Global Information Assurance Certification Paper

Applying Machine Learning Techniques to Measure Critical Security Controls

GIAC GCCC Gold Certification

Author: Balaji Balakrishnan, pingbalaji@gmail.com
Advisor: Adam Kliarsky
Accepted: September, 1st 2016

Abstract

Implementing and measuring Critical Security Controls (CSC) requires analyzing all data types (structured, semi-structured and unstructured). This implementation can be a daunting task. One of the goals of effective implementation of Critical Security Controls is to automate as much as possible. Machine learning techniques can help automate many of the measurements in Critical Security Controls. This paper proposes a method to integrate all types of data into a single data repository, extract relationships between different entities and perform machine learning to automate the analysis. This solution provides the security team the ability to analyze the information, and make data-driven security decisions.

1 Introduction

By implementing and maintaining the Critical Security Controls, defenders can reduce the attacker's opportunity by systematically reducing the attack surface. Implementing the Critical Security Controls is a daunting task. One of the goals of effective implementation of Critical Security Controls is to automate as much as possible. Machine learning techniques can help automate many of the measurements in Critical Security Controls such as ranking the most vulnerable asset.

Summary of the key concepts proposed are:

1) Collect all the data from different tools and solutions in one central data repository/data lake. Data collection should include all data types including structured, unstructured and semi-structured data
2) Apply security ontologies to identify relationships between the different data elements in Resource Description Framework (RDF) triples (Marklogic/Spark) or property graphs (Neo4J/GraphX). Define security ontologies, extract entities and identify relationships between entities from all data sources
3) Apply analytical and machine learning techniques to predict and measure security outcomes and risk. Steps involved in applying machine learning algorithms are to visualize and combine data cleansing with clever feature engineering, choose right metric/method for estimating model performance and then spending a lot of time tuning the parameters.

1.1 Overall architecture of the solution

Security teams collect valuable information from different tools and solutions. In most implementations, different tools are in silos and do not integrate to paint a complete risk picture. Most of the times, integration of the data is performed manually by an analyst. At scale, when dealing with lots of data, this does not work.

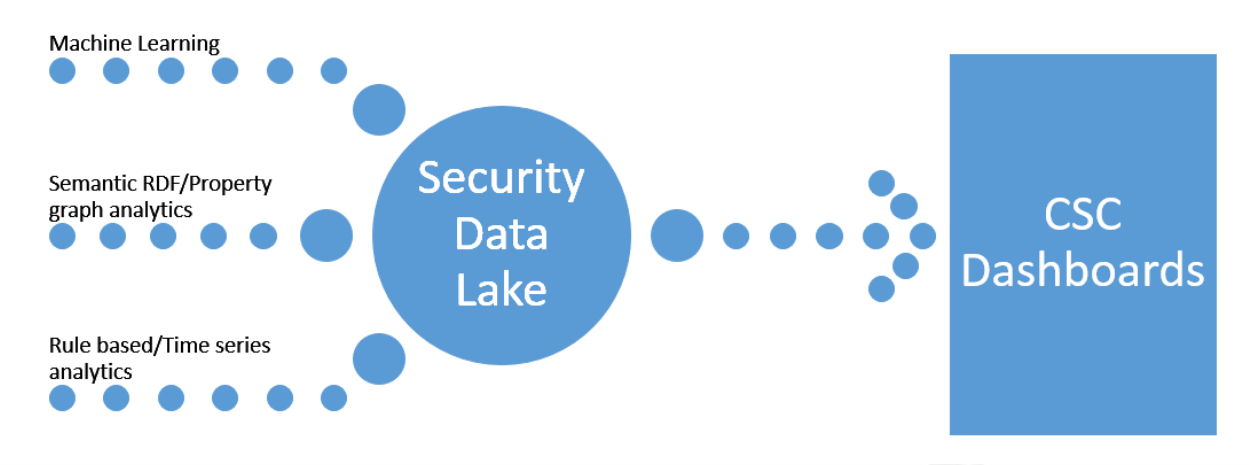

Figure 1 Conceptual diagram for the solution

Integrating all the data in one data repository, as shown above, provides the capability to the security team to analyze and derive data-driven security decisions.

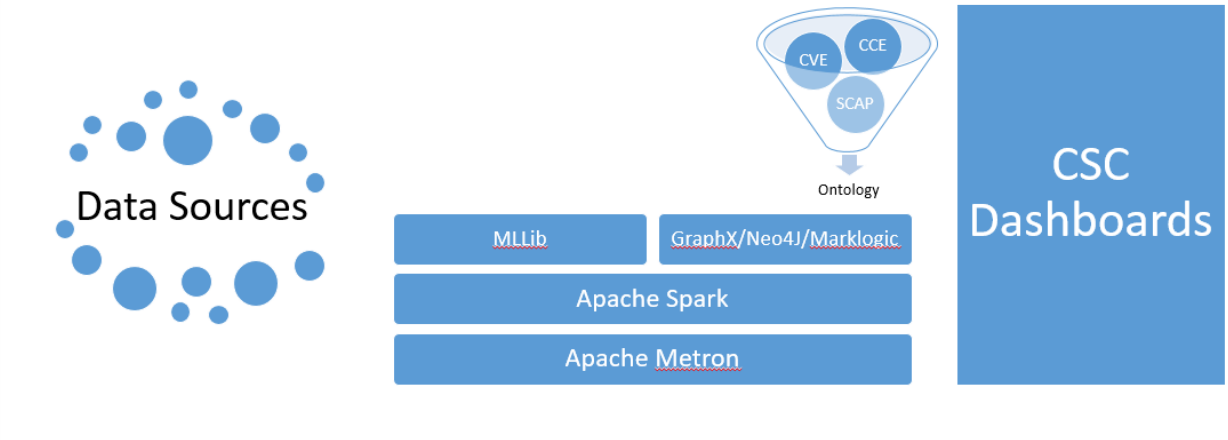

Figure 2 - Architecture Components of the solution

The first layer of the architecture is Apache Metron which is used to collect all relevant security data sources, including the PDF reports, JSON, and XML data. The next layer is Apache Spark processing the data. Apache Spark has additional libraries integrated into it to support a variety of data analysis and machine learning algorithms.

Some Security Information and Event Management (SIEM) solutions do not have the capability to capture unstructured data. There is valuable information in unstructured data. Traditional SIEM solutions are rule-based and do not support the application of machine learning algorithms to detect unknown threats. Some of the traditional SIEM solutions are proprietary and do not allow plug and play of various techniques. By

having a centralized open source big data analytics solution, it provides the capability to apply machine learning and semantically linked-data and other statistical techniques. Another big advantage of this solution is once a successful technique is identified using machine learning, many different problems can be solved using the same method. For example, if a solution is helpful in identifying suspicious access attempts from identity and access authentication data, the same technique can be applied to identify suspicious access attempts for cloud-based infrastructure.

Critical Security Controls 6.0 have clearly documented controls, sub-controls, and metrics which have entity relationship diagrams and provides information on all the data that needs to be collected. The metrics companion gives the measures required for a successful critical security controls implementation.

Following a structured process such as the Cross Industry Standard Process for Data Mining (CRISP-DM) to implement the CSC metrics ensures the success of the solution. CRISP-DM is a data mining process model that describes commonly used approaches that data mining experts use to tackle problems. Key steps involved in the process are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment (Jensen, 2013).

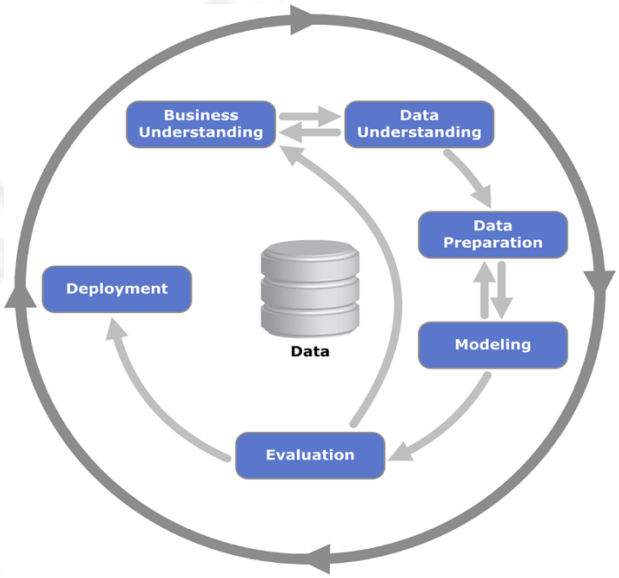

Figure 3- CRISP-DM (Jensen, 2013)

Some factors which make this solution feasible:

a) Advancement in big data technologies which enables us to store all types of data at scale. It provides the capability to extract information from

unstructured data and convert it to relationships and graphs for further analysis. For example, security awareness reports, penetration testing reports, and vulnerability analysis reports which are in PDF or XML format can be parsed (using natural language processing) and converted to RDF triples or property graphs for identifying relationships and ranking to prioritize the highest-risk assets.

b) Machine learning has always been there, but technologies like Apache Spark make it feasible to deploy Machine learning algorithms easily over different data types and use features from many different data sets. One of the other challenges with machine learning was false positives since there are not many features available with a single source (maybe network IDS). However, with recent advances in the integrations of multiple data sources to extract features, greater fidelity can be achieved.

c) Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) cloud technologies allow organizations to quickly and elastically deploy and manage big data platforms. These platforms allow the security teams to focus on creating rules and machine learning scripts to meet their needs and less on maintaining and supporting the infrastructure. IaaS/PaaS also reduce the dependency on having highly skilled resources to maintain big data infrastructure. Some examples are Microsoft Azure, Databricks, and Amazon AWS.

d) Many open source and commercial vendor solutions supporting Security Content Automation Protocol (SCAP) and similar standards enable integration of all the data seamlessly with common enumerations.

The first part of the paper will cover how open source big data platforms (Apache Metron), semantic linked-data and machine learning techniques can be applied to automate the implementation of the critical security controls. This part describes how the solution can be implemented with open source big data analytics solutions. The second part will implement a prototype using Azure ML Studio, R, and other tools. This paper also provides examples on how to perform advanced analytics and machine learning techniques to measure the effectiveness of Critical Security Controls.

2 Apache Metron - Big Data Analytics Platform

Apache Metron is highlighted as the big data security analytics platform since it is focused on cyber security. It makes data collection simple with Apache nifi. The data is also stored as real time index (SOLR) for search and Hbase/Hive storage for performing advanced analytics.

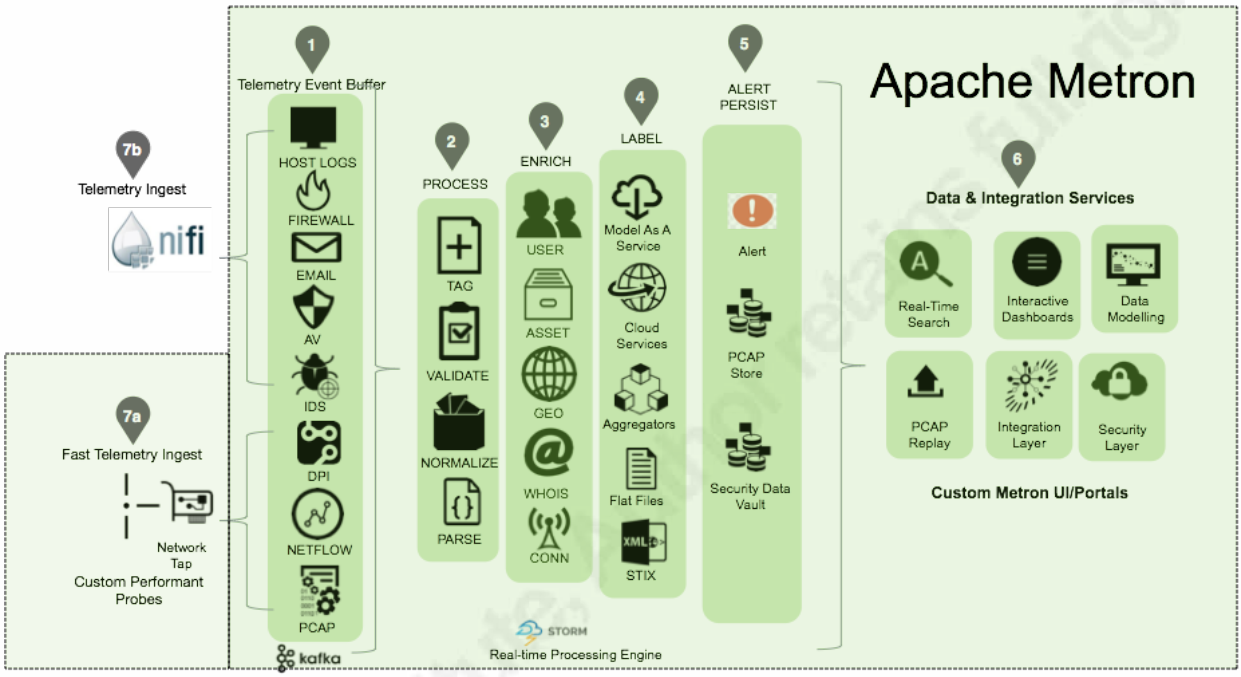

Figure 4- Metron Architecture (Apache, 2016)

Any big data solution which supports all data types can be used. Apache Metron was used as an example since it is security focused. Metron provides a good architectural framework required for big data security analytics platform including all components such as data ingestion, real-time processing engines, and data storage. Most big data platforms, open source or commercial, support NoSQL databases. These can be used to collect all of the relevant security data and store it in analytics-friendly formats.

Some data sources which are required for implementing and measuring critical security controls are Intrusion Detection Systems (IDS), Netflow, sniffer traffic dumps, firewall, Data Loss Prevention (DLP) and other logs from security applications. The entity relationship diagrams for the critical security controls provides guidance on choosing data sources. The data sources that are required to be collected for measuring and implementing critical security controls are extracted from the critical security control entity relationship tables and tabulated in Appendix A.

2.1 Machine Learning with Apache Spark

This section involves performing feature extraction and applying machine learning models to measure Critical Security Controls and displaying the results in interactive ranking dashboard/reports/alerts.

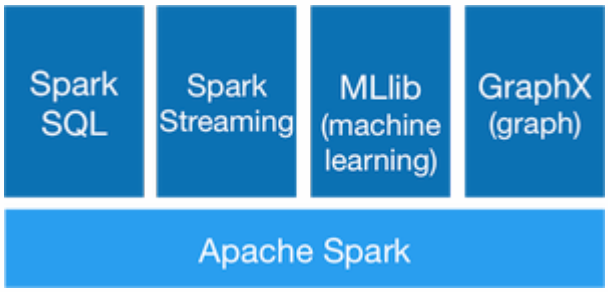

Figure 5 - Apache Spark Components

The key components that are very effective for machine learning and advanced analytics in Apache Spark are GraphX, MlLib, and Spark SQL. GraphX enables to implement advanced graph analytics to identify relationships in the data. MLLib provides the ability to seamlessly run complex machine learning algorithms on security data sets. Spark SQL provides extensive capabilities to transform data formats and can be used for advanced analytics and queries. The Apache Spark platform can be used to implement the machine learning models using MLLib libraries.

Machine learning

Machine learning has two major types: supervised and unsupervised learning. In Supervised Learning, the machine learning algorithm will be provided with data and labels (classification). The resultant model will try to predict the label (classification) given a set of features (AstroML, 2015). Some Classification Algorithms commonly used are Neural Networks, Random Forests, Support Vector Machines (SVM), Decision Trees, Logistic Regression, and Naive Bayes.

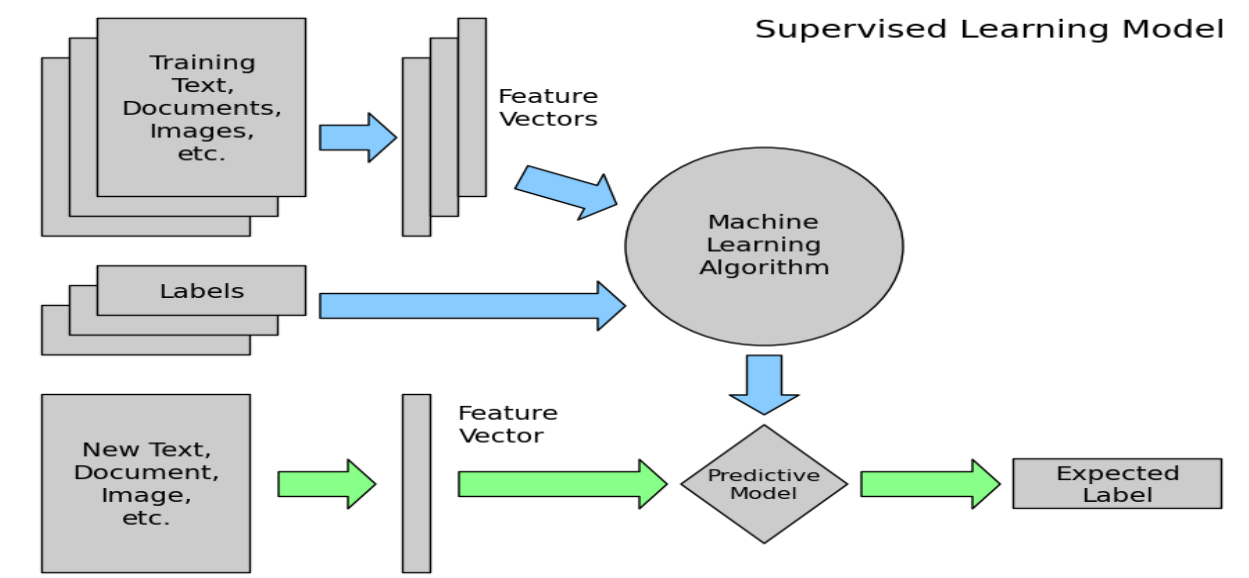

Figure 6- Supervised Learning Model (AstroML, 2015)

In Unsupervised Learning, the model tries to understand the data based on the features and the task is to identify patterns and anomalies from data. Unsupervised learning comprises tasks such as dimensionality reduction, clustering, and density estimation (AstroML, 2015).

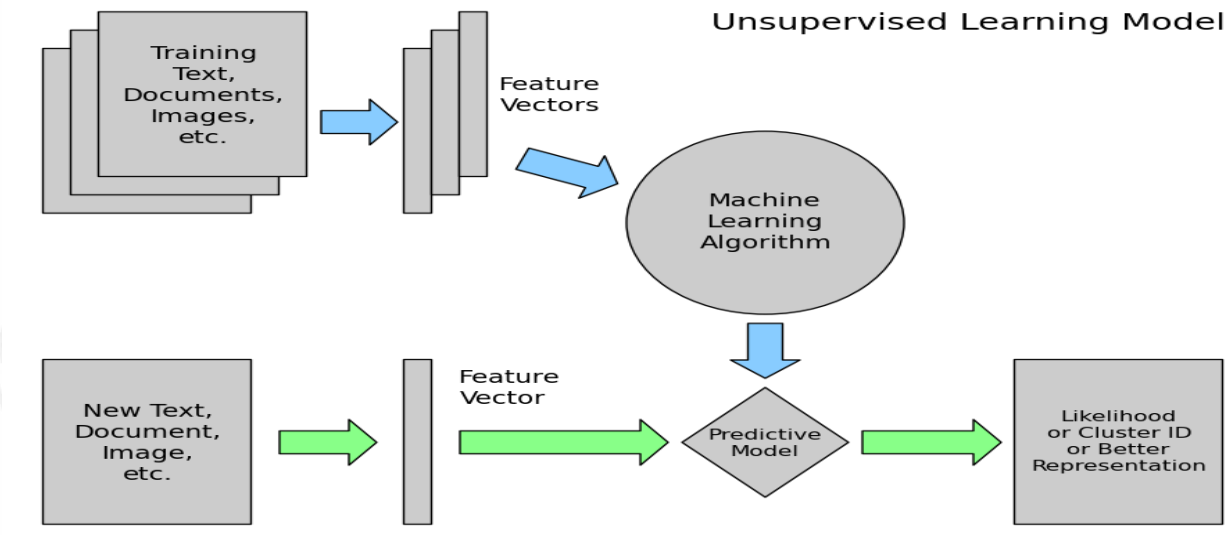

Figure 7- Unsupervised Learning Model

Some Common Unsupervised algorithms are K-Means clustering, Hierarchical clustering, and Hidden Markov models.

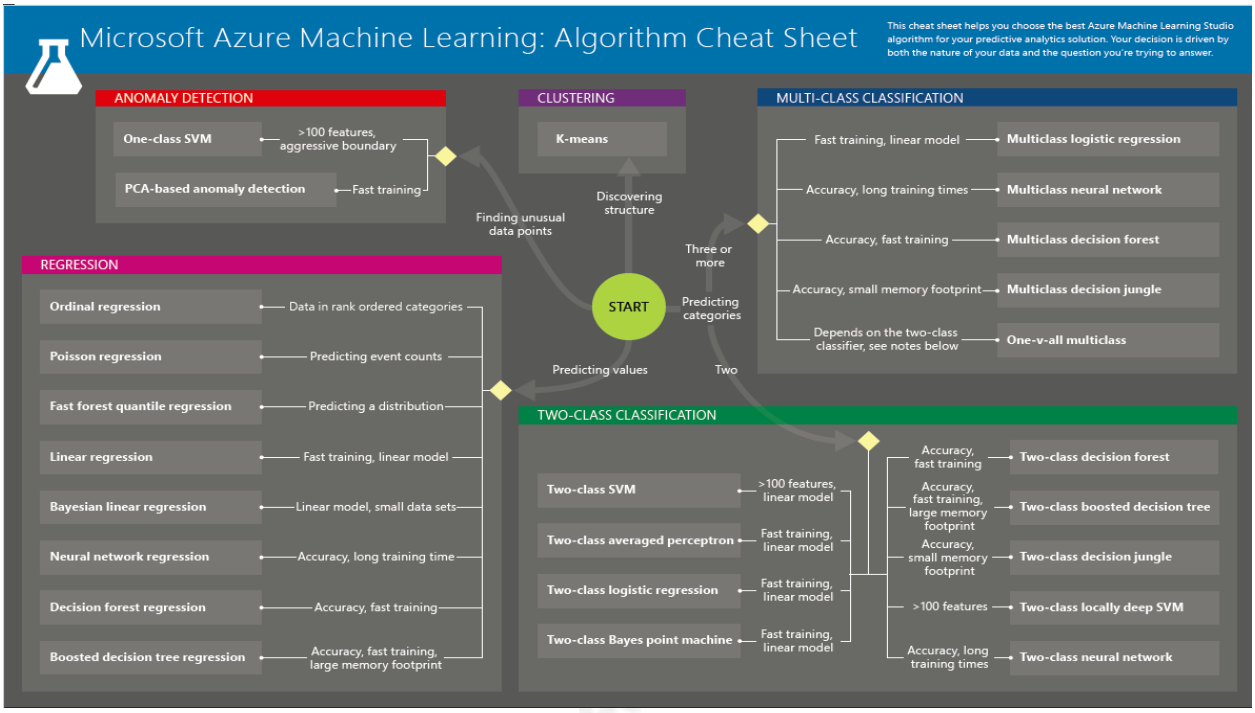

Figure 8 - Machine Learning Algorithm Cheat Sheet for Azure ML Studio

The solutions described below are dependent on having a big security analytics platform deployed and collecting all the required logs to perform the machine learning techniques.

The table in Appendix C captures the different machine learning techniques that can be applied to measure Critical Security Controls. The table of data sources is derived from the Entity Relationship diagrams in Appendix A.

Table 1 - Machine Learning Techniques to measure CSC

Use Case	Supervised – Classification/Regression Unsupervised – Clustering/Anomaly detection	Data sources	Applicable Critical Controls
Unauthorized Access analytics	Anomaly Detection	Identity and Access Management Windows AD Server access Privileged access	CSC 14, CSC 16

		HR Systems	
Threat intelligence	Anomaly Detection	Threat Intel Bro logs	CSC 8 , CSC 11 , CSC 12
Unauthorized Cloud access analytics	Anomaly Detection	Cloud Access logs Geo location Identity and Access management logs	CSC 12 , CSC 14 , CSC 16
Detect malicious network traffic – Detecting DGA	Classification	DNS logs Alexa Network	CSC 8 , CSC 11 , CSC 12
Detect malicious network traffic	Anomaly detection	Network - Nmap, Prads, Authentication logs	CSC 8 , CSC 11 , CSC 12
Highest Risk Asset Ranking	Ranking	All relevant logs	All CSC controls
Unauthorized Device	Classification/ anomaly detection	Network - Nmap, Prads, Authentication logs Asset database (ServiceNow)	CSC 1
Unauthorized software	Classification	Network Bro application detection Software Asset database	CSC 2

2.2 Machine Learning Prototypes

The prototype provides high-level examples of how some of the concepts discussed in machine learning can be applied. The dataset and tools used were mostly

open-source. A good source for some of the datasets is Security Repo [1]. The examples in this section are modified to match the test data. Security practitioners should modify all the examples according to the implementation and configuration of the solution used by the organization.

Follow the five basic steps to build an experiment to create, train, and score your model:

Create a model

- Step 1: Get data
- Step 2: Preprocess data
- Step 3: Define features

Train the model

- Step 4: Choose a learning algorithm and apply it

Score and test the model

- Step 5: Classify if the traffic is benign or malicious.

2.2.1 Azure ML Studio Prototype - Binary Classification: Network Intrusion Detection & Anomaly Detection

Azure ML Studio account can be created online[2]. The visual interface allows seamless implementation and evaluation of the machine learning models. Follow five basic steps discussed earlier to build an experiment in Machine Learning Studio to create, train, and score your model.

[1] http://www.secrepo.com/
[2] https://studio.azureml.net/

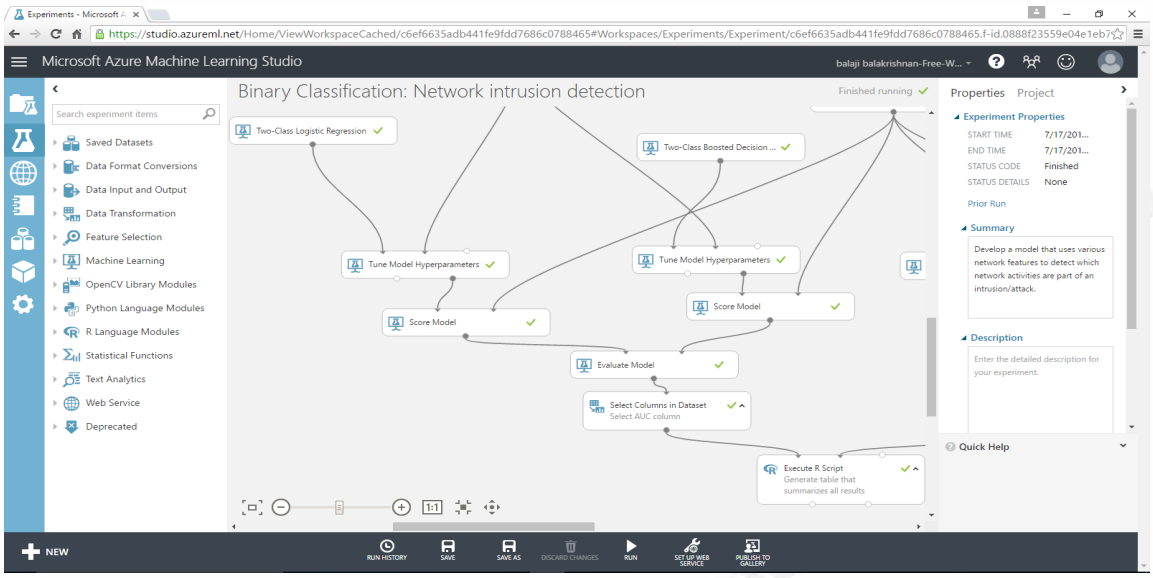

Figure 9 - Azure ML Studio - Classification

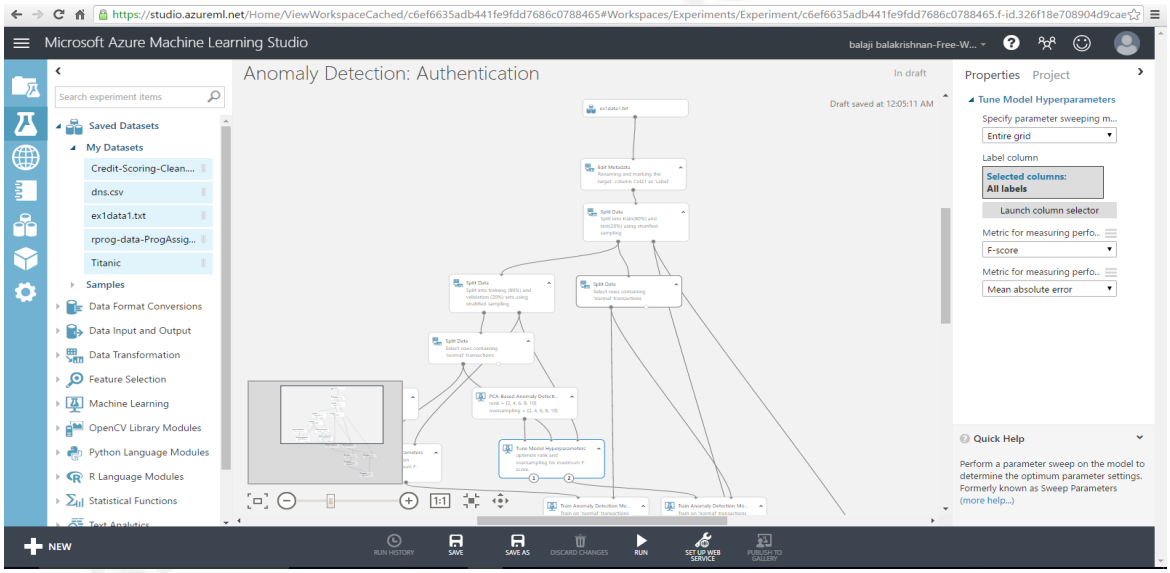

Figure 10 - Azure ML Studio - Anomaly Detection

As we start applying more algorithms the machine learning process gets complicated, Azure ML Studio visualizes the process and shows the steps in sequential order making it comparatively easy for experimenting and implementing the models. Azure ML Studio allows options to display the outcomes as web services. The output can also be fed back to the HBase data repository for further processing. Another example using the Azure ML Studio is anomaly detection. Anomaly detection machine learning models can be used to detect malicious traffic with outliers or malicious logins to the application. Appendix C goes through the different steps in Binary Classification: Network intrusion detection and Anomaly detection models.

2.2.2 Apache Spark Prototype – Random Forest & Classification Model – Classifying text

Databricks community edition provides users with access to an Apache Spark cluster. This environment can be used to import existing code examples and also create new Notebooks to experiment with different machine learning models. This allows the security team to focus on their data problems and models instead of solving the complexities around the distributed data infrastructure. Appendix C has the Random Forest & Classification model – Classifying text Databricks Apache Spark Notebook. It gives guidance on how Spark MLLib machine learning libraries can be used. The code can be edited to suit different data sets and use cases. This example on text classification can be used to create machine learning model for analyzing security sources.

2.2.3 Applying K-Means Clustering machine learning algorithm in Apache Spark

K-Means is a well-known clustering method, some common examples where K-Means clustering can be used are to cluster network traffic and authentication data.

```
# Load and parse the data

data = sc.textFile("kmeans_data.txt")

parsedData = data.map(lambda line: array([float(x) for x in line.split(' ')])).cache()

# Build the model (cluster the data)

clusters = KMeans.train(parsedData, 2, maxIterations = 10, runs = 1, initialization_mode = "kmeans||")

# Evaluate clustering by computing the sum of squared errors

def error(point):

        center = clusters.centers[clusters.predict(point)]

        return sqrt(sum([x**2 for x in (point - center)]))

cost = parsedData.map(lambda point: error(point)) .reduce(lambda x, y: x + y)

print("Sum of squared error = " + str(cost))

Dimension reduction + k-means

// compute principal components val points:

val mat = RowRDDMatrix(points)

val pc = mat.computePrincipalComponents(20)
```

```
// project points to a low-dimensional space
val projected = mat.multiply(pc).rows
// train a k-means model on the projected data
val model = KMeans.train(projected, 10)
```

Figure 11 K-Means Code Snippet (Xianguri, 2016)

These are just two prototypes which allow security practitioners to start using machine learning models. There are many resources available in the references section which might provide additional guidance on using the machine learning models to automate and measure critical security controls.

3 Security Ontology

Once we have all the data collected in Apache Metron (HBase), technologies like Jena, Marklogic, and/or Spark GraphX can be used for applying all the semantic models and machine learning models to help implement and measure the critical security controls. This section covers applying security ontology for extracting entities and identifying relationships between the entities.

One of the limitations in some SIEM implementations have been the normalization of data, converting all the different data sources into common entities like user, IP address, and performing analytics using the derived relationships. Semantic web/Graph databases overcome this limitation and allows the capability to ingest data from different data sources, extract entities and identify relationships between them. These relationships can be used to perform various queries to meet critical security control measurements.

The Semantic Web is a technology and a set of standards for sharing data using a model called the Resource Description Framework (RDF). RDF enables the representation of data as a set of linked statements, each of which consists of a subject, predicate, and object called a triple. RDF datasets, consisting of millions of triples, form a network of directed graph (DG) and are stored in systems called triple-stores. A query language standard, SPARQL, has also been developed to query RDF datasets (Mammo, 2014).

Ontologies are meant to represent information coming from all sorts of heterogeneous data sources. This makes ontologies ideal for dealing with all the

different structured, semi-structured, and unstructured data that comes in various formats and languages from across cyberspace (CSCSS, 2016).

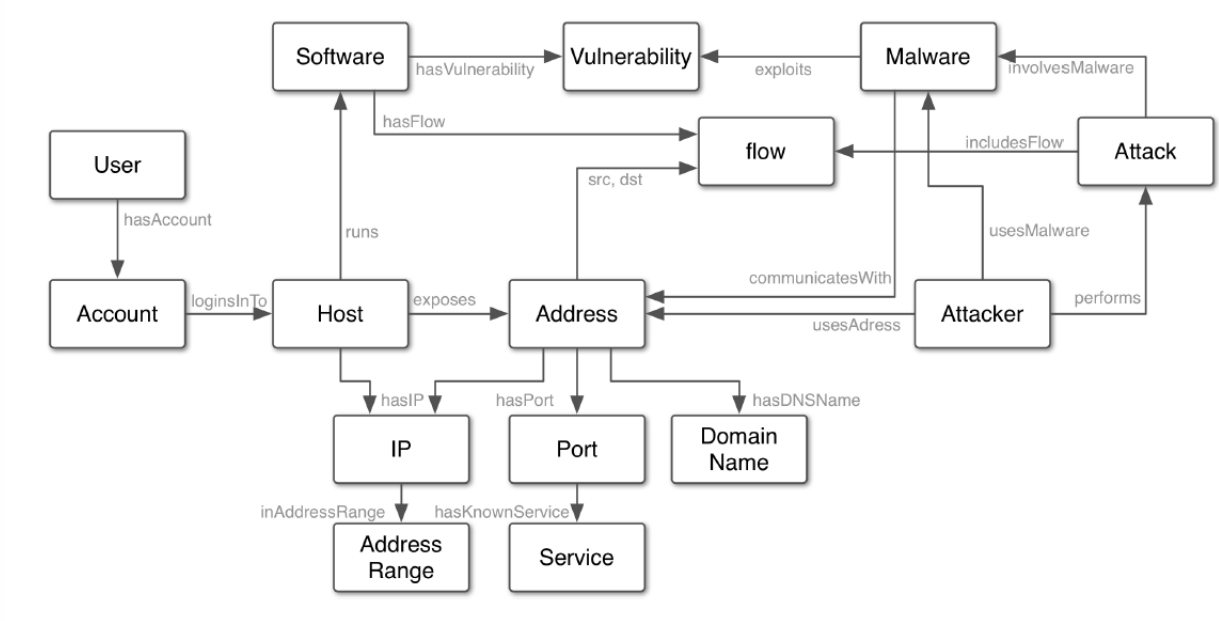

Figure 12- STUCCO Security Ontology

The first step is to create the RDFs in a database which supports RDFs. In some databases like Marklogic, once the RDFs are created, the RDFs will be applied to all the documents while ingesting them. This will identify relationships in various data sources and extract the entities and relationships. In some implementations, natural language processing should be used to extract the entities using machine learning algorithms. Once the relationships are extracted the database can be queried using SPARQL for understanding high-risk assets which are not compliant with critical security controls.

For example, consider the Critical Security Control 1 - Inventory of Authorized and Unauthorized Devices. The various sources that can be used as inputs for baseline and inventory of the assets are network monitoring tools, vulnerability assessment tools (passive and active) and an asset inventory database like Remedy, or Service Now. All the data from these sources should be converted to a single taxonomy/ontology so that the data can be normalized and queried for understanding various relationships between the datasets and authoritative baseline of inventory of assets can be obtained, and further intelligence can be derived.

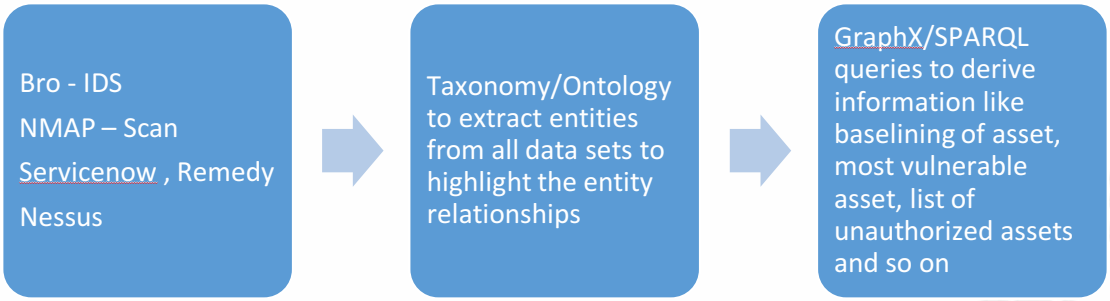

Figure 13- CSC 1 - Security Ontology Workflow

For critical security controls, below ontologies (RDFs) are required:

Table 2 – CSC SCAP Standards

Component	Version	Description
AI	1.1	Asset Identification (AI) is a specification for identifying assets
ARF	1.1	The Asset Reporting Format (ARF) is a specification describing a data model for asset reporting
CCE	5	The Common Configuration Enumeration ™ (CCE) is a nomenclature and dictionary of software security configurations
CCSS	1.0	The Common Configuration Scoring System (CCSS) is a specification for measuring the relative severity of system security configuration issues
CPE	2.3	The Common Platform Enumeration (CPE) is a specification measuring the relative severity of system security configuration issues
CVE	n/a	The Common Vulnerability Enumeration® (CVE) is a specification describing a nomenclature and dictionary of security-related software flaws
CVSS	2.0	The Common Vulnerability Scoring System is a language for representing system configuration information, assessing machine state, and reporting assessment results
OCIL	2.0	The Open Checklist Interactive Language (OCIL) is a language for representing checks that collect information from people or existing data stores made by other data collection efforts
OVAL	5.10.1	The Open Vulnerability and Assessment Language is a language for representing system configuration information, assessing machine state, and reporting assessment results
SCAP	1.2	SCAP is a specification for expressing and manipulating security data in standardized ways. SCAP uses several individual specifications in concert to automate continuous monitoring, vulnerability management, and security policy compliance evaluation reporting
TMSAD	1.0	The trust Model for Security Automation Data (TMSAD) describes a common trust model that can be applied to specifications within the security automation domain
XCCDF	1.2	Extensible Configuration Checklist Description Format (XCCDF) is a specification language for writing security checklists, benchmarks, and related kinds of documents
CMSS	n/a	Common Misuse Scoring System
CRE	n/a	Common Remediation Enumeration
CVRF	n/a	Common Vulnerabilities Reporting Framework
CWRAF	n/a	Common Weakness Risk Analysis Framework
CWSS	n/a	Common Weakness Scoring System
MMDEF	n/a	Malware Metadata Exchange Format
MAEC	n/a	Malware Attribute Enumeration and Classification
CybOX	n/a	Cyber Observable Expression

The reference section has additional information on the detailed functions of the different standards in the cyber ontologies. Appendix A provides a detailed table on the different security ontology applicable to critical security controls.

Table 3 - Security Ontology applicable to CSC

	Security Ontologies applicable to Critical Security Controls															
	VERIS	SCAP	CVE	CCE	CWE	OVAL	CAPEC	CVSS	XCCDF	OCIL	ARF	SWID	MMDEF	CybOX	STIX	MAEC
CSC 1	X										X					
CSC 2	X										X	X				
CSC 3	X	X	X	X	X	X	X	X	X	X						
CSC 4		X	X	X	X	X	X	X	X	X	X					
CSC 5	X			X					X							
CSC 6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CSC 7	X												X	X	X	X
CSC 8	X												X	X	X	X
CSC 9	X	X	X	X	X	X	X	X	X	X	X					
CSC 10		X	X	X	X	X	X	X	X	X						
CSC 11	X	X	X	X	X	X	X	X	X	X	X					
CSC 12	X												X	X	X	X
CSC 13	X	X		X	X	X			X	X	X					
CSC 14	X	X		X	X	X			X	X	X					
CSC 15	X	X		X	X	X			X	X	X					
CSC 16	X	X		X	X	X			X	X	X					
CSC 17	X								X							
CSC 18	X	X	X	X	X	X	X	X	X	X	X					
CSC 19	X	X					X	X		X	X	X	X	X	X	X
CSC 20	X	X	X	X	X	X	X	X	X	X	X					

Some examples of the SPARQL Queries that can be developed to measure Critical Security Controls are highlighted below:

- Highest Risk Asset Ranking
- Vulnerability Metrics – Most impactful vulnerability
- Configuration management metrics – Assets with most configuration issues
- Privileged User Monitoring

Many machine learning algorithms can also be applied once the relationships are established. The most useful algorithms are the ranking algorithms which can be used

to identify the highest ranking asset from the context of vulnerability or configuration issues or incidents or applications.

The key advantage of this methodology is abstracting information from various sources and putting in known entities which can then be used for executing many different queries and algorithms to find patterns or anomalies easily. This process of finding pattern or anomalies is not possible without establishing these relationships with common taxonomy.

3.1 Marklogic Prototype

Marklogic software can be downloaded from the Marklogic website. There is a free developer version which supports up to 1 TB. After installation of the Marklogic server the ml-cyber source code in Github demonstrate how multiple cyber ontologies can be imported into MarkLogic (ml-cyber, 2016). These ontologies can be used to represent various cybersecurity entities. SQARQL queries can be used to understand the relationships between the data.

3.2 Bloodhound Prototype

BloodHound uses Powershell scripts to collect active directory data and uses Neo4J graph database to identify relationships within the data. There are many pre-defined analytics queries and web interface for querying the graph database. This tool is very powerful to understand the effectiveness of graph database and will provide insights on how to identify hidden and unintentional relationships.

4 Implementing the Critical Security Control Metrics, measures, and thresholds

Once the platform is built and the entities and relationships are extracted, SPARQL queries, machine learning algorithms, and simple rules can be used to develop CSC dashboards. All the measures, metrics and thresholds in the CIS Critical Security Controls (Version 6) can be developed using the solution. As an example, the below measures, metrics and thresholds for CSC 1 can be calculated, and dashboard can be created.

CIS Critical Security Controls (Version 6): Measures, Metrics, and Thresholds

ID	Measure	METRICS		
		Lower Risk Threshold	Moderate Risk Threshold	Higher Risk Threshold
1.1	How many unauthorized devices are presently on the organization's network (by business unit)?	Less than 1%	1%-4%	5%-10%
1.2	How long, on average, does it take to remove unauthorized devices from the organization's network (by business unit)?	60 Minutes	1,440 Minutes (1 Day)	10,080 Minutes (1 Week)
1.3	What is the percentage of systems on the organization's network that are not utilizing Network Level Authentication (NLA) to authenticate to the organization's network (by business unit)?	Less than 1%	1%-4%	5%-10%
1.4	How many hardware devices have been recently blocked from connecting to the network by the organization's Network Level Authentication (NLA) system (by business unit)?			
1.5	How long does it take to detect new devices added to the organization's network (time in minutes - by business unit)?	60 Minutes	1,440 Minutes (1 Day)	10,080 Minutes (1 Week)
1.6	How long does it take to isolate/remove unauthorized devices from the organization's network (time in minutes - by business unit)?	60 Minutes	1,440 Minutes (1 Day)	10,080 Minutes (1 Week)

Figure 14 - CSC 1 Measures and Metrics

Some possible applications of the solution to meet CSC controls are highlighted:

CSC 1: Inventory of Authorized and Unauthorized Devices

CSC 2: Inventory of Authorized and Unauthorized Software

Potential applications of Machine Learning and Graph/Semantic model(RDF)

Classification algorithms can be deployed with features from multiple data sources for higher fidelity to classify authorized or unauthorized devices. One data source may be the network data which continuously captures the assets in the network. Another data source is the asset management database. Additional solutions using tools like Nmap can also be implemented to capture the active hosts in the network. In some cases, if the environment is well-defined simple, alerting logic can be deployed. Applying the RDFs on all data to extract asset related entities helps to correlate the

highest risk assets. Some examples metrics are Alerts for Unauthorized assets and Dashboard on highest risk asset.

CSC 3: Secure Configurations for Hardware and Software on Mobile Device Laptops, Workstations, and Servers

CSC 4: Continuous Vulnerability Assessment and Remediation

CSC 9: Limitation and Control of Network Ports, Protocols, and Services

CSC 11: Secure Configurations for Network Devices such as Firewall Routers, and Switches

CSC 15: Wireless Access Control

Potential applications of Machine Learning and Graph/Semantic model(RDF)

There are many solutions which provide dashboards on metrics on CSC 3 and CSC 4 like Nessus, OpenSCAP, and other tools. By combining the reports with other data sources, a holistic view of the environment is achieved. For example, identifying the relationships between the network intrusion detection logs to the vulnerability management data and penetration testing data would provide valuable insight to the security team. As another example, if a scan was detected, identifying that the asset does not have the vulnerability or is correctly configured, is a great tool to the security team. Another key advantage is that ranking algorithms can be deployed to provide a CSC controls-centric view, to determine which is the most effective control implemented in the environment. Anomaly detection algorithms can be applied to detect misconfigured systems.

CSC 5: Controlled Use of Administrative Privileges

CSC 14: Controlled Access Based on the Need to Know

CSC 16: Account Monitoring and Control

Potential applications of Machine Learning and Graph/Semantic model(RDF)

Anomaly detection algorithms to detect suspicious access events are very effective when features from HR data, identify and access management data and other sources like active directory events.

The same methods can also be applied to identify suspicious administrative access. The clustering (peer grouping) of access events will be an effective way to identify anomalies. These methods will also be useful with cloud access and cloud administrative privileges considering most of these are API-based. And since the

volume is typically very high, it is challenging to create rule-based access monitoring. Due to the volume and the level of access, machine learning algorithms perform well.

Different user populations have various levels of abnormality present. Peer groups can be calculated by assessing user's average activity for each estimated value (bytes out of network, removable media count, sensitive document access, etc.) in addition to their assigned roles. This additional comparison allows users own past actions to self-define who they operate most like and then to perform comparisons against like-acting peers.

CSC 6: Maintenance, Monitoring, and Analysis of Audit Logs

Applying the RDFs on all data to extract asset related entities helps to correlate the highest risk assets.

CSC 7: Email and Web Browser Protections

CSC 8: Malware Defenses

CSC 12: Boundary Defense

CSC 13: Data Protection

Anomaly Detection and Classification algorithms can be used for detecting malicious traffic from endpoints and network. Classification algorithms can be deployed with features from multiple data sources for higher fidelity to classify benign or malicious. One data source may be the network data which continuously captures the traffic in the network. Another data source might be Alexa domain reputation and authentication/user logs.

Different network segments have various levels of abnormality present. Peer groups can be calculated by assessing network segment's average activity for each estimated value (bytes out of network). This additional comparison allows network segments own past actions to self-define who they operate most like and then to perform comparisons against like-acting peers.

Threat intelligence and related new articles can be ingested into the solution, and all the entities can be extracted and relationships identified providing a holistic view of the threat.

For example, recently IBM Watson was tasked with reviewing news articles and learning the relationships and extracting entities. In just one page, relevant security information was captured and integrated.

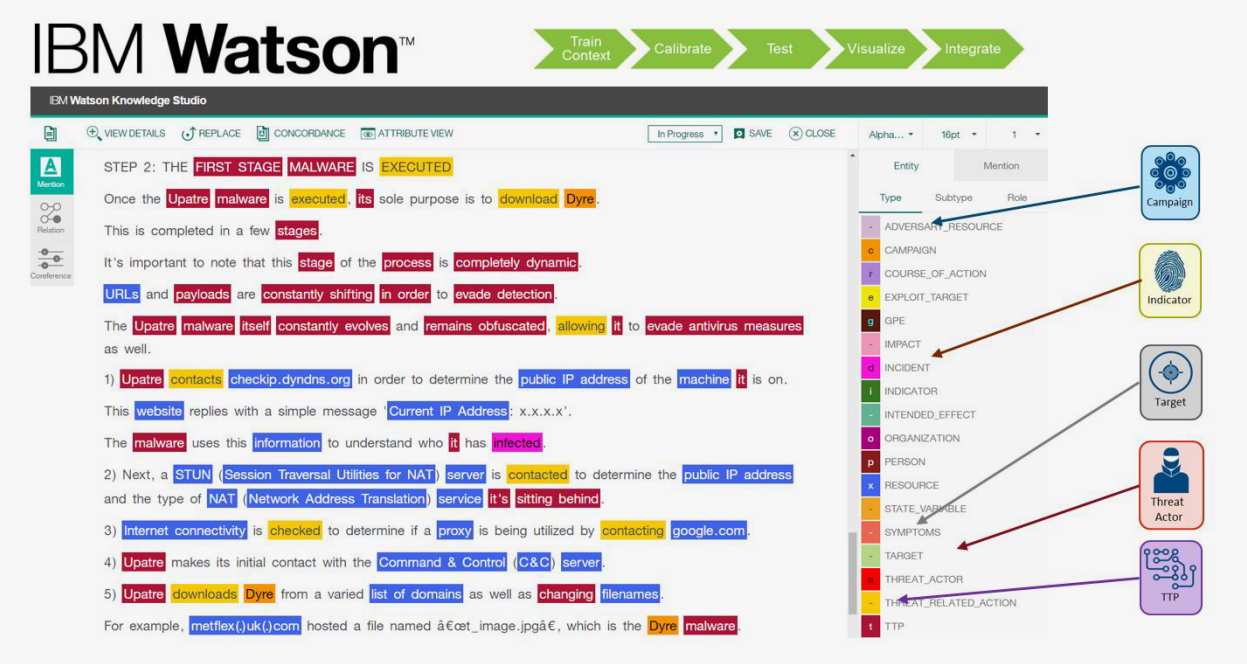

Figure 15- Watson for Cyber Security (Wired, 2016)

This CSC control is good example to understand the benefits of extracting asset related entities to help correlate the highest risk assets.

CSC 17: Security Skills Assessment and Appropriate Training to Fill Gaps

CSC 18: Application Software Security

CSC 19: Incident Response and Management

CSC 20: Penetration Tests and Red Team Exercises

Most of the data on security awareness are unstructured and is available in some form of reports. Extracting entities and relationships from those reports enable the organization to understand which section of users are assets are most vulnerable and tailor the awareness programs appropriately.

Similarly, most of the data from application security scanning, incidents, and penetration tests can be integrated with other data sets to provide a holistic view.

5 Conclusion

Security teams collect a lot of valuable information from different tools and solutions. In most implementations, the various tools operate independently and do not integrate to paint a complete risk picture. Often, security analysts have to manually combine the data, which is not scalable in medium and large environments. Aggregating all the data in one data repository provides the capability to the security team to analyze and derive data-driven security decisions.

The solution described collects and stores all types of data from all relevant security data sources. It then creates linked-data knowledge extractions and performs advanced analytics/Machine learning techniques to measure the effectiveness of Critical Security Controls.

Some other use cases which might benefit from this solution are Risk Management, Security Automation/Orchestration, User/Network Behavior Analytics, Fraud Detection, Threat Hunting, Threat Intelligence aggregation from various sources, Incident Response/Forensic Analysis, Cloud Access Security Monitoring.

Recently at the DefCon 2016 conference, Cyber Grand Challenge (CGC) was conducted and during the competition, each team developed code to automatically identified software flaws and scanned a network to identify affected hosts.

Google and Microsoft have been very active in this cognitive computing space and are releasing many open source tools like Tensorflow, cloud natural language processing APIs to solve problems and automate many tasks using AI enabled cognitive computing. With the advances in big data, artificial intelligence and machine learning technologies the capability is within reach in a few years. Sometime in the future, we may have autonomous systems, such as security operations centers, patch management, critical security controls enforcement - similar to autonomous self-driving cars. Hopefully, these machine learning and cognitive AI technologies along with skilled information security resources give us an edge over adversaries in future.

References

Welcome to the CIS Controls. (n.d.). Retrieved July 28, 2016, from https://www.cisecurity.org/critical-controls.cfm

Critical Security Controls - AuditScripts.com. (n.d.). Retrieved July 28, 2016, from http://www.auditscripts.com/projects/critical-security-controls/

CIS Critical Security Controls: Solution Directory. (n.d.). Retrieved July 28, 2016, from https://www.sans.org/critical-security-controls/vendor-solutions

A Small Business No Budget Implementation of the SANS 20 Security Controls. (n.d.). Retrieved July 28, 2016, from https://www.sans.org/reading-room/whitepapers/hsoffice/small-business-budgetimplementation- 20-security-controls-33744

Cross Industry Standard Process for Data Mining. (n.d.). Retrieved July 28, 2016, from https://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining

Metron Architecture. (n.d.). Retrieved July 28, 2016, from https://cwiki.apache.org/confluence/display/METRON/Metron Architecture

Machine Learning 101: General Concepts. (n.d.). Retrieved July 28, 2016, from http://www.astroml.org/sklearn_tutorial/general_concepts.html

Daedafusion/cyber-ontology. (n.d.). Retrieved July 28, 2016, from https://github.com/daedafusion/cyber-ontology

Invincealabs/icas-ontology. (n.d.). Retrieved July 28, 2016, from https://github.com/invincealabs/icas-ontology

Stucco-archive/ontology. (n.d.). Retrieved July 28, 2016, from https://github.com/stucco-archive/ontology

Machine learning algorithm cheat sheet for Microsoft Azure Machine Learning Studio. (n.d.). Retrieved July 28, 2016, from https://azure.microsoft.com/en-us/documentation/articles/machine-learning-algorithm-cheat-sheet/

Inside MARKLOGIC SERVER. (n.d.). Retrieved July 28, 2016, from http://cdn.marklogic.com/wp-content/uploads/resources/Inside-MarkLogic-Server.pdf

MarkLogic Community. (n.d.). Retrieved July 28, 2016, from https://docs.marklogic.com/guide/semantics/intro

IBM's Watson Has a New Project: Fighting Cybercrime. (n.d.). Retrieved July 30, 2016, from https://www.wired.com/2016/05/ibm-watson-cybercrime/

Semantic Interpretation of Structured Log Files. (n.d.). Retrieved July 28, 2016, from http://ebiquity.umbc.edu/_file_directory_/papers/812.pdf

Extracting Cybersecurity Related Linked Data from Text. (n.d.). Retrieved July 28, 2016, from http://ebiquity.umbc.edu/_file_directory_/papers/682.pdf

Developing Scientific Foundations for the Operational Cybersecurity Ecosystem. (n.d.). Retrieved July 28, 2016, from http://cscss.org/wp-content/uploads/2015/08/CSCSS-Science-of-Security-Developing-Scientific-Foundations-for-the-Operational-Cybersecurity-Ecosystem.pdf

Binary Classification: Network intrusion detection. (n.d.). Retrieved July 28, 2016, from https://gallery.cortanaintelligence.com/Experiment/Binary-Classification-Network-intrusion-detection-2?share=1

Stucco-Data Cyber security data sources. (n.d.). Retrieved July 28, 2016, from http://stucco.github.io/data/

MLlib: Scalable Machine Learning on Spark. (n.d.). Retrieved July 28, 2016, from http://stanford.edu/~rezab/sparkworkshop/slides/xiangrui.pdf

SCAP Overview. (n.d.). Retrieved July 30, 2016, from https://scap.nist.gov/events/2010/itsac/presentations/day1/SCAP_101-SCAP_Overview.pdf

Understanding SCAP NIST guidance and using SCAP tools to automate security. (n.d.). Retrieved July 30, 2016, from http://searchsecurity.techtarget.com/tip/Understanding-SCAP-NIST-guidance-and-using-SCAP-tools-to-automate-security

Balakrishnan, B. (n.d.). Insider Threat Mitigation Guidance. Retrieved July 30, 2016, from https://www.sans.org/reading-room/whitepapers/monitoring/insider-threat-mitigation-guidance-36307

Recordedfuture. (n.d.). Anticipating Cyber Vulnerability Exploits Using Machine Learning. Retrieved July 28, 2016, from https://go.recordedfuture.com/hubfs/reports/anticipating-cyber-exploits.pdf

Advanced Technologies/Tactics Techniques, Procedures: Closing the Attack Window, and Thresholds for Reporting and Containment. (n.d.). Retrieved July 30, 2016, from http://www.cisco.com/c/dam/en_us/about/security/intelligence/JNS_TTPs.pdf

KDnuggets. (n.d.). Retrieved July 30, 2016, from http://www.kdnuggets.com/2015/06/introduction-big-data-apache-spark.html

Mammo, M. (2014, November). Distributed SPARQL over Big RDF Data. Retrieved July 30, 2016, from https://repository.asu.edu/attachments/143388/content/Mammo_asu_0010N_14524.pdf

Cross Industry Standard Process for Data Mining. (n.d.). Retrieved July 30, 2016, from http://www.digplanet.com/wiki/Cross_Industry_Standard_Process_for_Data_Mining

Adaptivethreat/BloodHound. (n.d.). Retrieved August 06, 2016, from https://github.com/adaptivethreat/BloodHound

Sastafford/ml-cyber. (n.d.). Retrieved August 06, 2016, from https://github.com/sastafford/ml-cyber

Aktaion. (n.d.). Retrieved August 06, 2016, from https://github.com/jzadeh/Aktaion/blob/master/documentation/BsidesLVPresso2016.pdf

CylanceSPEAR/NMAP-Cluster. (n.d.). Retrieved August 06, 2016, from https://github.com/CylanceSPEAR/NMAP-Cluster/tree/master/clusteringnmap

Blackhat. (n.d.). Retrieved August 06, 2016, from https://www.blackhat.com/docs/us-16/materials/us-16-Wolff-Applied-Machine-Learning-For-Data-Exfil-And-Other-Fun-Topics.pdf

MITRE. (n.d.). Retrieved July 30, 2016, from https://www.itu.int/en/ITU-T/studygroups/com17/Documents/tutorials/2011/SCE-2011-8-26.pdf

ENISA Information Sharing Standard. (n.d.). Retrieved July 30, 2016, from www.enisa.europa.eu/publications/standards-and-tools-for-exchange-and-processing-of-actionable-information/at_download/

Resources. (n.d.). Retrieved July 30, 2016, from https://databricks.com/resources/type/example-notebooks

Appendix A: Security Ontology applicable to Critical Security Controls

Table 4- Security Ontology - CSC

Critical Control	Data Source	Data type	Semantic model(RDF)	Data Sources - Example
CSC 1: Inventory of Authorized and Unauthorized Devices	Asset Inventory Database	Structured	ARF, VERIS	Servicenow, Remedy
CSC 1: Inventory of Authorized and Unauthorized Devices	Passive/Active network scanner	Structured	ARF, VERIS	Bro, PRADS, nmap
CSC 2: Inventory of Authorized and Unauthorized Software	Asset Inventory Database	Structured	ARF, SWID Tags	Servicenow, Remedy
CSC 2: Inventory of Authorized and Unauthorized Software	Software Asset Management	Structured	ARF, SWID Tags	Nessus
CSC 3: Secure Configurations for Hardware and Software on Mobile Device Laptops, Workstations, and Servers	SCAP Configuration Scanner	Structured	SCAP, CVE, CCE, CWE, OVAL, CPE, CVSS, XCCDF	Nessus, OpenSCAP
CSC 3: Secure Configurations for Hardware and Software on Mobile Device Laptops, Workstations, and Servers	Configuration Enforcement System	Structured	SCAP, CVE, CCE, CWE, OVAL, CPE, CVSS, XCCDF	Nessus, OpenSCAP
CSC 4: Continuous Vulnerability Assessment and Remediation	SCAP Vulnerability Scanner	Structured	SCAP, CVE, CCE, CWE, OVAL	Nessus, OpenSCAP

CSC 4: Continuous Vulnerability Assessment and Remediation	Patch Management	Structured	SCAP, CVE, CCE, CWE, OVAL	Nessus, OpenSCAP
CSC 5: Controlled Use of Administrative Privileges	Authentication System	Structured	SCAP, CVE, CCE, CWE, OVAL	Active Directory
CSC 5: Controlled Use of Administrative Privileges	Identity and Access Management	Structured	VERIS	Active Directory
CSC 6: Maintenance, Monitoring, and Analysis of Audit Logs	Network Time Protocol(NTP) system	Structured	CVE, VERIS	Apache Metron
CSC 6: Maintenance, Monitoring, and Analysis of Audit Logs	Computing Systems	All data types	All applicable models	Apache Metron
CSC 7: Email and Web Browser Protections	Configuration Enforcement System	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	Fireeye, Palo Alto
CSC 7: Email and Web Browser Protections	URL / Email Filtering Proxy System	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	Fireeye, Palo Alto
CSC 8: Malware Defenses	Network Malware Detection	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	Fireeye, Palo Alto
CSC 8: Malware Defenses	Endpoint Protection Software / EMET	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	Fireeye, Palo Alto
CSC 9: Limitation and Control of Network Ports, Protocols, and Services	SCAP Vulnerability Scanner	Structured	SCAP, CVE, CCE, CWE, OVAL	Nessus, nmap, Bro
CSC 9: Limitation and Control of Network Ports,	Host / Application Firewall Systems	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	OSSEC, Microsoft , McAfee

Protocols, and Services				
CSC 10: Data Recovery Capability	Data Backup System	All data types	SCAP, CVE, CCE, CWE, OVAL	
CSC 10: Data Recovery Capability	Off-site / Off-line Backups	All data types	SCAP, CVE, CCE, CWE, OVAL	
CSC 11: Secure Configurations for Network Devices such as Firewall Routers, and Switches	Dedicated Administration Systems	All data types	SCAP, CVE, CCE, CWE, OVAL	
CSC 11: Secure Configurations for Network Devices such as Firewall Routers, and Switches	Authentication System	All data types	SCAP, CVE, CCE, CWE, OVAL	
CSC 11: Secure Configurations for Network Devices such as Firewall Routers, and Switches	Network Device Management System	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 12: Boundary Defense	Network Monitoring (IDS & IPS)	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	
CSC 12: Boundary Defense	Authentication System	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC	
CSC 12: Boundary Defense	Network Device Management System	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC, VERIS	
CSC 12: Boundary Defense	Configuration Enforcement System	Structured	MMDEF, CybOX, STIX, MAEC,	

			CAPEC, VERIS	
CSC 12: Boundary Defense	Application Firewall/Proxy System	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC, VERIS	
CSC 13: Data Protection	Endpoint Protection / Removable Media Control	Structured	MMDEF, CybOX, STIX, MAEC, CAPEC, VERIS	
CSC 13: Data Protection	Network & Host Based DLP	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 13: Data Protection	Encryption Systems	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 14: Controlled Access Based on the Need to Know	Network Device Management System	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 14: Controlled Access Based on the Need to Know	Encryption Systems	Unstructured	SCAP, CVE, CCE, CWE, OVAL	Reports
CSC 14: Controlled Access Based on the Need to Know	Host Based DLP	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 15: Wireless Access Control	Configuration Enforcement System	Structured	SCAP, CVE, CCE, CWE, OVAL	Nessus
CSC 15: Wireless Access Control	SCAP Vulnerability Scanner	Structured	SCAP, CVE, CCE, CWE, OVAL	Nessus

CSC 15: Wireless Access Control	Wireless Intrusion Detection System (WIDS)	Structured	SCAP, CVE, CCE, CWE, OVAL	Cisco WIDS
CSC 15: Wireless Access Control	Network Device Management System	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 16: Account Monitoring and Control	Authentication System	Structured	ARF, VERIS	
CSC 16: Account Monitoring and Control	Identity & Access Management System	Structured	SCAP, CVE, CCE, CWE, OVAL	
CSC 16: Account Monitoring and Control	Configuration Enforcement System	Structured	SCAP, CVE, CCE, CWE, OVAL, VERIS	
CSC 17: Security Skills Assessment and Appropriate Training to Fill Gaps	User Assessments	Unstructured	VERIS	Security awareness reports
CSC 17: Security Skills Assessment and Appropriate Training to Fill Gaps	Education Plans / Training Programs	Unstructured	SCAP, CVE, CCE, CWE, OVAL, VERIS	Security awareness reports
CSC 18: Application Software Security	Patch Management System	Structured	SCAP, CVE, CCE, CWE, OVAL, VERIS	Nessus
CSC 18: Application Software Security	Code Review / Vulnerability Scanner	Unstructured	SCAP, CVE, CCE, CWE, OVAL, VERIS	Fortify , Cenzic reports
CSC 19: Incident Response and Management	Third Party Authorities	Unstructured	VERIS, CAPEC	Incident Management Tool
CSC 19: Incident Response and Management	Incident Management Documentation	Unstructured	VERIS, CAPEC	Incident Management Tool
CSC 20: Penetration Tests and Red	Penetration Testers	Unstructured	SCAP, CVE, CCE, CWE, OVAL,	Penetration testing report, Metasploit

Team Exercises			VERIS, CAPEC	
CSC 20: Penetration Tests and Red Team Exercises	Penetration Testing Systems	Unstructured	SCAP, CVE, CCE, CWE, OVAL, VERIS, CAPEC	Penetration testing report, Metasploit

Appendix C: Microsoft Azure ML

This section shows how to create machine learning models using Azure ML Studio. Azure ML Studio website provides options for creating Azure ML account. Once the Azure ML account is created, different experiments can be created, and existing examples can be modified to meet our requirements. The below experiment discussed is modified from existing examples in Azure ML Studio.

Binary Classification: Network intrusion detection

The goal of this model is to use various network features to detect which network activities are part of an intrusion/attack.

Follow the below steps to build an experiment to create, train, and score your model in Azure ML Studio:

Create a model

This screenshot below shows how the first three steps of getting data, preprocessing data and defining the features for the model to be implemented.

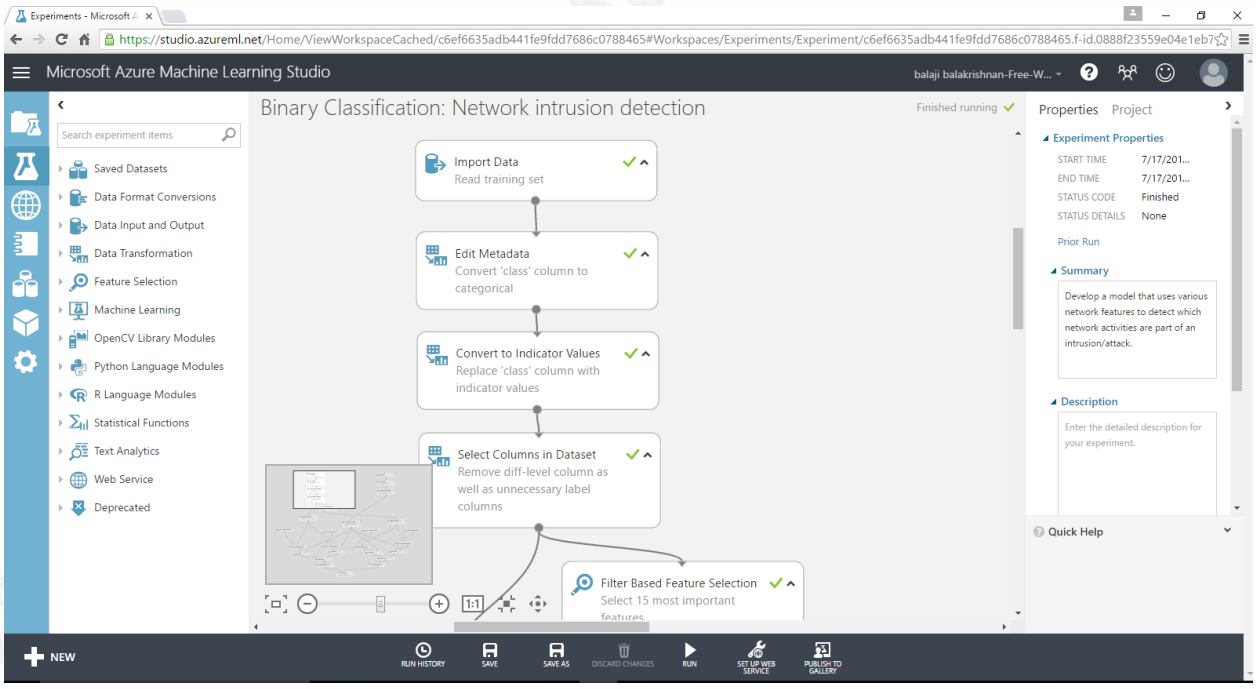

Figure 16 - Network Intrusion Detection – Preprocessing data and feature selection

Train the model

This screenshot below shows how to choose, apply and tune the parameters of a learning algorithm.

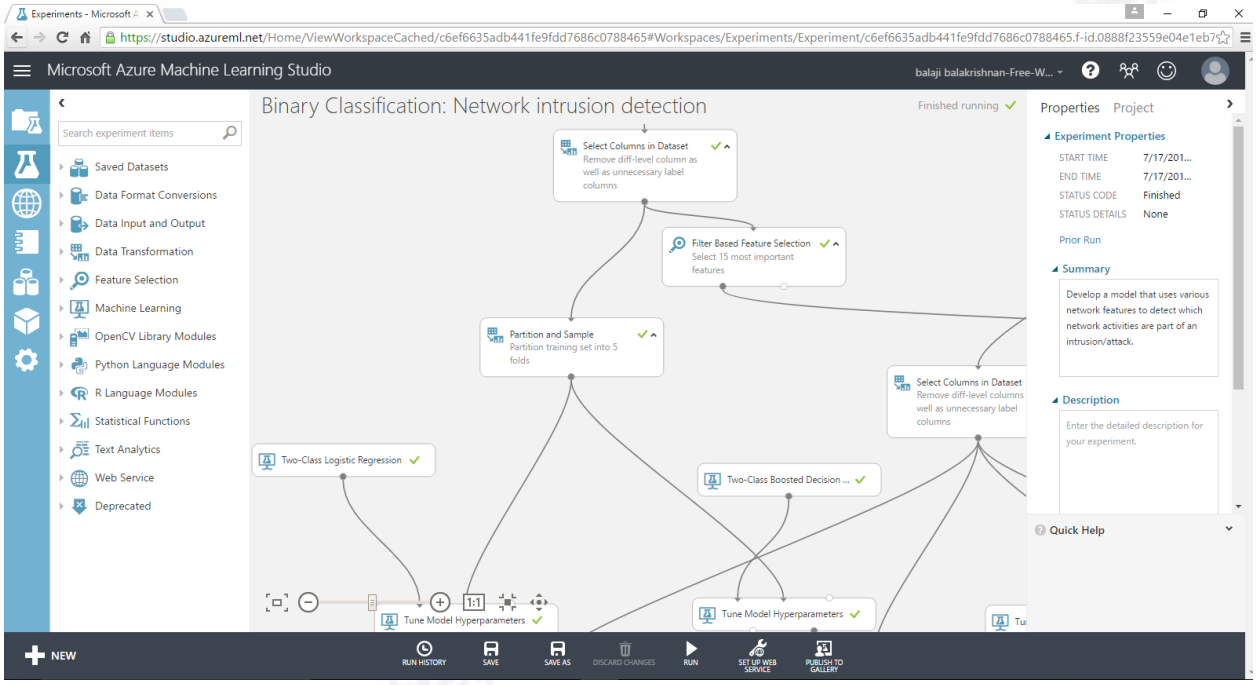

Figure 17 - Applying Two-class Logistic Regression algorithm

Score and test the model

This screenshot below shows how to evaluate and score the model for better performance.

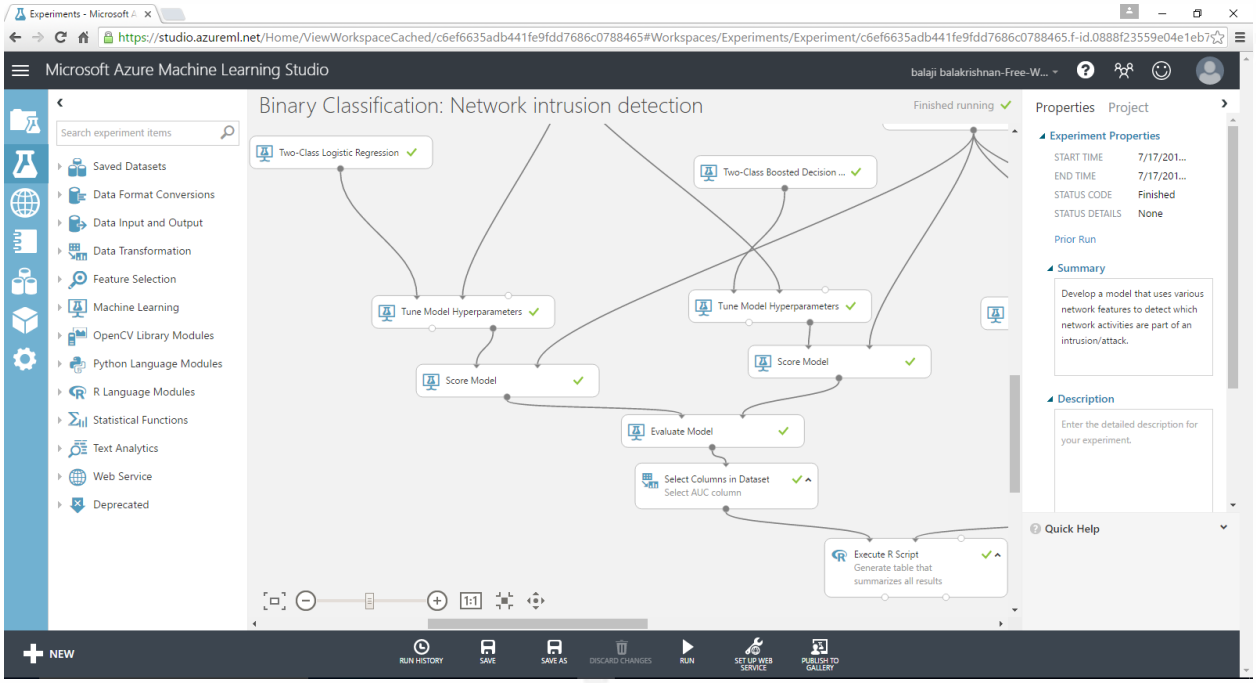

Figure 18 - Evaluate and score the model

The references section has additional experiments which can be updated to meet requirements specific to implementing critical security controls.

Appendix D: Apache Spark Prototype

Databricks community account edition provides an Apache Spark cluster. This environment can be used to import existing code examples and also create new Notebooks to experiment different machine learning models.

Databricks Community Spark Implementation of Random Forest

Databricks community account edition provides an Apache Spark cluster. This environment can be used to import existing code examples and also create new Notebooks to experiment different machine learning models. This step involves importing the random forest notebook, which can be modified to suit our needs like detecting DGA generated domains from DNS data set available in Security Repo [3].

The below screenshot shows how to import notebook into Databricks Spark cluster.

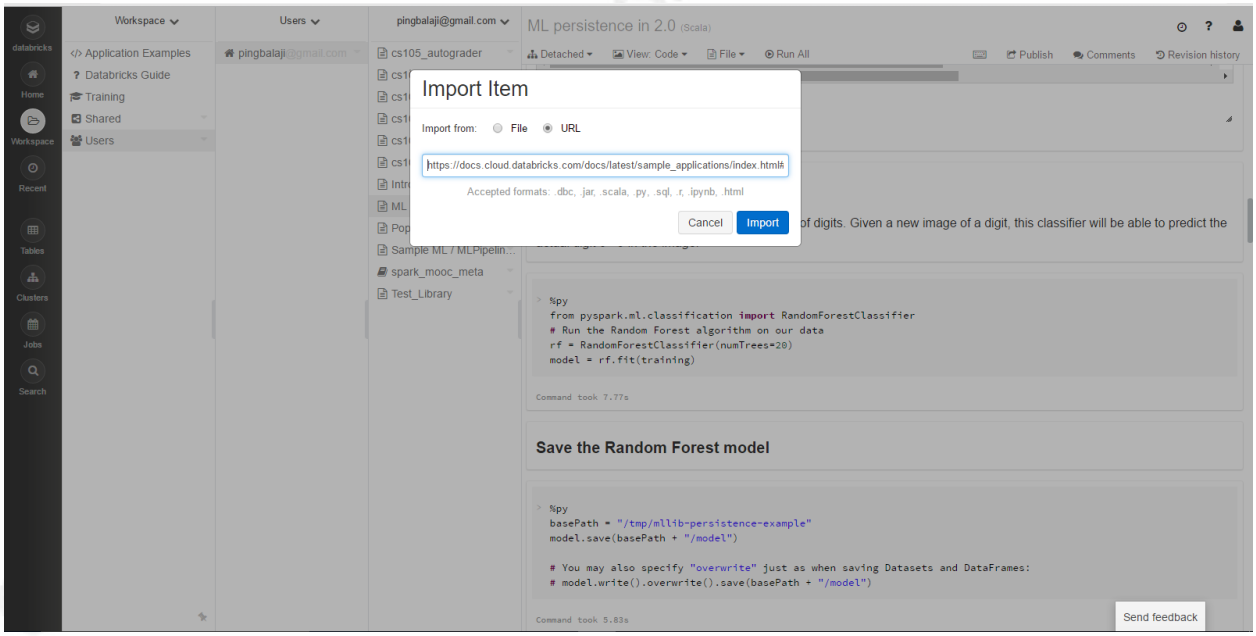

Figure 19 - Importing Notebook to Databricks

[3] http://www.secrepo.com/

The below screenshot shows how to train the model with training data set.

Figure 190 - Training Randomforest ML

The references section has additional experiment notebooks which can be updated to meet requirements specific to implementing critical security controls.