Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol.8 No.S1, 2019, pp. 74-79

Attack Impact Discovery and Recovery with Dynamic Bayesian Networks

T. Rajeshwari[1] and C. Thangamani[2]
[1]Research Scholar, [2]Associate Professor
[1&2]P.K.R Arts College for Women, Gobichettipalayam, Tamil Nadu, India
E-Mail: rajigee1995@gmail.com

***Abstract* - The network attacks are discovered using the Intrusion Detection Systems (IDS). Anomaly, signature and compound attack detection schemes are employed to fetch malicious data traffic activities. The attack impact analysis operations are carried out to discover the malicious objects in the network. The system objects are contaminated with process injection or hijacking. The attack ramification model discovers the contaminated objects. The dependency networks are built to model the information flow over the objects in the network. The dependency network is a directed graph built to indicate the data communication over the objects. The attack ramification models are designed with intrusion root information. The attack ramifications are applied to identify the malicious objects and contaminated objects. The attack ramifications are discovered with the information flows from the attack sources. The Attack Ramification with Bayesian Network (ARBN) scheme discovers the attack impact without the knowledge of the intrusion root. The probabilistic reasoning approach is employed to analyze the object state for ramification process. The objects lifetime is divided into temporal slices to verify the object state changes. The system call traces and object slices are correlated to construct the Temporal Dependency Network (TDN). The Bayesian Network (BN) is constructed with the uncertain data communication activities extracted from the TDN. The attack impact is fetched with loopy belief propagation on the BN model. The network security system is built with attack impact analysis and recovery operations. Live traffic data analysis process is carried out with improved temporal slicing concepts. Attack Ramification and Recovery with Dynamic Bayesian Network (ARRDBN) is built to support attack impact analysis and recovery tasks. The unsupervised attack handling mechanism automatically discovers the feasible solution for the associated attacks.**

Keywords: **Dynamic Bayesian Networks, Intrusion Detection Systems**

I. INTRODUCTION

Intrusion detection systems are the 'burglar alarms' of the computer security field. The aim is to defend a system by using a combination of an alarm that sounds whenever the site's security has been compromised and an entity—most often a Site Security Officer (SSO)—that can respond to the alarm and take appropriate action. This method should be contrasted with those that aim to strengthen the perimeter surrounding the computer system. Both of these methods should be used, along with others, to increase the chances of mounting a successful defense, relying on the age-old principle of defense in depth.

It should be noted that the intrusion can be one of a number of different types. For example, a user might steal a password and hence the means by which to prove his identity to the computer. Such a user is called as a masquerader. The detection of such intruders is an important problem for the field. Other important classes of intruders are people who are legitimate users of the system but who abuse their privileges and people who use pre-packed exploit scripts, often found on the Internet, to attack the system through a network. This is by no means an exhaustive list and the classification of threats to computer installations is an active area of research.

Two major principles are used in the intrusion detection process. They are anomaly detection and signature detection. The anomaly detection is relying on flagging all behavior abnormal for an entity. The signature based model uses the flagging behavior close to some defined pattern signature of a known intrusion. The problems with the anomaly detection approach are it does not necessarily detect undesirable behavior. The false alarm rates can be high in the anomaly based model. The problems with the signature based approach are absence of security policy.

An intrusion detection system consists of an audit data collection agent that collects information about the system being observed. This data is then either stored or processed directly by the detector. The output of detector is presented to the SSO. SSO can take further action with reference to the alarm.

II. INTRUSION DETECTION PRINCIPLES

A. Anomaly Detection

In anomaly detection, the system watches abnormalities in the traffic in question. The system takes the attitude which is abnormal and probably suspicious. The construction of such a detector starts by forming an opinion on what constitutes normal for the observed subject and then deciding on what percentage of the activity to flag as abnormal and how to make this particular decision. This detection principle thus flags behavior that is unlikely to originate from the normal process, without regard to actual intrusion scenarios.

Self-learning systems learn by example what constitutes normal for the installation; typically by observing traffic for

an extended period of time and building some model of the underlying process. A collective term for detectors that model the normal behavior of the system by the use of a stochastic model that does not take time series behavior into account. The system itself studies the traffic and formulates a number of rules that describe the normal operation of the system. In the detection stage, the system applies the rules and raises the alarm if the observed traffic forms a poor match with the rule base.

A system that collects simple and descriptive statistics from certain system parameters into a profile and constructs a distances vector for the observed traffic and the profile. If the distance is great enough, the system raises the alarm.This model is a more complex nature, taking time series behavior into account. Examples include techniques such as a Hidden Markov Model (HMM), an Artificial Neural Network (ANN) and other more or less exotic modeling techniques. An artificial neural network (ANN) is an example of a 'black box' modeling approach. The system's normal traffic is fed to an ANN, which subsequently 'learns' the pattern of normal traffic. The output of the ANN is then applied to new traffic and is used to form the intrusion detection decision. In the case of surveyed system, this output was not deemed of sufficient quality to be used to form the output directly, but rather was fed to a second level expert system that took the final decision.

The programmed class requires someone, be it a user or other functionary, who teaches the system programs it to detect certain anomalous events. Thus the user of the system forms an opinion on what is considered abnormal enough for the system to signal a security violation.These systems build a profile of normal statistical behavior by the parameters of the system by collecting descriptive statistics on a number of parameters. Such parameters can be the number of unsuccessful logins, the number of network connections, the number of commands with error returns, etc. In this class the collected statistics were used by higher level components to make a more abstract intrusion detection decision.

Here the user provides the system with simple but still compound rules to apply to the collected statistics.This is arguably the simplest example of the programmed—descriptive statistics detector. When the system has collected the necessary statistics, the user can program predefined thresholds that define whether to raise the alarm or not. An example is "number of unsuccessful login attempts > 3".The idea is to state explicitly the circumstances under which the observed system operates in a security-benign manner and to flag all deviations from this operation as intrusive. This has clear correspondence with a default deny security policy, formulating, as does the general legal system, which is permitted and labeling all else illegal. A formulation that while being far from common, is at least not unheard of.

In state series modeling, the policy for security benign operation is encoded as a set of states. The transitions between the states are implicit in the model, not explicit as when coded under a state machine in an expert system shell. As in any state machine, once it has matched one state, the intrusion detection system engine waits for the next transition to occur. If the monitored action is described as allowed, the system continues, while if the transition would take the system to another state, any state that is not explicitly mentioned will cause the system to sound the alarm. The monitored actions that can trigger transitions are usually security relevant actions such as file accesses, the opening of 'secure' communications ports, etc.

The rule matching engine is simpler than and not as powerful as a full expert system. There is no unification, for example. It does allow fuzzy matching, fuzzy in the sense that an attribute such as 'Write access to any file in the "tmp directory" could trigger a transition. Otherwise the actual specification of the security benign operation of the program could probably not be performed realistically.

B. Signature Detection

In signature detection, the intrusion detection decision is formed on the basis of knowledge of a model of the intrusive process and what traces it ought to leave in the observed system. The system can define in any and all instances what constitutes legal or illegal behavior and compare the observed behavior accordingly. It should be noted that these detectors try to detect evidence of intrusive activity irrespective of any idea of what the background traffic, i.e. normal behavior, of the system looks like. These detectors have to be able to operate no matter what constitutes the normal behavior of the system, looking instead for patterns or clues that are thought by the designers to stand out against the possible background traffic. This places very strict demands on the model of the nature of the intrusion. No sloppiness can be afforded here if the resulting detector is to have an acceptable detection and false alarm rate.

The system is programmed with an explicit decision rule, where the programmer has himself profiteered away the influence of the channel on the observation space. The detection rule is simple in the sense that it contains a straightforward coding of expected to be observed in the event of an intrusion. Thus, the idea is to state explicitly traces of the intrusion can be thought to occur uniquely in the observation space. This has clear correspondence with a default permit security policy, or the formulation that is common in law, i.e. listing illegal behavior and thereby defining all that is not explicitly listed as being permitted.

State-modeling encodes the intrusion as a number of different states, each of which has to be present in the observation space for the intrusion to be considered to have taken place. They are by their nature, time series models. Two subclasses exist: in the first, state transition, the states

that make up the intrusion form a simple chain that has to be traversed from beginning to end; in the second, petri-net, the states form a petri-net. In this case they can have a more general tree structure, in which several preparatory states can be fulfilled in any order, irrespective of the model where they occur.

An expert system is employed to reason about the security state of the system, given rules that describe intrusive behavior. Often forward-chaining, production-based tools are used, since these are most appropriate when dealing with systems where new facts are constantly entered into the system. These expert systems are often of considerable power and flexibility, allowing the user, access to powerful mechanisms such as unification. This comes at a cost to execution speed when compared with simpler methods.

String matching is a simple, often case sensitive, substring matching of the characters in text that is transmitted between systems, or that otherwise arise from the use of the system. Such a method is of course not in the least flexible, but it has the virtue of being simple to understand. Many efficient algorithms exist for the search for substrings in a longer string.These systems are similar to the more powerful expert system, but not as advanced. This often leads to speedier execution.

C. Compound Detectors

These detectors form a compound decision in view of a model of both the normal behaviour of the system and the intrusive behaviour of the intruder. The detector operates by detecting the intrusion against the background of the normal traffic in the system. The detectors are called as ‘signature inspired’ because the intrusive model is much stronger and more explicit than the normal model. These detectors would at the very least be able to qualify their decisions better, i.e. gives an improved indication of the quality of the alarm. Thus these systems are in some senses the most ‘advanced’ detectors surveyed. These systems automatically learn what constitutes intrusive and normal behavior for a system by being presented with examples of normal behavior interspersed with intrusive behavior.

The examples of intrusive behavior must thus be flagged as such by some outside authority for the system to be able to distinguish the two.There is only one example of such a system in this classification and it operates by automatically determining what observable features are interesting when forming the intrusion detection decision, isolating them and using them to form the intrusion detection decision later.

III. ATTACK AND ITS IMPACT DISCOVERY SCHEME

A. Graphical Model based Impact Analysis

In the last several decades, networked systems have grown in complexity and sophistication, introducing complex interdependencies amongst their umerous and diverse components. Attackers can leverage such interdependencies to penetrate seemingly well-guarded networks through sophisticated multi-step attacks. Explicit and implicit interdependencies exist at various layers of the hardware and software architecture. In particular, dependencies between vulnerabilities and dependencies between applications and services are critical for assessing the impact of multi-step attacks. These two classes of interdependencies have been traditionally studied using attack and dependency graphs respectively. Although significant work has been done in the area of both attack and dependency graphs, neither of these models can provide an accurate assessment of an attack’s impact, when used in isolation. To address this limitation, the system takes a mission-centric approach and present a solution to integrate these two powerful models into a unified framework that enables us to accurately assess the impact of multi-step attacks and identify high-impact attack paths within a network. This analysis can ultimately generate effective hardening recommendations, and can be seen as one phase of a continuous process that iteratively cycles through impact analysis and vulnerability remediation stages.

B. Probabilistic Mission Impact Assessment

Assessing and understanding the impact of scattered and widespread events onto a mission is a pertinacious problem. Current approaches attempting to solve mission impact assessment employ score-based algorithms leading to spurious results. A fourfold problem is identified with score-based algorithms: (1) score-based algorithms enforce deep training of experts to employed frameworks for specification (non-context-free), (2) require reference results for interpreting obtained results (non-bias-free), (3) require assessments outside of an experts’ expertise (non-local), and (4) require validation of end-results against ground truth. The model provides a formal, mathematical model for bias- and context-free mission impact assessment. Based on a probabilistic model the system reduces mission impact assessment to a well-understood mathematical problem based on definitions from local expertise and allow for a validation at data level. This is useful for areas and applications where qualitative assessments are required, such as assessments in critical infrastructures or military contexts.

C. Big Data Security Dependency Analyses

Intrusive multi-step attacks, such as Advanced Persistent Threat (APT) attacks, have plagued enterprises with significant financial losses and are the top reason for enterprises to increase their security budgets. Since these attacks are sophisticated and stealthy, they can remain undetected for years if individual steps are buried in background \noise." Enterprises is seeking solutions to connect the suspicious dots" across multiple activities. This requires ubiquitous system auditing for long periods of time, which in turn causes overwhelmingly large amount of

system audit events. Given a limited system budget, how to efficiently handle ever-increasing system audit logs is a great challenge.

The approach exploits the dependency among system events to reduce the number of log entries while still supporting high-quality forensic analysis. The aggregation algorithm preserves that preserves the dependency of events during data reduction to ensure the high quality of forensic analysis. Then the system uses an aggressive reduction algorithm and exploits domain knowledge for further data reduction. To validate the efficacy of the proposed approach, a comprehensive evaluation is contacted on real-world auditing systems using log traces of more than one month. The approach can significantly reduce the size of system logs and improve the efficiency of forensic analysis without losing accuracy.

D. Fault Detection in Large-Scale Cyber-Physical Systems

Detecting and isolating faults in Cyber-Physical Systems (CPSs), e.g., critical infrastructures, smart buildings/cities and the Internet-of-Things, are tasks that do generally scale badly with the CPS size. This work introduces a model-free Fault Detection and Diagnosis System (FDDS) designed having in mind scalability issues, so as to be able to detect and isolate faults in CPSs characterised by a large number of sensors. Following the model-free approach, the proposed FDDS learns the nominal fault-free conditions of the large-scale CPS autonomously by exploiting the temporal and spatial relationships existing among sensor data. A clustering method proposed to partition the large-scale CPS into groups of highly correlated sensors in order to grant scalability of the proposed FDDS. The design of model- and fault-free mechanisms to detect and isolate multiple sensor faults, and disambiguate between sensor faults and time variance of the physical phenomenon the cyber layer of CPS inspects.

IV. ATTACK RAMIFICATIONS USING TEMPORAL DEPENDENCY NETWORK

Assessing and mitigating the effects of successful attacks against computing systems are the natural and essential next step once attacks are detected. The central task of assessment and mitigation is to identify the ramifications of an attack, which include both malicious objects residing in a compromised system and objects that are contaminated by an attack.

However, successfully carrying out this task is faced with significant challenges. Specifically, attacks are usually very sophisticated, leveraging various vulnerabilities to compromise target systems and employing advanced attack vectors such as process injection/hijacking to subsequently contaminate system objects. Attacks' high complexity is further compounded by their increasing stealthiest, which commonly offers attackers a considerable amount of time before they are detected.

A generic strategy to solve these challenges is to monitor the information flow among objects in a computing system. For example, when a process reads from a file, a potential information flow is generated from the file to the process. Specifically, if the file contains malicious content such as exploits, a vulnerable process might be compromised. The dependency network serves as an effective method to model the information flow. A dependency network is a directed graph, where an edge e (v_i, v_j) indicates a potential information flow from the object vi to another object v_j . The provenance propagation methods employ dependency networks to identify all objects that have malicious information flows from the intrusion root, i.e., the entry point of an attack.

While these methods partially satisfy the objective to reveal malicious and contaminated objects in a system, their practical effectiveness is fundamentally constrained. The vast majority of these methods assume that the intrusion root is a known priori or can be easily located. Unfortunately, revealing intrusion root itself is a challenging task in practice considering the high complexity of object interactions in a system, the stealthiest of the attacks, and particularly the uncertainty caused by incomplete knowledge of all malicious actions. Therefore, such assumption is easily invalidated in practice, rendering these methods ineffective. These methods are also vulnerable todependency explosion, when a large number of intertwined but irrelevant object interactions are recorded and used to build dependency networks. For example, methods assume all objects that interact with a suspicious object are infected and use them to construct dependency networks. All processes that have read malicious files could be falsely considered as infected while only a few of them are actually contaminated by malicious content. Such approaches not only result in unnecessarily large graphs but also incur excessive false positives,

A few attempts have been proposed to mitigate dependency explosion by leveraging fine-grained logging or tracking. While they certainly lead to dependency networks that characterize the propagation of malicious information with higher fidelity, the performance cost could be prohibitively expensive. For example, BEEP divides a process to autonomous units and subsequently establish dependencies at unit-level. BEEP requires binary instrumentation of applications and mandates a priori analysis of applications. Therefore, it faces the difficulty in scaling to a large number of applications that typically run in modern systems. A more ambitious method has been proposed to perform byte-level dynamic taint tracking analysis, which incurs substantial run-time overhead.

In this paper, we present a novel, light-weight method to identify attack ramifications the method tackles the problem of undetermined intrusion root by leveraging dependency relationships of information flows between undetermined objects and a subset of objects with known security states. It overcomes dependency explosion by fusing evidence from

both known infected objects and known legitimate objects. Specifically, it leverages observations that i) an object could be infected if it has malicious information interactions with other known infected objects and ii) information interactions that involve known legitimate objects might be attack-irrelevant, thus providing clues for the real malicious information flows. It is worth noting that our method does not rely on fine-grained logging or tracking techniques. Instead, it uses only coarse-grained events, minimizes human efforts and incurs low run-time overhead. In order to increase the accuracy of description, our method splits the lifetime of an object into consecutive time slices to profile how the security state of this object changes over time, and considers explicit flows between different objects and implicit flows between different slices of an object.

Our method consists of three phases. First, it constructs a temporal dependency network (TDN) to correlate object-slices based on the inter object and intra-object information flows between them. Next, it builds a Bayesian network (BN) based infection model to characterize the infection propagation in TDN as a random process. Bayesian Network is used to take advantage of its capabilities of probabilistic inference over structured dependencies in TDN. Finally, it performs loopy belief propagation on the BN-based model to infer the security state of an object. To summarize, our work makes the following contributions:

1. We propose a new dependency network model, namely temporal dependency network (TDN), to include timing as an important dimension in the characterization of information flows. Our evaluation results based on extensive experiments have demonstrated that TDN, combined with our inference algorithm, boosts detection performance by 10.24% compared with existing coarse-grained dependency networks, which do not include timing information.
2. We devise a BN-based model to profile uncertainty introduced by the incomplete observation of attacks and real dependencies among objects.
3. We introduce a probabilistic inference method using loopy belief propagation to estimate the security states of undetermined objects. Our inference method does not only employ known compromised objects, but also, for the first time, incorporate evidence offered by objects with confirmed legitimate security states.
4. We have performed extensive evaluation using a large dataset of 389 attacks launched by real-world malware samples including highly sophisticated ones such as Stuxnet. These attacks cover a large variety of attack vectors, vulnerabilities, malware families, and background workloads. Evaluation results have shown that our proposed method is effective in attack ramification analysis with a 97.47% precision at 97.21% recall on averagewithout the knowledge of intrusion root, and with a98.24% precision at 97.87% recall with the knowledgeof intrusion root. In addition, our method incurs less than5% run-time overhead.

This work is an extension of our conference paper. We has made substantial improvements in following perspectives.

First, we compress the temporal dependency networkby merging redundant nodes and edges without affectingits effectiveness. This significantly reduces the size of thedependency network, making both storage and analysis muchmore efficient. Second, we construct a refined TDN of finergrained objects such as memory blocks, threads. The refinedTDN, or temporal dependency network with memory-relateddependencies (TDN-M), enables the analysis of sophisticatedsituations of novel attacks, such as localized infections.

Meanwhile, we propose a new method to capture executable-memory relateddependencies by combining system-level events and callstack traces. Third, we have redesigned probabilistic modelbetween each node and its parents as a two-step infectionpropagation process. The attack-specific information is explicitly modeled as random variables and shared amongmultiple dependencies. This facilitates the analysis of multistepattacks that the acquired evidence of some attack stepscan also contribute to the analysis of otherattack steps via these random variables.

Fourth, we have enhanced our Bayesian network model to infer over uncertainevidence introduced by inaccurate observations such as falsepositives or negatives of security checks. We have investigatedthe uncertainty of evidence and analyzed their impact onattack ramification identification. Finally, we conduct moreexperiments to evaluate the effectiveness of our method ona large dataset of 389 attacks launched by real-world malwaresamples including highly sophisticated ones such as Stuxnet.

V. CONCLUSION

The survey is conducted on the areas of the attack discovery and impact analysis domains. The Graphical Model based Impact Analysis, Probabilistic Mission Impact Assessment, Big Data Security Dependency Analyses and Fault Detection in Large-Scale Cyber-Physical System mechanisms are analyzed with different merits and demerits.The lightweight temporal dependency network method is able to identify attack ramifications without knowledge of intrusion root and less subject to dependency explosion. The lifetime of an object is split into consecutive time slices (object-slices) to profile how the security state of this object changes over time, a temporal dependency network (TDN) from system call traces to correlate object-slices according to information flows between them. A Bayesian network (BN) model is built to characterize the uncertainties of infection propagations in the TDN. Loopy belief propagation performed based on the BN model to infer the security state of an object by fusing evidence derived from infected and legitimate objects whose security states are known.

REFERENCES

[1] Yuan Yang, ZhongminCai, Chunyan Wang and Junjie Zhang, "Probabilistically Inferring Attack Ramifications Using Temporal Dependency Network", *IEEE Transactions on Information Forensics and Security*.,Vol. 13, No. 11, 2018.

[2] Z. Xu, Z. Wu, Z. Li, Jee, J. Rhee, X. Xiao and G. Jiang, "High fidelity data reduction for big data security dependency analyses," *in Proc. ACM CCS'16*., pp. 504–516, 2016.

[3] C. Alippi, S. Ntalampiras, and M. Roveri, "Model-free fault detection and isolation in large-scale cyber-physical systems," *IEEE Trans. Emerg. Topics Comput. Intell*., Vol. 1, No. 1. pp. 61–71, Feb. 2017.

[4] M. Albanese and S. Jajodia, "A graphical model to assess the impact of multi-step attacks," *J. Defense Modeling and Simulation*, Vol. 15, No. 1. pp. 79–93, Apr. 2017.

[5] A. Motzek and R. M¨oller, "Context- and bias-free probabilistic mission impact assessment," *Comput. Secur.*, Vol. 65. No. C. Mar. 2017.

[6] A. Motzek, R. M¨oller, M. Lange, and S. Dubus, "Probabilistic mission impact assessment based on widespread local events," *in Proc. NATO IST-128 Workshop: Assessing Mission Impact of Cyberattacks*., pp. 16–22, 2015.

[7] W. K. Sze and R. Sekar, "Provenance-based integrity protection for windows," *in Proc. ACSAC'1*, pp. 211–220, 2015.

[8] S. Ntalampiras, "Detection of integrity attacks in cyber-physical critical infrastructures using ensemble modeling," *IEEE Trans. Ind. Informat*., Vol. 11, No. 1. pp. 104–111, Feb. 2015.

[9] A. Motzek and R. M¨oller, "Indirect causes in dynamic bayesian networks revisited," *in Proc. IJCAI'15*, pp. 703–709, 2015.