jamk.fi

Methods for Managed Deployment of User Behavior Analytics to SIEM product

Mikko Seppänen

Bachelor's thesis
May 2021
Information and Communications Technology
Degree Programme in Information and Communications Technology

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Seppänen, Mikko	Opinnäytetyö, AMK	Toukokuu 2021
	Sivumäärä 63	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Menetelmät User Behavior Analyticsin hallitulle käyttöönotolle SIEM-tuotteeseen

Tutkinto-ohjelma
Tietojenkäsittely ja tietoliikenne

Työn ohjaaja(t)
Saharinen, Karo

Toimeksiantaja(t)
Nixu Oyj

Vastatakseen nykyaikaisen uhkaympäristön haasteisiin organisaatioiden on pitänyt miettiä kyberpuolustusstrategiaansa uudelleen. Perinteiset järjestelmät, jotka keskittyvät suojaamaan yrityksen verkkoa vain ulkoa tulevilta hyökkäyksiltä eivät kykene vastaamaan nykyaikaisien uhkien tarjoamiin haasteisiin. Uhkatoimijoista on tullut pitkäjänteisempiä ja vaikeammin havaittavia, ja organisaation verkon raja on muuttunut vähemmän konkreettiseksi laajemmin käyttöön otettujen pilviratkaisujen, IoT:n ja kannettavien henkilökohtaisten laitteiden ansiosta. Organisaatiot ovat alkaneet siirtyä Zero Trustin kaltaisiin ratkaisuihin, mutta tämäkään ei ole kaikenkattava vastaus tietomurtojen ehkäisemiseksi.

Organisaatioiden verkkoja ja laitteita valvovat tietoturvavalvomot kohtaavat myös uusia haasteita havainnointikyvyn tuottamiseksi yhä edistyneemmille hyökkäyksille. Käsiteltävän datan määrä kasvaa myös jatkuvasti, mikä venyttää perinteisen automaation ja analyytikoiden kyvyt äärimmilleen. *User and entity behavior analytics* ja koneoppimisratkaisujen tarkoituksena on parantaa tätä kyberpuolustuksen näkökohtaa lisäämällä mahdollisuuksia havaita hyökkääjän haltuun joutuneet käyttäjätilit tai laitteet ajoissa, tarjoamalla organisaation lokitietojen tehokkaampaa käsittelyä.

Opinnäytetyön tavoitteena oli luoda prosessi user behavior analyticsin käyttöönottamiseksi SIEM-tuotteeseen käytettäväksi MSSP SOC:ssa. Tärkeä osa prosessia oli määrittää, kuinka hyvin tuotetta voidaan räätälöidä erilaisiin asiakasympäristöihin. Alustavana suunnitelmana oli testata järjestelmä ja prosessi pilottiasennuksessa, mutta tuotteen kanssa testausympäristössä ilmenneiden teknisten ongelmien vuoksi asennusta lykättiin ja jätettiin tämän tutkimuksen ulkopuolelle. Käyttöönottoprosessia on tarkoitus käyttää tulevaisuudessa muissa SOC-asiakkaissa. Tärkein havainto on, että tuote näyttää lupaavalta ja muokattavalta mihin tahansa ympäristöön, mutta ilman testaamista ei voida sanoa järjestelmän tuomaa lopullista etua.

Avainsanat (asiasanat)

Kyberturvallisuus, SOC, User and entity behavior analytics, SIEM

Muut tiedot (salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Seppänen, Mikko	Bachelor's thesis	May 2021
		Language of publication: English
	Number of pages: 63	Permission for web publication: x

Title of publication

Methods for Managed Deployment of User Behavior Analytics to SIEM product

Degree programme

Information and Communications Technology

Supervisor(s)

Saharinen, Karo

Assigned by

Nixu Oyj

To keep up with the challenges of modern threat environment, organizations have had to rethink their cyber defensive strategies. Traditional perimeter-based defenses are unable to meet the challenge provided by modern threat landscape. Threat actors have become stealthier and more persistent, and the perimeter of the organization has also become less concrete due more widely adapted cloud solutions, IoT and portable personal devices. Organizations are starting to adapt solutions like Zero Trust, but the obstacles provided by the extra access controls are not all-encompassing solution to stop all breaches.

Security Operation Centers monitoring the organizations' assets are also facing new challenges to produce quality detections for the increasingly advanced attacks. The volume of data that needs to be processed is drastically increasing stretching the traditional automation and analysts' capabilities to a point where the systems are being overwhelmed. User and entity behavior and machine learning solutions aim to improve this aspect of cyber defense by increasing the chances that compromised user accounts or devices are detected in time by providing more effective processing of the organization's security data.

Objective of the thesis was to create a process for deployment of user behavior analytics to SIEM product for use in MSSP SOC. Important part of the process was to determine how well the product can be customized for varying customer environments. Initial plan was to test the system and process in a pilot deployment, but due to technical issues with the product in test environment the installation was postponed and left out of the scope of the thesis. The process for the deployment is planned to be used in future deployments for other SOC customers. Key finding is that the product seems promising and customizable for any environment, but without testing any conclusive statement of the benefits cannot be said.

Keywords/tags (subjects)

Cyber security, SOC, User and entity behavior analytics, SIEM

Miscellaneous (Confidential information)

Content

Figures

Tables

1 Introduction

Organizations cyber defense strategies have had to adapt in recent years from concentrating only on the traditional perimeter based defensive systems, where the focus is to defend the border of the company from outside intrusion to models like Zero Trust where the focus is shifted from the perimeter to identity verification of every person and device attempting to access resources inside the network.

This is because the perimeter of a company has become less and less concrete due to more widely adapted cloud solutions, IoT and portable personal devices. Users interact with company data from external networks using laptops or smartphones and the data and services are often in cloud services provided by third party vendors. (Wang 2017.)

According to IDG Security Priorities Study (2018, 5), only 13% of companies had adapted Zero Trust model, but the adaption has greatly accelerated in the last few years. In 2020 Okta surveyed over 500 security leaders and found that 40% of organizations globally are working on Zero Trust initiatives. (The State of Zero Trust Security in Global Organizations 2020.) But still today most companies are relying on the perimeter based defensive systems, and even if properly implemented, the Zero Trust will not be an all-encompassing solution to stop all breaches (Carder 2019). The compromised user accounts still need to be detected.

The present-day reality is that most organizations' computer systems are built in a way that single compromised user account can lead to the compromise of the whole organization's IT infrastructure if not detected in time. Most of the recent large-scale data breaches have started from a single compromised device or user account (Wang 2017).

One example of this is a story from CrowdStrike Cyber Intrusion Services Casebook (2018, 11) where the whole EU infrastructure of an organization described as "apparel manufacturer with a global presence" was compromised. This incident started with one employee taking their laptop to a coffee shop and visiting a malicious URL delivered by a phishing e-mail. This led to installation of Dridex banking trojan and

PowerShell Empire post-exploit toolkit. There were no indications of the initial compromise from the company's security systems, as they relied on the host being inside the company's network. After the host was joined back in the company network it was used as an entry point for lateral movement to other systems where the attackers were able to steal more credentials and eventually gain access to Domain Controller and business-critical backend servers.

The biggest weakness of most perimeter based defensive systems is that once the perimeter has been breached it will be extremely hard to detect the malicious activity inside the network. Attackers leverage legitimate user and service accounts and often use binaries that already exists in the target system in so called living-off-the-land attacks. In many cases the attackers have been able to spend months in the breached network before being noticed. (Kothari 2018.) In CrowdStrike Cyber Intrusion Services Casebook (2018, 6) it is stated that the average time the attackers were able to spend inside the compromised organization before being detected was 85 days.

For organizations today some of the users or devices getting compromised is not a question of "if", but rather "when". The question is then how to detect these compromised assets before they can cause further damage in the organization. (Wang 2017.)

UEBA solutions aim to improve this aspect of cyber defense by increasing the changes that compromised user accounts, devices or other types of malicious insider activity is detected in time. This is done by creating a baseline of the entity's normal behavior using machine learning and comparing future activities to this baseline by applying various analytics ranging from traditional pattern-based rules to advanced machine learning analytics. This will calculate a risk score for the entity and this accumulated risk score can then be used to generate alerts to focus the investigation to riskiest assets.

This thesis will examine the UEBA in the viewpoint of Security Operations Center. One of the challenges SOCs face is the constant need to develop the detection methods for security incidents in continuously evolving threat landscape and the ever-increasing amount of data that needs to be processed.

The raw log data ingested by SIEM is a prime example of streaming data that can scale beyond the traditional automation and human analysts' capabilities, stretching them to a point where the systems are overwhelmed. (Tuor, Kaplan, Hutchinson, Nichols & Robinson 2017.) Traditional pattern based SIEM rules have often a high false-positive ratio leading to alert fatigue (Feng, Wu, Li & Kunkle 2017).

UEBA and ML are tools that aim to improve the automation of existing SIEM system by providing more effective processing of the ingested log data to present the human analysts more relevant alerts. The information contained in the alerts generated by UEBA is also more comprehensive than that of a traditional SIEM correlation rule. The alerts include graphs of the user's past behavior and a collection of all the related anomalous events in a timeline to aid the analysis process.

2 Premise for the thesis

2.1 Employer

Assignment for the thesis was given by Nixu Oyj. Nixu is a cybersecurity services company located in northern Europe. Founded in 1988 in Helsinki, Nixu currently employs nearly 400 cybersecurity professionals and has market presence in Finland, Sweden, Denmark, and Netherlands. In 2020 Nixu had a revenue of 53,3 million euros. (Nixu Corporation n.d.)

In 2015 the Nixu Cyber Defense Center (CDC) was founded. CDC offers the traditional Cyber Operations Center (SOC) solutions with addition of complementary Managed Security Service Provider (MSSP) services. (Nixu Corporation n.d.)

2.2 Objectives

Objective for the thesis was to create a process for deployment of user behavior analytics (UBA) to SIEM product for use in MSSP SOC. The SIEM product used for this is IBM QRadar. The system was originally intended to be piloted in SIEM that is monitoring the employer's own network and assets, but due to technical issues with the product in test environment the schedule of the installation was postponed, and this piloting phase was left out from the scope of this thesis.

The process for the deployment is planned to be used in future deployments for other SOC customers. Important part of this process is to configure the system to serve customer specific needs since monitored environments vary in the structure of the network, assets, log data available for monitoring and the risk assessments of the company.

The user behavior analytics is integrated to the SIEM using two additional applications: User Behavior Analytics (UBA) for QRadar and Machine Learning Analytics app, which empowers the functionality of the former by enabling the use of advanced analytics.

Part of the thesis is also determining the differences and advantages UEBA brings to security monitoring in relation to traditional SIEM based monitoring without UEBA. This includes looking into how the system can be used to improve detection capabilities for existing use cases and what new use cases the system allows to monitor for that is ineffective with traditional methods.

The desired outcome by employer for this system is to improve the detection of security incidents by more effective processing of the available log data. This includes possibility to detect security incidents that might otherwise go unnoticed while potentially lowering the false-positive ratios of existing SIEM deployments. Also, the analysis process of security incidents related to user activity is expected to become more thorough since the data is processed and presented to analysts in greater detail.

3 Theory basis

3.1 Log data

Log is a record of events that has occurred within the monitored system or environment. Log consists of log entries which each contain information about single event that has taken place.

Logs are generated by wide array of systems and devices in the network such as firewalls, antivirus software, Active Directory servers and operating systems of endpoints. These are just some examples, and basically all devices and applications in the network generate some form of log entries; many of which contain information that can be relevant in security viewpoint. (Kent & Souppaya 2006, 2-1)

Originally logs were mostly used for debugging purposes but have evolved to serve various functions such as monitoring and optimizing system and network performance, monitoring user activity and providing data for security incident investigations. (Kent & Souppaya 2006)

The types of a log records and the information they contain vary a lot depending on the log source. Some of the log sources are specifically security oriented, for example Intrusion Detection Systems (IDS) or logs of endpoint protection tools. Other type of log sources that may not explicitly log security events can still contain information in the events that is relevant for security monitoring as they enrich the insight to the monitored environment and user activity. Examples of these could be switches and wireless access points. (Kent & Souppaya 2006)

Some of the most common log types that are relevant to security monitoring are coarsely:

- Antimalware software
- Firewall logs
- Logs from endpoints (e.g. EDR or Sysmon)
- Authentication server logs
- Directory server logs (i.e. Domain Controllers)
- DNS logs
- Web proxy
- Remote access software (e.g. VPN)
- Intrusion detection (IDS) and prevention (IPS) logs
- Logs from critical applications (specific to environment)
- Cloud service logs (e.g. Office 365)

As most of security monitoring depends on log entries generated within the monitored environment, the same is true for the user behavior analytics. All the data of the users' actions will be based on the log entries available for the system.

When we are looking at security monitoring in SOC standpoint, the monitoring generally covers most of the organizations network and assets. As a SOC might monitor large international companies with tens of thousands of assets, the log entries generated can reach extremely high numbers. The log velocity is often counted in Events Per Second (EPS) which can range up to 10 000 or more for single large organizations environment.

This amount of log entries is naturally too much for human analysts to go through manually. For this the common solution is to use automation provided by SIEM systems to process the data in real time and generate alerts and reports for the analysts.

3.2 SIEM

Security information and event management (SIEM) is a software product that aims to provide a comprehensive view into organization's IT security by providing real-time monitoring, alerting, reporting, and visualizing security related data. The basic principle of the system is that relevant data about organization's security is generated in multiple locations thorough the infrastructure and centralizing this data to one point allows to see trends and more easily detect anomalous patterns. (Agrawal & Makwana 2015, 1893-1894.)

SIEM combines the functions of security event management (SEM) and security information management (SIM) into one system. SEM consists of real-time monitoring, correlation of events, alerting, console, and dashboard views. SIM is the process for collecting the data to one centralized repository for trend analysis and reporting. (Agrawal & Makwana 2015, 1894-1896.)

The main data used by SIEM is the log and flow events generated by devices in the monitored environment. This often includes managed cloud services and Software as a Services (SaaS) like Microsoft Office 365. Additionally, SIEM can use data from other sources to support and enrich the detection capabilities and to provide contextual information about the monitored assets. This can include vulnerability scanning data, threat intelligence, and asset data of the monitored environment such as LDAP data for information about the users and other Directory Services objects.

Originally the Payment Card Industry Data Security Standard compliance drove the rise in SIEM adoption in large enterprises, but the rising concern of advanced threats facing companies today has led to increased deployment of SIEM products in wider variety of organizations. (Security Information & Event Management (SIEM) Market: Growth, Trends and Forecasts (2019-2024) - ResearchAndMarkets.com 2019.)

This is because traditional security controls such as firewalls and antivirus software alone are not adequate at detecting or stopping the more advanced threats. In 76% of organizations harmed by APT (Advanced Persistent Threat), antivirus software alone was not enough to provide an obstacle for the attackers (Rot & Olszewski 2017, 115). With SIEM it is possible to correlate events from multiple log sources to better detect more complex attack patterns.

3.2.1 Data aggregation and normalization

SIEM can collect data from various sources, such as servers, firewalls, IDS, and end-point protection tools. Most devices and applications that generate logs can be made to forward them to SIEM.

The log is usually forwarded to SIEM by log collectors. Log collectors are centralized servers that receive the log from the devices in the network, for example via syslog protocol. Linux and Unix based operating systems on network infrastructure like fire-walls, routers and switches usually support syslog forwarding natively. Devices that do not natively support log forwarding in the format required by the log collector can have a log forwarding agent installed. Examples of log forwarding agents are Rsyslog and Syslog-NG. (Todd, B 2017.) The log collector can also pre-process the data, for example adding headers that include information about the device that generated the log or filtering out unneeded information before it is forwarded to SIEM.

The traditional log collecting agents cannot be used for all log sources. Usually in cases of managed cloud services and SaaS applications installing these agents is not possible. Therefore, a direct integration using API is needed. Support for the most common services such as Office365 usually comes natively with the SIEM product. (SIEM Architecture: Technology, Process and Data, N.d.)

Logs from different products from different vendors can use varying formats in the log messages. Log normalization means turning this data into consistent form by reducing it to common event attributes. (Lane, A. 2010.) Most log sources include the same base event attributes such as IP addresses, timestamp, username, and hostname, but the format in the raw data can be very different.

The normalization is needed for SIEM products to process correlations and analytics with the log data effectively. For example, if SIEM is made to alert of connections to malicious destination domains, it needs to know which field in the log message presents the destination domain.

Part of the normalization also includes categorization and naming of events. This means mapping the events into common taxonomy of high- and low-level categories and giving the log message an event name. For successful logon message these could be following respectively: “authentication”, “authentication success” and “An account was successfully logged on”. These categories can then be used in searches and correlation rules without having to specify the log source or the exact events.

Most SIEM products can normalize generally used log formats by default and custom parsers can be made to support the more exotic ones. The received log data can also be enriched by information not contained in the log message. Example of this is geolocation and registrar data of external IP addresses.

3.2.2 Log correlation and alerting

Log correlation is comparing different events to each other or to contextual data by rule-based, statistical, or algorithmic methods. This can be done either in real-time or for historical data. (Chuvakin, A. N.d.)

Vast majority of log events ingested by SIEM are records of normal user, application, or operating system activity. Even if some of these events would be part of a security incident, when viewing a single event alone this may not become apparent, but when correlating with some other events can lead to an actionable indicator. Attacks usually consist of sequence of events and the purpose of correlation rules is to detect suspicious sequences that could indicate an attack.

The most basic way a correlation rule is built in SIEM is by defining a condition using basic logic operators and time or count constraints to normalized event attributes and choosing a response when these conditions are met. The response can be for example generating an alert or report for analysts to investigate or creating a new event or state for further correlation.

Example of a simple correlation rule that attempts to alert of successful password spraying attack could look something like this:

When more than 50 login failures are detected with different username and same source IP followed by login success from the same source IP in 30 minutes

Here the *login failure* and *login success* are normalized event categories and *source IP* and *username* normalized event attributes. This would likely need to be further correlated with asset data to whitelist source IPs that legitimately pass logons for multiple users, like ADFS servers.

This type of static threshold-based correlation rule can work in simple use cases, but when the attacks get more advanced the effectiveness of this type of simple rules quickly tapers off. It requires only small adjustment in the attacker's actions to go un-detected. The aforementioned rule would've likely already existed in early 2000's SIEM systems. (Chuvakin 2019.)

Modern threats and data volumes may require more modern approaches, such as systems utilizing ML and risk-based alerting.

3.2.3 QRadar

QRadar is a SIEM product by IBM. QRadar was Initially developed by Q1 Labs in 2007. IBM acquired Q1 Labs in 2011 and has been developing the product since. (Q1 Labs Inc n.d.) The basic architecture of the product is seen from figure 1.

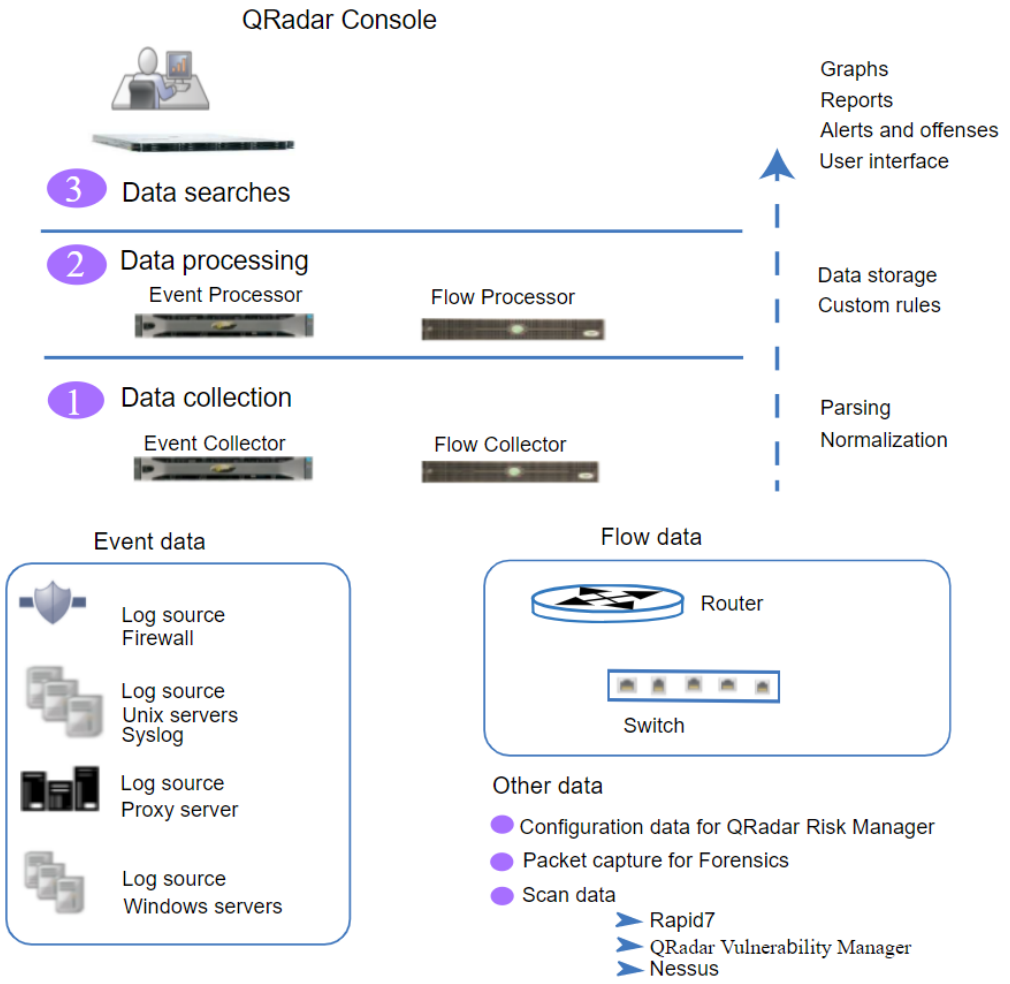

Figure 1. Overview of QRadar architecture (QRadar architecture overview n.d.)

Data collection

The first layer of operation is data collection. There are two main types of data that is ingested: log events and network flow. QRadar Event Collector collects events from log sources, parses, and normalizes the raw logs to structured format. Flow data is network traffic or session information which QRadar translates into normalized flow records. Each flow record represents a session between two hosts. After the collection and normalization, the event and flow data is sent to processing layer. (QRadar Architecture and Deployment Guide n.d.)

Custom Rule Engine (CRE)

On the processing layer the events and flow are run through the custom rule engine. CRE analyzes incoming events in real-time, matching them to rules and building blocks. Building blocks are used to group commonly used tests and to build complex

logic for rules. Building blocks use the same tests that rules use but do not have a rule response. (IBM QRadar building blocks n.d.) Whenever all conditions in a rule match, the rule generates a response. Most common rule response is to create an alert, which in QRadar terms is called an offense.

Data searches

On the last layer the collected and processed data is made available to the users. Users can run searches, generate reports, and investigate offenses. Searches can be made from user interface using given list of parameters, operators and inserting searchable value. Ariel Query Language (AQL) can be used to perform more advanced searches. (QRadar Architecture and Deployment Guide n.d.)

3.3 LDAP

LDAP (Lightweight Directory Access Protocol) is a standards-based protocol for interacting with directory servers (Learn About LDAP n.d.). Most common use for LDAP is to authenticate users and store information about directory service objects such as users, groups, and applications.

Entries in LDAP are collections of information about an entity. Entry consists of a distinguished name, collection of attributes and a collection of object classes. Distinguished name is a unique identifier for the entry which also acts as a fully qualified path to the entry in the directory information tree. Attributes contain the actual data for the entry. Each attribute has a name and one or more values. (Basic LDAP Concepts n.d.)

LDAP is used by most of the prevailing directory services, like Microsoft Active Directory, which is the most common directory service used in companies today by a wide margin. Some other examples are Red Hat Directory Service, OpenLDAP and Apache Directory Server. (Sobers 2020.)

Having read access to LDAP enables SIEM to maintain up to date information about relevant assets of the company. This is especially important in the case of UEBA as this information is needed to build the user profiles that allows peer group analysis and use cases related to monitoring of critical users or other critical assets.

3.4 Machine Learning

Machine learning is a category of algorithms that allow computers to predict outcomes without being explicitly programmed. An algorithm can be defined as a series of instructions that transform input into output. Example of a conventional algorithm is a sorting algorithm, where the input is a set of numbers and output an ordered list of these numbers. This can be explicitly programmed as we can write out the exact steps that will always lead to correct output with all possible input variations. (Alpaydın 2010.)

For some tasks adequate conventional algorithm cannot be written as the variance in the input can be effectively infinite and changing over time. Also, what can be considered as valid output can chance with time. Example of this could be an e-mail spam filter where the input is an e-mail file, and the output should be simply Boolean true or false, depending on if it is spam or not. In this case what the output should be is known, but we do not know how to transform this input into the output in a way that would work for most of the input variety. (Alpaydın 2010.)

Similarly, there are many other tasks that humans have learned to do manually, but cannot explain how we do it, and therefore cannot write the algorithm for the job. For example, facial recognition or turning heard speech into ascii characters on a computer. This is something we humans have learned to do inherently but describing every step in a way that could be coded into a program is not possible. (Alpaydın 2010.)

There are many problems like this where we do not know the algorithm for but have plenty of sample data available. For the spam filter, we can easily collect vast amounts of example spam and non-spam messages. The idea then is that the computer would extract the algorithm from this data. (Alpaydın 2010.)

Machine learning is often classified in following categories.

Supervised learning

In supervised learning the machine learning algorithms are fed a set of labeled data that contains the input and desired output. In the case of email spam filter the sys-

tem is fed example email data that is labeled simply with “spam” or “not spam”. After the system is trained using this labeled data it is tested by using non-labeled data to see if it can correctly predict the output.

Supervised learning problems include classification and regression. In classification problems the model categorizes input into two or more classes, such as in the email filters spam classification. In regression problems the output is a numeric value within a range. For example, a prediction of value of a car based on its attributes and past sales. This kind of learning from the data also allows for knowledge extraction; If the model correctly predicts the prices, looking at this model will also reveal the properties that makes up for the price. (Alpaydın 2010.)

Unsupervised learning

In unsupervised learning the fed input data is not labeled, classified, or categorized and the system reacts on the data without prior training. The aim is to study the innate structure of the input data to find regularities, creating groupings or clusters of the data.

Applications for this are segmenting datasets based on some shared attribute, detecting anomalous data that do not fit into any existing group and simplifying datasets by reducing their dimensionality. (Roman 2019.)

Reinforcement learning

In reinforcement learning a software agent interacts with an environment and takes actions available at its current state. The state can be for example the current position of chess pieces on a chess board. Every action returns a reward or a penalty for the agent and changes the state of the environment. The end goal is to find a sequence of actions that has the maximum cumulative reward, creating the best possible policy to solve the problem. (Alpaydın 2010.)

Deep learning

Deep learning (DL) is a subset of Machine Learning using multi-layered structure of algorithms called artificial neural network. This approach is inspired by the functionality of human brain. Neural networks at the basic level includes four components:

inputs, weight, a bias or threshold, and an output. The word “deep” in deep learning refers to depth of the layers in the neural network. (Kavlakoglu 2020.)

Deep learning has been the key at the recent years’ advances in machine learning, driving the success of applications that process unstructured data, such as speech recognition software and self-driving cars.

3.5 Machine learning in cybersecurity

Machine learning has been applied in many areas of science where large quantities of data need to be processed due to it offering great potential for adaptability, scalability, and ability to adjust to new and unknown challenges. (Ford & Siraj 2014.) These are also very prominent challenges in cybersecurity, especially for SOCs.

The problems ML is used to solve in cyber security is that network attacks and malware are constantly evolving and implementing methods to avoid detection from systems that base their monitoring on known patterns like file hashes, IDS signatures or known bad user behavior. Systems relying only on these pattern-based detection methods are only capable at finding attacks that have been observed before. This is also a challenge for machine learning, especially for supervised algorithms, as there naturally cannot exist a training dataset that includes the new never-seen-before attacks.

3.5.1 Methods of machine learning used in cyber security

On publication *On the Effectiveness of Machine and Deep Learning for Cyber Security* Apruzzese, Colajanni, Ferretti, Guido and Marchetti (2018) categorized the ML algorithms currently used in cyber security (see figure 2.)

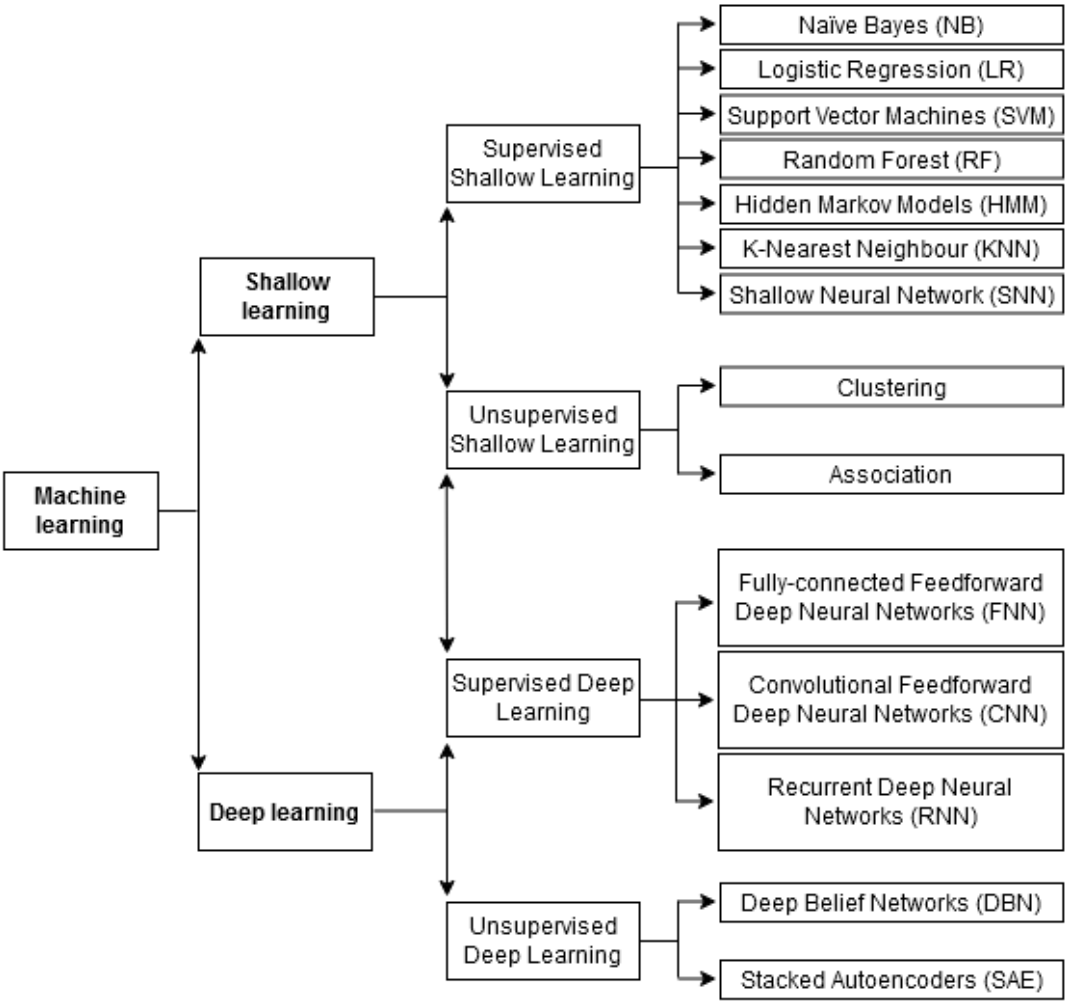

Figure 2 Classification of ML algorithms used in cyber security (Apruzzese et al. 2018, 374, Modified)

In this figure the algorithms are divided into Shallow learning (SL) and Deep learning (DL) and further into supervised an unsupervised category for both.

On the next table (table 1.) the algorithms are listed under the problem they are used to solve. Note that this table does not include UEBA. The algorithms used in UEBA will be listed in 4.3 using the QRadar UBA as example.

Table 1. Application of ML algorithms to cyber security problems (Apruzzese et al. 2018, 378, Modified)

	Network IDS	**Botnet detection**	**DGA detection**	**Malware Analysis**	**Spam Detection**

DL	Super-vised	RNN	RNN		DNN CNN RNN	
	Unsuper-vised	DBN SAE			DBN SAE	DBN SAE
SL	Super-vised	RF NB SVM LR HMM KNN SNN	RF NB SVM LR KNN SNN	RF HMM	RF NB SVM LR HMM KNN SNN	RF NB SVM LR KNN SNN
	Unsuper-vised	Cluster-ing Associa-tion	Clus-tering	Clustering	Clustering Association	Clustering Association

3.5.2 Applications of ML in cyber security

Most common applications for ML in cybersecurity are intrusion detection systems, network traffic analysis, malware analysis and email spam filters. The UEBA is a newer application applying the ML in cyber security; the term was coined by Gartner only in December 2015.

Following are few applications where ML has been applied to in cyber security.

Intrusion detection systems

Intrusion detection systems (IDS) aim to discover malicious activity in monitored network or computer. IDS systems have traditionally relied on signatures based on known patterns, like byte sequences in network traffic. While the signature-based detections are good at detecting known attacks the problem with this approach is inability to detect novel attacks and often having high number of false positives. Also,

the signatures need to be created and reviewed by experts before publishing them into production causing latency before the system can detect the attack.

Various ML approaches has been applied to IDS systems, either by training the models using labeled datasets, or using DL methods that are trained on larger unlabeled datasets in aim to find useful patterns. (Ahmad, Khan, Shiang, Abdullah & Ahmad 2020.)

NTA and NDR

While having similarities with IDS, NTA (Network Traffic Analysis) is a lot more complete system intercepting, recording, and analyzing network traffic patterns to detect security threats. This is further used by NDR (Network Detection and Response) products like Darktrace and Vectra.

ML is used for regression and prediction on packet parameters and comparing these to baseline, classifying and identifying classes of network attacks, such as scanning and clustering for forensics analysis. (Polyakov 2018.)
Few examples how NDR product Reveal(x) uses ML: First is clustering monitored assets to peer groups, device policies, and device and user roles by observing the behavior of every entity on the network. Second is building multiple predictive models based on different aspects of the behavior for each entity on the network. Lastly these are fed to detectors that attempt to alert of suspicious activities based on meaningful deviations on the models. (Wu, E. 2020.)
This is quite similar on how UEBA operates, but NDR is focused on network traffic and does not use log events from the systems or data from activity inside the endpoints.

Malware analysis

Modern malware attacks have created a need for new detection methods since signature-based detections are ineffective at detecting novel or polymorphic malware (Apruzzese et al. 2018, 377).

Heuristic detections have a better chance of detecting previously unknown malware, but even the heuristic detections rely on known methods of previously observed malware. Heuristic detections also have high chance for false positives. ML can be

used to enhance the heuristic detections accuracy by using the observed features to group the detections to different malware families. (Chumachenko, K. 2017.) The features can be for example API call signatures, DLL loads or data from PE headers.

DGA detection

Domain Generation Algorithms are a technique used by many present-day malware families to establish a command-and-control channel. Algorithm is used to generate domain names periodically which the malware attempts to contact instead of using hard coded list of domains or IP addresses. The domains are initially unregistered and the malware controllers knowing the algorithm and the random seed can predict which domains will be contacted and register these when needed. This makes it harder to detect or block the command-and-control traffic only by maintaining block-lists of known malicious domains. (Sivaguru, Choudhary, Yu, Tymchenko, Nascimento & De Cock 2018.)

Various type of ML approaches has been proposed for DGA detection and classification. These can be categorized in two types: retrospective classifiers and real-time detectors. (Sivaguru et al. 2018.) Retrospective classifiers use large set of domains as input and are designed as a reactionary system. This type of systems can often use contextual data to improve the detections, such as HTTP headers and passive DNS. (Woodbridge, Anderson, Ahuja & Grant 2016.) Real-time detectors attempt to classify using only the domain name string. Some use human provided features that are fed to the ML models, such as entropy or vowel to consonant ratio. Other approaches use DL for feature-less detection. (Woodbridge, Anderson et al. 2016.)

3.5.3 Challenges of ML in cybersecurity

This chapter explains some challenges that ML in cybersecurity faces, especially when compared to other market areas where ML has gained good success.

First is the problem with testing and training on pre-generated datasets, does it really give proper results for real-world situations? According to Sommer and Paxson (2010), systems that aim to find novel attacks only using simulated activity will often lack required realism or relevance.

Known issue in traditional rule-based detection methods is the constant need to frequently update the definitions, for example daily antivirus definition updates. Similarly, when using supervised ML algorithms that require labeled training datasets, the systems need to be re-trained regularly as relying on outdated datasets lead to inadequate detection performance. Problem is the availability of sufficiently large and comprehensive datasets. (Apruzzese et al. 2018, 385.) Manually building this kind of datasets is expensive and organizations are reluctant to share real data from their networks for the fear it may contain sensitive information. Also sanitizing this real-world data might alter relevant factors in the data so that it no longer represents the real-world situation (Tuor et al. 2017). This makes regular re-training extremely difficult, or even impossible (Apruzzese et al. 2018, 385). One exception to this is the email spam filter where the example data is easier to acquire, and the complexity of the problem is significantly lower than in user behavior or network anomaly detection problems.

Unlike many other ML applications, security related implementations must also face constant arms-race against attackers as the attackers are figuring out ways to avoid detection even from machine learning based anomaly detection applications. (Sommer & Paxson 2010.)

ML based systems are also susceptible to attacks that are aimed specifically against the algorithms. These include methods such as adversarial inputs and data poisoning. Again, using the e-mail spam filter as example, the attackers try to craft e-mails that fool the model into classifying the spam as safe by obfuscating words that the model might consider bad and inserting words that it might consider good. On data poisoning the adversaries attempt to contaminate the training data to control the prediction behavior of the model (Ma, Xie, Li, Maciejewski 2019). This requires the attacker to have access to the training pipeline of the model.

Another aspect in cyber security is that false positive and false negative detections have very different costs than in some other machine learning applications. Sommer and Paxson (2010) made a comparison to a product recommendation system in online store: For system like this, false positive does not cause significant issues, the customer will most likely continue shopping as usual. False negatives will also not cause big issues, at most the buyer might miss a product they would have otherwise

bought. This low risk factor allows for less prudent tuning of the system. (Sommer & Paxson 2010.) *In contrast, failing to detect malware, a network intrusion or a phishing email can compromise an entire organization* (Apruzzese et al. 2018, 384).

In most machine learning based security systems the reported detections will also have to be analyzed by human analysts. This means having low false positive ratio is essential as it will otherwise flood the analysts with alerts hindering the detection and remediation in case of a real security incident. (Sommer & Paxson 2010.) In systems where the remediation is automated like email spam filter a false positive can cause important legitimate emails to not be delivered to end user (Apruzzese et al. 2018, 384). In case of antivirus detections, in worst cases the antivirus software can halt the whole production by blocking or deleting critical software used by the company.

Another problem in cyber security is often having to find anomalies instead of similarities. Machine learning shines in finding similarities between events, but finding "unknowns", never seen before attacks that can be even crafted specifically against the monitored environment is a different challenge. Using the same example of online shopping; if the system would operate in similar fashion as anomaly detection, instead of finding products that are usually bought together it would be finding products that are usually not bought together. This would provide a very different challenge as most product pairs do not have a common customer group. (Sommer & Paxson 2010.)

There is also the issue of recognizing the difference between non-benign and benign anomaly. Most anomalies will not be caused by an actual attack. The systems with no way to tell between these two will have high number of false positives and most likely will not be fit for production use. The system also needs to support the analysts in understanding the alerts it generates. If the system only reports the detections as “Traffic on the host did not match normal profile.”, it will take a lot of additional resources from the analyst to determine what has happened. (Sommer & Paxson 2010.)

4 User and Entity Behavior Analytics

UEBA solutions build profiles of standard behavioral patterns for the entities it monitors. The entity can be for example user, host, application, network traffic in specific segment or a data repository. In the case of QRadar UBA the only monitored entities are currently the users, which explains the missing 'E'.

The profile is built for the entity's past behavior and the behavior of its peer groups. Peer groups can either be defined, for example by job position, or learned based on the activities. New activities are then compared to these baselines using various analytics in aim to detect meaningful anomalies from the behavior.

There exist two types of UEBA systems; standalone products and systems where the UEBA is integrated to other security tools, such as SIEM. The market in this is moving towards the integrated systems, either by SIEM vendors implementing UEBA solutions or standalone UEBA vendors adding SIEM capabilities to their products. Now every major SIEM vendor has an UEBA solution available in their product. (Sadowski, Litan, Bussa & Phillips 2018.)

UEBA has close relation to SIEM since the data required by the systems is basically the same; the analytics are performed on the organizations security data that is typically collected and stored by a SIEM. This means adding UEBA to an already implemented SIEM solution does not necessarily require any additional data sources. (User and Entity Behavior Analytics n.d.)

The most prominent use cases for UEBA technologies in information security can be summarized in following categories:

Compromised insider and advanced threats

The goal is to detect and analyze attacks that have successfully infiltrated the organization by compromising legitimate user or service accounts and moving laterally inside the company's IT infrastructure. These can range from a simple compromised email account due to non-targeted phishing attack, to more advanced targeted attack by an APT group. Especially in latter cases the activities performed by the attackers once the infrastructure has been breached are notoriously hard to detect as the attackers often use zero-day attacks and other methods that don't have any known

indicators of compromise. These threats can have complex operation models and their behavior might not yet be recognized as bad. (Sadowski et al. 2018.)

Like stated earlier, this makes traditional detection methods that rely on known patterns, thresholds, or correlation rules ineffective in detecting these advanced threats. Many of these attacks still require some alteration to the compromised entity's behavior which allows systems using UEBA to have a change to detect and alert of the suspicious behavior. (Sadowski et al. 2018.)

Data exfiltration

Detecting exfiltration of data in organizations. UEBA can be used to enhance existing DLP systems with advanced analytics, reducing their false positive rates and aiding in alert prioritization. Additionally, analytics on web proxy, firewall or endpoint data can be used to identify anomalous data transfers indicating possible exfiltration.

Identity and privileged access management (IAM and PAM)

Monitoring user behavior in relation to established access rights by detecting excessive privileges and abnormal accesses. UEBA can also be used to find and remove dormant accounts and user privileges that are set higher than needed. (Sadowski et al. 2018.)

Malicious insider

This is similar as the first use case, but difference is that the user account is not necessarily used by an outsider who has compromised the account, but rather a person who has authorized access to organization's critical information or systems misusing this access. This person is not necessarily an employee of the company, it could also be a contractor, third party vendor or a partner. (Sadowski et al. 2018.)

Incident prioritization

UEBA can be used to prioritize alerts that are generated by already existing security solutions across the organization. The techniques used by UEBA, like baselining behavior, helps to understand when incidents are especially abnormal. UEBA can also enrich the detections by contextual information. For example, information about the asset's criticality and user's role and access levels. Also, since UEBA shifts the incident

analysis from each individual alert to the riskiest entities, this can be used to lower the total volume of alerts that needs to be individually investigated.

Incident analysis process

While UEBA aims to increase the ability to detect security incidents, it also provides the analysts more information than regular rule-based alerts to aid in the investigation. Whenever an alert rises related to an entity monitored by UEBA, the system's dashboards can be used to help the investigation as they contain the data and graphs of the entity's past activities accumulated by the system. Easily readable and visually enhanced presentation of the entity's behavior can help the analyst to determine if the anomalous activity is true positive security incident.

4.1 Risk score and analytics methods

UEBA solutions use multiple types of analytics in conjunction. These include advanced analytics and traditional rule-based analytics that are combined to create a "defense in depth" approach. The events are analyzed in layers going from simple traditional analytics, including pattern matching and correlation rules, to more advanced analytics including all three main paradigms of ML: supervised learning, unsupervised learning, and reinforcement learning. (Sadowski et al. 2018.)

In UEBA supervised learning can be used to feed the system with sets of known good behavior and known bad behavior so it can learn to recognize the difference. Problem with this approach is the need to have labeled datasets with comprehensive examples of both. And even if this kind of dataset was available, this makes these analytics only able to recognize known bad behavior. Alternate approach is to use rule-based approach in conjunction with ML to create Bayesian networks. (Sadowski et al. 2018.)

Unsupervised learning in UEBA is used to learn normal behavior and detect anomalies from it. These analytics will not know if the anomalous behavior is good or bad, only if it is anomalous. This leaves the interpretation to analysts, but without any differentiation between good or bad anomaly this would have unacceptable false-positive ratio. UEBA vendors aim improve this ratio by focusing the analytics on events that have higher change of being indicative of threat. (Sadowski et al. 2018.)

Risk score

Rule based SIEM analytics are mostly deterministic by nature. This means whenever a rule triggers an alert is generated. Advanced analytics are heuristic by nature; the models compute the probability that the event is anomaly, how large the deviation from normal is and how likely it is indicative of a threat. These detections cannot be as binary since heuristic alerts are not guaranteed to be optimal. Therefore, a scoring system is often used. Each detection is given a value, for example within scale 0-100 depending on the likelihood and severity of the risk. (Sadowski et al. 2018.)

In UEBA the initial rules and analytics are independent indicators that each have a risk score associated. This individual risk score is not used as a basis of alert but are calculated together to form a total risk score for the entity. This can be counted for single user session, for a specific timeframe, or using system like in QRadar UBA, where the risk score does not have specific time or session frame, but decays over time to avoid inflation of the risk score. This use of risk score changes the focus from analyzing each individual alert to analyzing the riskiest entities (Sadowski et al. 2018).

To illustrate how the risk score can be calculated by UEBA is this is example from Exabeam UEBA, and very similar system is used by other vendors.

First the system collects independent indicators from the ingested log data. Some are based on statistical analysis, for example user accessing asset abnormally. Some indicators use pre-generated alerts from security devices, for instance an alert of malware deleted from host by antivirus software. Others involve ML such as detecting accesses to DGA domains. (Lin 2017.) These indicators can also include traditional pattern-matching, like user accessing domain that has been flagged as malicious by threat intelligence.

These initial indicators are meant to be statistically independent and easy to interpret (Lin 2017). This initial phase can also be considered as feature extraction, where the initial mass of log data is reduced to more manageable and relevant set for processing.

The risk score is used to fuse the output from these indicators together. In Exabeam UEBA this risk score is calculated for user sessions. Each indicator is given an initial

risk score called *anchor score*. This is assigned by human experts based on the relevance of the indicator.

There are few mechanics to make this approach more viable in varying environments. Since some indicators trigger more often across the userbase, often due to environment-specific reasons and some trigger more often for specific user accounts, this could lead to inflation of the risk score. Indicators that trigger more frequently have lesser informative value in security context. (Lin 2017.)

To mitigate the issue with risk score inflation, the initial anchor score is adjusted before applying it to the total score. First is comparing the triggering frequency to the user's past activities and the activities of the user's peer groups. If the indicator triggers for the first time for the user or its peer group, the risk score gets multiplied by a set factor. Another adjustment is based on Bayesian method that weighs the score of indicators by their past triggering frequency; the more frequent the indicator is the smaller its adjustment weight is. The final session score is the sum of the weighed scores of triggered indicators and alerts are only created if the session score reaches a certain set threshold. (Lin 2017.)

The use of scoring system like this has several advantages over generating an alert each time a rule is matched. First is the automatic correlation of all the events by associating them to the related entity, all the indicators that contributed to the total score will be included in the generated alert. Even though there is no requirement that the events are related, if significant number of anomalous events happen in close conjunction for the same entity, there is a good change they are. At least to the point that it is worth investigating if there is a common malicious denominator with the individual anomalies.

Use of risk score can also reduce the number of alerts that needs to be analyzed as each matching rule or analytic does not need to be analyzed individually. This also allows to monitor frequent low severity events without having to specify thresholds upon which to create an alert for each individual event, or the context which they are abnormal in, as this is learned by the baseline. When creating rules that generate risk score instead of a standalone alert the rules do not need to be as strictly tuned and

the logic can often be simpler as the system is designed in a way that automatically lessens the impact of false positive hits by understanding when it is normal.

Since single risky event in UEBA should never create alert by itself, there is still need for traditional correlation rules. These are cost-effective way to detect known bad user behavior, malware with known signatures or activities that are against company policies.

There also is not always a guarantee that the visibility to the monitored environment from the logs in SIEM is good enough to catch the whole attack pattern from the start to finish. It is very possible that there are only few log events among the mass that are the only available indicators of a successful attack. In cases like this, UEBA would not help in detection.

4.2 User Behavior Analytics for QRadar

UBA for QRadar is a separate application that is installed as an addon to existing QRadar SIEM deployments to provide two additional functions: risk profiling and unified user identities. UBA uses the existing QRadar database, interface, and rule engine to create new type of monitoring based around the risk profiles of users. (User Behavior Analytics for QRadar n.d.) UBA can be used without the Machine Learning Analytics app, in which case the UBA specific rules simply increase the risk score by a static amount when matching for the related user. Unified user identities means combining different username formats from logs to same profile using contextual data, mainly available from LDAP integration. UBA also includes dashboards to show the output of this process to analysts.

4.2.1 Used data and LDAP integration

First step of the process is collecting the data of users' activities. This is the same log data that is already ingested by the SIEM and large portion of this data is usable for UBA. Most relevant events are all the log messages that contain the information about the related user in the payload. Problem is that different log sources use different attributes as username even when the actor is the same entity. Example of

this is Office 365 suite using the User Principal Name as the username while Windows security event logs often use sAMAccountName. This means you cannot simply use the username string as the key.

For this issue QRadar UBA has the User Import wizard which can be used to import contextual data from multiple identity sources. With this QRadar can poll the data from LDAP servers, or it can be imported from a CSV file. (Configure user import. N.d.)

This data can contain all the LDAP attributes available from the server. These can include for example *userPrincipalName, cn, sn, mail* and *department*. From this UBA creates a profile for each user and maps all the uniquely identifiable attributes to the same profile. Another use of this data is grouping the users by non-unique values, such as group membership, country, or department. These groups can be used for peer group analysis.

Many log events do not contain the username and often the source IP is the only property that indicates the actor. For this it is possible to enable feature to search assets for username when username is not available for event or flow data. Only when the username is not found from the asset data the event is ignored. (IBM QRadar User Behavior Analytics (UBA) 2021.) The asset database is built by QRadar from events that contain asset information, such as authentication events that contain both source IP address and username or DHCP events that contain the assigned IP address, hostname, and MAC address.

Problem with this feature is the reliability of the asset data that is automatically generate by the QRadar asset profiler. From experience, it is quite common that dynamic IP addresses have incorrect user in the asset profile. For the reliability of UBA detections it is very important that risk score is not added for wrong users.

Even with this feature disabled, there might be some issues with certain log events that include the username when that username is mapped to the event by similar asset profiler by the log source. Examples of this are user mapping by next-generation firewalls, such as Paloalto's *User-ID*. Some of these systems provide reliable user-to-IP mapping, but depending on the product and how it has been setup, these are also

known to have wrong username from time to time. Hence, the reliability of the username information on the log event needs to be assessed before trusting it fully.

4.2.2 UBA process and rules

IBM described the process of analytics in following three step model:

1. Pattern matching
2. Machine learning and risk scores
3. Output to analysts

(Open Mic: User Behavior Analytics 2018.)

Pattern matching

In the first step the UBA specific rules look for matching events. The UBA rules initially work exactly like other QRadar rules, using the same rule engine and same tests when looking for matching events. Difference is that when events match to UBA specific rules, instead of the usual rule response, it creates a S*ense Event*. This event will have a *senseValue* that is a numerical value defined in the rule to indicate the severity of the issue found. (Process overview n.d.)

UBA app then pulls this sense event, takes the *senseValue* and username from the event and increases that user's risk score by that amount. The more user violates the rule, more the risk score is increased. (Process overview n.d.) This is the basic functionality of the UBA rules if no Machine Learning Analytics app is used. How the ML application interconnects to this process is discussed in chapter 4.3.

When QRadar UBA is installed, it also installs a content package with pre-made UBA-specific rules and custom properties. UBA 4.1.0 comes with 124 pre-made rules, divided into categories seen in the content pack summary (see figure 3). The sum of custom rules in the table is greater than the unique count as some of the rules are in

multiple categories and the table includes building blocks and other helper rules.

Content Pack	Custom Rules	Reference Data	Custom Properties	Property Expressions	QID Records
Access and Authentication	37 42 (UBA 4.1.0)	15	4	9	22 25 (UBA 4.1.0)
Accounts and Privileges	32	5	2	9	12
Browsing Behavior	20	0	2	14	19
Cloud	16	2	5	6	12
DNS Analyzer	5	0	0	0	4
Domain Controller	15	5	13	26	11
Endpoint	24 (UBA 3.7.0) 22 (UBA 3.8.0)	7 (UBA 3.7.0) 6 (UBA 3.8.0)	10	17 (UBA 3.7.0) 38 (UBA 3.8.0)	13 (UBA 3.7.0) 12 (UBA 3.8.0)
Exfiltration	24 27 (UBA 4.1.0)	1	3	17	11 12 (UBA 4.1.0)
Geography	12	4	0	0	7
Network Traffic	3 (UBA 3.7.0) 4 (UBA 3.8.0)	2	1 (UBA 3.7.0) 2 (UBA 3.8.0)	3 (UBA 3.7.0) 8 (UBA 3.8.0)	3 (UBA 3.7.0) 4 (UBA 3.8.0)
Threat Intelligence	19	6	7	17	14

Figure 3. Details of the UBA content pack (UBA content pack summary n.d)

These are two examples of the pre-made rules:

UBA: Possible TGT Forgery

- Detects Kerberos TGTs that contain Domain Name anomalies. These possibly indicate tickets that are generated by using pass the ticket exploits.

UBA: Pass the Hash

- Detects Windows logon events that are possibly generated during pass the hash exploits.

(Rules and tuning for the UBA app n.d.)

UBA also supports custom content. Since UBA uses the same rule engine as other QRadar rules, existing rules can be integrated to UBA either by changing the rule response of the existing rule or cloning it as a new UBA specific rule. (Integrating new or existing QRadar content with the UBA app n.d.)

Risk score

Risk score is the sum of all the risk events detected by the UBA rules. When this risk score for a user exceeds a threshold that has been specified in the UBA settings, UBA sends an event that triggers *UBA: Create Offense* -rule which in turn causes an offense to be created for that user. This threshold can be specified as static value or using dynamic setting where the threshold is updated hourly based on the distribution of risk score.

The risk score is reduced over time if no new events occur by percentage set by *Decay risk by this factor per hour* setting. Default for this is 0.5 and increasing it accelerates how fast the risk decays. This is used to mitigate risk score inflation. (Process overview n.d.)

To illustrate how the risk score works in UBA is following figure 4.

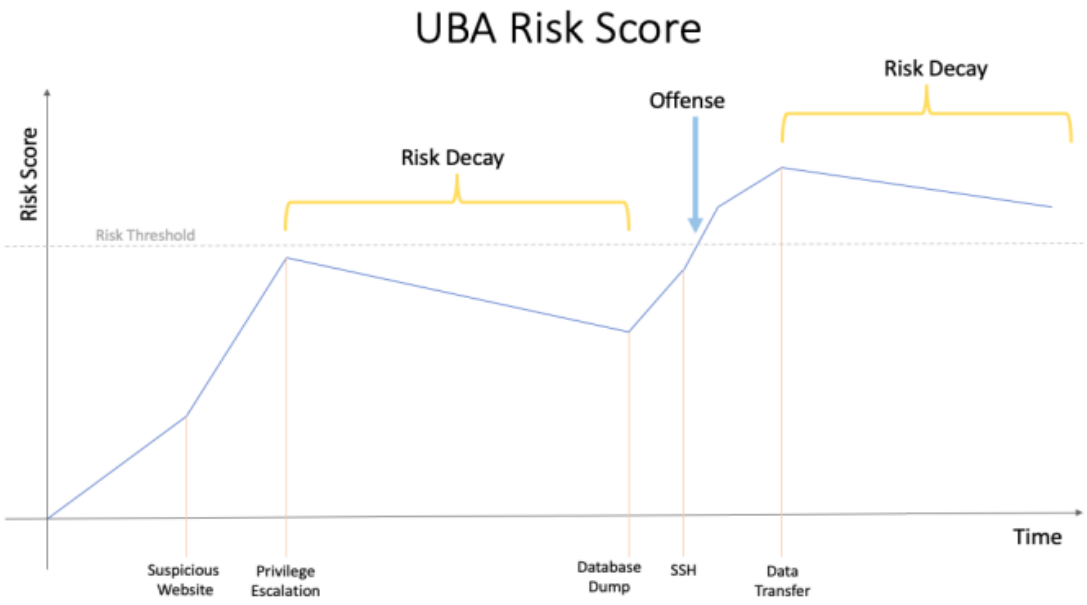

Figure 4. UBA Risk Score (Understanding the UBA Risk Score 2019.)

Output

Last part of the process is the output to analysts. Since UBA alerts usually consists of high number of different type of events, creating only text based alert descriptions or giving just a list of events to analysts would make analyzing the alerts slow and tedious process. For this UBA has dashboards with drilldowns to show what has caused the total risk score to accumulate. More on how the analysis process works and how it differs from analyzing regular QRadar offenses in chapter 4.4.

4.3 Machine Learning Analytics App for Qradar

The Machine Learning Analytics app extends the capabilities of QRadar by enabling advanced analytics such as time series predictive modeling and clustering and learning baselines of user behavior in the network. It also adds visualization in the UBA dashboards to show learned models of the current and expected behavior for users and alerts on points in time of deviations. (User Behavior Analytics for QRadar n.d.) The ML models are querying the same log data for the analytics as other QRadar rules.

ML model used by Machine Learning Analytics App

Following figure (see figure 5.) illustrates the used ML algorithms and the use case for them.

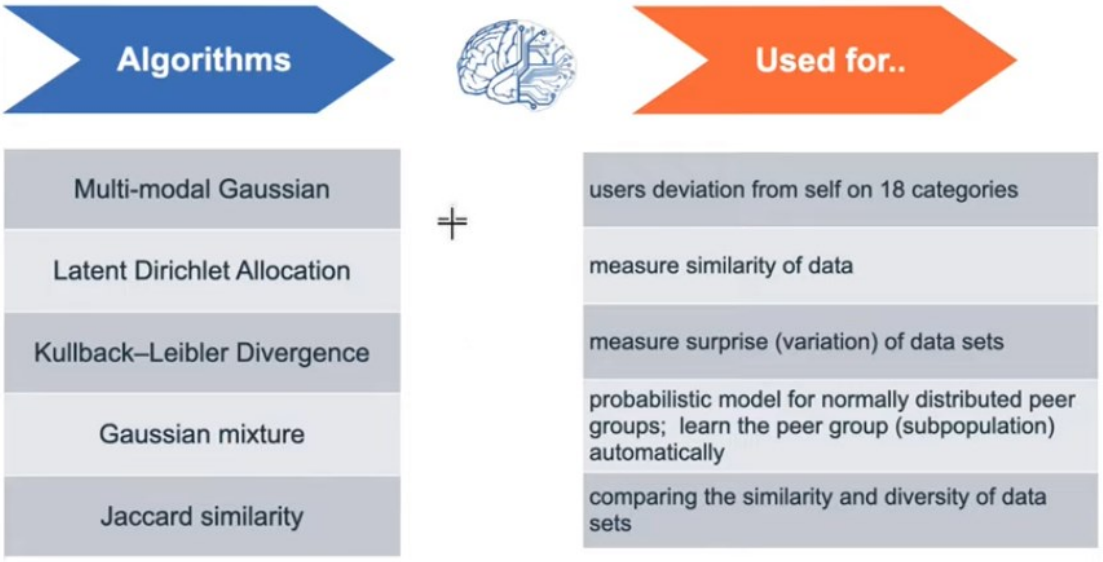

Figure 5. UBA's Machine Learning algorithms. (Bravo, J. 2020.)

The ML app with UBA version 4.1.0 comes with following 23 pre-made models divided in three categories:

Numeric user models

- **Access activity** - tracks a user's activity in the Access high-level category and creates a learned behavioral model for each hour of the day.
- **Aggregated Activity** - tracks a user's general activity by time and creates a model for the predicted weekly behavior patterns.
- **Authentication Activity** - tracks a user's activity in the Authentication high-level category and creates a learned behavioral model for each hour of day.

- **Data Downloaded** - monitors data that is downloaded for each user and then alerts on abnormal behavior.
- **Data Uploaded to Remote Networks** - monitors external domain data usage for each user and alerts on abnormal behavior.
- **DDL events** - tracks a user's DDL (Data Definition Language) events in time and creates a model for the predicted weekly score.
- **DML events** - tracks a user's DML (Data Manipulation Language) events in time and creates a model for the predicted weekly score.
- **HTTP Data Transfer Activity -** tracks a user's HTTP data transfer events in time and creates a model for the predicted weekly score.
- **Outbound Transfer Attempts** - monitors outbound traffic usage for each user and alerts on abnormal behavior.
- **Risk Posture** - tracks a user's risky activity by the rate of sense events generated and creates a baseline model.
- **Successful Access and Authentication Activity -** tracks a user's successful authentication and access events by time and creates a model for the predicted weekly behavior patterns.
- **Suspicious Activity -** tracks a user's activity in the Suspicious Activity high-level category and creates a learned behavioral model for each hour of the day.

(Individual (Numeric) user models 2020)

Observable user models

- **Lateral Movement: Internal Asset Usage -** tracks a user's internal destination asset activity by time and creates a model for the predicted weekly behavior patterns.
- **Lateral Movement: Internal Destination Port Activity** - tracks a user's activity to internal destination port activity by time and creates a model for the predicted weekly behavior patterns.
- **Lateral Movement: Network Zone Activity** - determines if a user's network zone is significantly different from the user's defined group.
- **Process Usage** - tracks a user's process usage activity in time and creates a model for the predicted weekly behavior patterns.

(Individual (Observable) user models 2020)

Peer group models

- **Activity Distribution** - learns behavior clusters based on LDAP group definition and searches for deviations from the normal distribution of these clusters over time.
- **Defined Peer Group** - shows how much a user's event activity deviates from the event activity of their defined peer group.
- **Internal Asset Access by Peer Group** - determines if a user's internal asset access is significantly different from the user's defined group.
- **Internal Destination Ports by Peer Group** - determines if a user's access to internal destination ports is significantly different from that user's defined group. If the user's access is deemed suspicious, a Sense Event is generated to increase the user's risk score.
- **Learned Peer Group** - identifies users who engage in similar activities and then places them into peer groups.
- **Network Zones by Peer Group** - determines if a user's network zone is significantly different from that user's defined group.

- **Process Execution by Peer Group** - determines if a user's process usage is significantly different from the user's defined group.

(Peer group models 2020)

The ML models use considerable amount of memory. Before deciding which models are used based on relevance to monitored environment, the size of the container and subsequent cost of upkeeping needs to also be assessed. Based on IBM documentation the maximum number of monitored users by any ML model is 40 000 per 5GB, up to 220 000 users total for 40GB (Machine Learning Analytics app n.d.)

Following settings for the models can be adjusted:

1. Risk value of sense event. This is the base amount how much risk score is increased when a sense event is triggered
2. Scaling of risk value. Optional setting that multiplies the risk value by a factor in range 1-10 depending how large the deviation is
3. Confidence interval to trigger anomaly. Percentage for how confident the ML algorithm needs to be before triggering an anomalous event. Default is 0.95
4. Data Retention Period. This defines how many days the model data is saved for. Default is 30 days
5. Show graph on User Details page. Define if a graph of the model should be displayed in User *Details page* in UBA
6. Group By field. For Peer Group and Activity Distribution models this defines which group is used by the model.
7. Optional AQL Search Filter. This can be used to narrow the data that the analytic queries for in QRadar. This can be used to reduce the data the analytic is using, for example by filtering out certain log source types or user groups.

(Machine learning user models n.d.)

The app also allows creation of custom models. All the custom models use time series to create a baseline of user activity by hour. To create a custom model, first step is to either choose a template which populates the search filter automatically or create a custom AQL query. There are three options in the model definition.

1. Property. Defines which property is used to build the model. For example, Source IP or Process
2. Function. The AQL function applied to the field. For example, COUNT, AVG or SUM
3. AQL search filter. Filter that is used to restrict the scope of the model to specific data.

This could create an example model with summary: This models the *COUNT* of the field *Destination IP* for users each hour. (Creating a custom model n.d.) Also, the same settings that can be used to adjust existing models are available for custom models.

Machine Learning Analytics app in UBA process

Most of the ML models are looking at the log data in QRadar, and as such working at same stage as the static UBA correlation rules. Just like the other UBA rules, whenever a ML model triggers an anomaly, it creates a S*ense Event.* But as the ML models are not deterministic by nature the *senseValue* in the event is adjusted based on the magnitude of the deviation.

Some of the models, namely *Risk Posture,* is looking at the output of other UBA rules and adjusting total risk score based on deviations on the expected rate of risk accumulation. This helps the system to fine-tune the otherwise static nature of non-ML based correlation rules which always increase the risk by set amount independent of user's normal behavior.

Most of the models have an individual graphs in the UBA dashboard showing the learned and actual behavior for the user. Following graph (see figure 6) is example of *Risk Posture* model.

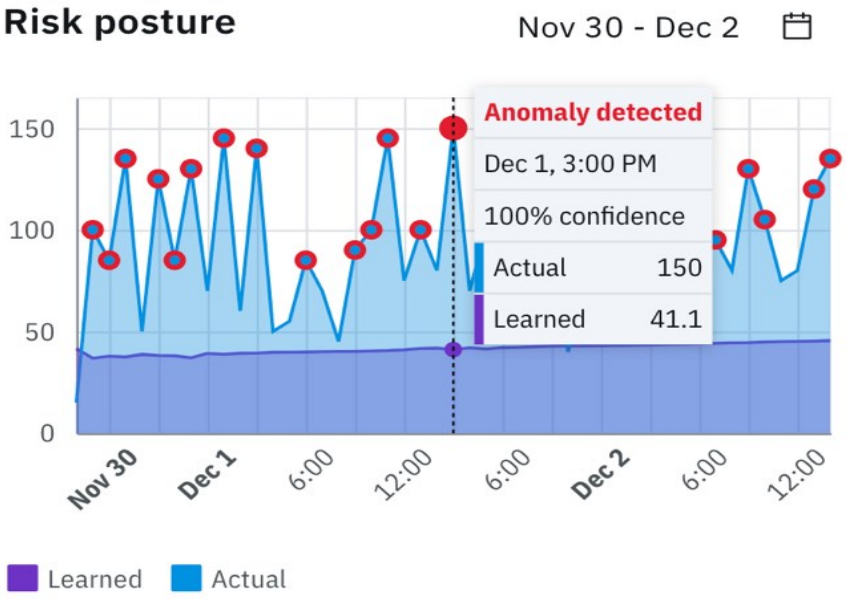

Figure 6. Risk posture (UBA dashboard with Machine Learning n.d.)

4.4 Investigating security incidents with UBA dashboard

UBA includes dashboards that are accessible from the QRadar user interface. The main view of the dashboard (figure 7) contains overview of the users and their risk

scores.

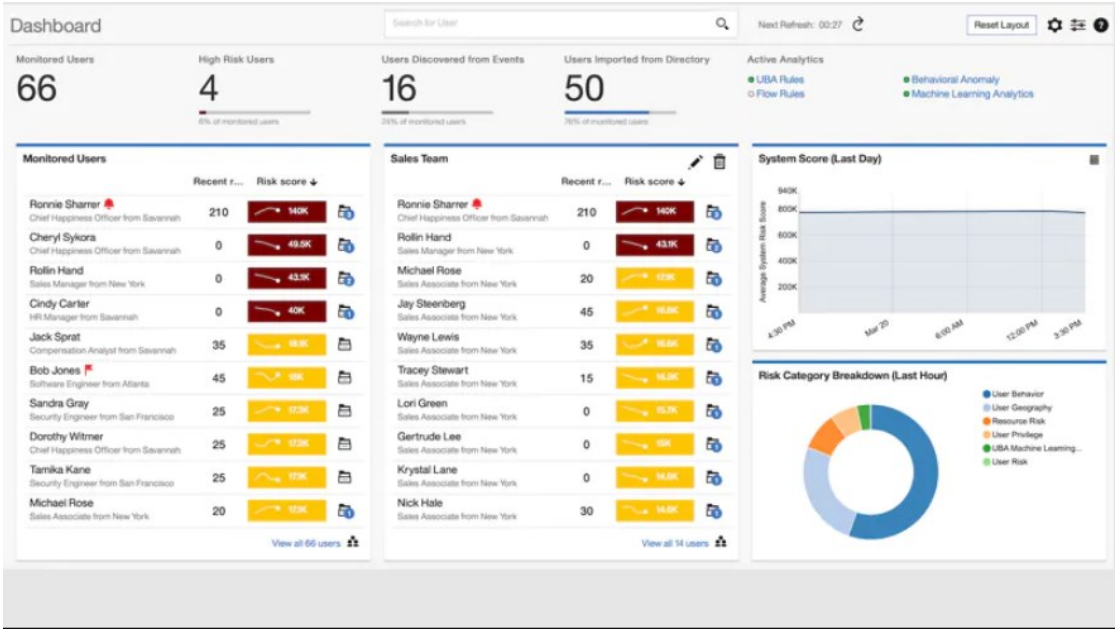

Figure 7. UBA overview (IBM QRadar User Behavior Analytics n.d.)

From this overview it is possible to access individual user's User Detail page by clicking on one of the users or by searching for a specific user. Example User Detail page seen in figure 8.

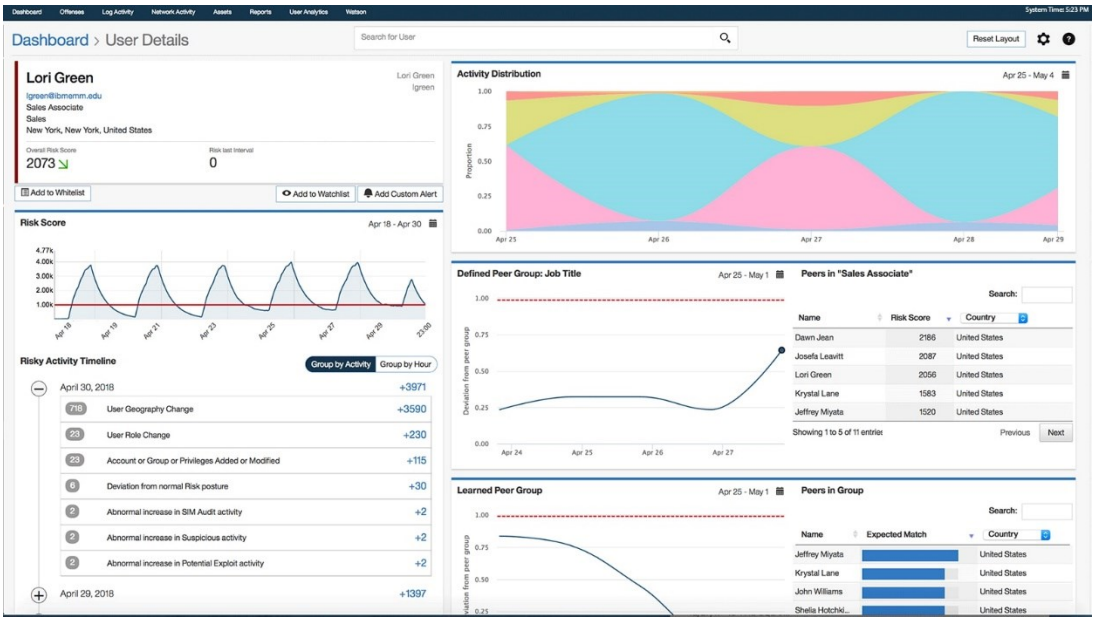

Figure 8 UBA User Details (IBM QRadar User Behavior Analytics n.d.)

The User Details page contains overview of all the information UBA has accumulated of the user. This includes the user profile compiled from the contextual data, graph of the trend of user's risk score, risky activity timeline which shows the total risk by

each contributing UBA rule, recent offenses related to the user and all the ML models that are set to generate an individual graph.

This page is also accessed when an offense has been generated by user breaching the alerting threshold. The overview this page provides is designed to give analysts quick first impression on why the risk threshold was breached.

For further analysis, any event on the timeline can be clicked to open an event viewer pane (see figure 9). This list all the log events that are associated with the activity. Clicking on individual event opens more details, such as the raw payload of the log event. Option at this point is also using “View in QRadar” button which opens the events in QRadar log activity tab. Analyst can then proceed to make their own queries to related log data.

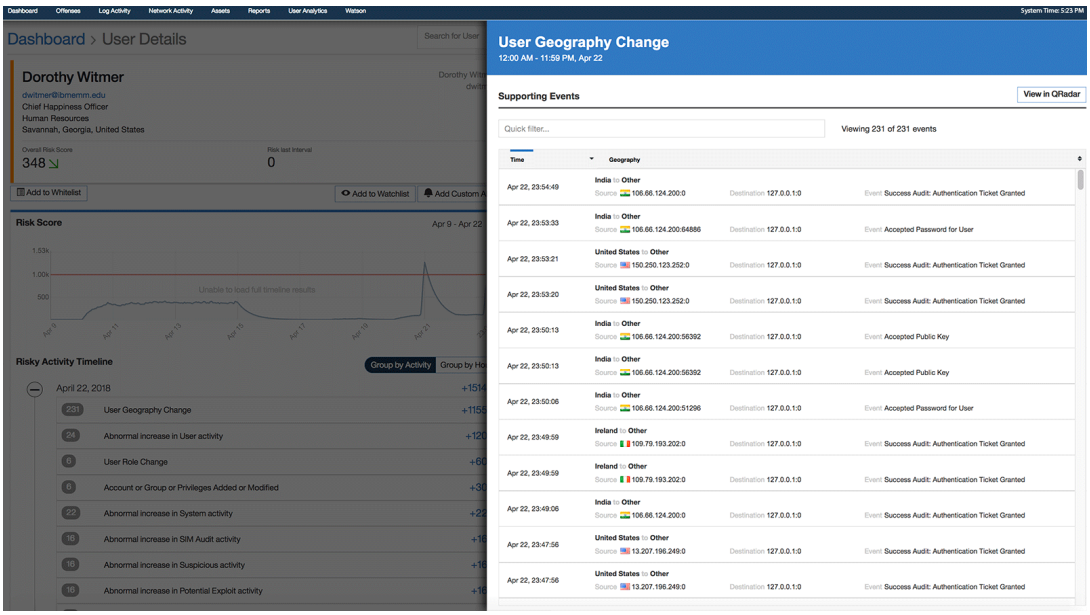

Figure 9 UBA Event viewer pane (IBM QRadar User Behavior Analytics n.d.)

Difference between analyzing UBA alert and normal QRadar offense

Initial triage for normal QRadar offenses is done by Nixu CDC using SOAR (Security Orchestration and Response) which is ingesting the offenses from QRadar. This includes growing amount of automation based on key details of the events that trigger the offense. This includes for example pre-made searches and details from events to populate ticket templates. In some simple use cases SOAR can process the alert completely automatically.

The UBA offense when created by the custom rule engine only includes the fact that the user breached the alerting threshold, and all analysis is expected to be done at the UBA dashboard. Looking at the UBA specific API at the point of writing this thesis it seems that there is no possibility to integrate the details from the dashboards to SOAR. This means that analysts will likely need to leave SOAR for the investigation in most cases.

One possibility for pre-made search used by SOAR is to display all the CRE events created by matched UBA rules and ML analytics. Each time any of the UBA analytics triggers it creates a new event containing the name of the analytic, related user and senseValue and is searchable from the log activity. SOAR could then search these events and display the count, or preferably sum of the senseValues of each recent analytic by the user. This is quite close how the UBA app itself creates the dashboards and sum of the risk score. Issue is that the unified user identities might be matched only at the point where UBA app ingests the Sense Events and not contained in the events themselves, in which case SOAR would also need to know the users' aliases.

While it may be that analyzing individual UBA alerts takes more time than that of regular QRadar rule, this is to be somewhat expected as each alert consists of much larger set of possibly correlating events. One of the ideas of risk-based alerting such as UBA is to lower the number of alerts that need to be analyzed so this might still be a net positive for analyst time spent. But without testing the system this can only be speculated.

Using the dashboards to analyze non-UBA specific rules is also possible. Some existing user related rules are currently challenging to thoroughly analyze, and UBA showing overview of the user's activity would be helpful for the investigation. Again, creating more conditions for analysts to leave SOAR is not ideal, but this could be used on Tier 2 or 3 phases when the initial triage does not provide enough information. It is also possible to use the same idea of pre-made search to show latest risky events in SOAR for username related offenses.

5 Plan for UBA deployment

According to Gartner's customer surveys *"Contrary to many vendor claims, UEBA solutions are not "set and forget" tools that can be up and running in days."* (Sadowski et al. 2018.) Gartner's clients have reported that it can take up to six months to get the UEBA system set up and tuned to such condition that it starts delivering on the use case it was deployed for.

One of the objectives of this thesis was to see how the system can be adjusted to different environments. This plan will include modifications that will be made for the pilot deployment and similar process can be used in future for other customers.

This plan will include:

- Reference data import settings.
- Process for choosing out-of-box use cases.
- Example custom use cases.
- Initial out-of-box ML models.
- Example of creating custom ML model.
- Process to define alerting threshold.

Will not include:

- Installation steps of the applications.
- Exact details of the customers applications.

5.1 Reference Data Import from LDAP

For the pilot deployment LDAP integration is created using the *User Import Wizard* with regular polling of data. This is a preferred method as the data is dynamic by nature; users and their roles change constantly. This also allows rules to be made that monitor the changes even when log event describing the change would not otherwise be available.

Setting up the LDAP integration naturally requires that the QRadar has access to the LDAP server. This can be defined either by hostname or directly with IP address. Another requirement is a bindDN: credentials that are used to authenticate against the LDAP.

After the initial setup of LDAP integration there are few modifications left for user. First is selecting which LDAP attributes are needed. Example by IBM of default values that work with Windows AD are following: *userPrincipalName, cn, sn, telephoneNumber, l, co, department, displayName, mail* and *title*. As this does not include group memberships adding *memberOf* might be a good idea, as this information is often relevant. This may need some testing how it behaves, as a single user can be in hundreds of different groups. Currently there may not be possibility to filter inside an attribute, for example to only list high privileged groups such as Domain Admins.

Chosen attributes are:

- userPrincipalName
- cn (Common Name)
- sAMAccountName
- l (City)
- co (Country)
- department (Department)
- displayName
- mail (email address)
- title (job title)

To be tested:

- memberOf (Group membership)

Next is setting up user coalescing. This means choosing the uniquely identifiable attributes that are used to combine activities with different username formats under the same entity. It is important not to choose attributes that have shared values across users. Selecting shared attribute, like country, would cause UBA to combine all users with the same country as a single user.

User coalescing will be set for following attributes:

- userPrincipalName
- sAMAccountName
- cn
- mail

Display fields for the user profile page can also be customized. The user profile page lists fields like Display name, Full name, and Group membership. The LDAP attribute

which value are displayed in these fields can be selected. This will only affect the profile page and has no effect on the analytics.

The LDAP data import is optional feature, but it is very important for the complete functionality of UBA. Without it, users' different aliases such as email address and sAMAccountName are considered as different users and tracked separately. It also enables many analytics and rules otherwise unavailable, such as defined peer group analysis and rules monitoring critical users.

Reference data that is not available from LDAP

LDAP covers most of the required contextual information about the users, but there is some information used by the analytics that is not available from it. One is the definition of critical and executive users. Information is needed from the customer to define which users, groups or organization units fit into these categories.

Second is information about assets other than users. Reference sets used by default UBA rules include: *Executive assets, Critical assets, Jump servers, Honeytoken accounts* and *Domain Controllers*. Most of these reference sets need to be populated from other sources, but a lot of this data is not exclusive to UBA monitoring and should already be available in existing SIEM deployments.

5.2 Out-of-box rules

The rules used in UBA is the most important part that determines how well it works in the specific environment. First step in the process should be evaluating how many of the pre-made rules can be used. This needs to be determined first by going through the available log data, does the SIEM deployment have required log source types that the rules need?

Second part is determining which of the rules that have log data available for should be used. Some of the pre-made rules are monitoring events that are not interesting to us and borderline illegal in Finland, for example *UBA: Browsed to Job Search Website.* Profiling users by web browsing habits like this does not seem purposeful. Reason why this rule exists in the first place is that this application is marketed for in-

sider threat detection, and this could be considered elevated risk of disgruntled employee planning for example data theft. Instead, we will concentrate on analytics that work for detecting potentially compromised user accounts.

List was made from the existing rules divided into following categories: *Access and authentication, Accounts and privileges, Browsing behavior, Cloud, Domain controller, Endpoint, Exfiltration, Geography, Network traffic and attacks and Threat intelligence*.

From this list it was decided which use cases will be initially enabled in the pilot based on available log data and which type of rules were seen to fit the environment, initially based on my expertise as SOC analyst. This list was further reviewed with the pilot customer and agreed that the monitored rules suit the use case of the SOC monitoring in the company. This list is not provided in this work as it would reveal the monitoring scope of the organization, but outline is that 76 of total available 120 of rules were decided to be used. This does not include the rules added in latest two versions of UBA as this assessment was done before the release.

This is only the initial setup of rules. Many of the rules might look good on paper, but before testing it is hard to determine how well they work in the environment. Eventual list of enabled features will be different from the planned one.

IBM uses following maturity model as guidance on how to setup the initial rules.

Table 2 Maturity Model (Open Mic: User Behavior Analytics 2018)

	Start with	**Intermediate**	**Advanced**
Visibility into	Account usage, account and privilege changes	Behavior on the network	Behavior on endpoints and applications
Log sources	LDAP, AD, Access and Authentication logs	Network, Routers, Proxy, Firewall, servers, VPN, IPS	Endpoints, Cloud, Flows, Applications, Policy, EDR
Benefits	Detect anomalous activity in accounts access, usage, privilege changes etc.	Detect anomalous behaviors of users on the	Detect abnormal behavior at the endpoint and application level. See users' deviation

		network, usage of networked resources and assets	from self or from their peer groups
Example use cases	• New Account Use Detected • Dormant Account Used • User Access Failed • Access to Critical Assets • User Accessing Account from Anonymous Source • User Accessing Risky Resources • Account Access from Unusual Locations • Attempted use of Suspended Account • Pass the Hash	• Browsing behavior rules • First Privilege Escalation • Suspicious Privileged Activity (First Observed Use) • Suspicious Privileged Activity (Rarely Used Privilege) • User Access Login Anomaly • Persistent SSH session • Potential TGT Forgery • User Geography Change • User Access at Unusual Times	• Abnormal transfer to external domain • Access by Service Accounts • Restricted Program Usage • Machine Learning detect abnormal deviations in: Volume of activity, Frequency of Activity, Peer groups • Potential Spam/Phishing • VPN Access By Service or Machine Account • Scanning in Network

This seems like reasonable starting point, as enabling the rules in patches allow for easier tuning and seeing how much risk each rule or category amounts to. Enabling all available rules from the beginning would likely lead to unmanageable tuning process.

The starting point includes account usage and privilege change related use cases. Log data used for this comes mostly from AD and LDAP and required log sources for these use cases should be available for most QRadar deployments. Rule categories at this phase are: *Access and authentication, Accounts and privileges* and *Domain Controller.*

Intermediate phase includes rules looking at the network traffic. Most SIEM deployments have at least some of the needed log sources integrated, such as NGFW, proxy logs or a network sensor. Rule categories are *Browsing behavior, Geography* and *Threat intelligence*

Advanced phase contains rules monitoring end point and cloud activity. Example log sources could include EDR, AV, Sysmon and AWS logs. Rule categories for this phase are *Endpoint, Exfiltration* and *Cloud*

There is no direct recommendation how long each phase should take, but at least a full week per phase is needed to get any idea how the rules behave as different weekdays are expected to have different activity.

After each phase there is need for a check for possible tuning needs. This can include whitelisting users, user groups or log sources from a specific rule, changing the rule's *senseValue*, completely reworking the rule logic or just disabling a rule altogether. If any of the rules need significant modification to the rule logic it will be copied as a new rule and the old one will be disabled. This way if a version update changes the pre-made rules it will not remove the modifications. The rules replaced this way will be added to the custom rules maintained by Nixu.

Each phase of enabling new features will cause the average risk profile of users to go up. This means that the alerting threshold needs to be adjusted at each step, or only set after all the features has been enabled and running for sufficient time.

5.3 Custom rules

After choosing and tuning the pre-made rules next step is looking into what data is available for custom rules and which existing standard SIEM rules can be copied to be used in UBA. Even though having duplicate rules for creating an offense directly and UBA sense event does not make sense in alerting perspective, it helps with analyzing user related alerts as the UBA dashboard will include those events in the graphs for better overview.

There are some log categories that are common in many environments but not covered by the pre-made rule set. Custom rule set will be made to cover some of these which can be used for various deployments. Categories like this are logs of Antivirus and EDR (Endpoint Detection and Response) software.

These should fit well for UBA. Normally antivirus logs of successfully mitigated threats may not be worth of alerting alone as these usually do not require manual actions and would flood the SIEM with alerts. These events can still be considered risky especially when related to other suspicious activity, or just in high trend for the user's context.

Idea for antivirus is to have different rules with varying risk level depending on the threat type and response of the antivirus software. For example, successful deletion is less risky than failed deletion, and severe infection like backdoor or hacking tool significantly higher risk than heuristic alert of adware. Some AV engines already included severity score, which could be translated into risk score by some suitable factor.

For EDR software same type of use cases will be made. EDR software often use behavior-based detections with a wide variety of confidence. This is because many of the behavior-based alerts attempt to detect legitimate binaries used for malicious purposes. And the more confident the alert is, the more specific the detection rule is, and therefore more false negatives it will have. This is again the issue of detecting known bad behavior. For example, using specific fingerprint of known malicious PowerShell script versus detecting the use of custom PowerShell scripts in general. Latter is often used by legitimate administration while also common in attack frameworks. Both activities also have evolving payloads making specific whitelisting or blacklisting difficult.

The lower credibility alerts can still be true positive depending on the context. As these cannot be omitted completely, the normal method is using correlation rule-based detections that are relying on thresholds; only if multiple low credibility detections by EDR are seen in a specified time window an alert is created. Issue is that without ML or UBA this cannot take the context into account very accurately and having set threshold in specified time window has a risk of things going under radar.

This is another good example where UBA can be useful. The detections that are currently relying on threshold in a single correlation rule can be used to add risk score for the user. As the system compiles all suspicious activities from all rules it will also correlate to other events that are currently used in similar fashion in separate threshold rules, for example brute force activity, potential Kerberoasting requests and low credibility threat intelligence matches.

Custom software used by companies can also be easily included into UBA monitoring. These could include for example any applications where sensitive data is stored. Only

requirement is that the system logs audit events which include the username and the action performed, and that these events can be forwarded to SIEM.

The way some of the use cases and most of the ML models are made a lot of this data is automatically included in them. This is because these are made for normalized log data and use high-level categories, for example Access Activity model uses Access high-level category and Authentication Activity uses Authentication category. This naturally requires the log data in SIEM to be normalized accordingly.

Example custom rules

Few custom UBA specific rules were made for the pilot as example. These rules are made for Microsoft Defender for Endpoint Antivirus logs. Other antivirus vendors will also be added to make these more generic rules that works in various environments. The normalization for this already exists, but to show this process the existing material is ignored, and the rules made from a scratch.

This will include total of 6 rules each with different associated risk score. The rules are quite similar so only one is described in detail. Following attributes need to be parsed from the log event: *Username*, *Event Type, Action, Action Success* and *Priority.*

Rules and the risk scores could initially be set like follows:

- UBA: Low Severity Malware Mitigated, 5
- UBA: Low Severity Malware Mitigation Failed, 20
- UBA: Medium Severity Malware Mitigated, 20
- UBA: Medium Severity Malware Mitigation Failed, 40
- UBA: High Severity Malware Mitigated, 50
- UBA: High Severity Malware Mitigation Failed, 100

Example rule seen in figure 10.

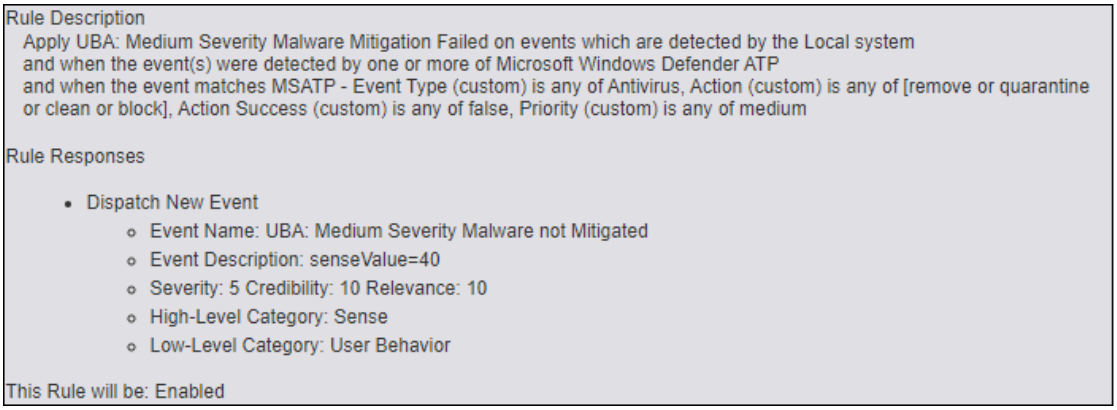

Figure 10. UBA: Medium Severity Malware Mitigation Failed

5.4 Machine Learning App

Out-of-box ML models

QRadar UBA is possible to be used without the ML app. But it is very integral part of the complete functionality of UBA. Without it there are no advanced analytics, such as learned baselines or peer group analysis.

The ML app comes with 23 pre-made models that are described in chapter 4.3 and a possibility to create custom models. First step again is to analyze how well the existing ML models work in the environment following the same steps as for the rules.

1. Is the data required by the models available?
2. Do the models support the use case which UBA is deployed for?
3. Create custom models for available data that is not covered by existing models.

After this assessment 11 of the pre-made models were decided to be used in the pilot. Enabling all the models at the same time might again not be optimal, but since it takes up to four weeks for some models to accumulate the required baseline data, using similar system where only a few are enabled at the same time and tuned before enabling the next patch would take months. Instead, the models will be enabled in close succession, only monitoring the performance overhead, initially using very low setting for *Risk value of the sense event.* This way the models can collect the baseline data without skewing the risk posture while being untuned.

Custom ML models

Custom ML model was planned for the pilot deployment to test how the process for custom model creation works. The model was made to monitor the customer's issue and project tracking and wiki software. Since the login events to this software are normalized under Authentication category, these are automatically monitored by Authentication Activity pre-made model.

The logs of these software include auditing for actions such as page edit, create, view or delete. Use case of the model is to detect anomalous usage of the software, which combined with other suspicious activity could indicate compromised user account illicitly accessing information.

The log includes a property *Event Type* which contains the action*,* which will be used by the model. Function used will simply be Count. The AQL filter is then made to concentrate the analytics on the specific log source types. This produces following ML model:

Summary: This models the *Count* of the field *Event Type* for users each hour

AQL filter: logsourceid in ('247','246') AND NOT REFERENCESETCONTAINS('RS: Whitelist username', username) AND NOT REFERENCESETCONTAINS('RS: Whitelist event type', "Event Type")

The AQL filter includes two reference sets that are empty by default, which can be used to whitelist usernames or certain event types if needed later. Changing the AQL filter when the model is running causes the model to be built again and all the old data will be discarded.

Rest of the settings will be as follows: Confidence interval to trigger anomaly will be set to default 0.95, data retention period will be set to 30 days, show graph on User Detail page will be enabled, risk value of the Sense Event will be set to 5 and scaling of the risk value will be enabled.

5.5 Scoring of use cases

The pre-made use cases by IBM have a scoring range between 5 to 25. This means that the riskiest possible event is only 5 times the risk of any low-risk event. This could lead to an issue where constant low risk events overshadow the rarer high-risk

events in the total risk score. On the other hand, having rules with extremely high risk-score in a way these would breach the risk threshold alone is not sensible; if there is an event known to be suspicious enough to warrant this a normal correlation rule can be made to always create an offense. Having the same event as very high-risk event in UBA would likely lead only to duplicate the alert. This would also clash with one of the purposes of UBA, compiling multiple suspicious events into alerts.

This could easily be changed by factoring the risk score of each use case to a more appropriate range, for example changing the scale to go from 5 to 100. Maintaining the lowest at 5 would retain the possibility to tune it down for some use cases if needed.

ML models will also need to be adjusted to reflect the change of the risk score made to UBA rules. Since the ML models have inherent property of having lower false positives due to learned baselines and taking the magnitude of the deviation into account, the risk score could be adjusted to have greater impact on the total risk than the normal UBA rules.

5.6 Watchlists and user groups

UBA allows a creation of watchlists that contain chosen sub-set of users. These watchlists can be populated by adding users individually, adding them from pre-existing reference sets or using regular expressions. The regular expression can be used to add all users matching a naming convention, for example “svc_” for service accounts.

Using watchlists can have various functions. First is a separate list in the dashboard where users of the watchlist are displayed. This allows to separately monitor the users of different groups. There is no functionality to have separate risk threshold for alerting.

The risk of the users in a watchlist can be scaled by factor. Default is one, but this can be changed to either lower the risk in case of certain group of users are a source of constant false positives or raised in case there is a need to monitor certain groups or users more closely.

ML tracking can be set for users in watchlists in following manner:

- High - Users are always tracked up to the maximum users per Machine Learning analytic.
- Normal - Users are tracked by highest risk after all the high users are included.
- Never - Users are not tracked by Machine Learning.

Initial use of watchlists will include the known service accounts and administrators. The reasoning behind this is that both groups behave differently from the most of the userbase and might require risk factor tuning to not overshadow rest of the users. Also, certain use cases might not work for service accounts that work for normal user accounts, since the pre-made list of use cases is tailored to monitor human user behavior. Having the accounts in separate watchlists allows to address these more easily.

The watchlists do not included a possibility to exclude groups from individual rules. If this needs to be done, the group needs to be whitelisted from the rule logic. ML analytics can also be tuned to exclude or only include certain users or groups by using AQL data filter functionality.

5.7 Alerting threshold

Alerting threshold can only be defined after all the use cases have been deployed and tuned, and after the risk scores have stabilized by the decay factor. Prior to testing it will be hard to estimate how long this will take.

Another way to tune the number of alerts is by adjusting the decay factor. Increasing this would have the effect of shortening the period in which the suspicious activities need to take place to generate an alert. This could negatively impact the change of detecting “low and slow” attacks that happen in a longer time frame.

For the pilot deployment the decay factor will be set as the default and no alerting will be set until all the use cases and ML models have been running for some time. After the risk scores have normalized the threshold for alerts will depend on the quality of detections. Initial plan is to use the static threshold, but both options may need to be tested.

Using the static option for alerting threshold will need re-examining the value with constant intervals since changes in the monitored environment can cause the risk

posture to change significantly. Few examples that can have considerable effect are introducing new rules, ML models or log sources - or decommissioning old ones. Especially in changes that cause the overall risk posture of the system to go down, there is a risk that the threshold will be left too high. False negatives are often harder to notice and react to than false positives.

Problem with the dynamic setting on the other hand is that it may force alerts to be created even when nothing interesting is happening if the generic risk posture of the system happens to be low. Daily false positives can cause analysts' trust to the system to decrease and make them lose the ardor for proper analysis. What will be ultimately the best approach is impossible to say without running the system first for sufficient time.

5.8 Tuning UBA

UBA will require constant upkeep and tuning, no matter how well the initial deployment was made.

Ratio between imported users and users discovered from events

If the user import is working correctly, this ratio should be around 90 percent, or even more being imported from Directory and rest discovered from events. This is because in almost all cases the LDAP should contain most of the users of the company, exceptions being only local user accounts. If the ratio is significantly skewed this could indicate either of following: LDAP import does not work as intended and does not import all the available data, or the usernames seen in events are in a format which does not match any of the LDAP attributes used for use coalescing. This makes UBA unable to associate users seen in events with existing profiles causing them to be imported as new users. (Bravo, J 2020.)

Indexing event properties

There are few event properties that are recommended to be indexed in QRadar: High Level Category, Low Level Category, senseValue, senseOverallScore and Username.

This is due to UBA heavily using these in searches and indexing will improve the performance. This will have some effect on the used disk space. (Enabling indexes to improve performance n.d.)

Tuning false positives

Even with the tuning done in the initial deployment, future changes in the environment may cause false positive risk to be created. It is important to address false positive alerts, especially in UBA as the alerting threshold will be set based on the risk posture, if rules cause significant amount of false positive risk score, the real risk will get drowned underneath it. This naturally means that only rising the risk threshold to lower alert numbers is not viable, as real alerts may then not then be triggered at all. Addressing false positive risk needs to be done by rule or ML model basis, using the steps described earlier in chapter 5.2.

6 Conclusions

As the pilot deployment was left out of the scope of this thesis the process for the deployment was based mostly on understanding of the system gained from the documentation. It is obvious that after running and testing the system the process will change, so spending substantial amount of time to create fine-grained process or attempting to analyze the usefulness of the analytics based on assumptions alone does not make too much sense. Therefore, this thesis became more of a look into the possibilities the product offers and what capabilities it has for including custom content.

The UBA does not contain any black-box analytics. All the analytics can be completely modified or made new from a scratch. This means that it has potential to be modified for any customer environment with enough work hours. Question then is that how much development time it needs before it becomes useful enough to warrant the extra cost. While the application itself is free to install for existing QRadar deployments, cost increases come from additional resources needed from hosting and work required from SOC developers. Based on the research done in this thesis, it is not

possible to conclusively state an answer to this, but the application does contain many features that would be very welcome as SOC analyst and developer.

The default rule set that comes with UBA is quite comprehensive for normal office environments, with only few categories that are completely missing. The quality of the rules is hard to assess without proper testing. Custom rule development done previously for QRadar is easy to integrate for UBA as the application uses mostly mechanics that are already familiar for people working with QRadar.

QRadar UBA seems to receive new updates with constant intervals. Latest major version was released in December 2020 and minor updates are release multiple times a year, latest in March 2021. Each version containing fair number of new features. It seems then that the product has healthy development cycle and expected to improve over time.

One of the largest additions would be making it UEBA instead of UBA. Currently limiting scope to only users can reduce the usability to some degree. Using for example IP address or hostname as entities would increase the percentage of log data that is usable for the system.

The analysis process by using UBA dashboards seems like a large improvement in many cases, especially compared to native QRadar offense analysis process. Only issue is its integration to SOAR being a question mark. The unified user identities is notable quality of life improvement, but how well it can be integrated to processes outside the UBA dashboards and analytics is still unknown.

The product is made by USA based company, where the laws and regulations concerning privacy and collecting data about users is much more lenient, especially after GDPR was introduced in EU. For companies operating in EU carefully assessing the legal aspect of using this type of software is important. It is also important for analysts using the system to not access the data collected about users without sound security related justification. Evaluating the legal aspect was not in the scope of this thesis.

Next step would be to test the system in production. Conclusion is that QRadar UBA is promising product, but how well it would detect compromised user accounts in real-life scenarios cannot be stated before testing it. Testing the product would need

to be done in real environment, so the background noise of normal user activity would be realistic. Testing the compromised user account scenario would optimally be done in purple team exercise, where advanced attack tactics would be simulated. For UBA this would provide one additional challenge, testing the usual security measures can be started initially with a fresh account that is given to the red team. For UBA the account would need to have a history of normal activity to present the most realistic scenario.

Another aspect to test is the ratio of false and true positive alerts. This will depend heavily on the deployed use cases, log sources and the nature of the monitored environment, as it does in all SOC monitoring. But it should be then comparable to the existing detection capabilities in the same environment. To get statistically significant results from this would take quite a long time.

References

Agrawal, K., Makwana, H. 2015. A Study on Critical Capabilities for Security Information and Event Management. International Journal of Science and Research, Vol 4, 1893-1896. Accessed on 4.5.2020 https://www.ijsr.net/archive/v4i7/15071503.pdf

Ahmad, Z., Khan, A.S., Shiang, C.W., Abdullah, J., Ahmad, F. 2020. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Survey Paper. Transactions on Emerging Telecommunications Technologies. 32. 10.1002/ett.4150.

Alpaydın, E. 2010. Introduction to Machine Learning. 2nd ed. MIT Press, Massachusetts Institute of Technology.

Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., Marchetti, M. 2018. On the Effectiveness of Machine and Deep Learning for Cyber Security. 2018 10th International Conference on Cyber Conflict. Tallinn, Estonia.

Bravo, J. 2020. UBA's Machine Learning in Action. Video uploaded in Youtube. Accessed on 20.4.2021. https://www.youtube.com/watch?v=bjvdbmOzdRg

Bravo, J. 2020. Tuning UBA Part One, Tips from Bruno. Video uploaded in Youtube. Accessed on 27.4.2021. https://www.youtube.com/watch?v=jhTzPUd9HG4

Basic LDAP Concepts N.d. Article at ldap.com website. Accessed 10.3.2021. https://ldap.com/learn-about-ldap/

Carder, J. 2019. What is the Zero Trust Model for Cybersecurity, Really?. Blog. Accessed on 3.5.2020. https://logrhythm.com/blog/what-is-the-zero-trust-model-for-cybersecurity/.

Chuvakin, A. N.d. The Complete Guide to Log and Event Management. White paper sponsored by NetIQ.

Chuvakin, A. 2019. Security Correlation Then and Now: A Sad Truth About SIEM. Anton on Security blog on Medium website. Accessed on 12.3.2021. https://medium.com/anton-on-security/security-correlation-then-and-now-a-sad-truth-about-siem-fc5a1afb1001

Chumachenko, K. 2017. Machine learning methods for malware detection and classification. Bachelor's Thesis, XAMK. Information Technology, Kaakkois-Suomen ammattikorkeakoulu.

Configure user import. N.d. IBM documentation in ibm.com website. Accessed on 17.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=app-configure-user-import

Creating a custom model. N.d. Documentation in ibm.com website. Accessed on 27.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=models-creating-custom-model

CrowdStrike Cyber Intrusion Services Casebook. 2018.

Enabling indexes to improve performance. N.d. UBA Documentation in ibm.com website. https://www.ibm.com/docs/en/qradar-common?topic=tuning-enabling-indexes-improve-performance

Feng, W., Wu, S., Li, X. & Kunkle, K. 2017. A Deep Belief Network Based Machine Learning System for Risky Host Detection.

Ford, V., Siraj, A. 2014. Applications of Machine Learning in Cyber Security. 27th International Conference on Computer Applications in Industry and Engineering, Caine 2014.

IBM QRadar User Behavior Analytics. N.d. Overview of IBM UBA product on ibm.com website. Accessed on 26.4.2021. https://www.ibm.com/products/qradar-user-behavior-analytics

IBM QRadar building blocks. N.d. QRadar documentation on ibm.com webstie. Accessed on 28.4.2021. https://www.ibm.com/docs/en/qsip/7.4?topic=phase-qradar-building-blocks

IDG Security Priorities Study. Executive Summary. 2018.

Individual (Numeric) user models. 2020. QRadar UBA documentation in ibm.com website. Accessed on 26.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=models-individual-numeric-user

Individual (Observable) user models. 2020. QRadar UBA documentation in ibm.com website. Accessed on 26.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=models-individual-observable-user

Integrating new or existing QRadar content with the UBA app. N.d. QRadar UBA documentation in ibm.com website. Accessed on 26.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=tuning-integrating-new-existing-qradar-content-uba-app

Kavlakoglu, E. 2020. AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?. Blog on ibm.com website. Accessed on 10.3.2021. https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks.

Kent, K., Souppaya, M. 2006. Guide to Computer Security Log Management. NIST Special Publication 800-92.

Kothari, P. 2018. Understanding Zero Trust: A New Strategy for Cyber Defense. Accessed on 23.1.2020. https://misti.com/infosec-insider/understanding-zero-trust-a-new-strategy-for-cyber-defense.

Lane, A. 2010. Understanding and Selecting SIEM/LM: Aggregation, Normalization, and Enrichment. Blog. https://securosis.com/blog/understanding-and-selecting-siem-lm-aggregation-normalization-and-enrichmen

Learn About LDAP. N.d. Article at ldap.com website. Accessed 10.3.2021. https://ldap.com/learn-about-ldap/

Lin, D. 2017. A User and Entity Behavior Analytics Scoring System Explained. Blog on exabeam website. Accessed on 13.4.2020. https://www.exabeam.com/ueba/user-entity-behavior-analytics-scoring-system-explained/

Ma, Y., Xie, T., Li, J., Maciejewski, R. 2019. Explaining Vulnerabilities to Adversarial Machine Learning through Visual Analytics. IEEE Transactions on Visualization and Computer Graphics, Vol 26, Issue 1

Machine Learning Analytics app. N.d. Documentation in ibm.com website. Accessed on 27.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=app-machine-learning-analytics

Machine learning user models. N.d. Documentation in ibm.com website. Accessed on 27.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=app-machine-learning-user-models

Nixu Corporation. N.d. Accessed on 11.4.2021. https://www.nixu.com/about.

Peer group models. 2020. QRadar UBA documentation in ibm.com website. Accessed on 26.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=models-peer-group

Polyakov, A. 2018. Machine Learning for Cybersecurity 101. Blog in towardsdatascience.com website. Accessed on 15.12.2020 https://towardsdatascience.com/machine-learning-for-cybersecurity-101-7822b802790b.

Process overview. N.d. QRadar UBA documentation in ibm.com website. Accessed on 26.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=analytics-process-overview

Q1 Labs Inc. N.d. Article on unb.ca website. Accessed on 28.4.2021. https://www.unb.ca/research/partner/successstories/q1-labs.html

QRadar architecture overview. N.d. QRadar documentation on ibm.com website. Accessed on 28.4.2021. https://www.ibm.com/docs/en/qsip/7.4?topic=deployment-qradar-architecture-overview)

QRadar Architecture and Deployment Guide. N.d. Accessed on 28.4.2021. https://www.ibm.com/docs/en/SS42VS_7.3.3/com.ibm.qradar.doc/b_siem_deployment.pdf

Roman, V. 2019. Unsupervised Machine Learning: Clustering Analysis. Blog on towardsdatascience.com. Accessed on 4.5.2020. https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e

Rot, A., Olszewski, B. 2017. Advanced Persistent Threats Attacks in Cyberspace. Threats, Vulnerabilities, Methods of Protection. Position papers of the Federated Conference on Computer Science and Information Systems, Vol 13, 115.

Rules and tuning for the UBA app. N.d. UBA documentation on ibm.com website. Accessed on 27.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=app-rules-tuning-uba

Sadowski, G., Litan, L., Bussa, T., Phillips, T. 2018. Market Guide for User and Entity Behavior Analytics. Gartner analyst report.

Security Information & Event Management (SIEM) Market: Growth, Trends and Forecasts (2019-2024) - ResearchAndMarkets.com. 2019. Summary of ResearchAndMarkets.com's report on businesswire.com website. Accessed on 5.5.2020. https://www.businesswire.com/news/home/20190729005302/en/Security-Information-Event-Management-SIEM-Market-Growth

SIEM Architecture: Technology, Process and Data, N.d. Article in Exabeam's website. https://www.exabeam.com/siem-guide/siem-architecture/

Sivaguru, R., Choudhary, C., Yu, B., Tymchenko, V., Nascimento, A., De Cock, M. 2018. An Evaluation of DGA Classifiers. 2018 IEEE International Conference on Big Data.

Sobers, R. 2020. The Difference Between Active Directory and LDAP. Blog. https://www.varonis.com/blog/the-difference-between-active-directory-and-ldap/

Sommer, R., Paxson, V. 2010. Outside the Closed World: On Using Machine Learning For Network Intrusion Detection. 2010 IEEE Symposium on Security and Privacy.

The State of Zero Trust Security in Global Organizations. 2020. Article about survey comissioned by Okta. Accessed on 27.4.2021. https://www.okta.com/resources/reports/state-of-zero-trust-security-in-global-organizations/

Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N. & Robinson, S. 2017. Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams. AI for Cyber Security Workshop at AAAI 2017. Accessed on 2.4.2020.

Todd, B. 2017. Creating a Logging Infrastructure. SANS Institute Information Security Reading Room. Accessed on 2.1.2020. https://www.sans.org/reading-room/whitepapers/logging/creating-logging-infrastructure-38130.

UBA content pack summary. N.d. Documentation in ibm.com website. Accessed on 17.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=app-uba-content-pack-summary

UBA dashboard with Machine Learning. N.d. UBA documentation at ibm.com website. Accessed on 24.4.2021. https://www.ibm.com/docs/en/qradar-common?topic=app-uba-dashboard-machine-learning

Understanding the UBA Risk Score. 2019. Blog post in qradarinsights.com. Accessed on 23.4.2021. https://qradarinsights.com/2019/03/15/understanding-the-uba-risk-score/

User and Entity Behavior Analytics. N.d. Article on exabeam.com website. Accessed on 11.5.2020 https://www.exabeam.com/siem-guide/ueba/.

User Behavior Analytics for QRadar. N.d. Overview of User Behavior Analytics on IBM X-Force App Exchange. Accessed on 17.4.2021. https://exchange.xforce.ibmcloud.com/hub/extension/IBMQRadar:UserBehaviorAnalytics

Wang, J. 2017. Deep Learning in Security: An Empirical Example in User & Entity Behavior Analytics (UEBA). Spark Summit 2017. Accessed on 3.2.2020. https://www.youtube.com/watch?v=aAhAJFk1OVc.

Woodbridge, J., Anderson, H., S., Ahuja, A., Grant, D. 2016. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks.

Wu, E. 2020. The Tricks of Our Trade: How Reveal(x) Uses Machine Learning. Blog on extrahop.com website. Accessed on 10.3.2021. https://www.extrahop.com/company/blog/2019/how-revealx-uses-advanced-machine-learning-ndr-product-explained/