International Journal of Engineering Research & Management Technology
ISSN: 2348-4039
May-June-2020 Volume 7, Issue-3
Email: editor@ijermt.org
www.ijermt.org

SYSTEM SECURITY-THREATS USING INNOVATIVE SECURITY TECHNIQUES IN PROACTIVE THREAT MANAGEMENT

Shri Gurinder Singh

Research Scholar, Dept. of Computer Science and Information Technology, Kalinga University

Dr. Yash Pal Singh

Professor, Dept. of Computer Science and Information Technology, Kalinga University

ABSTRACT

In order to prevent system compromise to the greatest extent possible, this work proposes a proactive approach to threat management to assist system designers in putting well-structured security processes into place from early phases of software development life cycle. In this security process, a hybrid technique has been developed for the elicitation of perfect and significant security requirements, which serve as the cornerstone of every secure software architecture. This ground-breaking method's hybrid process diagram aids in deciding which security features should be included as countermeasures in light of the evolving security requirements. This study adopts a multi-layered defence method as an improvement to overcome the limitations of single ring security. Due to the binary nature of current security methods, the system can only exist in one of two states: a secure state or a failed state. Fuzzy logic-based proactive defence mechanisms have been developed and implemented in this security procedure to prevent software system failure, which could be disastrous from the perspective of the user. The top quality sought after in security systems nowadays to combat skilled hackers' multifaceted threats is "adaptivity and intelligence." In order to fulfil the goal of software security, multi-agent system planning for threat avoidance (MASPTA), where autonomous agents cooperate and interact with one another, has been chosen as a proactive threat management technique. The Multi-Agent System Planning for Threat Avoidance Optimally (MASPTA-O) and the Optimal Countermeasures Identification Method (OCIM) are two more methodologies presented in this study that can be used to identify the best countermeasures for managing threats and vulnerabilities. It has evolved into the key component of every security initiative in the modern economic climate.

KEYWORDS: Multi-Agent System, Adaptivity, Intelligence, Defence, Security

INTRODUCTION

Today's media regularly features reports concerning cyberattacks and organisations being infiltrated. The functioning of all major institutions, including the government, the corporate sector, and the public, has been significantly impacted by cyberattacks. Many organisations use standard security measures and rely on patching and antivirus software to address random vulnerabilities, but they are unaware of the risks that have the highest likelihood of being compromised. Organisations nowadays must face the reality that every company has a high likelihood of compromising and may not even be aware of compromises that have already occurred within their organisation. Even if we might not be able to stop attacks today, detection is still essential. Attackers' jobs are made considerably easier since they only need to find one target, or vulnerability, but we must protect our IT assets from all threats and weaknesses. With the growth of ICT, internet penetration, and mobile devices, India has a very difficult cyber threat environment and is one of the top few nations with malicious activity. Attacks such as website defacement, invasions, and targeted attacks have been made against the government, the public sector, and the commercial sector. Such assaults have been mostly focused on information theft, phishing, and denial of service.

Today, breaking into websites includes setting up web shells (trapdoors) that give ongoing access to the databases housed inside. The assaults include routinely stealing sensitive data, stealing of Personally Identifiable Information (PII), and identity theft. These trapdoors have a self-destructive tendency and can end their lives after finishing a task or when they have a chance to be discovered. Web shells are designed to escape antivirus programmes, so they cannot be identified by them. Through a network of compromised computers known as BOTNETS, these trapdoors interact with their command and control centres. It is possible to communicate with C&C servers while remaining anonymous thanks to the usage of proxy servers and TOR (the onion ring) servers. Since these TOR and proxy servers are dynamic in nature, it is impossible to identify them and attribute them. Even though many businesses are aware of the terrible effects that Advanced Persistent Threats (APTs) have had on them, some are still in denial. They believe they are immune from these attacks. However, in the current context, attacks on organisations are not impossible; it is simply a matter of when.

MOTIVATION

In the current climate of global security, it might not always be viable to stop threats with the security measures outlined above when used alone. The suggested threat-oriented security architecture consequently incorporates these proactive techniques at several tiers for the best possible mitigation of known and unanticipated threats. Every layer of this architecture makes improvements to the security measures already in place and offers many security mechanisms for reducing, managing, and monitoring current security threats. In addition to the security measures that have been developed and discussed above, the idea of research honeytokens has been put forth for the detection of unanticipated threats, while meta-agents in conjunction with fuzzy logic have been used for the monitoring and management of threats in the proposed threat-oriented security model. Additionally, some built-in security recommendations have been suggested at the model's zero layer as a dual-protection method against potential invasions. Then, as part of proactive risk management, this security model is incorporated into the spiral framework's risk analysis section to strengthen and improve security. Since the spiral process is cyclical and iterative in nature, it gradually lowers the risk associated with threats faced during the development life cycle with each consecutive pass. But the proposed security model may need to be reviewed on a regular basis to stay up to date with changing threat perceptions.

REVIEW OF LITERATURE

IBM According to IBM (2014), "Proactive response to today's Advanced Persistent Threats"The article points out that APTs are becoming more dynamic and stealthy, and that high performance response systems are needed. The time lag between the disclosure of a vulnerability and the availability of the exploit code is rapidly shrinking and now only a few hours long. Attackers create exploits more quickly than application owners. Organisations must maintain ongoing compliance in order to stop threats, as well as analytical skills in order to fortify infrastructure against assaults. Pervasive real-time visibility, automatic remediation, and global scalability are necessary for the incident response process.

"Safeguarding business in a time of expanding and evolving cyber threats 2014" Verizon paper (IBM 2013)22 demonstrates how every security breach is a conflict between an attacker and a target. It is a race against time and a test of skill, and it frequently starts even before the assailant moves physically. Organizations may detect and respond to threats faster—in days rather than months—with the aid of more sophisticated cyber security threat management capabilities. According to the report, 69% of breaches were found by outside parties, and 66% go months without being noticed. By disrupting the sequence of events brought on by an incident or any APT stage, organizations can stop attacks. There is a concept of defending "threat actors" that conventional security procedures are unable to identify.

RESEARCH METHODOLOGY

For the study, both primary and secondary data were employed. Primary information was gathered from selected respondents in the organizations. The secondary material was gathered from newspapers, numerous magazines, organization websites, and CERTs from different nations. The core data was gathered using the questionnaire survey approach. The secondary data was acquired through online desk research. The information produced is both quantitative and qualitative. The qualitative data are the non-numerical data made up of views and observations, whereas the quantitative data are primarily the numerical data gathered from records and questionnaire replies.

SAMPLING UNIT: CIIs of India, such as the railroads, nuclear energy, private IT firms like Tata Teleservices, HCL Tech, and Wipro, as well as businesses that provide security-related goods and services, were selected as the sampling unit from which the study's requisite data could be gathered.

SAMPLING FRAME: The organization that was selected served as the basis for the sampling frame. For the questionnaire survey, the Chief Information Security Officers (CISOs) or Chief Information Officers (CIOs) of each organization were chosen. Where CISO/CIO appointments were not possible, the individuals in charge of computer security were included.

SAMPLE SIZE: Information gathered from both the public and private sectors has been examined. The ideal sample size was selected in order to satisfy the criteria for efficacy, representativeness, dependability, and flexibility. Additionally, the population variance was considered when choosing the sample size.

RESULTS AND DATA INTERPRETATION

INNOVATIVE SECURITY TECHNIQUES IN PROACTIVE THREAT MANAGEMENT

The implementation of several preventative measures in threat management for the security of software systems is covered in this chapter. As a result of threats to the assets from unscrupulous users, secure software systems play a crucial part in security needs. Therefore, a new hybrid technique for eliciting meaningful and realistic security needs has been described in this chapter. It was developed by combining the advantages of misuse scenarios with attack trees. The use of the defense-in-depth strategy is also covered in this chapter. By addressing the shallowness in the single-layer software security, it provides multi-level security protection to fend off modern threats from knowledgeable hackers. This chapter also makes a proactive suggestion for a novel method using fuzzy logic to break the current security mechanisms' binary principle-based jinx of brittleness. Fuzzy logic was used in this method to create a "partially secure state" that evolved between a "safe state" and a "failed state," which serves as a warning signal to take the necessary additional preventive steps to keep the system from reaching the failed state as much as feasible.

INNOVATIVE SECURITY TECHNIQUES IN PROACTIVE THREATMANAGEMENT

Protecting software and the resources it uses is the focus of software security. A significant source of security threats in software is design-level flaws. Security threats, or prospective assaults, such as misuses and anomalies that go against the security objectives of a system's intended operations, are created when these vulnerabilities are exploited. In order to apply security features for threat mitigation in safe software engineering, managing these threats indicates what, where, and how to do so. In this section, we've covered a few proactive threat management techniques like the hybrid technique for gathering security requirements, the multi-layered defence strategy for thwarting threats, and the use of fuzzy logic to prevent failed states as much as possible in software system security. To address the problems of the evolving security environment of the modern day, these strategies may be employed during the design phase of the software life cycle for software system security.

HYBRID TECHNIQUE FOR ELICITATION OF SECURITY REQUIREMENTS

By combining the benefits of attack trees and abuse situations, which are the fundamental foundation of software system security, a hybrid technique has been developed for the development of realistic and relevant security requirements. As shown below, Common Criteria, Misuse Cases, and Attack Trees are being used to elicit security needs.

Common Standards: The National Institute for Standards and Technology (NIST) created the Common Criteria (CC), a thorough, standardised procedure for security requirements elicitation, specification, and analysis. The output document created by this method is very readable and beneficial in assessing the general security requirements of information technology systems. A Protection Profile (PP) and a Security Target (ST) are the two types of documents that CC creates. The ST indicates what a product actually does that is security-relevant, whereas the PP identifies the desired security qualities of a product created by a group of consumers. The disadvantage of Common Criteria is that it is a complicated process that involves numerous specialised work products, including security objectives, security requirements, security policies, functional specifications, and security policy models. The process of developing reliable information security products requires these work products. Even those with a strong technical expertise may find it challenging to comprehend the work products and the connections between them.

TABLE -1: THREAT DESCRIPTIONS

Threat-Id	Threat Description	Threat Category	Risk LevelR	Probability of threat realization P(T) = R / 10
1	Malicious user views confidential on- the- wirepayroll data	InformationDisclosure	9	0.9
2	ker uploads rogue Webpages and code	Tampering with data	8.2	0.82
3	cker denies service toapplication	Denial of service	7.6	0.76
4	Spoof computer executing the process client request process	Spoofing	6.4	0.64
5	Attacker elevates privilege by leveraging the service client request process	Elevation of privilege	5.0	0.5

IMPLEMENTATION USING MATLAB

To create a fuzzy inference system, we employed the MATLAB Fuzzy Tool Box, a graphical user interface tool. The primary GUI tools in the toolbox used in this instance are the following: Fuzzy Inference System Editor, Membership Function Editor, Rule Editor, and Rule Viewer.

An overview of the fuzzy inference system is given by the fuzzy inference system (FIS) editor. It displays the relationship between the system's input and its matching output. Through the FIS editor, it is possible to modify the FIS's processing techniques and input variable names. Through the membership editor, names and numerical parameters of membership functions connected to all input and output variables can be specified and modified. For each input and output variable, we have utilised a triangular membership function to represent three areas. In Figure 4.8, the FIS Editor is used to represent the three input parameters Confidentiality (C), Integrity (I), and Availability (A), as well as the output parameter Security Level (SL), for the system.

ANALYSIS OF ADOPTED PROACTIVE SECURITY MEASURES

Adoption of the security measures mentioned above demonstrates how these cutting-edge methods play a crucial role in augmenting and fortifying the conventional security defences in threat management. These proactive procedures offer the security remedies required to handle the issues presented by the current digital era, as listed below.

A comparative analysis of three methods for gathering security requirements: attack trees, misuse cases, and common criteria. They have said that each strategy has its own advantages and disadvantages, and that when used in tandem, they can be advantageous. The Common Criteria are challenging to understand and apply, but simple to analyse. Misuse Cases offer output that is challenging to comprehend but are simple to learn and utilise. Attack Trees, on the other hand, produce unambiguous output but are challenging to analyse. They contend that by integrating different methodologies, it is possible to learn more about a system's security. By inventing a hybrid technique that combines the benefits of attack trees with misuse instances to create realistic and relevant security requirements, we have continued their work and strengthened the system's ability to successfully eliminate flaws in big and complex systems. Although this technique improves our understanding of analysis and design in threat management by merging threats, their mode of realisation, and related remedies, it has several drawbacks. This method could lead to the realisation of the threat because it does not convey the hint that mitigation actions have failed. The ensuing chapters provide separate discussions of the security solutions on this account. A single defence approach, which has been determined to be inadequate in the current security environment, is now used to keep hackers from breaching the system's security. Therefore, a multi-layered defence system has been suggested in this chapter as a proactive method of reducing the dangers brought on by threats. When used, this method lowers the risk of failure compared to a single layer defence system. This makes it extremely challenging for attackers to breach software security's outermost defences.

CONCLUSION

In contrast to the traditional security methods, the initial layer of this threat-oriented security architecture captures both known and unforeseen threats as part of proactive threat management. An integrated mechanism for the optimisation of threats detected in the first layer and in need of mitigation is presented in the second layer of the proposed model. Gidi Cohen's work, which simply provided outlines for detecting key threats, is improved by the layered threat elimination model augmented with the method presented in this layer for determining optimal countermeasures. Application of the integrated mechanism described in this study results in an average 37% reduction in the number of security measures needed to thwart threats, demonstrating its superior economic viability versus conventional security measures. The Pareto 80-20 Rule, which claims that 80% of dangers can be eliminated by mitigating only 20% of the system's principal threats, is further supported by the two-level optimisation in this layer.

The third layer of the suggested approach uses hybrid technique and multi-agent system planning to combat above optimised risks. In order to map the aforementioned risks into security requirements, a hybrid technique has been developed in this study that clearly outperforms the ones now in use by combining the strengths of attack trees and misuse situations. This ground-breaking method aids system designers in locating security characteristics that should be included as deterrents. Traditional security solutions now have a new dimension because to the threat mitigation provided by multi-agent system planning in this layer of the proposed paradigm. Here, a number of intelligent agents carry out their predetermined actions when the necessary preconditions are met without the need for human involvement. The fourth layer of the proposed model, which uses meta-agents in conjunction with fuzzy logic in a multi-agent environment, contributes to the monitoring and management of threats to the system. Fuzzy logic has been used in this layer's new guarded approach as a proactive solution to avoid the traditional security measures' curse of brittleness by establishing security zones between the safe state (green zone) and failing state (red zone). This defence mechanism is distinctive in that it responds to Olthoff's work, which has subtly pointed out limitations of current security mechanisms based on binary principle, by using meta-agents in conjunction with fuzzy logic for monitoring and management of threats in multi-agent environments.

REFERENCES

1. Li, X., Liu, R., Feng, Z., & He, K. (2009). Threat-Modeling Oriented Attack Path EvaluatingAlgorithm. Transactions of Tianjin University, Springer. 15(3), 162-167, doi: 10.1007/s12209-009-0029-y.
2. Lipner, S.B. (2004). The Trustworthy Computing Security Development Lifecycle. In 20th Annual Computer Security Applications Conference (pp. 2-13). Tucson, AZ, USA: IEEE Computer Society.
3. Madan, B.B., Popsotojanova, K.G., Vaidyanathan, K. & Trivedi, K.S. (2002). Modeling andQuantification of Security Attributes of Software System. In International Conference on Dependable Systems and Networks (pp. 505-514). Bethesda, MD, USA: IEEE Computer Society.
4. Mamdani, E.H., & Assilian, S. (1975). An experiment in linguistics synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 7(1), 2-15.

5. Marcus, D., & Sherstobitoff, R. (2012). Dissecting Operation Highrloller. McAfee, Santa Clara, CA.Retrieved from http://www.mcafee.com/us/resources/reports/rp-operation- high-roller.pdf.
6. Mauw, S. and Oostdijk,M. (2006). Foundations of Attack Trees. In 8th International Conference on Security and Cryptology (pp. 186-198). LNCS, Volume 3935/2006, Seoul, Korea: Springer, doi: 10.1007/11734727_17.
7. McAfee Labs (2012, First Quarter). McAfee Threats Report: First Quarter 2012. Retrieved from http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2012.pdf.
8. McConnell, S. (1993), Code Complete. Redmond, Washington: Microsoft Press.
9. McDermott J., & Fox, C. (1999). Using Abuse Case Models for Security Requirements Analysis. In 15th Annual Computer Security Applications Conference (pp. 55-64). Scottsdale, AZ, USA: IEEE Computer Society.
10. McGraw, G. (2004, March-April). Software Security. IEEE Security & Privacy, 2(2), 80-83.
11. McGraw, G. (2006). Software Security: Building Security In. Boston, United States ofAmerica: Addison Wesley.
12. Moradian, E., & Hakansson, A. (2010). Controlling Security of Software Development with Multi-agent System. In Knowledge-Based and Intelligent Information and Engineering Systems (pp. 98-107). Cardiff, UK: LNCS, Volume 6279, Springer.
13. Myagmar, S., Lee, A.J., & Yurcik, W. (2005, August). Threat Modeling as a Basis for Security Requirements. Paper in IEEE Symposium on Requirement Engineering for Information Security, Paris, France.
14. NIST (2011). National Vulnerability Database (NVD), Technical Report from National Institute of Standards and Technology. Retrieved from http://web.nvd.nist.gov/ view/vuln/detail?vulnId=CVE-2011-1851.
15. Oladimeji, E., Supakkul, S., & Chung, L. (2006). Security Threat Modeling and Analysis: A Goal-Oriented Approach. In 10th IASTED International Conference on SoftwareEngineering and Applications (pp. 178-185). Dallas, Texas, USA: ACTA Press.
16. Olthoff, K.G. (2001). Observations on Security Requirements Engineering. In Symposiumon Requirements Engineering for Information Security, Indianapolis, USA.
17. Opdahl, A.L., & Sindre, G. (2009). Experimental comparison of attack trees and misusecases for security threat identification. Journal of Information and Software Technology, Elsevier, 51(5), 916-932, doi: http://dx.doi.org/10.1016/j.infsof.2008.05.013.