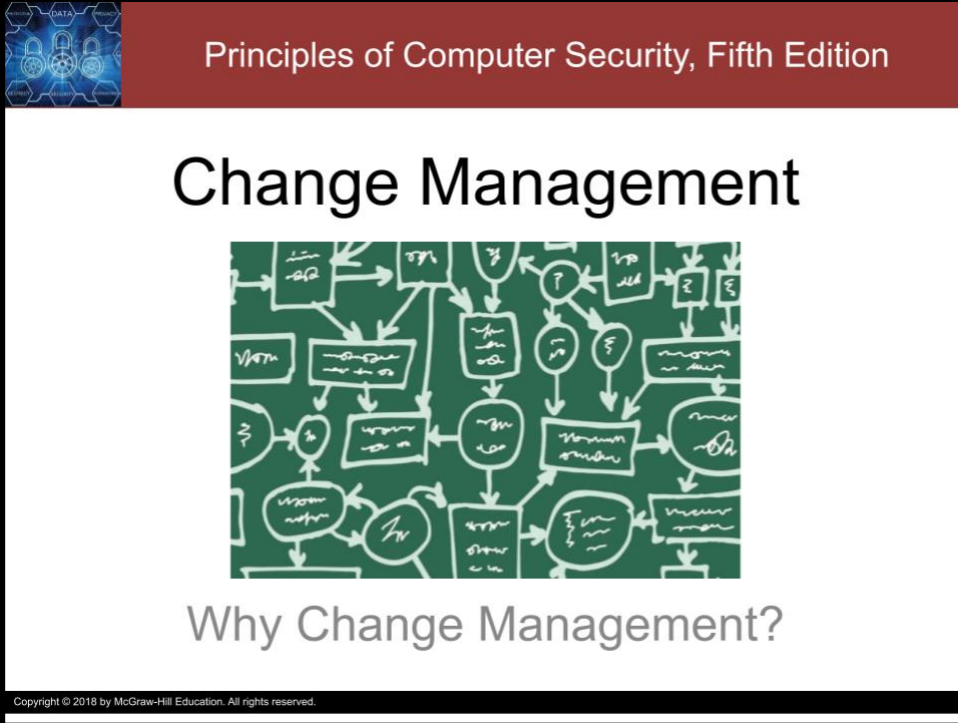


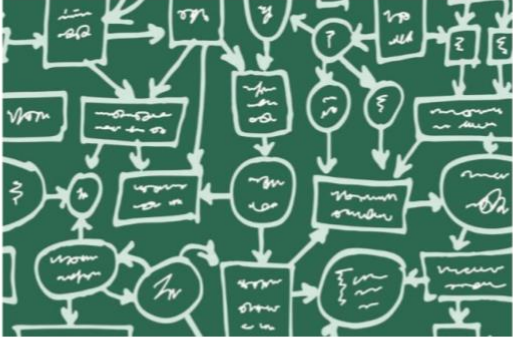
Change Management: Why change management?

Slide 1



Principles of Computer Security, Fifth Edition


Change Management



Why Change Management?

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we attempt to answer the question: Why change management?



Principles of Computer Security, Fifth Edition


Introduction

- **Change management** procedures can add structure and control to the development and management of large software systems as they move from development to implementation and during operation.
- The term **configuration management** is considered synonymous with change management and, in a more limited manner, version control or release control.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Change management procedures can add structure and control to the development and management of large software systems as they move from development to implementation and during operation. In this module, change management refers to a standard methodology for performing and recording changes during software development and system operation. The methodology defines steps that ensure that system changes are required by the organization and are properly authorized, documented, tested, and approved by management. Sometimes, the term **configuration management** is considered synonymous with change management and, in a more limited manner, version control or release control.

The term change management is often applied to the management of changes in the business environment, typically as a result of business process reengineering or quality enhancement efforts. The term change management as used in this module is directly related to managing and controlling software development, maintenance, and system operation. Configuration management is the application of change management principles to configuration of both software and hardware.



Principles of Computer Security, Fifth Edition

Why Change Management?


- Change management can be scaled to control and manage the development and maintenance of systems effectively.
- Change management should be used in all phases of a system's life:
 - Development, testing, quality assurance (QA), and production
- Change management is an essential part of creating a viable governance and control structure and is critical for compliance with the Sarbanes-Oxley Act.

Copyright © 2018, by Cengage Learning. All Rights Reserved.

To manage the system development and maintenance processes effectively, you need discipline and structure to help conserve resources and enhance effectiveness. Change management, like risk management, is often considered expensive, nonproductive, unnecessary, and confusing—basically, an impediment to progress. In some implementations, it is all of these things. But, it doesn't have to be! Like risk management, change management can be scaled to control and manage the development and maintenance of systems effectively.

Change management should be used in all phases of a system's life: development, testing, quality assurance, and production. Short development cycles, such as those used in Agile software development, have not changed the need for an appropriate amount of management control over software development, maintenance, and operation. In fact, short turnaround times make change management more necessary, because once a system goes active in today's service-oriented architecture environments, it often cannot be taken offline to correct errors—it must stay up and online or business will be lost and brand recognition damaged.

The Sarbanes-Oxley Act of 2002, officially entitled the Public Company Accounting Reform and Investor Protection Act of 2002, was enacted to help ensure management establishes viable governance environments and control structures to ensure accuracy of financial reporting. Section 404 outlines the requirements most applicable to information technology. Change management is an essential part of creating a viable governance and control structure and is critical for compliance with the Sarbanes-Oxley Act.



Principles of Computer Security, Fifth Edition

Types of Changes

- **Change** – The addition, modification, or removal of anything that could have an effect on IT Services.
 - The modification to a module to implement a new capability.
- **Standard Change** – A preapproved change that is low risk, relatively common and follows a procedure or work instruction.
 - Each month finance must make a small rounding adjustment to reconcile the General Ledger to account for foreign currency calculations.
- **Emergency Change** – A is a change that must be introduced as soon as possible.
 - Resolve a major incident or implement a security patch. The change management process will normally have a specific procedure for handling emergency changes.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

A **Change** is the addition, modification or removal of anything that could have an effect on IT Services. For example, the modification to a module to implement a new capability.


Generally speaking, there are two kinds of changes: standard and emergency.

A **Standard Change** is a pre approved change that is low risk, relatively common and follows a procedure or work instruction.

For example, each month finance must make a small rounding adjustment to reconcile the General Ledger to account for foreign currency calculations.

An **Emergency Change** is a change that must be introduced as soon as possible.

For example, to resolve a major incident or implement a security patch. The change management process will normally have a specific procedure for handling emergency changes.



Principles of Computer Security, Fifth Edition

Example Scenarios

- The developers can't find the latest version of the production source code.
- A bug corrected a few months ago mysteriously reappears.
- Development team members overwrote each other's changes.
- A programmer spent several hours changing the wrong version of the software.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Here are just a few scenarios that exemplify the need for appropriate change management policy and for procedures over software, hardware, and data:

The developers can't find the latest version of the production source code. Change management practices support versioning of software changes.


A bug corrected a few months ago mysteriously reappears. Proper change management ensures developers always use the most recently changed source code.

Development team members overwrote each other's changes. Today's change management tools support collaborative development.

A programmer spent several hours changing the wrong version of the software. Change management tools support viable management of previous software versions.

Just about anyone with experience in software development or system operations can relate to these and similar scenarios. However, each of these scenarios can be controlled, and impacts mitigated, through proper change management procedures.

Slide 6



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.

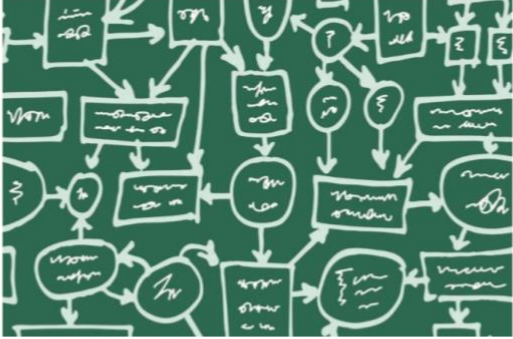
Change Management: Separation of Duties

Slide 1

DATA take care.

Principles of Computer Security, Fifth Edition


Change Management



The Key Concept: Separation of Duties

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss separation of duties, which is key to change management.



Principles of Computer Security, Fifth Edition

The Key Concept: Separation of Duties

- Involving more than one individual in a process can reduce risk.
- No one individual should be able to control all phases of a process or the processing and recording of a transaction □ **Separation of Duties**
 - discourage and prevent errors, fraud, and malicious acts.
 - establishes a basis for accountability and control
 - safeguard enterprise assets and protect against risks.
- Document, monitor, enforce.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

A foundation for change management is the recognition that involving more than one individual in a process can reduce risk.

Good business control practices require that duties be assigned to individuals in such a way that no one individual can control all phases of a process or the processing and recording of a transaction.


This is called separation of duties (also called segregation of duties).

It is an important means by which errors and fraudulent or malicious acts can be discouraged and prevented.

Separation of duties can be applied in many organizational scenarios because it establishes a basis for accountability and control.

Proper separation of duties can safeguard enterprise assets and protect against risks.

The specific segregation of duties should be documented, monitored, and enforced.




Principles of Computer Security, Fifth Edition

Example

- Management and payment of vendor invoices
- Potential for fraud
 - One person: create vendor, create invoice, authorize payment.
- Separation of duties defense
 - Two people: 1) create vendors, 2) create invoices and authorize payments
 - Fraud is more difficult – requires 2 corruptions

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

A well-understood business example of separation of duties is in the management and payment of vendor invoices. If a person can create a vendor in the finance system, enter invoices for payment, and then authorize a payment check to be written, it is apparent that fraud could be perpetrated because the person could write a check to himself for services never performed. Separating duties by requiring one person to create the vendors and another person to enter invoices and write checks makes it more difficult for someone to defraud an employer.



Principles of Computer Security, Fifth Edition

Focus on IT

- Design, implement, monitor, and enforce separation of duties for enterprise IT systems and operations.
- Rapid growth + inadequate IT controls = potential for exploitation.
- IT staff know enough to be dangerous.

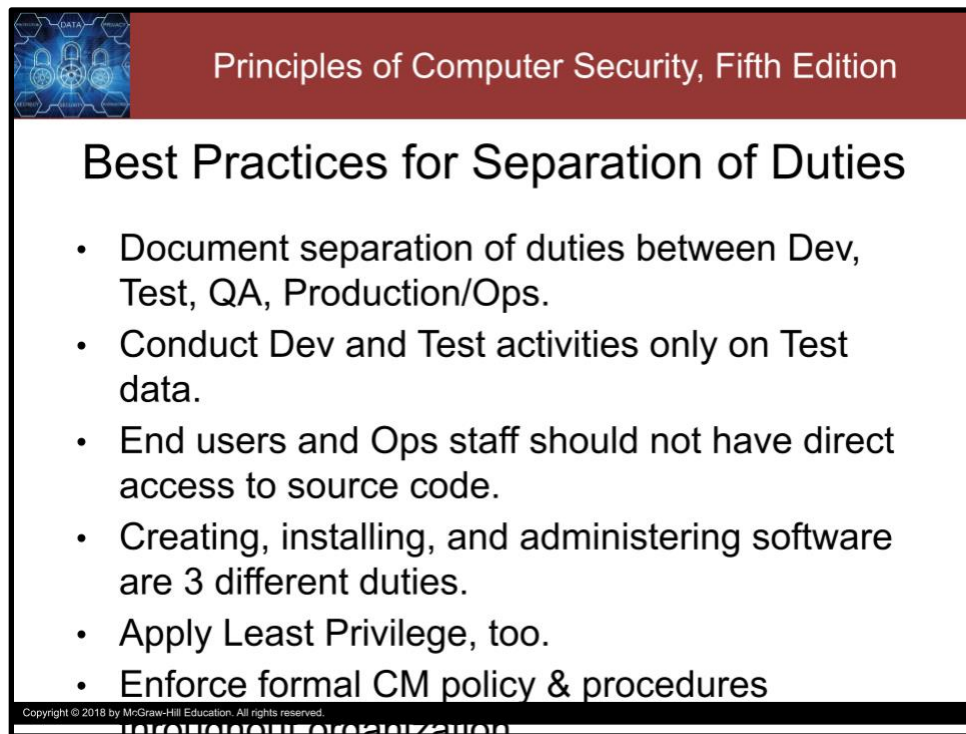
Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Information technology (IT) organizations should design, implement, monitor, and enforce appropriate separation of duties for the enterprise's information systems and processes.

Today's computer systems are rapidly evolving into an increasingly decentralized and networked computer infrastructure.

In the absence of adequate IT controls, such rapid growth may allow exploitation of large amounts of enterprise information in a short time.

Further, the knowledge of computer operations held by IT staff is significantly greater than that of an average user, and this knowledge could be abused for malicious purposes.



The slide features a blue header with the text "Principles of Computer Security, Fifth Edition" and a decorative graphic of padlocks and gears. The main content is a list of best practices for separation of duties, presented in a white box with a black border. The list includes six bullet points: documenting separation of duties, conducting activities on test data, restricting access to source code, separating creation, installation, and administration, applying least privilege, and enforcing formal CM policy.

Principles of Computer Security, Fifth Edition

Best Practices for Separation of Duties

- Document separation of duties between Dev, Test, QA, Production/Ops.
- Conduct Dev and Test activities only on Test data.
- End users and Ops staff should not have direct access to source code.
- Creating, installing, and administering software are 3 different duties.
- Apply Least Privilege, too.
- Enforce formal CM policy & procedures

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Some of the best practices for ensuring proper separation of duties in an IT organization are as follows:

Separation of duties between development, testing, QA, and production should be documented in written procedures and implemented by software or manual processes.

Program developers' and program testers' activities should be conducted on "test" data only. They should be restricted from accessing "live" production data. This will assist in ensuring an independent and objective testing environment without jeopardizing the confidentiality and integrity of production data.

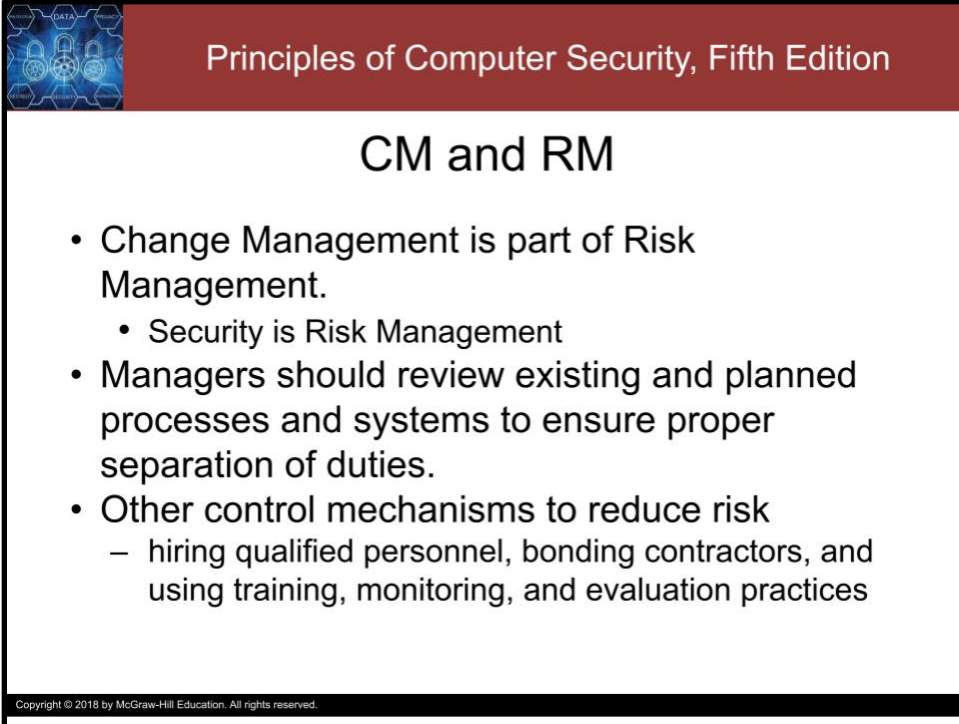
End users or computer operations personnel should not have direct access to program source code. This control helps lessen the opportunity of exploiting software weaknesses or introducing malicious code (or code that has not been properly tested) into the production environment either intentionally or unintentionally.

Functions of creating, installing, and administering software programs should be assigned to different individuals. For example, since developers create and enhance programs, they should not be able to install it on the production system. Likewise, database administrators should not be program developers on database systems they administer.

All accesses and privileges to systems, software, or data should be granted based on the principle of least privilege, which gives users no more privileges than are necessary to perform their jobs. Access privileges should be reviewed regularly to ensure that individuals who no longer require access have had their access removed.

Formal change management policy and procedures should be enforced throughout the enterprise. Any changes in hardware and software components (including emergency changes) that are implemented after the system has been placed into production must go through the approved formal change management mechanism.

Slide 6




Principles of Computer Security, Fifth Edition

CM and RM

- Change Management is part of Risk Management.
 - Security is Risk Management
- Managers should review existing and planned processes and systems to ensure proper separation of duties.
- Other control mechanisms to reduce risk
 - hiring qualified personnel, bonding contractors, and using training, monitoring, and evaluation practices

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Managers at all levels should review existing and planned processes and systems to ensure proper separation of duties. Smaller business entities may not have the resources to implement all of the preceding practices fully, but other control mechanisms, including hiring qualified personnel, bonding contractors, and using training, monitoring, and evaluation practices, can reduce any organization's exposure to risk. The establishment of such practices can ensure that enterprise assets are properly safeguarded and can also greatly reduce error and the potential for fraudulent or malicious activities.



Principles of Computer Security, Fifth Edition


CM and Separation of Duties

- Implement and enforce separation of duties
 - Add structure and management oversight to the Dev and Ops processes.
- Ensure that only correct and authorized changes are allowed to be made.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Change management practices implement and enforce separation of duties by adding structure and management oversight to the software development and system operation processes. Change management techniques can ensure that only correct and authorized changes, as approved by management or other authorities, are allowed to be made, following a defined process.

Slide 8



Principles of Computer Security, Fifth Edition

Attribution


- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.

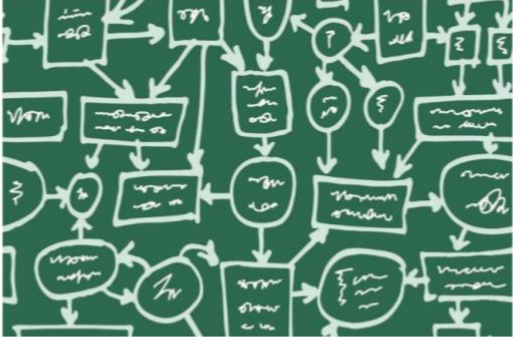
Change Management: Elements of Change Management

Slide 1



Principles of Computer Security, Fifth Edition


Change Management



Elements of Change Management

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss elements of change management.



Principles of Computer Security, Fifth Edition

Origin and Examples

- System engineering, “configuration management”.
- Modern HW and SW development require proper management structure and controls.
- See also:
 - Heartbleed:
<https://www.youtube.com/watch?v=1dOCHwf8zVQ>
 - Shellshock:
<https://www.youtube.com/watch?v=MyldPMn95kk>


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Change management has its roots in system engineering, where it is commonly referred to as configuration management.

Most of today’s software and hardware change management practices derive from long-standing system engineering configuration management practices.

Computer hardware and software development have evolved to the point that proper management structure and controls must exist to ensure the products operate as planned.

Issues such as the Heartbleed and Shellshock incidents illustrate the need to understand configurations and change.




Principles of Computer Security, Fifth Edition

Phases of Change Management

- Four general phases are defined under configuration management:
 - Configuration identification
 - Configuration control
 - Configuration status accounting
 - Configuration auditing

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Terminology may differ between perspectives, but there are four general phases of change management: Identification, control, status accounting, and auditing



Principles of Computer Security, Fifth Edition


Configuration Identification

- Identify which assets need to be managed and controlled.
 - These assets could be software modules, test cases or scripts, table or parameter values, servers, major subsystems, or entire systems.
 - These identified assets are called **configuration items** or **computer software configuration items**.
 - **Baseline** serves as a foundation for comparison or measurement and provides the necessary visibility to control change.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Configuration identification is the process of identifying which assets need to be managed and controlled. These assets could be software modules, test cases or scripts, table or parameter values, servers, major subsystems, or entire systems. The idea is that, depending on the size and complexity of the system, an appropriate set of data and software (or other assets) must be identified and properly managed. These identified assets are called **configuration items** or **computer software configuration items**.

Related to configuration identification, and the result of it, is the concept of a **baseline**. A baseline serves as a foundation for comparison or measurement. It provides the necessary visibility to control change. For example, a software baseline defines the software system as it is built and running at a point in time. As another example, network security best practices clearly state that any large organization should build its servers to a standard build configuration to enhance overall network security. The servers are the configuration items, and the standard build is the server baseline.




Principles of Computer Security, Fifth Edition

Configuration Control

- Control changes to items that have been baselined.
- Ensures
 - that only approved changes to a baseline are allowed to be implemented.
 - proper use of assets and avoids unnecessary downtime due to the installation of unapproved changes.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Configuration control is the process of controlling changes to items that have been baselined. Configuration control ensures that only approved changes to a baseline are allowed to be implemented. It is easy to understand why a software system, such as a web-based order entry system, should not be changed without proper testing and control—otherwise, the system might stop functioning at a critical time. Configuration control is a key step that provides valuable insight to managers. If a system is being changed, and configuration control is being observed, managers and others concerned will be better informed. This ensures proper use of assets and avoids unnecessary downtime due to the installation of unapproved changes.



Principles of Computer Security, Fifth Edition


Configuration Status Accounting

- Track and maintain data relative to each configuration item in the baseline.
- Closely related to configuration control.
- Involves gathering and maintaining information relative to each configuration item.
 - what changes have been requested; what changes have been made, when, and for what reason; who authorized the change; who performed the change; what other configuration items or systems were affected by the change.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Configuration status accounting consists of the procedures for tracking and maintaining data relative to each configuration item in the baseline. It is closely related to configuration control. Status accounting involves gathering and maintaining information relative to each configuration item. For example, it documents what changes have been requested; what changes have been made, when, and for what reason; who authorized the change; who performed the change; and what other configuration items or systems were affected by the change.

Returning to our example of servers being baselined, if the operating system of those servers is found to have a security flaw, then the baseline can be consulted to determine which servers are vulnerable to this particular security flaw. Those systems with this weakness can be updated (and only those that need to be updated). Configuration control and configuration status accounting help ensure that systems are more consistently managed and, ultimately in this case, the organization's network security is maintained. It is easy to imagine the state of an organization that has not built all servers to a common baseline and has not properly controlled its systems' configurations. It would be very difficult to know the configuration of individual servers, and security could quickly become weak.



Principles of Computer Security, Fifth Edition

Configuration Auditing

- Verify that the configuration items are built and maintained according to the requirements, standards, or contractual agreements.
- Two forms:
 - Functional – verify functional requirements are met
 - Physical – verify that all and only the right items are included


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Configuration auditing is the process of verifying that the configuration items are built and maintained according to the requirements, standards, or contractual agreements. It is similar to how audits in the financial world are used to ensure that generally accepted accounting principles and practices are adhered to and that financial statements properly reflect the financial status of the enterprise.

Configuration audits ensure that policies and procedures are being followed, that all configuration items (including hardware and software) are being properly maintained, and that existing documentation accurately reflects the status of the systems in operation.

Configuration auditing takes on two forms: functional and physical. A *functional configuration audit* verifies that the configuration item performs as defined by the documentation of the system requirements. A *physical configuration audit* confirms that all configuration items to be included in a release, install, change, or upgrade are actually included, and that no additional items are included—no more, no less.

Slide 8



Principles of Computer Security, Fifth Edition

Attribution

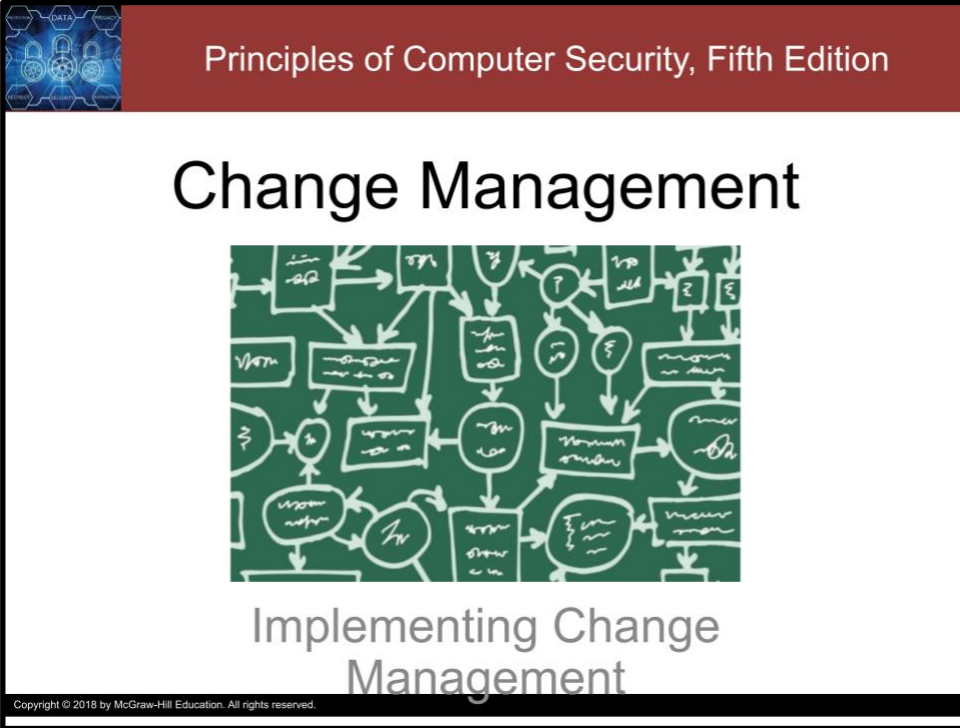
- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.

Change Management: Implementing Change Management

Slide 1




Principles of Computer Security, Fifth Edition

Change Management

Implementing Change Management

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss implementing change management.



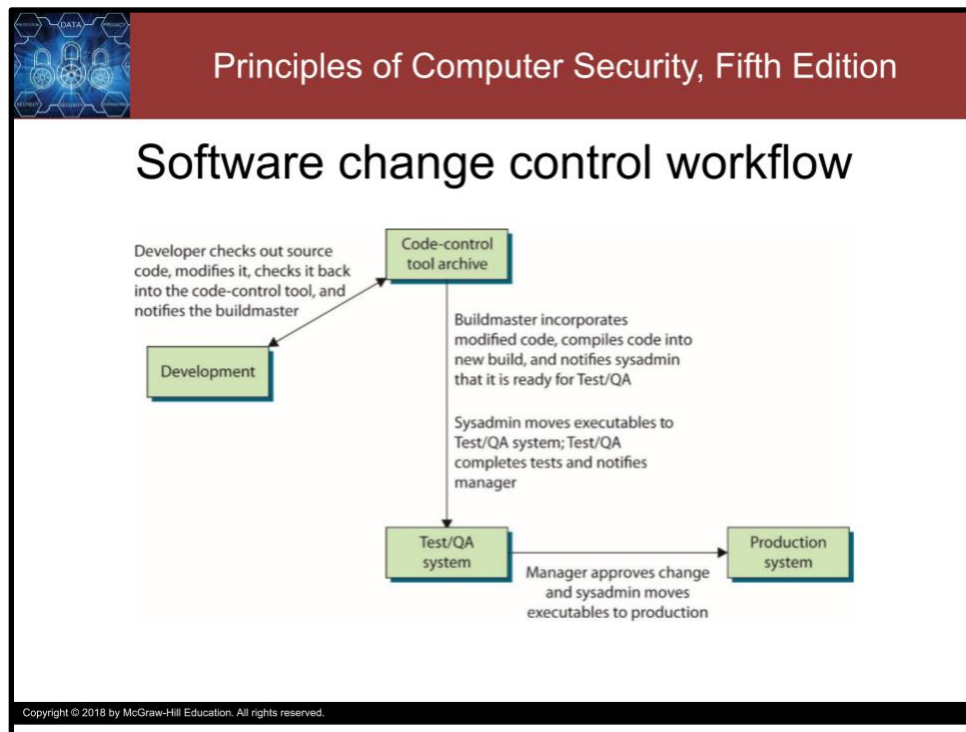
Principles of Computer Security, Fifth Edition

Implementing Change Management

- Change management requires some structure and discipline in order to be effective.

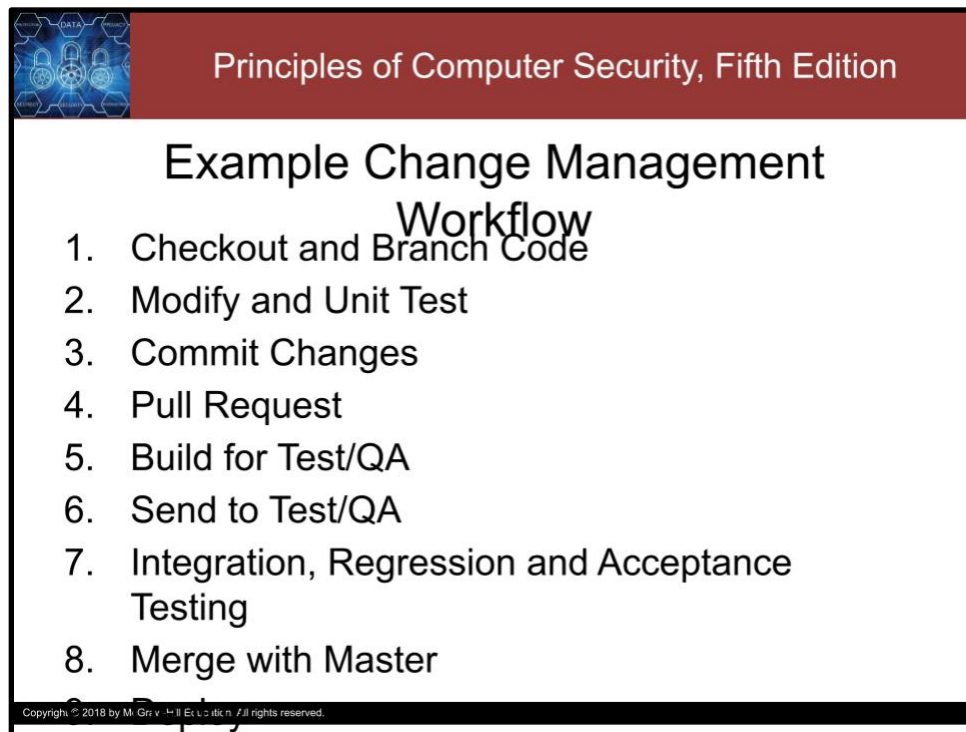
Copyright © 2018 by McGraw-Hill Education. All rights reserved.

I have heard from others that change management just gets in the way and is a headache to deal with. I understand that. There are a lot of processes that can feel that way, but, when implemented properly and followed consistently, result in higher quality work. Security processes tend to feel the most unnatural, I have found. This is why the principle of psychological acceptability is so important. If it sucks to use, people just won't use it and any security benefit it would have had is either nulled or, worse, subverted into a vulnerability. So, with that being said, effective change management requires structure and discipline. But, that does not mean implementing and adhering to a change management system has to be a headache. Or, maybe it does and you should just play a quick game of would-you-rather. Would you rather have headache or no head?



This figure illustrates a sample software change management flow appropriate for medium to large projects. It can be adapted to small organizations by having the developer perform work only on her workstation (never on the production system) and having the system administrator serve in the buildmaster function. The buildmaster is usually an independent person responsible for compiling and incorporating changed software into an executable image.

The figure also shows that developers never have access to the production system or data. It demonstrates proper separation of duties between developers, QA and test personnel, and production and therefore a distinct separation exists between development, testing and QA, and production environments. This workflow is for changes that have a major impact on production or the customer's business process. For minor changes that have minimal risk or impact on business processes, some of the steps may be omitted.



Principles of Computer Security, Fifth Edition


Example Change Management Workflow

1. Checkout and Branch Code
2. Modify and Unit Test
3. Commit Changes
4. Pull Request
5. Build for Test/QA
6. Send to Test/QA
7. Integration, Regression and Acceptance Testing
8. Merge with Master

Copyright © 2018 by M. G. K. v. 4.1. E. c. u. s. i. t. e. n. All rights reserved.

An example change management workflow proceeds as follows:

1. The developer checks out source code from the repository to the development system.
2. The developer modifies the code and conducts unit testing of the changed modules.
3. The developer checks the modified code into the repository.
4. The developer notifies the buildmaster that changes are ready for a new build and testing/QA.
5. The buildmaster creates a build incorporating the modified code and compiles the code.
6. The buildmaster notifies the system administrator that the executable image is ready for testing/QA and the system administrator moves the executables to the test/QA system.
7. QA tests the new executables.
8. If the tests are passed, test/QA notifies the manager. If tests fail, the process starts over. On success, the changes are merged into the trunk (main code branch).
9. Upon manager approval, the latest version is deployed to the production environment.



Principles of Computer Security, Fifth Edition

Backout Plan

- Key element of a change plan.
- Restore system to operational condition.
- Ultimate backout = complete backup
 - Can be expensive for big/complicated systems
 - Cheap and easy for code (a bunch of text files)


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

What happens if, in the middle of a change, something goes wrong? There should be a plan for that scenario, as it will almost inevitably happen at least once. One of the key elements of a change plan is a comprehensive backout plan. If, in the course of a planned change activity in production, a problem occurs that prevents going forward, it is essential to have a backout plan to restore the system to its previous operating condition.

A common “feature”, so to speak, of many operating system updates is the inability to go back to a previous version. This is fine provided that the update goes perfectly, but if for some reason it fails, what then? For a personal device, there may be some inconvenience. For a server in production, this can have significant business implications.

The ultimate in backout plans is the restoration of a complete backup of the system. Backups can be time consuming and difficult in some environments, but the spread of virtualization into the enterprise provides many more options in configuration management and backout plans. For code projects, keeping a copy of the last working build, even a complete history of all previous production builds, is cheap and easy.

Slide 6



Principles of Computer Security, Fifth Edition

Attribution

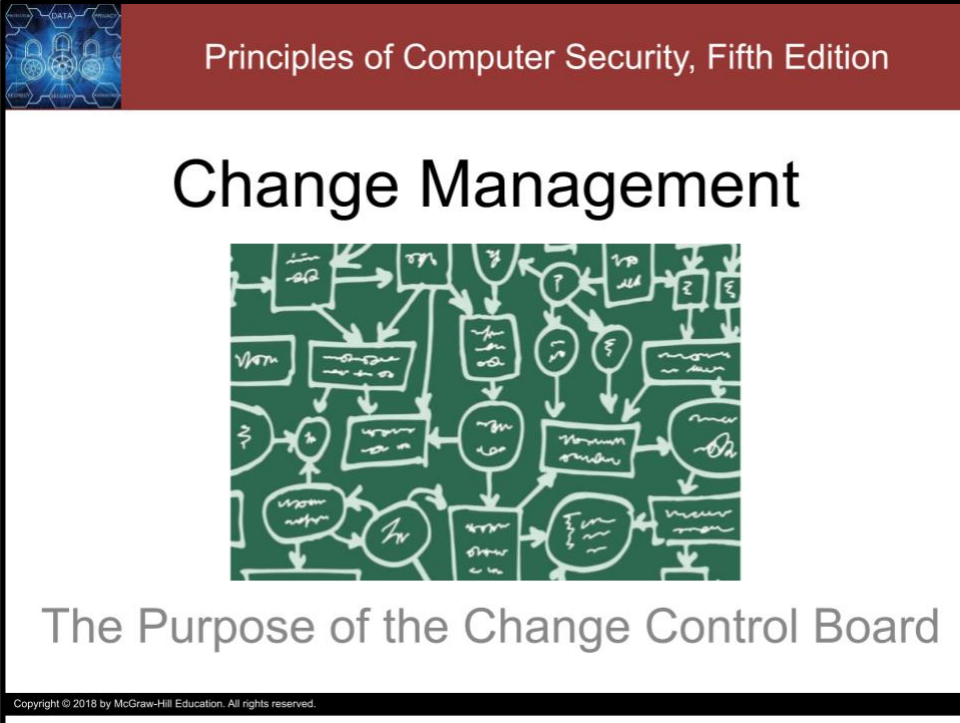
- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.

Change Management: The Purpose of the Change Control Board

Slide 1



The image shows the cover of the book "Principles of Computer Security, Fifth Edition". The top left corner features a blue graphic with the word "DATA" and several padlocks. The top right corner has a dark red banner with the text "Principles of Computer Security, Fifth Edition" in white. The main title "Change Management" is centered in a large black font. Below the title is a green chalkboard filled with a complex, hand-drawn flowchart of interconnected boxes and circles, representing a system or process. At the bottom of the chalkboard, the subtitle "The Purpose of the Change Control Board" is written in a light grey font. A small copyright notice is visible at the very bottom left of the slide.


Principles of Computer Security, Fifth Edition

Change Management

The Purpose of the Change Control Board

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss the purpose of the change control board.



Principles of Computer Security, Fifth Edition

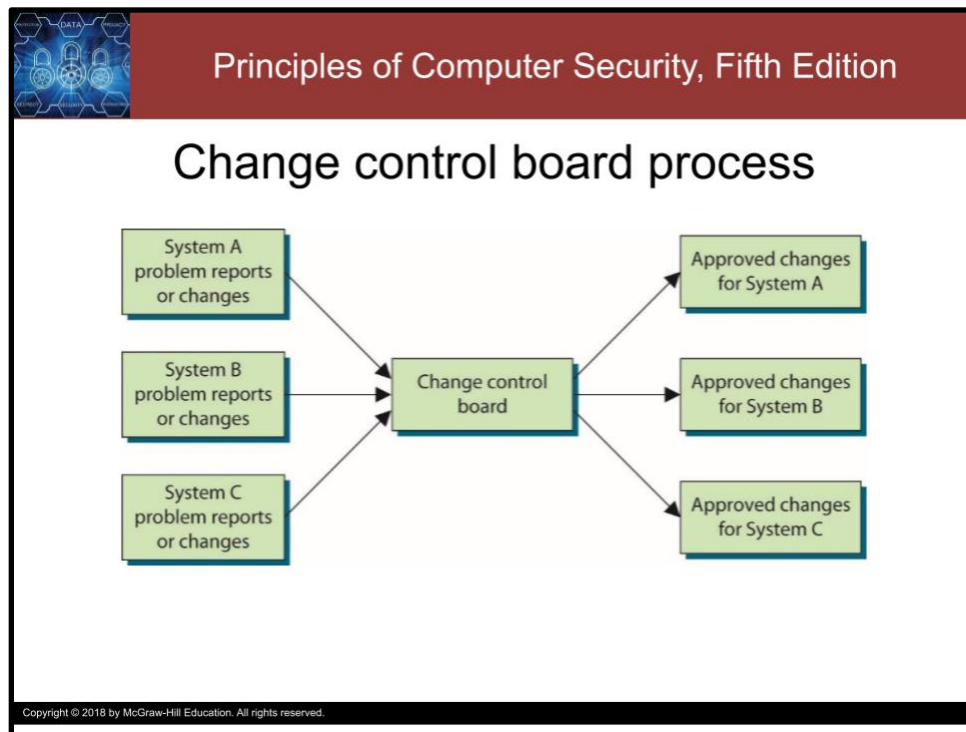
The Purpose of a Change Control Board (CCB)

- Oversee the change management process
- Also facilitates better coordination between projects.
- Convenes on a regular basis
 - can be convened for emergency or as-needed.
- Membership = {Dev project managers, Network admins, sysadmins, test/QA managers, InfoSec manager, Ops manager, helpdesk manager}

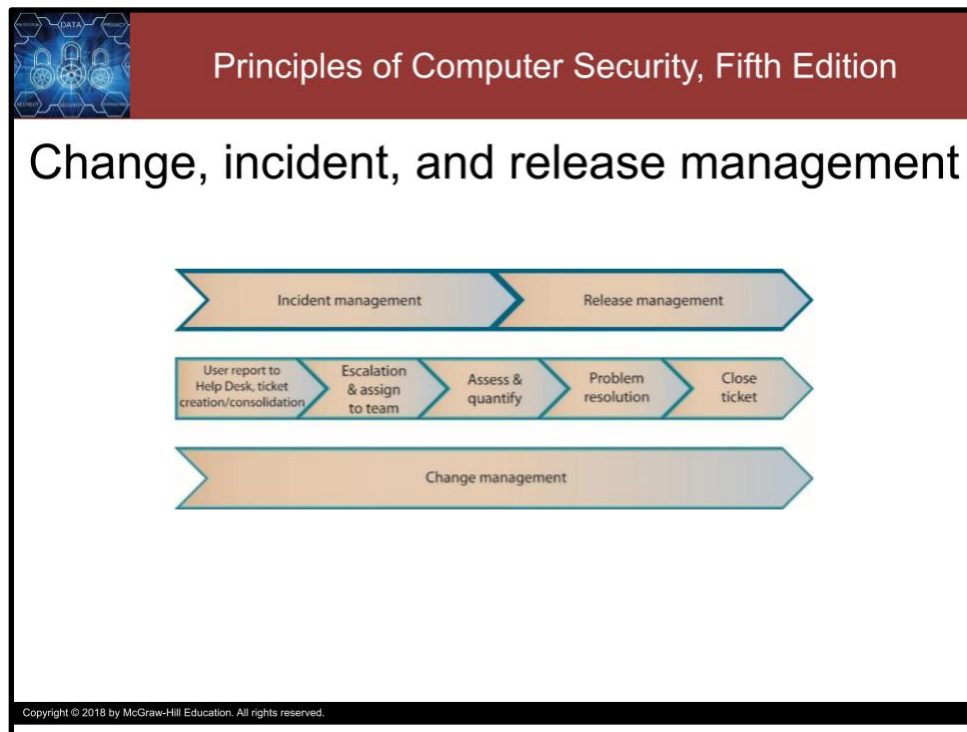
Copyright © 2018 by McGraw-Hill Education. All rights reserved.

To oversee the change management process, most organizations establish a **change control board (CCB)**. In practice, a CCB not only facilitates adequate management oversight, but also facilitates better coordination between projects. The CCB convenes on a regular basis, usually weekly or monthly, and can be convened on an emergency or as-needed basis as well.

The CCB's membership should consist of development project managers, network administrators, system administrators, test/QA managers, an information security manager, an operations center manager, and a help desk manager. Others can be added as necessary, depending on the size and complexity of the organization.




This figure shows the process for implementing and properly controlling hardware or software during changes. The CCB uses standard documents, such as change requests, in concert with business schedules and other elements of operational data, with a focus on system stability. The CCB also ensures that all elements of the change policy have been complied with before approving changes to production systems.



This figure shows the entire change management process and its relationship to incident management and release management.

As you can see, change management is happening continuously throughout the incident and release management processes, from the initial report and ticket creation, through escalation, assessment, resolution, and ticket close out.



Principles of Computer Security, Fifth Edition

Code Integrity


- CM □ code consistency and integrity assurance
- Automated CM systems simplify deployment
 - better control for ensuring executable and source-code integrity
- Code integrity is critical
 - Modifications can introduce vulnerabilities
 - Verified by host-based IDS
 - Use hashing to detect tampering

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

One key benefit of adequate change management is the assurance of code consistency and integrity. Whenever a modified program is moved to the production source-code library, the executable version should also be moved to the production system. Automated change management systems greatly simplify this process and are therefore better controls for ensuring executable and source-code integrity. Remember that at no time should the user or application developer have access to production source and executable code libraries in the production environment.

In today's networked environment, the integrity of the executable code is critical. A common hacking technique is to replace key system executable code with modified code that contains backdoors, allowing unauthorized access or functions to be performed. Executable code integrity can be verified using host-based intrusion detection systems. These systems create and maintain a database of the size and content of executable modules. Conceptually, this is usually done by computing a secure hash of the executable modules and storing the results in a database. The operation is performed on a regular schedule against the executable modules, and the results are compared to the database to identify any unauthorized changes that may have occurred to the executable modules.

Slide 7



Principles of Computer Security, Fifth Edition

Attribution


- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care!

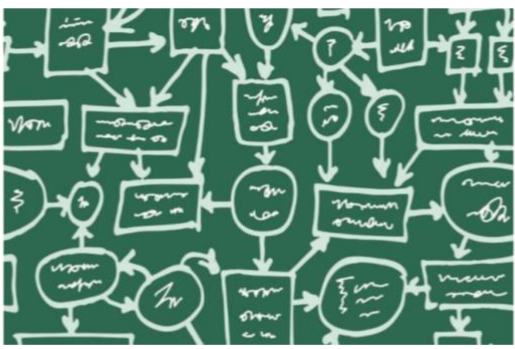
Change Management: The Capability Maturity Model Integrations

Slide 1



Principles of Computer Security, Fifth Edition


Change Management



The Capability Maturity Model Integrations

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss the Capability Maturity Model Integration for development.



Principles of Computer Security, Fifth Edition

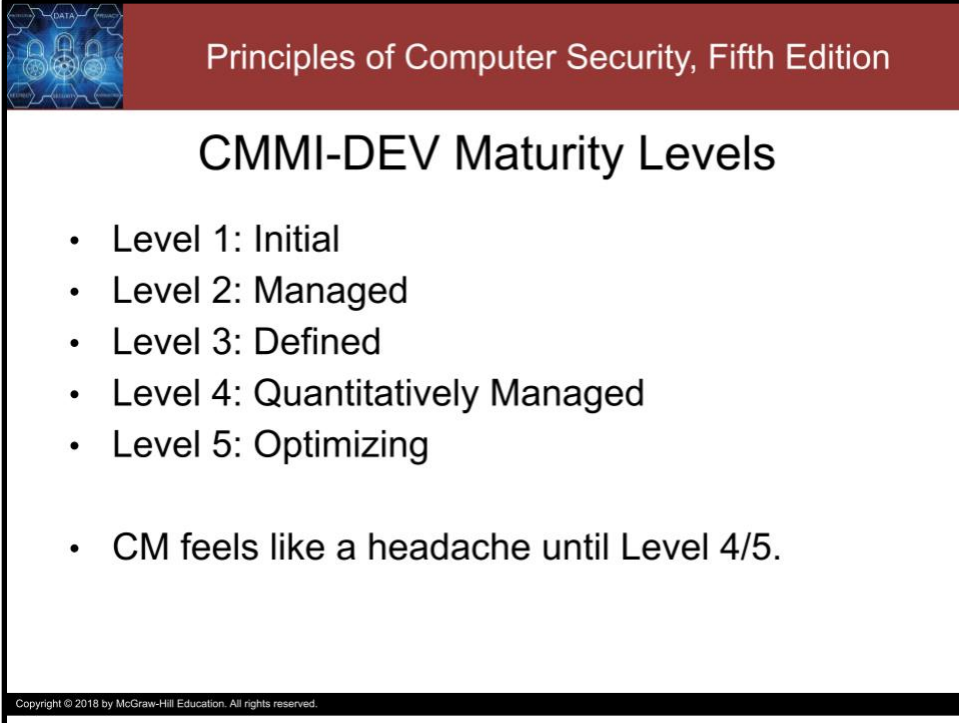
The Capability Maturity Model Integrations (CMMI)

- Important set of process models (developed by SEI)
- Three CMMIs
 - Capability Maturity Model Integration for Acquisition (CMMI-ACQ)
 - **Capability Maturity Model Integration for Development (CMMI-DEV)**
 - Capability Maturity Model Integration for Services (CMMI-SVC)
- CM is fundamental to CMMI-DEV

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The **Capability Maturity Model Integrations (CMMI)** are an important set of process models. They were developed at Carnegie Mellon University's Software Engineering Institute (SEI). SEI created three capability maturity model integrations: The Capability Maturity Model Integration for Acquisition (CMMI-ACQ), The Capability Maturity Model Integration for Development (CMMI-DEV), And the Capability Maturity Model Integration for Services (CMMI-SVC). We'll look at CMMI-DEV, which is representative of the three models.

One of the fundamental concepts of CMMI-DEV is configuration or change management, which provides organizations with the ability to improve their software and other processes by providing an evolutionary path from ad hoc processes to disciplined management processes



Principles of Computer Security, Fifth Edition

CMMI-DEV Maturity Levels

- Level 1: Initial
- Level 2: Managed
- Level 3: Defined
- Level 4: Quantitatively Managed
- Level 5: Optimizing

- CM feels like a headache until Level 4/5.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The CMMI-DEV defines five maturity levels:

Level 1: Initial – At maturity level 1, processes are generally ad hoc and chaotic. The organization does not provide a stable environment to support processes.

Level 2: Managed – At maturity level 2, processes are planned and executed in accordance with policy. The projects employ skilled people who have adequate resources to produce controlled outputs; involve relevant stakeholders; are monitored, controlled, and reviewed; and are evaluated for adherence to their process descriptions.

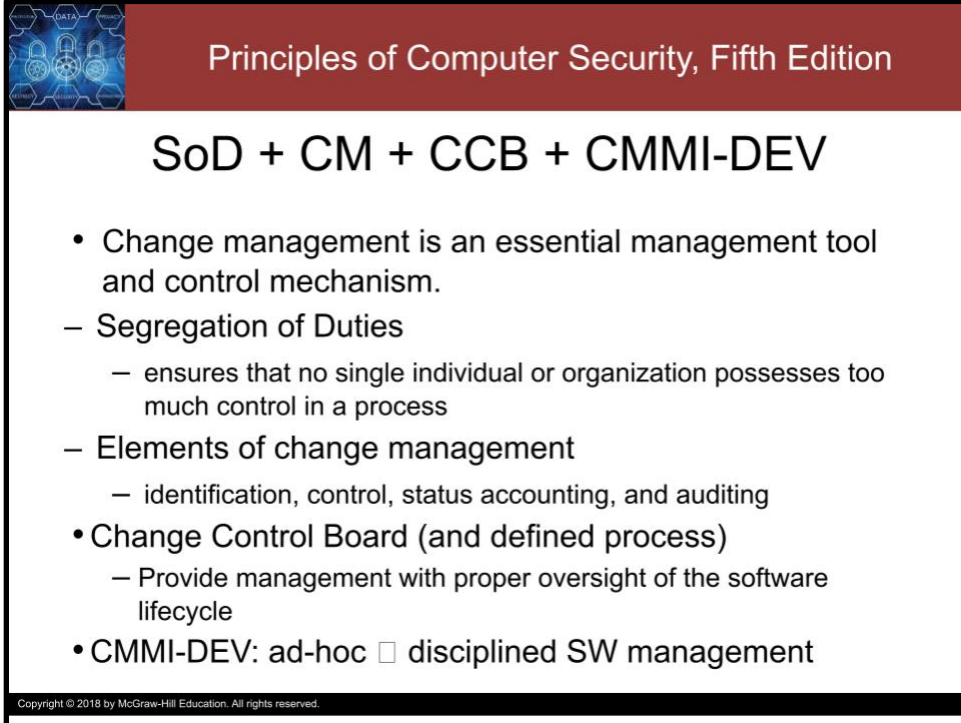
Level 3: Defined – At maturity level 3, processes are well characterized and understood, and are described in standards, procedures, tools, and methods. These standard processes are used to establish consistency across the organization.

Level 4: Quantitatively Managed – At maturity level 4, the organization establishes quantitative objectives for quality and process performance and uses them as criteria in managing projects. Quantitative objectives are based on the needs of the customer, end users, organization, and process implementers. Quality and process performance is understood in statistical terms and is managed throughout the life of projects.

Level 5: Optimizing – At maturity level 5, an organization continually improves its processes based on a quantitative understanding of its business objectives and performance needs. The organization uses a quantitative approach to understanding the variation inherent in the process and the causes of process outcomes.

Change management is a key process to implementing the CMMI-DEV in an organization. For example, if an organization is at CMMI-DEV level 1, it probably has minimal formal change management processes in place. At level 3, an organization has a defined change management process that is followed consistently. At level 5, the change management process is a routine, quantitatively evaluated part of improving software products and implementing innovative ideas across the organization. For an organization to manage software development, operation, and maintenance, it should have effective change management processes in place.

Slide 4



Principles of Computer Security, Fifth Edition

SoD + CM + CCB + CMMI-DEV

- Change management is an essential management tool and control mechanism.
- Segregation of Duties
 - ensures that no single individual or organization possesses too much control in a process
- Elements of change management
 - identification, control, status accounting, and auditing
- Change Control Board (and defined process)
 - Provide management with proper oversight of the software lifecycle
- CMMI-DEV: ad-hoc disciplined SW management


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Change management is an essential management tool and control mechanism.

Segregation of duties ensures that no single individual or organization possesses too much control in a process, which helps prevent errors and fraudulent or malicious acts

The elements of change management—configuration identification, configuration control, configuration status accounting, and configuration auditing—coupled with a defined process and a change control board, will provide management with proper oversight of the software lifecycle.

Once process and management oversight exists, the company can use CMMI-DEV to move from ad hoc activities to a disciplined software management process.



Principles of Computer Security, Fifth Edition

Attribution

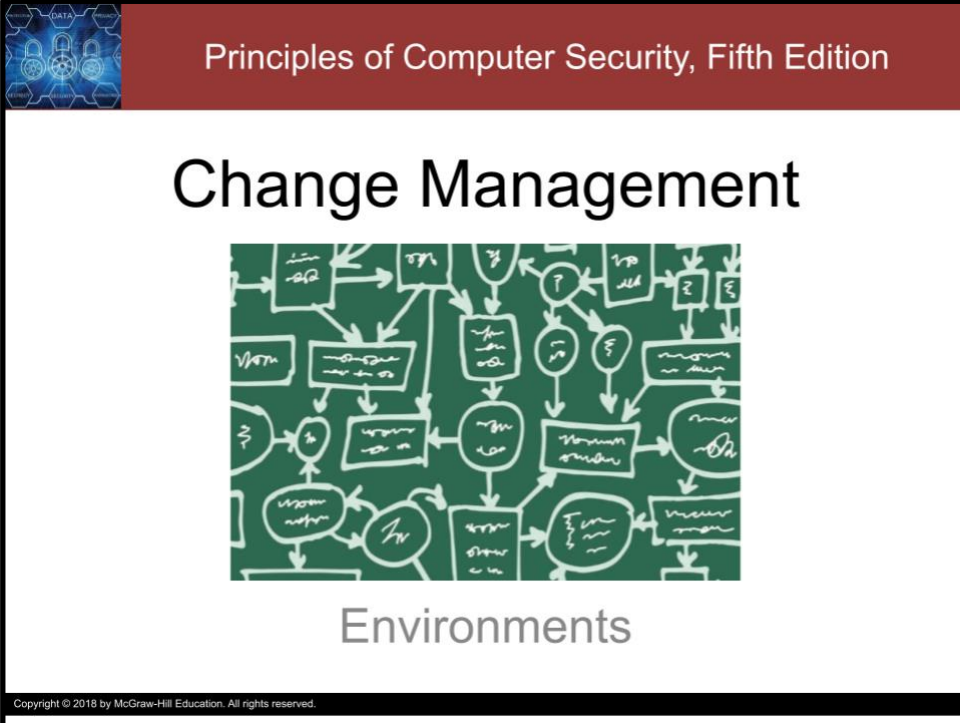
- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care!


Change Management: Environments

Slide 1



The image shows the cover of the book "Principles of Computer Security, Fifth Edition". The top left corner features a blue graphic with the word "DATA" and several padlocks. The top right corner has a dark red banner with the book title in white text. The main title "Change Management" is centered in a large black font. Below it is a green chalkboard-style diagram with white hand-drawn boxes, circles, and arrows representing a complex process flow. The word "Environments" is written in a light grey font below the diagram. At the bottom left, there is a small copyright notice: "Copyright © 2018 by McGraw-Hill Education. All rights reserved."

Howdy! In this video, we discuss the Dev, Test, Staging, and Production environments.




Principles of Computer Security, Fifth Edition

Environment

- Modern environments contain multiple, separate environments.
 - Designed to isolate development, test, and production functions
 - Prevents accidents arising from untested code ending up in production.
 - Segregated by access control list and hardware
- Special accounts that can access both are used to move code between environments

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Within a modern environment, there are multiple, separate environments designed to isolate development, test, and production functions. These are primarily to prevent accidents arising from untested code ending up in production. These environments are segregated by access control lists and hardware to prevent users from accessing multiple different levels of the environment. For moving the code between environments, special accounts that can access both are used to avoid contamination.




Principles of Computer Security, Fifth Edition

Development (Dev)

- Dev systems are sized, configured, and set up for developers to develop.
 - HW does not have to scale like production
 - Does not need to be as responsive as production
 - Needs to be the same type of system as production.
- After code is developed, it is moved to a test system.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The development system is one that is sized, configured, and set up for developers to develop applications and systems. Development hardware does not have to scale like production and does not need to be as responsive as production HW used in production. To avoid headaches, the development platform ought to be of the same type of system as the production environment. After code is developed, it is moved to a test system. Modern development practices have begun incorporating test into development. The model of developers developing then testers testing, throwing the code back and forth over a wall between different teams, is inefficient. Nonetheless, whoever is testing, whenever they test, it is done in a test environment, NOT in the dev environment.



Principles of Computer Security, Fifth Edition

Test


- Fairly closely mimics Production.
 - same versions of software, down to patch level; same sets of permissions, file structures, and so on.
 - may not scale like production.
 - SW/HW footprint looks exactly like production.
- Enables a system to be fully tested prior to being deployed into production.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The test environment is one that fairly closely mimics the production environment. It has the same versions of software, down to patch levels, and the same sets of permissions, file structures, and so on.

The test environment may not scale like production, but, from the perspective of the software/hardware footprint, it will look exactly like production.

The purpose of the test environment is to enable a system to be fully tested prior to being deployed into production.




Principles of Computer Security, Fifth Edition

Staging

- Optional environment.
 - Common with multiple production environments
- After test, then staging, then deploy to production.
- Sandbox after testing / before production.
- Staged deployment.
 - SW deployed to part of the enterprise and then the process is paused to watch for unforeseen problems.
 - Limits impact of errors discovered in production

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The staging environment is an optional environment. It is commonly found when there are multiple production environments. After passing testing, the system moves into staging. It can then be deployed to the different production systems. The primary purpose of staging is as a sandbox between testing and production. One method of deployment is a staged deployment where software is deployed to part of the enterprise and then the process is paused to watch for unforeseen problems. If there are no problems, deployment continues to update more and more production systems. This helps to limit the impact of errors that are only revealed once in production (which is also why the test environment tries to be as similar to production as possible).



Principles of Computer Security, Fifth Edition


Production

- Where the systems work with real data, doing the business that the system is supposed to perform.
- Virtually no changes except as approved through the system's change management process.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Production is the environment where the systems work with real data, doing the real business that the real system is supposed to really perform.

This is an environment where there are by design virtually no changes, except as approved and tested through the system's change management process.



Principles of Computer Security, Fifth Edition


Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

How many testers does it take to change a light bulb?

None. Testers do not fix problems; they just find

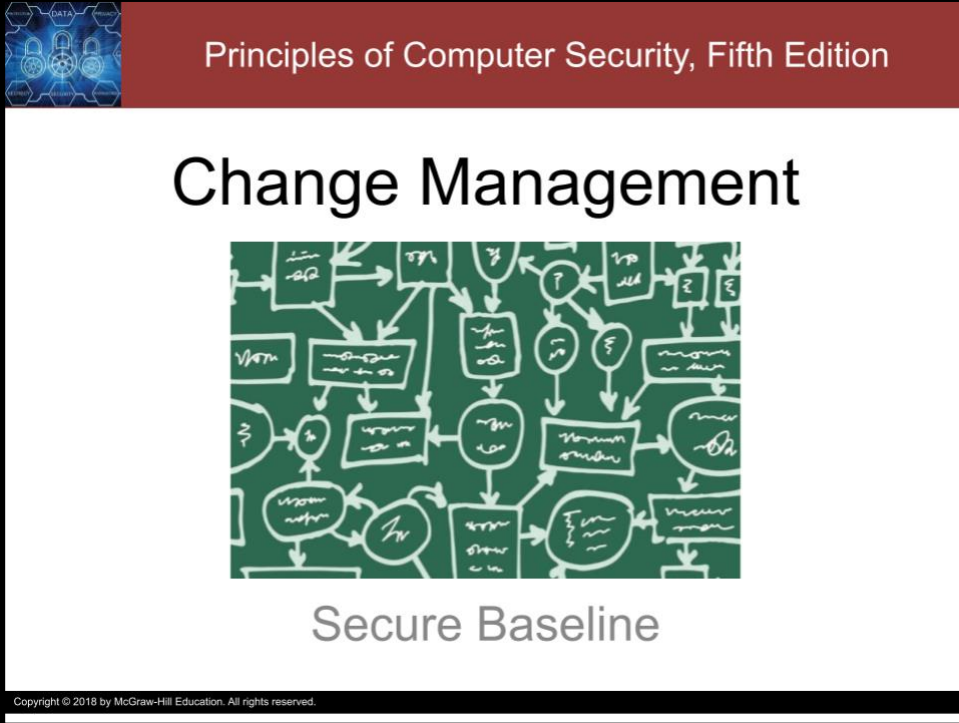
Copyright © 2018 by McGraw-Hill Education. All rights reserved.



Thank you and take care.

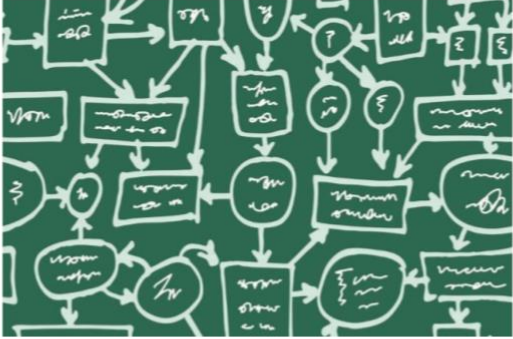
Change Management: Secure Baseline

Slide 1



Principles of Computer Security, Fifth Edition


Change Management



Secure Baseline

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we introduce baselining.



Principles of Computer Security, Fifth Edition

Secure Baseline

- A reproducible configuration with a known (and maximal) security level.
- Creating a secure baseline:
 - Anything that is not required for operations should be removed or disabled on the system
 - All appropriate patches, hotfixes, and settings should be applied to protect and secure it

Copyright © 2018 by McGraw-Hill Education. All rights reserved.


To secure the software on a system effectively and consistently, you must take a structured and logical approach.

This starts with an examination of the system's intended functions and capabilities to determine what processes and applications will be housed on the system.

As a best practice, anything that is not required for operations should be removed or disabled on the system.

Then all the appropriate patches and configurations should be applied to protect and secure it.

This becomes the system's secure baseline.



Principles of Computer Security, Fifth Edition

Baselining

- Establish software's base security state.
- Allows the software to run safely and securely.
- SW & HW must be considered together.
- Once done for a particular SW+HW, can be reproduced on similar systems
- Uniform baselines are critical for managing large numbers of systems.


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Terminology may differ between perspectives, but there are four general phases of change the process of establishing software's base security state is called baselining and the resulting product is a security baseline that allows the software to run safely and securely.

When it comes to security, software and hardware can be interdependent and so must be considered together.

Once a particular SW+HW combination has been baselined, similar systems can be configured with the same baseline to achieve the same level of security.

Uniform baselines are critical in large-scale operations because maintaining separate configurations and security levels for many machines is very resource intensive.



Principles of Computer Security, Fifth Edition

Initial Baseline Configuration

- A secure state and reference point
- Establishes a reference to help keep the system secure.
- Can be used as a template when similar systems and network devices are deployed.


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

After administrators have finished patching, securing, and preparing a system, they often create an initial baseline configuration.

This represents a secure state for the system or network device and a reference point for the software and its configuration.

This information establishes a reference that can be used to help keep the system secure by establishing a known-safe configuration.

If this initial baseline can be replicated, it can also be used as a template when similar systems and network devices are deployed.



Principles of Computer Security, Fifth Edition

Attribution

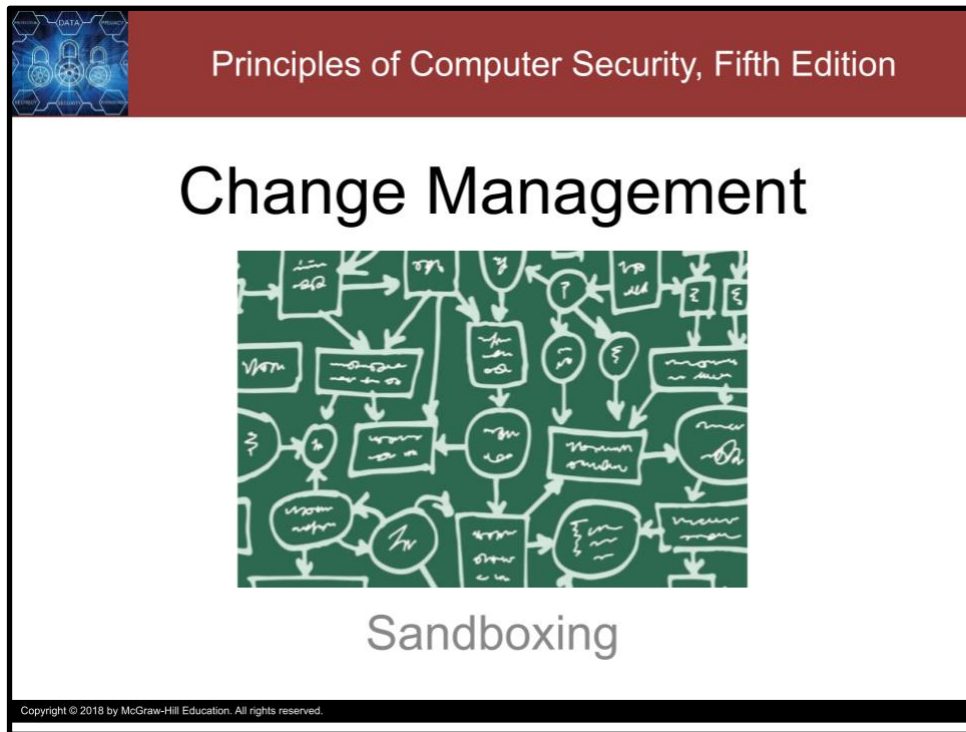
- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.

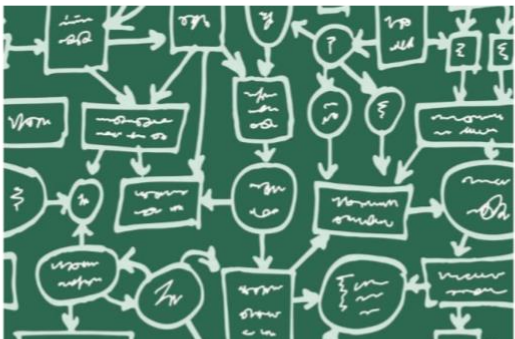
Change Management: Sandboxing

Slide 1



Principles of Computer Security, Fifth Edition


Change Management



Sandboxing

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we briefly discuss sandboxing in relation to change management.



Principles of Computer Security, Fifth Edition

Sandboxing

- Isolating a system from its surroundings.
- Standard practice to run programs with an increased risk surface inside a sandbox
 - Limit the interaction with the CPU, memory, and other processes.
 - Protect system and other processes
- Change Management Utility: configuration control and auditing
- Sandbox all the things with virtualization/containers
 - Beware of malware that watches the watcher

Copyright © 2018 by McGraw-Hill Education. All rights reserved. <http://www.ces-os.org/>

Sandboxing refers to the quarantine or isolation of a system from its surroundings.

It has become standard practice for some programs with an increased risk surface to operate within a sandbox, limiting the interaction with the CPU, memory, and other processes, which prevents aberrant behavior from affecting the OS or other programs on the system.

With respect to change management, sandboxing is useful for the configuration control and auditing phases.

Virtualization and containers can be used as a form of sandboxing with respect to an entire system.


You can build a VM or container, test something inside it, and, based on the results, make a decision with regard to security, stability, or whatever concern was present.

This is not a foolproof method because it is possible for malware to detect that they are in a VM or a container and choose not to misbehave while being watched so closely. So, a solution which is becoming quite popular is to run all applications in their own sandboxes.

Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

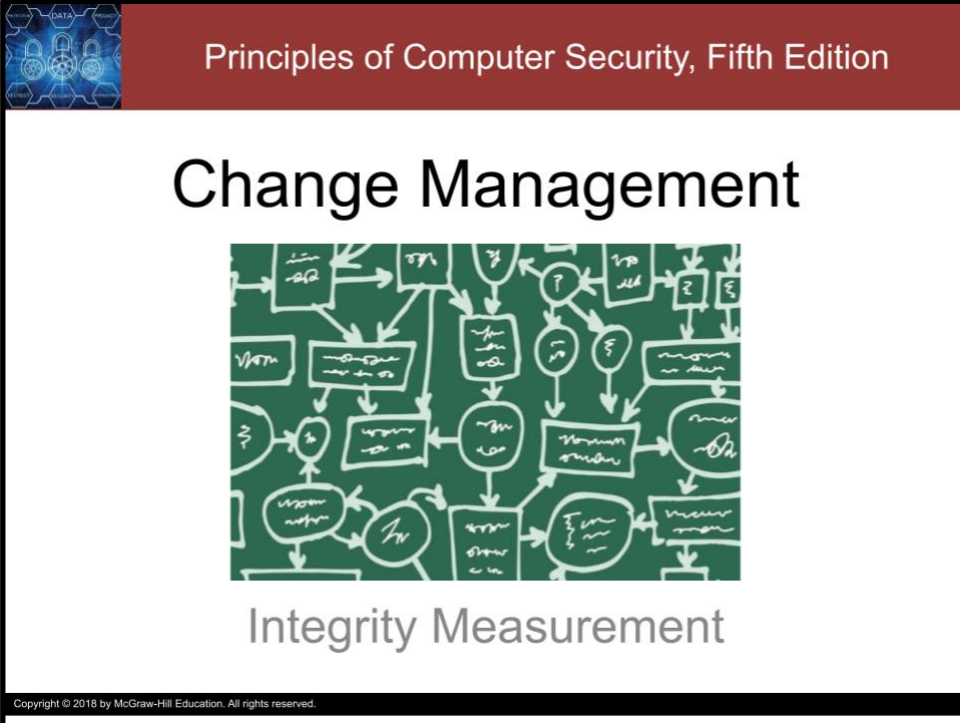


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.

Change Management: Integrity Measurement

Slide 1



Principles of Computer Security, Fifth Edition


Change Management

Integrity Measurement

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The image shows the cover of the textbook 'Principles of Computer Security, Fifth Edition'. The top section is a dark red banner with the title in white. Below this is a white area with the main title 'Change Management' in large black font, followed by a green square containing a white flowchart diagram. The diagram consists of numerous interconnected boxes and circles, some containing mathematical symbols like sigma, pi, and lambda, representing a complex system or process. Below the diagram is the subtitle 'Integrity Measurement' in a smaller grey font. At the very bottom, a small black bar contains the copyright notice.

Howdy! In this video, we briefly introduce integrity measurement for change management.



Principles of Computer Security, Fifth Edition

Integrity Measurement


- Detecting changes to a system away.
- Hashing □ TPM secure boot
 - TPM = Trusted Platform Module
- Compare observed behavior to expected behavior
 - Behavior could be static data
 - Deviation indicates loss of integrity
- Change Management utility: configuration audit
- How Windows 10 uses TPM
 - <https://docs.microsoft.com/en-us/windows/security/information-protection/tpm/how-windows-uses-the-tpm>

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Integrity measurement is the measuring and identification of changes to a specific system away from an expected value.

From the simple changing of data as measured by a hash value to the TPM-based integrity measurement of the system boot process and attestation of trust, the concept is the same: securely store a hash or other keyed value representing expected behavior, and then at the time of concern, take a measurement and calculate the appropriate function and compare the two values. If the values mismatch (or are too far apart), then something bad probably happened and the system might not be in a safe state.

In the case of a TPM-mediated system, where the TPM chip provides a hardware-based root of trust anchor, the system is specifically designed to calculate hashes of a system and store them in the Platform Configuration Register (PCR). This register can be read later and compared to a known or expected value. If the values differ, that means there is a trust violation. Certain BIOSs, UEFIs, and boot loaders can all work with the TPM chip in this manner, providing a means of establishing a trust chain during system boot. This is useful for the configuration audit phase of change management.



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.