Automated Penetration Testing Based on a Threat Model

Norah Ahmed Almubairik
Imam Abdulrahman Alfaisal University,
Saudi Arabia
naalmubairik@uod.edu.sa

Gary Wills
University of Southampton
United Kingdom
gbw@ecs.soton.ac.uk

Abstract—**The aim of this work is to propose a systematic penetration testing algorithm guided by a threat model. The use of the threat model in penetration testing ensures that all existing threats are checked and no threat is overlooked through the penetration test process. The objectives of this work are as follows: assembling a package of penetration testing tools (toolkit) to test the security of a system. Moreover, considering standard methodologies to design the automated penetration testing. A number of methodologies have been followed during the design of the algorithm. First, a threat model designed at the IT Innovation Centre was used extract threats. These threats were used as a starting point for the penetration testing. Second, the NIST 800-115 standard for penetration testing was followed. Applying the proposed automated penetration testing algorithm to a real system contributes to the reduction of consequences which can result from malicious attacks.**

Keywords: Automated penetration testing; system model; threat model.

I. INTRODUCTION (HEADING 1)

A penetration test is only one of several security testing types. Wai [8] defines penetration testing as an attacking activity carried out by a trusted person instead of malicious hackers. In addition, Geer and Harthorne [2] define the penetration testing as the ability to illegally gain something through legal authority. Saindane [7] illustrates that penetration testing is an essential part of the risk assessment strategy of an organisation.

The threat model entails all malicious activities that could be done by an attacker. Myagmar et al. [6] justify that a threat model is a backbone for security requirements. In other words, modelling potential threats, which reduce the value of an organisation's assets, from an adversary's point of view is essential in order to address the security challenges during system design phase, Xu et al. [9] demonstrate how automated security testing with the respect to a threat model is operational and beneficial.

II. PROBLEMS AND RELATED RESEARCH

There are multiple issues related to the manual penetration testing procedure. First, the penetration testing team reasonably examines the security of a system based on their knowledge and expertise, but they might not check all existing threats. On the other hand, a single ignored threat can compromise an entire system and, eventually, lead to unfavourable situations (i.e. identity theft, information exposure, stealing customers' banking account details, etc.). That is to say, certification does ensure that penetration testers have the basic minimum assumptions of skills and attitude, but does not ensure professionalism [3]. Consequently, with manual penetration testing, there is probability of leaving a significant portion of attack space uninvestigated.

Another issue related to the manual penetration testing process is that testing all variations of threats is time-consuming and exhausting for the penetration testers. Thus, the process is expensive as it is labour-intensive work in nature.

In penetration testing, there is currently a limited awareness of the automated penetration testing procedure. The meaning of automated penetration testing is explained by Chapple [1], where the security experts carry out a careful attack on systems and applications to be tested. Hope and Walther (2016) observe the apparent lack of knowledge and resources in automating the methodologies of security testing. Also, Xu et al. [9] emphasise that a whole or partial automation of the security testing procedure is far preferable. Therefore, the goal of this work is to introduce an automated penetration testing algorithm based on a threat model.

III. METHODS

This research aims to propose an algorithm to systematically generate a penetration testing plan guided by a threat model. To achieve this goal, number of methodologies has been followed.

First, a threat model designed at the IT Innovation Centre was used to model systems, highlight potential threats, and apply optional control strategies. These threats were used as a starting point for the penetration testing (see Figure 1).

Then, to automate the penetration testing process based on a threat model, the NIST 800-115 penetration testing technique, is utilised. This automated penetration testing consists of seven different phases: planning, system modelling, threats extraction using the threat model, vulnerability discovery, threat and vulnerability mapping, attack and reporting (see Figure 2).

The IT Innovation Centre is an applied research center and part of the Electronic and Computer Science College at the University of Southampton that focuses on designing, developing, and engineering technological innovations: http://www.itinnovation.soton.ac.uk/

Figure 1. Modelling a system and extracting threats using the threat model

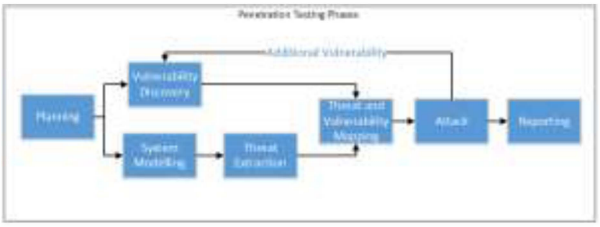

Figure 2.Penetration testing phases

Then, an algorithm has been devised according to the flow of testing phases. The algorithm works as a road map for penetration testers to properly conduct the testing driven by the threat model. In this work, the problem needed to be solved is a graph problem. The graph is set of points called "nodes" and some of them are linked by lines called "edges" [4]. This graph problem matches the automated penetration testing problem where the system under test consists of assets (nodes). These assets are linked through different relations: uses, controls, connect, and hosts. These relations are the edges of the graph problem. So, the algorithm has to run over a graph consist of nodes and edges and, thus provide a complete penetration testing.

To verify that the automated penetration testing algorithm is feasible in a real-world scenario, Face Validity measure was used. It checks if a test and its components seem valid and useful to people taking this test. In other words, does the algorithm will measure what it is supposed to measure or not.

IV. Findings And Argument

The algorithm was successfully designed to evaluate the tested system's immunity against malicious attacks. The algorithm was written for human reading rather than for machine reading; it was also written using mathematical notations. The algorithm was then tested to ensure its validity.

The algorithm's simplicity depends on the size of a system model. The more threats involve in the model; the more testing is needed.

Applying this algorithm on a real-life system has several advantages. The algorithm will enable businesses to take security countermeasures in order to minimise the impact of sensitive data exposures; it also encourages organisations to comply with legal regulations where the security of electronic records is obligatory. Finally, it ensures business continuity.

V. Conclusion

This work proposes a threat model driven approach for automated penetration testing. Automating the process of testing helps security experts to avoid overlooking threats; automation also allows testers employ their expertise in the most valuable part [1]. Given that the penetration testing is a time-consuming and exhausting process, the automation, will relieve the tester from monotonous tasks and he can use his expertise in the most challenging parts.

Despite recent efforts to automate the penetration testing process, further research is required to cover all threats that are reported by the threat model, including SQL threats, cross-site scripting, and mail threats, among others. Covering these remaining threats will add to the test's completeness.

References

[1] Chapple, M. (2014) An intro to automated penetration testing. Available at: http://searchsecurity.techtarget.com/tip/An-intro-to-automated-penetration-testing.

[2] Geer, D. and Harthorne, J. (2002) 'Penetration Testing: A Duet', IEEE.

[3] (ISC)2 Government Advisory Council Executive Writers Bureau (2013) Penetration testing: Pros and cons of attacking your own network. Available at: https://gcn.com/articles/2013/02/04/pros-cons-penetration-testing.aspx.

[4] Levitin, A. (2011) Introduction to the design and analysis of Algorithms. 2nd edn. Boston: Pearson Addison-Wesley.

[5] Microsoft (2016) Common types of network attacks. Available at: https://technet.microsoft.com/en-us/library/cc959354.aspx (Accessed: 16 December 2016).

[6] Myagmar, S., Lee, S. and Yurcik, W. (2005) 'Threat Modeling as a Basis for security requirement'.

[7] Saindane, M. (2012) 'Penetration Testing – A systematic Approach'

[8] Wai, C. (2002) Auditing & Assessment. Available at: https://www.sans.org/reading-room/whitepapers/auditing/conducting-penetration-test-organization-67 (Accessed: 20 December 2016).

[9] Xu, D., Manghui, T., Sanford, M., Thomas, L., Woodraska, D. and Xu, W. (2012) 'Automated Security Test Generation with Formal Threat Model', IEEE Transactions on Dependable and Secure Computing, 9(4).