anomaly, multi-factor authentication logs and cyber threat intelligence feeds were also queried to increase the fidelity of the login failure alerts if specific criteria were present.

When evaluating the first set of alerts for Login Time Anomalies, when cross-correlated with logs associated with potential password spraying campaigns, there was a significant reduction in alerts that warranted investigation. Out of the original 3,005 alerts, 1994 had authentication failures related to the offending username, resulting in a 34% reduction in alerts.

Continuing to utilize the correlation above for the multi-factor authentication evaluation, the username was used as the key variable in the query between log sources. The resulting query contained a total of 149 logs that included a multi-factor authentication failure, which warrants SOC investigation to ensure that the user, in fact, mistakenly failed the authentication. This results in a 95% reduction in alerts.

The final correlation of alerts using the source IP as the key variable in the correlation against a cyber threat intelligence feed resulted in zero hits. Once again, although this did not produce any results within this data set, it does not indicate that this evaluation is not worth implementing. The ability to be able to build a script for automated correlation doesn't always have to produce immediate results; however, having the logic built into a SOC analyst's workflow provides endless possibilities. Continuing to enhance existing scripts or creating new ones that help produce the most

meaningful and context-rich alerts for SOC analysts help provide a prioritized view of the detections that have triggered within your newly integrated toolsets.

4.3. Testing Results Summary

The following charts summarize the two different alerts that were analyzed within this testing:

Impossible Travel Anomaly

Log Source Correlation of Impossible Travel to:	# of alerts remaining	# of alerts w/ remote authentication failures	# of alerts with MFA failure logs	# of alerts w/ CTI
Remote authentication failure	799		34	0
Multi-factor authentication	34	2		0
CTI	0	0	0	

Login Time Anomaly

Log Source Correlation	# of alerts remaining	# of alerts w/ remote authentication failures	# of alerts with MFA logs	# of alerts w/ CTI
Remote authentication failure	1994		149	0
Multi-factor authentication	149	14		0
CTI	0	0	0	

This summary demonstrates the importance of leveraging additional data sources at the analyst's disposal to effectively manage the number of alerts generated from the emerging machine learning algorithms. By combining the data enrichment correlations, organizations can effectively manage which alerts SOC analysts spend their time on. This research proves that these machine learning anomaly detections no longer need to be ignored entirely or classified as a false-positive but rather leveraged when investigating other detection mechanisms to provide context around the activity presented within those alerts. The power of layering in the correlation logic takes the total alerts that require

immediate investigation completed by an analyst from the original 1,013 Impossible Travel Anomalies down to two, and of the 3,005 Login Time Anomalies, down to 14.

5. Recommendations and Implications

While the research presented here focuses on a small subset of anomaly detection mechanisms, there are many opportunities to implement similar correlations to bubble up alerts that are most likely to be true-positives for analyst review. Similar anomaly detections that typically result in high false-positive classifications that would benefit from this integration would be login failure anomaly, successful user login anomaly, or user login location anomaly, to name a few. The variables present in each of these anomaly detections lend themselves to similar correlation techniques to reduce the amount of noise generated for the SOC. Furthermore, as the different vendors continue to emerge on the XDR stage, practitioners need to look to standardize how the data presented within those tools can be utilized and consumed in various other methods.

5.1. Recommendations for Practice

Many security practitioners are familiar with several different ways to script integrations. The importance of centralizing the concepts discussed in this research cannot be overstated. If organizations do not leverage a SOAR platform today, one might consider starting with an open-source tool such as TheHive (https://thehive-project.org/) to begin the journey. The capabilities within this tool are highly configurable to fit an organization's needs. This kind of option could provide analysts with the experience needed to properly evaluate some of the enterprise-grade solutions in the SOAR space.

The scripts outlined below can be implemented in an automated fashion so that the security analysts do not have to manually initiate these alert enrichment techniques, which ultimately saves the SOC team a quantifiable amount of time. The high-level implementation would be carried out according to this flow chart:

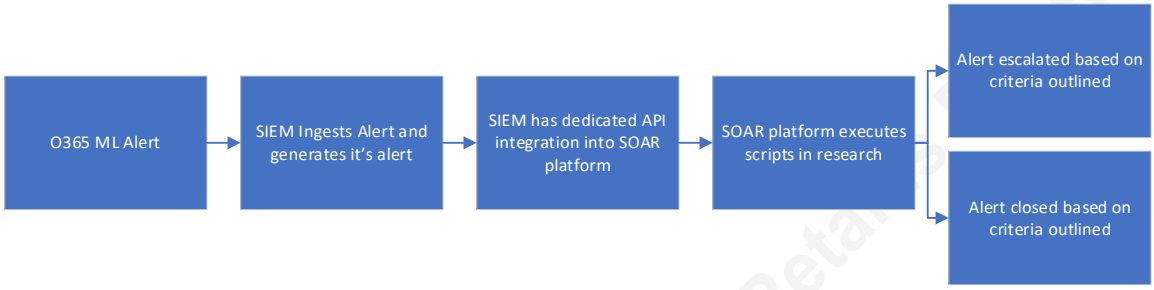

Figure 1. Automation workflow

Impossible Travel Anomaly and the User Login failure were the fourth and seventh overall alerts triggered within the 30-day time period for the correlations used to evaluate the alerts generated within this environment. As previously mentioned, the key variables detailed in the scripts below allow for easy re-use to begin reducing the noise in the high alert generating detections coming out of the machine learning platforms that are used to gain visibility into the activity taking place in the environment.

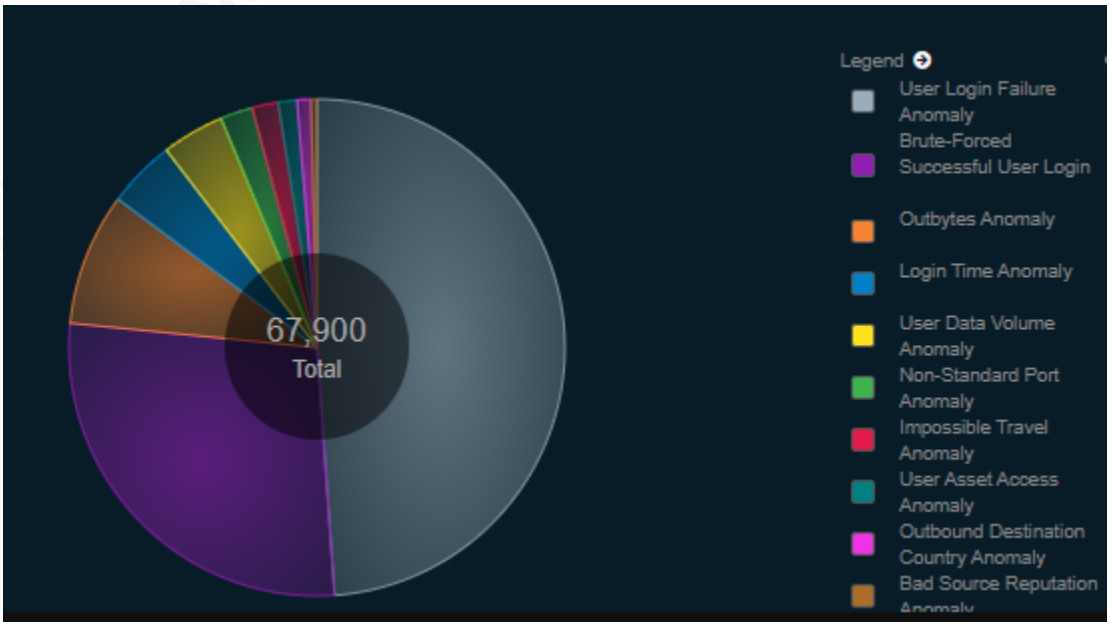

Figure 2. Total machine learning alerts during 30-day data evaluation

5.1.1. Automating Fidelity Enrichment Logic

With the total alerts present in this test data being 67,900, or roughly 2,263 alerts per day, the SOC analysts responsible for evaluating these alerts would most likely possess the same thoughts and feelings of frustration as those in the surveys previously mentioned. To begin making progress towards tool integration and automation, the author of this research has successfully integrated the enrichment logic presented in the research above into automated scripts within a SOAR platform. The benefits of real-time automation significantly improve the results above due to the additional variable of time. It is recommended to utilize something similar to the scripts below (Figures 3-6). Notice that the system queries are executing with a variable of time minus three hours compared to the research, which evaluates the three scenarios across the entire 30-day period. An example comparison is if User A generates an Impossible Travel Anomaly on the 10th day of the 30 days, but the failed authentication log to the Azure-based services is seen on the 15th days, then this correlation is irrelevant. The initial Impossible Travel Anomaly would have been ignored if the script were executed immediately upon alert generation to evaluate the 3 hours leading up to that alert. It is important to note that the variables contained within the scripts below should be tuned to meet your organization's risk appetite. For example, if a SOC analyst within your organization believes an attacker could wait longer than three hours before attempting another logon attempt, then the time variable should be adjusted accordingly.

The script has been modified for ease of understanding and needs to be adjusted to meet the specific variable names within your SIEM infrastructure. In this script securityAlert is the JSON record fed into the SOAR platform from the ELK SIEM. The variable azureFailedLogins is created and produces another JSON record resulting from a query back into the SIEM to obtain the correlation data needed for evaluation. In this scenario, the SIEM is queried using the creation time of the initial Impossible Travel alert and going back three hours from that point to capture all Azure Active Directory login attempts that did not result in a success. This JSON record is then searched in the if statement to look for the presence of the username from the initial security alert. If the username is present in the failed login attempts query, this could indicate a successful password spraying campaign. The fidelity field within the security alert is then elevated

to 100 to rise to the top of the SOC analyst's alert queue resulting in an immediate investigation. If the username is not present, it is deemed safe for alert closure since the resulting SOC analyst investigation would most likely result in a classification of false-positive.

```
import securityAlert
from datetime import timedelta

azureFailedLogins = securityAlert.createTime()-timedelta(hours=3)[msg_origin.source=:"office365"
 and workload="AzureActiveDirectory" and not login_result="success"]

if (securityAlert.username in azureFailedLogins.username)
    print("Increase fidelity")
    securityAlert.fidelity = 100
else:
    print("Close alert")
    securityAlert.status = closed
```

Figure 3. Failed Azure authentication comparison script

The script to query the multi-factor authentication logs within this test environment to determine if the user successfully passed or failed the multi-factor authentication request is displayed in figure 4 below. Once again, the original JSON record is imported, called the securityAlert. A JSON record in the variable called mfaLogs is created by executing the query in brackets, capturing multi-factor authentication logs from Azure Active Directory that did not result in a success within the three hours leading up to the creation of the original Impossible Travel alert. That JSON record is then queried, looking for the presence of the original username from the security alert to see if a failed multi-factor attempt occurred leading up to the successful login record that ultimately generated the Impossible Travel alert. As with the failed login attempts script in Figure 3, the same escalation or closure actions are performed on the original alert.

```
import securityAlert
from datetime import timedelta

mfaLogs = securityAlert.createTime()-timedelta(hours=3)[status.additionalDetails="MFA" an
workload="AzureActiveDirectory" and not login_result.keyword:"success"]

if (securityAlert.username in mfaLogs.username)
    print("Increase fidelity")
    securityAlert.fidelity = 100
else:
    print("Close alert")
    securityAlert.status = closed
```

Author Name, email@address

Figure 4. Failed Multi-Factor Authentication comparison script

The script shown in Figure 5 below is utilized to evaluate the IP addresses found in the Microsoft logs for their presence on a threat intel feed, ultimately classifying the IP address with a bad reputation. If the IP addresses associated with the login messages appear on any of the threat intel feeds integrated into the SIEM, then the alert should be immediately escalated to the SOC for analysis. Alternatively, if all IP addresses come back clean, then the alert should be evaluated against additional data points such as those described in Figures 3 and 4. While the presence of an IP address on a known bad list is something in need of immediate investigation, the non-existence in those feeds doesn't necessarily make the alert a false-positive on its own. Additional correlation or analysis is needed.

```
import securityAlert

if (securityAlert.reputation != 'good'
   print("Increase fidelity")
   securityAlert.fidelity = 100
else:
   print("Not on threat intel feeds")
```

Figure 5. Failed Multi-Factor Authentication comparison script

To utilize the power of scripting and combine the three correlations evaluated in this research, the script in Figure 6 will provide even more fidelity than the individual analysis conducted in the previous scripts. In this script, the same JSON records for correlation as previously described are built and first evaluated against the threat intel feed. If present, then we will immediately escalate to the SOC with the highest fidelity. However, if not present on a threat intel feed, then we need to perform further correlations. In the first elif statement in Figure 6, the logic is combined, meaning that the username must show up in both failed authentication logs as well as failed multi-factor authentication logs. This will once again raise the alert to the highest level for SOC analyst review. However, suppose the username is only seen in one of the correlations. In that case, the alert is set to a fidelity of 75, where it is still brought to the attention of the SOC but would leave room for other correlations taking place within the system that might warrant higher priority. As with each of the variables within these scripts, the

fidelities should also be set to the proper levels of priority based on how an individual SOC operates.

```
import securityAlert
from datetime import timedelta

azureFailedLogins = securityAlert.createTime()-timedelta(hours=3)[msg_origin.source=
"office365" and workload="AzureActiveDirectory" and not login_result="success"]
mfaLogs = securityAlert.createTime()-timedelta(hours=3)[status.additionalDetails="MFA" and
workload="AzureActiveDirectory" and not login_result.keyword:"success"]


if (securityAlert.reputation != 'good'
    print("Increase fidelity")
    securityAlert.fidelity = 100

elif (securityAlert.username in azureFailedLogins.username) and (securityAlert.username in
mfaLogs.username)
    print("Increase fidelity")
    securityAlert.fidelity = 100

elif (securityAlert.username in azureFailedLogins.username)
    print("Increase fidelity")
    securityAlert.fidelity = 75

elif (securityAlert.username in mfaLogs.username)
    print("Increase fidelity")
    securityAlert.fidelity = 75

else:
    print("Close alert, no additional correlation found")
    securityAlert.status = closed
```

Figure 6. Combination of all scripts

5.2. Implications for Future Research

Security analysts and engineers continue developing ways to detect and thwart the various attack methods the adversaries utilize. As different detection mechanisms are introduced into the tools that the operations center leverage, analysts and engineers must find ways to integrate those data points across their toolsets. Conducting additional research to determine which common data points are the most powerful within the security toolsets could prove extremely valuable. These variables could then be standardized into common formats, ultimately making the concepts outlined in this research much easier to implement. The vulnerability management community has proven success in this ideology by creating "The Security Content Automation Protocol (SCAP) [which] is a synthesis of interoperable specifications derived from community ideas" (Computer Security Division). Suppose the emerging machine learning toolsets could standardize their API or data format into the SCAP standard. In that case, the

concepts and scripts leveraged in this research will make tool automation and integration easier and ultimately much more powerful. Additional research could be conducted to identify the most common fields or variables across the different detection technologies and whether they align with those utilized in SCAP. This research will help determine if integration with SCAP is feasible or if a new standardization is needed.

6. Conclusion

The continual evolution of the technologies available to security teams has extremely powerful detection mechanisms to help identify and thwart the adversaries. Organizations must be careful to find the right balance between too many tools and data versus overwhelming and burning out the SOC analysts responsible for utilizing the data generated from these platforms. The concepts presented here demonstrate the importance of tool integration. While the examples in this research were specific to two different detection mechanisms, security analysts can leverage the concepts within this research to reprioritize the alerts generated from individual log sources and present the SOC analysts with higher fidelity alerts. By prioritizing alerts, analysts subsequently spend their valuable time investigating the higher fidelity alerts rather than wasting their time on alerts that tend to have an extremely high rate of a false-positive classification. Allowing your security team to begin utilizing these concepts will start freeing up time and empower them to find ways to use automation repetitively. The true power a SOC team should strive for is to automate the detection capabilities and start to leverage these concepts to automate their response capabilities, which significantly reduces the meantime to resolution. Overall, this might arguably be the most critical metric a security team should measure.

Author Name, email@address

References

2020 Devo SOC Performance Report, Ponemon Institute. (2020, June) Retrieved from https://www.devo.com/2020-devo-soc-performance-report/

AlienVault - Open Threat Exchange. AlienVault Open Threat Exchange. (n.d.). https://otx.alienvault.com/api.

Automated Indicator Sharing. Cybersecurity and Infrastructure Security Agency CISA. (n.d.). https://www.cisa.gov/ais.

Brzezinska, Ann. (2020, April) *Forrester: The 2020 State of Security Operations.* Retrieved from https://start.paloaltonetworks.com/forrester-2020-state-of-secops.html

Computer Security Division, I. T. L. (n.d.). *Security Content Automation Protocol: CSRC*. CSRC. https://csrc.nist.gov/projects/security-content-automation-protocol/.

Cyber Kill Chain®. Lockheed Martin. (n.d.). https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

Dias, L. F., & Correia, M. (2020). *Big Data Analytics for Intrusion Detection: An Overview*. In Ganapathi, P., & Shanmugapriya, D. (Ed.), Handbook of Research on Machine and Deep Learning Applications for Cyber Security (pp. 292-316). IGI Global.

Emerging Threats Intelligence: Proofpoint US. Proofpoint. (2021, January 5). https://www.proofpoint.com/us/products/advanced-threat-protection/et-intelligence.

Author Name, email@address

Hall, J., Flores, J., Robbins, M., Coulter, D., Foulds, I., Turscak, M., Manoj Reddy, K. N., McLaughlin, M., Jin, Y., Sharkey, K., Bahall, D., Mohanram, P., & Ross, E. (2020, May 15). *Use the sign-ins report to review Azure AD Multi-Factor Authentication Events.* Retrieved from https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-reporting

Johnson, M., Davis, C., Mumzhu, O., Coulter, D., Mazzoli, R., Huy, B., Narayanan, S., Simpson, D., Jupudi, A., Leavitt, S., Ahna, S., Halfin, D., Woitasen, D., Borys, A., Bailey, C., Ako-Adjei, K., Cole, L., Kay, J., Baumgartner, P., Travis, M., Vikramju, & Fosmark, T. (2021, May 5). *Search the audit log in the compliance center.* Retrieved from https://docs.microsoft.com/en-us/microsoft-365/compliance/search-the-audit-log-in-security-and-compliance?view=o365-worldwide

NIST. (2020, August 7) *Security Content Automation Protocol.* Retrieved from https://csrc.nist.gov/projects/security-content-automation-protocol/

Pettey, C. (2020, September 17) *Gartner Top 9 Security and Risk Trends for 2020*. Retrieved from https://www.gartner.com/smarterwithgartner/gartner-top-9-security-and-risk-trends-for-2020/

Richards, K. (2017, December). *UBA Leads the Security Analytics Race.* Retrieved from https://searchsecurity.techtarget.com/feature/User-behavior-analytics-leads-the-security-analytics-charge

Sagir, S., Martinez, J., Baldwin, M., Sharkey, K., Cai, S., Karlin, R., Stewart, M., Lamos, B. (2021, January 5). *Get behavioral analytics and anomaly detection*. Retrieved

Author Name, email@address

from https://docs.microsoft.com/en-us/cloud-app-security/anomaly-detection-policy#

Author Name, email@address