Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 4 (2012) 487 – 494

Procedia Technology

C3IT-2012

Threat-Oriented Security Framework: A Proactive Approach in Threat Management

Vandana Gandotra[a], Archana Singhal[a], Punam Bedi[a]

[a]*University of Delhi, Delhi,India*

Abstract

Present day sophisticated and innovative attacks have resulted in exponentially increasing security problems. This paper therefore presents a three phased threat-oriented security model to meet the above security challenges as a part of proactive threat management. Integration of threat management with development process in the proposed work provides necessary security cover against both known and unknown threats. Identification of these threats has been made possible by fusion of threat modeling process and research honeytokens in conjunction with statistical model in the first phase. Necessary security measures to mitigate above identified threats have been adopted in the second phase using multi-agent system planning. Risk reduction as a result of adoption of countermeasures has been evaluated in the third phase using meta-agents in multi-agent environment. This three-phased model is placed in the risk analysis segment of the spiral model to enhance and strengthen the security as a part of proactive risk management. This work gives a new and innovative approach to provide security against all types of threats and has an edge over traditional techniques which only cater to predictable threats in risk management.

Keywords: threat modeling process; research honeytokens; unknown threats; multi-agent system planning; meta-agents; spiral model

1. Introduction

With changing threat perceptions in the present day security scenario the necessity of threat management in the system development life cycle is simply indispensable [4]. This paper therefore presents a three phased threat-oriented security model adopted as a proactive step in risk management to meet all types of threats both known and unknown. Identification and prioritization of threats is the function of first phase. Here threat modeling process has been used for identification of known threats whereas a detection mechanism has been planted using research honeytokens in conjunction with statistical model for detection of unknown threats.

In the proposed work data fusion and information correlation from two sources i.e. statistical model in conjunction with research honeytokens ensures that as many as possible threats are discovered by developers not by malicious users.

The above identified threats are then neutralized using multi-agent system planning in the second phase. Evaluation of risk reduction in respect of both known and unknown threats after adopting necessary

2212-0173 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.protcy.2012.05.078

countermeasures is the responsibility of the third phase. Meta-agents have been inducted in this phase to provide control of software agents by keeping track on them [6, 7]. This is to ensure whether the countermeasures applied in Phase-2 are able to mitigate the threats as identified in Phase-1. These meta-agents automatically monitor the performance of software agents in a multi-agent system and provide a security checklist to the designers and developers for taking appropriate corrective actions. This three-phased model is embedded in the risk analysis segment of the spiral model to augment and strengthen the software security as a part of proactive threat management. As the spiral process is cyclic and iterative in nature subsequent passes of spiral process progressively result in elimination of all possible threats in developing secure software systems.

This paper is organized as follows: Section 2 provides an overview of the pioneering works done by leading researchers in this area. Proposed threat-oriented security model has been discussed in Section 3. Adoption of the proposed model in the spiral framework has been explained in Section 4. Analysis of the proposed model is given in Section 5. Section 6 concludes the paper and gives outlines for future work in this area.

2. Previous Research

Security must be deeply integrated into the full software development life cycle to match the evolutionary nature of threats manifested these days. Different authors and researchers are already working in this direction for developing innovative techniques to meet these security challenges [1, 4]. New concepts regarding secure software systems have been given by Essafi et al. in their paper [8]. Threat modeling approach for identifying, documenting and mitigating security threats to a software system has been given by Swiderski and Synder [2]. Nowadays agent technology and multi-agent system planning is gaining popularity in information security and risk management. Research in the area of multi-agent system planning for enhancing the security of the software systems has been presented by Bedi et al. in their approach MASPTA [1]. Moradian and Hakansson [6, 7] have introduced the approach to solving security problems in software systems using meta-agents in multi-agent environment.

Honeypots, a new technology with enormous potential is used these days for detecting and gathering information about threats. Lance Spitzner [10, 13] has given a new dimension to this area of research and has formalized the idea of honeytokens. Different authors [14, 16] have also worked in this field to further strengthen the security of software systems by developing responses to new attacks of black hats. Spiral model developed by Boehm based on a risk-driven and cyclic approach has also contributed towards risk reduction for developing secure software systems [3, 11].

3. Proposed Threat Oriented Security Model

Traditional technologies used these days can prevent known threats but they are unable to identify and avert unknown threats. Evidently failure of software security in this information age can be catastrophic and is unacceptable as it carries confidential data of various applications. Threat-oriented security model is a step in this direction and is embedded in risk analysis segment of spiral process to identify and eliminate all possible threats progressively for risk reduction to the acceptance level of software security. Three phases of the proposed model are explained below.

3.1. Identification of Threats

This phase deals with identification of all possible threats whether known or unknown to the system under development. Known or predictable threats have been identified using threat modeling process while research honeytokens in conjunction with statistical technique has been adopted for identification of unknown threats as detailed below.

3.1.1. Identification of Known Threats using Threat Modeling Process

Threat modeling process provides a structured way to secure software design which involves understanding an adversary's goal in attacking a system based on system's assets of interest. Threat modeling process consists of characterizing the security of the system, identifying assets and access points and determining threats [2]. Various entities defined during the threat modeling process and their relationship has been indicated in the Threat Entity Relationship (T-E-R) diagram as shown in Fig. 1.

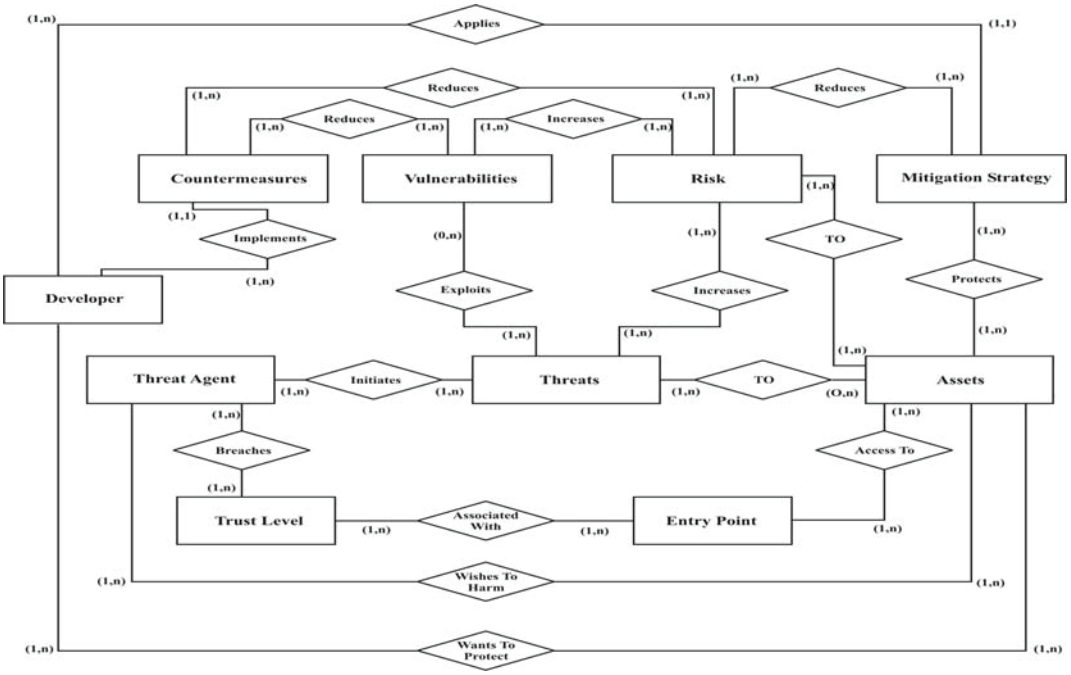

Fig. 1. Threat-Entity Relationship Diagram

As shown in diagram given above, a threat agent is an adversary who is motivated to harm/exploit the system in an unauthorized manner to drive benefit. He is primarily responsible for the threat to the assets of the system and can traverse the privilege boundaries in an illegal way to reach the access points to realize the threat. Vulnerabilities not covered by countermeasures are very easily exploited by threat agents to meet their objective of exploiting the system maliciously and fraudulently to their advantage. Responsibility of the developer is to adopt countermeasures to reduce the identified vulnerabilities to the minimum. In this way threat modeling helps in identifying all the vulnerable points in the system that an attacker can exploit to carry out threats and is a base of any secure software system. It also helps the designers anticipate attack goals and determine: 'what' the system is designed to protect; from 'whom' and 'how' it can be done. Microsoft's Threat Modeling Tool supports and facilitates the designers and developers in this respect [5].

Although the above mechanism provides broad view of identification of known threats but security has no meaning unless unknown threats to the system are not identified and taken care of in the security framework of proactive threat management. These unknown threats may be due to exploitation of vulnerabilities left unnoticed during the design phase or adaptive threats by sophisticated hackers who are coming out with new threat perceptions. The strategy for capturing these unknown threats to the system adopted in this proposed model is given below.

3.1.2. Identification of Unknown Threats using Research Honeytokens in Conjunction with Statistical Model:

Research Honeytokens is a new concept to the computer security arena which in fact is most interesting implementation of a honeypot to detect unknown threats. The term honeytoken was first coined by Augusto Paes de Barros in 2003 in a discussion on honeypots. Honeytokens are honeypots that are not physical devices but are a digital entity that would look attractive and useful to an attacker [16]. This concept has been further augmented by Lance Spitzner to proactively gather information about security threats by providing a real system with real applications to adopt remedial measures against them. A research honeytoken is a highly flexible tool with no production value and any interaction with honeytoken means malicious or fraudulent activity. Research honeytokens deployed should be advanced one and updated regularly so that they are capable of attracting more attackers without them knowing it [10, 13]. Here, research honeytokens are being used as a digital entity planted during the design phase of the software life cycle itself to procure information about attacker profiles in respect of unknown threats. A research honeytoken is then closely monitored by human or software agents to identify new or unseen vulnerabilities which are being exploited by the attacker to compromise the security of the system. Further forensic analysis of collected data from honeytokens gives preferred attack patterns against unknown threats.

While research honeytokens represent a powerful tool to identify threats but they have their own limitations. There may be cases when the attacker may not interact with them and may have a little value as observable. Moreover the attackers can also use a research honeytoken maliciously to attack other non - honeytoken systems. This deficiency has been overcome to a certain degree by adopting statistical technique with honeytokens in the proposed model. A statistical based systems use statistical models to detect malicious activity. These systems adapt to different system behaviors or occurrences and try to develop a usage pattern. Predefined variables monitored over a time period are calculated for a test value. If this value is above the user defined threshold, then an alert is triggered. This approach does not require any predefined attack patterns and is capable of detecting new or unknown attacks [15]. Fusion and information correlation of data received from these two sources provides much better detection ability for wide range of threats and attacks at an early stage. Moreover, the normalcy depends on correlations among different parameters. The independent values of two different parameters determined using research honeytokens and statistical model can be taken normal, but their combination can show abnormality [9, 12]. In this way correlating information from multiple sources helps in detecting many potential unknown attacks which can result in security compromise of a system.

3.1.3. Integrated Framework for Identification of all types of Threats for Mitigation

This model presents a new integrated framework to identify both known and unknown threats as shown in Fig. 2 below.

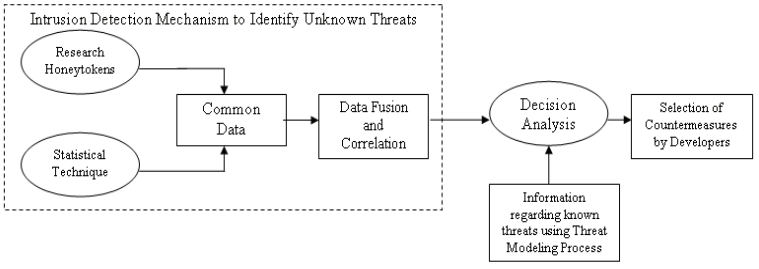

Fig.2. Integrated Framework for Identification of Threats Known and Unknown

The above diagram shows a central common data system which collects data from two sources i.e. research honeytokens and statistical model. Once fused, correlated and analyzed, it indicates new or unknown threats and their mode of attack which is fed as input to decision analysis process. Similarly known threats identified using threat modeling process as shown above are also incorporated as input in decision analysis process which gives threats to be mitigated as output. These threats are then communicated to the developers for selection and adoption of appropriate safeguards as given below.

3.2. Mitigation of Threats using Multi-agent System Planning

In this phase Multi Agent System Planning for Threat Avoidance (MASPTA) has been adopted for averting above identified threats. MASPTA as the name suggests is a system that works in a multi-agent environment for avoiding threats to a web based system where multiple autonomous agents coordinate and communicate with each other to achieve the goal of software security [1]. Threat trees have been created in this phase for above identified threats requiring mitigation to analyze how a threat is manifested through attack paths. In a threat tree, the threat is at the root node which needs to be mitigated and an attack path is a route from a leaf condition to the root threat that an attacker can take to achieve its goal as shown in example threat tree in Fig. 3. Here, AND refinement means all attacks should succeed for the corresponding threat to occur while OR refinement means presence of at least one attack can cause the threat to occur. The attacks at leaf nodes can not be refined further and are executed by the attacker to accomplish the threat. Agents are therefore inducted against the attacks at leaf nodes as shown in the diagram below to prune the attack branches from the threat tree to avoid the threat at the root level. These agents execute their predetermined action plans to avert threats to the system [1].

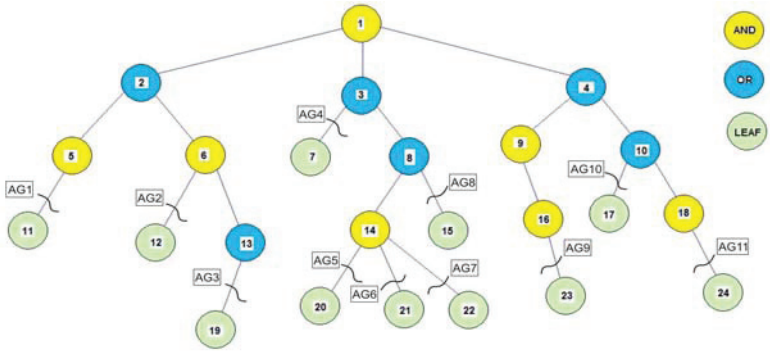

Fig.3. Threat Tree with Agents

3.3. Monitoring and Management of Threats

As explained above, both possible known and unknown threats are identified and appropriate countermeasures are applied to mitigate these threats using multi-agent system planning. Security measures so applied for risk reduction are now evaluated in this phase to assess the effectiveness of the adopted countermeasures using meta-agents in multi-agent environment. Meta–agents constitute a special case of agents and can be used for meta-reasoning i.e. reasoning about reasoning. Their goals are performing reasoning, planning actions and can have a method to evaluate individual agents [6, 7]. These meta-agents communicate with each other and have the ability to work with other agents to achieve a common goal of software security. In the proposed model these meta-agents are superior to the software agents applied in Phase-2 and has been applied to monitor them regarding execution of their specified action plans to avoid threats. They automatically create a security checklist for designers and developers in the light of abnormalities observed in the security measures applied in the step above. Checklist created

by meta-agents identifies the left-over security holes which can be exploited by the attackers to compromise the security of the system and are communicated to the designers. These meta-agents monitor the applied safeguards continuously during execution to update the security checklist. The designers/developers in turn evolve necessary strategies and contingency plans for taking appropriate corrective actions to plug these noticed security holes. These plans may supplement the applied safeguards to ensure security of the software system under development.

4. Adoption of Threat Oriented Security Model in the Spiral Framework

The proposed threat-oriented security model is now inducted in risk analysis segment of the spiral model as shown in Fig. 4. The first circuit around the threat-oriented security model embedded in risk segment of the spiral process has been used for identification of known threats and adoption of necessary countermeasures to mitigate them using multi-agent system. An intrusion detection mechanism consisting of research honeytokens in conjunction with statistical model has also been planted during this circuit only to detect unknown threats.

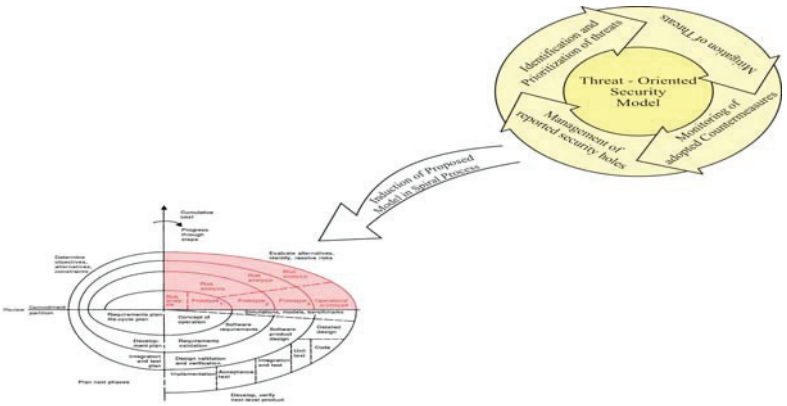

Fig. 4. Induction of Threat Oriented Security Model in the Spiral Framework

Furthermore, meta-agents in this mechanism help to determine if the safeguards already deployed to mitigate the known threats are adequate or not. For instance if a detection mechanism is attacked, it proves that an attacker found a way to this mechanism. With the knowledge of these attacks, it will be easier for the designers to determine and close security holes, as well as adopt additional safeguards to close the exploits to avoid real attacks [14] .This process is then carried out to the next segments of the spiral process for development of the software prototype. This prototype is then deployed for feedback in respect of following security parameters:

- Detection of new or unknown threats which may be due to exploitation of vulnerabilities that have gone unnoticed during the first iteration with the help of planted detection mechanism.
- Performance of the countermeasures applied against identified threats using meta-agents.
- New security features required to be adopted to cater to additional requirements given by the customer after delivery of prototype.

On the basis of above feedback necessary security features may be incorporated by the designers to remove the observed anomalies in the subsequent passes around the spiral process to counter all types of threats known or unknown. In this way every reiteration will result in reduction of vulnerabilities at the design level and progressive risk reduction till it reaches the acceptance level to the satisfaction of customer.

5. Analysis of the Proposed Model

The proposed threat-oriented security model is primarily meant to enhance and strengthen the security of the software systems to meet innovative and sophisticated mode of threats and fraudulent attempts by hackers of present era. Every phase of this proposed model contributes towards enhancement of the security measures deployed at present. A comparative study of the advantages of this proposed model over the traditional security mechanisms is as follows:

- The first phase of the proposed model captures both known and unknown threats as part of proactive threat management whereas in the existing security mechanisms only known threats are taken care of.
- Intelligent system using multi-agent system planning has been adopted for mitigation of all possible threats in the second phase as against traditional security solutions involving artificial intelligence which is not that significant.
- Induction of meta-agents in the third phase is revolutionary in nature and provide security checklist to designers for adoption of necessary security features to make-up for the left over security holes to save the system from being compromised. Use of meta-agents in multi-agent environment in the existing security mechanisms for the purpose is still in its infancy.
- Integration of the proposed model in the risk segment of the spiral model has resulted in significant decrease in design level vulnerabilities and progressively brought the risk reduction to the acceptance level which is our ultimate goal. This step supplements the traditional security process models for secure software engineering.
- Threat Mitigation, Monitoring and Management Plan (TMMP) in the proposed model is adaptive in nature and goes on updating with every subsequent round of spiral process to customer's satisfaction.

Although this integrated approach has many advantages over the traditional security solutions, it has its own limitation as any security hole left uncovered and exploited by malicious users can be problematic.

6. Conclusion and Future Work

Traditional risk management techniques used at present have been found to be quite inadequate as they cater to known or predictable threats only. It has therefore become necessary to go in for an integrated and comprehensive approach in proactive threat management to meet all types of potential threats. This work is an attempt wherein different traditional and innovative techniques detailed above have been fused together to evolve a threat–oriented security model which can counter all types of threats both known and unknown for secure software engineering. As a future work, we are extending this area of research to make this model cost-oriented.

References

1. Punam Bedi, Vandana Gandotra, Archana Singhal, Vandita Vats and Neha Mishra, "Avoiding Threats Using Multi Agent System Planning for Web Based Systems". 1st International conference on Computational Collective Intelligence – Semantic Web, Social Networks and Multiagent Systems, Wroclaw, Poland, October 2009, LNAI, Springer-Verlag Berlin Heidelberg, pp.709-719, 2009.
2. Frank Swiderski and Window Synder, Threat Modeling, Microsoft Press, 2005.
3. Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill, 2005.
4. Gary McGraw, Software Security: Building Security In, Addison-Wesley Software Security Series, 2006.
5. Microsoft ACE Team. Microsoft Threat Analysis and Modeling [EB/OL]. http://msdn.microsoft.com/en-us/security/default.aspx.

6. Esmiralda Moradian and Anne Hakansson, “Approach to solving security Problems Using Meta-Agents in Multi Agent System”, In the Proceedings of 2nd International KMS Symposium on Agents and Multi-Agent Systems: Technologies and Applications , LNAI 4953, pp.122-131, 2008.
7. Esmiralda Moradian and Anne Hakansson “Controlling Security of Software Development with Multi-agent System”, In the Proceedings of 14th Knowledge-Based and Intelligent Information and Engineering Systems, LNCS, Volume 6279/2010, pp. 98-107, 2010.
8. Mehrez Essafi, Lamia Labed Jilani and Henda Hajjami, “Towards a comprehensive View of Secure Software Engineering”, In the Proceedings of International Conference on Emerging Security Information, Systems and Technologies, pp.181-186, 2007.
9. Herve Debar and Andreas Wespi, “Aggregation and Correlation of Intrusion-Detection Alerts”. In Proceedings of Recent Advances in Intrusion Detection (RAID), pp. 87-105, 2001.
10. Lance Spitzner, Honeypots: Tracking Hackers, Addison-Wesley, 2002.
11. Barry W. Boehm, “ A Spiral Model of software Development and Enhancement”, In ACM Computer Journal, volume 21, issue 5, pp. 39-45, 1988.
12. Dipankar Das Gupta, “Immuno-Inspired Autonomic System for Cyber Defense” In Information Security Technical Report, Elsevier , volume 12, issue 4, pp. 235-241, 2007.
13. Lance Spitzner, “Honeypots: Catching the Insider Threat”, In the Proceedings of 19th Annual Computer Security applications Conference, pp. 170, 2003.
14. Arash Barfar and Shahriar Mohammadi, “Honeypots: Intrusion Deception”, The Information Systems Security Association (ISSA Journal), pp. 28-31, 2007.
15. Park Yongro, “A Statistical Process Control Approach for Network Intrusion Detection”, Dissertation, Georgia institute of Technology, 2005.
16. Jonathan White, “Creating Personally Identifiable Honeytokens”, Innovations and Advances in Computer Sciences and Engineering, Springer Science + Business Media, pp. 228-233, 2010.