Section Quiz

Candidate: Dunkan Gibson (dunkan.gibson)
Date: 4/5/2022 7:30:36 pm • Time spent: 03:44

Score: 80% Passing Score: 80%

Question 1: ✔ Correct

Which of the following attacks is a form of software exploitation that transmits or submits a longer stream of data than the input variable is designed to handle?

- Data diddling
- Buffer overflow attack
- Smurf attack
- Time-of-check to time-of-use attack

EXPLANATION

A buffer overflow occurs when software code receives more input than it was designed to handle. This normally occurs because the programmer of that code failed to include input validation checks. When a buffer overflow occurs, the extra data is pushed into the execution stack and processed with the security context of the system itself. In other words, a buffer overflow attack often allows the attacker to perform any operation on a system.

A time-of-check to time-of-use (TOCTOU) attack occurs when the results of an attack are realized or initiated after the attack itself is perpetrated. Data diddling is the purposeful altering of data. A smurf attack is a form of distributed-reflective denial of service.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_buffer_01_secp7.question.fex

Question 2: ✔ Correct

Having poor software development practices and failing to program input validation checks during development of custom software can result in a system vulnerable to which type of attack?

- Superzapping
- Denial-of-service attack
- Buffer overflow attack
- Dictionary attack

EXPLANATION

Poor software development practices and failing to program input validation checks can leave a system vulnerable to buffer overflow attacks. A buffer overflow occurs when software code receives more input than it was designed to handle because the programmer of that code failed to include input validation checks. When a buffer overflow occurs, the extra data is pushed into the execution stack and processed with the security context of the system itself. In other words, a buffer overflow attack often allows the attacker to perform any operation on a system.

Denial-of-service attacks exploit vulnerabilities in implementation and coding errors. Dictionary attacks are waged against logon prompts or stolen copies of a security account's database. Superzapping attacks are specific attacks that use a specialized utility named superzap to bypass the security of IBM mainframes to perform system alterations.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_buffer_03_secp7.question.fex

Question 3: ✕ Incorrect

Which type of attack is the act of exploiting a software program's free acceptance of input in order to execute arbitrary code on a target?

- Data diddling
- Buffer overflow attack
- ~~TOCTOU~~
- Covert channel exploitation

EXPLANATION

The act of exploiting a software program's free acceptance of input in order to execute arbitrary code on a target is called a buffer overflow.

Data diddling is the change or corruption of data. TOC/TOU is a logon session replay attack. Covert channel exploitation is the use of timing or storage mechanisms to bypass security controls in order to leak information out of a secured environment.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_buffer_04_secp7.question.fex

Question 4: ✔ Correct

Which of the following is an attack that injects malicious scripts into web pages to redirect users to fake websites to gather personal information?

- Drive-by download
- SQL injection
- XSS
- DLL injection

EXPLANATION

Cross-site scripting (XSS) is an attack that injects scripts into web pages. When a user views the web page, the malicious scripts run, allowing the attacker to capture information or perform other actions.

- XSS often relies on social engineering or phishing to entice users to click on links to web pages that contain the malicious scripts.
- Some scripts redirect users to legitimate websites, but run in the background to capture information sent to the legitimate site.
- Scripts can be written to read (steal) cookies that contain identity information (such as session information).
- Scripts can also be designed to run under the security context of the current user. For example, scripts might execute with full privileges on the local system, or the scripts might run using the credentials used on a financial website.

A drive-by download is an attack where software or malware is downloaded and installed without explicit consent from the user. An SQL injection attack occurs when an attacker includes database commands within user data input fields on a form, and those commands subsequently execute on the server. A DLL injection attack occurs when a program is forced to load a dynamic-link library (DLL). This DLL then executes under the security context of the running application, and executes malicious code included with the injected DLL.

REFERENCES

10.3.11 Preventing Cross-Site Scripting

10.3.14 Web Application Attack Facts

q_webattk_cross_secp7.question.fex

Question 5: ✔ Correct

Which of the following is specifically meant to ensure that a program operates on clean, correct, and useful data?

- Application hardening
- Process spawning
- Error and exception handling
- Input validation

EXPLANATION

Input validation is the process of ensuring that a program operates on clean, correct, and useful data. Input validation uses routines (also called validation rules or check routines) that check for correctness, meaningfulness, and secureness in data input to the system.

Application hardening is the process of preventing vulnerability exploitation in software applications. Error and exception handling is a programming language construct designed to handle the occurrence of exceptions (which are special conditions that change the normal flow of program execution). Process spawning is the creation of a new process (also called a child process) by an existing process (also called a parent process).

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_input_secp7.question.fex

Question 6: ✔ Correct

You have a website that accepts input from users for creating customer accounts. Input on the form is passed to a database server where the user account information is stored.

An attacker is able to insert database commands in the input fields and have those commands execute on the server.

Which type of attack has occurred?

- Buffer overflow
- DLL injection
- Cross-site scripting
- SQL injection

EXPLANATION

A SQL injection attack occurs when an attacker includes database commands within user data input fields on a form, and those commands subsequently execute on the server. The injection attack succeeds if the server does not properly validate the input to restrict entry of characters that could end and begin a database command. SQL injection attacks are prevented by proper programming methods that prevent commands from occurring within form data or that filter data to prevent such attacks.

A buffer overflow occurs when an operating system or application does not properly enforce boundaries for how much and which type of data can be inputted. Hackers submit data beyond the size reserved for the data in the memory buffer, and the extra data overwrites adjacent memory locations. The extra data sent by the attacker could include executable code that might then be able to execute in privileged mode.

Cross-site scripting (XSS) is an attack that injects scripts into web pages. When the user views the web page, the malicious scripts run, allowing the attacker to capture information or perform other actions. A DLL injection attack occurs when a program is forced to load a dynamic-link library (DLL). This DLL then executes under the security context of the running application and executes malicious code included with the injected DLL.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_sql_02_secp7.question.fex

Question 7: ✕ Incorrect

An attacker inserts SQL database commands into a data input field of an order form used by a web-based application. When submitted, these commands are executed on the remote database server, causing customer contact information from the database to be sent to the malicious user's web browser.

Which practice would have prevented this exploit?

- Using the latest browser version and patch level
- Installing antivirus, anti-spyware, pop-up blockers, and firewall software
- Implementing client-side validation
- ~~Implementing a script blocker~~

EXPLANATION

Client-side validation should have been used on the local system to identify input errors in the order form before the data was ever sent to the server. In this example, if the user entered SQL commands in an order form field, the error would have been immediately detected and blocked before the data was submitted to the server.

Using the latest browser version and patch level, installing anti-malware software, and using a script blocker are valuable security measures. But these would not have prevented the exploit in this scenario.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_sql_03_secp7.question.fex

Question 8: ✔ Correct

Which of the following functions does a single quote (') perform in an SQL injection?

- Indicates that everything after the single quote is a comment
- ➡ Indicates that data has ended and a command is beginning
- Indicates that code is ending and a comment is being entered
- Indicates that the comment has ended and data is being entered

EXPLANATION

A single quote (') indicates that data has ended and a command is beginning.

The double dashes (--) indicate that code is ending and a comment is being entered. Comments are code that a program does not execute and are usually used for explanations or reminders for the coder. Applications know to ignore the comments.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_sql_06_secp7.question.fex

Question 9: ✔ Correct

As you browse the internet, you notice that when you go to some sites, multiple additional windows are opened automatically. Many of these windows contain advertisements for products that are inappropriate for your family to view.

Which tool can you implement to prevent these windows from showing?

- Anti-spyware
- Anti-adware
- Pop-up blocker
- Antivirus
- Phishing filter

EXPLANATION

Use a pop-up blocker to prevent windows from automatically opening when you visit a web site. Pop-up blockers typically do not block pop-ups that show when you click a button or a link, but they do prevent the pop-up windows that open automatically as you navigate to other sites.

Use antivirus software to scan attachments, downloads, or your system for malicious programs. Use anti-adware and anti-spyware software to prevent software that tracks your browsing history. While removing adware might prevent some pop-ups, it does not prevent all pop-ups unless the anti-adware software includes a pop-up blocker. Use a phishing filter to remove phishing emails or to prevent navigating to links that are disguised as legitimate links.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_web_01_secp7.question.fex

Question 10: ✔ Correct

While using a web-based order form, an attacker enters an unusually large value in the Quantity field.

The value he or she entered is so large that it exceeds the maximum value supported by the variable type used to store the quantity in the web application. This causes the value of the quantity variable to wrap around to the minimum possible value, which is a negative number.

As a result, the web application processes the order as a return instead of a purchase, and the attacker's account is credited with a large sum of money.

Which practices would have prevented this exploit? (Select two.)

- [ ] Using the latest browser version and patch level
- [x] Implementing server-side validation
- [ ] Installing antivirus, anti-spyware, pop-up blockers, and firewall software
- [x] Implementing client-side validation
- [ ] Installing the latest operating system updates

EXPLANATION

Client-side validation and server-side validation should have been used to identify input errors in the order form. In this example, if the user entered an invalid quantity in an order form field, client-side validation would have detected and blocked the error before the data was submitted to the server. Server-side validation should have also been used after the data was sent to the server to detect errors. Experienced attackers can circumvent client-side validation techniques by sending data to the server from outside the application's standard user interface, bypassing any input validation measures that may have been implemented on the client.

Using the latest browser version and patch level, installing the latest operating system updates, and using a script blocker are valuable security measures, but they would not have prevented the exploit in this scenario.

REFERENCES

10.3.14 Web Application Attack Facts

q_webattk_web_02_secp7.question.fex