Online Detection and Control of Malware Infected Assets

Hasan Cam[1]

***Abstract*—Malware infection activities need to be detected as early as possible to minimize the adverse impact of malware delivery, infection, exploitation, and spreading. Given that cybersecurity observations over a network are usually incomplete, noisy, uncertain, and need to be extracted from a big data size, it is a challenge to extract quality information for identifying vulnerable, infected, or exploited assets and then taking appropriate actions to mitigate the impact of malware infection and spread. This paper presents an integrated model of logistic regression and Partially Observable Markov Decision Process (POMDP) along with online data analytics on temporal causality and dependency relationships of observations. New regression and malware infection features are developed by capturing and formulating the cross relationships of observations. Logistic regression of these new features is used to estimate the initial probability values that sensor measurements are indicative of vulnerability exploitations, which help infer the infection status of those assets associated with the vulnerabilities to be likely exploited. The results of the logistic regression on the infection status of assets are considered as the initial belief state of POMDP. The integrated model of logistic regression and POMDP is designed to iteratively collaborate in identifying and controlling malware infection and spread. Experimental results show the efficiency of having such collaboration in identifying and controlling malware-infected assets.**

I. INTRODUCTION

A cyber-resilient system needs to adaptively resist against attacks, minimize the adverse impact of vulnerability exploitations, and recover any compromised asset of the system. Such a resilient system requires not only detection and assessment of its vulnerabilities, attacks, infections, and exploitations, but also determining those actions needed to keep the system resilient against malware infection and spread. In today's cyber defense environment, it is reasonable to state that a high-value target is either already infected or likely will be infected in the near future, especially if attacker is persistent. Therefore, it is highly desirable to have adaptive and active cyber defense. Here, active cyber defense refers to a "synchronized, real-time capability to discover, detect, analyze, and mitigate threats and vulnerabilities" [1]. To build such a cyber defense against malware, this paper proposes using a combination of online data analytics of all observations, identifying malware-infected assets, and mitigating the adverse impact of malware infection and spreading using an integrated model of Partially Observable Markov Decision Process (POMDP) [2] and logistic regression along with learning decision trees, semantics-based malware analysis, data analytics on temporal causality, and dependency relationships of observations.

The disadvantage of having noisy and incomplete cyber observations in dealing with malware infection during all phases of cyber kill chain can be mitigated by establishing temporal causality analysis and feature extraction of observations in near real-time. It is also highly desirable to take adaptive actions to recover infected assets, patching vulnerabilities, and controlling malware spread upon detection of malware-infected assets. Therefore, this paper's approach is based on three observations: (*i*) the logistic regression model [3,4] can be used to estimate the initial exploit likelihood value of vulnerabilities based on the indicators of all cyber data including vulnerability, intrusion, traffic, log data, etc.; (*ii*) POMDP fits well for identifying malware-infected assets and taking appropriate actions to mitigate its adverse impact and malware spread within a cybersecurity system whose system is not known exactly, and therefore actions to be taken for malware control are not obvious; and (*iii*) to assist in making the process of identifying malware-infected assets faster, more accurate, tractable, more beneficial, and more efficient, it is essential to consider online, context-specific processing on those assets that are most likely targets predicted by current observations.

Logistic regression helps modeling based on past and present states by dynamically determining the exploit likelihood of vulnerability expectations based on all types of sensor measurements and observations, which was considered in our previous work [7-11]. It also helps focus context-specific relationships of observations. POMDP enables us to model the past, present, and future states of the real-world environment, observations, and actions. In POMDP, the exact current state is not known, but the belief state showing the distribution of probabilities over states is known. Data analytics determine the temporal causality and dependencies among features of all types of sensor measurements and observations.

[1]Hasan Cam is with the Network Science Division, Army Research Laboratory, Adelphi, MD 20783 USA (email: hasan.cam.civ@mail.mil).

In POMDP, the state of an environment is only partially observable, the observations are mapped to states via a probability distribution, and mapping of actions to states is probabilistic [2,5,6]. Solving POMDP refers to selecting the best action in a given state in order to maximize the value. The existing POMDP-solve techniques usually require *a priori* knowledge on state transitions, mapping observations to states, reward functions, and initial belief states. However, this paper leverages the logistic regression model and temporal causality analytics of observations to estimate the initial belief state. The proposed approach aims at learning model parameters over time so that they are eventually assigned more realistic values, based on temporal causality relationships and data analytics of forthcoming observations, along with the incident reports of cybersecurity analysts if available. That is, the belief state, state transitions, observations, and actions of POMDP are learnt gradually. This in turn provides more accurate modeling of malware-infected assets in identifying their present infection status, predicting their future status, and taking effective actions to mitigate the adverse impact of malware infection, exploitation, and spread.

This paper makes the following contributions: (*i*) not only detecting malware-infected assets but also taking actions to mitigate the adverse impact of malware infection and spread within a dynamic cybersecurity environment; (*ii*) integrating logistic regression and POMDP models and temporal causality relationships of all cyber observations such that the parameters of the models are assigned values collaboratively, where exploit likelihood of vulnerabilities is computed using logistic regression; (*iii*) developing learning decision tree to establish relationships between malware infection and indicators, and then to dynamically update the weights of relationships; (*iv*) extracting new features and patterns from the joint data analytics of various cyber sensor measurements, which are taken as inputs to the regression and POMDP models; (*v*) determining context-specific targeted assets using online data analytics and logistic regression, and, then, the POMDP model is used to infer malware-infected assets and to take effective actions for controlling malware infection and spread. Note that these observations of the logistic regression and data analytics refer to all available observations of the network, whereas the observations of the POMDP model refer to the observations of the environment of only those assets that are likely exploited. The advantage of applying POMDP model to a small number of assets is to make their state space tractable.

The remainder of the paper is organized as follows. Section 2 introduces the proposed protocol, called Online Detection and Mitigation of Infected Assets (ODMIA), by describing its steps using data analytics, logistic regression, malware decision diagram, regression analysis of features, and POMDP. Section 3 discusses the experiment. Section 4 contains concluding remarks.

II. The Protocol ODMIA

This section first describes the proposed protocol ODMIA, and then it discusses a detailed implementation of its steps by providing new algorithms, models, and examples.

Protocol ODMIA

Input: All types of cyber sensor measurements for intrusion detection, vulnerability scanning, network traffic and monitoring, and incident reports of analysts (if available).

Output: Malware-infected assets are detected, actions are taken to mitigate the adverse impact of malware infection and spread using POMDP; the initial exploit likelihoods of vulnerabilities are estimated using logistic regression and analytics of observations.

1. Perform pre-processing of all input datasets to align their formats and to adjust their sizes (if needed), and then feed them as input into the program performing the operations of the following steps.
2. Determine the relationships among the features of the same type of datasets (e.g., vulnerability scanning type of datasets may include host-based and network-based vulnerability scanning), where dataset types could be vulnerability scanning, intrusion detection, and flow traffic.
3. Add new data features to capture the joint relationships among datasets, and then determine the joint and cross relationships of all datasets by applying joint data analytics.
4. Obtain the initial exploit likelihoods of those vulnerabilities associated with the IP addresses appearing in intrusion alerts by running logistic regression program with new features, where exploit likelihood of a vulnerability correspond to the probability that observations and sensor measurements are indicative of its exploitation.
5. Develop a learning decision tree to help infer malware infection status of assets, where the root node checks out whether the vulnerability exploit likelihood of an asset exceeds a certain threshold.

6. Use the proposed POMDP model with reinforcement learning to assess the infection status of the selected assets, and then take actions to mitigate impact of malware infection and spread.
7. Validate the POMDP belief state of assets with the help of new observations and potential new patterns of malware infection and spread: Go to Step 2 to re-run all the above steps until no more improvement can be made in the detection and recovery of malware-infected assets.

A. Determining Relationships and Characteristics among the Features of Datasets (Steps 2 and 3 of Protocol ODMIA)

Steps 2 and 3 of ODMIA provide the ability to analyze and correlate vulnerability data within the datasets of the same type as well as across the datasets of different types such as intrusion detection, vulnerability scanning, and flow traffic types. The triggering temporal and causal relations between malicious activity and vulnerability data are inferred using the correlations of all features among datasets including intrusion, vulnerability, and network data. Figure 1 shows how the IP address of an intrusion detection system (IDS) alert and some ports' activities can be correlated to indicate any suspicious malware activity on a relevant host of the system.

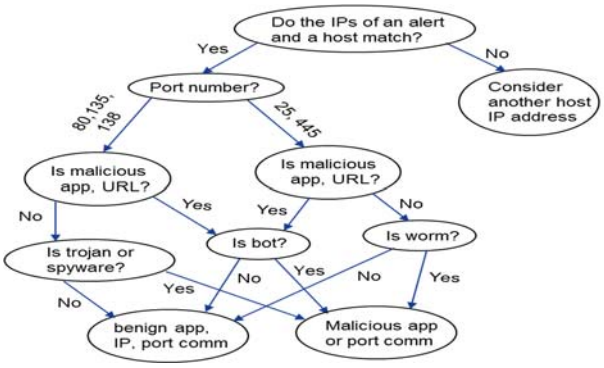

Figure 1. The relationships diagram of the alerts' IP and port addresses between cyber alerts and hosts of assets.

Next, we provide an example, shown in three phases, to infer the relationships among features of different types of sensor measurements, and then we establish the dependencies.

Phase 1 (see Figure 2):
(*i*) Given IDS alerts of an intrusion detection dataset, determine whether those assets whose IP addresses are the same as the source or destination IP addresses of IDS alerts are associated with some known vulnerabilities and common vulnerabilities and exposures (CVE) IDs, and (*ii*) especially consider those vulnerabilities of these assets with high severity.

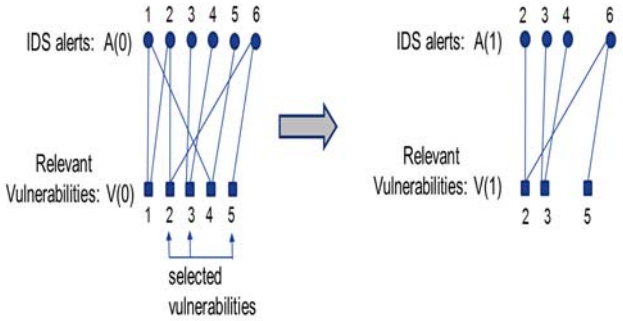

Figure 2. IDS alerts (e.g., Snort alerts) are associated with relevant vulnerabilities.

Phase 2 (see Figure 3):
(*i*) Search a set of known exploits, denoted $e_1, e_2, \ldots, e_m$, that are expected to involve the exploitation of the selected vulnerabilities in Step 1. Let $E(1) = \{e_1, e_2, e_3\}$.
(*ii*) Explore and capture all relationships among these exploits of E(1), IDS alerts of A(1), and vulnerabilities of V(1).

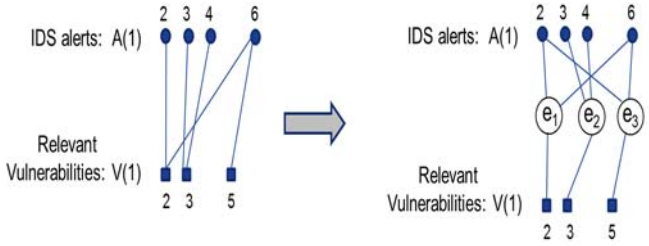

Figure 3. Determining exploits that are pointed to by IDS alerts and are capable of exploiting the relevant vulnerabilities.

Phase 3 (see Figure 4):
(*i*) Determine the IP addresses of assets associated with the selected vulnerabilities, and
(*ii*) Construct a profile of these assets' properties, dependencies, and causalities with respect to operating system (OS) services, processes, applications, communication links, etc.

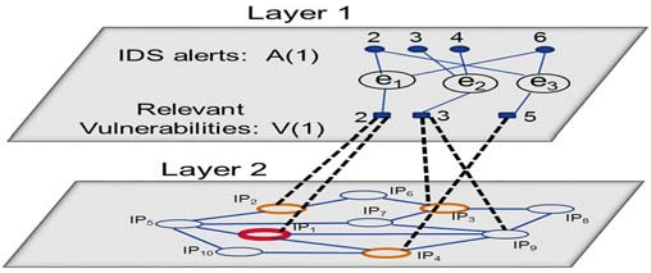

Figure 4. The dependencies and causality relationships among the above IP addresses associated with IDS alerts and relevant vulnerabilities.

To capture the relationships between intrusion detection, vulnerability scanning, and network monitoring data, new features are developed. For instance, new features such as exploit impact, traffic abnormality, time interval, hot IP, and vector time are developed for data analytics. For example, we introduced a new feature called CVE_exploitability to capture the relationships among the suspected IP and port addresses, the vulnerability feature CVE_severity of the assets with the same IP addresses, and their flow traffic. Figure 5 illustrates the regions A, B, and C of a regression tree for the new feature CVE_exploitability. These regression tree regions are used to quantify the significance of relationships of features.

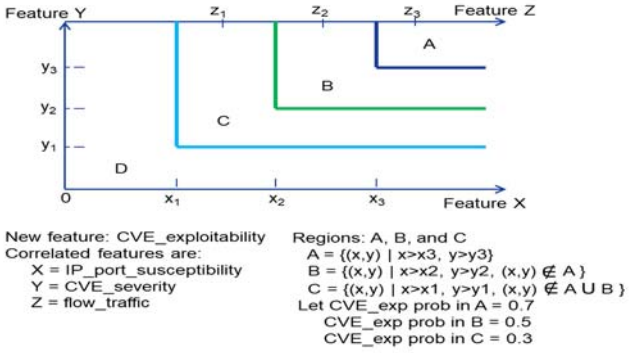

Figure 5. The regression tree partitions with regions A, B, C, and D of a new feature CVE_exploitability obtained by correlating the features of IP_port_susceptibility, CVE_severity, and flow_traffic.

B. Obtaining the Initial Exploit Likelihood of Vulnerabilities Associated with Intrusion Alerts' IPs via Logistic Regression (Step 4 of Protocol ODMIA)

Logistic regression predicts results on a binary outcome variable using one or more predictor variable. Logistic regression infers the "probability" of a particular outcome using a logistic function whose output always takes values between zero and 1, whereas its input can take an input with any value from negative to positive infinity. In ODMIA, the logistic regression is used to estimate the initial probabilities that all available datasets including intrusions, vulnerabilities, flow traffic, and incident reports are indicative of vulnerability exploitations. Then, those vulnerabilities have high probability of exploitation likelihoods are selected to focus on a context-specific set of assets.

Logistic function $\sigma(z)$ is defined as

$$\sigma(z) = e^z/(e^z+1) = 1//(1+e^{-z})$$
$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m,$$

where β_0 is the intercept and x_i is a predictor variable; logistic regression finds β_i parameter through an iterative search process. Note that the logistic regression technique is used to compute the probability that the features of intrusion alerts and vulnerabilities indicate the exploitation of a vulnerability. This probability corresponds to the probability value of an observation.

C. Decision Tree for Malware Infection (Step 5 of Protocol ODMIA)

We first developed new features for malware infection and spread, namely, vulExploit_prob_th, malDelivery, malBeacon, malAdvCC, malSpread, malStopSpread, malDegrade, and malNeutralize. Then, as shown in Figure 6, these new features, along with their indicators shown in dashed rectangles, are used to construct a malware infection decision diagram, where each rectangle requires a decision to be made based on the indicators obtained through all observations.

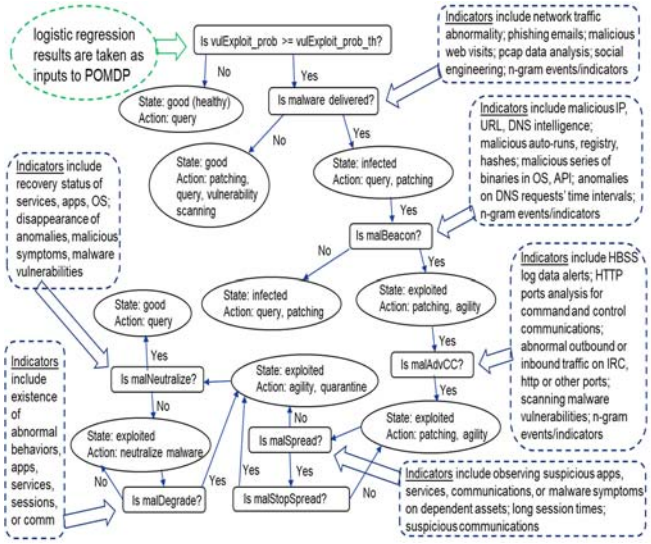

Figure 6. Malware infection decision diagram with indicators.

D. POMDP Model with Reinforcement Learning (Step 6 of Protocol ODMIA)

In identifying and controlling malware-infected assets, the POMDP state of an asset refers to its all-possible status in malware infection and spread. We consider three cases of POMDP states, namely, 2 States: [good, infected], 3 States: [good, infected, exploited], and 4 States: [good, infected, exploited, exploitedSpread]. For these actions, we consider five action groups: 1) ask: {query, ask next observation, ask patching vulnerability}; 2) resistMalDel: {patch, compliance, detect anomalies, data analysis}; 3) resistInfect: {patch, reinstall, reset, compliance}; 4) resistExploit: {shut down, quarantine, evict, disconnect}; and 5) resistExpSpread: {disconnect, quarantine}. Likewise,

we consider four observation groups: 1) detectGood: {detectMalNeutralize, detectNoMalSympthom}; 2) detectInfection:{detectMalDelivery}; 3) detectExploitation:{detectMalBeacon, detectMalAdvCC}; and 4) detectExpMalSpread: {detectMalSpread}.

For example, consider a POMDP state diagram with 2 states, 3 actions, 5 observations for a single asset, shown in Figure 7. The two states are good (i.e., healthy), and infected: S1: good, S2: infected. The three actions are as follows: (a1) ask; (a2) resistMalDel; and (a3) resistInfect. The five observations are: (z1) notInfected; (z2) malDelivery; (z3) malBeacon; (z4) malAdvCC; and (z5) malNeutralize. The rewards are determined, based on action and state.

Although the initial belief state is usually given (e.g., [0.6–0.4]), our approach determines it from the exploit likelihood of vulnerabilities in the logistic regression results. As shown in Figure 8, a POMDP policy graph state diagram indicates what actions should be taken based on observations. It represents policy as a finite state controller such that nodes represent vectors in value function, and edges represent transitions based on observations. Depending on how many infected or exploited assets are recovered or prevented from being exploited, the accuracy and benefit of policy graph can be evaluated, compared with a non-POMDP model, as illustrated in Figure 9.

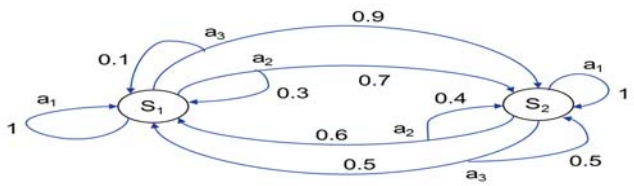

Figure 7. Two-state diagram with 3 actions for a single asset in POMDP.

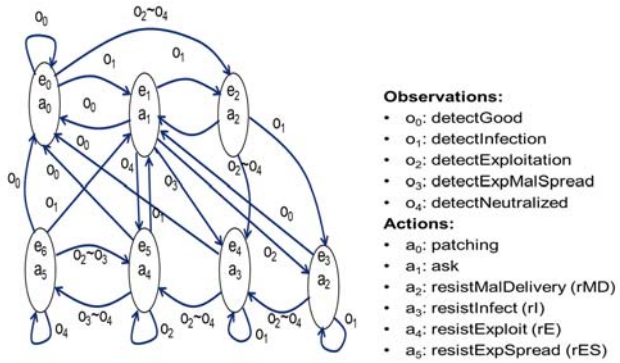

Figure 8. A policy transfer graph with five observations and six actions in POMDP.

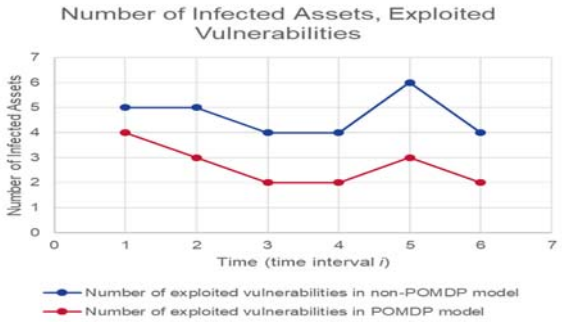

Figure 9. Comparing the number of infected assets in POMDP and non-POMDP models.

III. Experiments

We collected various types of cybersecurity real data including network vulnerability scanning, IDS alerts such as Snort alerts, and flow traffic flow data over a 3-week period in an organization. For these data, we also received the corresponding incident reports of cyber analysts. The data are pre-processed, and their formats are aligned. Due to the overwhelming size of data after vulnerability scanning, intrusion detection, and flow traffic, we reduced their sizes with the help of timestamp information in the incident reports. Then, they are taken as input to our python scripts that we developed in python, where we used the pandas module for data analytics. The features of all these data are cross-correlated, and then we introduced new features whose values are estimated with the regression tree algorithms that we developed using the regression tree partitions, such as the one shown in Figure 5.

To illustrate the correlation of features of different datasets, Figure 10 shows vulnerability features that include datetime, CVE, IP, port, CVSS vector, basescore, severity, whereas network traffic flow features contain datetime, protocol, srcIP, dstIP, srcport, dstport, client bytes, client packets, server bytes, and server packets. We built the following new features: hot_IP, CVE_exploitability, CVE_matching, flow_traffic_abnormality, exploit_Impact, vulnerability_exploit, and analyst decision. The values of these new features are estimated using all sensor measurements obtained over a three-week period. Then, these features' values are taken as input to a logistic regression model, making use of the packages statsmodels and sklearn. As illustrated in Figure 11, the response value (i.e., vulnerability exploit probability) of logistic regression shows the probabilities that sensor measurements are indicative of the exploitation of vulnerabilities. These probabilities correspond to the exploit likelihoods of vulnerabilities.

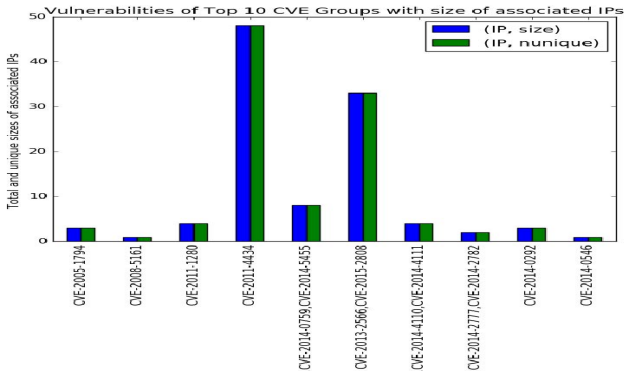

Figure 10. The data analytics results regarding the top 10 vulnerability CVE IDs along with the sizes of their associated IPs.

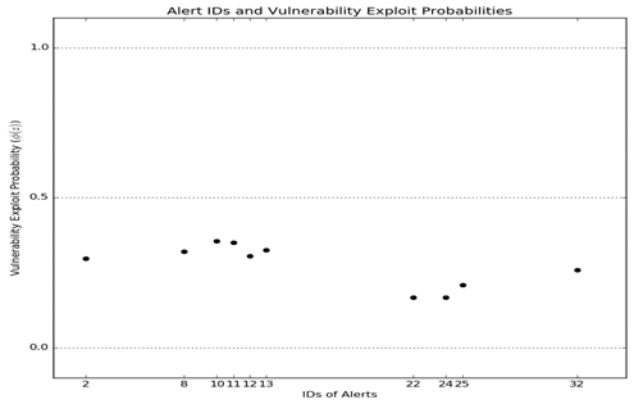

Figure 11. The estimation of vulnerability exploit likelihood by logistic regression model using IDS alerts along with all the relevant vulnerability and network traffic data.

IV. Conclusion

Antivirus software, intrusion detection, and vulnerability scanning tools can systematically search hosts for suspicious activity, signatures of malicious code, or security flaws. The difficulty with these tools is that they generate far too many alerts and indicators, many of which do not truly have security implications unless they are correlated and analyzed with the alerts and features of other types of cybersecurity observations in near real-time. In addition, the past and present patterns of a cybersecurity environment should be captured by appropriate scalable models. However, current cybersecurity tools and methods have limited capability, extensibility, and scalability to deal with such complicated situations and big data in real-time. A typical use of the existing models including POMDP is not scalable or practical. To help overcome these difficulties of these models, this paper has presented an online collaborative approach that can first select small-size, context-specific targeted assets that are likely infected based on logistic regression and temporal causality relationships of various types of cybersecurity data. Then, the proposed approach uses POMDP to take effective actions that are not necessarily optimal at a given time interval. Finally, this paper shows how the models of logistic regression and POMDP, along with the data analytics of temporal causality relationships, can collaboratively work in the detection and control of malware-infected assets.

References

[1] U.S. Army. "Army Network Campaign Plan, 2020 and Beyond", February 2015.

[2] L. P. Kaelbing, M. L. Littman, and A. Cassandra. "Planning and acting in partially observable stochastic domains". *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[3] T-F. Yen, V. Heorhiadi, A. Oprea, M. K. Reiter, and A. Juels. "An Epidemiological Study of Malware Encounters in a Large Enterprise". In: *Proceedings of CCS'14*, November 3–7, 2014. Scottsdale, AZ.

[4] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levine, and H. Owen. "Honeystat: Local Worm Detection using Honeypots". In: *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, pp. 39–58, 2004.

[5] S. M. McCarthy, A. Sinha, M. Tambe, and P. Manadhata. "Data Exfiltration Detection and Prevention: Virtually Distributed POMDPs for Practically Safer Networks". In: *Proceedings of Decision and Game Theory for Security (GameSec 2016)*, 2016.

[6] Y. Liu, S. Hu, and T.-Y. Ho. "Leveraging Strategic: Detection Techniques for Smart Home Pricing Cyberattacks". *IEEE Trans. on Dependable and Secure Computing*, vol. 13, no. 2, March/April 2016.

[7] H. Cam. "Risk Assessment by Dynamic Representation of Vulnerability, Exploitation, and Impact". In: *Proceedings of Cyber Sensing 2015, SPIE Defense, Security, and Sensing*, April 20–24, 2015, Baltimore, MD.

[8] J. R. Morris-King and H. Cam. "Controlling Proximity-Malware Infection in Diverse Tactical Mobile Networks Using K-Distance Pruning". In: *Proceedings of the MILCOM 2016*, November 1–3, 2016, Baltimore, MD.

[9] J. R. Morris-King and H. Cam. "Ecology-Inspired Cyber Risk Model for Propagation of Vulnerability Exploitation in Tactical Edge". In: *Proceedings of MILCOM 2015*, October 26–28, 2015, Tampa, FL.

[10] J. Morris-King and H. Cam. "Modeling Risk and Agility Interaction on Tactical Edge". In: *NATO Workshop IST-128-RWS-019 on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impac*t, June 15–17, 2015, Istanbul, Turkey.

[11] B. Thompson, J. R. Morris-King, and H. Cam, "Effectiveness of Proactive Reset for Mitigating Impact of Stealthy Attacks on Networks of Autonomous Systems". In: *Proceedings of the 2016 IEEE Conference on Communications and Network Security (CNS): International Workshop on Cyber-Physical Systems Security (CPS-Sec)*, October 17–19, 2016, Philadelphia, PA.