Strategic Monitoring for Efficient Detection of Simultaneous APT Attacks with Limited Resources

Fan Shen, Zhiyuan Liu, Levi Perigo
Department of Computer Science, University of Colorado Boulder
Boulder, Colorado 80309

Abstract—**Advanced Persistent Threats (APT) are a type of sophisticated multistage cyber attack, and the defense against APT is challenging. Existing studies apply signature-based or behavior-based methods to analyze monitoring data to detect APT, but little research has been dedicated to the important problem of addressing APT detection with limited resources. In order to maintain the primary functionality of a system, the resources allocated for security purposes, for example logging and examining the behavior of a system, are usually constrained. Therefore, when facing multiple simultaneous powerful cyber attacks like APT, the allocation of limited security resources becomes critical. The research in this paper focuses on the threat model where multiple simultaneous APT attacks exist in the defender's system, but the defender does not have sufficient monitoring resources to check every running process. To capture the footprint of multistage activities including APT attacks and benign activities, this work leverages the provenance graph which is constructed based on dependencies of processes. Furthermore, this work studies the monitoring strategy to efficiently detect APT attacks from incomplete information of paths on the provenance graph, by considering both the "exploitation" effect and the "exploration" effect. The contributions of this work are two-fold. First, it extends the classic UCB algorithm in the domain of the multi-armed bandit problem to solve cyber security problems, and proposes to use the malevolence value of a path, which is generated by a novel LSTM neural network as the exploitation term. Second, the consideration of "exploration" is innovative in the detection of APT attacks with limited monitoring resources. The experimental results show that the use of the LSTM neural network is beneficial to enforce the exploitation effect as it satisfies the same property as the exploitation term in the classic UCB algorithm and that by using the proposed monitoring strategy, multiple simultaneous APT attacks are detected more efficiently than using the random strategy and the greedy strategy, regarding the time needed to detect same number of APT attacks.**

Keywords—Advanced persistent threats; intrusion detection; LSTM; multi-armed bandit

I. Introduction

Multiple malicious activities can happen simultaneously on a host or system, especially when it performs mission-critical tasks [1]. If the monitoring capacity (also referred to as monitoring resources throughout this paper) is limited, without thoughtful allocation of these resources, it is possible that some malicious activities will not be captured and identified. A generic way of allocating limited resources is to assign the monitoring resources to the most important or suspicious objects, which is also called a greedy strategy. One dilemma of the greedy strategy, however, is that, if the early perceptions about objects are not reliable, benign objects might consume more monitoring resources than malicious objects. Unfortunately, unreliable early detection of Advanced Persistent Threats is common, because such advanced attacks are stealthy and it is possible that an APT attack in its earlier stages is less suspicious than a benign activity. When using a greedy strategy to allocate resources, the Matthew Effect can cause some of the attacks to be undetected, because less monitoring resources are assigned to them. Therefore, an enhanced method to allocate monitoring resources, compared with the greedy strategy, is needed for the detection of simultaneous long-term attacks with limited resources, because this threat model has not been extensively addressed by existing works in the area of anomaly detection.

The research in this paper focuses on the detection of a sophisticated cyber attack, APT[2], and proposes a strategy to allocate limited security resources for monitoring, in order to efficiently detect APT activities when multiple APT activities are ongoing concurrently in a system.

II. Related Work

APT has attracted the attention of industry and academia since the 2010s, when several unseen yet powerful APT malware were discovered including Stuxnet, Duqu, Flame and Gauss. Although it is a multistage, complicated attack, the typical stages and behavioral patterns of APT are extracted by existing studies. For example, attack chain models based on Cyber Kill Chain [3] and the attack pyramid model [4] are proposed to characterize multiple APT stages and their relations. MITRE ATT&CK [5] constantly publish common tactics, techniques, and procedures (TTP) of APT attacks. Then according to the characteristics of APT, researchers use different methods to effectively detect it from illegitimate system access, suspicious network traffic patterns, and abnormal system resource utilization. Based on the categories of methods, APT detection studies are mainly divided into two categories: signature-based and behavior-based.

Signature-based detection methods match system behaviors with known attack patterns, once a match is found, pre-configured actions will be taken. Snort [6] is an example of signature-based intrusion detection system, which generates explicit rules from known attacks, if any rule is matched, it will trigger actions such as SNMP traps, event logging, and allow/deny traffic. The author [7] leverages the Intrusion Kill Chain [8] model defining rules to identify each APT stage by its attack mechanisms from multiple sources of logs and build the kill chain by comparing the timestamps of found APT stages. HOLMES [9] defines explicit rules to map a low-level system event to an abstracted APT TTP of MITRE ATT&CK and use the rules to identify each APT stage. As Command and Control (C2) communication is common in an APT campaign, network packet inspection or network flow analysis is used in some works to identify the C2 communication. The authors of [10] found consistent patterns in the network traffic between the APT malware of interest and the C2 server, and used protocol-aware rules, special strings in a URL, and malformed images to effectively detect C2 communication. Signature-based detection methods are effective to attacks whose characteristics are known and can be well-represented by a set of matching rules, and when compared with behavior-based methods, signature-based methods are less complicated and easier to implement. However, the performance of signature-based methods can degrade quickly when dealing with variants of known attacks and novel or zero-day attacks.

Behavior-based detection methods do not rely on known patterns, and they profile behaviors, either benign or malicious, using statistical or machine learning methods [11]. The advantage of such detection methods is being able to accommodate variations and uncertainties of behaviors and characterize behaviors that can not be represented by explicit matching rules. To identify suspicious hosts that might be involved in C&C communications, the authors in [12] characterized each host periodically using features extracted from network flows, then assigned a risk score to each host based on the deviation from its historical positions, distances from other hosts, and magnitude of increment in the feature space. The authors [13][14] focus on detection of spear phishing emails that are used for initial penetration in APT. The authors of [13] extract static and dynamic features of PDF files attached in emails, and propose a classification model using Support Vector Machine, to detect malicious PDF files. Active learning is also integrated in their model to cope with unseen PDF files. The researchers in [14] used Naïve Bayes theorem to detect spam emails containing links that redirect the victim to malicious websites, which could help APT attacker establish backdoors inside the victim's system. The study [15] proposes an ensemble RNN-based model to detect different APT steps by analyzing network traffic data. Host-level system logs are analyzed in [16] to identify different APT phases. The authors of [16] first translated system log sequences into abstracted states by using the hidden Markov model, then fed high-level state sequences to three multi-classification models: LSTM, one-dimensional CNN, and SVM, to predict the APT phase. Since the detection of a single APT stage does not necessarily indicate a multistage APT activity, some works [4][17][18][19] detect APT by correlating detection results of different APT stages based on models that describe dependencies between stages.

In existing studies on APT detection, the resource allocation problem is rarely addressed; whereas, it is a popular topic in game-theoretic studies [20][21] on APT prevention. Resource constraints, however, do exist in APT detection, either only a portion of CPU and memory can be used for monitoring activities in a system, or only limited insights can be obtained from enormous monitoring data in a timely manner. In a worse-case scenario, if multiple attacks concurrently exist in a system, and monitoring resources are not enough to cover all attacks, it is critical that the system operator allocate limited resources efficiently so that the most attacks possible are detected. Therefore, this work focuses on multistage APT attacks, and proposes a novel strategy that allocates monitoring resources, not only based on the current malevolence of activities, but also introducing an exploration mechanism to eliminate the side effect of a greedy strategy when the malevolence calculation based on early stage information of APT is not reliable.

III. Threat Model

In this work, simultaneous and continuous APT attack activities exist in the defender's system. For each multistage APT attack, each of its stages can be detected if relevant behavior is monitored. It is assumed that a whole-system provenance graph is used to obtain all paths including non-APT paths and APT paths. However, to reflect a real-world scenario, the monitoring capability of the defender's system is limited, so that only the activity on some, but not all, paths can be monitored and analyzed at each timestamp. Every monitoring timestamp provides a classification result indicating that the monitored activity is benign or one of APT stages. Therefore, for each path on the provenance graph, the defender has a sequence of temporal but incomplete detection results about the activities on the path. Thus, the goal of the defender is to efficiently detect as many simultaneous APT attacks as possible, while utilizing limited resources.

IV. Problem Setting

To capture the footprint of system activities, this work leverages the provenance graph which is constructed based on the dependencies of processes. It is assumed that the provenance graph is complete and not compromised in this paper. Therefore, the movement of a multistage APT attack is represented by one of the paths on the provenance graph, and in the threat model of this paper, the defender needs to detect those attack paths efficiently with limited resources. In addition, to decide the identity of a path, benign or APT, monitoring and investigation is used to check the state of nodes on a path. At each timestamp, however, not all running nodes can be checked because the monitoring resources are limited. Therefore, the research question addressed in this paper is, how to allocate limited monitoring resources, (i.e. select which running nodes should be evaluated at each timestamp), so APT attacks are detected effectively in terms of the number of detected attacks, and efficiently in terms of time needed? And a strategic monitoring model which considers both the estimated malevolence of system activities and the uncertainty of that estimation is proposed to solve the problem.

The problem setting of this paper is described as follows. There are n processes running on a host, which correspond to n paths on the provenance graph, but at each timestamp, only k $(k < n)$ processes can be monitored and investigated to get the current state of the corresponding path. In this paper, when a process is chosen to be monitored, its local structure is fed into the detection engine developed by [11] which outputs a classification result. Mathematically, for each path i $(i \in [1, n])$, at time j, if the process corresponding to path i is monitored, the current state of path s_{ij} is obtained from the detection engine and $s_{ij} = \{0, 1, 2, 3, 4, 5, 6\}$; if the process corresponding to path i is not monitored, the current state of path s_{ij} is unknown and is represented by $s_{ij} = -1$. In this paper, the defender decides which k processes are selected to be monitored at each timestamp, based on historical and temporal state information of each path.

V. Strategic Monitoring Model

In this paper, a scoring mechanism is proposed to decide which k processes to monitor at each timestamp, in other words, to solve a sequential decision making problem to identify all simultaneous APT attacks as early as possible. The proposed scoring rule is derived from a classic multi-armed bandit algorithm Upper Confidence Bound (UCB) [22]. The purpose of a multi-armed bandit problem is to maximize the cumulative reward by sequentially selecting arms to pull, by assuming the existence of uncertainties in the reward of each arm [23]. The UCB algorithm integrate both the exploitation effect (observed mean reward of each arm so far) and the exploration effect (number of times each arm has been pulled) when prioritizing arms. The index of arm i at time t is calculated as follows:

$$I_i(t) = \bar{\theta}_i(t) + \sqrt{2\frac{\log t}{\tau_i(t)}} \tag{1}$$

where $\bar{\theta}_i(t)$ is the sample mean reward of arm i, which is the exploitation factor meaning that arms with higher historical reward are prioritized; $\tau_i(t)$ is the number of times arm i has been pulled, which is the exploration factor meaning that the less pulled arms are prioritized.

It is appropriate to apply UCB to solve the monitoring strategy in the threat model of this paper, because the detection of multistage APT relies on sequential decisions of monitoring to obtain temporal state of each system behavior path. Therefore, the proposed scoring mechanism is a variation of the UCB algorithm, replacing the sample mean reward with the current malevolence value of a path. One novelty of this paper is that, it extends the UCB algorithm to the multistage APT detection scenario by proposing to use a long short-term memory (LSTM) based malevolence value as the exploitation driver, and showing that this modification to the UCB algorithm is reasonable. The proposed scoring rule is formulated as follows:

$$I_i(t) = \alpha \cdot f_i([s_{i1}, s_{i2}, ..., s_{it}]) + (1-\alpha) \cdot \sqrt{2\frac{\log t}{\tau_i(t)}} \tag{2}$$

where $I_i(t)$ is the index or score of path i at time t which is a weighted sum of an exploitation term and an exploration term; $f(\cdot)$ is a neural network that takes the temporal state information of path i as input and outputs the current malevolence value the path i; the exploration term is the same as in Equation 1; and α is a constant in $[0, 1]$.

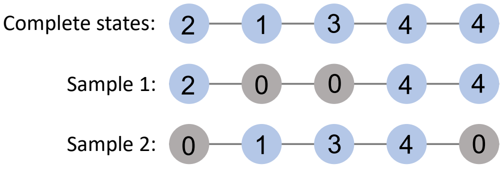


Fig. 1. Different samples generated from a sequence of complete states of a system behavior path.

A LSTM neural network is used to assign a malevolence value to a system behavior path based on its temporal states information. Samples of paths with various temporal states are used to train the neural network. In this paper, these samples are generated from complete state sequences of both benign behavior paths and APT behavior paths, by randomly selecting some states and making them unknown. For example, as shown in Fig. 1, the top row indicates that the state of an "APT attack" path at every timestamp is known, thus the sequence of states is complete (1 represents the state is benign, 2-4 represent different APT stages). To generate samples e.g. "Sample 1" and "Sample 2" resulted from incomplete monitoring in a scenario where resource constraints exist, some timestamps of the complete sequence are masked with value 0 representing that the state information at those timestamps become unknown. Note that, the class of "Sample 1" and "Sample 2" is still "APT attack". By choosing different values for the number of timestamps that are masked, samples with different incompleteness are generated, and together with their classes they are used to train the LSTM model which outputs how likely a sequence of states is an APT attack. There are two classes in the data: "1" for APT attack behavior paths; "0" for benign behavior paths. The LSTM neural network is trained to output a value in $[0, 1]$. A higher LSTM output value indicates a more suspicious behavior path, and this value is used as the exploitation factor in Equation 2: the more suspicious the behavior path is, the more likely it would be monitored next time. The second term in Equation 2 is the exploration factor, meaning that the less frequently the behavior path was monitored, the more likely it would be monitored next time.

VI. Experiments

To evaluate the performance of the proposed monitoring strategy developed in this research, synthetic behavior paths with temporal states for both benign scenario and APT attack scenario are generated and implemented. Two principles are followed when generating the synthetic data: (1) all states at a timestamp ({1: "benign"; 2: "APT stage 1"; 3: "APT stage 2"; 4: "APT stage 3"; 5: "APT stage 4"; 6: "APT stage 5"; 7: "APT stage 6"}) can appear on a behavior path of benign scenario and APT attack scenario; (2) the temporal order of states is differentiated on a behavior path of benign scenario and on a behavior path of APT attack scenario. More specifically, the states corresponding to APT stages on the behavior path of a benign scenario are uncorrelated; however, the states corresponding to APT stages on the behavior path of an APT attack scenario are correlated in the sense that, without interruption an APT attacker gradually moves from lower APT stages to higher stages because the attacker does not have incentive to move from higher APT stages to lower stages. In addition, a random number of benign states appear between APT related states on behavior paths of both benign scenarios and APT attack scenarios. By following these principles, 100 attack paths and 100 benign paths with 80 complete temporal states are generated. Then, each complete path generates 79 incomplete paths by randomly hiding i ($i \in [1, 79]$) states. Eventually, 8000 APT attack paths and 8000 benign paths with various degrees of incompleteness are generated and used to train and test the LSTM model in this paper.

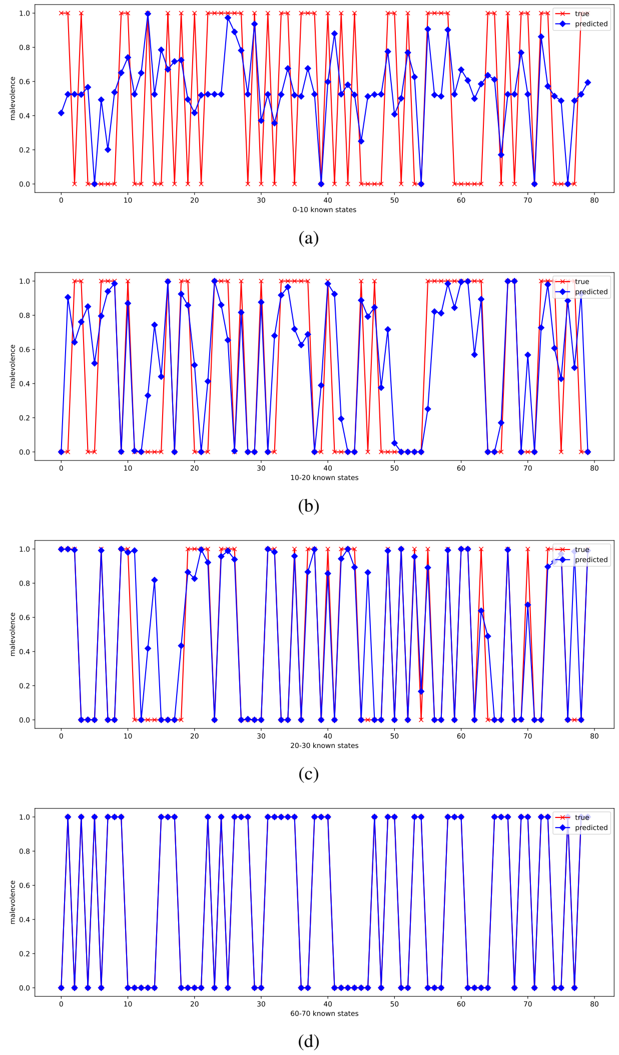
The first part of the experiment is to demonstrate that using the output of a novel LSTM neural network as the exploitation term in the proposed model is effective, in other words, the trained neural network should satisfy the same property as the exploitation term defined in the classic UCB algorithm. In the classic UCB algorithm, the property of the exploitation term is that, as an arm is pulled more often, the estimation to its reward becomes more accurate. Therefore, the proposed LSTM neural network is tested on behavior paths of which the defender has different degrees of information incompleteness. If the LSTM neural network is effective, it is desired that its estimation to the malevolence of paths is more accurate as the defender's information incompleteness of paths decreases.

The second part of the experiment is to demonstrate the performance of the proposed monitoring strategy. Before evaluation, 20 behavior paths are randomly selected from the data set, including four APT attack paths and 16 benign paths. Then, the the proposed monitoring strategy, a random strategy and a greedy strategy are evaluated respectively, in terms of when the attack paths are detected and the number of false positives. In the experiment setting, only five paths can be monitored at each timestamp, a random strategy means that the five monitored paths are randomly selected; a greedy strategy means that the paths with five highest malevolence scores are selected, in other words, only the exploitation effect in the proposed strategy is considered; the proposed strategy selects the 5 paths with the highest scores where the score is defined as the weighed sum of the exploitation factor and the exploration factor are selected. In addition, two termination conditions are applied when implementing the proposed strategy: (1) when the malevolence score of a path is greater than β, the path is determined as an APT attack and will no longer be a candidate of being monitored; (2) when the malevolence score of a path is smaller than γ, the path is determined as a benign scenario and will no longer be a candidate to be monitored. By testing different sets of parameter values, the best parameter values of the proposed strategy are used to compare the proposed strategy with other strategies, including the weight parameter α in Equation 2 ($\alpha = 0.82$), and two threshold parameters β ($\beta = 0.9$) and γ ($\gamma = 0.05$).

VII. Results and Analysis

Fig. 2 shows the performance of the LSTM neural network in the proposed strategic monitoring model. From Fig. 2(a) to Fig. 2(d), the number of known states of paths increases, in other words, the defender's information incompleteness of paths decreases. The red line represents the true malevolence of paths, and the blue line represents the predicted malevolence of paths by the LSTM neural network. It can be seen that the difference between the true malevolence values and predicted malevolence values by the LSTM neural network becomes smaller, when the number of known states increases. Therefore, the estimation of the LSTM neural network to the malevolence of a behavior path becomes more accurate along the path that is monitored more frequently, making the output of the LSTM neural network an effective exploitation term in the proposed monitoring strategy model.

To demonstrate the performance of the proposed monitoring strategy, it is compared with a random strategy and a greedy strategy, and the results of the three strategies including the paths monitored at each timestamp as well as when attacks are detected are visualized in Fig. 3, Fig. 4, Fig. 5 respectively. Note that, in Fig. 3 to 5, each row i represents a behavior path and the rows with light red shade means that the row represents an APT path. Each column j represents a timestamp, and at each time j, the defender can pick $\max(5, u_j)$ paths to monitor where u_j is the number of undetermined paths at time j. For a position (i, j) where $i \in [1, 20]$ and $j \in [1, 80]$, if it is in blue, it means path i is monitored at timestamp j; if it is in red, it means path i is classified as an APT attack scenario at timestamp j and it will no longer be monitored which is represented by marking its future states as grey; if it is in green, it means path i is classified as a benign scenario at timestamp j and it will no longer be monitored which is represented by marking its future states as grey. Therefore, the number of undetermined paths u_j is the number of paths that are not in grey at time j.



(a)

(b)

(c)

(d)

Fig. 2. Comparison of true and predicted malevolence of paths with various number of known states.

Fig. 3 shows the results of using a random strategy, which means that the defender randomly picks paths to monitor at each timestamp. The four APT attacks paths are detected at timestamp 57, 50, 49 and 41. In addition, the number of false positives is 2.

Fig. 4 shows the result of using a greedy strategy, which

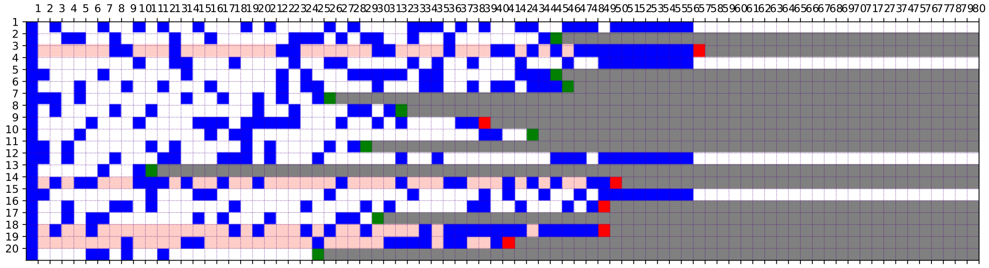


Fig. 3. Performance of the random monitoring strategy.

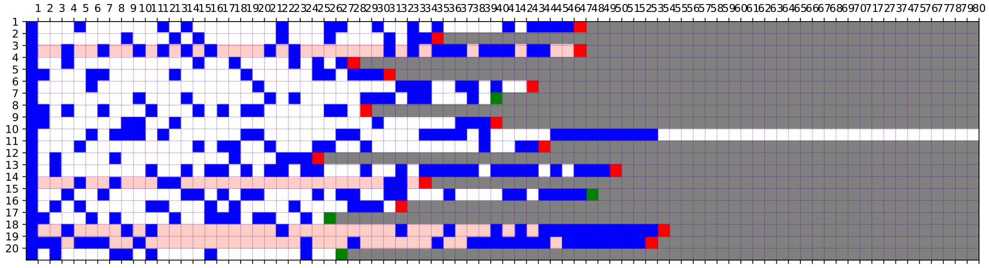


Fig. 4. Performance of the greedy monitoring strategy.

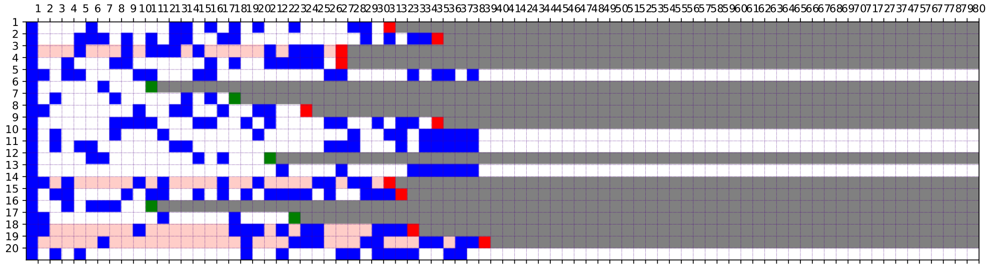


Fig. 5. Performance of the proposed monitoring strategy.

means that the defender picks paths with the highest malevolence values predicted by the LSTM neural network. The 4 APT attacks paths are detected at timestamp 47, 34, 54 and 53. In addition, the number of false positives is 11.

Fig. 5 shows the result of using the strategy developed by this paper, which means the defender considers both exploitation and exploration then picks paths with the highest values calculated by Equation 2. The 4 APT attacks paths are detected at timestamp 27, 31, 33 and 39. In addition, the number of false positives is 6.

Compared with the other two strategies, the monitoring strategy proposed in this paper detects all 4 APT attacks significantly earlier, specifically nearly 17 timestamps earlier than the random strategy, and 15 timestamps earlier than the greedy strategy. The key to the efficiency of the proposed strategy is that by considering exploration when the malevolence estimation is not as accurate in early timestamps, the defender is able to identify some benign paths quickly, which reduces the number of competitors of limited monitoring resources. However, the random strategy treats benign paths and attacks paths equally and the greedy strategy can be misled by inaccurate malevolence in earlier stamps, thus they are less efficient in the detection of simultaneous APT attacks.

When facing advanced attacks like APT, false positive is more acceptable compared to false negative, because missing an APT attack is more devastating than looking into a benign activity which is falsely classified as attack. Regarding the number of false positives in the experiments, the proposed strategy is better than the greedy strategy, but is worse than the random strategy. This is as expected, because from Fig. 3, it takes longer and relies on more information for the defender to determine the identity of a path when using the random strategy. With more information, the malevolence estimation to a path is more accurate at shown in Fig. 2, however, the random strategy is the least efficient regarding the time needed to detect simultaneous APT attacks. Therefore, overall the proposed strategy outperforms the other two strategies regarding the metric of efficient detection of simultaneous APT attacks with limited resources. And the improvement on other metrics is left as a future extension.

VIII. Conclusion

The work in this paper addresses the issue of resource constraints in the detection of multiple simultaneous APT attacks. It proposes a monitoring strategy to efficiently detect APT attacks with incomplete information about activities in a system. The key of the proposed strategy is that it considers both the "exploitation" effect and the "exploration" effect in resource allocation, which is beneficial for finding the optimal strategy in circumstances with high uncertainties. The novel contributions of this work to address the research question are as follows.

First, differing from existing works that allocate security resources based on the estimated malevolence of system activities only, this work emphasizes the importance of "exploration" in APT detection, because the perception to advanced and stealthy attacks based on its earlier stage information is usually not accurate. This work is the seminal paper to consider both the "exploitation" effect and the "exploration" effect in monitoring resource allocation, and apply the classic multi-armed bandit algorithm, UCB, to solve optimal resource allocation problems in APT defense.

Second, this work proposes a novel LSTM neural network to measure the malevolence of a path on the provenance graph based on its incomplete temporal information, and replaces the exploitation term in the classic UCB algorithm with this malevolence value. The experimental results show that by using the proposed monitoring strategy, multiple simultaneous APT attacks are detected more efficiently than using a random strategy and a greedy strategy, regarding the time needed to detect same number of attacks.

Although the proposed model shows the advantage of detecting simultaneous APT attacks efficiently with limited resources, a future extension to this work is to enhance the model in terms of more metrics, for example, reducing false positives by exploring more features of APT to differentiate it from benign activities more effectively.

References

[1] H. Zhang *et al*, "Efficient strategy selection for moving target defense under multiple attacks," *IEEE Access*, vol. 7, pp. 65982-65995, 2019.

[2] A. Alshamrani *et al.*, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851-1877, 2019.

[3] Lockheed Martin, "The cyber kill chain," Available at https://www.lockheedmartin.com/us/what-we-do/aerospace-defense/cyber/cyber-kill-chain.html, Accessed Dec 2022.

[4] P. Giura and W. Wang, "A context-based detection framework for advanced persistent threats," In *Proc. 2012 International Conference on Cyber Security*, pp. 69-74, 2012.

[5] MITRE ATT&CK, Available at https://attack.mitre.org, Accessed Dec 2022.

[6] M. Roesch, "Snort: Lightweight intrusion detection for networks," *Lisa*, vol. 99, no. 1, pp. 229-238, 1999.

[7] P. Bhatt *et al.*, "Towards a framework to detect multi-stage advanced persistent threats attacks," In *IEEE 8th international symposium on service oriented system engineering*, pp. 390-395, 2014.

[8] 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, Apr. 2011.

[9] M. M. Sadegh *et al.*, "Holmes: real-time apt detection through correlation of suspicious information flows," In *Proc. 2019 IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, May 2019, pp. 1137-1152.

[10] N. Villeneuve and J. Bennett, "Detecting APT activity with network traffic analysis," *Trend Micro Incorporated Research Paper*, pp. 1-13, 2012.

[11] F. Shen, L. Perigo and J. Curry, "SR2APT: A Detection and Strategic Alert Response Model Against Multistage APT Attacks," *Security and Communication Networks* (forthcoming), DOI:10.1155/1969/6802359.

[12] M. Marchetti *et al.*, "Analysis of high volumes of network traffic for advanced persistent threat detection," *Computer Networks*, vol. 109, pp. 127-141, Nov. 2016.

[13] N. Nissim, "Detection of malicious PDF files and directions for enhancements: A state-of-the art survey," *Computers & Security*, vol. 48, pp. 246-266, Feb. 2015.

[14] J. V. Chandra, N. Challa and S. K. Pasupuleti, "A practical approach to E-mail spam filters to protect data from advanced persistent threat," In *Proc. 2016 international conference on circuit, power and computing technologies (ICCPCT)*, pp. 1-5, 2016.

[15] H. N. Eke *et al*, "Framework for Detecting APTs Based on Steps Analysis and Correlation," *Security and Resilience in Cyber-Physical Systems*, pp. 119-147, 2022.

[16] M. AbuOdeh *et al.*, "A novel AI-based methodology for identifying cyber attacks in honey pots," In *Proc. AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 15224-15231, May 2021.

[17] J. Sexton, C. Storlie and J. Neil, "Attack chain detection," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 8, no. 5-6, pp. 353-363, Oct. 2015.

[18] G. Brogi and V. V. T. Tong, "Terminaptor: Highlighting advanced persistent threats through information flow tracking," In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-5, 2016.

[19] I. Ghafir *et al.*, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, pp. 349-359, Dec. 2018.

[20] M. V. Dijk *et al.*, "FlipIt: The game of stealthy takeover," *Journal of Cryptology*, vol. 26, no. 4, pp. 655-713, 2013.

[21] M. Zhang *et al.*, "A game theoretic model for defending against stealthy attacks with limited resources," In *International Conference on Decision and Game Theory for Security*, pp. 93-112, 2015.

[22] P. Auer, N. Cesa-Bianchi and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235-256, 2002.

[23] Z. Liu *et al.*, "Incentivized exploration for multi-armed bandits under reward drift," In *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4981-4988, 2020.