Common Attack Pattern Enumeration and Classification (CAPEC) Schema Description

January 15, 2008

This work was performed under contract to:

Department of Homeland Security

Mr. Sean Barnum
Principal Consultant

Cigital, Inc.

21351 Ridgetop Circle
Suite 400
Dulles, VA 20166
Phone: + 1 (703) 404-9293

www.cigital.com

Table of Contents

Document Revision History

Version	Modification	Date	Author
1.0	Initial document	9/25/06	Sean Barnum
1.1	Revisions and extensions to schema up to initial draft release of CAPEC content for community review	3/12/07	Sean Barnum
1.2	Formalized description schema enhancements	11/20/07	Sean Barnum
1.3	Added pattern abstraction levels	11/30/07	Sean Barnum

Preface

Purpose

The purpose of this document is to define a standard schema for representing attack patterns and to describe in adequate detail the meaning and intent of each constituent schema element. This schema will form the foundation for the Common Attack Pattern Enumeration and Classification (CAPEC).

Audience

This document is intended for contributors to and users of the CAPEC. For contributors, it is a mechanism for discussion of and feedback into decisions regarding the appropriate useful schematic elements as well as terminology and formatting. For users of the CAPEC, it is an explanatory guide to assist in understanding the purpose of attack patterns as well as the meaning of individual attack pattern elements.

1 Primary Schema Elements

1.1 Identifying Information

These schema fields are intended to provide information to the reader that will allow the clear and unambiguous identification of each specific attack pattern.

1.1.1 Attack Pattern ID

This field contains a unique integer identifier for the pattern. Externally, patterns will be referenced using this ID in the form "CAPEC-####" (e.g., CAPEC-12).

1.1.2 Attack Pattern Name

This field contains a short descriptive name for the pattern. It should be kept as short as possible but also clearly convey the nature of the attack being described.

1.2 Describing Information

These schema fields are intended to provide information to the reader that clearly and effectively describe the attacks defined by this attack pattern.

1.2.1 Description

This field contains a detailed description of the attack including the chain of actions taken by the attacker. More comprehensive descriptions could include relevant attack trees [Schneier 99] and/or exploit graphs [McGraw 06] to more clearly elaborate this type of attack.

1.2.1.1 Summary

Brief description of what the attack involves and what weaknesses it is leveraging for exploit.

1.2.1.2 Attack Execution Flow

Outline of the steps involved in an attacker executing the typical flow of the attack.

1.2.1.2.1 Attack Phase

Segment the attack steps into the various phases of attack. One of three phases "Explore," "Experiment," or "Exploit." Each phase should appear at most once, and attack steps should be grouped by what kind of activities the attacker is carrying out. The exploration and experimentation phases may or may not occur during a particular attack, because the attacker may already know exactly how to exploit a system.

Name

"Explore," "Experiment," or "Exploit."

Attack Step

Brief description of an individual action step in carrying out the attack.

Attack Step Title

This field contains a short descriptive title for the attack step. It should be kept as short as possible but also clearly convey the nature of the attack step being described.

Attack Step Description

This field contains a brief description of the attack step.

Attack Step Technique

This field captures various techniques that the attacker can use to achieve the attack step's goal. For example, an attacker may use tools such as WebScarab and Tamper Data in the experimentation phase of a SQL Injection attack pattern. The techniques include references to environments, because not all techniques work in all environments

Attack Step Technique Description

This field contains a brief description of the attack step technique.

Environments

References the defined environments where this attack step technique is applicable.

Indicators of Susceptibility

These are indicators that the application may or may not be susceptible to the given attack step (not necessarily the pattern as a whole).

ID

This field contains a unique integer identifier for the indicator.

Type

Each indicator has a mandatory type attribute that can be one of the values "Positive," "Negative," or "Inconclusive." For example, a positive indicator of susceptibility to parameter tampering is the existence of parameters in the URL. Although it does not guarantee susceptibility, it indicates a cause for further examination. A negative indicator for the technique of privilege escalation is a lack of credentials and user identifiers in an application. Again, this is not a

conclusive measure of resistance to attack, but an indicator that the attack step technique is unlikely to bear significant fruit. An inconclusive indicator of susceptibility to dynamic code injection is a page whose URL ends in .jsp, .asp, or .do but which has no visible explicit parameters. Such URLs typically indicate dynamic processing, but since no visible parameters are passed, it is inconclusive whether dynamic code could be injected into the application.

Indicator Description

This field contains a brief description of the indicator.

Environments

References the defined environments where this indicator of susceptibility is applicable.

Outcome

This field captures possible outcomes for this attack step.

ID

This field contains a unique integer identifier for the outcome.

Type

An outcome has a mandatory *type* attribute that can be one of the values "success," "failure," or "inconclusive." It indicates what results of executing the attack step techniques should be considered successes, which should be considered failures, and which ones are inconclusive. Outcomes' successes are determined relative to the attacker's point of view. It is a success if the attack step got the attacker closer to his goal of attacking the application. It is a failure if the attacker got no closer to his goal.

Security Control

This field captures security controls for this attack step that describe ways in which the attack step can be detected, corrected, or prevented. These are presented from a defender's point of view, where the defender may be a developer, tester, operations administrator, or other resource resisting the attacker.

ID

This field contains a unique integer identifier for the security control.

Type

Each security control has a mandatory *type* attribute that can be one of the values "Detective," "Corrective," or "Preventative." Detective controls detect an attacker's activities in the attack step, whether the activities are successful or not. Corrective controls attempt to mitigate an attacker's success by responding to a successful outcome. They are not related to or normalized against outcomes. Preventative controls are those that make the attack step unlikely or impossible to succeed.

1.2.2 Related Weaknesses

Which specific weaknesses does this attack target and leverage? Specific weaknesses (underlying issues that may cause vulnerabilities) reference the industry-standard Common Weakness Enumeration (CWE[1]). This list should include not only those weaknesses that are directly targeted by the attack but also those whose presence can directly increase the likelihood of the attack succeeding or the impact if it does succeed.

1.2.2.1 Related Weakness

CWE_ID

The CWE_ID is a field that exists for all weaknesses identified in the Common Weakness Enumeration (CWE). It is a unique value that allows each weakness to be unambiguously identified. The CWE_ID field for the attack pattern contains the value of the CWE_ID for the specific related weakness.

Weakness Name

The CWE Name is a rendered field that is displayed for all. It is a unique, short text description of the weakness that allows each weakness to be clearly identified. The Weakness Name field for the attack pattern is only displaying the value of the CWE Name field for the specific targeted weakness it is not part of the CAPEC schema.

Weakness Relationship Type

This field describes the nature of the relationship between this weakness and the attack pattern. Weaknesses that are specifically targeted by the attack are of type "Targeted". Weaknesses which are not specifically targeted but whose presence may increase the likelihood of the attack succeeding or the impact of the attack if it does succeed are of type "Secondary".

[1] http://cwe.mitre.org

Weakness Relationship Types
Targeted
Secondary

1.2.3 Related Vulnerabilities

What specific vulnerabilities does this attack target and leverage? Specific vulnerabilities should reference industry-standard identifiers such as Common Vulnerabilities and Exposures (CVE[2]) numbers and/or US-CERT[3] numbers. As vulnerabilities are much more specific and localized than weaknesses, it is typically rare that an attack pattern will target specific vulnerabilities. This would most likely occur if they are targeting vulnerabilities in underlying platforms, frameworks, libraries, etc.

1.2.3.1 Related Vulnerability

Vulnerability ID

This field uses the unique reference ID for a specific related vulnerability utilizing an industry standard vulnerability listing (e.g., CVE-2006-4192, VU#650769, etc.).

Vulnerability Description

This field contains a short textual description of the specific related vulnerability taken from the industry standard vulnerability listing.

1.2.4 Methods of Attack

This field describes the mechanism of attack used by this pattern. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined vectors which is currently incomplete and will grow as new relevant vectors are identified. This field can help define the applicable attack surface required for this attack.

1.2.4.1 Method of Attack

This field describes an individual method of attack.

Enumerated Choices
Injection
Modification of Resources
Protocol Manipulation
Analysis

[2] http://cve.mitre.org

[3] http://www.us-cert.gov/

Enumerated Choices
API Abuse
Brute Force
Flooding
Time and State
Spoofing
Social Engineering

1.2.5 Examples-Instances

This field contains explanatory examples or demonstrative exploit instances of this attack, which are intended to help the reader understand the nature, context and variability of the attack in more practical and concrete terms.

1.2.5.1 Example-Instance

Example-Instance Description

This field describes in detail a specific example or exploit instance of this attack. It should outline the context of the attack, the targeted software, the targeted weaknesses or vulnerabilities, the specific set of actions involved in the attack and the resulting impact of the attacks success or failure (in the case of counterexamples).

Example-Instance Related Vulnerabilities

This field lists the specific vulnerabilities targeted by this exploit instance of the attack. Specific vulnerabilities should reference industry-standard identifiers such as Common Vulnerabilities and Exposures (CVE) numbers and/or US-CERT numbers.

1.2.6 References

This field enumerates reference resources that were used to develop the definition of this attack pattern and those that could prove valuable to the reader looking for further information on this attack.

1.2.6.1 Reference

This field should describe the reference clearly and unambiguously by name and with some method of address such that the reader can locate the resource for further reference.

1.3 Prescribing Information

These schema fields are intended to provide information to the reader regarding specific recommended actions to be taken in regards to this attack pattern.

1.3.1 Solutions and Mitigations

This field describes 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.

1.3.1.1 Solution or Mitigation

This field describes an individual blocking solution or mitigation.

1.4 Scoping and Delimiting Information

These schema fields are intended to provide information to the reader to assist in determining which attack patterns are appropriate for a given context. They should help answer the question, "Which ones should I care about?"

1.4.1 Typical Severity

On a rough scale (Very Low, Low, Medium, High, Very high), what is the typical severity of impact to the targeted software if this attack occurs? The severity of a specific attack instance can vary greatly depending on the specific context of the target software under attack. This field is intended to capture an overall typical average value for this type of attack with the understanding that it will not be completely accurate for all attacks.

1.4.2 Typical Likelihood of Exploit

On a rough scale (Very Low, Low Medium, High, Very High), what is the overall likelihood of this type of attack typically succeeding considering the attack prerequisites, targeted weakness attack surface, skill required and resources required as well as available and likely implemented blocking solutions? The likelihood of exploit of a specific attack instance can vary greatly depending on the specific context of the target software under attack. This field is intended to capture an overall typical average value for this type of attack with the understanding that it will not be completely accurate for all attacks.

1.4.3 Attack Prerequisites

This field describes the conditions that must exist or the functionality and characteristics that the target software must have or behavior it must exhibit for an attack of this type to succeed.

1.4.3.1 Attack Prerequisite

This field describes an individual attack prerequisite.

1.4.4 Attacker Skill or Knowledge Required

This field describes the level of skill or specific knowledge required by an attacker to execute this type of attack. This should be communicated on a rough scale (Low, Medium, High) as well as in contextual detail.

For example:

Low - Basic computer familiarity

Low - Basic SQL knowledge

Medium - Moderate scripting and shell experience and ability to disassemble and decompile

High - Expert knowledge of LINUX kernel

High - Detailed knowledge of target software development practices and business context (former employee)

Etc.

1.4.5 Resources Required

This field describes the resources (CPU cycles, IP addresses, tools, etc.) required by an attacker to effectively execute this type of attack.

1.4.6 Attack Motivation-Consequences

What is the attacker trying to achieve by using this attack? This is not the end business/mission goal of the attack within the target context but rather the specific technical result desired that could be leveraged to achieve the end business/mission objective. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined motivations/consequences which is currently incomplete and will grow as new relevant possibilities are identified. This information is useful for aligning attack patterns to threat models and for determining which attack patterns are relevant for a given context.

Attack Motivation-Consequence

This field describes an individual attack motivation-consequence.

Enumerated Choices
Denial of Service
Run Arbitrary Code
Information Leakage
Data Modification
Privilege Escalation

1.4.7 Context Description

This field describes the contexts in which this pattern is relevant and/or gives further background context for the attack. This information is useful for better understanding the nature of this type of attack.

1.4.8 Purpose

This field describes the generalized purpose of the pattern in order to enable the capture of pattern composibility. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined purposes which may be currently incomplete and may grow as new relevant possibilities are identified.

Enumerated Choices
Reconnaissance
Penetration
Exploitation
Obfuscation

1.4.9 CIA Impact

This field describes the typical impact of this pattern on the standard security characteristics of confidentiality, integrity and availability. The actual impact of a specific attack instance can vary greatly depending on the specific context of the target software under attack. This field is intended to capture an overall typical average value for this type of attack with the understanding that it will not be completely accurate for all attacks.

1.4.9.1 Confidentiality Impact

This field describes the typical impact of this pattern on the confidentiality characteristics of the targeted software and related data.

1.4.9.2 Integrity Impact

This field describes the typical impact of this pattern on the integrity characteristics of the targeted software and related data.

1.4.9.3 Availability Impact

This field describes the typical impact of this pattern on the availability characteristics of the targeted software and related data.

1.4.10 Technical Context

This field describes the technical context in which this pattern is relevant. This could involve factors such as platform, OS, language, architectural paradigm, etc. The ultimate specific factors to be part of this field have yet to be determined. This field currently contains the factors outlined here. This field will be iteratively refined as more is learned

through the identification and elaboration of new attack patterns. This information is useful for aligning attack patterns to attack surfaces and for determining which attack patterns are relevant for a given context.

1.4.10.1 Architectural Paradigm

This field outlines the architectural paradigms of target software where the pattern is possible and relevant. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined paradigms which may be currently incomplete and may grow or change as new relevant possibilities are identified.

Enumerated Choices
Mainframe
Client-Server
n-Tier
SOA
Other

1.4.10.2 Framework

This field outlines the frameworks used as part of the target software where the pattern is possible and relevant. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined frameworks which may be currently incomplete and may grow or change as new relevant possibilities are identified.

Enumerated Choices
J2EE
.NET
Struts
Spring
Hibernate
Other

1.4.10.3 Platform

This field outlines the platforms (OS) of the target software where the pattern is possible and relevant. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined platforms which may be currently incomplete and may grow or change as new relevant possibilities are identified.

Enumerated Choices
Windows
UNIX-LINUX
Solaris
Other

1.4.11 Keywords

This field is intended to be a catch-all field for assisting in searching and subsecting a collection of patterns according to criteria not contained elsewhere in this schema.

1.4.11.1 Keyword

This field contains an individual keyword to be used for tagging and searching.

1.4.12 Pattern Abstraction Level

This field defines an appropriate abstraction level for the pattern which helps guide the appropriate information needed for its definition as well as where and how it is most usefully leveraged.

Enumerated Choices
Meta
Standard
Detailed

While CAPEC by design covers a range of abstraction levels for attack patterns, this field is intended to give some distinction between levels to aid in their creation, interpretation and usage. Given a rather complex continuum of abstraction, there are no hard and fast rules for absolutely assigning a particular abstraction level. However, the characterization and heuristics below are useful in guiding such assignments.

1.4.12.1 Meta abstraction attack patterns

- Context
  - Technology and functional context non-specific
  - Often defines approaches to thinking on how to violate assumptions made in software
- Relevant knowledge resources
  - Security principles
  - Common weaknesses
- Process activities where value is provided

    - Writing high-level security policy
- Solutions and mitigations
    - Solutions & mitigations are typically educational or platitudenal in nature
- General value proposition
    - Supports software security learning and high-level thinking

1.4.12.2 Standard abstraction attack patterns

- Context
    - Typically functional context-specific but technology context non-specific
    - Often defines actionable approach to exploiting specific weaknesses in software
- Relevant knowledge resources
    - Common weaknesses
    - Design patterns
    - Security patterns
    - Secure coding standards
    - Security guidelines
- Process activities where value is provided
    - Writing detailed security policy
    - Security requirements (guidance for abuse case creation)
    - Security test case development (moderate value)
    - Application penetration testing
- Solutions and mitigations
    - Solutions & mitigations are both design and implementation-specific in nature
- General value proposition
    - Supports software security analysis

1.4.12.3 Detailed abstraction attack patterns

- Context
    - Technology and/or functional context specific
    - Often defines specific delivery vector and/or attack content
- Relevant knowledge resources
    - Common weaknesses

  - Common vulnerabilities
  - Secure coding standards
- Process activities where value is provided
  - Security requirements (templates for abuse case creation)
  - Secure code review (indirectly through targeted weaknesses)
  - Security test case development (strong value)
  - Application penetration testing
- Solutions and mitigations
  - Solutions & mitigations are typically implementation-specific and occasionally design- specific
- General value proposition
  - Approaches or achieves support for clear attack definition and/or automation

1.4.12.4 CAPEC schema recommendations for each abstraction level

Schema Elements	Meta	Standard	Detailed
Attack Pattern ID	R	R	R
Attack Pattern Name	R	R	R
Description	R (High-level typically without Attack Execution Flow)	R (Attack Execution Flow, Phases and Steps defined with minimal deep Attack Step Technique detail)	R (Detailed with specific Attack Step Techniques and Security Controls defined)
Related Weaknesses	S (Typically high-level weaknesses)	R (All types of weaknesses)	R (Typically low-level, specific weaknesses)
Related Vulnerabilities	S (Rarely applicable)	S (Occasionally applicable)	S (Often applicable)
Methods of Attack	R	R	R
Examples-Instances	O	S	R
References	S	S	S
Solutions and Mitigations	R (Typically educational, policy or process)	R (Typically implementation and design)	R (Typically implementation and occasionally design)
Typical Severity	O	R	R
Typical Likelihood of Exploit	O	R	R

Schema Elements	Meta	Standard	Detailed
Attack Prerequisites	S	S	S
Attack Skill or Knowledge Required	S (Loosely defined)	S	R (Tightly defined)
Resources Required	S	S	R
Attack Motivation-Consequences	R	R	R
Context Description	S	S	S
Purpose	R	R	R
CIA Impact	S	R	R
Technical Context	O	S	R
Keywords	S	S	S
Injection Vector	O	S	S
Payload	O	S	S
Activation Zone	O	S	S
Payload Activation Impact	O	S	S
Probing Techniques	S	S	S
Indicators-Warnings of Attack	S	S	S
Obfuscation Techniques	S	S	S
Related Attack Patterns	S	S	S
Relevant Security Requirements	S	S	S
Relevant Design Patterns	S	S	S
Relevant Security Patterns	S	S	S
Related Security Principles	S	S	S
Related Guidelines	S	S	S
R = Required S = Suggested, where appropriate O = Out of Scope			

1.5 Administrative Information

These schema fields are intended to provide information to the reader of an administrative nature for this pattern.

1.5.1 Source

This field captures at a very high level where this pattern came from and how it has been modified throughout its history.

1.5.1.1 Submission

This field describes where this pattern originally came from. There can only be one submission field for a pattern.

Submitter

This field describes the identity of the pattern submitter.

Submitter Organization

This field describes the organization of the pattern submitter.

Submission Date

This field describes the date on which the pattern was submitted to CAPEC.

Submission Comment

This field captures any relevant description or explanation of the submission.

1.5.1.2 Modification

This field describes modification milestones for this pattern after it was submitted to CAPEC.

Modifier

This field describes the identity of the pattern modifier.

Modifier Organization

This field describes the organization of the pattern modifier.

Modification Date

This field describes the date on which the pattern was modified in CAPEC.

Modification Comment

This field captures any relevant description or explanation of what was modified and why.

2 Supporting Schema Elements

The schema fields outlined here provide valuable information regarding the relevant attack patterns but are not universally applicable to all attack patterns. Attack patterns where this information is relevant and available should include the appropriate supporting schema elements.

2.1 Describing Information

These schema fields are intended to provide information to the reader that clearly and effectively describe the attacks defined by this attack pattern.

2.1.1 Injection Vector

This field describes, as precisely as possible, the mechanism and format of an input-driven attack of this type. Injection vectors must take into account the grammar of an attack, the syntax accepted by the system, the position of various fields, and the ranges of data that are acceptable. [Hoglund & McGraw 04]

2.1.2 Payload

This field describes the code, configuration or other data to be executed or otherwise *activated* as part of an injection-based attack of this type.

2.1.3 Activation Zone

This field describes the area within the target software that is capable of executing or otherwise *activating* the payload of an injection-based attack of this type. The activation zone is where the intent of the attacker is put into action. The activation zone may be a command interpreter, some active machine code in a buffer, a client browser, a system API call, etc. [Hoglund & McGraw 04]

2.1.4 Payload Activation Impact

This field is a description of the impact that the activation of the attack payload for an injection-based attack of this type would typically have on the confidentiality, integrity or availability of the target software.

2.2 Diagnosing Information

These schema fields are intended to provide information to assist the reader in identifying and diagnosing whether this type of attack may be imminent, may be currently in progress or may have occurred in the past.

2.2.1 Probing Techniques

This field describes techniques typically used to probe and reconnoiter a potential target to determine vulnerability and/or to prepare for this type of attack.

2.2.1.1 Probing Technique

This field describes an individual probing technique.

2.2.2 Indicators-Warnings of Attack

This field describes activities, events, conditions or behaviors that could serve as indicators that an attack of this type is imminent, in progress or has occurred.

2.2.2.1 Indicator-Warning of Attack

This field describes an individual indicator/warning.

2.2.3 Obfuscation Techniques

This field describes techniques typically used to disguise the fact that an attack of this type is imminent, in progress or has occurred.

2.2.3.1 Obfuscation Technique

This field describes an individual obfuscation technique.

2.3 Enhancing Information

These schema fields are intended to provide information to enhance the value and broader context of this pattern. This will include information such as relevant cross-referencing with other knowledge catalogs.

2.3.1 Related Attack Patterns

This field identifies other attack patterns that are somehow related to, dependent on, typically chained together, etc. with this pattern.

2.3.1.1 Related Attack Pattern

Related Attack Pattern ID

This field contains a unique integer identifier referencing the Attack Pattern ID field of the related pattern.

Related Attack Pattern Name

This field contains the short descriptive name of the related pattern contained in that pattern's Attack Pattern Name field..

Related Attack Pattern Relationship Type

This field defines the nature of the relationship between this pattern and the related pattern. In order to assist in normalization and classification, this field involves a selection from an enumerated list of defined relationship types. This enumerated list of choices is likely to grow as new, useful types of relationships are identified.

Enumerated Choices
More Abstract
More Detailed
Occasionally Precedes
Occasionally Follows

Relationship Description

This field provides a mechanism to provide further context and description to the nature of the relationship.

2.3.2 Relevant Security Requirements

This field identifies specific security requirements that are relevant to this type of attack and are general enough to offer opportunities for reuse.

2.3.2.1 Relevant Security Requirement

This field describes an individual relevant security requirement.

2.3.3 Relevant Design Patterns

This field identifies specific relevant design patterns that are recommended as providing resistance or resilience to this attack, or which are not recommended as they are particularly susceptible to this type of attack.

2.3.3.1 Recommended Design Pattern

This field describes an individual design pattern that is recommended due to its likelihood of increasing the software's resistance or resiliency to this type of attack.

2.3.3.2 Non-Recommended Design Pattern

This field describes an individual design pattern that is not recommended due to its likelihood of decreasing the software's resistance or resiliency to this type of attack.

2.3.4 Relevant Security Patterns

This field identifies specific security patterns that are recommended as providing resistance or resilience to this type of attack.

2.3.4.1 Relevant Security Pattern

This field describes an individual relevant security pattern.

2.3.5 Related Security Principles

This field identifies existing security principles (ideally those outlined in the Principles content area of the DHS Build Security In website) that are relevant to identifying or mitigating this type of attack.

2.3.5.1 Related Security Principle

This field describes an individual security principle.

2.3.6 Related Guidelines

This field identifies existing security guidelines that are relevant to identifying or mitigating this type of attack.

2.3.6.1 Related Guideline

This field describes an individual related guideline.

3 References

[Hoglund & McGraw 04]	Hoglund, Greg; & McGraw, Gary. *Exploiting Software: How to Break Code*. Addison-Wesley, 2004. http://www.exploitingsoftware.com
[McGraw 06]	McGraw, Gary. *Software Security: Building Security In*. Addison-Wesley, February, 2006. http;//www.buildingsecurityin.com
[Schneier 99]	Schneier, Bruce. "Attack Trees: Modeling Security Threats." Dr. Dobb's Journal, December, 1999.

For More Information

For more information about this document, contact:

Contact	Title	Organization	Phone #	Email Address
Mr. Sean Barnum	Principal Consultant	Cigital, Inc.	703 404-5762 direct 703 404-9293 main	sbarnum@cigital.com

About Cigital, Inc.

Cigital helps commercial and government clients assure software quality and improve software development processes. Our Software Quality Management (SQM) solutions drive down the cost of deploying quality software and ensuring software reliability, security and performance. Cigital's expert Consultants measure software quality by combining proprietary methodologies, tools and knowledge to perform full-lifecycle testing via a risk management framework. The resulting metrics are used to drive application readiness decisions and identify the most cost-effective areas for software process improvement. Founded in 1992, Cigital (www.cigital.com) is headquartered in Northern Virginia with additional offices in Boston.