AUTO-TTPHunt: Active Investigation Threat Reasoning Framework for Attack Diagnosis and Predictability

Ehab Al-Shaer
UNC Charlotte
Charlotte, NC, USA
ealshaer@uncc.edu

Mohiuddin Ahmed
UNC Charlotte
Charlotte, NC, USA
mahmed27@uncc.edu

***Abstract*—**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

In recent year, there is an increase of attack including advanced persistence threat(APT) [1] and the techniques used by the attacker in these attacks have reached an unprecedented sophistication [2]. According to [1], [2], APT attacks have increased from approximately 500 attacks per year in 2009 to almost 1500 APT attacks per year in 2016. Those APTs use different attack techniques and previously unseen attack procedures, which can span up to multiple steps to reach a target. One such APT is sending spear phishing email [3] to get initial access on the target host, performing drive by download attack [4] to exploit a vulnerability in the target, and later exfiltrating confidential data from the compromised host [5], [6]. Sometimes, a typical attacker can perform an attack even with benign program. All of these attacker activities can be monitored and stored using different monitoring tools [7], [8]. The monitored activities will be stored in the end device as generated logs. However, these generated low-level logs lack high-level understanding of attacker activities and the relationship among different activities. Thus, defenders require a comprehensive method to extract the high-level attacker activities and the relationship among the activities, which can then be used to detect and predict future APT techniques or threats.

MITRE ATT@CK framework recently published a public knowledge base of 29 tactics, 485 techniques used by the APT. Every Mitre ATT@Ck tactic published is a high-level attacker goal in a specific kill chain phase. Every Mitre ATT@CK technique consists of one or multiple procedures the attacker used to achieve a certain goal whereas each procedure is a unique way to achieve corresponding goal. Mitre ATT@CK framework also provides 91 APT group and a subset of publicly reported technique used by each APT Group. Tools like RedCannary, Carbon black, OSQuery, LogRythm developed detectors and mapped each detector to the mitre technique. However, those detectors can detect only single stage of an APT attack.

The more recent trend in cybersecurity is not only to detect attack but also to predict future threats based on current attacker behavior. Previous work on attack prediction [4], [9]–[11] can predict only binary outcome like as whether an attack will happen or not. For example- [12] predicts whether a data breach will happen or not, [13] predicts whether a host or website will be compromised by the attacker. Those approaches are not able to give what are the risky component of the system going to be attacked, what are the exact action an attacker is going to perform. To predict those exact actions of the attacker, [12]proposes a framework based on deep learning.

All of the approaches described up to now are based on passive monitoring through Sysmon [14] or network traffic monitor like intrusion detection system [7]. However, an attacker can delete generated logs and can also add spurious logs to misdirect the defender. Deletion of activity logs and addition of spurious activity logs will degrade performance of the passive approaches. In this case, active monitoring through investigation of the unobserved or conflicting behaviors will increase the accuracy of the system. However, active monitoring will increase the detection time of the system. Thus, a trade-off need to be determined when to use passive or active approach or a combination of both [15].

New Inro

Though threat hunter uses logs generated by existing rule- or signature based detection systems and correlates the logs using security information and event management systems such as SPlunk, QRadar; those detection system can be easily evaded by the use of polymorphic worms. Additionally, those systems detect attack at the later stage of the attack and work in a reactive manner rather than being proactive. Cyber threat hunting process also uses IOCs extracted from cyber threat intelligence. However, IOC does not provide any threat context and can be easily evaded by changing the IOCs. Moreover, there are too many logs or observations generated by the attacker or benign user activities which makes monitoring everything an impossible task. Thus one of the goal of this paper is to determine what is the minimum number of observations that can produce maximum information gain which will require online evidential reasoning as a combination of passive and active diagnosis.

The recent trend in cyber threat hunting is the use of Mitre Attack framework with traditional attack detection system such as IDS, SIEM. Threat hunter uses cyber threat analytic for attack analysis. Cyber Threat hunting is a proactive process to identify attack whereas Intrusion detection and prevention system are reactive taking action after getting the alert [?], [?]. Hunting is about taking a proactive as opposed to a reactive approach to identifying incidents. And threat hunters are not simply waiting to respond to alerts or indicators of compromise (IOCs). They are actively searching for threats to prevent or minimize damage. The goal of threat hunting is to detect and respond to the attacker activities as early in the kill chain as possible. Hunting starts with a good question, selecting the best hypothesis to verify [?], [?]. Otherwise, it will be like looking for a needle in a hey-stack. The key, however, is to constantly look for attacks that get past security systems and to catch intrusions in progress rather than after attackers have completed their objectives and done worse damage to the business. Organizations need to consider that hunting is more than advanced security monitoring and move their hunting mindset strategically through targeted approaches using threat intelligence to provide hunting accuracy. The advancement of attacker capabilities and techniques makes detection of the attack a daunting task [?], [?]. However, the trails left by the attacker actions known as indicator of compromise and the casual relationship among the adversary actions and IOCs are a promising indicators to determine hypothesis to verify and starting point of attack analysis.

Problem to solve. The problem we are going to solve here is to detect an ongoing attack(can span a long period of time) goal an attacker trying to achieve during the cyber threat hunting process with the use of reported(observed) techniques, what are the techniques used by the attacker(backward analysis) and predict the additional technique the attacker going to use to achieve corresponding goal.

The problem has following characteristics to be considered-

- Hypothesis Generation. Since the first steps of a cyber threat hunting process is to select a plausible hypothesis based on the observations and verify it, selection of the most plausible hypothesis should be done efficiently. Otherwise, time and resources will be wasted on verifying a wrong hypothesis.
- Hypothesis Validation. Selected hypothesis needs to be validated considering all of the TTP chains used to achieve at least one goal in the hypothesis.
- Investigation of Unseen Technique. If a hypothesis can not be validated using observed techniques, active investigation needs to be performed. However, verifying all unseen techniques using active investigation is not a feasible solution because of cost and time constraints. Thus, minimum number of active investigation action that can verify maximum number of unseen techniques within a budget constraint needs to be selected.

Contributions. By addressing all the above mentioned characteristics of the problem, Auto-TTPHunt presents a framework to detect goal of an attack and predict additional techniques the attacker is going to used. Auto-TTPHunt starts with a list of observed technique and a correlation matrix between the technique and goal. Our key contributions and importance of them are described below, details of each contribution is given in section V.

The first contribution of AUTO-TTPHunt is to generate a plausible hypothesis as a list of goals the attacker is trying to achieve. An algorithm is given to generate the hypothesis by considering the goal with highest contribution in explaining observed techniques and given correlation matrix. This hypothesis will be used by the hunters as a starting point of hunting process.

Second, A scientific metric is provided to validate the generated hypothesis based on the goals in the hypothesis, TTP chain used to achieve the goal, observed techniques and the given correlation matrix. The use of a metric to calculate the fidelity of a hypothesis is required to avoid unnecessary active investigations. It can be case that enough techniques are already observed and correlation among the observed technique to build the TTP chain is sufficient to verify that hypothesis through passive evidential reasoning which is the indication of high fidelity score. However, situation can arise in which case enough observations are not available and the hypothesis can not be verified without performing further investigation which is the indication of low fidelity score.

The third contribution of this paper is to provide heuristics to select minimum number of investigation action that can verify maximum number of unseen techniques within a budget constraint. Since investigating all unseen techniques is infeasible and one investigation action can verify one or multiple techniques, a list of investigation actions are determined based on the data sources of each techniques where data sources are the types of event logs we have to monitor to verify corresponding technique.

At the end, attack goals are detected based on the improved fidelity score and techniques used by the attack to achieve the goal will be reported by performing backward analysis on the TTP chains used. The additional technique the attack is going to use to achieve the validated goals will also be predicted by performing forward analysis on the TTP chains used.

Evaluations. AUTO-TTPHunt is evaluated using synthetic attack data and different attack use cases. Synthetic attack data is generated by using the correlation of techniques as reported in 91 APT groups provided by Mitre ATT@CK framework. Since order of technique occurrence in the 91 APT group reported by MITRE is not provided, synthetic attack chain are generated based on the technique usage trends present in the already reported APT groups. The evaluation of Auto-TTPHunt using synthetic attack data and APT usecases shows that it can determine what is the goal of the attacker based on the observed techniques and can also predict which additional techniques can be used by the corresponding attack to achieve that goal. The details of the data generation and evaluation results are reported on section VIII.

II. Related Work

III. Approach Demonstration through an Example

IV. Motivation, Problem Formalization and Challenges

A. Motivation

Though there exist many rule- or signature-based detection systems, those are ineffective because they can be easily evaded (use of polymorphic malware) and they detect attack in later stage of kill chain phase and works in a reactive manner rather than being proactive. Additionally, detection system based on IOC can also be easily evaded by changing IOC and existing IOC extraction system does not provide any threat context of the IOC which makes the determination of threat priority a infeasible task. Again, it is infeasible to monitor everything in a machine, attacker can remove evidence and misdirect defender by reporting spurious evidence in a compromised host. Those situations will reduce the accuracy and performance of the passive attack diagnosis which requires active investigation of the reported observables. However, active threat diagnosis does not scale well because of the high cost of active testing and performance overhead on the networks and host machines, and excessive number of observables reported for each host machine and corresponding network devices. Moreover, some attacks using C2 communication through uncommonly used port, protocol and encryption techniques will not be even identified if only active diagnosis of reported observables is applied. However, those type of attacks can be easily detected if passive diagnostics of reported observables is used since abnormal usage of those ports, protocols and encryption techniques will increase the entropy of the corresponding observables which is a indicators of a specific attack using C2 communication. Thus, the integration of active and passive attack diagnosis is paramount to localize attack and develop efficient attack diagnosis solutions.

To integrate active and passive attack diagnosis, we propose an Observable-Technique-TTP-Goal-InvestigationAction model as shown in figure 1. In this model, investigation action(IA) can be performed by the monitoring device already installed in the network of hosts. Those investigation actions include analysis of network traffic for abnormal data flow(e.g. client sending significantly more data than it receives), uncommon network utilization(usage of protocols and ports that are rarely seen), and monitoring process, registry key access, file system access and modification, and many to count. Since any actions that are performed should be reflected in the generated logs, monitoring of corresponding entity will reflect the absence or presence of malicious observables which can be used to verify missing or spuriously reported observables.

B. Problem Definition

In this paper, $G = \{g_1, g_2, ..., g_n\}$ denotes set of goals which can be achieved by adversary through the execution of a set of techniques (C). Such a set of techniques is called TTP. These goals can be considered as terminating techniques of TTPs. $Q = \{q_1, q_2, ..., q_\mathrm{m}\}$ will be used to denote set of attack techniques as proposed by the MITRE attack framework, $A = \{a_1, a_2, ..., a_\mathrm{n}\}$ denotes a set of investigation action which will be used to verify an unobserved observable and $O = \{o_1, o_2, ..., o_\mathrm{y}\}$ denotes set of observables that will be used to diagnose attack techniques which are responsible for these observables. The Observable-Technique-TTP-Goal-InvestigationAction model can be considered as an *k-partite graph* where $k = 5$. In this graph, every observable is associated with one single *technique (q)* and a set of techniques forms a *TTP (C)* which can be used by the attackers to reach a $goal(g_\mathrm{i})$.

Notation	Definition
O_{q_i}	A set of all observables that is caused by technique q_i
O_O	A set of all reported observables so far
O_{O_i}	A set of observed observables that are caused by technique q_i
O_{U_i}	A set of lost observables that are caused by technique q_i
h_i	A possible hypothesis consisting of set of goals which can explain O_O or Q_O
ϕ	A set of all different threat hypothesis, h_i where each h_i can fully explain O_O or Q_O
O_N	A set of diagnosed but not-yet-observed observables associated with any technique that is used to reach any goal in the selected hypothesis
O_U	A subset of O_N containing observables, the non-existence of which is confirmed
O_V	A subset of O_N containing observables, the existence of which is verified
G_{q_i}	A set of goals which are reachable from the technique q_i
$C_{C_j}(g_i)$	Contribution of goal g_i in technique chain or TTP C_j
Q_O	A set of reported techniques
Q_h	A set of techniquesfrom where at least one of the goal of hypothesis h is reachable
$Q_{O_{C_j}}$	A set of reported techniques that is used in TTP C_j
Q_{C_j}	A set of techniques that is used in TTP C_j which contains both reported and not seen techniques
Q_{O_i}	A set of reported techniques that is used to reach goal g_i
Q_{g_i}	A set of techniques from where goal g_i is rachable

TABLE I
Active Investigation Threat Reasoning notation

The causality matrix $P_{\mathrm{Q} \times \mathrm{G}} = \{p(q_\mathrm{i}|g_\mathrm{j})\}$ defines the causal certainty between technique $q_\mathrm{i}(q_\mathrm{i} \in Q)$ and the goal $g_\mathrm{j}(g_\mathrm{j} \in G)$ that the attackers are trying to achieve by executing technique q_i. Since most of the techniques have unique observables, the posterior probability, $p(o_\mathrm{k}|q_\mathrm{i})$ will be either close to 1 or 0. The causality matrices can be deterministic or probabilistic. If $p(q_\mathrm{i}|g_\mathrm{j}) = 0 \textit{ or } 1$, we consider such a model deterministic model; otherwise we call it a probabilistic model. We will also use $A = \{a_1, a_2, ..., a_\mathrm{m}\}$ to denote set of investigation actions which will be used to verify the existence of certain unreported observables. These investigation actions are collected from adversarial tactics, techniques and procedures, and corresponding data source which is defined by MITRE ATT&CK framework through common knowledge gathered by research and post access adversary activities. The relationship among investigation actions and observables can be represented as a bipartite graph as shown in lower part of figure 1. Hence, the *Observable-Technique-TTP-Goal-InvestigationAction* graph can be described as a 9-tuple $(G, C, Q, O, A, E_1, E_2, E_3, E_4)$, where goal set G, TTP set C, attack technique set Q, Observable set O and investigation action set A are five independent vertex sets. Every edge in E_1 connects a vertex in G with another vertex in C to indicate what are the attack TTP or set of techniques required to achieve a goal. Every edge in E_2 connects a vertex in C with another vertex in Q to define what are the techniques (Q) that forms a TTP (C) to achieve a goal (G). Every edge in E_3 connects a vertex in Q with another vertex in O to denote the casual relationship among attack techniques and observables. Every edge in E_4 connects a vertex in A with another vertex in O to denote which observables can be verified by each investigation action. The notations used in this paper to describe active investigation cyber threat reasoning are introduced in Table I.

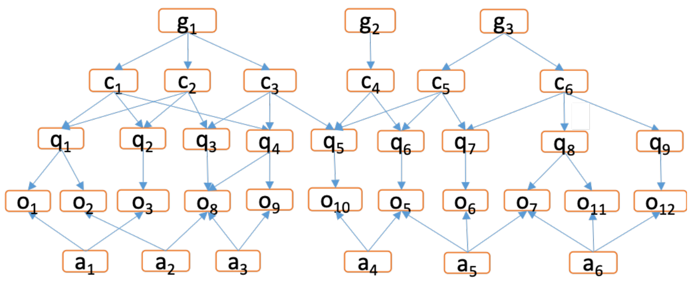

Fig. 1. Observable-Technique-TTP-Goal-InvestigationAction Model

The basic *Observable-Technique-TTP-Goal-Investigation Action* model can be described as follows-

- For every goal, associate a goal vertex $g_{\mathrm{j}},\ g_{\mathrm{j}} \in G$;
- For every TTP which is a set of technique, associate a TTP vertex $c_{\mathrm{m}},\ c_{\mathrm{m}} \in C$;
- For every attack technique, associate an adversarial technique vertex $q_{\mathrm{i}},\ q_{\mathrm{i}} \in Q$;
- For every observable, associate an observable vertex $o_{\mathrm{k}},\ o_{\mathrm{k}} \in O$;
- For every investigation action, associate an investigation action vertex $a_{\mathrm{i}},\ a_{\mathrm{i}} \in A$;
- For every goal g_{j}, associate an edge to each TTP c_{m} which is executed by the attacker to reach goal g_j;
- For every TTP c_{m}, associate an edge to each technique q_{i} which is executed by the attacker to reach goal g_j ;
- For every technique q_{i}, associate an edge to each observable o_{k} which is observed for the execution of this adversarial technique;
- For every investigative action a_{i}, associate an edge to each observable o_{i} with a weight equal to investigative action cost incurred to verify corresponding observable;

The evaluation of the proposed solution to diagnose attack includes two factors- performance and accuracy. The threat detection time, DT, which is the time between observable report time and the time when a threat is detected is synonymous with performance of the system. The less time required to diagnose a threat, high will be the performance of the system. Accuracy of threat diagnosis depends on two factors:

1) The threat detection rate (α), number of truly detected threat instance(T_{d}) to total number of occurred threat instance(T_{t}), formally $\alpha = \frac{|T_d \cap T_h|}{|T_h|}$;
2) The false positive rate (β), number of falsely detected threat instance to total number of detected threat instance(T_{d}), formally $\beta = \frac{|T_d - T_d \cap T_h|}{T_d}$;

Hence, the goal of the proposed system is to minimize the threat detection time, DT and the false positive rate, β and maximize the threat detection rate, α.

C. Challenges

Since Auto-TTPHunt expects reported observable as Mitre technique and the attacker activities reported in end host and network devices are low-level event log data, the first challenge is to convert reported low-level event logs to corresponding Mitre technique. Tools like RedCannary already developed almost 800 detectors mapped to different Mitre techniques which can be used to get Mitre technique corresponding to each reported observable as attacker activity. Tools like LogRythm with OSQuery can also be used to detect Mitre attack technique for a reported attacker activities.

Since there is no standard list of goals an attacker can try to achieve, another challenge is to determine what are the list of goals an attacker can try to achieve.

Moreover, an attack can span to multiple hosts and network devices. After converting low-level event logs to attack technique, the next challenge is to stitch together all of the techniques that belongs to the same attack. Since the reported event logs and corresponding attack techniques contain host address and process id related to those techniques, stitching techniques belonging to the same attack is trivial.

The first step of cyber threat hunting is to select a threat hypothesis. However, selecting a hypothesis and trying to verify it without following a scientific method may waste a lot of time of hunter and hunting resources. Additionally, it is not feasible to monitor all activities in all end hosts and network devices. Considering the partial observability of attacker activities, the fourth challenge to solve is to select the best hypothesis based on the observed techniques that leads to optimized detection and accuracy of the threat. The details of hypothesis generation is described in section V.

Since it is infeasible to monitor everything in a network, performing only passive diagnosis through hypothesis generation and selection will fail to diagnose attack which calls for the active investigation of unobserved evidence in the selected hypothesis. However, if sufficient evidence is available, sometimes passive diagnosis is enough to localize an attack. Thus, determining the trade-off between passive diagnosis through hypothesis generation and selection, and active investigation is another challenge. The details of trade-off determination is described in section V.

Given the selected hypothesis and a list of possible TTP which can be used by the attacker to reach goals in hypothesis, the threat hunters require to investigate unobserved techniques in the TTP. However, investigating all of the unobserved techniques is not a feasible solution. Selection of minimum number of investigation action that can verify maximum number of unobserved techniques is another challenge to solve. The details of investigation action selection is described in section V.

In summary, the development of the proposed solution includes following three problems to solve-

1) Given the correlation matrix among attack goal and Mitre technique, construct a set of the most plausible hypothesis, $\Phi = \{h_1, h_2, ..., h_\mathrm{q}\}$, which can explain currently observed evidences;
2) Given a set of plausible of hypotheses, find the most credible hypothesis h, which can give the best explanation for the currently observed evidences;
3) Given the most credible hypothesis h that gives best explanation to the observed techniques, determine the trade-off between passive diagnosis and active investigation.
4) If the selected hypothesis doesn't satisfy the fidelity requirement, then given the unobserved evidences Q_N, find the minimum cost investigative action set to search for an acceptable hypothesis;

In the next section, we are going to discuss solution of these four problems in cyber security threat hunting.

V. Active Investigation Threat Reasoning Framework Overview

The Active Investigation Threat Reasoning (*Auto-TTPHunt*) framework (Fig. 2) includes three functional modules: Threat Reasoning (FR), Fidelity Evaluation (FE), and Investigative Action Selection (IAS).

Reported observables are converted to observed attacker technique through the use of TTPDrill [16] and feeds to the TR module as input. The Threat Reasoning module takes passively observed techniques Q_O as input and returns the threat hypothesis set Φ as output. The threat hypothesis set Φ might include a set of hypotheses $(h_1, h_2, \ldots, h_n)$, where each one contains one or multiple goals that explains all observed techniques up to that point. Then, Φ is sent to the Fidelity Evaluation module to check if any hypothesis h_i ($h_i \in \Phi$) is satisfactory. If most correlated techniques necessary to explain the threat hypothesis h_i are observed (i.e. high fidelity), then the threat reasoning process terminates. Otherwise, a list of unobserved techniques, Q_N, that contribute to explain the threat hypothesis h_i of the highest fidelity is sent to the Investigative Action Selection module to determine which TTP have been executed by the attacker to achieve a goal in hypothesis h_i. If there are multiple hypotheses having the same highest fidelity, we choose the one having $min(Q_N)$.

The Investigative Action Selection module will select the minimum number of investigation action to verify maximum of the unobserved techniques within a budget constraint. After the investigative action selection and verification of unobserved techniques Q_N within a budget constraint, the IAS module will return a set of verified techniques Q_V and non-verified techniques Q_U(non-existing), and will be sent the FR module to recalculate the fidelity score by adding verified techniques, Q_V in input set of observed techniques. The corresponding fidelity score might be increased or decreased based on the selected investigative action in IAS module. If the newly calculated fidelity score is satisfactory, then the reasoning process terminates; otherwise, (Q_O, Q_U) will be sent as new input to the Threat Reasoning module to create a new hypothesis. This process is repeated until a hypothesis with high fidelity score is found. The calculation of fidelity score is explained later in this section. In the following section, we describe the three modules in detail and the complete active investigation threat Reasoning algorithm.

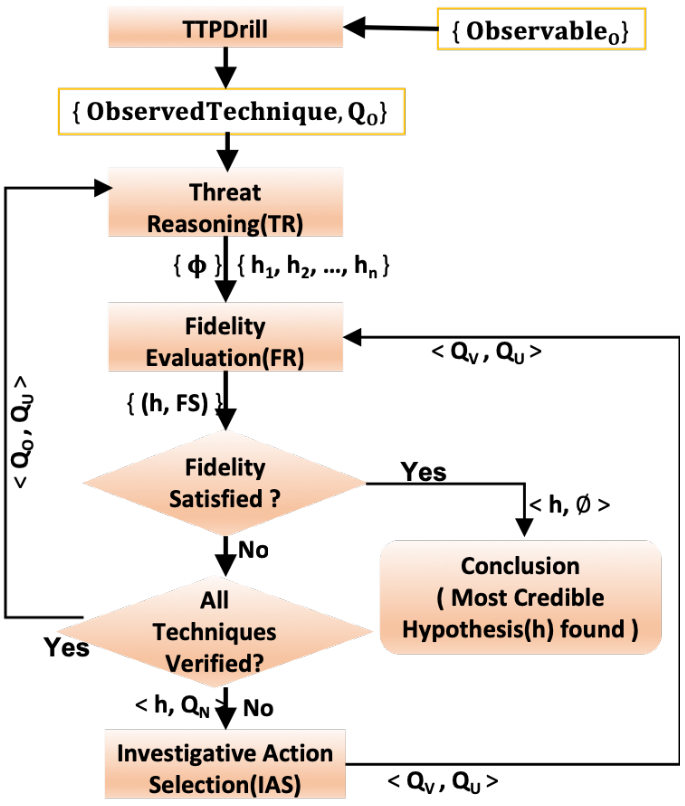

Fig. 2. Active Investigation Threat reasoning Approach

VI. ThreatHunter: Active Investigation Threat Reasoning

In this section, we will elaborate the development of active investigation threat reasoning system (*Auto-TTPHunt*) for attack diagnosis which uses both passive reasoning and active investigation of observed techniques. The goal of Auto-TTPHunt is to balance the use of active and passive measurements. Here, Active Investigative action selection process will be used only if the passive reasoning is not sufficient to explain adversary activities. If there are enough and highly quality reported observables, then use of active investigation may not be necessary.

A. Observales to Technique mapping

Since the threat reasoning model will take technique as input and techniques are high level procedure used by attacker to achieve a mean whereas observables are low level event logs, a mapping among observables from low level event logs to high level Mitre technique needs to be created. Though, generation of mapping between attacker activities and Mitre technique is out of the scope of this paper, there are previous research work TTPDrill [16] and tools like RedCannery [?], logRythm with OSQuery [?] exist. For our paper, we used TTPDrill to generate the mapping.

B. Heuristic Algorithm for Threat Reasoning

The Threat Reasoning module will use contribution function, $C(g_i)$ as a scientific matrix to find set of techniques or TTP that is used to reach goal, g_i based on the maximum contribution of goal, g_i on explaining the observed techniques used by the attackers. In the probabilistic model, technique q_i can be used by the adversary to achieve one goal or multiple goals, each $g_i, (g_i \in G_{q_i})$ with different posterior probabilities of terminating techniques (goals). We assume that our used correlation matrix between technique and goal is sufficient enough to explain attack TTP which implies that prior probability of a technique execution or a goal is very low. Hence, technique q_i will not be reported or observed if the adversary do not try to achieve any of the goal g_i. Strictly speaking, if q_i is reported or observed, the attacker is trying to achieve at least one $g_i(g_i \in G_{q_i})$. However, posterior probability $p(q_i|g_i)$ itself does not truly reflect attackers' chance of achieving goal g_i by considering the observed techniques q_i used by the attacker. For example, in Fig. 1, there are two possible goal that the adversary are trying achieve from observing q_6: (1) g_2 can be attained using TTP C_4, or (2) g_3 can be attained using TTP C_5, or (3) both g_2 and g_3 can be attained using TTP C_4 and C_5.

In order to measure the contribution of each goal g_i to the observation of q_i, we normalize the conditional probability $p(q_i|g_i)$ to the normalized conditional probability $\tau(q_i|g_i)$ to reflect the relative contribution of each goal g_i to the observation of q_i.

$$\tau(q_i|g_i) = \frac{p(q_i|g_i)}{\sum_{g_i \in G_{q_i}} p(q_i|g_i)} \tag{1}$$

using $\tau(q_i|g_i)$, we can compute normalized posterior probability $\mu(g_i|q_i)$ as given below-

$$\mu(g_i|q_i) = \frac{\tau(q_i|g_i)p(g_i)}{\sum_{g_i \in G_{q_i}} \tau(q_i|g_i)p(g_i)} \tag{2}$$

$\mu(g_i|q_i)$ denotes the relative probability of achieving goal g_i by observing the technique q_i.

The contribution function $C(g_i)$ given below is the explainability of goal g_i given the observed techniques. It evaluates all contribution factors $\mu(g_i|q_i)$, $q_i \in Q_{O_i}$, and selects the g_i with maximum contribution value $C(g_i)$ to the currently unobserved techniques as the best candidate.

Algorithm 1 Hypothesis Update Algorithm **HU**(h, Q_K, G_C)

Input: hypothesis h, observed but uncovered technique set Q_K, goal candidate set G_C
Output: threat hypothesis set Φ

```
1: $c_{max} = 0$
2: for all $g_i \in G_C$ do
3:     if $C(g_i) > c_{max}$ then
4:         $c_{max} \leftarrow C(g_i)$
5:         $G_Q \leftarrow \emptyset$
6:         $G_Q \leftarrow G_Q \cup \{g_i\}$
7:     else
8:         if $C(g_i) = c_{max}$ then
9:             $G_Q \leftarrow G_Q \cup \{g_i\}$
10:        end if
11:    end if
12: end for
13: for all $g_i \in G_Q$ do
14:     $h_i \leftarrow h \cup \{g_i\}$
15:     $Q_{K_i} \leftarrow Q_K - Q_{O_i}$
16:     $G_{C_i} \leftarrow G_C - \{g_i\}$
17: end for
18: for all $Q_{K_i}$ do
19:     if $Q_{K_i} = \emptyset$ then
20:         $\Phi \leftarrow \Phi \cup \{h_i\}$
21:     end if
22: end for
23: if $\Phi \neq \emptyset$ then
24:     return $< \Phi >$
25: else
26:     /* No $h_i$ can explain all $Q_O$*/
27:     for all $h_i$ do
28:         $HU(h_i, Q_{K_i}, G_{C_i})$
29:     end for
30: end if
```

$$C(g_i) = \frac{\sum_{j=1}^{n} C_{C_j}(g_i)}{n} \tag{3}$$

$$where, C_{C_j}(g_i) = \frac{\sum_{q_i \in Q_{O,C_j}} \mu(g_i|q_i)}{\sum_{q_i \in Q_{C_j}} \mu(g_i|q_i)} \tag{4}$$

$C_{C_j}(g_i)$ is the explainability of goal g_i in TTP C_j given the observed techniques.

Therefore, threat reasoning becomes a process of searching for the goal (g_i) with maximum $C(g_i)$. This process continues until all observed techniques are explained. The contribution function $C(g_i)$ can be used for both the deterministic and probabilistic models.

In the deterministic model, the higher the number of observed techniques, the stronger the indication that the attacker are trying achieve the corresponding goals. Meanwhile, we should not ignore the influence of prior technique or goal (terminating technique) probability $p(q_i)$ or $p(g_i)$, which represents long-term statistical observation. Since $p(q_i|g_i) = 0$ or 1 in the deterministic model, the normalized conditional probability reflects the influence of prior probability of technique q_i. Thus, the same contribution function given in equation 3 can be used to combine the effect of $p(q_i)$ or $p(g_i)$, and the ratio of $\frac{|q_{O_i}|}{|q_{g_i}|}$ together.

The threat reasoning algorithm first finds the candidate goal set G_C, including all goals that can be reachable from at least one observed technique q_i, ($q_i \in Q_O$). Then it calls the function $HU()$ to generate and update the hypothesis set Φ until all observed techniques Q_O is explained. According to the contribution $C(g_i)$ of each goal g_i, ($g_i \in G_C$), Algorithm 1 iteratively searches for the best explanation (i.e. the goal with the highest contribution) (lines 4-6) of Q_K, which are the currently observed techniques not explained by the hypothesis h_i (lines 4-12). Here $Q_K = Q_O - \cup_{G_i \in h_i} Q_{O_i}$ (lines 14-16) and initially $Q_K = Q_O$. If multiple goals have the same contribution, multiple hypotheses will be generated (lines 13-17). The searching process (HU) will recursively run until all observed techniques (i.e., Q_{K_i}) are explained (i.e., $Q_{K_i} = \emptyset$) (lines 18-24). Note that only those hypotheses with a minimum number of goals that cover all observed techniques are included in Φ (lines 23-24).

This threat Reasoning algorithm can be applied to both deterministic and probabilistic models with the same contribution function $C(g_i)$ but different conditional probability $p(q_i|g_i)$.

1) Finding TTP:

C. Fidelity Evaluation of Threat Hypotheses

The Threat hypotheses generated by the Threat Reasoning algorithm may not accurately determine the correct TTP used by the adversary because of the absence of sufficient attacker activities or the deletion of observables by the attackers or insertion of spurious attacker activities to misdirect the defenders or the false positive rate associated with each technique detection. Thus, the task of the Fidelity Evaluation is to measure the credibility of the hypothesis created in the reasoning phase given the corresponding observed techniques where objective evaluation of the threat reasoning is crucial in threat hunting process.

Fidelity function $FD(h)$ is used to measure the plausibility of the hypothesis h given the observed techniques by using the fidelity score of each TTP chain used to achieve at least one goal from hypothesis h. We use the Refutability function $FD(h)$ to measure the credibility of a TTP c_i in hypothesis h given the observed techniques Q_O by subtracting refutability score, $R(c_i)$ of hypothesis h from 1.

- For the deterministic model:

$$R(c_i, h) = \sum_{g_i \in h} \frac{|Q_{U_i}|}{|Q_{g_i}|} \tag{5}$$

- For the probabilistic model:

$$R(c_i, h) = \frac{p(Q_{c_i}|h)}{p(Q_{U,c_i}|h)} \tag{6}$$

$$FD(h) = 1 - R(c_i, h), \; where\, i = \arg\min_j R(c_j, h) \tag{7}$$

Where, $P(Q_{C_i}|h)$ is the probability of all required techniques(observed and unobserved) of TTP c_i occurred as true positive given the hypothesis h and $P(Q_{U,c_i}|h)$ is the probability of all unobserved techniques of TTP c_i occurred as true positive given the hypothesis h.

$$p(Q_{c_i}|h) = \prod_{q_i \in \bigcup_{g_i \in h} Q_{c_i, g_i}} (1 - FP(q_i))(1 - \prod_{g_i \in h} (1 - p(q_i|g_i))) \tag{8}$$

$$p(Q_{U,c_i}|h) = \prod_{q_i \in Q_{U,c_i}} (1 - FP(q_i))(1 - \prod_{g_i \in h} (1 - p(q_i|g_i))) \tag{9}$$

Thus, refutability score of each TTP, c_i is calculated and the TTP with lowest $R(c_i)$ will be selected as the plausible TTP since the TTP with lowest $R(c_i)$ will be the TTP with highest fidelity score and will be the most plausible TTP to achieve a goal in the hypothesis, h.

Obviously in the deterministic model, if the hypothesis h is correct, $FD(h)$ must be equal to 1 since the corresponding techniques can be either observed or verified. In the probabilistic model, if related techniques are observed or verified, $FD(h)$ of a credible hypothesis can still be less than 1, since some evidence of execution of techniques may not present in the reported observables even when the hypotheses are correct and the false positive rate of the technique detection. In both cases, our fidelity algorithm takes into consideration a target Fidelity Threshold, FD_{TH}, that the user has to configure to accept the hypothesis. System administrators can define the threshold based on long-term observation and previous experience. If the threshold is set too high, even correct hypothesis will be ignored and will increase false negative rate of threat detection; however if the threshold is too low, then a less credible hypothesis might be selected and will increase the false positive rate of threat detection.

The fidelity evaluation function, $FD(h)$ is used to evaluate each hypothesis and decide if the result is satisfactory by comparing them to the pre-defined threshold value. If an acceptable hypothesis that matches the fidelity threshold exists, the threat diagnosis process will terminate. Otherwise, the best available hypothesis and a non-empty set of techniques (Q_N) or observables (O_N) would be sent to IAS module to determine a satisfactory hypothesis in the next iteration.

D. Action Selection Heuristic Algorithm

During the hypothesis generation, the goals in the generated hypothesis advertises TTP to achieve the goal where the TTP contains both the observed (Q_O) and unobserved (Q_N) techniques. Those unobserved techniques are required to be verified through the active investigation to validate the hypothesis. The reason behind the verification of observables or techniques rather than goals is that observabels are visible evidence of the techniques used to achieve the goals which is easier to track and verify. The task of Investigative Action Selection is to find the least-cost investigation actions within a budget constraint to verify Q_N (unobserved techniques) of the hypothesis that has the highest fidelity. As the size of Q_N grows large, the process of selecting the minimal cost investigation action that verifies Q_N becomes non-trivial. Since fidelity evaluation module outputs unobserved techniques (Q_N) rather than unobserved observables and IAS module uses unobserved observables, both the Technique-observables and Investigation-Action-observables correlation graph will be used in optimal investigation action selection process.

The Technique-observables correlation graph can be described as a 3-tuple (Q, O, E_1) graph where Q and O are two independent vertex sets representing attack techniques and observables respectively, and every edge $e_k \in E_1$ connects a vertex $q_i \in Q$ with a vertex $o_j \in O$. The Investigation-Action-observables correlation graph can be represented as a 3-tuple (A, O, E_2) graph such that A and O are two independent vertex sets representing Investigative Actions and observables respectively, and every edge $e_k \in E_2$ connects a vertex $a_j \in A$ with a vertex $o_i \in O$ with a corresponding cost (t_{ij}) to denote that a_j can verify o_i with cost $w_{ij} = w(s_i, a_j) > 0$. If there is no association between o_i and a_j, then $w_{ij} = 0$. Moreover, it might be the case that a set of investigation actions are required to verify an observables, we represent such case with a composite investigation action vertex, v_j. Thus, the composite investigation action vertex v_j associates a set of conjunctive investigation actions to the corresponding observable. However, if multiple investigation actions are directly connected to an observable, any of these investigative actions can be used disjunctively to verify this observable. The relationship between the composite investigation action vertex, v_j and the corresponding observable, o_i is represented using a bipartite graph as follows- we (1) set the cost of verifying o_i with composite investigation action to $w(o_i, v_j) = w_{i1} + w_{i2} + ... + w_{ik}$ if composite investigation action set, $A_c = \{a_1, a_2, ..., a_k\}$ which equals to the total cost of the conjunctive action set; (2) then create virtual vertex v_j that corresponds to the conjunctive investigation action set; and (3) associate v_j with all observables that can be verified by any combination of investigation action in the conjunctive investigation action set.

The goal of the Investigative Action Selection algorithm is to select the actions that cover all Observables O_N with a minimal action cost within a budget constraint. With the representation of the Observable-Action bipartite graph, we can model this problem as a weighted set-covering problem. Thus, the Investigative Action Selection algorithm searches for A_i such that A_i includes the set of actions that cover all of the observables in the Observable-Action correlation graph with total cost within a budget constriant. We can formally define A_i as the covering set that satisfies the following conditions: (1) $\forall o_i \in O$, $\exists a_j \in A_i$ such that $t_{ij} > 0$, and (2) $\sum_{a_i \in A_i, s_j \in S_N} t_{ij}$ is the minimum and less than budget constraint.

The weighted set-covering is an NP-complete problem [17]. Thus, we developed a heuristic greedy set-covering approximation algorithm to solve this problem. The main idea of the Algorithm is simply to select the investigative action (a_i or v_i) that has the maximum *relative covering ratio*, $R_i = \frac{|O_{a_i}|}{\sum_{o_j \in O_{a_i}} t_{ij}}$ first, where this investigative action is added to the final set A_f and removed from the candidate set A_c that includes all actions. Here, O_{a_i} is the set of observables that action a_i can verify, $O_{a_i} \subseteq O_N$. Then, we remove all observables that are covered by this selected action from the not seen observable set O_N. This search continues to find the next action a_i ($a_i \in A_c$) that has the maximum ratio R_i until all observables are covered (i.e., O_N is empty). Thus, intuitively, this algorithm appreciates actions that have more observable correlation or aggregation. If multiple actions have the same relative covering ratio, the action with more covered observables (i.e., larger $|O_{a_i}|$ size) will be selected. If multiple actions have the same ratio, R_i, and same $|O_{a_i}|$, then each action is considered independently to compute the final selected sets for each action and the set that has the minimum cost is selected. Finally, it is important to notice that each single action in the A_f set is necessary for the threat diagnosis process, since each one covers unique observables.

E. Algorithm for Active Investigation Threat Reasoning

The main contribution of the *AITR* framework is incorporation of active investigation actions with passive threat reasoning by determining the appropriate trade-off between them. Passive threat reasoning could work well if enough observables can be observed correctly. However, it is infeasible to monitor everything, and in most cases, we need to deal with deletion of observables by attacker for persistence, insertion of spurious evidences to misdirect the threat diagnosis and false positive rate of technique detectors. During the passive threat reasoning, the generated hypothesis suggests a set of techniques Q_N that are unobserved but expected to happen based on the highest fidelity hypothesis. If fidelity evaluation of generated hypothesis is not acceptable, optimal investigation actions are selected to verify Q_N (Line 9). Investigative Action execution will either increase fidelity evaluation of the previous hypothesis or bring new evidence to generate a new hypothesis. By performing the optimal investigation action, the IAS module and fidelity evalution module will iteratively evaluate the hypothesis and suggest the hypothesis with FD_{score} greater than FD_{TH} as the final goal the attacker is trying to achieve. Since the accepted hypothesis also advertises the TTP used by the attacker to reach the final predicted goal, the system also predicts which are the next provable attack techniques the adversary is going to execute.

Algorithm 2 illustrates the complete process of the *AUTO-TTPHunt* framework. Initially, the system takes observed technique Q_O as input (Line 1). Threat Reasoning module is used to search the best hypothesis Φ (Line 3). Fidelity is the key to associate passive reasoning to active investigation actions. Fidelity Evaluation is used to measure the correctness of corresponding hypothesis h ($h \in \Phi$) and produce expected unobserved techniques Q_N (Line 4). If the Fidelity score of hypothesis h is satisfied, the process terminates with the current hypothesis as output (Line 5 - 6). If fidelity score is not acceptable, the new verified evidence from previous observation is used to search for another hypothesis (Line 3) until the fidelity score is satisfied. At any point, if either the fidelity evaluation does not find unobserved techniques

Algorithm 2 Active Investigation Threat Reasoning Q_O

```
Input: $Q_O$
Output: threat hypothesis $h$
 1: $Q_N \leftarrow Q_O$
 2: while $Q_N \neq \emptyset$ do
 3:     $\Phi = HU(null,\ Q_O,\ G_C)$
 4:     $< h, Q_N, S > = FD(\Phi)$
 5:     if $Q_N = \emptyset$ or $FD_{TH} <= S$ then
 6:         return $< h >$
 7:     else
 8:         /*schedule active threat diagnosis*/
 9:         $< Q_V, Q_U >= AIAS(Q_N)$
10:     end if
11:     $Q_O \leftarrow Q_O \cup Q_V$
12:     $< h, Q_N, S > = FD(\{h\})$
13:     if $Q_N = \emptyset \parallel Q_V = \emptyset \parallel FD_{TH} <= S$ then
14:         return $< h >$
15:     end if
16: end while
```

to verify (Q_N is $\emptyset$), or none of the verified techniques had occurred (Q_V is $\emptyset$), the program will terminate and return the current selected hypothesis (Line 13-14). In either case, this is an indication that the current selected hypothesis is credible.

VII. Data Sets

To build the correlation matrix, we used attack instance provided by the mitre ATT@CK framework. Later, a tool is developed to implement the proposed framework. Network end device logs will be collected using different monitoring tools and used as input to the ThreatHunter framework

VIII. Active Investigation Reasoning framework Evaluation

IX. Conclusion

Since the diagnosis of attack to detect and prevent is quintessential, our proposed framework provides an efficient way to detect and defend against attack. In future, we will implement the active investigation action selection algorithm which will be used if currently generated hypothesis does not satisfy the set threshold.

References

[1] "Symmantec. attack listing." https://www.symantec.com/security-center/a-z.

[2] L. D. Ping Chen and C. Huygens, "A study on advanced persistent threats," in *In IFIP International Conference on Communications and Multimedia Security*, 2014.

[3] C. G. Z. L. C. P. S. Stevens Le Blond, Adina Uritesc and E. Kirda, "A look at targeted attacks through the lense of an ngo," in *In USENIX Security Symposium*, 2014.

[4] C. K. Marco Cova and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code," in *In International Conference on World Wide Web (WWW)*, 2010.

[5] P. P. H. D. H. Y. S. L. B. D. M. Brown Farinholt, Mohammad Rezaeirad and K. Levchenko, "To catch a ratter: Monitoring the behavior of amateur darkcomet rat operators in the wild," in *In IEEE Symposium on Security and Privacy*, 2017.

[6] L. C. B. G. M. S. R. K. C. K. Brett Stone-Gross, Marco Cova and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeove," in *In ACM Conference on Computer and Communications Security(CCS)*, 2009.

[7] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," in *Expert Syst. Appl*, 2009.

[8] C. G. H. R. S. A. Peng Ning, Dingbang Xu, "Building attack scenarios through integration of complementary alert correlation methods," in *Expert Syst. Appl*, 2004.

[9] V. Y. M. W. F. Guofei Gu, Phillip A Porras and W. Lee, "Detecting malware infection through ids-driven dialog correlation," in *In USENIX Security Symposium*, 2007.

[10] M. J. V. P. Grant Ho, Aashish Sharma and D. Wagner, "Detecting credential spearphishing in enterprise settings," in *In USENIX Security Symposium*, 2017.

[11] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks.," in *In ACM Conference on Computer and Communications Security (CCS)*, 2003.

[12] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, "Tiresias: Predicting security events through deep learning," in *In ACM Conference on Computer and Communications Security (CCS)*, 2018.

[13] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *In USENIX Security Symposium. 625640*, 2014.

[14] "Sysmon." https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon.

[15] I. E. A.-S. M. I. Yongning Tang, Member and R. Boutaba, "Efficient fault diagnosis using incremental alarm correlation and active investigation for internet and overlay networks," in *IEEE Transactions on Network and Service Management , Volume: 5, Number: 1, Pages: 36–49*, 2008.

[16] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC 2017, (New York, NY, USA), p. 103115, Association for Computing Machinery, 2017.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, second edition ed., 2001.