Contents

[Lecture 15: Database 2](#_Toc155118701)

[Lecture 16: CSRF 2](#_Toc155118702)

[Lecture 17: XSS 2](#_Toc155118703)

[Lecture 18: Authentication 2](#_Toc155118704)

[Lecture 19: SSL 2](#_Toc155118705)

[Lecture 20: IPsec 2](#_Toc155118706)

[Intrusion Detection System 2](#_Toc155118707)

[Firewalls 2](#_Toc155118708)

[Ports 2](#_Toc155118709)

[Confidentiality Policy 2](#_Toc155118710)

# Lecture 15: Database

SQL Injection (SQLi) is performed by an attacker when he gives malicious queries(unexpected commands) to database server. This can result in

* DOS attack
* Execute arbitrary system commands.
* Modify or delete data

Types of attacks:

* In-band attack (tautology, piggyback, end of line)
* Out-of-band attack(email)
* Inferential attack(illegal queries( descriptive error page), blind sql injection(true/false questions by attacker to infer data))

Defenses:

* Encrypt Database
  + Full – bad idea
  + Every instance (value) – bad idea
  + Records or columns - best
* Database Access Control options
* Defensive Coding(stronger data validation checks in input/forms)
* Runtime Prevention
* Detection

Also take prevention against Inference attacks by

* Detect possible attacks in DB design
* Runtime query detection

# Lecture 16: CSRF

Cross Site Reference Forgery is an attack depending on the weakness of the web application occurring when a user’s cookie saved in browser can be used to access a trusted site by attacker without user’s knowledge. Its impact can vary from just accessing the trusted site to performing transactions or changing the credentials to fully take over.

Defenses:

* Not performing actions based on just saved cookies but instead authenticate user first for every session.
* STP (Synchronizer Token Pattern)
* Origin / Referrer Headers

# Lecture 17: XSS

Cross Site Scripting is an attack where user’s interaction with a vulnerable application can be compromised by an attacker (adds a malicious script). So an attack can do:

* Cookie Theft
* Keylogging
* Phishing

There are three types:

* Reflected

Reflected XSS involves the injection of a malicious script that is reflected back to the user in the web page's response.

* + Attacker stealing info with user interaction.
  + Providing the user with a bad link (a vulnerable site) and including a script in it.
  + The vulnerable site echoes back user’s input that can be accessed by attacker because of the script he included earlier.
  + Vulnerable site
  + Bad input
* Stored

Stored XSS involves permanently storing the script on the target server.

* + Attacker compromising server to steal info of user.
  + Malicious script is inserted into victim server.
  + User accesses vulnerable site himself and malicious script gets executed.
  + Attacker receives info.
* DOM Based

DOM-based XSS occurs when the injection and execution happen within the Document Object Model on the client side.

Defenses:

* Encoding

Escaping user input so browser only interpret it as data not code.

* Validation

Allowing code but filtering out malicious parts

* + Whitelisting/blacklisting
  + Sanitize/reject.
* Input handling context
* Inbound/outbound (validation before receiving/validation before sending)
* Client-Side/Server-Side (to prevent DOM based)
* Validate all headers, cookies, input fields, query strings etc.
* Adopt the policy “what is allowed”. (What is not allowed policy is difficult to manage and not fully possible)

# Lecture 18: Authentication

* Repudiable – password and id
* Non-repudiable - biometric
* Basic auth involving Server.
  + Password based (use salt)
  + Biometric (False acceptance, False rejection)
* Challenge-Response – Symmetric and Asymmetric
* Centralized auth – Kerberos (client, AS and TGS, Application Server)
* OATH (an organization developing open standards for security), adding multiple layers of security like two factor authentication.

# Lecture 19: SSL

At transport layer, above TCP

Handshake after TCP handshake

Client: Hello extension (list of algos, session key)

Server: Hello extension (selected algo, session key)

Client: Pre Master key (secret key either client provides or generated by p^ab mod q)

Client server exchange random number

Session Creation

Master keys = (secret, client random, server random)

Master key = Pseudo Random function (master keys)

Key + seq no = mac generation

Add to compressed data

So compressed data + mac + padding

Now perform encryption with algo key (optional)

* Protected data

# Lecture 20: IPsec

Defenses:

* Origin / Referrer Headers

# Intrusion Detection System

Three classes of intruders:

Masqueraders:

Not authorized

Penetrate system’s access controls.

Exploits a legitimate user’s account.

Usually outside

Misfeasors:

Legitimate user

Access not authorized OR

Misuses access

Usually Inside

Clandestine User:

Seizes supervisory control.

Evades auditing and access controls. OR

Suppresses audit collection.

Can be inside or outside.

Types of attacks 7

Components 3

Basic Principles

Lessen damage. (remove intruder asap)

Prevent intrusion. (act as deterrent)

Improve intrusion prevention measures from data collected.

Ways of classifying IDS

Anomaly Based

* + Collection of information about legitimate user
  + Then statistical tests to observe
  + Any distinct from usual behavior is intrusion
  + **Primary strength:** recognize novel attacks
  + Usually generate many false alarms
  + Does not require prior knowledge of security flaws

Misuse Detection/ Signature based

* + Observe events on system
  + Set of rules to identify attack
  + Auto analyze and generate rules for historical audit
  + Requires large database rules to be effective.
  + Does not require prior knowledge of security flaws

Specification Based - Hybrid Detection:

* + System behavior described through functionalities and policies
  + Any thing outside this is considered intrusion
  + Combines strengths of anomaly based detection(novel attacks detection ) and misuse detection (known attacks detection)
  + Specification development difficult and expensive.

Not proper detection – false security

Too false alarms – time waste

Types of IDS

Host based:

Monitors events on a particular host

Drawback: what info to record and what not (to reduce analysis burden)

System specific activity.

No additional hardware needed.

Network based:

Monitor network traffic, devices, and protocols.

Cost of ownership reduced.

Operating system independent.

Honey pots: lure a potential attacker away from critical system

# Firewalls

**Packet Filtering**

Rules defined (no match/default -> reject recommended), uses info on transport level only (packet details)

**Stateful Packet Inspection**

Extension to traditional packet filtering, instead of each packet maintains state (connection information) hence easy to define rules for connections instead of each packet , but also overhead

**Application Proxy**

Allows data in/out of process based on its type e.g allowing only HTTPS traffic to a website

Can act on single computer or network layer

More secure than packet filters

Overhead on each connection

**Circuit-level Proxy**

Relays TCP segments from one connection to other without examining contents.

Security functions consists of determining which connections will be allowed.

Firewall vs IDS

Firewalls are essential for controlling access and protecting against unauthorized entry

IDS play a crucial role in identifying and responding to suspicious activities that may evade the firewall’s initial barrier.

Both complement each other and together provide defense-in-depth.

Firewalls block traffic, focus on access control.

IDS alerts, focus on monitoring and detecting

## Ports

Port 80 – web sever

Port 25 – mail server

Port 53 – DNS traffic

# Confidentiality Policy