Lecture 2 🌱

> **Class**: Security policy and Software security
>
> **Date**: 11.30AM , 10 Nov
>
> **Author**: Lasal Hettiarachchi

Key learnings:

Definitions

- Busniness continuity planning(BCP)
  - The activities required to keep the organization running during a period of displacement or interruption of normal operations
  - Getting some form of business running
- Disaster Recovery Planning(DRP)
  - The process of rebuilding the operations or infrastructure after the disaster has passed.
  - Getting business running at 100%

Business contінity plan

- Focus on Availability of CIA
- Gets critical systems to another environment while repair of original facilities, gets the right people to the right places.
- Allow performing the business in a different mode until regular conditions are back in place.
- 3 Methods (Cost increases and the speed as we go down)
  - Cold site : an empty facility located offsite with required infrastructure
  - Hot site : a site with hardware, software and network installed and compatible to original site. (processes are not running)
  - Mirrored site : a site equipped with identical facilities to the original system with mirroring capability. Data is mirrored and backed up.
- Conflicting interests when creating a plan
  - Ensure redundancy
  - Minimize costs

Disaster recovery plan

- Focus on Availability of CIA
- Minimize the effect of the disaster by taking necessary steps to ensure resources, people,, and business processes are able to resume normal operations in a timely manner.
- This might take weeks or even months
- Require functionality for disaster recovery is built into the system

- How:
  - Non rupidiation
  - Backups
  - Dual factor authentication
  - Recovery of accounts

DRP - Key activities

- Identification of critical systems
- Data Backup and Restoration
  - Databases
  - Partial backups
  - Incremental backups
- Server and System administration
- System Shutdown and start-ups
  - Automatic shutdown and startup of systems

Disaster Recovery Goals

1. Risk Reduction
2. Resume operation
3. Addresing Investors/ Owner concerns
4. Industry Concerns

Disaster Recovery Types

Seperate plans required for different systems

1. Virtualized DRP
2. Network DRP
3. Cloud DRP
4. Data centre DRP

Benifits of DCP and DRP

- Reduce Risk
- Process improvements
- Improved organizational maturity
- Improved availability and reliability
- Marketplace advantage : competitive advantages

Software Security

Physical security , network security and administrative security

- Software security and application security are 2 things well first focus of software security

- software security : Concept of **implementing** mechanisms during the **construction** of software to help it remain functional (or resistant) to attacks. Software that is secure from design without having to add additional security elements to add additional layers of security
    - Holistic long term approach
    - Root cause analysis rather than threat modeling: why an exploit might happen
    - Organizational change
- Application security: After development of the software we take actions, vulnarability testng , mitigations. After design and development by adding additional layers. The way to secure the software after deployment is complete
    - Issue based short term approach : Issues faced by the company
    - Penetrate and patch: Ethical hacking
    - Threat modeling
    - Code review
    - Yearly penetration tests
    - Vulnerability assestment

Typres of software security

1. Network Security: The security between different devices located on the same network. In this case, both software security and hardware security are important.
2. Endpoint Security: Securing end user devices. This means that laptops, phones, computers, tablets, etc. are secure (again, both software and hardware) to avoid unwanted users sneaking in. Proper OS
3. Internet Security: This is what is commonly known as cybersecurity and deals with the transit and use of information.
4. Cloud Security: Secured hipervisors and cloud infrestucture

Practices of software security

- Concept of Least privilege
- Keep software up-to-date and patched
- Use of automation for software security tasks
- User Educatio
- Document, monitor, and measure: softwares integrate with other software therefore monitoring is important
- Make a plan for failure

Attacks on software

- leakage: Information leaving system
- Tampering: Unauthorixed information altering
- Resource stealing: illegal use of resources such as hardware
- Vandalism: disturbing correct system operation
    - Buffer overflow attacks

- Denial of service: Disrupting legitimate system use
    - Doing the attcak legitimately

Methods of attacks

- Eavesdropping: Obtaing messages copies without authority
- Masquerading(spoofing): using identity of another principle without authority
- Message tampering: changer messages in transition by intercepting and altering
- Replaying: Storing messages and sending them later.
    - Storing encrypted passwords and sending again
    - Challenge response authentication
- Flooding: sending too many messages
- Vulnerabilities: 41% - server 36% - non server

Issues with security breaches

1. Immediate financial loss
2. Reputation
3. Lawsuits

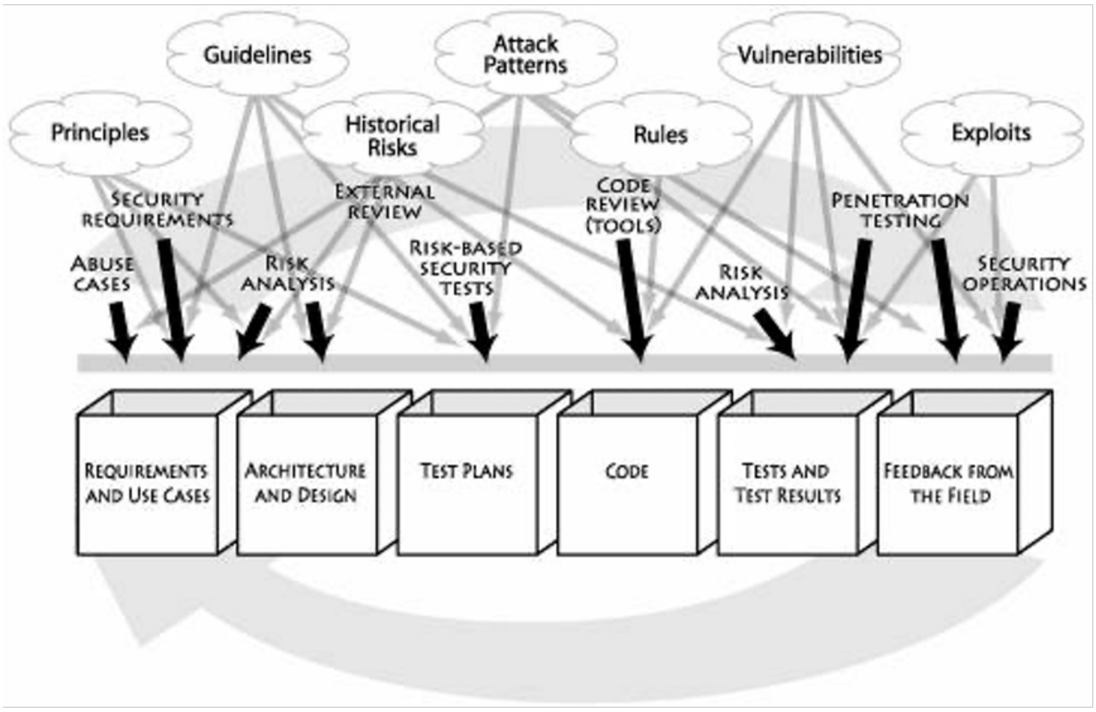
Why application security is not good enough

- Applying and managing security patches may be costly
- 0-day vulnerability: Vulnarability is known to the public but a patch has not been released
- Patches may not fix the cause
- Firewalls and IDSs may not be sufficient
- Build security in

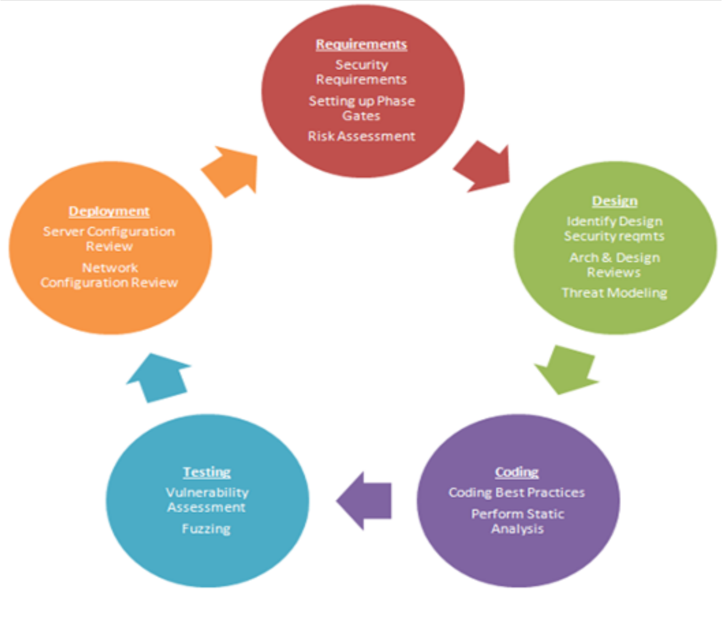
Software Security Lifecycle

S-SDLC improvement

- focus early on security in the LC → reduce cost in patching

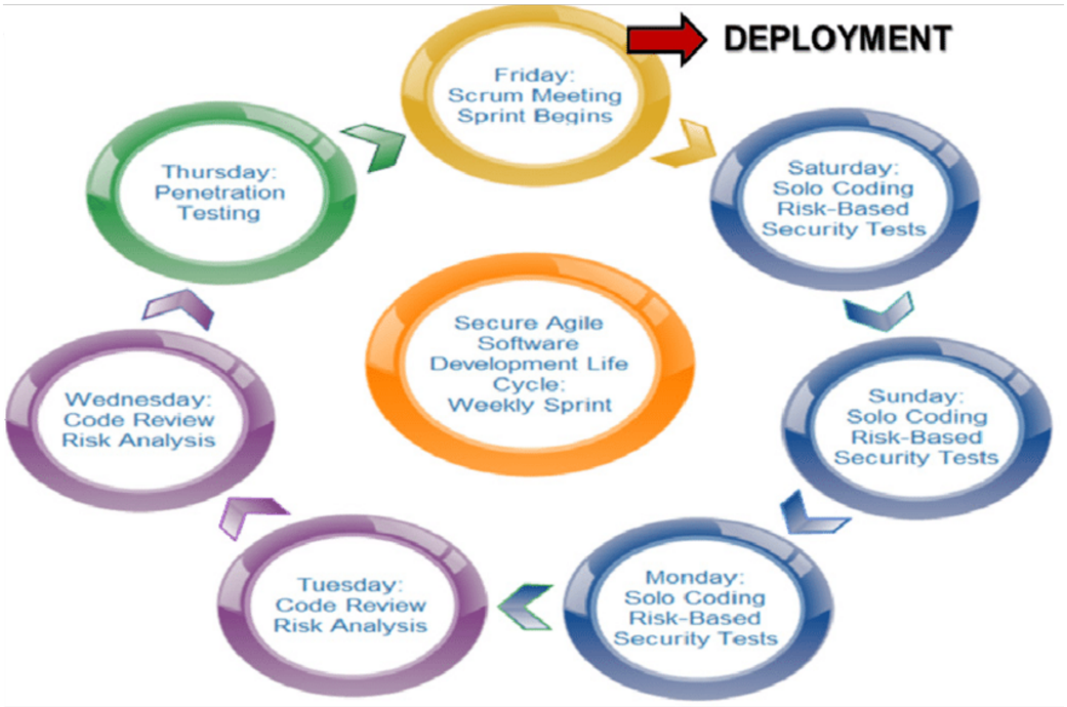


- Make security a mandatory part of SDLC and forcus on security on each phase(Theres no special SSDLC , SDLC is taken and adapted)



Sample S-SDLC

- In agile sprints level attention is given to functional requirements and security recuirements

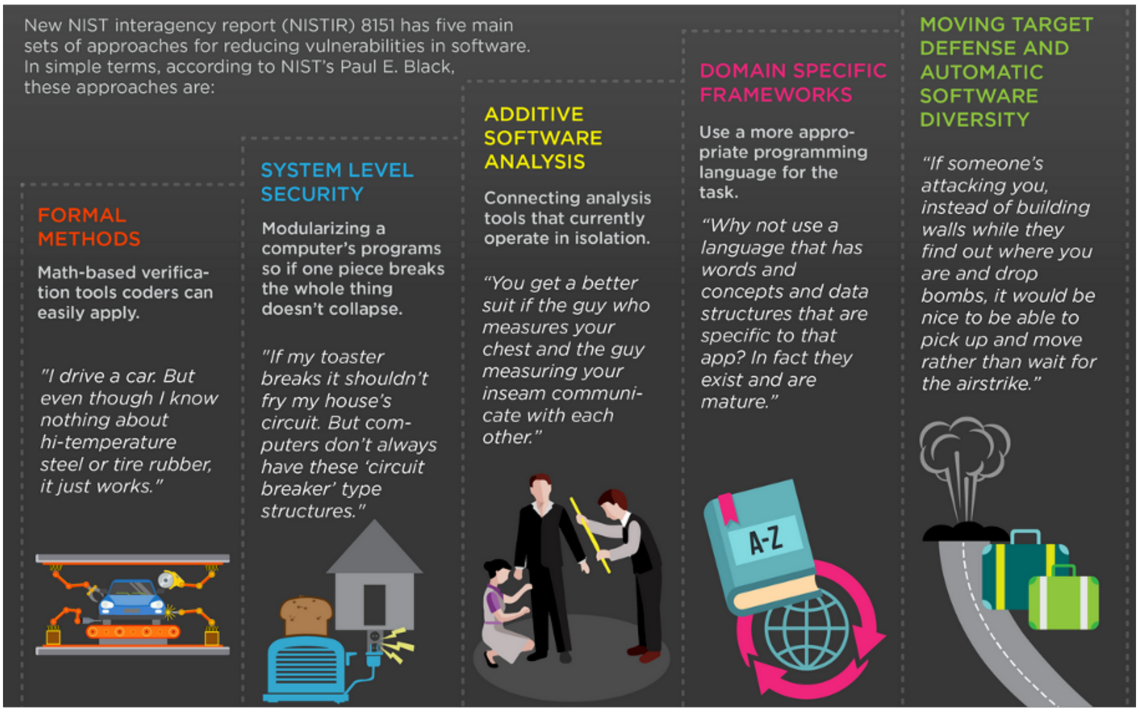

DEPLOYMENT
Friday: Scrum Meeting Sprint Begins
Saturday: Solo Coding Risk-Based Security Tests
Sunday: Solo Coding Risk-Based Security Tests
Monday: Solo Coding Risk-Based Security Tests
Tuesday: Code Review Risk Analysis
Wednesday: Code Review Risk Analysis
Thursday: Penetration Testing
Secure Agile Software Development Life Cycle: Weekly Sprint

How to implement a S-SDLC

- Get support from the project stakeholders
- Develop and deploy security policies and procedures
- Adapt the SDLC
- Provide security focus user training
- Security Service level agreement (Explain to the customer why security is important and why extra time and money is going)

Implementing software security

- Reducing vulnerabilities
- Formal methods: Math-based verification tools coders can easily apply.
    - Automate the security aspects
- System level security: Modularizing a computer's programs so if one piece breaks the whole thing doesn't collapse.
    - If single module is insecure it can be changed and updated
- Additive software analysis
    - Connecting analysis tools that currently operate in isolation.
    - Network security , server security , Intrution detection/prevention system.
- Domain specific frameworks:
    - Use a more appropriate programming language for the task.
- Moving target defence and automatic software diversity (Application security)
    - technologies are advancing, we need to update and patch our software
    - In S-SDLC maintance is also a part of the system



New NIST interagency report (NISTIR) 8151 has five main sets of approaches for reducing vulnerabilities in software. In simple terms, according to NIST's Paul E. Black, these approaches are:

FORMAL METHODS

Math-based verification tools coders can easily apply.

"I drive a car. But even though I know nothing about hi-temperature steel or tire rubber, it just works."

SYSTEM LEVEL SECURITY

Modularizing a computer's programs so if one piece breaks the whole thing doesn't collapse.

"If my toaster breaks it shouldn't fry my house's circuit. But computers don't always have these 'circuit breaker' type structures."

ADDITIVE SOFTWARE ANALYSIS

Connecting analysis tools that currently operate in isolation.

"You get a better suit if the guy who measures your chest and the guy measuring your inseam communicate with each other."

DOMAIN SPECIFIC FRAMEWORKS

Use a more appropriate programming language for the task.

"Why not use a language that has words and concepts and data structures that are specific to that app? In fact they exist and are mature."

MOVING TARGET DEFENSE AND AUTOMATIC SOFTWARE DIVERSITY

"If someone's attacking you, instead of building walls while they find out where you are and drop bombs, it would be nice to be able to pick up and move rather than wait for the airstrike."

Aspects of Software Security

1. Code level security(Input verificatio) :
2. Differences in Programming Languages and Operating Systems and how it affects to Software Development
    - confidentiality : Encryption
    - Integrity: DIgital signatures , hash functions
    - Access control: Biometrics, and password time tokes
    - Data in transforts: SSL(Secure socket layer),TLS(Transport layer security), PDP(Pretty good privacy)
3. Cryptography
4. Access Control Mechanisms
5. Security on data-at-rest

Build Security In is a collaborative effort that provides practices, tools, guidelines, rules, principles, and other resources that software developers, architects, and security practitioners can use to build security into software in every phase of its development.

Resources:

Common Vulnerability Scoring System Version 3.0 Calculator
https://www.first.org/cvss/calculator/3.0
https://www.first.org/cvss/calculator/3.0

WSO2 Secure Engineering Guidelines

This document summarizes the 'Secure Coding Guidelines' that should be followed by WSO2 engineers while engineering W...

https://wso2.com/technical-reports/wso2-secure-engineering-guidelines/

SLIIT CourseWeb: Log in to the site

https://courseweb.sliit.lk/pluginfile.php/104348/mod_resource/content/0/STRIDE_Reference_Sheets.pdf

https://courseweb.sliit.lk/pluginfile.php/104348/mod_resource/content/0/STRIDE_Reference_Sheets.pdf