UDC 004.891, 004.492 DOI: 10.30748/soi.2017.148.25

Y.Y.Stefinko, A.Z.Piskuzub

National University "Lviv Polytechnic", Lviv

THEORY OF MODERN PENETRATION TESTING EXPERT SYSTEM

The paper considers the models and algorithms of intelligent components intended for active analyzing. The offered approach is based on simulation of computer attacks by using attack graphs, logical trees and solving the probabilistic questions. This article discusses developing of the security expert system by using new methods for efficient and automated penetration testing. Methodologies of automated planning and partially observable Markov decision processes are suggested in order to improve results and build flexible security evaluation expert system.

Keywords: *penetration testing, ethical hacking, artificial intelligence, expert system, planner, automation, attack graphs, pentesting system, algorithm, modelling, POMDP.*

Introduction

After beginning of the new computerized century people faced many issues with information security as well as with financial loss or politic scandals after computer hacker's intrusion. Judging from the global statistics, the main reason is a low security level of majority of systems connected to the Internet. The most common vulnerabilities exist in operating systems (OS) and application configurations, user management and administration, incorrect physical access policy and improper access control settings, existence of vulnerable or easily exploited services and malicious software. Therefore now vulnerability detection and estimation of security level of computer networks are actual tasks of information security.

Using passive security strategy (firewalls, antiviruses etc.) often became ineffective, so that more proactive techniques are growing during the last time. Ethical hacker's community has been steadily growing since BackTrack OS was introduced at 2006 and a lot of software and tools are developed or improved [1]. Those tools help to simulate real penetration testing (pentesting, ethical hacking) surely only after official agreements between customer and ethical hacker – called Rules of Engagement (ROE), according to PCI DSS and OSSTMM methodologies [1].

Security experts have spent a lot of time for investigation in ethical hacking sphere for last years in order to find out the best approach to defend their systems and proactively react on attacks. Pentesting process in general can be divided into 10 well-known phases from Target Scoping to Reporting [1]. Every step has different goal, so that not all of them can be automated. However this research will discuss all possible approaches to avoid manual human work.

For the security evaluation and attack execution famous ethical hacking system was developed - Kali Linux (BackTrack descendant). It include stack of known network and security tools. Many of them are open-source, however you need to be experienced security and network expert to perform full penetration testing using this OS. For successful pentesting we also still need strong development and even tester skills. Also many tasks should be done manually by advanced scripting and still we expect high attention from the advanced pentester. It becomes ineffective to use manual solutions during big pentesting process which include many phases of security evaluation. Human errors are still possible and many companies are searching for automated software to be able to perform those tests as often as needed. So automation as well as pentesting system preparation (installation, deployment etc.) are those issues which need deeper research and improvements nowadays [3].

In artificial intelligence (AI), an expert system is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning about knowledge, represented mainly as *if–then* rules rather than through conventional procedural code. The first expert systems were created in the 1970s. Expert systems were among the first truly successful forms of artificial intelligence software.

An expert system is divided into two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities. Figure 2 representing the general structure of the expert system and consider the main components.

In this article known algorithms are examined, building attack trees and graphs, automation and improvements for penetration testing are discussed in order to speed-up the testing process and highlight the main problematic parts. Suggested algorithm in the output should include the best from well-known methodologies and try to resolve issue with multi-attack execution.

Related work

In the latest research [4] C.Sarraute mentioned Markov decision process (MDP) as a way of improvement for penetration testing planning and automated execution. So building success algorithm is one of the most important parts of planning automated pentesting. As we know not all attacks can lead to successful results, so that many papers were written to research problem of penetration testing planning. Also security experts investigated security issues in the cloud environment in [5]. It was proved by security experts that it's necessary to implement automation for all existing pentesting approaches and improve detection of possibility to compromise the attacked target OS by executing specific attack chains. Decision making problem was discussed in few research works and thesis [2] by J.Hoffman. Artificial intelligence usage, or in other words, full automation of penetration testing still can be ineffective due to problems with creativity, intuition and question the assumptions, which human mind can solve even in the most complex situations [6]. For example, during entire pentest modelling, the issue with computational complexity theory can appear due to planning attack graphs complexity as it needs to resolve PSPACE problem [6]. In computational complexity theory, PSPACE is the set of all decision problems that can be solved by a Turing machine using a polynomial amount of space [2].

By using suggested pentesting model during real attack simulation we can improve efficiency of pentest and find the smallest vulnerabilities in target systems. For sure, most of them can't be detected by simple manual pentesting, but it needs much more time than with automated artificial intelligence mechanisms.

Modeling penetration testing

Industrial ready penetration testing systems are not simple and transparent as security analytics can imagine. Firstly, mathematical and theoretical basis should include various graphs and models for real life simulation of the potential attacker.

Therefore, correlation and aggregation of security events is needed both for proactive security, which deals with preventing possible attacks, and for reactive security, which deals with detecting actual intrusions.

In previous researches we looked on penetration testing as on process which can be done strictly in one-by-one steps consequence without carrying about preparation and planning. This wasn't efficient and real demonstration of ethical hacking. We should think that hacker can be smarter and execute few attack in one moment or use one compromised host in creative manner for getting to the next vulnerable point in the target system. Therefore, vulnerabilities are not only exploited in isolation. Once an attacker has found a way in a network she, as a rational being, will try to maximize her return on investment, as reported by the media and reflected in several research works [9, 10]. Furthermore, network administrators have to manage not only isolated vulnerabilities but the risk of vulnerabilities composed in multi-step attacks. Not only vulnerable hosts can be used to compose attack steps. Having access to a credential may allow attackers to also exploit non-vulnerable hosts. All in all, solving every particular vulnerability, for example by patching, deactivating services and hosts, is never a solution for multi-step attacks.

Finding steps of possible attacks turns out to be a rather challenging problem due to the complexity and size of networks, and the high number of possible combinations among steps which represent opportunities for potential attackers. To address the problem of proactively finding possible multi-step attacks in networks, Franqueira's thesis [7] contains the MsAMS (Multi-step Attack Modeling and Simulation) solution: an approach that uses a variation of Mobile Ambients as modeling paradigm, and Heuristic Search as simulation paradigm, supported by Link Analysis Ranking algorithms as a source of metrics. As an evidence of feasibility of this approach in networks of realistic size, this thesis introduces the MsAMS proof-of-concept tool, and shows that this is scalable to realistic networks.

From the attack execution point we can see many possible ways for compromising the target (Fig.1). For example, attacker can use hybrid approach including social engineering and professional hacker's tool like Metasploit or sqlmap.

Fig. 1 example graph shows three attack scenarios for how a user's bank account credentials can become compromised. The leftmost subgraph is a keylogger attack, the middle subgraph is a SSL spoof attack, and the right subgraph is a phishing attack. So building specific attack graphs is one of the components of the successful pentesting system.

Building algorithm for pentesting expert system

On the other hand, automated network penetration testing, an apriori completely unrelated area to critical constrained planning, will constitute the second major part of work. Due to the growing size of today's networks, it is getting harder (for large company networks even impossible) to identify security threats by hand. Rather, companies are using automated, and semi automated tools to analyse vulnerabilities of their networks. Using planning to simulate attacks to networks is a very promising future direction in (semi-) automated network security testing. A company called Core Security is already commercially using a planner inside one of their tools to generate possible attack plans that are provided to a human security officer to guide the attention to particular, possible security threatening, regions of the network. One problem of this approach is that it requires a global and exact model of the network, including all the network host configurations. In practice, this is clearly impossible to get and to maintain. Ideally, the model should start with minimal possible knowledge about the network and host configurations.

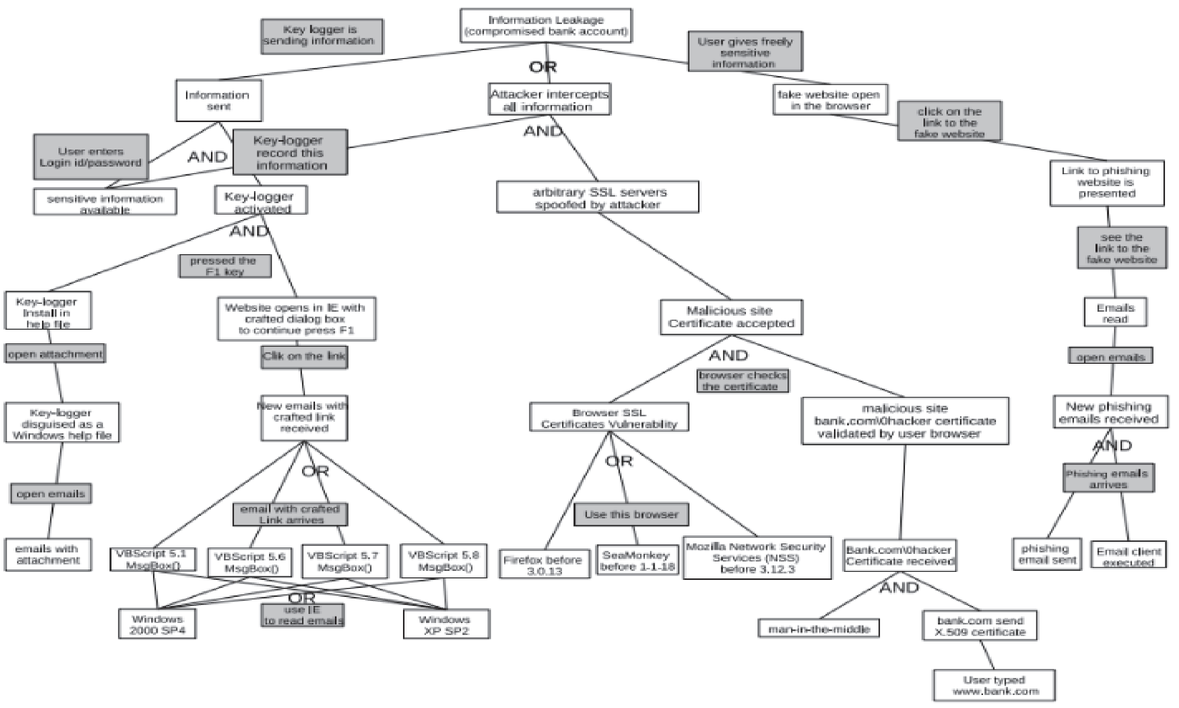
Information Leakage (compromised bank account)
OR
Key logger is sending information
User gives freely sensitive information
Information sent
Attacker intercepts all information
fake website open in the browser
click on the link to the fake website
User enters Login id/password
AND
Key-logger record this information
AND
sensitive information available
Key-logger activated
arbitrary SSL servers spoofed by attacker
Link to phishing website is presented
AND
pressed the F1 key
see the link to the fake website
Key-logger Install in help file
Website opens in IE with crafted dialog box to continue press F1
Malicious site Certificate accepted
Emails read
open attachment
Clik on the link
AND
open emails
browser checks the certificate
Key-logger disguised as a Windows help file
New emails with crafted link received
Browser SSL Certificates Vulnerablity
malicious site bank.com\0hacker certificate validated by user browser
New phishing emails received
OR
OR
AND
open emails
email with crafted Link arrives
Use this browser
Phishing emails arrives
emails with attachment
VBScript 5.1 MsgBox()
VBScript 5.6 MsgBox()
VBScript 5.7 MsgBox()
VBScript 5.8 MsgBox()
Firefox before 3.0.13
SeaMonkey before 1-1-18
Mozilla Network Security Services (NSS) before 3.12.3
Bank.com\0hacker Certificate received
phishing email sent
Email client executed
OR
AND
Windows 2000 SP4
use IE to read emails
Windows XP SP2
man-in-the-middle
bank.com send X.509 certificate
User typed www.bank.com

Fig. 1. An instantiated attack graph augmented with user actions

This idea was followed by Sarraute et al. in their design of network penetration testing as solving partially observable Markov decision processes (POMDPs) [4]. Unfortunately, solving such POMDP models is only feasible for a very small number of hosts, and thus does not scale to real world networks. Hoffmann's work was continued on finding feasible, efficient, yet realistic models of penetration testing [2]. One very typical aspect of probabilistic models of penetration testing is that the probability of reaching a goal state will be low. In other words, the probability of reaching dead ends will be very high. So, dead end detection will play an important role to solve penetration testing problems efficiently, as well.

During research the idea of modelling the problem in terms of POMDP was tested. This grounds penetration testing in a well-researched mathematical formalism, highlighting important aspects of this problem's nature. POMDPs allow modelling information gathering as an integral part of the problem, thus providing for the first time a means to intelligently mix scanning actions with actual exploits.

Planning based on Markov Decision Processes (MDPs) is designed to deal with non-determinism, probabilities, partial observability, and extended goals. It is based on the following conventions:

- A planning domain is modelled as a *stochastic* system, that is a nondeterministic state-transition system that assigns probabilities to state transitions.
- Goals are represented by means of *utility functions*, numeric functions that give preferences to states to be traversed and/or actions to be performed. Utility functions can express preferences on the entire execution path of a plan, rather than just desired final states.
- Plans are represented as *policies* that specify the action to execute in each belief state.
- The planning problem is seen as an *optimization problem*, in which planning algorithms search for a plan that maximizes the utility function.
- Partial observability is modelled by observations that return a probability distribution over the state space, called *belief states*.

A Markov Decision Process (MDP), also called stochastic system [4], is a nondeterministic state-transition system with a probability distribution on each state transition. It is defined by a tuple:

$$\Sigma = \langle S, A, T \rangle, \tag{1}$$

where:

- The state space S is a finite set of states.
- The action space A is a finite set of actions.
- $T : S \times A \to \Pi(S)$ is the state-transition function, giving for each world state *S* and agent action A, a

probability distribution over world states. We write T (s, a, s') for the probability of ending in state s' given that the agent starts in state S and executes action a.

- $r: S \times A \rightarrow R$ is the reward function, giving the expected immediate reward gained by the agent for taking action a in state *s*.

Policy executions correspond to infinite sequence of states, called histories, which are Markov Chains. Given a policy, we can compute the probability of a history. Let π be a policy and $h = \langle s_0, s_1, s_2 ... \rangle$ be a history. The probability of h induced by π is the product of all transition probabilities induced by the policy [4]:

$$\Pr(h \mid \pi) = \prod_{i \geq 0} T(s_i, \pi(s_i), s_{i+1}). \quad (2)$$

We call $E_{\pi^*}(s)$ the optimal reward in state s.

From the Bellman Equation we have that:

$$E_{\pi^*}(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') E_{\pi^*}(s') \right\} \quad (3)$$

As penetration testing is about acting under partial observability, POMDPs are a natural candidate to model this particular problem. They allow to model the problem of knowledge acquisition and to account for probabilistic information, e.g., the fact that certain configurations or vulnerabilities are more frequent than others. In comparison, classical planning approaches assume that the whole network configuration is known, so that no exploration is required. The present section discusses how to formalize penetration testing using POMDPs.

As we shall see, the uncertainty is located essentially in the initial belief state. This is different from modeling the uncertainty in pentesting using probabilistic action outcomes, which does not account for the real dynamics of the system. Also, as indicated previously, unlike our POMDPs, the approach of Sarraute only chooses exploits, assuming a naive a priori knowledge acquisition and thus ignoring the interaction between these two [4].

Generating a POMDP model for pentesting requires knowledge about possible states, actions, and observations, plus the reward function and the initial belief state. Note first that the POMDP model may evolve from one pentest to the next due to new applications, exploits or tests. Action and observation models for the various possible tests and exploits can be derived from the documentation of testing tools (see, e.g., nmap's manpage) and databases such as CVE (Common Vulnerabilities and Exposures). Information could presumably be automatically extracted from such databases, which are already very structured. In our experiments, we start from a proprietary database of Core Security Technologies. The two remaining components of the model – the reward function and the initial belief state- involve quantitative information which is more difficult to acquire. In our experiments, this information is estimated based on expert knowledge or experience.

The Planning Domain Definition Language (PDDL) is the predominant language for specifying planning tasks in the research area of Automated Planning, which focuses on the actual development of planning systems.

The PDDL description language serves as the bridge between the pentesting tool and the planner. Since exploits have strict platform and connectivity requirements, failing to accurately express those requirements in the PDDL model would result in plans that cannot be executed against real networks. This forces our PDDL representation of the attack planning problem to be quite verbose.

For this reason, PDDL strives for a compromise between the expressive power required for practical applications and limitations that keep a solution of the planning problem still achievable. The PDDL description is given as input to the planner. The advantage of using the PDDL language is that we can experiment with different planners and determine which best fits our particular problem. In recent research works experts evaluated similar pentesting model using both SGPlan and Metric-FF planners [2]. Planner can be integrated in famous pentesting frameworks like Core Impact or Metasploit, which allows executing and validating the resulting plans against a test bench of scenarios.

Finally, the following is an example of an action: an exploit that will attempt to install an agent on target host t from an agent previously installed on the source host s. To be successful, this exploit requires that the target runs a specific OS, has the service ovtrcd running and is listening on port 5053.

Some typical rule based expert system was examined on fig.2. In addition to the rule base, the expert system must give a helpful user interface, a facility for the user to question the program, a technique of learning from experience and an ability to give a reasoned explanation for conclusions that have been reached.

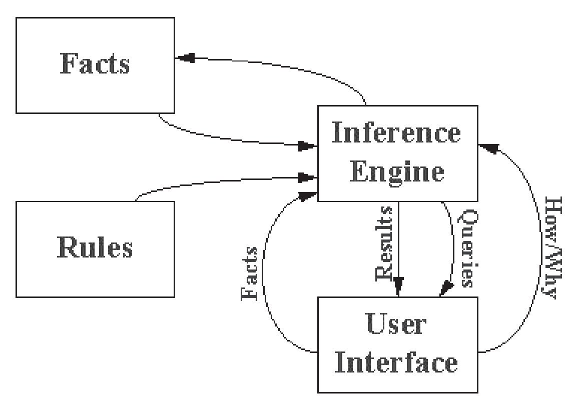

Fig. 2. Structure of the typical expert system

In theory, penetration testing (or security) expert systems should consist of the same main parts. Facts will be collected from the public vulnerability databases and stored in database like MySQL. All computation related to the conclusions will be pefrormed in the interference engine which can be written on python and bash programming language. Graph teory and calculation with POMDP should be also included it in for automated and for hightening results precision.

Conclusion & future work

This research proved that penetration testing is the sophisticated process which has many issues with automation and building strong algorithm. However, building attacks graphs and modeling penetration testing by using POMDP are useful for building efficient and modern penetration testing expert system.This system will kepp all knowedge in the database after generation attcks trees. All gathered knowledge will be used then by the inference engine for execution of the pentest.

One of the main disadvantages of Markov decision process integration is that POMDP is incompatible with scaling target systems for large network infrastructure. So scaling should be investigated in future.

PDDL example was suggested to describe how we can implement this approach with special exploitation software (e.g. Metasploit), in order to automate attacks. As we see planning techniques are very perspective for pentesting future and this area looks very promising. All of the solutions are ready for cloud integration, build and deploy, hence in future the own pentesting expert system for ethical hacking should be deployed.

From the received results we can assume that it is possible in theory to build such sophisticated expert system for security purposes. In future research the main task will be development of the user interface and interference engine which should be able to "learn" and then provide some results to the security team about target examined system. For the next papers we would like to investigate deeper to the building of the own pentesting automated expert system which can include mentioned algorithms and graphs in one semi-automated penetration testing phase.

References

1. *Stefinko Y. Manual and automated penetration testing. Benefits and drawbacks. Modern tendency / Y. Stefinko, A. Piskozub, R. Banakh // TCSET 2016 Proceedings, Lviv, Ukraine, 2016. – P. 488-491. doi: 10.1109/TCSET.2016.7452095.*
2. *Hoffman J. "Simulated Penetration Testing: From "Dijkstra" to "Turing Test++"", ICAPS 2015 Proceedings. Published by The AAAI Press, Palo Alto, CA, 2015.*
3. *Stefinko Y. "Efficient and automated pentesting by using docker in cloud",VI Inter University Conference of Students,PhD Students and Young Scientists "Engineer of XXI Century, Monograph***,** *Bielsko-Biala:ATH, 2016, pp.383–393.*
4. *Sarraute C. "Automated attack planning"***,** *Ph.D. thesis, School of Engineering, Buenos Aires, Argentina, July 2nd, 2012.*
5. *Fernandes D.A.B., Soares L.F.B., Gomes, J.V. et al.: Security Issues in Cloud Environments - A Survey: Int. J. Inf. Secur. (2014), Issue 2, pp 113–170. doi:10.1007/s10207-013-0208-7*
6. *Sheyner O.M. "Scenario Graphs and Attack Graphs", Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, April 14, 2004.*
7. *Franqueira V.N.L. "Finding multi-step attacks in computer networks using heuristic search and mobile ambients" – Dissertation to obtain the degree of doctor at the University of Twente - Enschede, The Netherlands, 2009. - http://dx.doi.org/10.3990/1.9789036529235.*

Надійшла до редколегії 10.01.2017

Рецензент: д-р техн. наук проф. В.В. Хома, Національний університет «Львівська політехніка», Львів.

СУЧАСНА ТЕОРІЯ ЕКСПЕРТНИХ СИСТЕМ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Я.Я. Стефінко, А.З. Піскозуб

У статті розглядаються моделі і алгоритми інтелектуальних компонентів, призначених для активного аналізу. Запропонований теоретичний підхід заснований на моделюванні комп'ютерних атак з використанням графів атак, логічних дерев і вирішення імовірнісних питань. Обговорюється розробка експертної системи безпеки з використанням сучасних методів ефективного і автоматизованого тестування на проникнення. Методології автоматизованого планування і процеси прийняття рішень Маркова запропоновані для того, щоб поліпшити результати і побудувати гнучку експертну систему оцінки безпеки.

Ключові слова*: тестування на проникнення, етичне хакерство, штучний інтелект, експертна система, планувальник, автоматизація, графи атак, алгоритм, моделювання, POMDP.*

СОВРЕМЕННАЯ ТЕОРИЯ ЭКСПЕРТНЫХ СИСТЕМ ТЕСТИРОВАНИЯ НА ПРОНИКНОВЕНИЕ

Я.Я. Стефинко, А.З. Пискозуб

В статье рассматриваются модели и алгоритмы интеллектуальных компонентов, предназначенных для активного анализа. Предлагаемый теоретический подход основан на моделировании компьютерных атак с использованием графов атак, логических деревьев и решения вероятностных вопросов. Обсуждается разработка экспертной системы безопасности с использованием современних методов эффективного и автоматизированного тестирования на проникновение. Методологии автоматизированного планирования и процессов принятия решений Маркова предложены для того, чтобы улучшить результаты и построить гибкую экспертную систему оценки безопасности.

Ключевые слова*: тестирование на проникновение, этичное хакерство, искусственный интеллект, экспертная система, планировщик, автоматизации, графов атак, алгоритм, моделирование, POMDP.*