Technische Universität München

Automated Attack Planning using a Partially Observable Model for Penetration Testing of Industrial Control Systems

Master Thesis

Supervisors: Prof. Dr. Alexander Pretschner, Alei Salem
Email: alexander.pretschner, salem @ in.tum.de
Phone: +49 89 289 – 17, 314
Starting date: immediately

Fakultät für Informatik
Lehrstuhl 22
Software Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3 85748
Garching bei München

Tel: +49 89 289 17885,
+49 89 289 17314

Web: http://www22.in.tum.de

Context

Industrial control systems govern important industrial processes as well as many other areas of daily life including building automation and the energy infrastructure. In the past, such systems were rarely networked if at all. Recently, however, a need for a higher degree of communication between different systems as well as other company resources has arisen. While leading to increased productivity and the development of new processes altogether, this also introduced a whole new class of possibles attack vectors. As such systems are usually optimized for their specific purpose and otherwise offer limited computational resources, they often lack security mechanisms found elsewhere. This makes industrial control systems a very interesting and promising target for malicious intruders trying to cause damage on often critical and costly infrastructure. Consequently, there is a desparate need to continuously ensure that ICS are free of devastating vulnerabilities.

Penetration testing, also known as *Pentesting*, is a classical, proactive security testing methodology that strives to detect and alleviate existing security vulnerabilities. Although some simple vulnerabilities can be detected somewhat automatically e.g. by common using static code analysis tools, vulnerability scanners, and so forth, identifying more complex–and often more devastating–vulnerabilities require the expertise and skillsets of professionals. This semi-manual process is, hence, both time-consuming and subjective, due to the reliance on human capabilities. In fact, pentesting is widely adopted as an artform, rather than a systematic process [3][1]. To address those two issues, automated pentesting is regarded as a method that is time-efficient and systematic.

Planning an attack is a fundamental aspect of pentesting, especially within the automated setting. Failure to come up with a reliable attack plan undermines the effectiveness and usefulness of the entire testing process. Planning commences by specifying the attackers access to the target system. The two extreme settings are black- and white-box testing. If the attacker knows little or no information about the system, black-box testing is adopted, whereas if they have full access to the system, the white testing counterpart is considered. In this work, we adopt the more realistic gray-box testing, which simulates an internal attacker with partial knowledge of the target system. The primary hypothesis of this work is that given partial information about the system, we can utilize approaches such as the *Partial Obervable Canadian Hacker Problem (PO-CHP)* [2] to generate technically-sound and time-saving penetration testing plans that target a valuable asset within the network e.g. a PLC unit within a powerplant.

Goal

As mentioned earlier, in this thesis, we make the following assumptions about available information and system behavior. Firstly, we assume that the attacker possesses complete knowledge of the network layout. Secondly, we assume that the attacker has primitive information about the static configuration of the nodes in the network e.g. whether a node has been previously compromised, et cetera. Thirdly, we assume that a previously compromised asset can not be lost and actions do not have any unintended side-effects e.g. applying an exploid causing a machine crash. Under those three assumptions, the goal of

this thesis is to develop a tool that computes an attack plan that orders possible attacks in such a way that a specific target host can be compromised incurring the lowest cost possible according to some metric e.g. time required for the attack to succeed, the probability of a successful attack, and so forth.

In order to calculate the cost of exploiting a given node, we adopt a Bayesian approach that continuously updates our initial/prior belief in exploiting a node according to information acquired from common vulnerability databases. To come up with prior beliefs, we utilize scanning tools e.g. *NMAP* and statistical information about the exploitability of identified systems. As for explotation likelihoods, we consult vulnerability databases such as *National Vulnerability Database (NVD)*.

The contributions of this thesis are (a) generate reliable testing plans that guide pentesters/automated explotation tools through the pentesting process, and (b) calculate the least expensive method to exploit a valuable asset/goal, which is not offered by any tool as per our literature review. Having said that, we plan to evaluate our tool according to three metrics viz., time consumption, resource utilization e.g. processing power, RAM space, disk utilization, et cetera, and–most importantly–technical soundness of the generated plans. For the first two metrics, we plan to measure the time taken and resource utilized to generate the testing plan, and compare such figures against similar tools. As for the soundness, due to lack of tools/platforms that deliver the proposed functionality, we plan to utilize human expertise to assess the soundness of the plans. For future works, we can augment the generated plan to automated exploitation tools and compare the expected and actual success figures. Lastly, the sample networks on which we evaluate our approach and tool comprise simulated and real-world ICS networks provided by IABG.

Work-plan

1. Familiarization with industrial control systems and the security thereof.
2. Research into and implementation of the automated planning technique.
    a. Familiarization with the current state of research.
    b. Identify methods to assess the probability of exploiting a node.
        - Identify host configuration e.g. using NMAP, Nessus, etc.
        - Infer prior belief in exploitability.
        - Consult vulnerability databases using the configurations.
        - Infer likelihood of successful attack.
        - Update exploitability belief and cost.
    c. Identifying the best algorithms for implementing the proposed approach.
        - Find graph traversal and shortest path algorithms.
    d. Finalization of the design and implementation decisions of the tool.
        - Design tool building blocks.
        - Choose a programming language.
        - Ensure reliable logging of utilized resources for evaluation.
    e. Implementation of the proposed approaches.
3. Evaluation of the developed techniques.
    a. Setting up the testing environment.
    b. Testing the sample systems using the developed tool.
    c. Compare time and resource utilization against similar tools.
    d. Assess the technical soundness of the generated testing plans.
4. Writing of the final thesis containing:

a. Description of the problem and motivation.
b. State of the art survey of automatic planning of penetration testing as well as applicable artificial intelligence techniques.
c. Rationale for using the selected techniques.
d. Implementation description.
e. Evaluation.
f. Conclusion and further work.

Deliverables

- Final thesis report written in conformance with TUM guidelines.

References

[1] D. Geer and J. Harthorne. Penetration testing: a duet. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 185–195, 2002.

[2] Jörg Hoffmann. Simulated penetration testing: From" dijkstra" to" turing test++". In *ICAPS*, pages 364–372, 2015.

[3] S. Robinson. The art of penetration testing. In *Security of Distributed Control Systems, 2005. The IEE Seminar on*, pages 71–76, Nov 2005.