Abstracting Parent Mitigations from the CAPEC Attack Pattern Dictionary

Patrick H. Engebretson, Joshua J. Pauli, and Kevin F. Streff

College of Business and Information Systems
Dakota State University
Madison, SD, 57042, USA

Abstract

We propose the addition of a more abstract class of mitigations (parent mitigation) derived directly from the currently prescribed mitigations in the CAPEC Release 1 Dictionary (child mitigations). The currently prescribed mitigation strategies are too detailed to be useful in a corporate or non-academic environment. Therefore, we propose the parent mitigations as an additional element to the CAPEC library. The purpose of this new element is to logically group the child mitigation strategies of the 101 Attack Patterns of CAPEC into a useable, manageable, and serviceable list of mitigation strategies. Our parent mitigations provide the CAPCEC standard with a much more applicable set of mitigations to strengthen the current CAPEC Dictionary and aid in the adoption and acceptance of the standard.

Keywords: Attack Pattern, CAPEC, Mitigation Strategies, Abstraction, NIST 800-53

1. Introduction

Attack patterns are defined by the CAPEC Release 1 Dictionary as a formalized representation of a computer attacker's tools, methodologies, and perspective [1]. CAPEC provides a formal definition of each attack by providing descriptive textual fields. These fields, defined as elements, provide explicit details for each identified exploit. The current CAPEC release includes a list of 101 specific security attacks and the textual elements of each. CAPEC was created to make up for the shortcoming in attack-specific security research which lacked a standard that provided a consistent documentation of attacks [2]. One drawback to the current standard is that the current CAPEC dictionary is so large, it is wrought with the possibility of user error [3].

While CAPEC's Release 1 Dictionary provides a solid framework, the current format and sheer volume of information renders the new standard nearly useless to anyone outside of the academic field [4]. The current CAPEC standard provides an element entitled "Solutions and Mitigations" which is defined as "the actions or approaches that can potentially prevent or mitigate the risk of this type of attack. These solutions and mitigations are targeted to improve the resistance of the target software and thereby reduce the likelihood of the attack's success or to improve the resilience of the target software and thereby reduce the impact of the attack if it is successful." [5]

This element is a required field in order to make the standard effective for mitigating attacks. Ideally, a user concerned with a given attack pattern should be able to review the CAPEC standard for that attack, and formulate a plan for reducing exposure to the attack. However, we have found that the current mitigations outlined are either far too detailed, far too numerous, or far too inconsistent to be useful.

For example, anyone concerned with a "Sever Side Include" attack (CAPEC Attack Pattern #101), can consult the "Solutions and Mitigations" element assigned to this attack pattern. One of the currently prescribed mitigations is "Set the OPTIONS IncludesNOEXEC in the global access.conf file or local .htaccess (Apache) file to deny SSI execution in directories that do not need them" [1]. While this level of detail does have benefits, there is simply too much granularity with this mitigation strategy. This currently prescribed mitigation strategy could lead adopters to believe they are safe from a Server Side Include attack if they do not make use of the Apache web server.

It is important to note that we are not advocating for the replacement of the current "Solutions and Mitigations" element, rather we provide a more manageable version of the mitigation strategies for all 101 attack patterns. The current "Mitigations and Solutions" element (defined as "child mitigations" in our approach) will still be readily available to provide detailed strategies.

Attack patterns are relatively new, having been introduced within the past decade [6]. It is the goal of this paper to leverage this vast repository of attack pattern information while simultaneously adding an addition layer of information to provide a uniform standard for mitigation strategies for each attack pattern. This is accomplished through the introduction of the new parent mitigations.

Section 2 covers related work that our current research is based on. Section 3 outlines our parent mitigations and why we chose them. Section 4 introduces the process of adding parent mitigations. Section 5 covers future work and we conclude in section 6.

2. Related Work

Within the past five years, the fields of network, computer, and software security have begun to shift their focus away from perimeter defensive models, such as border routers, firewalls, and intrusion detection systems, to more proactive defensive models [2]. Until recently many companies have simply relied on a patch-when-exploited methodology to writing secure software [7]. This patch and penetrate methodology does nothing to address the underlying security issues. In order to better instantiate a proactive defense model, one must include software security and make sure that these priorities are carried throughout every phase of the software development lifecycle. Good security and the ability to combat malicious code is the byproduct of understanding said code's mechanics as well as its motivations [8].

Fostering a deep understanding of attack patterns can lead to the permeation of security throughout the software development life cycle, as well as heighten awareness of known exploits, vulnerabilities and weaknesses [9]. Integrating and increasing attack pattern knowledge can result in adding security by creating less exposure to identified bugs and known flaws [2]. Attack patterns can be used to create a security checklist, which in turn can lead to a higher level of security [10].

The origins of attack patterns can be traced back to concepts outlined by Gamma, et. al. when the foundation for today's attack patterns where established as the concept of a general, repeatable solution to identified system development problems [11]. More recently the concept of presenting from an attacker's perspective was done on an individual, or attack by attack basis, with no agreed upon formula, structure, or common language for consistently presenting such a viewpoint [2].

The lack of a common or united vocabulary makes it difficult to gather, analyze, and share pertinent information in meaningful ways which could be used to advance the discipline of software security. Moore began to formalize a concept of combining various types of malicious attacks (i.e. the attacker's perspective) with the pattern framework [6]. Hoglund and McGraw built upon the foundation of Moore's paper by more formally defining attack patterns and identifying 48 distinct attack patterns [2].

The National Cyber Security Division of the Department of Homeland Security in conjunction with Cigital and MITRE Corporation agreed to sponsor CAPEC. The final result of this collective effort was published in March of 2007 and included a formalized attack driven perspective of software security with 101 different attack patterns outlined [12].

The Common Attack Pattern Enumeration and Classification (CAPEC) list provides an official schema and formal representation for defining individual attack patterns [12]. Given the sheer volume of information included in the Release 1 Dictionary, which includes not only the defined 101 attack patterns but their descriptive elements as well, there tends to be considerable confusion and information overload when individuals are first introduced to the concept of attack patterns [3].

CAPEC formally organizes and presents each attack pattern by gathering and displaying both primary and supporting data elements [13].

Primary elements include:

- Attack Pattern ID
- Attack Pattern Name
- Description
- Related Weaknesses
- Related Vulnerabilities
- Methods of Attack
- Examples-Instances
- References
- Solutions and Mitigations
- Typical Severity
- Typical Likelihood of Exploit
- Attack Prerequisites
- Attacker Skill or
- Knowledge Requirements
- Resources Required
- Attack Motivation-Consequences
- Context Description

Supporting elements include:

- Injection Vector
- Payload
- Activation Zone
- Payload Activation Impact
- Probing Techniques
- Indicators/Warnings of Attack
- Obfuscation Techniques
- Related Attack Patterns
- Relevant Security Requirements
- Relevant Design Patterns
- Relevant Security Principles
- Related Guidelines

In order to efficiently manage the CAPEC Release 1 Dictionary, we proposed the creation of a prototype tool. [4]. The prototype tool described would allow for the collection, organization and mapping of several key attack pattern elements including mitigation strategies.

Exploration and examination of the various techniques used by malicious attackers is an important step in providing better security for our technology resources [14]. McGraw points out that the best penetration tests are built on a solid understanding of both design and risks [15]. This type of understanding can only be achieved when we have a formal set of definitions to build and share knowledge. CAPEC provides such a framework.

3. Selecting Parent Mitigations

The CAPEC Release 1 Dictionary includes nearly 400 individually prescribed controls which can be used to mitigate or reduce the effects of the defined attack patterns. This current level of detail in the "Solutions and Mitigations" element tends to be inconsistent. Some attack patterns provide an extremely granular level of detail. For example, one of the prescribed mitigations for attack pattern #42 (MIME Conversion) calls for disabling "the 7 to 8 bit conversion by removing the F=9 flag from all Mailer specifications in the sendmail.cf file." This level of detail may lead CAPEC adopters to believe that they need not be concerned with MIME Conversion attacks if they implement a Microsoft Exchange server rather than a Sendmail-based email server. Such a mistake could lead to an increased attack exposure and a false sense of security.

The reverse is also true; some attack patterns provide only a high level overview of potential mitigation strategies. Attack pattern #9 (Buffer Overflow in Local Command-Line Utilities) includes the "Do not unnecessarily expose services" mitigation. This is too vague, undefined and unclear to be of use to many users.

In order to increase the effectiveness and consistency of mitigation strategies, we propose the inclusion of a new element to the CAPEC standard. Our "Parent Mitigation" element is directly abstracted from the currently prescribed CAPEC "Solutions and Mitigations" element.

We examined several standards when looking for a complete set of parent mitigation strategies to complement the CAPEC Dictionary. It is vital to make use of a predefined, currently accepted, standardized list of controls. The implementation and use of an accepted standard removes the heuristic tone of an ad-hoc approach.

We reviewed COBIT 4.1 [16], ISO 27002:2005 [17] and NIST SP 800-53 [18] for an acceptable list of controls to use as "Parent Mitigations" in our approach. After reviewing the controls outlined in each of these standards, we choose to make use of NIST 800-53 (revision 2). Both NIST and CAPEC have strong ties to the United States Federal government. NIST is a non-regulatory federal agency funded through the U.S. Department of Commerce while CAPEC is the direct result of funding from the Department of Homeland Security [19]. CAPEC is a federally funded classification of attacks and NIST is a federally funded list of controls; the union of these two standards is logical. During this selection process, we were able to reject the controls outlined in the COBIT standard [20]. This research determined that the COBIT control framework is less specific to Information Systems or Information Technology details than the controls outlined in ISO [20]. Because of the technical nature of attack patterns, we focus on controls which provide the most technical details. ISO is a "management system, not a technology specification" [21]. We are providing a technical specification for mitigations as part of our approach. We view NIST as a stronger match than the business process-oriented ISO standard.

We chose to use NIST because the controls provide a ready-made hierarchy which fits within our parent-child model. This additional level of detail and structure not only correlates directly with our work, but will also be used in future work to further extend the relationship between NIST and CAPEC.

NIST 800-53 provides a usable hierarchy already in place. At the top level, this hierarchy consists of "Family" controls which are general and wide reaching. The standard further breaks down each of the "Family" controls into a series of detailed controls. The final draft of 800-53-r2 includes a total of 17 "Family" level controls [18]:

- Access Control
- Awareness and Training
- Audit and Accountability
- Certification, Accreditation, and Security Assessments
- Configuration Management
- Contingency Planning
- Identification and Authentication
- Incident Response
- Maintenance
- Media Protection
- Physical and Environmental Protection
- Planning
- Personnel Security
- Risk Assessment
- System and Services Acquisition
- System and Communications Protection
- System and Information Integrity

Our approach introduces the appropriate NIST control into the existing CAPEC dictionary as a "Parent Mitigation" in order to provide a more generalized mitigation strategy for each of the 400 attack patterns.

4. Abstracting Child Mitigations

The process of abstracting “Parent Mitigations” is accomplished by compiling the entire attack pattern mitigations from CAPEC into a single list. A line item review of each mitigation strategy is then completed. Using the control definitions outlined in NIST 800-53, we correlate each CAPEC control with a corresponding NIST control. Although we are only interested in the NIST “Family” control, we map each of the current CAPEC mitigations to the detailed controls in NIST 800-53 to ensure completeness. Once this process is complete, we are able to determine the appropriate family level controls for inclusion into the CAPEC standard.

We use a subset of the CAPEC dictionary to illustrate our approach. Attack Pattern ID #3 is “Using Leading 'Ghost' Character Sequences to Bypass Input Filters”. Examination of the CAPEC Dictionary provides three mitigations for this attack:

1. Perform white list, rather than black list, input validation.
2. Cononicalize all data prior to validation.
3. Take an iterative approach to input validation (defense in depth)

Upon careful review of each of these mitigations and using the comprehensive guidelines provided as part of the NIST 800-53 standard, we are able to match each of these to one or more of the detailed NIST controls. The first control outlined by CAPEC is correlates to the following NIST controls:

1. AC-3 Access Enforcement
2. AC-4 Information Flow Enforcement
3. IA-3 Device Identification and Authentication

These detailed NIST controls are part of the following NIST “Families”:

1. Access Control
2. Identification and Authentication

The second control outlined by CAPEC is “Conicalize all data prior to validation”. Using the NIST 800-53 guidelines, we correlate this with the following NIST controls:

1. SI-9 Information Input Restrictions
2. SI-10 Information Accuracy, Completeness, Validity, and Authenticity

Both of these controls fall under the NIST “System and Information Integrity” control “Family”.

The final mitigation is, “Take an iterative approach to input validation (defense in depth)”. Upon careful review of the NIST 800-53 guidelines, we correlate this CAPEC mitigation with the following NIST control:

1. SI-10 Information Accuracy, Completeness, Validity, and Authentication

SI-10 belongs to the “System and Information Integrity” control “Family”. These three mitigations, and the relationships among them, are introduced in figure 1.

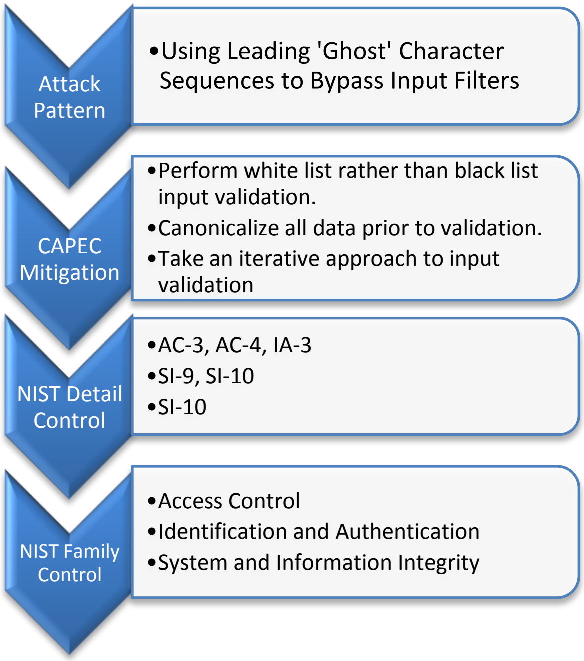
Attack Pattern
•Using Leading 'Ghost' Character Sequences to Bypass Input Filters
CAPEC Mitigation
•Perform white list rather than black list input validation.
•Canonicalize all data prior to validation.
•Take an iterative approach to input validation
NIST Detail Control
•AC-3, AC-4, IA-3
•SI-9, SI-10
•SI-10
NIST Family Control
•Access Control
•Identification and Authentication
•System and Information Integrity

Figure 1. Relationship among CAPEC and NIST for Attack Pattern #3.

CAPEC mandates three controls and our process of abstraction results in the same number of controls needed to mitigate the risk. We are not concerned with reducing the number of controls for each attack pattern. Rather, we are attempting to formalize and reduce the total number of possible mitigations. Our approach reduces the total mitigations from nearly 400 (from CAPEC) to no more than 17 (from the NIST “Family”).

This same process can be followed for attack pattern 4 (Using Alternative IP Address Encodings). Figure 2 introduces the results of our approach on this attack pattern.

There is significant value in completing this abstraction process. Adding the “Parent” mitigation into the CAPEC dictionary brings a level of consistency and standardization. The CAPEC Dictionary’s mitigation strategies are now standardized into 17 “Parents” (down from the nearly 400) at the same level of abstraction. This allows managers to make better use of the CAPEC dictionary. By abstracting these mitigations into 17 categories, users are less likely to dismiss a particular attack pattern because the mitigation is too detailed or too specific. This is currently a risk for CAPEC adopters who believe that they are not at risk for a given attack because they do not have the specific technology mentioned in the CAPEC mitigation

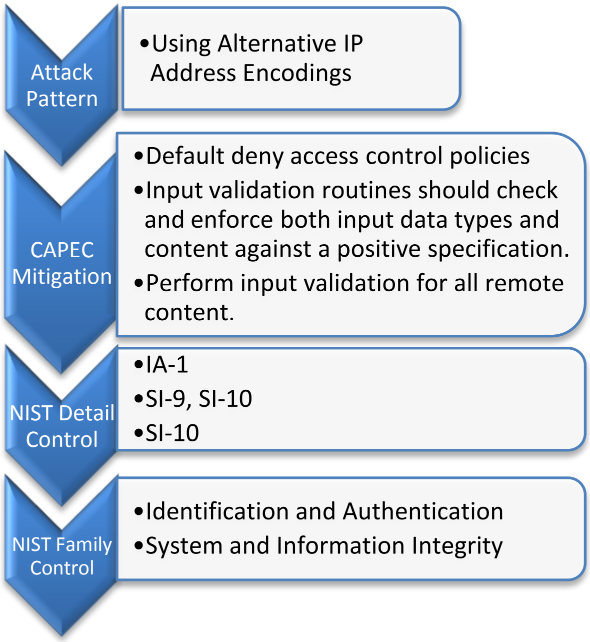
Attack Pattern
•Using Alternative IP Address Encodings
CAPEC Mitigation
•Default deny access control policies
•Input validation routines should check and enforce both input data types and content against a positive specification.
•Perform input validation for all remote content.
NIST Detail Control
•IA-1
•SI-9, SI-10
•SI-10
NIST Family Control
•Identification and Authentication
•System and Information Integrity

Figure 2. Relationship among CAPEC and NIST for Attack Pattern #4.

5. Future Work

The immediate future work for this approach is to complete the remaining attack patterns. Using the process outlined in this paper, each of the mitigations specified in the CAPEC Dictionary needs to be abstracted. Upon completion of this task, a formal request will be made to the managers of the CAPEC standard asking for the inclusion of the newly constructed “Parent” element.

Other future work calls for the re-inclusion of parent threats. While these threats are currently available on the CAPEC web site, they are not formally defined by any descriptive element.

Work on trimming the number of attack pattern elements as will also be considered [3]. This approach to make the CAPEC standard more usable will speed adoption and acceptance.

6. Conclusions

While the current CAPEC standard provides a significant amount of information, there are tremendous variations in the depth and breadth of the “Mitigations and Solutions” currently outlined for each attack pattern. Some attack patterns provide detail that is too granular while others provide information that is vague. Our approach injects a “Parent Mitigation” element into the dictionary to provide consistency to the CAPEC Release 1 Dictionary. Because the current “Mitigation and Solutions” element provides valuable information, we are not advocating its removal. Rather our intention is to add a “Parent” element to provide a manageable and consistent number of more abstracted mitigations. This is a valuable step to the increased adoption and wide spread acceptance of the CAPEC Release 1 Dictionary.

References

[1] "CAPEC - Common Attack Pattern Enumeration and Classification (CAPEC)," 2007.

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

[3] J. Pauli and P. Engebretson, "Hierarchy-Drive Approach for Attack Patterns in Software Security Education," in *5th International Conference on Information Technology : New Generations*, Las Vegas, 2008.

[4] J. Pauli and P. Engebretson, "Towards a Specification Prototype for Hierarchy-Driven Attack Patterns," in *5th International Conference on Information Technology : New Generations (ITNG 2008)*, Las Vegas, 2008.

[5] S. Barnum, "Common Attack Pattern Enumeration and Classification (CAPEC) Schema Description," 2008.

[6] A. P. Moore, R. J. Ellison, and R. C. Linger, *Attack Modeling for Information Security and Survivability*: Carnegie Mellon University, Software Engineering Institute, 2001.

[7] E. B. Fernandez, "A Methodology for Secure Software Design," in *Conference on Software Engineering Research and Practice (SERP'04)*, Las Vegas, Nevada, 2004, pp. 21-24.

[8] I. Arce, "Why Attacking Systems Is a Good Idea," in *Security and Privacy, IEEE*. vol. 2, 2004, pp. 17-19.

[9] M. Gegick and L. Williams, "Matching attack patterns to security vulnerabilities in software-intensive system designs," pp. 1-7, 2005.

[10] S. Barnum, "Attack Patterns as a Knowledge Resource for Building Secure Software," A. Sethi, Ed. OMG Software Assurance Workshop: Cigital, 2007.

[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1995.

[12] S. Barnum and S. Amit, "Further Information on Attack Patterns," in *Build Security In Setting a Higher Standard for Software Assurance*, U. S. D. o. H. Security, Ed.: Cigital, Inc., 2006.

[13] S. Barnum, "Attack Patterns: Knowing Your Enemy in Order to Defeat Them," in *Black Hat DC 2007* Washington DC, 2007.

[14] E. Skoudis and T. Liston, *Counter Hack Reloaded A Step-by-Step Guide to Computer Attacks and Effective Defenses*, 2nd ed. Upper Saddle River: Prentice Hall, 2006.

[15] G. McGraw, *Software Security: Building Security In*: Addison-Wesley Professional, 2006.

[16] ISACA, "COBIT 4.1," Rolling Meadows, Illinois 2008, p. Cobit has been developed and is maintained by the Information Systems Audit and Control Association (IACSA) http://www.iacsa.org.

[17] ISO, "27002:2005," I. S. Organization, Ed. Geneva 2005.

[18] NIST, "800-53 Rev. 2," in *Recommended Security Controls for Federal Information Systems*, U. D. o. Commerce, Ed., 2007.

[19] NIST, "800-53," in *Recommended Security Controls for Federal Information Systems*, U. D. o. Commerce, Ed., 2006.

[20] S. Flowerday and R. Von Solms, "Real-Time Information Integrity = Systems Integrity + Data Integrity + Continuous Assurances," *Computers & Security*, vol. 24, pp. 604-613, November 2005 2005.

[21] A. Calder, *Information Security Based on ISO 27001/ISO 17799: A Management Guide*: Van Haren Publishing, 2006.