http://dx.doi.org/10.14236/ewic/ICS2018.2

SAMIIT: Spiral Attack Model in IIoT Mapping Security Alerts to Attack Life Cycle Phases

Amin Hassanzadeh, Robin Burkett
Accenture Technology Labs
Arlington, Virginia, USA
{*amin.hassanzadeh, robin.l.burkett*} *@accenture.com*

Sophisticated attacks such as NightDragon and Crashoverride have shown a multi-step multi-domain attack life cycle in Industrial Internet of Things (IIoT). Security analysts use cyber kill chain reference model to describe attack phases and adversary actions at each phase, link individual attacks into broader campaigns, and also identify courses of action. Although the model is widely studied and applied by IT security people, less is known and used in IIoT. In this research, we first review and evaluate several models proposed for attack life cycle in IT and IIoT. Next, a spiral attack model is proposed to map IIoT cyber intrusions to different attack phases and architectural levels of IIoT environments. Finally, we present a machine learning classification approach for mapping security alerts to IIoT attack phases and architectural layers. The results show the accuracy of the mapping mechanism and how it helps analysts in security operation centers to prioritize alerts and derive risk scores corresponding to each alert.

Industrial Internet of Things, Attack Life Cycle, Security Alerts, Machine Learning Classification

1. INTRODUCTION

Industrial Internet of Things (IIoT) networks have been targeted by extremely sophisticated attacks such as Crashoverride (1), Dragonfly (2)(3), the BlackEnergy campaign (4) and Stuxnet (5). These complex attacks are implemented by well-rehearsed and coordinated single-step attacks, are advanced (they use customized malware) organized and focused on specific targets, and they follow well-defined plans to reach their objectives. Similar to advanced persistent threats targeting enterprise networks, complex IIoT attacks follow a multi-phase life cycle that usually lasts few months; a low and slow process taken by the adversary before executing the last action against the final target. Public analyses on IIoT incidents (1)(2)(4)(5)(6)(7)(8) have also shown a similar pattern in which the adversary first compromises the corporate networks (IT world) and then acts as an insider to gain access to the final object in the Operational Technology (OT) domain.

It is observed (9)(10) that adversaries are often presented with ample time to execute the attack. However, despite their known patterns and time frame, yet detecting Advanced Persistent Threats (APT) remains an increasingly difficult challenge for security analysts. The reason for not being successful in preventing these attacks is multi-fold: (1) lack of technology (e.g., to detect zero-day anomalies); (2) lack of knowledge and resources (i.e., well-trained and experienced security analysts to review logs and alerts); and (3) lack of adequate incident triage and investigation process. Assuming that IIoT Security Operations Center (SOC) is utilizing the most recent and advanced technologies and very skilled people, an inadequate security event analysis process can simply fail in detecting APTs (11). More than 70% of security analysts participated in a survey conducted by Ponemon institute (12) wanted their Security Information and Event Management (SIEM) tool to generate fewer alerts that are more accurate, prioritized and meaningful, and also automate specific tasks that allow response teams to focus on priorities. In this research, we aim to improve processes that help security analysts focus on more accurate, prioritized and meaningful alerts during an attack life cycle. More specifically, a research question arises here, *how can we help security analysts prioritize security events, understand the current adversary's status, and concentrate on relevant logs and alerts?*

An important step in understanding adversary's status and mapping security events to it is to know attacks model and life cycle. In 2011, Lockheed Martin researchers proposed Cyber Kill Chain^{TM} (CKC) to assist the decision-making process for better understanding and reacting to adversary intrusions (13). Since then, inspired by cyber kill chain, several other attack reference models have been created (e.g., Mitre ATT&CK^{TM} (14), Mandiant (15)). In efforts to apply CKC in identifying evidence for most of attack phases and proposing correspondent defense mechanisms, researchers have perceived that it cannot identify all phases of complex attacks (e.g., recurring phases (16) or post-compromise attack phases (14)). Moreover, due to the different

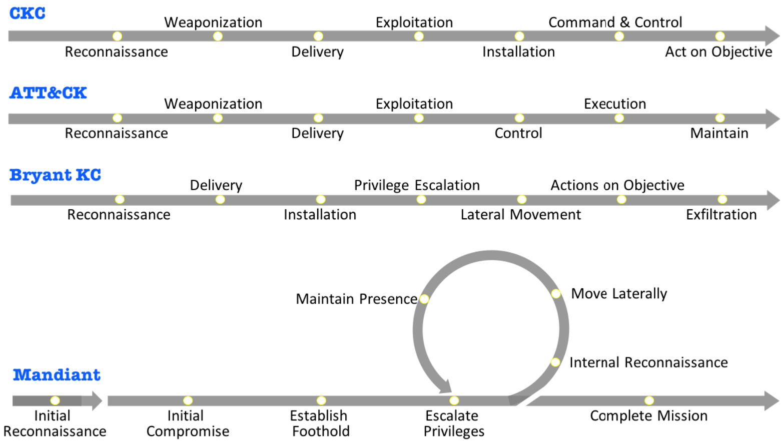

Figure 1: *Several Attack Models Proposed for Multi-Phase APTs: CKC (13), ATT&CK (14), Bryant KC (16), and Mandiant (15).*

architectural design in IIoT networks, and their IT/OT interconnections, CKC and other IT-specific attack models cannot be directly applied to them. Accordingly, some IIoT attack models such as SANS Industrial Control Systems (ICS) cyber kill chain (17) and ICS cyber defense triage process (18) have been proposed recently. However, these models are still based on CKC and of course not covering all of the phases of IIoT attacks. In fact, a very recent report by US-CERT on Russian government activities targeting critical infrastructure (19) shows that CKC model (not ICS CKC) is still used to analyze, discuss, and dissect ICS attacks.

In this research, we first survey state-of-the-art IT and IIoT attack models, review their pros and cons in identifying adversarial steps in details, and then propose a spiral IIoT attack model that takes all of those details (e.g., pre- and post-compromise steps, recurring phases, and IIoT architectural layers) into consideration. We believe applying a thorough and accurate IIoT attack model would help security analysts better understand attack life cycle, prioritize and focus on specific security controls and their logs/alerts at each phase, and take the courses of actions accordingly. (Note: from now on we refer to security alerts and system and security logs as “alert”.) In order to help analysts do so, we then propose a machine learning based approach in classifying alerts that automatically maps each of them to an attack phase. We will apply our method to two datasets collected from public and our test environment and discuss the challenges and results.

2. ATTACK LIFE CYCLE

In this section, we review state of the art in modeling complex multi-phase attack life cycle for both enterprise and IIoT networks. As shown throughout this section, more accurate and detailed attack models have been created in the last decade since the first cyber kill chain was proposed. Moreover, recent interests in IIoT security have inspired security researchers to define IIoT-specific attack models that consider architectural aspects of these networks.

2.1. Cyber Attack Reference Models

Researchers from Lockheed Martin expanded the concept of military kill chain, a systematic process an adversary uses to make desired effects on the target, for cyber intrusions. Cyber Kill Chain (13) models an intrusion life cycle for analysis and also driving defensive courses of action. The model is build based on the fact that APT utilizes intrusion after intrusion (multi-phase), and adjusts each operation based on the success/failure of the previous one, and one mitigation can break the entire chain and prevent the attack. As shown in Figure 1, CKC phases are defined as reconnaissance, weaponization, delivery, exploitation, installation, command and control (C2), and actions on objectives. The first three phases describe the process of target identification and selection, payload development and transmission to break into the trusted boundary. Next three phases focus on the process of establishing a presence in the target environment through vulnerability exploitation and remote control. Finally, the adversary will take actions towards the objectives (e.g., data exfiltration) or may just use the initial victim to move laterally inside the target network. CKC model can be used for designing defensive capabilities based on adversary actions (as proposed in its courses of action matrix (13)). However, several aspects of this model need further details and granularity in order to become an actionable intelligence; for instance, *pre-compromise vs. post-compromise* phases and *recurring* phases are some of those aspects considered in the models we have studied here.

FireEye’s M-Trends (21) reveals that during 2015, the average time to discover a security breach in an enterprise was 146 days. Also according to (13), a detection at any phase means that the preceding phases have already executed successfully. Therefore, unlike conventional incident response procedures that initiate after exploitation phase, defenders must move detection and analysis up the kill chain. However, pre-compromised activities are not in the enterprise's field of view (e.g., reconnaissance and weaponization) and mostly impossible to detect. Hence, defenders must expand (1) pre-compromise abilities: to monitor and deny attacker's activity outside the enterprise boundaries, and (2) post-compromise abilities: to detect and prevent them after the exploitation phase (i.e., the enterprise has been compromised but the adversary hasn't reached to the final target yet). ATT&CK (14) (as depicted in Figure 1) models the attack life cycle based on pre-compromise and post-compromise phases (i.e., before and after exploitation) and provides details on the tactics and techniques [1] that adversary uses at each phase. Adversary pre-compromise techniques and tactics are not always detectable or preventable (e.g., people weakness identification), moreover, APTs show a slow and multi-step post-compromise procedure between exploitation and end of the mission. Hence, understanding post-compromise ATT&CK tactics (e.g., privilege escalation, lateral movement, and exfiltration) and their corresponding techniques, and mapping events to each of them help defenders break the kill chain or at least increase adversary's costs to achieve the final objectives.

ATT&CK addresses the lack of details in post-compromise intrusion steps of CKC, however, the overall life cycle is still very similar to CKC. For example, security analysts can use techniques information to better understand adversary's tactical goal at each step, but they cannot predict what may happen next. This is because ATT&CK tactics are defined to be applicable from one endpoint to another as adversary laterally moves across the target network, while CKC considers the broader attack life cycle. Thus, the order of tactics varies from one attack to another. Unlike ATT&CK, most of state of the art on attack life cycle modeling propose an order of steps as the reference model where few steps might be skipped depending on the adversary's objectives. Bryant kill chain (16), shown in Figure 1, considers privilege escalation and lateral movement as two main steps in the kill chain. This model focuses on phases that leave trails inside the victim's network and can be observed within sensor data. In fact, (16) provides a detailed list of sensor data and attack indicators corresponding to each phase that can be used for forensics purposes (16).

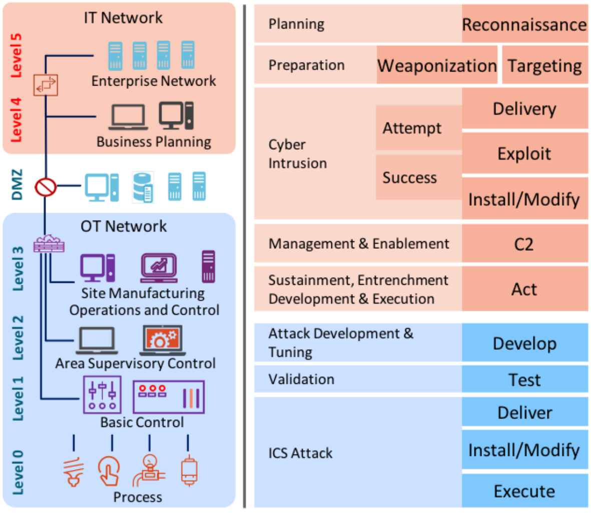

Figure 2: *IIoT Zoned Architecture (20) (left) and Cyber Kill Chain for Industrial Control Systems (17) (right).*

Once attackers compromise a target network, they establish a foothold to maintain remote access and move laterally inside the network until they reach to the final target. As shown in Figure 1, Mandiant attack model (15) defines a set of recurring phases after the initial compromise and before completing the mission by the attacker. These are basically steps towards finding next weak links/nodes and vulnerabilities, exploiting them and laterally move inside the victim's network, and finally executing the planned scenarios on the last target in the chain. Due to the architectural design of enterprise and IIoT networks, recurring phases are APT's most common patterns, and including them in the attack life cycle will help security analysts better understand complex attacks and predict and mitigate future intrusions. Microsoft Advanced Threat Analytics (ATA) (22) and FusionX expanded kill chain (23) are two other examples of attack life cycle models considering recurring phases in the post-compromise life of attacker.

2.2. IIoT Attack Reference Models

As IIoT offers a convergence of enterprise network and ICS infrastructure, a new reference architecture that includes end-to-end IT and OT networking components has been emerged. As depicted in Figure 2, a typical IIoT network consists of multiple zones, in which a *zone* is a set of network and operational entities grouped together to form a subclass of services and applications. In (10), we have reviewed the operational and security functions corresponding to each architectural level of IIoT reference architecture along with the IIoT attack patterns. In general, the target in industrial attacks can be an asset in either the IT, or in the DMZ, or in the OT zone, however the ultimate target in most of ICS incidents was shown to be a host/device in the control zone (levels 0-2). As observed in the

[1] "Tactics describe why an adversary performs an action, and techniques describe how they do it." (14)

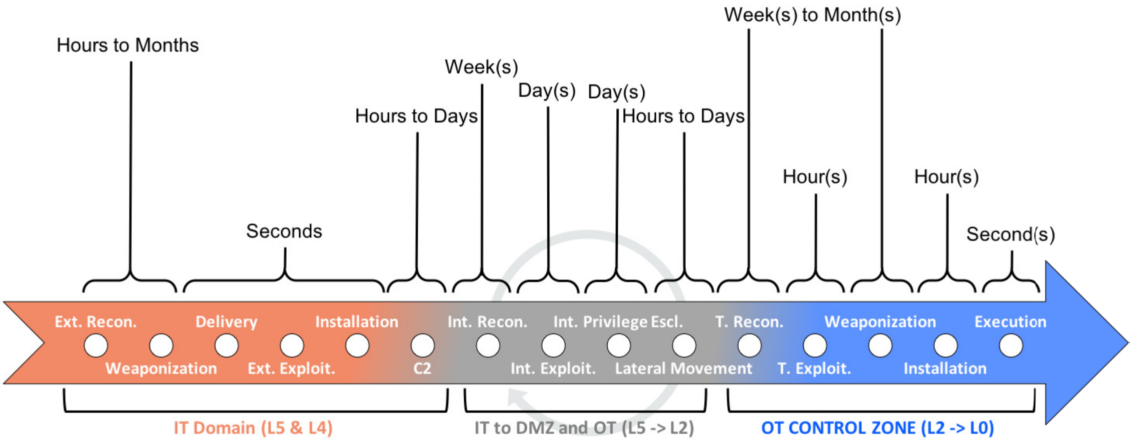

Figure 3: *A comprehensive, granular IIoT Attack Life Cycle.*

recent APT patterns in IIoT (10), the main attack vector focuses on gaining entry through IT systems and traversing to the OT infrastructure by launching multiple low and slow attacks. Therefore, APTs in IIoT are most likely *multi-domain*, multi-step attacks that require one or more recurring set of phases explained above. In other words, an actionable attack reference model for IIoT must take the architectural levels (or zones) into consideration.

ICS Kill Chain published by SANS (17) is developed based on CKC and zonal architecture of industrial networks. As shown in Figure 2, ICS kill chain is a two-stage attack model where the first stage (shown in pink) consists of exact same phases as CKC in IT; implying that an adversary needs to first compromise the IT network, gain knowledge about business and operational information, and finally target OT components. Upon gathering sufficient information about the cyber physical systems and processes in the OT, the attacker needs to develop and test a capability (e.g., a PLC configuration in (1)) for attacking it meaningfully (i.e., stage two shown in blue). For example, it may take few weeks/months for the attacker to learn what controller is used in the target control zone, acquire a similar component to play with, develop a malicious code for it, and finally install and execute it on the target. Note that all of these steps are taken outside the defender's field of view, and relying on detection of install/modify phase might be too late for preventing the attacker from damaging a critical asset.

ICS kill chain is the first IIoT attack model that includes the reference architecture of IIoT networks. However, it does not provide details regarding *post-compromise* and *recurring* phases and how an attacker may move laterally from IT to DMZ or DMZ to levels 3 or 2 in OT. In fact, in this model, stage two (and its five phases) only applies to level 1 and level 0 of IIoT reference architecture that includes controllers and physical processes. Thus, all pre- and post-compromise phases and recurring steps certainly take place in the upper levels; attacker first compromises internet-facing levels (e.g., corporate or enterprise networks) and then moves laterally within the IIoT network towards lower levels in the OT zones. A very recent research on ICS cyber defense triage process (18) shows this concept based on Mandiant attack model and IIoT reference architecture. However, the researchers did not elaborate on how to map security alerts to each phase and architectural level.

3. SAMIIT

In this section, we propose an improved IIoT attack model that considers IIoT architectural levels, pre- and post-compromise and recurring phases in the attack life cycle. Figure 3 shows a comprehensive, granular attack model for IIoT that is color coded to show architectural IIoT zones previously illustrated in Figure 2. Additionally, we show the time typically required for the adversary to complete a phase successfully. We explain this model in the following paragraphs in more details.

3.1. Attack Phases and Zones

The pre-compromise part of the model is consistent with CKC and ATT&CK early phases. In order for the adversary to be able to compromise a target network, they would need to start with target identification and selection (i.e., external reconnaissance). Next, the adversary identifies weaknesses (e.g., technical, people, and organizational) and vulnerabilities that can be exploitable and then compromise the target network through weaponization, delivery, and external exploitation phases similar to CKC and ATT&CK. When the trojan is installed and successfully beaconed outbound to the remote controller, the network is compromised and the attacker has established a foothold in the victim's network. As observed in several cyber incidents in interconnected IIoT (1)(3)(6)(7), internet facing IT assets are the main target of pre-compromise phases. Therefore, Figure 3 suggests analysts to focus on security sensors at levels 4-5 to detect and prevent these types of malicious activity. In case of Stuxnet, however, it is concluded that the first phase was internal exploitation in the OT zone.

Once inside the network, the attacker constantly searches for new vulnerable systems to exploit and move laterally towards the final target. These are a set of recurring tasks (all internal) to find the next vulnerable point in the chain, exploit it, escalate the privilege and move forward (as shown in the gray area of Figure 3). Depending on the network and security architecture of the target network, the attacker may move horizontally within an architectural level or vertically to a lower level. In almost all of the IIoT cyber incidents (1) (2)(3)(4)(5) the final target was a Human Machine Interface (HMI) or a controller at level 2 and level 1, respectively (except for (9) that could be due to a different adversarial plan). Hence, the recurring phases can happen anywhere between IT, DMZ, and OT zone depending on the initial compromise point and the final target. This helps defenders effectively and efficiently assign security controls to the architectural levels(10) and identify the most relevant data sources to look for evidence.

When the last component in the chain is exploited, the attacker either quickly executes a malicious command or slowly exfiltrates confidential data (to avoid detection). In the case of IIoT though, the final target can be a level 1 Programmable Logic Controller (PLC) or a Remote Terminal Unit (RTU) that requires a longer time for the attacker to understand the physical process, develop a malicious configuration accordingly, and finally install and launch it. Therefore, the second weaponization phase in Figure 3 (in the blue area) refers to develop, test, and deliver phases of (17) that is taken outside the target network.

3.2. Attack Timeline

In order for defender to prevent the adversary from successfully achieving their desired objective, at least one step in the entire kill chain has to fail. If countermeasures are implemented faster than adversary's next steps, the adversarial costs will raise, and the kill chain will likely break (13). According to (23)(25), the average time to complete each phase is not necessarily the same and those that take longer provide an opportunity for defender to act faster that adversary. Therefore, it is critical for defenders to identify all critical phases in IIoT kill chain where there is enough time to break the chain.

We have shown the typical time required for completing each phase of IIoT attack life cycle in Figure 3. As depicted, there are phases that take longer and performed in a very low and slow manner, e.g., internal/target reconnaissance. During these phases, defenders have enough time to collect evidence for prior malicious activities, not the activities relevant to the current phases as they are likely not complete yet. In fact, an attack phase (e.g., exploitation or C2) that is normally followed by longer phases (e.g., internal reconnaissance or weaponization) is what defenders may invest more to efficiently prevent adversaries to proceed. For example, constantly looking for privilege escalation or lateral movement logs anywhere in IT to level 2 OT while attacker is performing another internal reconnaissance, or looking for HMI exploitation in level 2 while attacker is developing a malicious PLC configuration at their end, will help defenders take actions before the adversary moves.

3.3. Spiral Model

The gray area in Figure 3 can be extended to multiple cycles depending on the network and security architecture of the target environment. For example, if standard IIoT architecture (ISA-62443) is applied and zones are segmented by restrict firewall rules, the gray area will take very long and repeat multiple times (vertically and horizontally within the entire IIoT network). Inspired by OODA loop concept and its applications in APT simulation (23)(24), we propose SAMIIT, a Spiral Attack Model in IIoT (Figure 4) that not only does it include all the aforementioned details of attack life cycle (e.g., pre- and post-compromise, recurring phases, architectural levels), but also visualizes the life cycle for better analysis and mitigation.

Figure 4 depicts SAMIIT that consists of two parts: linear and spiral. The phases in the linear part (similar to the orange area in Figure 3), if successful, will result in a network breach (let's call it *exploitation* where a host is compromised). Assuming that in most IIoT incidents this breach starts from IT zone, the exploitation will be followed with several recurring phases until the adversary reaches to the final target and execute the last attack. Spiral part of SAMIIT includes both gray and blue areas of Figure 3 as well as architectural levels of IIoT networks. When *exploitation* is successful and a host is compromised, the adversary will either execute the final action and the attack life cycle ends (unlikely in IIoT) or laterally moves inside the network; whether horizontally in one level (circle) or vertically from one level (circle) to a lower level (inner circle). The white spiral arrow

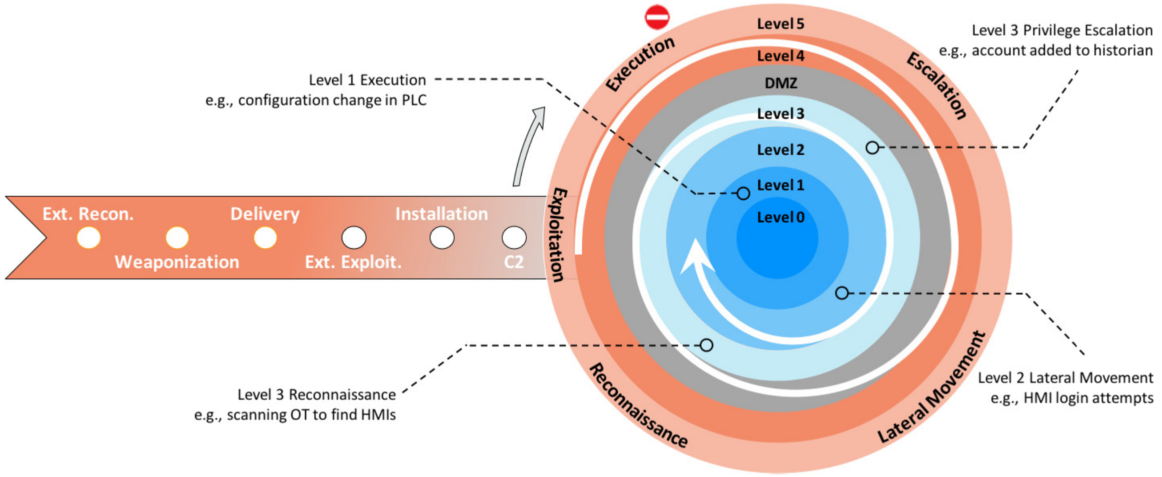

Figure 4: *SAMIIT: Spiral Attack Model in IIoT.*

in Figure 4 shows how the attacker can move within an IIoT network. For instance, a log that shows an account is added to the historian server will be mapped to level 3 privilege escalation, that can be followed by login attempts at the same level or scanning activities in a lower level to find HMIs.

Security analysts can use SAMIIT to visualize and classify security events to track adversary moves in each level and predict next possible activities. However, it is almost impossible for SOC analysts to review every single alert and manually map them to a sector and a level in that sector manually. Therefore, an automated event classifier that maps alerts/logs to these sectors and levels is of great help and improves the performance of SOC processes remarkably. In the next section, we propose a machine learning classification approach to map alerts/logs to SAMIIT.

4. MAPPING ALERTS TO ATTACK PHASES

Cyber security researchers have been applying neural network and data mining approaches to intrusion detection (or other security tools) for more than a decade. However, using machine learning based processes than can extract information automatically, without on-line guidance from users, is a promising approach in future SOC (26)(27). The machine learning algorithm proposed in (26), is used to help SOC analysts classify events based on their risk levels. In (27), a classifier is used to categorize intrusion detection alerts and determine which analyst should review what alerts (i.e., to better deal with large volumes of alerts). To the best of our knowledge, there is no prior work in using machine learning for labeling alerts with attack life cycle phases.

A fundamental question for this research is *why do we need a learning mechanism for automated alert mapping, and what is its value compared to conventional rule-based systems (e.g., domain specific expert systems)?* In a typical SOC environment, security events are collected from ~100 different sensors (commercial or open-source, standard or proprietary formats, etc.) and new tools and devices are always developed and added to SOC environments. Hence, the amount of data an analyst receives daily and the number of different data formats each sensor uses to report security events is countless. In fact, this number grows constantly with ever-evolving attack techniques and new security tools being developed. In addition, human mistakes in developing rule-based systems always exist. Thus, an automated mechanism for alert labeling (mapping to phases) that can learn from a large number of training examples that utilizes domain expert knowledge can produce a more accurate and efficient results. Since there are different types of machine learning systems based on human supervision, based on learning rate (e.g., batch or online), and based on their generalization, next research question is *what type of machine learning system is suitable for this application?*

In security operation environments, several security sensors generate alerts with similarities in their formats, source or destination hosts, severity levels, etc., that if used by unsupervised clustering algorithms will result in groups irrelevant to our attack phases. Additionally, analysts in different SOCs and IIoT networks may have different perspectives and customized attack phases that matches their environments properly. Therefor, it is more reasonable and practical to train the system with supervised data (i.e., a train dataset of alerts already labeled by experts). We will later discuss how semisupervised or reinforcement learning might be used for this application. Regarding learning rate, although new tools and alerts are constantly added to the system, the rate is not very high and learning can be performed offline (e.g., weekly or monthly on powerful SOC servers). Finally, generalization

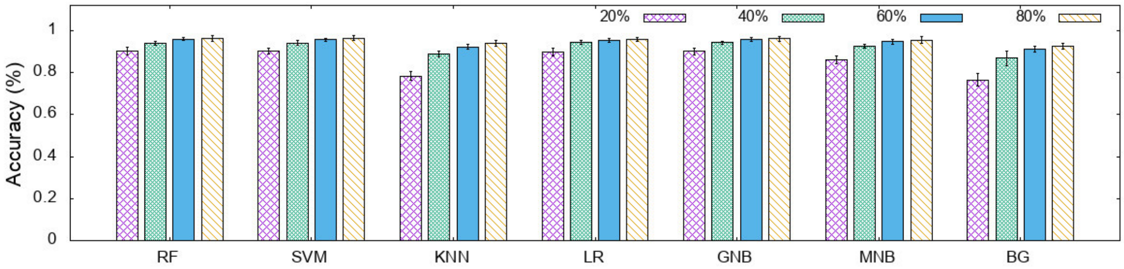

Figure 5: *Accuracy of Classifiers Applied to Mitre Dataset Using Six Features.*

means whether the learning system is instance-based or model-based. In this section, we will apply both model-based and instance-based algorithms and compare the results.

4.1. Proof of Concept Experiment

Mitre researchers have been working on an open-source project called Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK^{TM})[2] that provides a model and also a knowledge base for attack life cycle and target platforms of each phases. ATT&CK matrix categorizes adversarial techniques (i.e., how an attack is performed) into different tactics (i.e., the goal of the attack) in which a technique may span several tactics implying that they can be used for multiple purposes. The project covers both pre and post compromise sets of phases for Windows, Linux, and Mac platforms. More than 1500 techniques are added to this knowledge base where additional information such as permission required, example malware, data sources, and detection mechanisms for each technique are also available. Although these are neither actual alerts from SOC nor extracted directly from security event samples, the features and descriptions provided for each of them are similar to security logs and events (e.g., operating system logs or antivirus alerts) ingested to SIEM tools. In addition, we will show how this knowledge base and threat intelligence data can be used to enrich alerts before mapping them to attack phases. In other words, the additional information they can add to security alert and logs will remarkably improve alert labeling and consequently the performance of SOC analysts.

Each technique, a row in the dataset, has 16 features (columns) including the name of malware/tool that uses the technique, its type (software or group), the usage, data source to find a footprint of it, target platform, technical description, and multiple indexing numbers. Though, few of those features such as permission, network, and system requirements are missing for the majority of rows and we have to exclude them. We finally selected six features (columns) such as *Technique*, *Name*, *Usage*, *Data Sources*, *Technical Description*, and *Platform* to build a model that predicts *Category*. Category is a tactic or multiple tactics that a technique belongs to. ATT&CK suggests 10 individual adversarial tactics for post-compromise activities, however, there are 23 unique labels in the column *Category* in which the other 13 labels are different combinations of tactics. For example, “Service Registry Permissions Weakness” technique can be used for persistence or privilege escalation purposes (in ATT&CK definition), therefor it is labeled as “Persistence or Privilege Escalation.” It is worth mentioning that we do not interpret this as two different labels for that technique (as it negatively impacts the learning mechanism) and instead it is considered as a new class.

We implemented our machine learning algorithm using Scikit-learn library in Python. Multiple classifiers such as Random-Forest (RF), Support Vector Machine (SVM), K-Nearest Neighborhood (KNN), Logistic Regression (LR), Gaussian Naive Bayes (GNB) and Multinomial Naive Bayes (MNB), and Bagging were used for performance evaluation. In order to avoid overfitting or underfitting, we built our models using 20%, 40%, 60%, and finally 80% of dataset as the training data. For each classifier and split ratio, we repeated the experiment 20 times and calculated the average accuracy among them. Figure 5 shows the average accuracy of each classifier for different sizes of training data. The results show 0.95 and higher accuracy for all classifiers when 80% of data is used for training. For smaller training sets, however, an instance-based approach like KNN does not perform very well. Overall, RF, SVM, and GNB outperformed other algorithms with higher accuracy and very low standard deviation over 20 different runs for each experiment. A grid search was performed to find the optimal SVM parameters; the results are obtained using linear kernel while the c-value was set to 1. We note here that all other parameters were set to their default values in all of classifiers.

Security logs and alerts collected in real-world SOCs usually provide information such as time, source and destination addresses, data source, and a short message describing the event. When

[2] attack.mitre.com

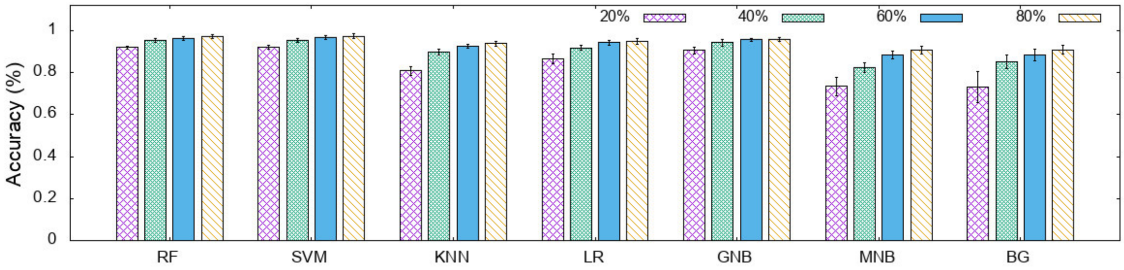

Figure 6: *Accuracy of Classifiers Applied to Mitre Dataset Using NLP on Technical Description Only*

compared with ATT&CK techniques, security alerts and logs are much less descriptive and the short description they provide is significantly helpful for analysts to learn about attack status. Hence, in another implementation, we reduced our feature set to only one (*Usage* or *Technical Description*) to see if techniques can be classified only based on their description. This is to show *how keywords in a security alert or log may help machine learning algorithms automatically map alerts to attack phases.*

We performed two experiments using a Natural Language Processing (NLP) technique known as Term Frequency - Inverse Document Frequency (TF-IDF). TF-IDF is a term weighting scheme to show how important a term is to a document. In the first experiment we used Technical Description field only and calculated the TF-IDF value of frequent words in that field for all samples in the training dataset. We then applied all aforementioned classifiers and split ratios for labeling techniques in the testing dataset. As depicted in Figure 6, the accuracy of top three classifiers (RF, SVM, and GNB) is similar to the results obtained in the initial six-feature experiment ($\geq$0.95), however, the accuracy of the other four classifiers is slightly lower ($\sim$0.02) than those obtained from the initial six-feature experiment. In the second experiment we used *Usage* field only but the accuracies were much lower. This is because *Usage* explains what tools are usually used in a technique or what the impacts will be; unlike Technical Description that explains the goal of the attack too. In conclusion, the results show the critical role a descriptive message can play in alert labeling. Thus, security tools can enhance mapping process by adding more context to alerts and logs regarding their consequences (i.g., attacker goals).

4.2. System Implementation

In this section, we show how SAMIIT can be used in ICS SOC and how the alert mapping algorithm performs in real-world implementations. We have created a test environment using several industrial, networking, and security devices and tools. Although developing an ICS testbed and its challenges are beyond the scope of this paper, we briefly present the architecture of our testbed and its IT and OT components. As shown in Figure 7, our testbed is based on standard IIoT architecture (ISA-62443) except for levels 2 and 3 that are merged. Level 0 has two I/O panels controlled by a PLC and two RTUs in level 1. There are two HMIs in level 2 that monitor and command lower level devices. Several workstation and servers (e.g., SIEM and endpoint security server) exist in the IT. The entire testbed is connected to a network switch and a firewall/router, and zoning is implemented using VLAN and firewall rules.

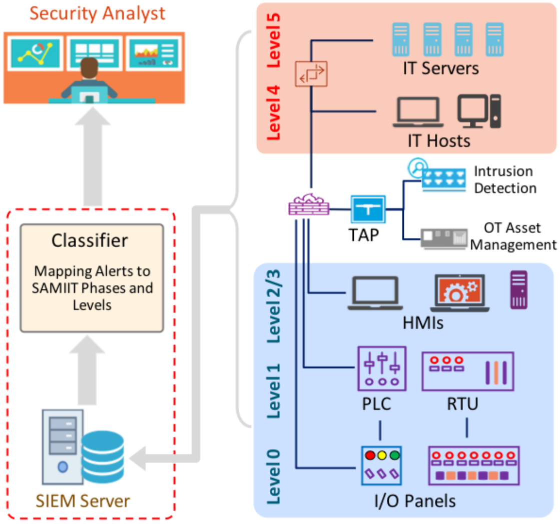

Figure 7: *Implementation on an ICS testbed*

We use Snort[3] intrusion detection running both IT and OT specific rules, and a commercial OT asset discovery and anomaly detection tool. They are both connected to the span port of the switch (using a network tapping as shown by TAP in Figure 7) to be able to inspect the entire ICS traffic. All devices from level 1 to level 5 send their security logs (e.g., syslog, IDS and anti-malware alerts) to a commercial SIEM. For this experiment, we launched multiple attacks against different architectural levels both manually and automated using penetration testing and APT simulation tools. We only use logs and alerts collected from firewall, Snort and the commercial asset discovery tool for performance

[3] www.snort.org

evaluation (∼135K samples for a month) since they cover the majority of events reported from IT and OT zones. Among them, there exists 12 types of attacks detected by Snort (e.g., access to vulnerable web app, information leak, web application attack), 4 types of alerts reported by firewall/router (mainly IT-OT traffic blocked and login failures), and 6 types of threats detected by OT asset discovery and anomaly detection tool (e.g., baseline deviation and new asset detection).

The main challenge for building a classifier and testing it against testbed dataset is that this dataset is not labeled. We previously discussed the drawbacks of unsupervised techniques and why we prefer supervised classification approaches for this problem. Using supervised algorithms, we performed two separate experiments on the dataset both based on NLP on the technical description field, and did not use the multi-feature approach. In the first experiment we used the models created by ATT&CK dataset (explained in Section 4.1) for labeling alerts collected from our testbed. In the other experiment we asked an expert to label a subset of alerts using domain knowledge and used that as the training set. In both experiments, mapping alerts to the architectural levels of SAMIIT is achieved through a lookup table created from OT asset discovery and domain knowledge; we use source/destination address and data source of each alert to identify the architectural level.

Since the dataset was not labeled, we were unable to calculate the accuracy rate, however, our analysis shows that in the first experiment, the model(s) built from ATT&CK dataset could not label SIEM dataset very accurately. For example, using different classifiers such as SVM or KNN, most of the alerts were labeled by only few labels such as command and control or defense evasion. Although this might be due to limited types of attack we have launched for this experiment, we believe that technical descriptions and terms extracted from them in our train dataset are highly enriched as compared to short description of SIEM alerts. For instance, any blocked communication between IT and OT is labeled as command and control that might be reconnaissance (or even exfiltration, but we know that we didn't launch this type of attack). More than 95% of alerts from OT asset discovery tool were labeled as command and control because they reported baseline deviation (i.e., a communication that was not observed during training mode of the device but happened during execution phase). Although it sounds reasonable, again it could be reconnaissance (device enumeration in OT) or even execution. Please note that another reason for all of these false labelings is because we only have post-compromise labels in our models built from ATT&CK, but the SIEM dataset includes other phases as well. Snort alerts were labeled more accurately since the alerts are more descriptive. This experiment shows although ATT&CK dataset can be helpful for labeling real-world SIEM alerts, it is not very effective to build a model only based on this dataset and directly apply it to SIEM tools. Security alerts are not as descriptive as ATT&CK dataset and have fewer features. One solution to this is to first enrich alerts using internal data, for example by adding source of information (e.g., API monitoring, system calls, process monitoring, process command-line parameters, loaded DLLs, Windows Registry), as well as ATT&CK data and then run it against the model. Another solution might be to use domain expert knowledge (and ATT&CK information) to manually label some real-world SIEM alerts for training sets, and then test it in operation.

In the second experiment, an expert labeled 7% of alerts (∼9K out of 136K) in a way that all of different attack types reported by the three tools had a sample in the training data. We then build an NLP-based model using the description of alerts and logs and labeled the rest of alerts. The results showed a surprisingly high accuracy rate. Although we cannot calculate the accuracy rate in operation mode due to unlabeled test data, an expert randomly checked alerts from different tools and types of anomalies and confirmed that (Note: we also used 50% of labeled data as training and tested it against the other 50% of data and the accuracy was 98%). In rare cases that the train samples were small, e.g., login attempts reported by firewall/router, the error rate was a little high (∼18% - 2 alerts out of 11). The expert believes that this is because login failure samples are labeled as lateral movement or privilege escalation in the training data (i.e., depends on the source of the attempt).

5. CONCLUSION AND FUTURE WORKS

In this research, we reviewed several attack life cycle models proposed for IT and IIoT networks. Considering some details missing in those models and their inability to accurately model IIoT complex attacks, we proposed SAMIIT, a spiral attack model to map IIoT cyber intrusions to different attack phases and architectural levels of IIoT environments. We then presented a machine learning classification approach for mapping security alerts to IIoT attack phases and architectural layers. As a proof of concept implementation, we evaluated our classifiers using ATT&CK dataset and showed the high accuracy of classification algorithms. We also investigated the performance of our proposed solution in an ICS testbed built using real-world devices and tools.

Taking the performance evaluation results into consideration, we envision several interesting future directions: alert enrichment in SIEM tools to provide more context to alerts and improve the accuracy of alert labeling mechanisms; reinforcement learning can be used to maximize the accuracy of alert labeling through online analysts' feedback and rewards; the security community needs to put together an ICS-specific ATT&CK type of dataset to educate analysts working in ICS/IIoT SOC; finally, we believe that alert labeling can help predictive analytics techniques to forecast adversarial options and next possible attacks, therefore being able to mitigate them faster and break the kill chain.

REFERENCES

[1] Dragos Inc., "Crashoverride: Analyzing the threat to electric grid operations," *Hanover, Maryland*, 2017.

[2] Symantec, "Cyberespionage attacks against energy suppliers," *Mountain View, California*, 2014.

[3] N. Nelson, "The impact of dragonfly malware on industrial control systems," *SANS Institute*, 2016.

[4] ICS-CERT, "Ongoing sophisticated malware campaign compromising ICS," *https://ics-cert.us-cert.gov/alerts/ICS-ALERT-14-281-01B*, 2014.

[5] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, vol. 5, 2011.

[6] G. E. Cyberattacks, "Night Dragon," *McAfee Foundstone Professional Services and McAfee Labs*, 2011.

[7] Defense Use Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.

[8] Accenture Security, "Dealing with the threats posed by triton/trisis destructive malware," *Industrial Control System Technical Report*, 2018.

[9] C. Wueest, "Targeted attacks against the energy sector," *Symantec Security Response, Mountain View, CA*, 2014.

[10] A. Hassanzadeh, S. Modi, and S. Mulchandani, "Towards effective security control assignment in the industrial internet of things," in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 795–800.

[11] A. Torres, "Building a world-class security operations center: A roadmap," *SANS Institute, May*, 2015.

[12] Ponemon Institute, "Challenges to achieving SIEM optimization," *Ponemon Institute LLC*, 2017.

[13] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[14] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding cyber threats with ATT&CK-based analytics," *Mitre*, 2017.

[15] Mandiant Intelligence Center, "APT1: Exposing one of China's cyber espionage units," *Mandiant*, 2013.

[16] B. D. Bryant and H. Saiedian, "A novel kill-chain framework for remote security log analysis with SIEM software," *computers & security*, vol. 67, pp. 198–210, 2017.

[17] M. J. Assante and R. M. Lee, "The industrial control system cyber kill chain," *SANS Institute InfoSec Reading Room*, vol. 1, 2015.

[18] A. Cook, H. Janicke, R. Smith, and L. Maglaras, "The industrial control system cyber defence triage process," *Computers & Security*, vol. 70, pp. 467–481, 2017.

[19] US-CERT, "Russian government cyber activity targeting energy and other critical infrastructure sectors," *https://www.us-cert.gov/ncas/alerts/TA18-074A*, 2018.

[20] L. Obregon, "Secure architecture for industrial control systems," *SANS Institute InfoSec Reading Room*, 2015.

[21] FireEye, "M-trends 2016," *https://www2.fireeye .com/M-Trends-2016.html*, 2016.

[22] Microsoft, "What threats does ATA look for?" *https://docs.microsoft.com/en-us/advanced-threat-analytics/ata-threats*, 2015.

[23] Sean Malone, "Using an expanded cyber kill chain model to increase attack resiliency," *Blackhat*, 2016.

[24] T. J. Grant, H. Venter, and J. H. Eloff, "Simulating adversarial interactions between intruders and system administrators using OODA-RR," in *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. ACM, 2007, pp. 46–55.

[25] A. Rege, Z. Obradovic, N. Asadi, B. Singer, and N. Masceri, "A temporal assessment of cyber intrusion chains using multidisciplinary frameworks and methodologies," in *International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*. IEEE, 2017, pp. 1–7.

[26] C. Feng, S. Wu, and N. Liu, "A user-centric machine learning framework for cyber security operations center," in *Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on*. IEEE, 2017, pp. 173–175.

[27] S. McElwee, J. Heaton, J. Fraley, and J. Cannady, "Deep learning for prioritizing and responding to intrusion detection alerts," in *Military Communications Conference (MILCOM), MILCOM 2017-2017 IEEE*. IEEE, 2017, pp. 1–5.