Hacking Goals: A Goal-Centric Attack Classification Framework

Francesco Caturano, Gaetano Perrone, Simon Pietro Romano

Hacking Goals: a goal-centric attack classification framework

F. Caturano[1], G. Perrone[1], and S. P. Romano[1]

University of Napoli Federico II, Department of Electrical Engineering and Information Technology, Via Claudio 21, 80125 Napoli, Italy
{francesco.caturano,gaetano.perrone, spromano}@unina.it

Abstract. Attack classification does represent a crucial activity in different security areas. During security assessment, it makes it easier to define which attacks must be performed. When conducting threat modeling activities, it simplifies the definition of attack graphs. Many works have addressed the attack taxonomy problem, by introducing different ways to classify attacks. However, these classifications are centered around vulnerabilities and have all been designed from the point of view of those defending a system. Nowadays, companies have a growing interest in Penetration Testing activities, as they have proven effective in detecting vulnerabilities. Penetration testers perform their activity by focusing on goals rather than attack types. In this paper we introduce a "goal-centric" methodology to classify attacks in terms of Hacking Goals.

1 Introduction

Attack Classification provides an important contribution in different security fields. According to [6], an attack classification approach can be leveraged to build secure systems, to identify vulnerabilities for which security defenses do not yet exist, to provide a uniform language for reporting incidents to response teams. These are all defense perspectives. There is also an offensive perspective that is used to detect vulnerabilities by simulating malicious activities. Such activities follow known methodologies, such as those mentioned in [4]. However, in the literature there are just a few contributions that try and formalize these methodologies.

This article proposes an attacker-centric methodology for attacks classification.

2 Setting the scenario

Penetration Testing (PT) is the process of finding IT security vulnerabilities in a system, by emulating the behaviour of a malicious attacker. In black box PT, the team has no information about the target, and tries to sneak into the system by exploiting vulnerabilities. An introduction to PT tasks and tools can be found in [10]. Different phases can be identified:

- *Information Gathering*: in this phase the attacker finds publicly available information that can be used in subsequent phases, e.g., domain names, subnets owned by the target organization, systems that appear to be 'alive' in the network.
- *Scanning*: in this phase the attacker detects running TCP and UDP services exposed by the target hosts.
- *Enumeration*: in this phase the attacker enumerates running services. The goal here is to detect versions of running services and look for potential vulnerabilities;
- *Exploit*: when the attacker has detected vulnerabilities in the system, she/he tries to exploit them and get inside the target;
- *Post-Exploitation*: the attacker tries to obtain higher privileges and persistence inside hacked systems, and performs "lateral movement" activities to gain access to other internal systems.

Final deliverable of a PT activity is a detailed report, containing an executive summary, i.e., a synthesis of detected vulnerabilities, ordered by risk level.

3 Related Work

Many authors have defined methodologies to classify attacks in computer systems. V. M. Igure and R. D. Williams [6] give a formal definition of attack taxonomies and offer a complete overview of the existing ones. Authors suggest to create a layered taxonomy in order to provide an objective methodology to identify vulnerabilities. This is the most important hacking goal classification feature, as by using a goal-centric attack classification you need to focus on hacking goals dependencies. more intricate than that.

Common Attack Platform and Enumeration (CAPEC) [1] is a community resource for identifying and understanding attacks. It offers a search engine that allows users to search for specific attacks. The classification is very useful because it reports description and relationships between attacks. It describes prerequisites to perform an attack. CAPEC classifies attacks by using a target-centric approach, as some prerequisites depend on the target. When using an approach focused on hacking goals, prerequisites are instead "attacker-centric".

Kotenko and Doynikova [9] have created a generator of attack scenarios for network security evaluation. This is of interest to us, since a goal-centric classification allows simplifying the realization of attack graphs, while also defining a test result evaluation methodology.

Different authors have explored security testing by leveraging planning models. Obes et al. [7] show how is it possible to create a PDDL (Planning Domain Definition Language) representation of an attack model. PDDL contains interesting properties such as domain definition, action definition, preconditions required to perform an action and output of an action. Goal-centric classification can be used to define a hacking methodology, so it has a wider scope when compared to PDDL.

4 Hacking Goal

In this section we provide a formal definition of the proposed Hacking Goal classification, as illustrated in Fig. 1.

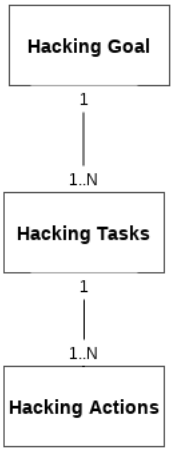

Fig. 1. Hacking Goal, Hacking Tasks and Hacking Actions relationships

A *Hacking Goal* is a macro objective that the attacker is going to achieve. An attacker performs different *Hacking Tasks* to fulfil her/his final goal. Depending on the chosen goal, the related hacking tasks metrics can change. A *Hacking Action* is a single action that an attacker executes while performing a specific hacking task. When the attacker performs Hacking Actions, she/he acquires knowledge about the target environment. For instance, when the attacker makes a TCP scan against a target, she/he "observes" which services are running on that target.

4.1 Hacking Task Properties

Table 1 summarizes the main properties of a Hacking Task, by also providing a short description for each of them.

With respect to Hacking Task metrics, they strongly depend on the specific Hacking Goal the task in question is associated with. Companies might, e.g., be interested in the effectiveness of their attack response strategies. In such a case, they carry out *Red Team* campaigns, that are an evolution of the Penetration Testing activity. While with standard Penetration Testing the target is aware of Penetration Testers attacks and purposefully disables security controls (since there's an interest in having vulnerabilities be disclosed), with Red Team scenarios the attacker needs to evade security controls and thus must necessarily

Property Name	Property Description
ID	An identifier. This can either be custom or refer to a standard Security Test classification methodology.
Name	A name that helps understand what is the intent of the current hacking task.
Description	A brief description of the hacking task.
Prerequisites	A list of prerequisites that a hacking task must satisfy in order to be executed. Prerequisites might be the output of a previous hacking task.
Dependencies	The list of hacking tasks that must be completed before the execution of the hacking task in question. For example, before trying an anonymous FTP login, the attacker should detect the presence of a running FTP service inside the system.
Category	A phase of the ongoing security assessment (e.g., Enumeration, Scanning, Exploitation).
Results	Output generated upon completion of a hacking task.
Metrics	A performance indicator that describes how is it possible to evaluate the effectiveness of hacking actions with respect to performing a chosen hacking task.

Table 1. Hacking Task properties

behave in a "stealthy" way. In this case, a Hacking Goal might include "stealthiness" requirements, and the related hacking tasks might assign a higher weight to the actions that do not trigger Intrusion Detection Systems alarms. Hacking Task metrics should in this case include such stealthiness properties.

4.2 Hacking Tasks Tree

Hacking task dependencies generate a *Hacking Tasks Tree*.

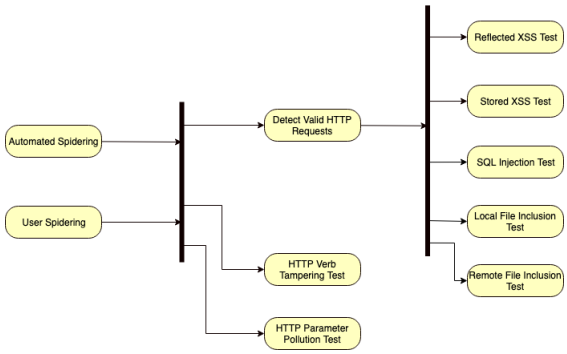

Fig. 2. Hacking Tasks Tree Example for Web Applications

Fig. 2 shows Hacking Task dependencies in a Web Application Penetration Testing model. Each box is a single hacking task. A Hacking Goal in the example is "Find all injection vulnerabilities". An injection vulnerability occurs when a Web Application does not properly validate user input in an HTTP Request. In the example, the Reflected XSS Test detects Cross-Site Scripting vulnerabilities, the SQL Injection Test detects SQL Injection vulnerabilities, the Local File Inclusion Test detects LFI vulnerabilities and the Remote File Inclusion Test detects RFI vulnerabilities.

In order to find an injection vulnerability, the attacker must have chosen a valid path, a valid HTTP request and a parameter of the HTTP request that she/he wants to test. In order to choose a parameter, all forms inside HTML pages have to be found by sending valid HTTP requests to the target. In the model, the "Detect Valid HTTP Requests" hacking task is executed to the purpose. In order to send valid requests, the attacker needs to know available paths at the web server. So, before finding valid HTTP requests, she/he performs the "Automatic Spidering" and "User Spidering" tasks in order to enumerate all paths.

Hacking Goal Classification can use existing sources. As an example, in the case of Web Applications useful resources might be the OWASP [2] (Open Web Application Security Project) Testing Guide and the well-known Web Application Hackers Handbook [5].

5 Strengths and weaknesses of goal-centric classification

A goal-centric attack classification approach makes it easier to find a mapping with Penetration Testing methodologies, since Penetration Testers use hacking

methodologies that are focused on goals rather than on the types of attacks they can perform. Through goal-centric classification it is possible to formalize metrics and evaluate attacks. For instance, if the goal is "Enumerate all Paths of a Web Server", a metric to estimate the effectiveness of performed actions might be the ratio of the number of discovered paths to the number of HTTP requests sent to the Web Server.

The proposed approach might also be used to design intelligent agents. An intelligent agent performs actions inside an environment, and monitors the environment through sensors. It is also important to define agent tasks. Russel [8] defines the concept of "task environment", by using the PEAS (Performance/Environment/Actuators/Sensors) model. In our case, *Performance* refers to the metric used to evaluate the chosen Hacking Goal, *Environment* is the target that the Penetration Tester is analyzing, *Actuators* are the tools and techniques used by the tester and *Sensors* are the "observations" deriving from the executed actions. As part of our ongoing activities, we are formalizing an attacker model based on PEAS, with the aim of showing how it is possible to create a link between Hacking Goal classification and an attacker's behavioral model.

On the downside, the formalization of a goal-centric attack classification model requires proficiency in the security field, as well as specific efforts to properly define metrics that might change depending on the specific hacking task to be performed.

6 Conclusion

In this paper we have proposed a switch of perspective with respect to the definition of proper taxonomies in the cybersecurity field. Namely, we have embraced an attack-centric point of view for the classification of attacks. The model we propose is a hierarchical one and helps identify macro-objectives (*Hacking Goals*) that can be further decomposed into constituent *Hacking Tasks*. For each such task, we have identified finer grained components (*Hacking Actions*), each associated with a specific attack activity.

We have formalized the above concepts as a unified taxonomy framework, illustrated ways for leveraging existing hacking goal classification approaches as sources of information and discussed strengths and weaknesses of a goal-centric attack taxonomy.

References

1. Common Attack Pattern Enumeration and Classification (CAPEC) [online] Available: https://capec.mitre.org.
2. "OWASP", Owasp.org, 2019. [Online]. Available: https://www.owasp.org/index.php/Main_Page. [Accessed: 03- Nov- 2019].
3. RFC 4949 - Internet Security Glossary, Version 2", Tools.ietf.org, 2019. [Online]. Available: https://tools.ietf.org/html/rfc4949. [Accessed: 03- Nov- 2019].
4. "The Penetration Testing Execution Standard", Pentest-standard.org, 2019. [Online]. Available: http://www.pentest-standard.org/index.php/Main_Page. [Accessed: 03- Nov- 2019].
5. D. Stuttard and M. Pinto, The web application hacker's handbook. Hoboken, N.J.: Wiley, 2013.
6. V. M. Igure and R. D. Williams, "Taxonomies of attacks and vulnerabilities in computer systems," in IEEE Communications Surveys & Tutorials, vol. 10, no. 1, pp. 6-19, First Quarter 2008. doi: 10.1109/COMST.2008.4483667
7. J. Obes, C. Sarraute and G. Richarte, "Attack Planning in the Real World", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1306.4044. [Accessed: 04- Nov- 2019].
8. "Artificial Intelligence: A Modern Approach", Aima.cs.berkeley.edu, 2019. [Online]. Available: http://aima.cs.berkeley.edu/. [Accessed: 03- Nov- 2019].
9. I. Kotenko and E. Doynikova, "The CAPEC based generator of attack scenarios for network security evaluation," 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, 2015, pp. 436-441. doi: 10.1109/IDAACS.2015.7340774
10. H. M. Z. A. Shebli and B. D. Beheshti, "A study on penetration testing process and tools," 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, 2018, pp. 1-7.doi: 10.1109/LISAT.2018.8378035