Carnegie Mellon University
Software Engineering Institute

Insider Threat Vulnerability Assessment (ITVA)

Software Engineering Capability Area

https://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM23-0882

Table of Contents

Introduction

The insider threat vulnerability assessment was developed by staff in the CERT® Division at the Software Engineering Institute (SEI), a federally funded research and development center at Carnegie Mellon University. The assessment, which is based on hundreds of actual insider threat cases, enables organizations to gain a better understanding of insider threat and an enhanced ability to assess and manage associated risks. The assessment was designed to be completed over a period of three weeks. Week one is the pre-assessment week, where assessment team members review organization-supplied documents to become familiar with organization practices and policies. During week two, the assessment team spends three to five days onsite at an organization. During that time, the assessment team reviews documents, interviews key personnel, and observes processes to substantiate each capability. During the final week, the assessment team prepares an insider threat vulnerability assessment final report, describing how prepared an organization is to prevent, detect, and respond to insider threats.

The purpose of the software engineering capability area is to determine the degree to which the organization being assessed is vulnerable to insider threats resulting from the organization's software engineering activities.

Software engineering activities include those related to software requirements, design, implementation, testing, build, release, and deployment. The focus is not just on software developed in-house, but also on systems assembled from commercial off-the-shelf (COTS) products.

The assessment team will work with the organization being assessed to identify roles and responsibilities within the organization for addressing vulnerabilities to insider threats related to software engineering. Typical roles include software project managers, requirements engineers, developers, testers, release managers, and configuration managers.

® CERT® is a registered mark owned by Carnegie Mellon University.

Generic Clarifications

An insider is defined as any person who supports the organization, including contractors, subcontractors, and business partners.

All capabilities containing the phase ***"prevent, detect, and respond to"*** require that the organization can do all three: prevent insider threat incidents, detect incidents if they occur, and respond to incidents when they occur.

A ***policy*** is an administrative control commonly used as a prevention method. However, for an organization to achieve a capability involving a policy, the policy's existence is not sufficient on its own. The assessment team will be looking for the following attributes of a policy:

- documented
- communicated
- maintained
- routinely and consistently applied
- enforced
- monitored

Without defined policies and procedures, it can be difficult to discipline, terminate, or prosecute employees who engage in insider threat activity. To be effective, the policies and procedures must be consistently and routinely enforced.

Capability Sequence # SE1.1: Configuration Management Policy

The organization has a configuration management policy.

Clarification/Intent

A well-implemented configuration management process begins with a policy for configuration management. In particular, a configuration management policy should address the organization's expectations about access and changes to software-related work products.

Assessment Team Guidance

A lack of a configuration management policy can leave an organization without enforcement options when unauthorized access or changes to software-related work products occur.

Questions to ask include the following:

- Does a policy for configuration management exist?
- Has this policy been communicated?
- How does the organization manage repeated policy violations?
- What are the consequences of repeated policy violations?

MERIT Example

The insider was employed as a lead software engineer by the victim organization, a government entity. The insider operated an un-related side business with several co-workers. The insider led a team developing a software suite for the organization. After major issues were found with the first implementation of the software suite, the organization's management requested that the insider document all source code in order to implement configuration management and central control of development. The insider learned that future development of the suite would be outsourced and consequently he would be demoted and have his pay reduced. The demotion would also require the insider to relocate to another organizational office. To undermine the transition, the insider wrote the code in an obscure way. The insider filed a grievance and took a leave of absence. The grievance was denied, and the insider resigned. Prior to resigning, the insider copied source code and encrypted it with a password. The insider deleted the source code from his laptop, which he turned in at the time of his resignation. The insider explained that he intentionally deleted the source code, but did not disclose that he retained a copy. An investigation eventually led to discovery of the encrypted copy of the software at his home, to which he eventually provided the crypto key and admitted his guilt. The duration of the incident was 9 months. The insider was arrested, convicted, sentenced to 1 year imprisonment, and ordered to pay $13,000 in fines and restitution.

Organization Response

Evidence Sought

Auto Verification

Additional Information

Scoring Criteria

Level 1

A score of Level 1 indicates failure to meet the requirements for the higher levels.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 2

☐ A documented configuration management policy exists.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ The configuration management policy is communicated to the organization and relevant trusted business partners.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 3

☐ The configuration management policy is routinely and consistently applied.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Violations of the configuration management policy are identified and remediated.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 4

☐ The configuration management policy is monitored for effectiveness.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ The configuration management policy is reviewed for currency on a periodic basis and updated as needed.

Doc Rev

Dir Obs

Intvw

Score: ☐ **Not applicable** ☐ **1** ☐ **2** ☐ **3** ☐ **4**

Justification

Evidence Collected

Document Review		**Direct Observation**		**Interview**	

Notes (from documentation, observations, and interviews)

Capability Sequence # SE1.2: Unauthorized Changes to Software

The organization prevents, detects, and responds to unauthorized changes to software work products.

Clarification/Intent

Work products to be placed under configuration management should be identified. Examples include source code, designs, data, executables, and regression tests. Often, a tool is used for configuration management.

Configuration baselines may be needed to determine approved versions of configuration items at different milestones.

Once items are placed under configuration management, changes must be tracked and controlled.

Next, there should be some method, usually configuration audits, to verify that configuration policies and practices are being implemented and that the integrity of items under configuration management is maintained.

Finally, violations found during configuration audits should be addressed.

Assessment Team Guidance

Work products to be placed under configuration management should be identified. Questions to ask include the following:

- Which software-related work products are placed under configuration management?
- Is a tool used for configuration management?
- Are configuration baselines created?
- When is a baseline created?
- Who creates it?
- What triggers creation of a new baseline?
- If a baseline is not created, is version control used?

Once items are placed under configuration management, changes should be tracked and controlled.

Questions to ask include the following:

- How are changes and/or change requests to items under configuration management handled?
- How are changes initiated?
- How are they tracked?
- How are they validated?

Finally, there should be some method of verifying that configuration policies and practices are being implemented and that the integrity of items under configuration management is maintained.

Questions to ask include the following:

- How are configuration management practices verified?
- Are all work products identified as under change management in fact under change management
- Are configuration audits performed?
- If so, how often? How are exceptions and violations addressed?
- Have they ever had instances of unauthorized changes?
- Are people allowed to update software without going through a CM system?
- Has software ever been released with unauthorized changes?
- Is there actually any way possible to install software that has not gone through configuration control?

MERIT Example

The insider was formerly employed as chief scientist by the victim organization, a software company. The insider's accomplice was employed by the victim organization as a senior research scientist. During his employment, the insider maintained a great deal of secrecy regarding the project he was working on in order to limit the amount of information that the victim organization could obtain. Specifically, the insider failed to document source code and purposefully omitted comments from source code in order to maintain control of the source code. The insider was warned about his deficient documentation methods. The insider also maintained a copy of the project's source code on his home computer. The insider resigned and used the stolen source code to form a competitor beneficiary organization with his accomplice. The beneficiary organization was named after the project that the insider worked on at the victim organization. The insider's plan was to use his beneficiary organization to market the project's product before the victim organization was able to do so. The beneficiary organization's website offered a directly competitive product. Two months after the beneficiary organization was founded, the victim organization filed suit against the company, claiming the insiders breached their employment contracts, misappropriated trade secrets, and infringed on the victim organization's project's trademarked name. The beneficiary organization was forced to change its name after a restraining order was issued to prevent the beneficiary organization from using the project's trademarked name. The civil suit was settled out of court. The beneficiary organization agreed to a permanent injunction against using or disclosing the victim organization's trade secrets and to pay the victim organization an undisclosed sum.

Organization Response

Evidence Sought

Auto Verification

Additional Information

Scoring Criteria

Level 1

A score of Level 1 indicates failure to meet the requirements for the higher levels.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 2

☐ Items to be placed under configuration management are identified.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ All software work products are placed under configuration management.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Responsibility for configuration management of products is assigned.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Resources are allocated to perform configuration management activities.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Identified configurations are baselined.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Changes to baselines are controlled and tracked.

Doc Rev

Dir Obs

Intvw

☐ The integrity of configured items is maintained.

Doc Rev

Dir Obs

Intvw

Level 3

☐ Configuration management processes/activities are monitored and controlled.

Doc Rev

Dir Obs

Intvw

☐ Violations of configuration management policies for specific software products are addressed.

Doc Rev

Dir Obs

Intvw

☐ Configuration baselines and revisions are backed up and recoverable if needed.

Doc Rev

Dir Obs

Intvw

Level 4

☐ A configuration management plan exists.

Doc Rev

Dir Obs

Intvw

☐ Affected employees are trained in configuration management practices and/or tools.

Doc Rev

Dir Obs

Intvw

☐ Stakeholders and product owners are identified and involved in configuration management activities and decisions.

Doc Rev

Dir Obs

Intvw

☐ The configuration management process is objectively evaluated for adherence.

Doc Rev

Dir Obs

Intvw

Score: ☐ **Not applicable** ☐ **1** ☐ **2** ☐ **3** ☐ **4**

Justification

Evidence Collected					
Document Review		**Direct Observation**		**Interview**	

Notes (from documentation, observations, and interviews)

Capability Sequence # SE1.3: Unauthorized Access to High-Value Software

The organization prevents, detects, and responds to unauthorized access to high-value software assets.

Clarification/Intent

An organization should define and manage high-value software assets that are targets for unauthorized access from insiders.

Examples of high-value information assets include intellectual property (IP) such as source code, designs, and algorithms, as well as customer data used for testing. Access is usually limited to people directly involved in developing the software products.

High-value software assets must first be identified. Then, physical, technical, and administrative controls that keep these assets viable must be selected, implemented, and managed.

Assessment Team Guidance

Software assets have several attributes that should be considered. These include

- availability (accessible to authorized users whenever needed)
- confidentiality (accessible only to authorized users)
- integrity (being in the condition intended)

Questions to ask include the following:

- Are high-value software assets identified? How?
- How is user authorization determined?
- How is access controlled?
- How are availability requirements determined?
- How is it assured that only authorized changes are made to the assets? Note that configuration management policies and procedures are likely to apply here.
- What policies are in place? Examples include policies for
  - classification of software assets by sensitivity
  - disposition of software assets
  - backup and archiving of software assets
  - removal of software assets from the workplace
  - physical access to software assets
- How are attempts at unauthorized access tracked?
- How are violations addressed?
- Examples of unauthorized access include access to source code by sales personnel and physical access to production servers by software developers.
  - An example of physical control is limiting access to backup tapes.
  - An example of administrative controls is organizational policies.

An example of technical control is access control lists to the source code configuration management tool.

MERIT Example

The insider was formerly employed as a software packager for the victim organization, a software developer. As a function of his job, the insider had access to proprietary source code. The insider was terminated for undisclosed reasons. Subsequently, several of the victim organization's competitors received anonymous emails offering to sell the victim organization's source code. It remains unknown whether the insider stole the source code during his employment or retained access to the organization's network after his termination. However, the date of version of source code offered in the email preceded the insider's termination, leading to the presumption that the insider stole the source code prior to his termination. The competitor organizations reported the emails to the victim organization. The insider had signed, and violated, an intellectual property (IP) agreement. The insider was arrested, convicted, and sentenced to 9 months imprisonment followed by 15 months supervised release.

Organization Response

Evidence Sought

Auto Verification

Additional Information

Scoring Criteria

Level 1

A score of Level 1 indicates failure to meet the requirements for the higher levels.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 2

☐ Policy exists specifying access control to high-value software assets.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ High-value software assets and their security requirements are identified.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Role-based access to high-value software assets is established.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Technical access control mechanisms are established, including user and privilege management and access management.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Physical access to high-value software assets is monitored and controlled.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 3

☐ Unauthorized access is detected via access logs or other means.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Access controls on high-value software assets are consistently enforced.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Violations of access controls on high-value software assets are addressed.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ High-value software assets are backed up and replicated.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 4

☐ The disposition of high-value software assets (retiring or deleting) is controlled.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Employees working with high value software assets are trained on the importance of maintaining access controls.

Doc Rev _____

Dir Obs _____

Intvw _____

Score: ☐ **Not applicable** ☐ **1** ☐ **2** ☐ **3** ☐ **4**

Justification

Evidence Collected					
Document Review		**Direct Observation**		**Interview**	

Notes (from documentation, observations, and interviews)

Capability Sequence # SE1.4: Software Development Peer Review

The organization uses peer reviews to prevent, detect, and respond to malicious code insertion during software development.

Clarification/Intent

Software development organizations should prevent, detect, and respond to insertion of malicious code during the software development lifecycle.

Peer reviews are an important part of software verification and a proven mechanism for effective defect and malware removal.

Peer reviews involve a methodical examination of work products to identify defects and/or malware and other needed changes.

Examples of work products to be peer reviewed include designs, source code, and test cases.

Reviews include checks for malware

- back doors
- logic bombs
- disabling of integrity controls
- circumvention of role-based access

Assessment Team Guidance

The organization can show

- code review checklists
- review schedules
- defects/malware logs of defects/malware found during reviews

Peer reviews are sometimes known by alternate terms such as reviews or inspections.

MERIT Example

The insider, a non-United States native, was initially employed as a computer specialist and later as a securities trader by the victim organization, an investment bank. While employed as a computer specialist, the insider created a risk assessment program designed to aid bond traders in deciding which bonds to buy and sell. At the time of the incident, the insider was no longer involved in the organization's computer operations. For unknown reasons, the insider became angry with his bosses. The insider may have been displeased with his bonus, even though he made over $125,000 a year. The insider, motivated by revenge, inserted a logic bomb into the program he created. The logic bomb was designed to encourage customers to take precarious financial deals. The insider constructed the logic bomb so that risks would increase in tiny increments; therefore, traders would not realize their deals were getting riskier. The insider planned for the organization and its customers to lose $1 million over the course of a year. A programmer, who was trying to modify the program's code, realized that someone had tampered with the program and subsequently discovered the logic bomb. During a telephone conversation on a tapped line, the insider, speaking in a foreign language, bragged to a friend about the logic bomb. The organization was able to prevent any major damage from occurring, but spent $50,000 repairing the damage. The insider later claimed that the program was created for personal use, but contradicted this by revealing that a trader made a large profit using the insider's program. The insider was terminated, arrested, and convicted, but sentencing details are unknown.

Organization Response

Evidence Sought

Auto Verification

Additional Information

Scoring Criteria

Level 1

A score of Level 1 indicates failure to meet the requirements for the higher levels.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 2

☐ A policy specifying peer reviews for software products exist.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer reviews are conducted.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Discovered malware/defects are removed or fixed.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 3

☐ Documented peer review procedures exist.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer review results are recorded and tracked.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Fixed malware/defects are verified.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer review policy violations are addressed.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer review procedures are consistently applied and enforced

Doc Rev _____

Dir Obs _____

Intvw _____

Level 4

☐ Peer review activities are planned.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer review training is provided.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer review processes are managed.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Peer review processes are periodically reviewed for effectiveness and updated as needed.

Doc Rev _____

Dir Obs _____

Intvw _____

Score: ☐ Not applicable ☐ 1 ☐ 2 ☐ 3 ☐ 4

Justification

Evidence Collected

Document Review		**Direct Observation**		**Interview**	

Notes (from documentation, observations, and interviews)

Capability Sequence # SE1.5: Insider Threats in SDLC

The organization considers insider threats during the software development lifecycle.

Clarification/Intent

Software product development should include both functional and nonfunctional attributes that make it resistant to insider threats.

Functional attributes include the specific functions performed by the system.

Nonfunctional attributes include additional issues or characteristics such as maintainability, safety, reliability, and security. Particular nonfunctional attributes that might be included from an insider threat perspective could include but is not limited to multiparty authorization, separation of duties, and robust error handling such that insiders cannot take advantage of a system in an error state.

Software security requirements should address insider threat.

During software architecture and design, malicious exploitation of the product or service by an insider should be considered, for example, by threat modeling. Based on the results of the model, mitigating controls should be identified and included as appropriate, these can include: exception handling; role-based access; audit trail reporting; and backup, restore, and replay.

Implementation, testing, and deployment activities should ensure that insider-threat-related requirements and design are incorporated and validated.

Software product development can include acquired software applications or components (e.g., COTS, GOTS, custom developed software components) that are integrated to produce a software-reliant system. This also includes continued development or modification of the product.

Assessment Team Guidance

Insider threat should be considered throughout the software development lifecycle, including requirements, architecture, design, implementation, testing, integration, and deployment activities.

- Do requirements make the product resistant to potential insider threats?
- Are insider threats considered during requirements elicitation and development?

Is there a clear traceability (tracking and audit trail) showing that insider-threat-related requirements have been implemented and tested?

MERIT Example

To Be Supplied

Organization Response

Evidence Sought

Auto Verification

Additional Information

Scoring Criteria

Level 1

A score of Level 1 indicates failure to meet the requirements for the higher levels.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 2

☐ Insider threat is addressed during software requirements definition and testing by

Doc Rev _____

Dir Obs _____

Intvw _____

- ☐ establishing multi-party authorization requirements as appropriate

  Doc Rev _____

  Dir Obs _____

  Intvw _____

- ☐ establishing separation of duties as appropriate

  Doc Rev _____

  Dir Obs _____

  Intvw _____

- ☐ implementing audit trail reporting

  Doc Rev _____

  Dir Obs _____

  Intvw _____

- ☐ implementing backup, restore, and replay functionality for the products or services

  Doc Rev _____

  Dir Obs _____

  Intvw _____

☐ implementing exception handling alerts and robust error handling

Doc Rev _____

Dir Obs _____

Intvw _____

☐ implementing role-based access requirements as appropriate

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Insider threat is addressed during software integration and testing by identifying ways that insiders may attempt to access, modify, or damage products and systems and trying those methods as part of testing, validation, and verification.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Insider threat is addressed during software deployment by identifying ways that insiders may attempt to access, modify, or damage products and systems and trying those methods as part of the deployment.

Doc Rev _____

Dir Obs _____

Intvw _____

Level 3

☐ Insider threat is addressed in software architecture and design by

Doc Rev _____

Dir Obs _____

Intvw _____

☐ considering how the product or service might be exploited in a malicious way by an insider

Doc Rev _____

Dir Obs _____

Intvw _____

☐ threat modeling insider potential actions or behaviors in relation to the product or service

Doc Rev _____

Dir Obs _____

Intvw _____

☐ including role-based access requirements

Doc Rev _____

Dir Obs _____

Intvw _____

☐ including audit trail reporting requirements

Doc Rev _____

Dir Obs _____

Intvw _____

☐ including exception and error handling requirements

Doc Rev _____

Dir Obs _____

Intvw _____

Level 4

☐ Potential insider threat actions and behaviors are addressed in software security requirements for both in-house developed and acquired software.

Doc Rev _____

Dir Obs _____

Intvw _____

☐ Traceability is maintained from insider threat requirements through other development activities to demonstrate and verify implementation.

Doc Rev _____

Dir Obs _____

Intvw _____

Score: ☐ Not applicable ☐ 1 ☐ 2 ☐ 3 ☐ 4

Justification

Evidence Collected

Document Review		**Direct Observation**		**Interview**	

Notes (from documentation, observations, and interviews)