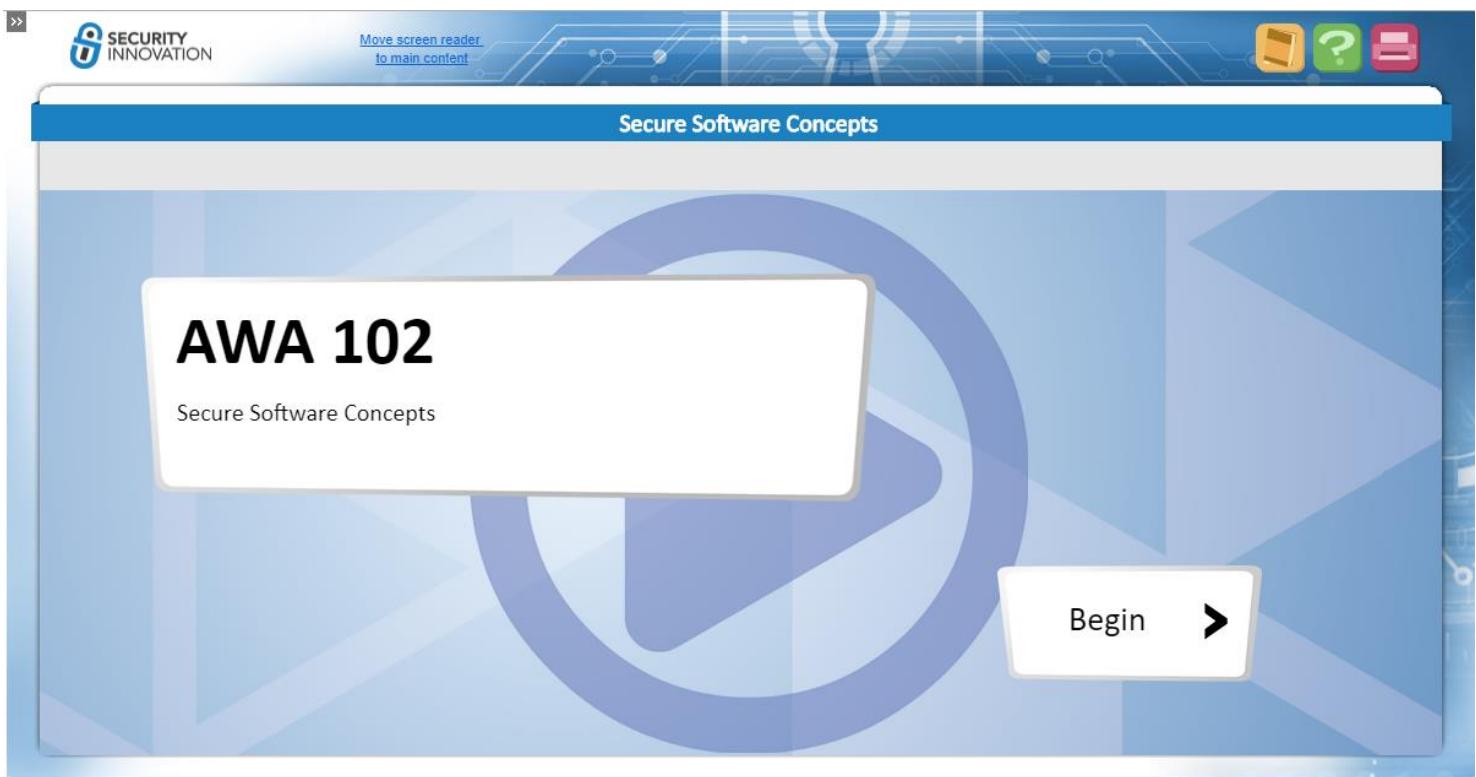


# Secure Software Concepts

---



## Narration

## On Screen Text

**AWA 102**

Secure Software Concepts

# Secure Software Concepts

## Course Overview and Objectives

The screenshot shows a course interface titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A blue banner across the top includes a "Move screen reader to main content" link and three icons: a yellow book, a green question mark, and a red equals sign. Below the banner, the title "Secure Software Concepts" is displayed above a sub-section titled "Course Overview and Objectives". On the right, a progress indicator shows "01/25". The main content area features a large graphic of a padlock on the left. In the center, a box lists the learning objectives:

When you have completed this course, you will be able to:

- Describe the current threat landscape
- Identify several common security vulnerabilities
- List several resources for evaluating and mitigating common security risks
- Identify security-related tasks for each stage in a secure software development lifecycle
- List resources for implementing a security strategy
- Describe how to apply several security best practices

A "Continue" button with a right-pointing arrow is located at the bottom right of the content area.

### Narration

This course provides a high-level overview of secure software concepts for applications, including application security, security standards, secure development methodologies, and security best practices.

When you have completed this course, you will be able to describe the current threat landscape and identify several common security vulnerabilities. You will also be able to list several resources for evaluating and mitigating the most common application security risks.

You will be able to identify security-related tasks for each stage in a secure software development lifecycle, and list resources for implementing a security strategy based on your organization's actual risk profile, and leveraging other organization's experiences with secure development practices. Finally, you will be able to describe how to apply several security best practices to harden your security stance.

### On Screen Text

#### Course Overview and Objectives

## **Secure Software Concepts**

---

**When you have completed this course, you will be able to:**

- Describe the current threat landscape
- Identify several common security vulnerabilities
- List several resources for evaluating and mitigating common security risks
- Identify security-related tasks for each stage in a secure software development lifecycle
- List resources for implementing a security strategy
- Describe how to apply several security best practices

# Secure Software Concepts

## Application Security Threats

The screenshot shows a digital learning interface. At the top left is the 'SECURITY INNOVATION' logo. To its right is a link 'Move screen reader to main content'. The top center has a decorative circuit board graphic. The top right contains three icons: a yellow folder, a green question mark, and a red document. The main title 'Secure Software Concepts' is at the top center of the slide area. Below it, the specific slide title 'Application Security Threats' is displayed. In the bottom right corner of the slide area, there is a '02/25' indicator. At the bottom right of the entire slide area is a white rectangular button labeled 'Continue' with a right-pointing arrow. The slide content itself features four stylized cartoon characters on the left side: a woman with brown skin and large hoop earrings, an older man with white hair and sunglasses, a blonde woman, and a man with blue hair. On the right side, a man with dark skin and short black hair is shown from the waist up, wearing a pink t-shirt, looking through a blue doorway while holding a small white device.

### Narration

Despite our greatest efforts to address the most common security threats, we now face increasingly sophisticated and targeted attacks. Even with our own applications secure, there is an additional attack surface from third-party libraries and external cloud services.

The current threat landscape not only includes individual hackers, but also coordinated attacks from state-sponsored groups, organized crime, botnets, ransomware, and distributed attacks from Internet of Things (IoT) devices and mobile malware.

Furthermore, there is increasing pressure from regulatory bodies, potential litigation, and greater customer expectations, all demanding the highest security.

### On Screen Text

#### Application Security Threats

# Secure Software Concepts

## Common Vulnerabilities

The screenshot shows a mobile application interface titled "Secure Software Concepts". At the top, there is a navigation bar with icons for back, forward, and search, along with the "SECURITY INNOVATION" logo and a link to move the screen reader to main content. Below the title, the page header reads "Common Vulnerabilities" and "03/25". The main content area displays five large, rounded square icons representing different types of vulnerabilities: a red one with a screwdriver, a red one with an envelope, an orange one with three gears, an orange one with a lightning bolt, and an orange one with a person climbing a wall. In the bottom right corner of the content area, there is a "Continue" button with a right-pointing arrow.

## Narration

Software security vulnerabilities are code defects that are exploitable and reproducible. Software security is the process of identifying and eliminating these defects.

Some of the most common vulnerabilities found in applications are:

Injection vulnerabilities – Failing to properly filter user input, thus leading to injection of arbitrary database, shell, or other commands.

Exposing sensitive data – Failing to properly encrypt or otherwise secure sensitive system and customer information.

Misconfigured security settings – Failing to properly configure a system or platform with the most secure options available.

Cross-site scripting - Failure to encode untrusted data for the correct output context.

And, cross-site request forgery – Failing to properly handle web requests to ensure users did intend to make sensitive changes or perform sensitive actions.

## On Screen Text

Common Vulnerabilities

# Secure Software Concepts

## Standards, Regulations, and Policies

The screenshot shows a web-based training interface. At the top left is the 'SECURITY INNOVATION' logo. A link 'Move screen reader to main content' is visible. The title 'Secure Software Concepts' is at the top center. Below it, the page title 'Standards, Regulations, and Policies' is on the left, and the date '04/25' is on the right. The main content area features three large logos: 'ASVS' (with a yellow oval), 'CWE' (in purple and blue), and 'SANS' (in green). A small circular icon with a leaf and a striped circle is positioned above the 'CWE' logo. In the bottom right corner of the content area is a 'Continue >' button.

### Narration

Whether it be healthcare, finance, e-commerce, or government entities, many organizations fall under the domain of one or more regulatory bodies. As custodians of sensitive user data, standards and regulatory bodies ensure organizations comply with minimum standards and practices.

Even without external regulations, it is important to implement widely-accepted application security guidelines in your organization. Secure coding best practices recommended by the Open Web Application Security Project (OWASP) Top 10, OWASP Application Security Verification Standard (ASVS), and the Common Weakness Enumeration (CWE)/SysAdmin, Audit, Network, Security (SANS) Institute Top 25 all provide guidance for the most common security threats.

As you design and build web applications, it is vital that you integrate adherence to security policies, standards, and regulations into the entire development lifecycle.

### On Screen Text

## Standards, Regulations, and Policies

# Secure Software Concepts

## About Web Application Security

The screenshot shows a web-based training module. At the top left is the "SECURITY INNOVATION" logo with a blue 'i' icon. A "Move screen reader to main content" link is visible above the main title. The main title "Secure Software Concepts" is centered at the top. Below it, the specific topic "About Web Application Security" is shown, along with the date "05/25". The central image is a large red wax seal impression containing a stylized letter 'B'. In the bottom right corner of the slide area, there is a "Continue >" button.

### Narration

As more organizations move their operations online and increasingly rely on web-based applications and services, the urgency of web application security becomes increasingly apparent. For those charged with managing and mitigating the risks of web applications, it can be a considerable task to balance the business, legal, and financial requirements of the organization.

Web applications are a primary target for hackers because they are the most visible targets and most likely to contain serious vulnerabilities; they are typically connected to database and other back-end servers that could be a path into the internal network; and, attacks can be easily automated to identify common weaknesses.

Application security includes designing software on a solid foundation of standards and best practices, using structured development frameworks, and implementing secure development methodologies. The failure to secure web applications can quickly escalate to a breach of an organization's entire network.

### On Screen Text

#### About Web Application Security

# Secure Software Concepts

## OWASP Top 10

The screenshot shows the OWASP Top 10 mobile application interface. At the top, there is a header bar with the OWASP logo, a screen reader icon, and three navigation icons. Below the header, the title "Secure Software Concepts" is displayed above the "OWASP Top 10" section. The main content area lists the 10 vulnerabilities in a grid format:

Rank	Vulnerability Name	Icon
1.	Injection	Icon of a syringe
2.	Broken Authentication + Session Management	Icon of a key and a lock
3.	Sensitive Data Exposure	Icon of an envelope with a lock
4.	XML External Entities	Icon of a bottle and a glass
5.	Broken Access Control	Icon of a door with a lock
6.	Security Misconfiguration	Icon of gears
7.	Cross Site Scripting	Icon of a script with a lightning bolt
8.	Insecure Deserialization	Icon of a broken puzzle piece
9.	Using Components with Known Vulnerabilities	Icon of an umbrella in the rain
10.	Insufficient Logging + Monitoring	Icon of a clipboard with a document

A "Check Your Knowledge" button is located in the bottom right corner of the main content area.

## Narration

The Open Web Application Security Project (OWASP) Top 10 list describes the most common web application attacks and vulnerabilities, along with example code vulnerabilities, example attacks, and guidance on mitigating risks.

The list is compiled based on a consensus of information security professionals and by analyzing actual attack data. Each item on the Top 10 list includes analysis of the risks, and the exploitability, prevalence, and detectability of the attack. The guidance also includes example scenarios and steps to prevent exposure to the vulnerability.

OWASP updates the Top 10 list every few years to help you focus on vulnerabilities that pose the most current risks. However, as with any security model, standard, or guideline, following the OWASP Top 10 does not necessarily eliminate all possible risks.

## On Screen Text

### OWASP Top 10

1. Injection
2. Broken Authentication + Session Management

## **Secure Software Concepts**

---

3. Sensitive Data Exposure
4. XML External Entities
5. Broken Access Control
6. Security Misconfiguration
7. Cross Site Scripting
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging + Monitoring

# Secure Software Concepts

## Knowledge Check

The screenshot shows a knowledge check interface. At the top left is the Security Innovation logo. A link "Move screen reader to main content" is visible. On the right are icons for a book, a question mark, and a refresh symbol. The title "Secure Software Concepts" is at the top center, followed by a sub-section title "Knowledge Check" and a progress indicator "07/25". The main content area contains a question: "Web applications are a primary target for hackers for all the following reasons except:". Below the question is a list of four options, each preceded by a radio button:

- They are the most likely to contain serious vulnerabilities
- They typically offer a potential path into the internal network
- Attacks can be easily automated to identify common weaknesses
- Regulatory bodies make web applications the most difficult to hack

## Narration

### On Screen Text

#### Knowledge Check

**Web applications are a primary target for hackers for all the following reasons except:**

They are the most likely to contain serious vulnerabilities

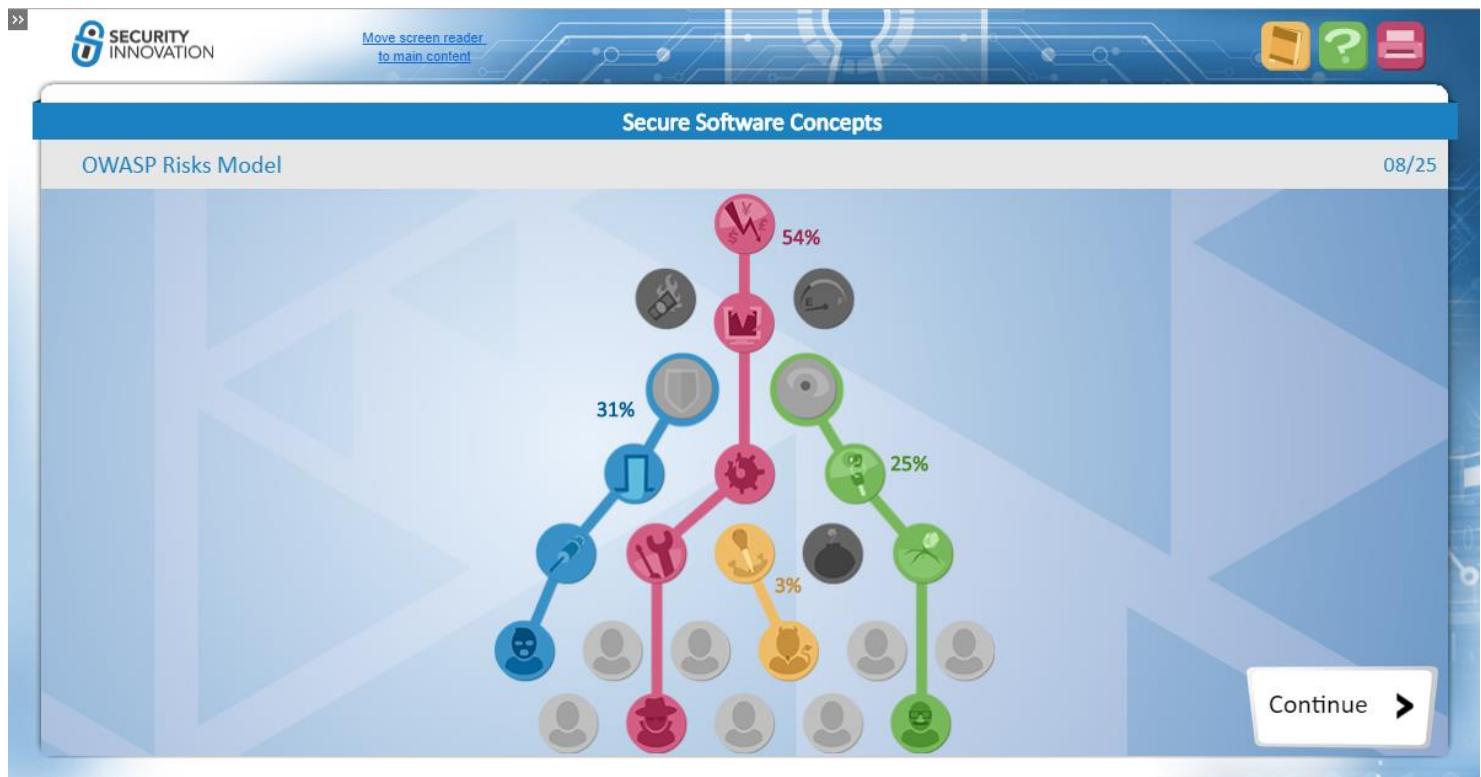
They typically offer a potential path into the internal network

Attacks can be easily automated to identify common weaknesses

Regulatory bodies make web applications the most difficult to hack

# Secure Software Concepts

## OWASP Risks Model



### Narration

For the OWASP Top 10, risk is a measurement of the likelihood that any particular attack path—from threat agent to security controls—will have a significant impact on your organization.

There may be many different attackers, attack vectors, security weaknesses, and security controls that exist in any web application. Any number of attackers—or threat agents—might employ different attack vectors looking for security weaknesses. A single security weakness coupled with insufficient security controls might lead to a security compromise. A security compromise could have a serious impact on both technical and business operations.

To determine risk, an organization must understand who the threat agents are and take into consideration the attack vectors, security weaknesses, and associated security controls for an application.

Finally, you must assess the likelihood of a particular compromise having a significant impact on your technical or business operations. Throughout the OWASP Top 10 guide, you will see risk ratings for each vulnerability.

### On Screen Text

#### OWASP Risks Model

31%

## Secure Software Concepts

---

54%

25%

3%

# Secure Software Concepts

## OWASP ASVS

The screenshot shows a web page titled "Secure Software Concepts" under the "OWASP ASVS" section. At the top left is the "SECURITY INNOVATION" logo. A navigation bar includes links for "Move screen reader to main content", "Secure Software Concepts", "OWASP ASVS", and a date "09/25". On the right are icons for a book, a question mark, and a refresh symbol. The main content area features a large orange circle containing the letters "ASVS". Below it, a text box states: "ASVS defines detailed verification requirements for levels 1 and above, whereas level 0 is meant to be flexible and is customized by each organization." To the right, four levels of verification are shown as colored triangles: "Advanced" (blue), "Standard" (green), "Opportunistic" (orange), and "Cursory" (pink). Each level has a plus sign icon next to its name. To the right of the levels is a numbered list of 19 verification requirements, starting with "1. Architecture, design, and threat modeling" and ending with "19. Configuration". A "Continue >" button is at the bottom right.

## Narration

The OWASP Application Security Verification Standard (ASVS) project provides a standard for testing web application security, with a list of verification requirements in nineteen key areas. Whereas the OWASP Top 10 provides a focused list of the most common security issues, the ASVS provides a comprehensive guide for addressing most common application security risks.

ASVS defines four levels of verification. Each level provides for a more in-depth set of requirements than the previous level. Applications that meet these requirements are considered an ASVS Level N application, where N is the level of compliance met. You can use the ASVS standard as both guidance and as a metric for measuring and comparing the security of an application.

## On Screen Text

### OWASP ASVS

ASVS defines detailed verification requirements for levels 1 and above, whereas level 0 is meant to be flexible and is customized by each organization.

Advanced

Standard

Opportunistic

Cursory

1. Architecture, design, and threat modeling
2. Authentication verification requirements
3. Session management verification requirements
4. Access control verification requirements
5. Malicious input handling verification requirements
6. Output encoding/escaping
7. Cryptography at rest verification requirements
8. Error handing and logging verification requirements
9. Data protection verification requirements
10. Communications security verification requirements
11. HTTP security configuration verification requirements
12. Security configuration verification requirements
13. Malicious controls verification requirements
14. Internal security verification requirements
15. Business logic verification requirements
16. Files and resources verification requirements
17. Mobile verification requirements
18. Web services verification requirements
19. Configuration

# Secure Software Concepts

## CWE/SANS Top 25

The screenshot shows a web-based report titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A link "Move screen reader to main content" is visible above the main table. The table has a blue header row with columns for Rank, Score, ID, and Name. The body of the table lists 25 entries, each with a rank, score, ID, and a brief description of the weakness. The weaknesses listed are: [1] Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection'), [2] Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection'), [3] Buffer Copy without Checking Size of Input ('Classic Buffer Overflow'), [4] Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'), [5] Missing Authentication for Critical Function, [6] Missing Authorization, [7] Use of Hard-coded Credentials, [8] Missing Encryption of Sensitive Data, [9] Unrestricted Upload of File with Dangerous Type, [10] Reliance on Untrusted Inputs in a Security Decision, [11] Execution with Unnecessary Privileges, [12] Cross-Site Request Forgery (CSRF), [13] Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal'), [14] Download of Code Without Integrity Check, [15] Incorrect Authorization, [16] Inclusion of Functionality from Untrusted Control Sphere, [17] Incorrect Permission Assignment for Critical Resource, [18] Use of Potentially Dangerous Function, [19] Use of a Broken or Risky Cryptographic Algorithm, [20] Incorrect Calculation of Buffer Size, [21] Improper Restriction of Excessive Authentication Attempts, [22] URL Redirection to Untrusted Site ('Open Redirect'), [23] Uncontrolled Format String, [24] Integer Overflow or Wraparound, and [25] Use of a One-Way Hash without a Salt.

Rank	Score	ID	Name
[1]	93.8	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	83.3	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[3]	79.0	CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[4]	77.7	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[5]	76.9	CWE-306	Missing Authentication for Critical Function
[6]	76.8	CWE-862	Missing Authorization
[7]	75.0	CWE-798	Use of Hard-coded Credentials
[8]	75.0	CWE-311	Missing Encryption of Sensitive Data
[9]	74.0	CWE-434	Unrestricted Upload of File with Dangerous Type
[10]	73.8	CWE-807	Reliance on Untrusted Inputs in a Security Decision
[11]	73.1	CWE-250	Execution with Unnecessary Privileges
[12]	70.1	CWE-352	Cross-Site Request Forgery (CSRF)
[13]	69.3	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[14]	68.5	CWE-494	Download of Code Without Integrity Check
[15]	67.8	CWE-863	Incorrect Authorization
[16]	66.0	CWE-829	Inclusion of Functionality from Untrusted Control Sphere
[17]	65.5	CWE-732	Incorrect Permission Assignment for Critical Resource
[18]	64.6	CWE-676	Use of Potentially Dangerous Function
[19]	64.1	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
[20]	62.4	CWE-131	Incorrect Calculation of Buffer Size
[21]	61.5	CWE-307	Improper Restriction of Excessive Authentication Attempts
[22]	61.1	CWE-601	URL Redirection to Untrusted Site ('Open Redirect')
[23]	61.0	CWE-134	Uncontrolled Format String
[24]	60.3	CWE-190	Integer Overflow or Wraparound
[25]	59.9	CWE-759	Use of a One-Way Hash without a Salt

## Narration

The MITRE Common Weakness Enumeration (CWE) group and SysAdmin, Audit, Network, Security (SANS) Institute published a report on the top 25 most dangerous software errors. The list is intended to help developers understand, recognize, and prevent common security coding errors that could lead to serious vulnerabilities in their applications.

Each of the CWE/SANS Top 25 errors is accompanied by a wealth of information such as remediation cost, attack frequency, consequences, and mitigation techniques.

Although the last joint report was published in 2011, nearly all the weaknesses cited are still relevant today. Additionally, the community-developed formal list of over 700 common weaknesses continues to be updated as an ongoing effort sponsored by the U.S. Department of Homeland Security.

## On Screen Text

### CWE/SANS Top 25

## Secure Software Concepts

---

Rank	Score	ID	Name
[1]	93.8	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	83.3	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[3]	79.0	CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[4]	77.7	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[5]	76.9	CWE-306	Missing Authentication for Critical Function
[6]	76.8	CWE-862	Missing Authorization
[7]	75.0	CWE-798	Use of Hard-coded Credentials
[8]	75.0	CWE-311	Missing Encryption of Sensitive Data
[9]	74.0	CWE-434	Unrestricted Upload of File with Dangerous Type
[10]	73.8	CWE-807	Reliance on Untrusted Inputs in a Security Decision
[11]	73.1	CWE-250	Execution with Unnecessary Privileges
[12]	70.1	CWE-352	Cross-Site Request Forgery (CSRF)
[13]	69.3	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[14]	68.5	CWE-494	Download of Code Without Integrity Check
[15]	67.8	CWE-863	Incorrect Authorization
[16]	66.0	CWE-829	Inclusion of Functionality from Untrusted Control Sphere
[17]	65.5	CWE-732	Incorrect Permission Assignment for Critical Resource
[18]	64.6	CWE-676	Use of Potentially Dangerous Function
[19]	64.1	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
[20]	62.4	CWE-131	Incorrect Calculation of Buffer Size
[21]	61.5	CWE-307	Improper Restriction of Excessive Authentication Attempts
[22]	61.1	CWE-601	URL Redirection to Untrusted Site ('Open Redirect')
[23]	61.0	CWE-134	Uncontrolled Format String

## Secure Software Concepts

---

Rank	Score	ID	Name
[24]	60.3	CWE-190	Integer Overflow or Wraparound
[25]	59.9	CWE-759	Use of a One-Way Hash without a Salt

# Secure Software Concepts

## Knowledge Check

The screenshot shows a knowledge check interface. At the top, there's a header bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons (a book, a question mark, and a document). Below the header is a blue navigation bar with the title 'Secure Software Concepts'. On the left, a sidebar says 'Knowledge Check'. On the right, it shows '11/25'. The main content area contains a question: 'Which one of the following guidelines provides a standard for addressing application security risks in 19 key areas?'. Four options are listed with radio buttons:

- OWASP Top 10
- CWE/SANS Top 25
- OWASP ASVS
- OWASP Risks

## Narration

### On Screen Text

### Knowledge Check

Which one of the following guidelines provides a standard for addressing application security risks in 19 key areas?

OWASP Top 10

CWE/SANS Top 25

OWASP ASVS

OWASP Risks

# Secure Software Concepts

## Secure SDLC

The screenshot shows a web page titled "Secure Software Concepts". The main content area is titled "Secure SDLC". A large graphic in the center features a green circle with a white cross inside, surrounded by blue arrows forming a circular flow. Below the graphic is a dashed orange bar with five gray dots. To the right of the graphic is a text box containing the following text:

A secure SDLC facilitates building a system that is well-aligned with security and compliance requirements throughout each stage of development by incorporating security-related tasks in each lifecycle stage.

*Click each of the gray dots below to learn more.*

The top navigation bar includes the "SECURITY INNOVATION" logo, a "Move screen reader to main content" link, and three icons: a yellow square, a green circle with a question mark, and a red rectangle.

### Narration

A secure software development lifecycle—or SDLC—ensures that proper security controls and safeguards are an intrinsic part of an application’s development. By treating security as a design requirement, it becomes ingrained into every component of the system.

An SDLC facilitates building a system that is well-aligned with security and compliance requirements throughout each stage of development by incorporating security-related tasks in each lifecycle stage. Let’s look at an example.

### On Screen Text

#### Secure SDLC

A secure SDLC facilitates building a system that is well-aligned with security and compliance requirements throughout each stage of development by incorporating security-related tasks in each lifecycle stage.

*Click each of the gray dots below to learn more.*

# Secure Software Concepts

## Secure SDLC

The screenshot shows a web page titled "Secure Software Concepts" with a sub-section titled "Secure SDLC". The page features a cartoon illustration of a security professional wearing a headset and sunglasses, holding a clipboard. A speech bubble contains the text: "In the **requirements** stage you would identify security and compliance objectives, establish security standards, and perform risk assessments." The page has a blue header and footer, and a navigation bar with icons for search, help, and print.

## Narration

In the requirements stage you would identify security and compliance objectives, establish security standards, and perform risk assessments.

## On Screen Text

### Secure SDLC

In the **requirements** stage you would identify security and compliance objectives, establish security standards, and perform risk assessments.

# Secure Software Concepts

## Secure SDLC

The screenshot shows a course interface titled "Secure Software Concepts". The top navigation bar includes the "SECURITY INNOVATION" logo, a "Move screen reader to main content" link, and three icons. The main content area has a blue header "Secure SDLC" and a progress bar at the bottom. A callout box contains the text: "During **design** you would analyze the application's attack surface, perform **threat modeling**, and document the application's security architecture." Below the text is an illustration of a person working at a computer with various charts and graphs.

## Narration

During design you would analyze the application's attack surface, perform threat modeling, and document the application's security architecture.

## On Screen Text

### Secure SDLC

During **design** you would analyze the application's attack surface, perform threat modeling, and document the application's security architecture.

# Secure Software Concepts

---

## Secure SDLC

The screenshot shows a web-based training application. At the top, there's a header with the "SECURITY INNOVATION" logo and a "Move screen reader to main content" link. Below the header is a blue navigation bar with the title "Secure Software Concepts". On the left, a sidebar has a "Secure SDLC" button and a "12/25" progress indicator. The main content area features a green background with a stylized "X" pattern. Two cartoon characters, a boy and a girl, are shown at a desk with a computer monitor. A callout box contains the text: "Once in **development** you would address security guidelines as well as perform [code reviews](#) and [static analysis](#)". A dashed orange arrow at the bottom points from left to right, with a red circle highlighting the third dot.

## Narration

Once in development you would address security guidelines as well as perform code reviews and static analysis.

## On Screen Text

### Secure SDLC

Once in **development** you would address security guidelines as well as perform code reviews and static analysis.

# Secure Software Concepts

## Secure SDLC

The screenshot shows a slide from a presentation titled "Secure Software Concepts". The slide has a blue header bar with the title and a navigation menu. Below the header is a green section containing a cartoon illustration of a person with black hair and a white shirt using a power drill on a computer monitor. A dashed orange line with circular markers runs across the bottom of this section. To the right of the illustration is a speech bubble containing text about the testing phase. The slide is numbered 12/25 in the top right corner.

In the **testing** phase you would perform [dynamic analysis](#), review the application's attack surface, and execute [penetration tests](#).

## Narration

In the testing phase you would perform dynamic analysis, review the application's attack surface, and execute penetration tests.

## On Screen Text

### Secure SDLC

In the **testing** phase you would perform dynamic analysis, review the application's attack surface, and execute penetration tests.

# Secure Software Concepts

## Secure SDLC

The screenshot shows a course interface titled "Secure Software Concepts". The top navigation bar includes the "SECURITY INNOVATION" logo, a "Move screen reader to main content" link, and icons for download, help, and search. The main content area has a blue header "Secure Software Concepts" and a sub-header "Secure SDLC". A progress bar at the top right indicates "12/25". The main content area contains a text box with the following text:

Finally, in preparation for **deployment** you would perform final security reviews, create an incident response plan, and document compliance with the appropriate policies and standards.

To the right of the text box is an illustration of a person wearing a green beret with yellow stars and sunglasses, holding a red megaphone. Next to the person is a whiteboard with a black and red diagram consisting of arrows and crosses. Below the illustration is a dashed orange path with five circular markers, ending in a red circle with a white arrow pointing right. A "Continue >" button is located at the bottom right of the path.

## Narration

Finally, in preparation for deployment you would perform final security reviews, create an incident response plan, and document compliance with the appropriate policies and standards.

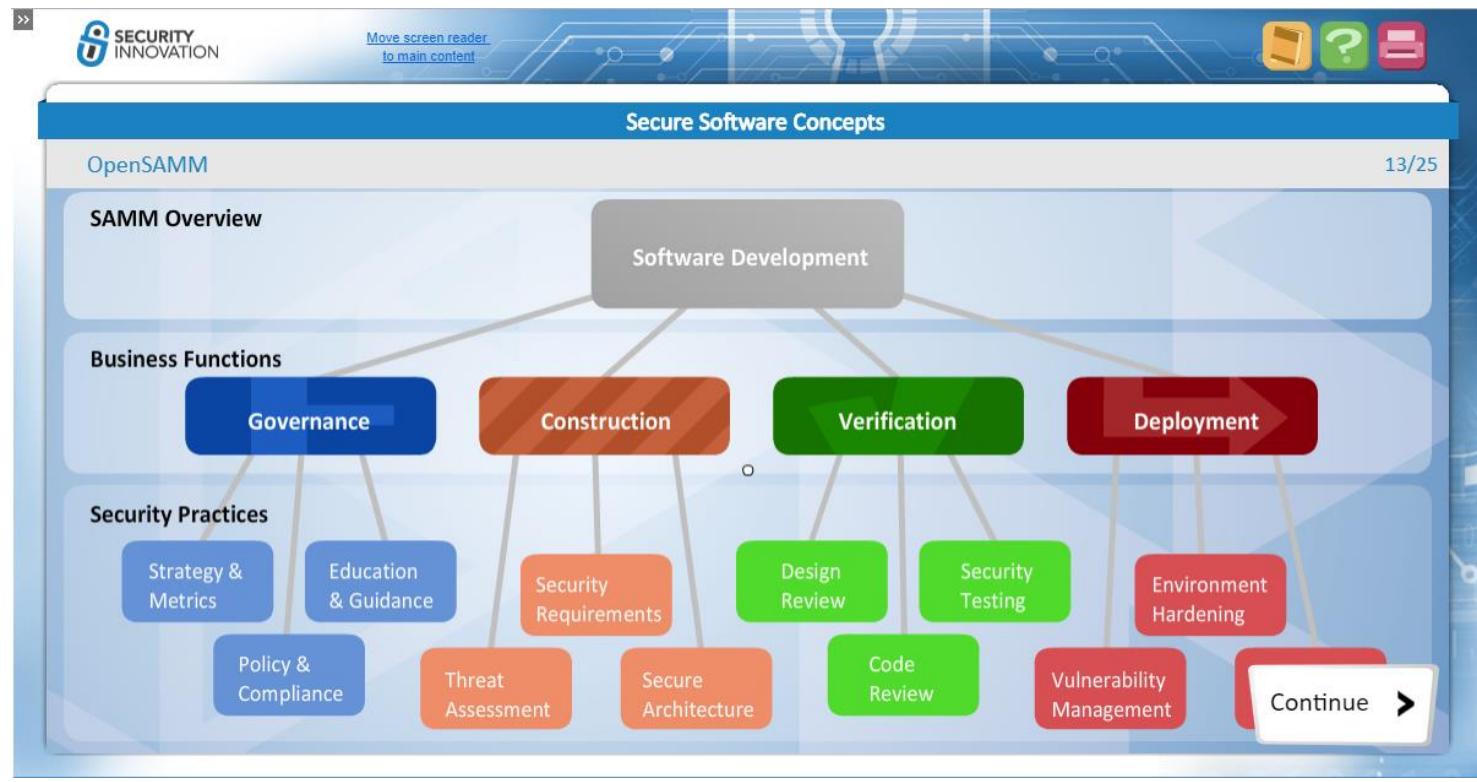
## On Screen Text

### Secure SDLC

Finally, in preparation for **deployment** you would perform final security reviews, create an incident response plan, and document compliance with the appropriate policies and standards.

# Secure Software Concepts

## OpenSAMM



## Narration

The Open Software Assurance Maturity Model (OpenSAMM) is an open software security framework that helps organizations devise and implement a software security strategy that is aligned to its actual risks.

The OpenSAMM project provides guidance to help organizations evaluate existing software security practices, build a software security program, demonstrate improvements to a security assurance program, and define and measure security-related activities.

OpenSAMM is designed to help evaluate an organization's current software development security practices, build a well-balanced software security program, produce demonstrable improvements in software security management, and define and measure all software security tasks within an organization.

## On Screen Text

OpenSAMM

# Secure Software Concepts

## BSIMM

The screenshot shows a web-based interface for the BSIMM Secure Software Concepts. At the top, there's a header bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons. Below the header is a blue navigation bar with the title 'Secure Software Concepts' and the text 'BSIMM' on the left and '14/25' on the right. A sub-instruction 'Click each domain to learn more.' is displayed. The main area features four colored boxes representing domains: 'Governance' (blue), 'Intelligence' (green), 'SSDL Touchpoints' (orange), and 'Deployment' (red). Each box contains a small icon related to its domain.

## Narration

The Building Security In Maturity Model (BSIMM) is a study of real-world software security initiatives from companies such as Adobe, Microsoft, PayPal, and many others.

It does not represent a specific security development model, but instead provides an aggregate of the experiences and lessons learned by the participating organizations.

BSIMM describes over 100 activities that your organization can implement. Those activities are organized in a free software security framework (SSF) that defines 12 practices organized into four domains: Governance, Intelligence, Secure Software Development Lifecycle (SSDL), and Deployment.

## On Screen Text

**BSIMM**

# Secure Software Concepts

## BSIMM

The screenshot shows a web-based interface for 'Secure Software Concepts'. At the top, there's a header bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons. Below the header is a blue navigation bar with 'Secure Software Concepts' and 'BSIMM' tabs, and a '14/25' indicator. The main content area has a light blue background with a grid pattern. On the left, a white box contains text about organizing, managing, and measuring software security initiatives, followed by a bulleted list: 'Strategy and Metrics', 'Compliance and Policy', and 'Training'. To the right of this text are four colored boxes: 'Governance' (grey), 'Intelligence' (green with a brain icon), 'SSDL Touchpoints' (orange), and 'Deployment' (red with a large arrow).

## Narration

Practices that help you organize, manage, and measure your software security initiative including staff development include strategy and metrics, compliance and policy, and training.

## On Screen Text

### BSIMM

#### Governance

Practices that help you organize, manage and measure your software security initiative including staff development.

- Strategy and Metrics
- Compliance and Policy
- Training

# Secure Software Concepts

## BSIMM

The screenshot shows a web-based interface for 'Secure Software Concepts'. At the top, there's a header bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons. Below the header is a blue navigation bar with 'Secure Software Concepts' and 'BSIMM' on the left, and '14/25' on the right. The main content area has a light blue background with a white rounded rectangle containing text and a bulleted list. To the right are four colored boxes: 'Governance' (blue), 'Intelligence' (grey with a brain icon), 'SSDL Touchpoints' (orange), and 'Deployment' (red with a right-pointing arrow). The 'Intelligence' box is highlighted with a grey border.

Practices that result in collections of knowledge to use to carry out software security activities throughout your organization.

- Attack Models
- Security Features and Design
- Standards and Requirements

## Narration

Practices that result in collections of knowledge to use to carry out software security activities throughout your organization include attack models, security features and design, and standards and requirements.

## On Screen Text

### BSIMM

#### Intelligence

Practices that result in collections of knowledge to use to carry out software security activities throughout your organization.

- Attack Models
- Security Features and Design
- Standards and Requirements

# Secure Software Concepts

## BSIMM

The screenshot shows a slide titled "Secure Software Concepts" under the "BSIMM" category. On the left, a box lists common practices: Architecture Analysis, Code Review, and Security Testing. To the right, four colored boxes represent touchpoints: Governance (blue), Intelligence (green), SSDL Touchpoints (grey), and Deployment (red).

Common practices associated with analysis and assurance of particular software development artifacts and processes.

- Architecture Analysis
- Code Review
- Security Testing

Governance

Intelligence

SSDL Touchpoints

Deployment

## Narration

Common practices associated with analysis and assurance of particular software development artifacts and processes include architecture analysis, code review and security testing.

## On Screen Text

### BSIMM

#### SSDL Touchpoints

Common practices associated with analysis and assurance of particular software development artifacts and processes.

- Architecture Analysis
- Code Review
- Security Testing

# Secure Software Concepts

## BSIMM

The screenshot shows a web-based interface for 'Secure Software Concepts'. At the top, there's a navigation bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons. Below the header, the title 'Secure Software Concepts' is displayed above a horizontal menu bar with 'BSIMM' on the left and '14/25' on the right. The main content area features four rounded rectangular boxes arranged in a 2x2 grid. The top-left box contains text about practices interfacing with security and maintenance organizations, followed by a bulleted list: Penetration Testing, Software Environment, Configuration Management, and Vulnerability Management. The other three boxes are labeled 'Governance' (blue), 'Intelligence' (green), 'SSDL Touchpoints' (orange), and 'Deployment' (grey).

## Narration

Practices that interface with traditional network security and software maintenance organizations include penetration testing, software environment config management, and vulnerability management.

## On Screen Text

### BSIMM

### Deployment

Practices that interface with traditional network security and software maintenance organizations.

- Penetration Testing
- Software Environment
- Configuration Management
- Vulnerability Management

# Secure Software Concepts

## Knowledge Check

The screenshot shows a knowledge check interface. At the top left is the Security Innovation logo. A blue banner across the top center contains the text "Secure Software Concepts". Below the banner, the title "Knowledge Check" is displayed, along with a progress indicator "15/25". The main content area contains a question and four options. The question is: "In which stage of a secure development lifecycle should you identify security and compliance objectives, establish security standards, and perform risk assessments?". The options are: Requirements, Design, Development, and Testing. Each option is preceded by a radio button.

In which stage of a secure development lifecycle should you identify security and compliance objectives, establish security standards, and perform risk assessments?

- Requirements
- Design
- Development
- Testing

## Narration On Screen Text

### Knowledge Check

In which stage of a secure development lifecycle should you identify security and compliance objectives, establish security standards, and perform risk assessments?

Requirements

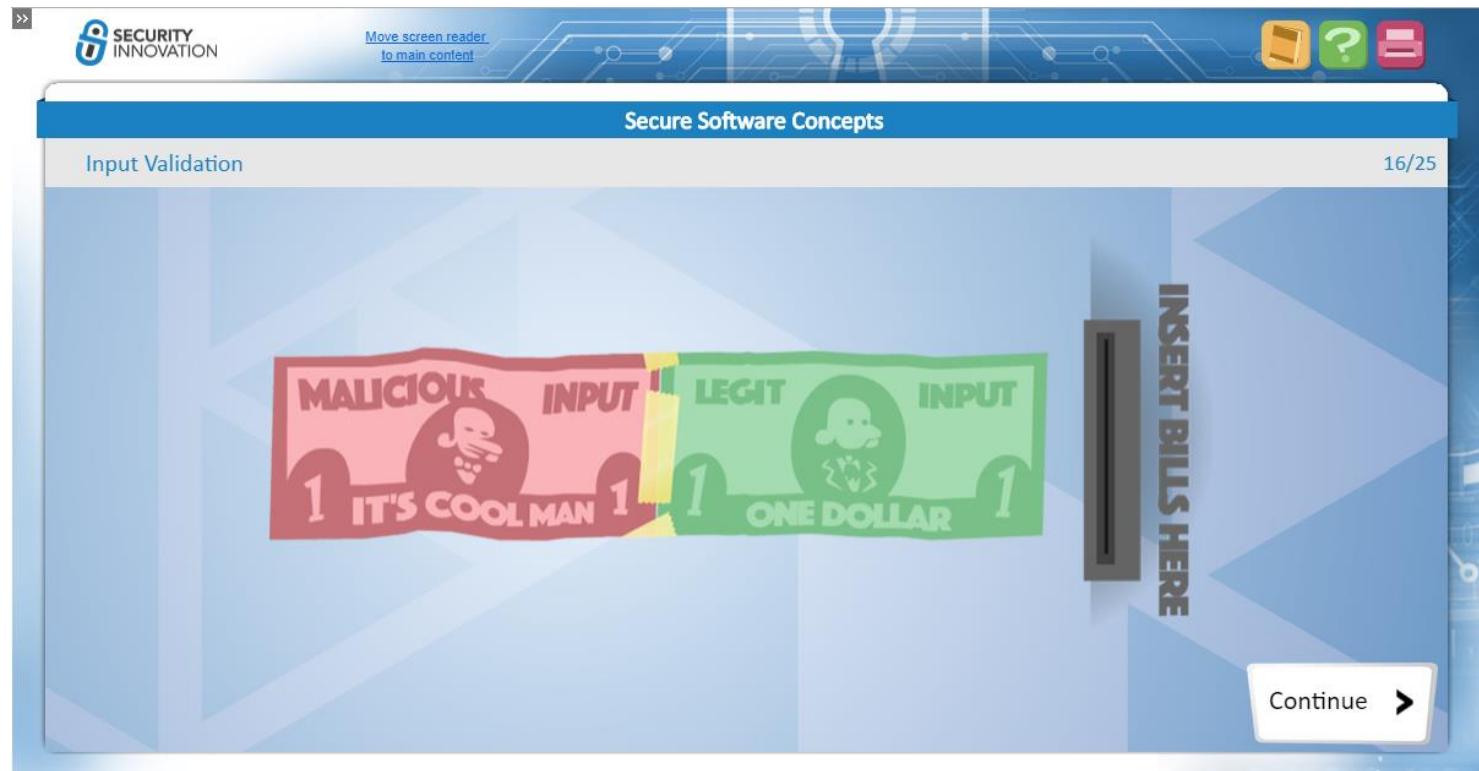
Design

Development

Testing

# Secure Software Concepts

## Input Validation



## Narration

As you learn about application security, one practice you will hear about often is input validation, primarily because it plays such a vital role in preventing security flaws.

## On Screen Text

### Input Validation

# Secure Software Concepts

## Input Validation Process

The slide features a blue header with the title 'Secure Software Concepts'. Below the header is a sub-header 'Input Validation Process'. In the center of the slide is a graphic of a red envelope being washed by a purple sponge in water, with bubbles around it. The background has a light blue geometric pattern. At the bottom right is a 'Continue' button with a right-pointing arrow. The top right corner shows '17/25'.

### Narration

Input validation occurs in multiple steps:

First, there is client-side validation, which is there mostly to assist users in data entry and to improve the user experience, for example, by ensuring that a phone number does not contain letters or other invalid characters.

After the server receives the data from the web browser, there should then be a normalization (or canonicalization) process. This step helps ensure that the data is formatted consistently for validation. This step would include counteracting any attempts to obscure an attack, such as removing encoded characters.

With data normalized, the next step would be to validate it using a whitelist approach. Rather than blocking known bad data—a blacklist approach—you would only allow data based on acceptable ranges, formats, data types, and lengths. For example, a financial transaction should only contain numbers, decimals, commas, and currency symbols.

Optionally, the application might perform an additional step to strip off any data not needed, such as hyphens, parenthesis, commas, etc.

And finally, if necessary, the application might perform any character encoding required for data storage.

### On Screen Text

### Input Validation Process

# Secure Software Concepts

## Input Validation Strategies

The screenshot shows a web-based training module. At the top, there's a header bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons. Below the header is a blue navigation bar with the title 'Secure Software Concepts' and the page number '18/25'. The main content area has a light blue background with a large graphic of a dollar bill. On the left, a white callout box contains the text: 'Some basic strategies for input validation are:' followed by a bulleted list: • Document the Data Flow • Keep it Centralized • Keep it Balanced • Use a Library. To the right of the callout is a graphic of a dollar bill labeled 'LEGIT INPUT' and a red bill labeled 'MALICIOUS INPUT IT'S COOL MAN 1'. A 'Continue >' button is located in the bottom right corner of the slide area.

### Narration

Input validation isn't simply checking the values for each input; doing it right requires a comprehensive and consistent strategy across the entire application.

Some basic strategies for input validation are:

Document the Data Flow – Dissect the application to identify trust boundaries, data flows, entry points, and exit points. The key here is to know what data enters your application, where it enters, and where it originates.

Keep it Centralized – Centralizing data validation code provides a valuable chokepoint for data entering the application, and it also allows for consistent and easy-to-audit validation code.

Keep it Balanced – While validating data is important, it should not detract from the user experience. Some data can be automatically sanitized—such as removing dashes from a phone number—rather than rejected for not matching a specific format.

Use a Library – Many application frameworks and common libraries have robust data-validation functions built in. Make use of these features when possible as they have often been well-tested in a variety of scenarios.

## On Screen Text

### Input Validation Strategies

Some basic strategies for input validation are:

- Document the Data Flow
- Keep it Centralized
- Keep it Balanced
- Use a Library

# Secure Software Concepts

## Least Privilege

The screenshot shows a web-based training module. At the top, there's a header bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons. Below the header is a blue navigation bar with the title 'Secure Software Concepts' and the current page 'Least Privilege'. On the right of the navigation bar is a progress indicator '19/25'. The main content area has a light blue background with a white callout box containing text and a bulleted list. To the right of the text is a cartoon illustration of a person in a hard hat working on a server or firewall. A 'Continue >' button is at the bottom right of the main content area.

Examples of implementing the principle of least privilege are:

- Using a limited-user account context
- Removing write privileges for the web application's user
- Configuring firewall to only allow HTTP or (HTTPS)
- Setting file permissions that prevent modification of web content files

## Narration

The least privileges principle begins with the idea that all software can and will eventually be compromised by a malicious user. To reduce the impact of a compromise, we must design applications using the minimal set of privileges required to function correctly. This principle applies across all layers of the application, including the underlying operating system, databases, and cloud accounts.

The principle of least privilege accomplishes two things. First, it reduces the attack surface and thus lessens the likelihood of successful attacks. Second, it limits capabilities after a successful attack and thus makes it more difficult to execute lateral attacks.

Examples of implementing the principle of least privilege are: Using a limited-user account context for running the web, database, and application server software; removing write privileges for the web application's user if an application doesn't need to write to a database table; configuring a firewall to only allow a web server to communicate via Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS); and, setting file permissions that prevent the web server platform from modifying web content files.

## On Screen Text

### Least Privilege

## **Secure Software Concepts**

---

The principle of least privilege accomplishes two things:

- Reduces the attack surface and thus lessens the likelihood of successful attacks.
- Limits capabilities after a successful attack and thus makes it more difficult to execute lateral attacks.

Examples of implementing the principle of least privilege are:

- Using a limited-user account context
- Removing write privileges for the web application's user
- Configuring firewall to only allow HTTP or (HTTPS)
- Setting file permissions that prevent modification of web content files

# Secure Software Concepts

## Least Privilege Best Practices

The screenshot shows a slide from a course titled "Secure Software Concepts". The title bar includes the "SECURITY INNOVATION" logo and a "Move screen reader to main content" link. The slide has a blue header with the title "Secure Software Concepts" and a subtitle "Least Privilege Best Practices". In the top right corner, it says "20/25". The main content area contains a list of best practices:

- Start with nothing
- Segment your application
- Granting temporary privilege and revoke upon completion
- Have stakeholder buy-in

To the right of the list are three badge-like icons representing different users or roles. A red wax seal with a large letter "B" is placed over the top badge. Below the badges is a "Continue" button with a right-pointing arrow.

### Narration

As you implement the principle of least privilege security into your application, keep in mind the following tips:

Start with nothing, treating every user or process as having no privileges and only granting those privileges necessary to perform each action.

Segment your application so that you can easily implement a role-based approach to security.

If necessary to give a user or process higher privileges for one action, consider granting that privilege only temporarily and revoking it as soon as possible after completion of the action.

Finally, because least privilege can limit what certain users can do, make sure you have stakeholder buy-in before introducing restrictions.

### On Screen Text

## Least Privilege Best Practices

Least Privilege Best Practices:

- Start with nothing
- Segment your application

## **Secure Software Concepts**

---

- Granting temporary privilege and revoke upon completion
- Have stakeholder buy-in

# Secure Software Concepts

## Defense in Depth

The screenshot shows a web-based training module titled "Secure Software Concepts". The main content area is titled "Defense in Depth". On the left, there is a list of measures for defense in depth. On the right, there is a diagram illustrating multiple layers of security defenses represented by nested shapes and keyholes.

For a web application, [defense in depth](#) might include:

- Implementing a web application firewall
- Implementing web server and other platform protections
- Hardening the server's operating system
- Properly validating all application input
- Setting database constraints to ensure proper data formats
- Creating audit logs to track application operations

Continue ➤

## Narration

Another important security principle is defense in depth—in short creating multiple layers of security defenses to protect from failures in any one layer.

Even the most sophisticated security techniques are prone to failure. With multiple layers of defense, however, each layer increases the difficulty and effort required to compromise the application. Defense in depth ensures that all but the most sophisticated and determined attacks will fail. Defense in depth not only provides redundancy, but it can also address unexpected attack vectors, such as internal threats or physical attacks.

For a web application, defense in depth might include: implementing a web application firewall; implementing web server and other platform protections; hardening the server's operating system; properly validating all application input; setting database constraints to ensure proper data formats; and, creating audit logs to track application operations.

## On Screen Text

### Defense in Depth

## **Secure Software Concepts**

---

For a web application, defense in depth might include:

- Implementing a web application firewall
- Implementing web server and other platform protections
- Hardening the server's operating system
- Properly validating all application input
- Setting database constraints to ensure proper data formats
- Creating audit logs to track application operations

# Secure Software Concepts

## Cryptography

The screenshot shows a web-based learning platform. At the top, there's a navigation bar with the 'SECURITY INNOVATION' logo, a 'Move screen reader to main content' link, and three icons (yellow square, green circle, red rectangle). Below the bar, the title 'Secure Software Concepts' is displayed, followed by the section name 'Cryptography' and a progress indicator '22/25'. The main content area features a large blue circular icon containing a stylized gear or key symbol, set against a background of blue geometric shapes. In the bottom right corner of this area, there's a white button labeled 'Check Your Knowledge' with a right-pointing arrow. The overall theme is cybersecurity and technology.

## Narration

Protecting private user information means encrypting data both in transit and on disk. Cryptography is a key security principle, but it isn't just about encryption. Cryptography also provides for authentication, confidentiality, and integrity. Authentication is the verification of a claim. That claim could be the assertion that a person is who they say they are or that software is what it claims to be. Authentication can also apply to web sites, allowing you to ensure that you are connecting to the proper host.

Confidentiality is the assurance that communications between two parties are private and cannot be intercepted by any third party.

Integrity is proof that data has not changed in transit from one location to another or from one medium to another. Public key cryptography provides a means for authentication through the use of public key certificates, confidentiality with asymmetric encryption, and integrity by means of digital signatures.

Cryptography in general is mature and appropriately secure when implemented with well-established libraries and following the proper best practices.

## On Screen Text

### Cryptography

# Secure Software Concepts

## Knowledge Check

The screenshot shows a knowledge check question. At the top left is the Security Innovation logo. To its right is a link "Move screen reader to main content". On the far right are three icons: a yellow book, a green question mark, and a red document. The main title "Secure Software Concepts" is at the top center. Below it, the section title "Knowledge Check" is on the left, and "23/25" is on the right. The question asks: "Which one of the following is the correct order of steps for input validation?". Four options are listed, each preceded by a radio button:

- Whitelist validation, encoding, canonicalization
- Client-side validation, blacklist validation, encoding
- Encoding, canonicalization, client-side validation
- Canonicalization, whitelist validation, encoding

## Narration

### On Screen Text

#### Knowledge Check

**Which one of the following is the correct order of steps for input validation?**

Whitelist validation, encoding, canonicalization

Client-side validation, blacklist validation, encoding

Encoding, canonicalization, client-side validation

Canonicalization, whitelist validation, encoding

# Secure Software Concepts

## Course Summary

The screenshot shows a course summary page titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A blue banner across the top has the text "Move screen reader to main content". On the right side of the banner are three icons: a yellow folder, a green question mark, and a red document. Below the banner, the title "Secure Software Concepts" is centered above a large blue graduation cap icon. To the left of the cap, the text "Click each tab to learn more." is displayed. Below the cap are four tabs: "Application Security", "Security Standards", "Secure Development Methodologies", and "Security Best Practices". The background features a blue and white geometric pattern.

## Narration

### On Screen Text

#### Course Summary

*Click each tab to learn more.*

# Secure Software Concepts

## Course Summary

The screenshot shows a course summary page titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A blue banner at the top right contains icons for a book, a question mark, and a refresh symbol, along with the text "Move screen reader to main content". Below the banner, the page title "Secure Software Concepts" is displayed. On the left, there's a large graphic of a padlock. A callout box labeled "Application Security" contains text about the topic. At the bottom, four tabs are visible: "Application Security", "Security Standards", "Secure Development Methodologies", and "Security Best Practices". A note at the bottom left says "Click each tab to learn more."

## Narration

In this topic, you learned about application security, which involves designing software on a solid foundation of standards, best practices, and secure development methodologies. You also learned about the increasing diversity of the current threat landscape, and about several of the most common security vulnerabilities applications face.

*Click here to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

## On Screen Text

### Course Summary

#### Application Security

In this topic, you learned about application security, which involves designing software on a solid foundation of standards, best practices, and secure development methodologies. You also learned about the increasing diversity of the current threat landscape, and about several of the most common security vulnerabilities applications face.

## **Secure Software Concepts**

---

*Click [here](#) to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

*Click each tab to learn more.*

# Secure Software Concepts

## Course Summary

The screenshot shows a course summary page titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A blue banner across the top has a circuit board pattern and includes icons for a document, a question mark, and a menu. Below the banner, the title "Secure Software Concepts" is centered. On the left, a sidebar contains the text "Course Summary" and "Security Standards". The main content area contains a paragraph about security policies, standards, and regulations, mentioning the OWASP Top 10, ASVS projects, and CWE/SANS Top 25. It also includes a link to review the section again and a note to use the TOC to navigate to the Thank You page. To the right, there is a large graphic featuring the logos for ASVS (Application Security Verification Standard), CWE (Common Weakness Enumeration), and SANS. Below the main content are four tabs: "Application Security", "Security Standards" (which is selected), "Secure Development Methodologies", and "Security Best Practices". A footer at the bottom left says "Click each tab to learn more."

## Narration

In this topic, you learned that security policies, standards, and regulations should be integrated into the entire development lifecycle to inform secure coding best practices. This topic also discussed the OWASP Top 10 and ASVS projects and the CWE/SANS Top 25 list for evaluating and mitigating risks from the most common application security vulnerabilities.

*Click here to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

## On Screen Text

### Course Summary

#### Security Standards

In this topic, you learned that security policies, standards, and regulations should be integrated into the entire development lifecycle to inform secure coding best practices. This topic also discussed the OWASP Top 10 and ASVS projects and the CWE/SANS Top 25 list for evaluating and mitigating risks from the most common application security vulnerabilities.

*Click here to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

## **Secure Software Concepts**

---

*Click each tab to learn more.*

# Secure Software Concepts

## Course Summary

The screenshot shows a course summary page titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A blue banner across the top includes a "Move screen reader to main content" link and three icons: a yellow folder, a green question mark, and a red equals sign. Below the banner, the page title "Secure Software Concepts" is displayed above a "Course Summary" section. On the right, a progress bar shows "24/25". The main content area contains a section titled "Secure Development Methodologies" with a descriptive paragraph. Below this is a callout box with the text: "Click [here](#) to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course." At the bottom of the content area, there are four tabs: "Application Security", "Security Standards", "Secure Development Methodologies" (which is highlighted in blue), and "Security Best Practices". To the right of the tabs is a graphic featuring a green dollar bill labeled "LEGIT INPUT" and a red banner labeled "MALICIOUS INPUT IT'S COOL MAN 1".

## Narration

This topic provided examples of security-related tasks you might perform at each stage of a secure software development lifecycle. It also described OpenSAMM, a framework which can help you implement a security strategy aligned to the actual risks faced by your organization, and BSIMM, an aggregate report of the lessons learned by major corporations in their experiences implementing secure development practices.

*Click [here](#) to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

## On Screen Text

### Course Summary

#### Secure Development Methodologies

This topic provided examples of security-related tasks you might perform at each stage of a secure software development lifecycle. It also described OpenSAMM, a framework which can help you implement a security strategy aligned to the actual risks faced by your organization, and BSIMM, an aggregate report of the lessons learned by major corporations in their experiences implementing secure development practices.

## **Secure Software Concepts**

---

*Click [here](#) to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

*Click each tab to learn more.*

# Secure Software Concepts

## Course Summary

The screenshot shows a course summary page titled "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A "Move screen reader to main content" link is visible. On the right are icons for a book, a question mark, and a print function. The main content area has a blue header "Secure Software Concepts" and a sub-header "Course Summary". Below this is a large text box containing the "Security Best Practices" section. The text describes four best practices: input validation, least privilege, defense in depth, and cryptography. It also includes a link to review the section again and a note to use the TOC to navigate to the Thank You page. A red wax seal icon with a large letter "B" is prominently displayed on the right side. At the bottom left is a callout "Click each tab to learn more." with tabs for Application Security, Security Standards, Secure Development Methodologies, and Security Best Practices. A "Continue >" button is at the bottom right.

Move screen reader to main content

Secure Software Concepts

Course Summary 24/25

Security Best Practices

In this topic you learned about four security best practices: input validation, which is the first line of defense against many attacks; least privilege, which minimizes the potential impact of successful attacks; defense in depth, which strengthens your security position; and cryptography, which when applied properly can protect sensitive data and provide assurances of confidentiality, integrity, and authenticity.

Click [here](#) to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.

Click each tab to learn more.

Application Security Security Standards Secure Development Methodologies Security Best Practices

Continue >

## Narration

In this topic you learned about four security best practices: input validation, which is the first line of defense against many attacks; least privilege, which minimizes the potential impact of successful attacks; defense in depth, which strengthens your security position; and cryptography, which when applied properly can protect sensitive data and provide assurances of confidentiality, integrity, and authenticity.

*Click here to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

## On Screen Text

### Course Summary

#### Security Best Practices

In this topic you learned about four security best practices: input validation, which is the first line of defense against many attacks; least privilege, which minimizes the potential impact of successful attacks; defense in depth, which strengthens your security position; and cryptography, which when applied properly can protect sensitive data and provide assurances of confidentiality, integrity, and authenticity.

*Click [here](#) to review this section again. When you are done, use the TOC to navigate to the Thank You page to complete this course.*

## **Secure Software Concepts**

---

*Click each tab to learn more.*

# Secure Software Concepts

---

## Thank You

The screenshot shows a course completion page for "Secure Software Concepts". At the top left is the "SECURITY INNOVATION" logo. A blue banner across the top has the course title "Secure Software Concepts" in white. Below the banner, the text "Thank You" is on the left and "25/25" is on the right. A large central white box contains the text: "This concludes the **Secure Software Concepts** course. Thank you." Below this, in italicized text, is the instruction: "Click the “Take the Exam” button to proceed to the exam." In the bottom right corner of the main box is a white button with the text "Take the Exam" and a right-pointing arrow. The background features a blue and grey geometric pattern.

## Narration

## On Screen Text

### Thank You

This concludes the **Secure Software Concepts** course. Thank you.

*Click the “Take the Exam” button to proceed to the exam.*