Domain 8: Software Development Security

MIS-5903

https://community.mis.temple.edu/mis5903sec711summer2021/

1

Perimeter Controls

- Programmers do not implement security in code development
- Many security professionals are not software developers
- Most software developers are not security professionals
- Software vendors trying to get products to market in quickest time
- We're used to receiving software with flaws and applying patches

2

Mainframes didn't require security because

- Only a handful knew how to run them
- Users worked on (dumb) terminals that could not introduce malicious code
- Environments were closed

3

Environment vs. Application

- Operating system controls can be circumvented within application
- Firewalls and access controls prevent attackers' exploitation of known vulnerabilities
- Application and DB Management controls are specific to needs
- Security controls can be built within the application

4

Functionality vs. Security – Happy Medium

- Perform necessary calculations on business data
- Rogue functionality could lead to data loss.
- Security controls slow down and/or prevent functionality
- Checking input data minimizes program malfunction

5

Software Development Life Cycle

- Requirements gathering – why? What? For whom?
- Design – how the software will accomplish identified goals?
- Development – programming to meet specifications laid out during Design phase. Integration of new code with existing software.
- Testing/Validation – verifying that software works as planned and that goals are set.
- Release/Maintenance – deploying software, ensuring software is properly configured, patched, and monitored.
- Identification
  - Subjects supply identification information
  - Username, user ID, account number
- Authentication
  - Verifying the identification information
  - Passphrase, PIN value, thumbprint, smart card, one-time password
- Authorization
  - Using identity of subject with other criteria to determine authorized actions
  - "I know who you are, now what can you do?"
- Accountability
  - Audit logs and monitoring to track subject activities with objects

6

Project Plan

- Security plan – ensures that security is not overlooked
- Statement of Work (SOW)
- Work Breakdown Structure (WBS)

7

Requirements Gathering – Security Plan

- Security requirements
- Security risk assessment
- Privacy Risk Assessment
- Privacy Impact Rating (PIR)
  - P1, High Privacy Risk
  - P2, Moderate Privacy Link (user-initiated)
  - P3, Low Privacy Risk (no PII stored on machine)
- Risk-level acceptance

8

Design Phase

- Software Requirements use models:
  - Informational – the type of information to be processed and how it will be processed
  - Functional – outlines tasks and functions the application needs to perform
  - Behavioral – Explains the states the application will be during execution
- Security Plan
  - Attack Surface Analysis
  - Threat Modeling

9

Development Phase

- Computer Aided Software Engineering (CASE) tools
- Secure Coding
  - CWE/SANS Top 25 Most Dangeroud Software Errors https://www.sans.org/top25-software-errors/
- Input Validation
- Buffer Overflows

10

Testing/Validation

- Test against vulnerabilities identified
- Separation of Duties – environments (dev, testing, production)
- Testing Types
  - Unit testing – individual components in a controlled environment
  - Integration testing – components work together as per design specifications
  - Acceptance testing – code meets customer requirements
  - Regression testing – after a change, retesting to ensure functionality, performance, and protection

11

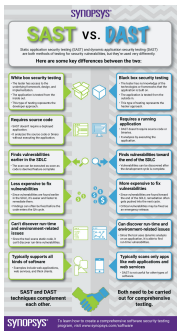
Testing Methods

- Fuzzers – use complex input to impair program execution
  - Large amounts of malformed, unexpected, or random data
- Manual testing – look for logical errors
  - Attackers manipulate program flow by using special program sequences
  - Code auditing by security-centric programmers
  - Dynamic analysis – real time, when running

12

Automated Testing

- Static Application Software Testing
- Dynamic Application Software Testing



13

Release/Maintenance

- Newly discovered problems
- Interoperation with environment

14

Open Web Application Security Project (OWASP)

- Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfigurations
- Sensitive Data Exposure
- Missing Function Levels Across Controls
- Cross-Site Request Forgery
- Using Components with Known Vulnerabilities
- Unvalidated Redirects

15

Initiatives

- U.S. Department of Homeland Security (DHS)
  - Build Security In (BSI)

16

Software Development Models – Build & Fix

- no architectural design/planning
- Not a formal SDLC model; SDLC is hardly involved
- No feedback mechanisms to allow for improvement

17

Software Development Models – Waterfall

- Uses a linear-sequential life-cycle
- ALL requirements gathered in initial phase; no formal way to integrate changes
- Rigid approach
- All requirements must be fully understood
- Dangerous for large projects
- Advantageous for smaller projects with full requirements

18

Software Development Models – V-Shaped (V-Model)

- Built upon Waterfall
- Higher chance of success
- Requires testing throughout the phases, not just the end

19

Software Development Models – Prototyping

- Rapid prototype – quickly create a sample, gain understanding
  - Aka throwaway – prototype is not built upon, but thrown away
- Evolutionary prototype – feedback gained through phases used to get closer to final stage
- Operational prototypes – designed to be used in a production environment as being tweaked

20

Software Development Models – Incremental

- Multiple development cycles
- Aka "Multi Waterfall"
- Each phase results in a deliverable that is an operational product
- Working software available at early development stages
- Changes can take place during each iteration
- Early understanding of risk, complexity, funding, functionality requirements

21

Software Development Models – Spiral

- Iterative approach
- Initial understanding and requirements
- New requirements can be added and addressed

22

Software Development Models – Rapid Application Development

- Combines prototyping and iterative development procedures
- Accelerating the software development process
- Customer is involved in the prototyping process

23

Software Development Models – Agile

- Umbrella term for several development methodologies
- Considered "lightweight"
- Focuses on individual interaction
- User stories
- Scrum - most widely adopted agile methodology
  - Projects of any size
  - Features can be added, changed, removed – at clearly defined points
    - SPRINT –
      - fixed duration development that is usually two weeks in length
      - with specific deliverables
  - Customer involvement, no surprises

24

Software Development Models – Agile (2)

- Extreme Programming (XP)
  - Reliance on test-driven development
  - Pair programming
  - Reduces errors, improves overall quality
- Kanban
  - Production scheduling system developed by Toyota
  - Adopted by IT
  - Visual tracking of all tasks so team can prioritize
  - Right features, right time
  - React to changing or unknown requirements

25

Other Models

- Exploratory Model – no clear objectives
- Joint Application Development (JAD) – team approach in a workshop-oriented environment
  - Includes members other than coders
- Reuse model – modifying pre-existing prototypes
  - Reduces development cost and time
- Cleanroom – structured and formal methods of developing and testing
  - Used for high-quality and mission-critical applications for certification

26

Integrated Protect Team

- Multidisciplinary development team with representation (e.g. accounts payable stakeholders)
- Management technique
- DevOps – changing the culture
  - Software Development
  - IT Operations
  - Quality Assurance

27

Capability Maturity Model Integration

1. Initial – ad-hoc development; heroics.
2. Repeatable – no formal process models defined
3. Defined – Formal processes carried out in each project
4. Managed – Formal processes, Quantitative Data fed into Improvement Program
5. Optimizing – budgeted and integrated plans for continuous improvements

28

Change Management

1. Make a formal request for a change.
2. Analyze the request
   a. Develop the implementation strategy
   b. Calculate the costs of this implementation
   c. Review security implications
3. Record the change request
4. Submit the change request for approval
5. Develop the Change
   a. Recode segments
   b. Link changes in code to formal change request
   c. Submit software for testing and quality control
   d. Repeat until quality is adequate
   e. Make version changes
6. Report results to management

29

Software Configuration Management (SCM)

- Identifies versions of software at points of time
- Versioning – revisions, roll-back
- Synchronization between multiple copies
- Code repository
- "Air Gapped"
- Software Escrow
  - Compiled code – not readable by humans

30

Software Generations

1. Machine Language
2. Assembly Language
3. High-Level Language
4. Very High-Level Language
5. Natural Language

31

Assemblers, Compilers, Interpreters

- Assembler – converts assembly language source code into machine code
- Compiler – converts high-level language statements into the necessary machine-level format for specific processors to understand.
  - One software may be compiled five times for five different systems
- Interpreters – platform independent, but cannot run on its own
  - Java Virtual Machine executes Java, for instance

32

Object Oriented Concepts

- Object Oriented Programming (OOP) works with
  - Classes – has attributes (color, size, cost) (e.g. furniture)
  - Objects – inherit class' attributes when instantiated (e.g. table)
  - *An object is an instantiation of a class*
  - Modularity – building blocks
  - Deferred commitment – internal components of an object can be redefined without changing other parts
  - Reusability – Classes are reused by other programs
  - Naturalness – Map to business needs and solutions
  - Shared Portion (API) messages
  - Private Portion (data hiding) ; encapsulated

33

Data Modeling & Structures

- Data Modeling considers data independently of how data is processed and components that process the data
- Data structure – logical relationship between elements of data
- Cohesion – how many different types of tasks a module can carry out
  - One or very similar – high cohesion
  - Multiple different tasks = low cohesion
- Coupling – level of interaction needed to carry out tasks
  - No communication = low coupling
  - High communication / High Changed = high coupling

34

Distributed Computing

- Distributed Computing Environment
  - Developed by the Open Software Foundation (OSF) also called Open Group
- Component Object Model (COM)
  - Object Linking and Embedding (OLE)
- Distributed Component Object Model (DCOM)
  - Replaced with .net framework (Common Language Runtime)
- CORBA – Common Object Request Broker Architecture
  - Object Management Group (OMG)
  - Two parts: Object Request Brokers (ORBs) and Object Services
- Java Platform Enterprise Edition (Java EE)
- Simple Object Access Protocol (SOAP) – XML-based; web environments

35

Mobile Code

- Java applets – run on web browser, java virtual machine sandbox
- ActiveX controls – relies upon digital certificates (Authenticode)

36

Web Security

- Administrative Interfaces
- Authentication and Access Control
- Input Validation
  - Path or directory traversal
  - Unicode encoding (%c1%1c)
  - URL encoding (%20)
  - Client-side validation (before upload); pre-validation
- Parameter Validation
- Session Management

37

Database Management

- Database – collection of data stored in a meaningful way
- Database Management System (DBMS)
- Models:
  - Relational
  - Hierarchical – no indexes, no links
  - Network – built upon hierarchical
  - Object-Oriented – store variety of data types, e.g. images, audio
  - Object-relational

38

Database Terms

- Record – collection of related data items
- File – collection of records of same time
- Database – cross-referenced collection of data
- Tuple – row in a two-dimensional database
- Attribute – Column in a two-dimensional database
- Primary Key – columns that make each row unique
- View – virtual relation defined by database administrator
- Foreign Key – attribute of one table that is related to the primary key of another table
- Cell – Intersection of row and column
- Schema – structure of the database
- Data Dictionary – central repository of elements and relationships

39

Database Programming Interfaces

- Open Database Connectivity (ODBC)
- Object Linking and Embedding Database (OLE DB)
- ActiveX Data Objects (ADO)
- Java Database Connectivity (JDBC)

40

Database Integrity

- Semantic integrity – data types, logical values, uniqueness constraints
- Referential integrity if all foreign keys reference existing primary keys
- Entity integrity – all tuples are uniquely identified by primary key

41

Database Security Issues

- Aggregation – user obtains sensitive information by piecing together two less-sensitive records
- Inference – deduction of the full story from pieces
- Content-Dependent Access Control
- Context-Dependent Access Control
- Database Views
- Polyinstantiation
- Views

42

Online Transaction Processing (OLTP)

- Atomicity – units of work; all (or none) modifications take effect
- Consistency – integrity policy followed
- Isolation – transactions execute in isolation until completed
- Durability – Once transaction is verified as accurate on all systems, the transaction is committed and cannot be rolled back

43

Data Warehousing / Mining / Big Data

- Data warehousing – combines data from multiple data sources into a large databases
  - Data is normalized; redundant data is removed
- Data Mining – metadata is produced to show unseen relationships
  - Metadata – data about the data
  - Knowledge Discovery in Database (KDD)
    - Classification
    - Probabilistic
    - Statistical
- Big Data – distinct term; very large data sets that are unsuitable for traditional analysis techniques

44

Next Steps…

- Continue Discussion on Class Website
- Prepare for Presentations
- Questions?

45