Hierarchy-Driven Approach for Attack Patterns in Software Security Education

Joshua J. Pauli
College of Business and Information Systems
Dakota State University
Madison, SD, 57042, USA
josh.pauli@dsu.edu

Patrick H. Engebretson
College of Business and Information Systems
Dakota State University
Madison, SD, 57042, USA
pat.engebretson@dsu.edu

Abstract

We propose a hierarchy-driven approach to facilitate student learning and foster a deeper understanding of the importance of attack patterns in computer, network, and software security. This is a fundamental point in computer and software security education because the "patch and pray" mentality of software security is insufficient. The importance and significance of our approach is justified by accentuating the deficiencies in previous ad-hoc approaches to teaching attack patterns. Because of the vast amount of information in attack pattern repositories, it is unrealistic to expect students to fully comprehend attack pattern fundamentals and its place in computer, network, and software security.

Keywords: Attack Trees, Attack Patterns, Refinement, Hierarchy.

1. Introduction

Within the past five years, the fields of network, computer, and software security has begun to shift its focus away from perimeter defensive models, such as border routers, firewalls, and intrusion detection systems, to more proactive defensive models [1]. Until recently many companies have simply relied on a patch-when-exploited methodology to writing secure software [2]. In order to better instantiate a proactive defense model, one must begin with software security and make sure that these priorities are carried throughout every phase of the software development lifecycle.

The goal of teaching attack patterns to students is to provide them with a semi-formalized representation of the attacker's perspective. Equally as important is providing each student with a security-focused, expert level understanding of software development and the various ways in which software is currently exploited. Good security and the ability to combat malicious code is the byproduct of understanding said code's mechanics as well as its motivations [3]. By applying this approach, students are able to see how the elements of each attack pattern are related to each other and how each element is related to elements of other attack patterns.

2. Hierarchy-Driven Model

The CAPEC list creates a forum for researchers, developers, security experts, and students to collaborate and share information through a common dialect. However, because of the vast quantities of information within CAPEC, an instructor has a daunting task to introduce the concept of attack patterns and attempt to make use of the CAPEC resource. With over 100 attack patterns and the corresponding Primary and Supporting elements for each, students can be easily overwhelmed by the size, scope and nature of learning such a system.

Given the significance of the subject matter and importance of such concepts, we developed our approach for teaching attack patterns. Our model will rely on utilizing a hierarchy to present attack pattern information logically. One of the major categories for learning strategies is the creation or use of a hierarchy [4]. Furthermore, the process of learning can be facilitated through the implementation of a teaching strategy and good teaching can be defined as one which encourages students to think and remember concepts for themselves [4]. Utilization of a hierarchy will help to bring order as well as facilitate a deeper understanding of relevant attack pattern elements. In order to help solidify the new model, students will first be introduced to the concept of a hierarchy model outside the realm of attack patterns. Upon completion of the abstracted model, students will be asked to apply the hierarchical model to the CAPEC Release 1 attack pattern list.

Specifically, we will focus on a slim element set to incorporate into our hierarchy model [1]. We utilize a top-down approach with the highest level being the most

978-0-7695-3099-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ITNG.2008.15

general element. Subsequent hierarch levels will become more specific in nature and scope. Students are required to follow a minimum 1:1 ratio for each level of the hierarchy as refinement continues from level to the next. While higher ratios of refinement may be possible at each level, the 1:1 ratio between each level is sufficient to introduce attack patterns in an educational setting.

Our hierarchy model is introduced in Figure 1 where more details are realized about the attack pattern as refinement continues to subsequent levels of abstract. Vulnerabilities are at the highest level of abstraction to effectively group the attack patterns in understandable contexts for students.

The purpose is to give students a solid introduction to attack patterns, as well as introduce the CAPEC Release 1 list of attack patterns. As the hierarchy is populated, more details are known about each attack pattern. This information is more useful in a hierarchical format than a textual description with no clear connection between the elements because it is now known what elements influence other elements in the same attack pattern. Students can clearly see these connections between each level of refinement for each attack pattern; this knowledge can then be leveraged when analyzing and designing secure software.

In order to foster a deeper understanding for the students, a significant portion of time was spent specifying and documenting consistent definitions for our approach before populating the hierarchy. We use existing and accepted definitions as summarized below [3].

Level 1: Vulnerability is defined as a large or general classification used to group a collection of related of errors that an attacker can exploit. Every attack pattern outlined by CAPEC is a child of one of the following vulnerabilities: Abuse of Functionality, Spoofing, Probabilistic Techniques, Exploitation of Authentication, Resource Depletion, Exploitation of Privilege/Trust, Injection, Data Structure Attacks, Data Leakage Attacks, Resource Manipulation, Protocol Manipulation, and Time State Attacks.

Level 2: Attack Pattern is a high level blueprint that describes various types of software attacks.

Level 3: Exploit describes a specific instance of an attack pattern. Level 3.1 Bug / Flaw is used to explain difference between a logical or design issue (flaw) and an implementation or coding issue (bug). CAPEC does not specify specific attack patterns as either a bug or flaw. Level 3.1 has been added to force the student to think deeply about the exploit being explored.

Level 4: Activation Zone is the area in a software package which is capable of activating or executing a payload or exploit.

Level 5: Injection Vector refers to the actual format of the input used in an attack. Level 5.1: Payload references any input given to the software in order to carry out an exploit. Note, the hierarchy allows for the fact that not every Attack Pattern makes use of a payload.

Level 6: Reward is the output event or desired outcome of a successful exploit.

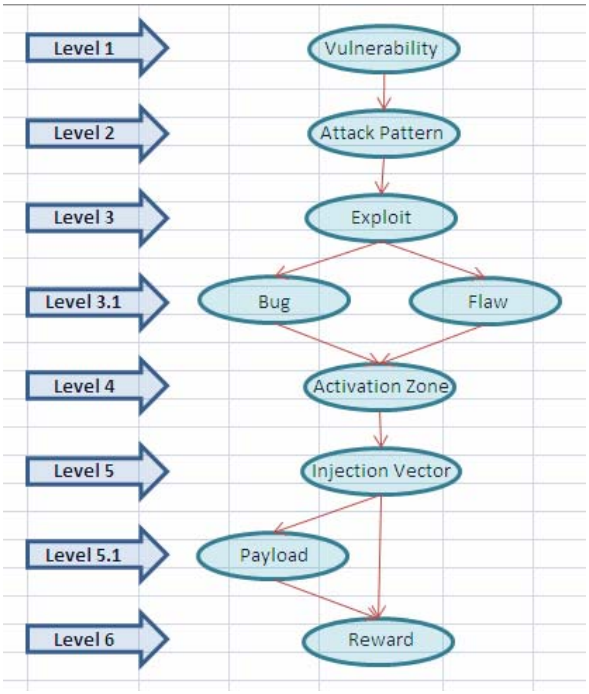

Figure 1. Hierarchy Framework

3. Conclusions

Because of the significance of the attack patterns in relationship to software security it is important that a student's overture to the topic be positive. In order to avoid a negative experience for the students, our introduction to attack patterns and CAPEC focuses on a pared down number of elements and applying them to a hierarchy to aid in retention of information.

References

[1] I. Arce. *Why Attacking Systems Is a Good Idea*, in *Security and Privacy, IEEE*. 2004. p. 17-19.

[2] E. Fernandez. *A Methodology for Secure Software Design*. In *Proc of the Conference on Software Engineering Research and Practice (SERP'04)*. 2004. Las Vegas, Nevada.

[3] G. Hoglund and G. McGraw, *Exploiting Software: How to Break Code*. 2004: Pearson Higher Education.

[4] C. Weinstein and R.E. Mayer, *The teaching of learning strategies*. Handbook of research on teaching, 1986. **3**: p. 315-327.