CySA+
Cybersecurity Analyst

CCI
Post Office Box 9627
Mississippi State, MS 39762

CySA+

Part 4

Security Architectures

Secure Software Development

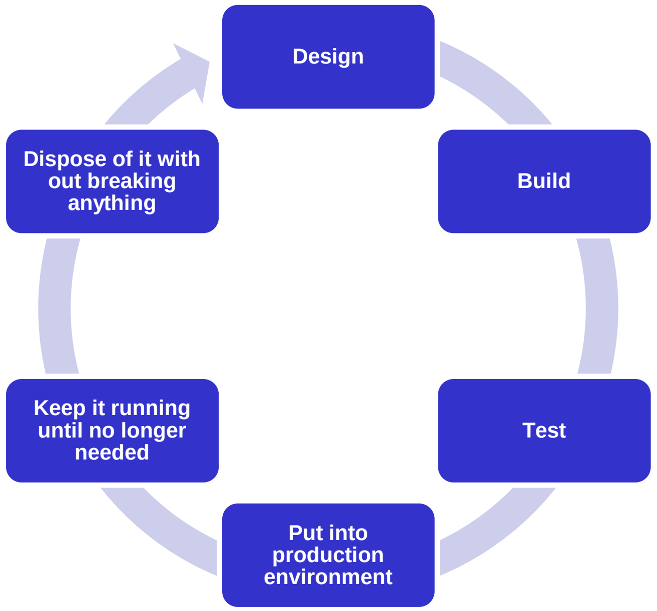
Chapter 14

Outline

- The Software Development Lifecycle
- Secure Software Development
- Best Practices
- Center for Internet Security
- Quiz

The Software Development Lifecycle

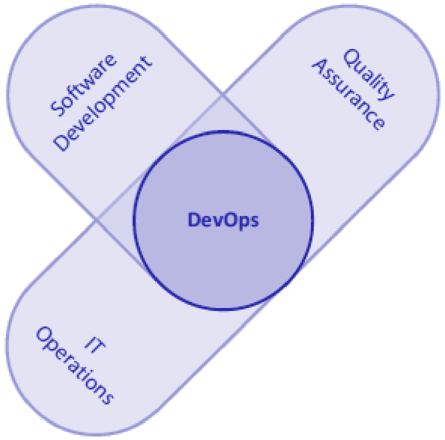
- Secure Software Development for an Analysts
  - Knowing what software developers do will help security analysts better understand insecure software development practices
- Software Development Lifecycle

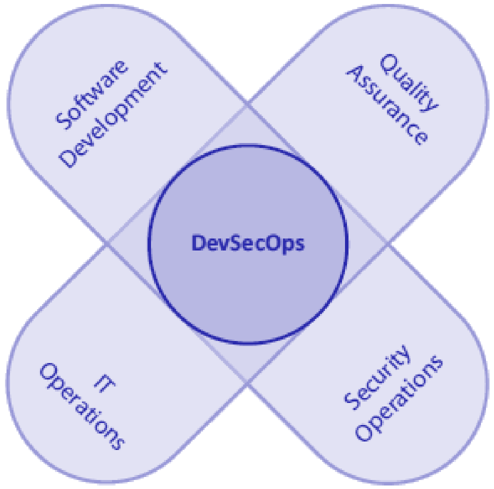

Design
Build
Test
Put into production environment
Keep it running until no longer needed
Dispose of it with out breaking anything

[1]

The Software Development Lifecycle

- **DevOps**
  - **Development team, IT team, and quality assurance (QA) team all working together**
  - **Incentive is to enable frequent, efficient, and reliable releases of software products**
- **DevSecOps**
  - **Includes the cybersecurity team**
  - **Incentive is to also create secure software products**





[1]

The Software Development Lifecycle

- Requirements
  - All software development should start with the identifying the requirements
- Functional Requirements
  - Defines what the software must preform
    - What inputs, processing, and outputs should the software have
- Nonfunctional Requirements
  - Defines how the software must preform
    - The characteristics, constraints, or limitations of the system
    - Sometimes called the quality requirements
- Security Requirements
  - Defines the behavior and characteristics the software must have to be at an acceptable level of security

[1]

The Software Development Lifecycle

- **Development**
  - **First, design an architecture that will address the nonfunctional requirements first**
  - **Second, design the functional requirements**
  - **Third implementation phase**
- **Implementation**
  - **Ensure DevOps or DevSecOps is used from the beginning**
    - **This ensures issues are dealt with before the implementation phase**
  - **User Acceptance Testing**
    - **Software is not accepted until the user declares that all features have been implemented as expected**
    - **Agile development methodologies continuously engages the user through out the development process**

[1]

The Software Development Lifecycle

- **Operation and Maintenance (O&M)**
  - **O&M represents ~75% of the total cost of ownership (TCO)**
    - **~35% of O&M costs are related to correcting vulnerabilities**
    - **~26% of the TCO is spent on fixing software defects**
  - **It is important to spend extra time on designing, secure developing, and testing the system before it goes to O&M**

[1]

Secure Software Development

- Secure Software Development
  - Not difficult, it just takes a lot of planning
  - Requires time, discipline, and attention to detail
    - This means significate cost for the organization
    - Reason for many organization developing insecure code
    - Fixing and maintaining bad code is more costly in the long run
- Secure Coding
  - It is nearly impossible to develop code that has no vulnerabilities
  - Securing coding is reducing vulnerabilities to a manageable degree
  - Implement controls in the operational environment to mitigate unexpected vulnerabilities

[1]

Secure Software Development

- Input Validation
  - Context-sensitive whitelisting
    - Allow only values that are expected
  - Client-side validation
    - Pros and Cons
      - Errors are caught at the point of entry
      - Tools can be used to negate validation
    - Client-side validation should be done to enhance the client's experience but not relied on for security
  - Server-side validation
    - It is best practice to validate any input on the server-side regardless of it being checked on the client-side
- Parameter Validation
  - Validating values coming in from the application
    - All values coming into function should have defined limits before being processed

[1]

Secure Software Development

- Static Code Analysis
  - A technique used to identify software defects or security policy violations
    - This analysis is done by examining the code without executing the code
  - Static Analysis
    - This term usually refers to automated tools that assist analysts and developers
    - IntelliJ IDEA
      - Java IDE with a feature that automatically highlights typical security errors in static code
        - » This is comparable to Microsoft Word highlighting grammatical errors in a Word document

[1]

Secure Software Development

- Code Reviews
  - This requires someone other than the author of the code manually examining the code
  - Best Practice
    - Use a combination of code review and static analysis
- Regression Testing
  - The formal process of testing modified code to ensure no features or security characteristics have been compromised or modified
  - Only as useful as the developed standardized suite of tests

[1]

Secure Software Development

- **Web App Vulnerability Scanning**
  - **Are external tests conducted from the perspective of the adversary**
  - **Common vulnerability checks**
    - **Outdated server components**
      - **E.g. servers with required patches**
    - **Misconfigured server**
    - **Secure authentication of users**
    - **Secure session management**
      - **E.g. Random session tokens**
    - **Information leaks**
      - **E.g. revealing too much information about the server**
    - **Cross-site scripting (XSS)**
    - **Improper use of HTTPS**

[1]

Secure Software Development

- Interception Proxies
  - A software tool place in between two communicating endpoints
    - This allows for examining, modifying, or logging messages between the two endpoints
  - Usually will be on the same network or the same host as one of the two endpoints
  - Used to exam the security of web apps and mobile apps
  - Typical security tests
    - Input validation
    - Parameter credentials
    - Plaintext credentials
    - Session Tokens

[1]

Secure Software Development

- **Fuzzing**
  - **Technique used to find flaws in the software**
  - **Fuzzers**
    - **Can sends a lot of random data to the target program to cause failures**
    - **It is more effective to construct semi-normal data and continuously make small changes to trigger failures**
    - **Adversaries manipulate these failures to inject their own code into the system**
  - **Commonly identified flaws are**
    - **Buffer overflows, DoS vulnerabilities, injection weaknesses, validation flaws, etc.**

[1]

Secure Software Development

- **Stress Testing**
  - **Determines the robustness of the software by placing it under extreme demands**
    - **The purpose is to compromise the availability of the system by overwhelming it with too much of something**
- **Resource Starvation**
  - **Attempts to compromise the availability of the system by depleting its resources**
  - **Commonly attacks**
    - **Network bandwidth**
      - **E.g DDoS attack**
    - **Memory**
      - **E.g. If the system has memory leaks this will deplete the availed memory**
    - **CPU**
      - **E.g. Causing the system to generate a lot of key pairs, would exhaust the CPU**

[1]

Best Practices

- Software Engineering Institute (SEI)
  - Founded at Carnegie-Mellon University
    - Federally funded by the DoD since 1984 due to the need for secure coding practices
  - Top ten secure coding practices developed by SEI
    1. Validate all input
    2. Don’t ignore compiler warnings
    3. Architect for security
    4. Avoid unnecessary complexity
    5. Deny by default
    6. Use least privilege
    7. Don’t share data you don’t have to
    8. Defend in depth
    9. Strive for quality
    10. Use secure coding standards

[1]

Best Practices

- Open Web Application Security Project (OWASP)
  - Is an organization that specializes in web security issues
  - OWASP has chapter meetings through out the world
    - These groups provide development guides, testing procedures, and code review steps
  - OWASP Top Ten Web-Based Software Vulnerabilities
    - Injection
    - Broken Authentication and Session Management
    - Cross-Site Scripting (XSS)
    - Insecure Direct Object Refreneces
    - Security Misconfiguration
    - Sensitive Data Exposure
    - Missing Function Level Access Controls
    - Cross-Site Request Forgery (CSRF)
    - Using Components with Known Vulnerabilities
    - Unvalidated Redirects and Forwards

[1]

Best Practices

- SANS Institute
  - Is an organization the specializes in information security and cybersecurity training
  - A list of SANS recommendations and best practices
    - Display generic error messages
      - Detailed error messages are very useful to an attacker
    - Implement account lockouts
      - Lock an account after so many failed password attempts
    - Limit sensitive data use
      - This can be a problem when form auto-completion is permitted
    - Use HTTPS everywhere
      - This protects data that transvers the Internet
    - Use parameterized SQL quires
      - Prevents SQL injection
    - Automate application deployment
      - Complex systems have dozens of configuration parameters it is important that every deployment is done exactly the same

[1]

Center for Internet Security

- Center for Internet Security (CIS)
  - Nonprofit organization that works to enhance the cybersecurity of private and public organizations around the world
  - It has four core divisions
    - The Integrated Intelligence Center
      - Receives threat intelligence from organizations and shares this information with subscribers
    - The Multi-State Information Sharing and Analysis Center (MS-ISAC)
      - Preforms similar functions but focuses on state, local, tribal, and territorial (SLTT) government partners
    - The Trusted Purchasing Alliance
      - Obtains deals that individual organizations would not be able to negotiate by leveraging its SLTT and nonprofit partners
    - Security Benchmark Division
      - Produces two product streams
        - » System design recommendations
        - » Benchmarks

[1]

Center for Internet Security

- **System Design Recommendations**
  - **CIS identifies best practices for secure system designs**
  - **CIS has 20 controls that can significantly improve an organization’s security posture**
  - **CIS offers guidance on how to implement the 20 CIS Controls in any organizations information system**
- **Benchmarks**
  - **CIS Benchmarks are detailed guides on securing a specific platform**
  - **CIS offers pre-hardened images of some open source platforms**

[1]

Quiz

Chapter 14

Question #1

- **1. The practice of testing user input to reduce the impact of malformed user requests is referred to as what?**
  A. **Input validation**
  B. **Static code analysis**
  C. **Manual inspection**
  D. **Stress testing**

[1]

Answer #1

- A
  - Input validation is an approach to protecting systems from abnormal use input by testing the data provided against appropriate values

[1]

Question #2

- **2. Which phase in the software development lifecycle often highlights friction between developers and business units due to integration and performance issues?**
  - **A. Implementation**
  - **B. Design**
  - **C. Planning**
  - **D. Maintenance**

[1]

Answer #2

- A
  - Implementation is all about seeing how the software works in its production environment
  - Although problems are bound to surface, the most productive organizations have mature mechanisms for feedback for improvement

[1]

Question #3

- **3. To reduce the amount of data that must be examined and interpreted by a web application, what method can be used to catch errors before submission?**
  A. Server-side validation
  B. Proxy validation
  C. Client-side validation
  D. Stress validation

[1]

Answer #3

- C
  - Client-side validation checks are those performed on data in the user browser or application before the data is sent to the server
  - This practice is used alongside server-side validation to improve security and to reduce the load on the server

[1]

Question #4

- **4. What key process is often used to determine the usability and suitability of newly develop4ed software before implementation across an organization?**
  A. User acceptance testing
  B. Parameter validation
  C. Regression testing
  D. Data filtering

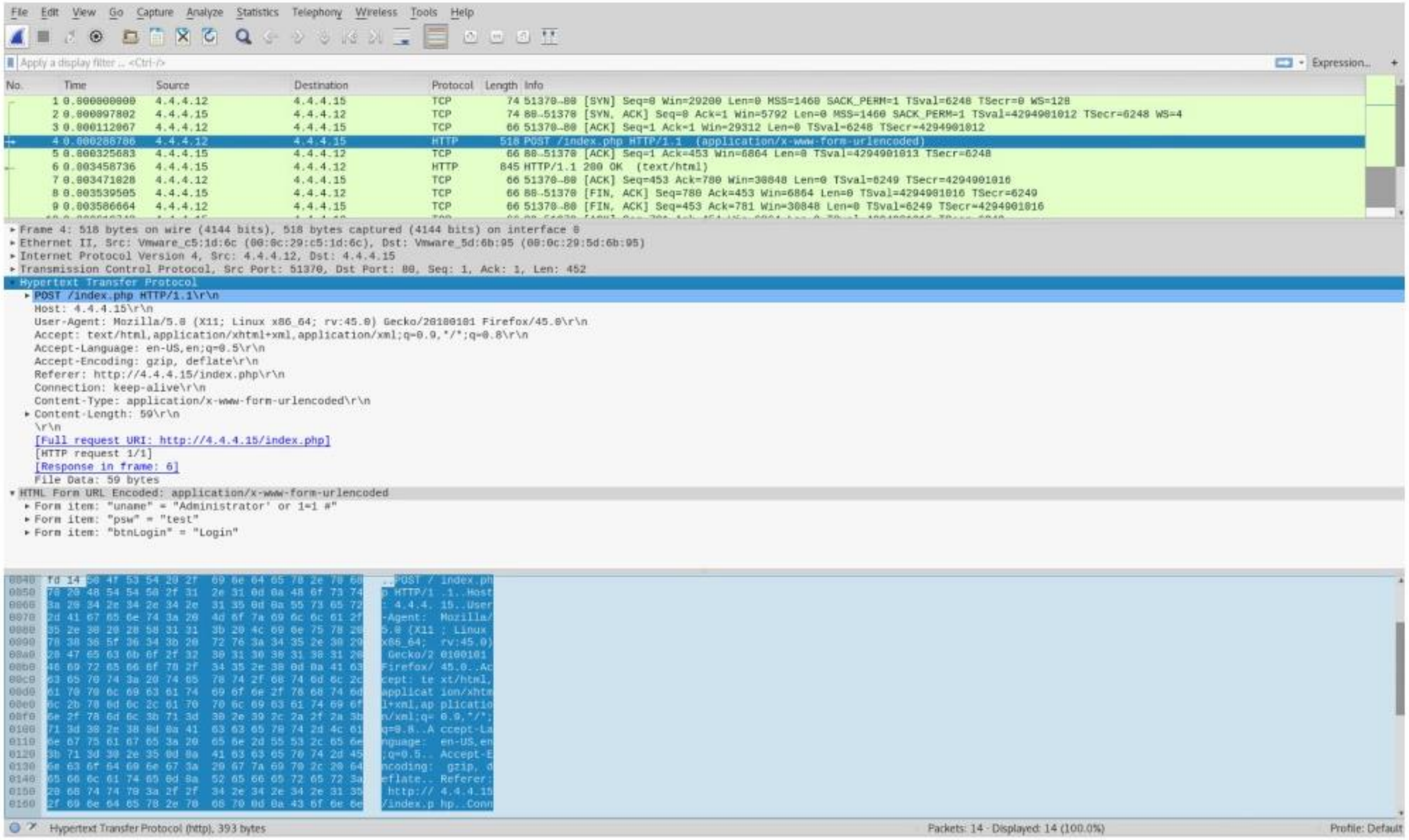
[1]

Answer #4

- A
  - User acceptance testing is a method to determine whether a piece of software meets specifications and is suitable for the business processes

[1]

Scenario for Questions 5-10


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

```
No.  Time         Source      Destination  Protocol Length Info
   1 0.000000000  4.4.4.12    4.4.4.15     TCP        74 51370→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=6248 TSecr=0 WS=128
   2 0.000097802  4.4.4.15    4.4.4.12     TCP        74 80→51370 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=4294901012 TSecr=6248 WS=4
   3 0.000112067  4.4.4.12    4.4.4.15     TCP        66 51370→80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=6248 TSecr=4294901012
   4 0.000286786  4.4.4.12    4.4.4.15     HTTP      518 POST /index.php HTTP/1.1  (application/x-www-form-urlencoded)
   5 0.000325683  4.4.4.15    4.4.4.12     TCP        66 80→51370 [ACK] Seq=1 Ack=453 Win=6864 Len=0 TSval=4294901013 TSecr=6248
   6 0.003458736  4.4.4.15    4.4.4.12     HTTP      845 HTTP/1.1 200 OK  (text/html)
   7 0.003471028  4.4.4.12    4.4.4.15     TCP        66 51370→80 [ACK] Seq=453 Ack=780 Win=30848 Len=0 TSval=6249 TSecr=4294901016
   8 0.003539505  4.4.4.15    4.4.4.12     TCP        66 80→51370 [FIN, ACK] Seq=780 Ack=453 Win=6864 Len=0 TSval=4294901016 TSecr=6249
   9 0.003586664  4.4.4.12    4.4.4.15     TCP        66 51370→80 [FIN, ACK] Seq=453 Ack=781 Win=30848 Len=0 TSval=6249 TSecr=4294901016
```

```
▸ Frame 4: 518 bytes on wire (4144 bits), 518 bytes captured (4144 bits) on interface 0
▸ Ethernet II, Src: Vmware_c5:1d:6c (00:0c:29:c5:1d:6c), Dst: Vmware_5d:6b:95 (00:0c:29:5d:6b:95)
▸ Internet Protocol Version 4, Src: 4.4.4.12, Dst: 4.4.4.15
▸ Transmission Control Protocol, Src Port: 51370, Dst Port: 80, Seq: 1, Ack: 1, Len: 452
▾ Hypertext Transfer Protocol
  ▸ POST /index.php HTTP/1.1\r\n
    Host: 4.4.4.15\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: http://4.4.4.15/index.php\r\n
    Connection: keep-alive\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  ▸ Content-Length: 59\r\n
    \r\n
    [Full request URI: http://4.4.4.15/index.php]
    [HTTP request 1/1]
    [Response in frame: 6]
    File Data: 59 bytes
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "uname" = "Administrator' or 1=1 #"
  ▸ Form item: "psw" = "test"
  ▸ Form item: "btnLogin" = "Login"
```

```
0040  fd 14 50 4f 53 54 20 2f  69 6e 64 65 78 2e 70 68   ..POST / index.ph
0050  70 20 48 54 54 50 2f 31  2e 31 0d 0a 48 6f 73 74   p HTTP/1 .1..Host
0060  3a 20 34 2e 34 2e 34 2e  31 35 0d 0a 55 73 65 72   : 4.4.4. 15..User
0070  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
0080  35 2e 30 20 28 58 31 31  3b 20 4c 69 6e 75 78 20   5.0 (X11 ; Linux 
0090  78 38 36 5f 36 34 3b 20  72 76 3a 34 35 2e 30 29   x86_64;  rv:45.0)
00a0  20 47 65 63 6b 6f 2f 32  30 31 30 30 31 30 31 20    Gecko/2 0100101 
00b0  46 69 72 65 66 6f 78 2f  34 35 2e 30 0d 0a 41 63   Firefox/ 45.0..Ac
00c0  63 65 70 74 3a 20 74 65  78 74 2f 68 74 6d 6c 2c   cept: te xt/html,
00d0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 68 74 6d   applicat ion/xhtm
00e0  6c 2b 78 6d 6c 2c 61 70  70 6c 69 63 61 74 69 6f   l+xml,ap plicatio
00f0  6e 2f 78 6d 6c 3b 71 3d  30 2e 39 2c 2a 2f 2a 3b   n/xml;q= 0.9,*/*;
0100  71 3d 30 2e 38 0d 0a 41  63 63 65 70 74 2d 4c 61   q=0.8..A ccept-La
0110  6e 67 75 61 67 65 3a 20  65 6e 2d 55 53 2c 65 6e   nguage:  en-US,en
0120  3b 71 3d 30 2e 35 0d 0a  41 63 63 65 70 74 2d 45   ;q=0.5.. Accept-E
0130  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 20 64   ncoding:  gzip, d
0140  65 66 6c 61 74 65 0d 0a  52 65 66 65 72 65 72 3a   eflate.. Referer:
0150  20 68 74 74 70 3a 2f 2f  34 2e 34 2e 34 2e 31 35    http:// 4.4.4.15
0160  2f 69 6e 64 65 78 2e 70  68 70 0d 0a 43 6f 6e 6e   /index.p hp..Conn
```

Hypertext Transfer Protocol (http), 393 bytes   Packets: 14 · Displayed: 14 (100.0%)   Profile: Default

[1]

Scenario for Questions 5-10

- **Your accounting department's administrator has reached out to your team because one of the department's analysts has discovered a discrepancy in the accounting reports. Some of the department's paper documents do not match the stored versions, leading them to believe the database has been tampered with. This database is for internal access only, and you can assume that it hasn't been accessed from outside the corporate network. The administrator tells you that the database software was written several years ago by one individual and that they haven't been able to update the system since the initial rollout. You are also provided traffic capture data by the local admin to assist with the analysis.**

[1]

Question #5

- **5. After hearing the description of how the software was developed by one person, what process do you know would have improved the software without needing to run it?**
  A. **Runtime analysis**
  B. **Just-in-time analysis**
  C. **Fuzzing**
  D. **Code review**

[1]

Answer #5

- D
  - Code review is the systematic examination of software by someone not involved in the initial development process
  - This ensures an unbiased perspective and promotes adherence to coding and security standards

[1]

Question #6

- **6. What tool might you employ to monitor communication between endpoints and the server to observe exchanges and assist you in discovering the flaw?**
  A. Jump box
  B. Interception proxy
  C. Regression testing
  D. Client-side validation

[1]

Answer #6

- B
  - An interception proxy is a software tool that is inserted between two endpoints, usually on the same network, to monitor traffic and help with security testing

[1]

Question #7

- **7. Based on the traffic-capture data, what is the most likely method used to gain unauthorized access to the web application?**
  A. **Regression**
  B. **Replay attack**
  C. **SQL injection**
  D. **Request forgery**

[1]

Answer #7

- C
  - SQL injection is a technique of manipulating input to gain control of a web application's database server
  - It is effective and powerful, and often facilitates data manipulation or theft

[1]

Question #8

- **8. What practice might have prevented this particular type attack from being successful?**
  A. Network segmentation
  B. SSL
  C. Two-factor authentication
  D. Input validation

[1]

Answer #8

- D
  - Input validation is the practice of constraining and sanitizing input data
  - This is an effective defense against all types of injection attacks by checking the type, length, format, and range of data against known good types

[1]

Question #9

- **9. To prevent input from being interpreted as actual commands, what method should the developer have used?**
  A. Regression testing
  B. Generic error messages
  C. Session tokens
  D. Parameterized queries

[1]

Answer #9

- D
  - Using parameterized queries is a developer practice for easily differentiating between code and user-provided input

[1]

Question #10

- **10. You have updated the server software and want to actively test it for new flaws. Which method is the least suitable for your requirement?**
  - A. Fuzzing
  - B. Static code analysis
  - C. Stress testing
  - D. Web app vulnerability scanning

[1]

Answer #10

- B
  - All methods except for static code analysis are considered active types of assessments

[1]

References

1. Maymí Fernando, and Brent Chapman. CompTIA CSA+ Cybersecurity Analyst Certification All-in-One Exam Guide (Exam CS0-001). McGraw-Hill Education, 2018.