Exam Report: 8.6.12 Practice Questions

Date: 3/31/28 10:41:12 pm
Time Spent: 2:04

Candidate: Garsteck, Matthew
Login: mGarsteck

Overall Performance

Your Score: 50%

Passing Score: 80%

View results by: Objective Analysis Individual Responses

Individual Responses

Question 1: Incorrect

Which of the following enters random data to the inputs of an application?

- Validation rules
- ~~Application hardening~~
- Fuzzing
- Routines

Explanation

Fuzz testing (also known as *fuzzing*) is a software testing technique that exposes security problems by providing invalid, unexpected, or random data to the inputs of an application. Fuzzing programs come in two types:

- *Mutation-based* programs, which mutate existing data samples to create test data.
- *Generation-based* programs, which define new test data based on models of the input.

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.

References

LabSim for Security Pro, Section 8.6.
[All Questions SecPro2017_v6.exm APP_DEV_01]

Question 2: Correct

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

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

LabSim for Security Pro, Section 8.6.
[All Questions SecPro2017_v6.exm APP_DEV_02]

Question 3: Incorrect

During the application development cycle, an application tester creates multiple virtual machines on a hypervisor, each with a different version and edition of Windows installed. She then installs the latest build of the application being developed on each virtual machine and evaluates each installation for security vulnerabilities.

Which assessment technique was used in this scenario?

- Code review
- ~~Baseline reporting~~
- Configuration testing
- Fuzzing

Explanation

Configuration testing is the process of testing an application under development on systems that have various combinations of hardware and software. In this scenario, the tester is evaluating the security of the application on various versions and editions of the Windows operating system.

A code review is systematic examination of an application's source code. It is intended to find and fix overlooked mistakes, improving the overall quality and security of software. A code review is sometimes called a *peer review*. Fuzzing is a software testing technique that exposes security problems by providing invalid, unexpected, or random data to the inputs of an application. Input validation is the process of ensuring that a program operates on clean, correct, and useful data.

References

LabSim for Security Pro, Section 8.6.
[All Questions SecPro2017_v6.exm APP_DEV_03]

Question 4: Correct

During the application development cycle, a developer asks several of his peers to assess the portion of the application he was assigned to write for security vulnerabilities.

Which assessment technique was used in this scenario?

- Code review
- Fuzzing
- Baseline reporting
- Input validation

Explanation

A code review is a systematic examination of an application's source code. It is intended to find and fix overlooked mistakes, improving the overall quality and security of software. A code review is sometimes called a *peer review*.

Baselines are of a set of consistent requirements that establish a standard configuration for all systems. With a baseline established, you can more easily identify abnormal activity and areas that need improvement. Fuzzing is a software testing technique that exposes security problems by providing invalid, unexpected, or random data to the inputs of an application. Input validation is the process of ensuring that a program operates on clean, correct, and useful data.

References

LabSim for Security Pro, Section 8.6.
[All Questions SecPro2017_v6.exm APP_DEV_04]

Question 5: Correct

You've been assigned to evaluate NoSQL databases as a part of a big data analysis initiative in your organization.

You've downloaded an open source NoSQL database from the internet and installed it on a test system in an isolated lab environment.

Which of the following are likely to be true about this test system? (Select two.)

- [x] Data is stored in the database in an unencrypted format.
- [ ] The database is more susceptible to SQL injection attacks than traditional SQL databases.
- [ ] The default admin user password is admin.
- [ ] By default, data is stored in the database in an encrypted format.
- [x] The database admin user has no password assigned.

Explanation

Because NoSQL is a relatively new database architecture and less mature than traditional SQL, there are several key security issues to be aware of:

• Most NoSQL database implementations do not require authentication by default, even for the admin user. After installation, user account passwords must be manually configured by the database administrator. Otherwise, anyone can access the database and view the information it contains without supplying a password.
• In most NoSQL database implementations, any logged in user can access every collection on the server by default. The database administrator must manually configure access controls to prevent unauthorized access to the information in the database.
• Most NoSQL database implementations do not implement data encryption for either data at rest or data in transit.
• Because NoSQL does not support most aspects of the SQL language, NoSQL databases are less susceptible to SQL injection attacks when compared to traditional SQL database implementations.

References

LabSim for Security Pro, Section 8.6.
[All Questions SecPro2017_v6.exm APP_DEV_05]

Question 6: Incorrect

You've been given an assignment to evaluate NoSQL databases as a part of a big data analysis initiative in your organization.

You've downloaded an open source NoSQL database from the internet and installed it on a test system in an isolated lab environment.

What should you do to harden this database before implementing it in a production environment? (Select two.)

- [x] Implement an application-layer protocol to encrypt data prior to saving it in the database
- [ ] Implement an IDS to detect SQL injection attacks on the database
- [x] ~~Enable data encryption in the database configuration~~
- [ ] Disable anonymous access
- [ ] Enable anonymous access

Explanation

To harden a NoSQL implementation, consider using the following measures:

- Configure user accounts and assign strong passwords to them.
- Disable anonymous access and require authentication.
- Configure access controls to restrict access based on the user account.
- Encrypt data using an Application-layer protocol prior to saving it in the database. Databases don't like encryption, but this ensures that the data at rest is stored in an encrypted format within the database.
- Encrypt data in transit using SSL.
- Implement a NoSQL database only in in a hardened, secure environment protected by traditional network security mechanisms, such as firewalls, VLANs, and ACLs. NoSQL databases have limited security.

Because NoSQL does not support most aspects of the SQL language, NoSQL databases are less susceptible to SQL injection attacks when compared to traditional SQL database implementations.

References

LabSim for Security Pro, Section 8.6.
[All Questions SecPro2017_v6.exm APP_DEV_06]