sait.ca

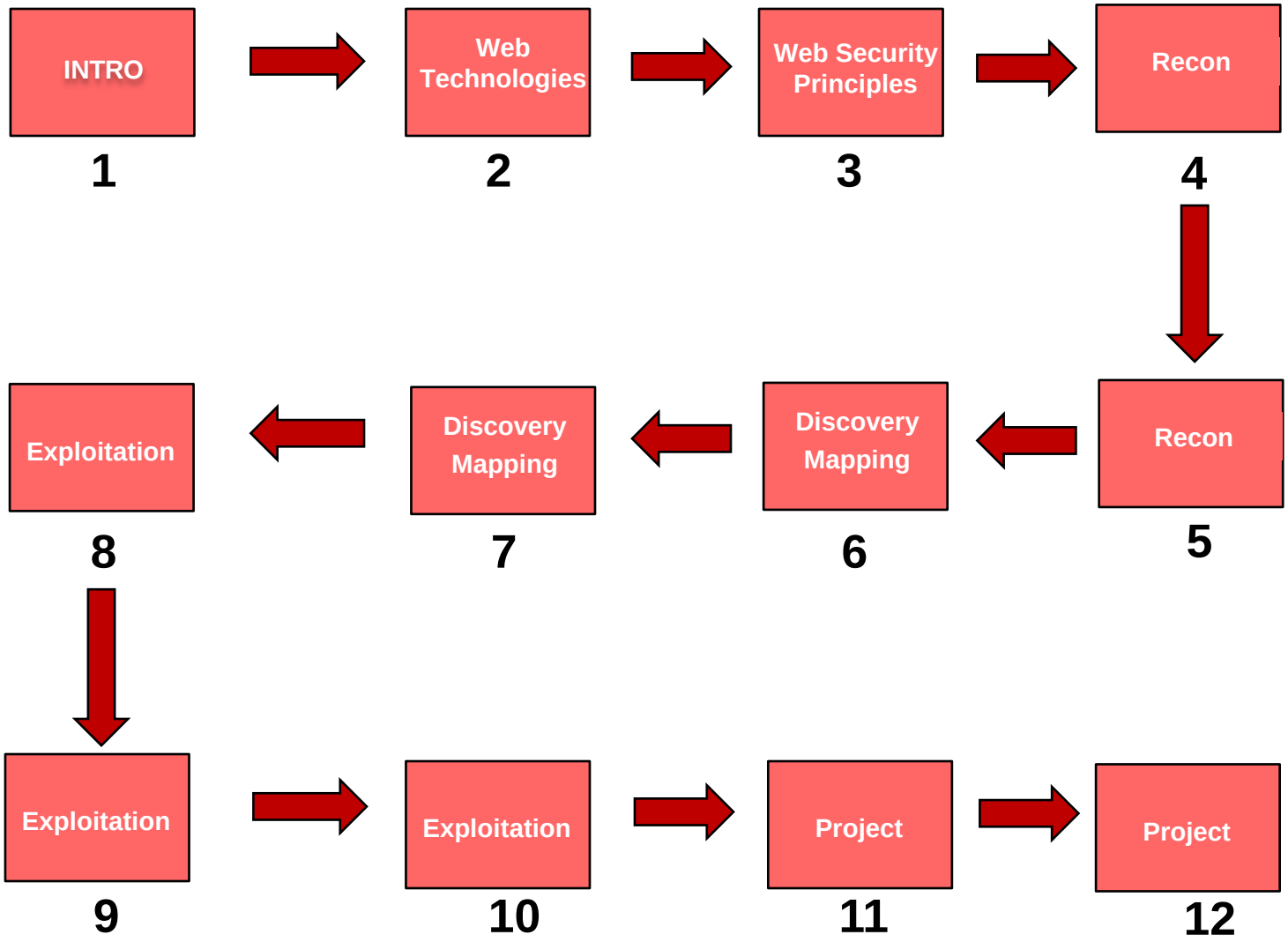
SAIT POLYTECHNIC

FURTHER YOUR PASSION

Web App & Data Base Security

Web Principles

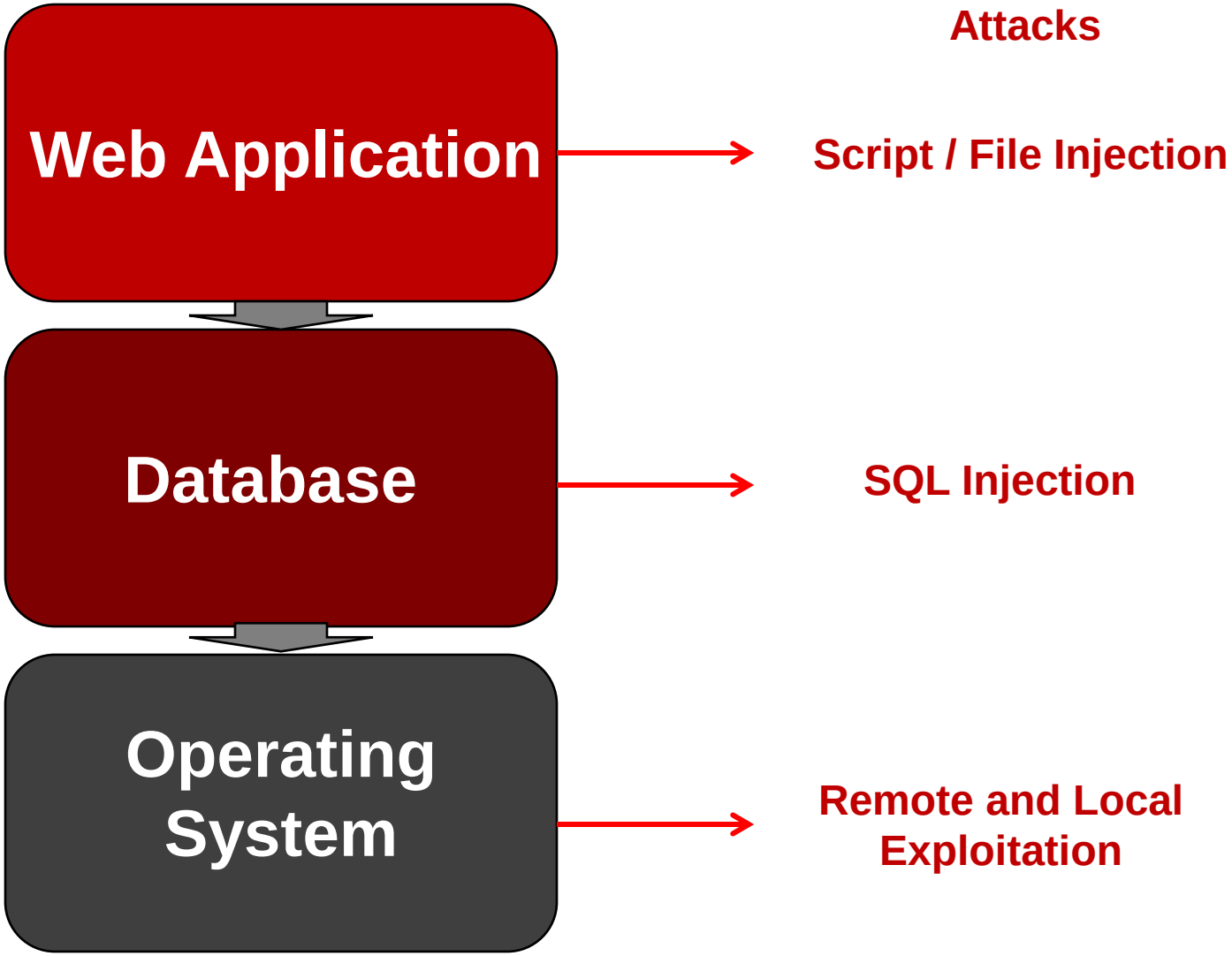
Web App & Data Base Security



Agenda

- OWASP;
- OWASP Top 10 security vulnerabilities;
- Lab1 – Reviewing the exploitation process.

Web App & Security Architecture

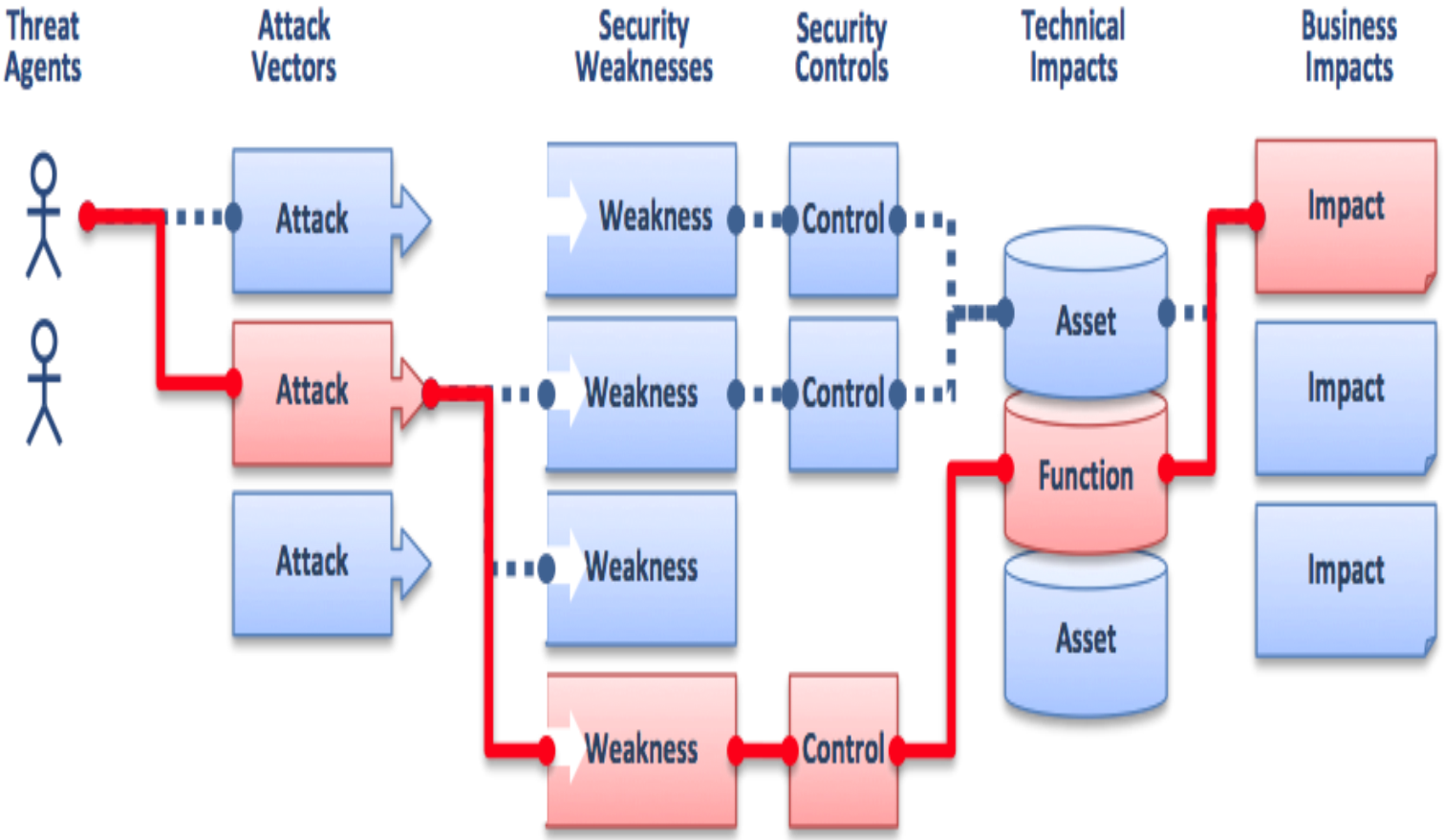


About OWASP (Open Web App Security Project)

- It's an open source application security project;
- This community works to create freely-available articles, methodologies, documentation, tools, and technologies;
- www.owasp.org.

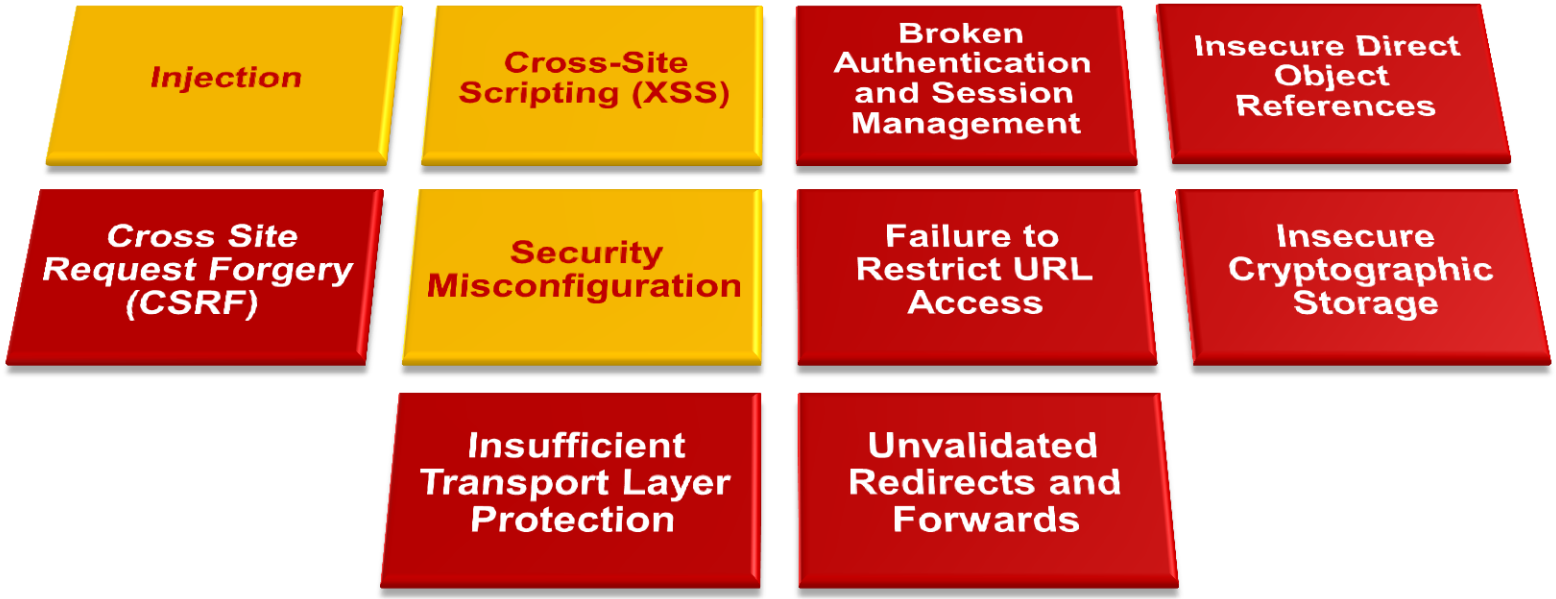
OWASP
The Open Web Application Security Project
http://www.owasp.org

About OWASP (Open Web App Security Project)



https://www.owasp.org/index.php/Top_10_2010-Main

The OWASP Top Ten (2010 Edition)



https://www.owasp.org/index.php/Top_10_2010-Main

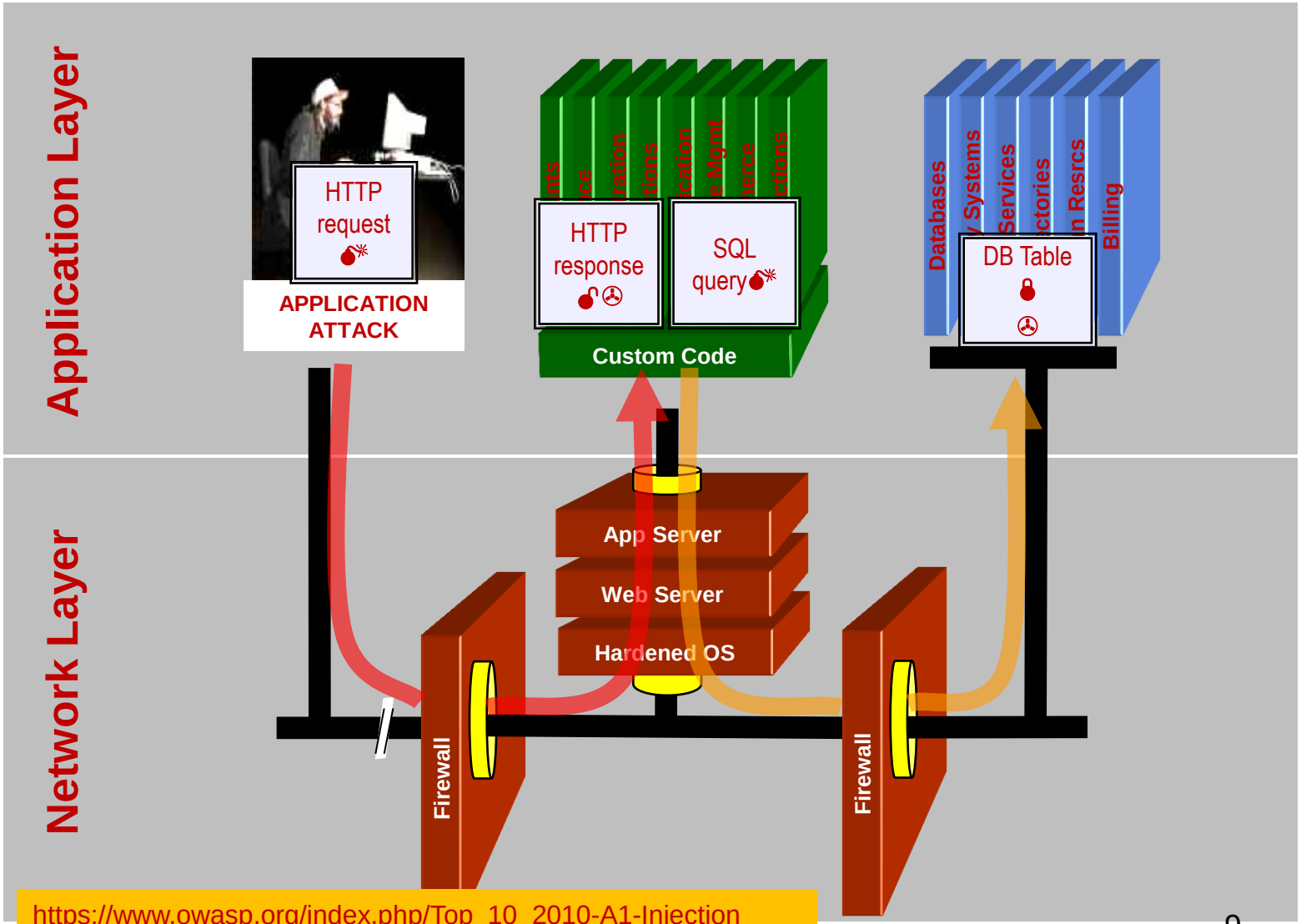
Injection

- Injecting malicious commands into the input string;
- The primary goal is to find a way to run the attacker's code on the clients web server thru poor designed input validation in the application;
- If he can do that, he may be able to read valuable confidential data stored in the clients databases;
- The impact is severe since the entire data base can be read or modified;
- SQL Injection, File injection (WAR File via an application server) and command injection are example of injections;
- WAR file (Web application ARchive) is a JAR file used to distribute a collection of JavaServer Pages, Java Servlets, Java classes, XML files and other resources that together constitute a Web application.

Injection

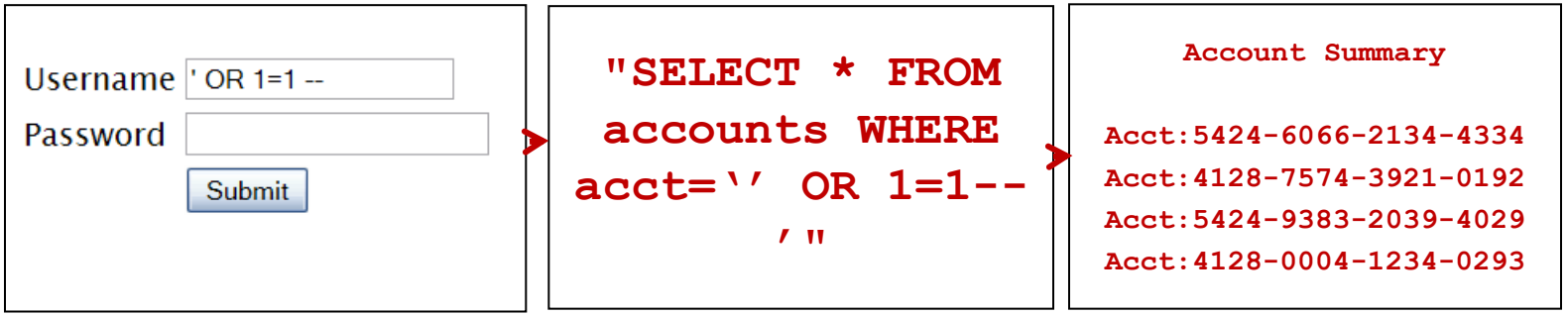
SQL Injection – Illustrated

Injection



https://www.owasp.org/index.php/Top_10_2010-A1-Injection

SQL Injection – Illustrated



Injection

1. Application presents a form to the attacker;
2. Attacker sends an attack in the form data;
3. Application forwards attack to the database in a SQL query;
4. Database runs query containing attack and sends encrypted results back to application;
5. Application decrypts data as normal and sends results to the user.

https://www.owasp.org/index.php/Top_10_2010-A1-Injection

Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS)

A specific type of injection vulnerability in which the attacker injects his own script (such as JavaScript) or HTML into a vulnerable web page.

- Data from attacker is sent to the user's browser;
- Steal user's session, user passwords, steal sensitive data, rewrite web page, redirect user to phishing or malware site;
- Install XSS proxy which allows attacker to observe and direct all user's behavior on vulnerable site.

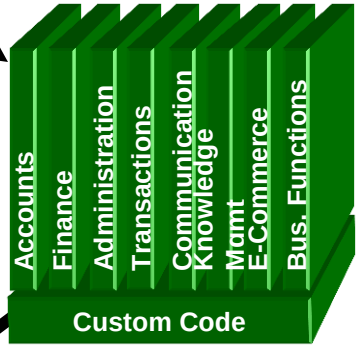
https://www.owasp.org/index.php/Top_10_2010-A2-Cross-Site_Scripting_(XSS)

Cross-Site Scripting Illustrated

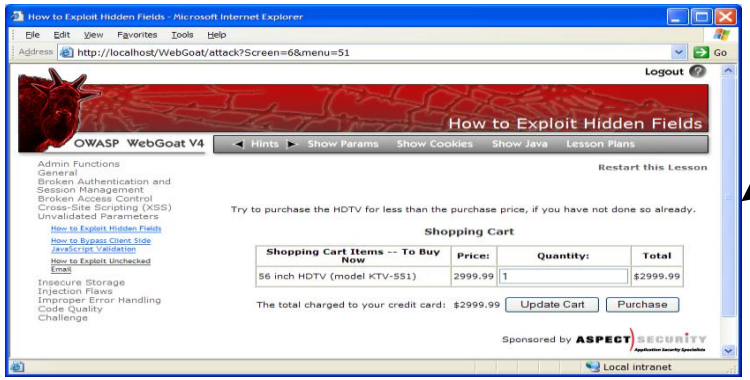
Cross-Site Scripting (XSS)

1 Attacker finds a vulnerable web app

Application with stored XSS vulnerability



2 Victim views page – sees the web page



3 Script silently sends attacker Victim's session cookie

https://www.owasp.org/index.php/Top_10_2010-A2-Cross-Site_Scripting_(XSS)

Broken Authentication and Session Management

- When using a web app, your browser communicates with the application web server by sending and receiving messages using HTTP protocol;
- Since HTTP is a stateless protocol, which means the server doesn't remember who you are between HTTP requests;
- The web apps are forced to implement their own state keeping method;
- Usually, the way they do this is to generate a unique token (session ID) for each user;
- The next request from the same user will have the SID with the request;

Broken Authentication and Session Management

Broken Authentication and Session Management

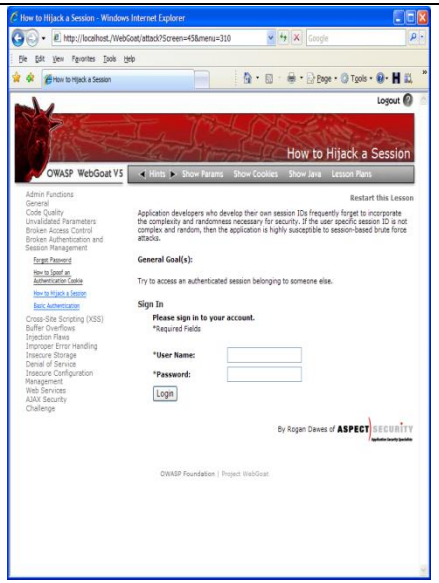
- Change my password, remember my password, forgot my password, secret question, logout, email address, etc;
- User accounts compromised or user sessions hijacked;
- Should use SSL for everything requiring authentication.

Broken Authentication Illustrated

Broken Authentication and Session Management

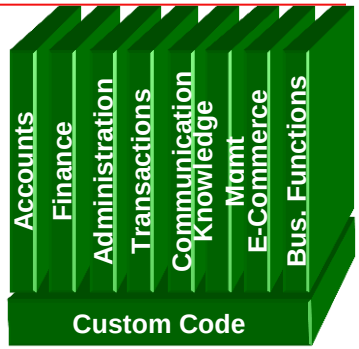
1 User sends credentials

www.sait230.com?JSESSIONID=9FA1DB9EA...



Site uses URL rewriting
(i.e., put session in URL)

2



3 User clicks on a link to http://www.hacker.com in a forum

4 Hacker checks referer logs on www.hacker.com and finds user's JSESSIONID

5 Hacker uses JSESSIONID and takes over victim's account

https://www.owasp.org/index.php/Top_10_2010-A3-Broken_Authentication_and_Session_Management

Insecure Direct Object References

Insecure Direct Object Reference

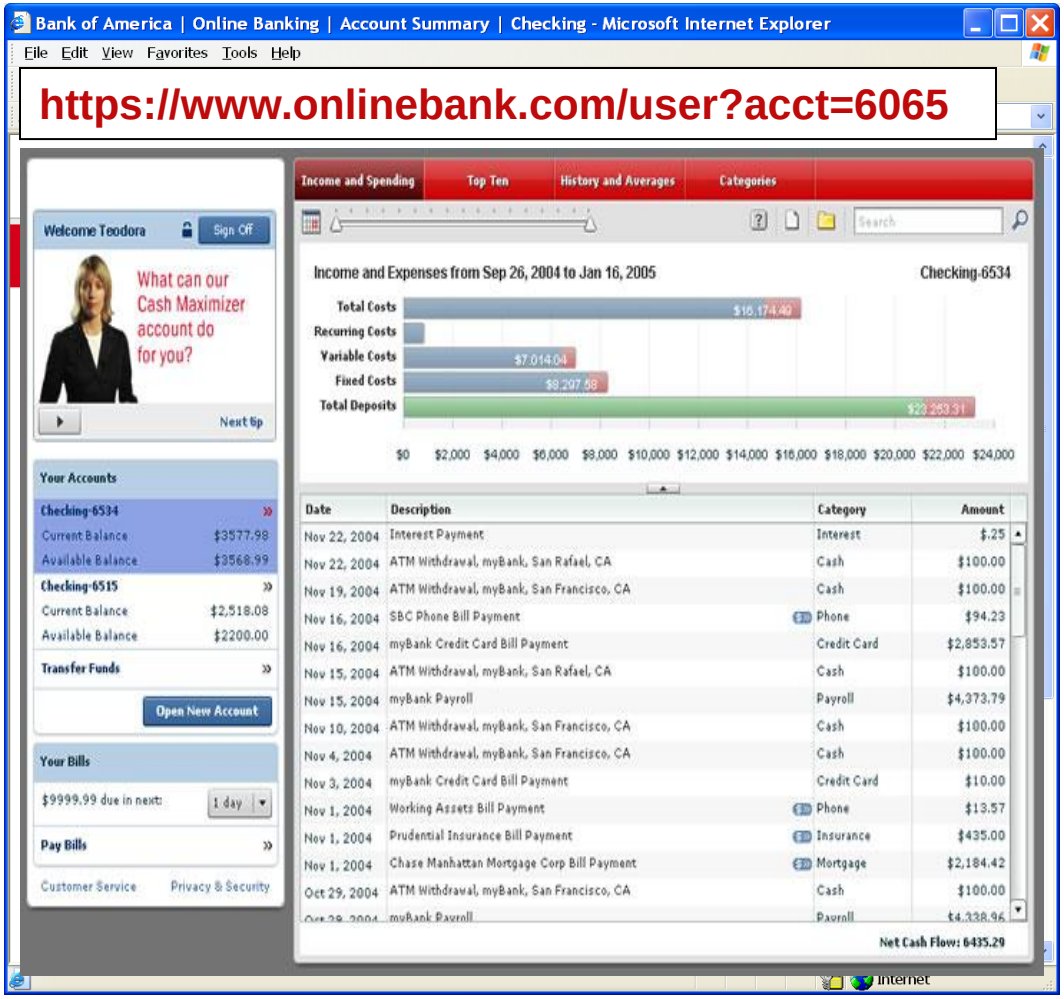
- There is usually no good reason for the web app to reveal any internal resource names such as data file names;
- http://www.sait230.ca/page?datafile=12345.txt;
- http://www.sait230.ca/../../../passwords.txt;
- Only listing the ‘authorized’ objects for the current user;
- Hiding the object references in hidden fields;
- Not enforcing these restrictions on the server side;

Insecure Direct Object References

Insecure Direct Object Reference

- Not enforcing these restrictions on the server side;
- This is called presentation layer access control, and doesn't work;
- Attacker simply tampers with parameter value;
- Users are able to access unauthorized files or data.

Insecure Direct Object References Illustrated



- Attacker notices his acct parameter is 6065

  ?acct=6065

- He modifies it to a nearby number

  ?acct=6066

- Attacker views the victim's account information

https://www.owasp.org/index.php/Top_10_2010-A4-Insecure_Direct_Object_References

Cross Site Request Forgery (CSRF)

Cross-Site Request Forgery

- Attacker creates forged HTTP requests and tricks a victim into submitting them via image tags, XSS, or numerous other techniques. If the user is authenticated, the attack succeeds;
- A web browser will automatically sends any cookies it's holding for a web site back to that web site every time it makes a request there;
- Type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts.

Cross Site Request Forgery (CSRF)

Cross-Site Request Forgery

- The following characteristics are common to CSRF:
  - Involve sites that rely on a user's identity;
  - Exploit the site's trust in that identity;
  - Trick the user's browser into sending HTTP requests to a target site;
  - Involve HTTP requests that have side effects.

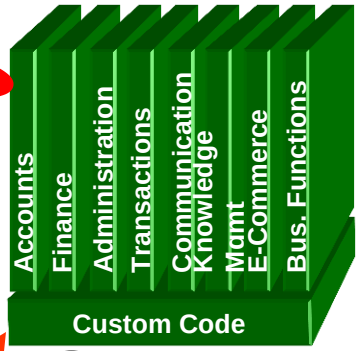
CSRF Illustrated

1

Attacker sets the trap on some website on the internet (or simply via an e-mail)

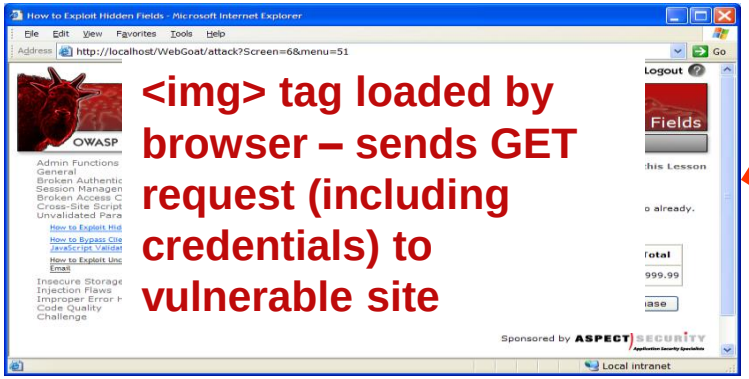
Hidden <img> tag contains attack against vulnerable site

Application with CSRF vulnerability



2

While logged into vulnerable site, victim views attacker's site



<img> tag loaded by browser – sends GET request (including credentials) to vulnerable site

3

Vulnerable site sees legitimate request from victim and performs the action requested

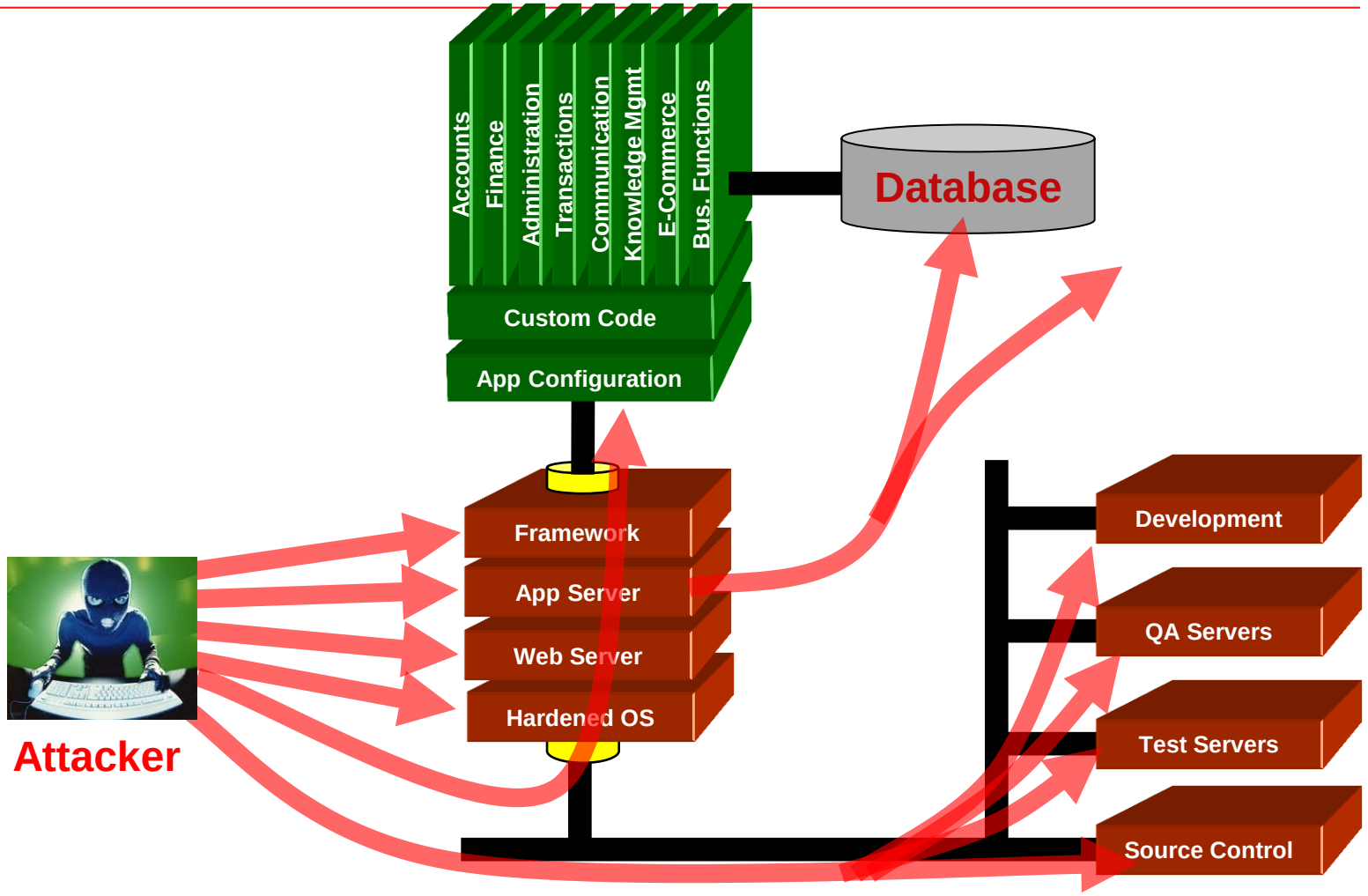
Security Misconfiguration Illustrated

Security Misconfiguration

- Attacker accesses default accounts, unused pages, unpatched flaws, unprotected files and directories, etc. to gain unauthorized access to or knowledge of the system;
- Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, framework, and custom code.
- Developers and network administrators need to work together to ensure that the entire stack is configured properly;
- Automated scanners are useful for detecting missing patches, misconfigurations, use of default accounts, unnecessary services, etc.

Security Misconfiguration Illustrated

Security Misconfiguration



https://www.owasp.org/index.php/Top_10_2010-A6-Security_Misconfiguration

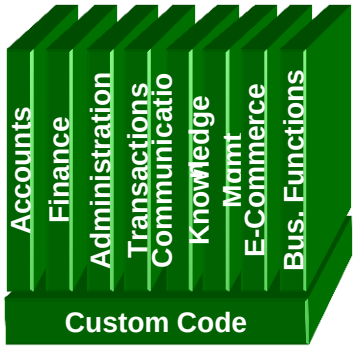
Insecure Cryptographic Storage

Insecure Cryptographic Storage

- Sensitive data like passwords should never be stored unencrypted in plaintext on a server;
- It's better to store a one-way cryptographic hash of a user's password rather than the password itself;
- Attackers access or modify confidential or private information:
  - e.g, credit cards, health care records, financial data (yours or your customers).
- Attackers extract secrets to use in additional attacks.

Insecure Cryptographic Storage Illustrated

1 Victim enters credit card number in form



2 Error handler logs CC details because merchant gateway is unavailable

3 Logs are accessible to all members of IT staff for debugging purposes

4 Malicious insider steals 4 million credit card numbers

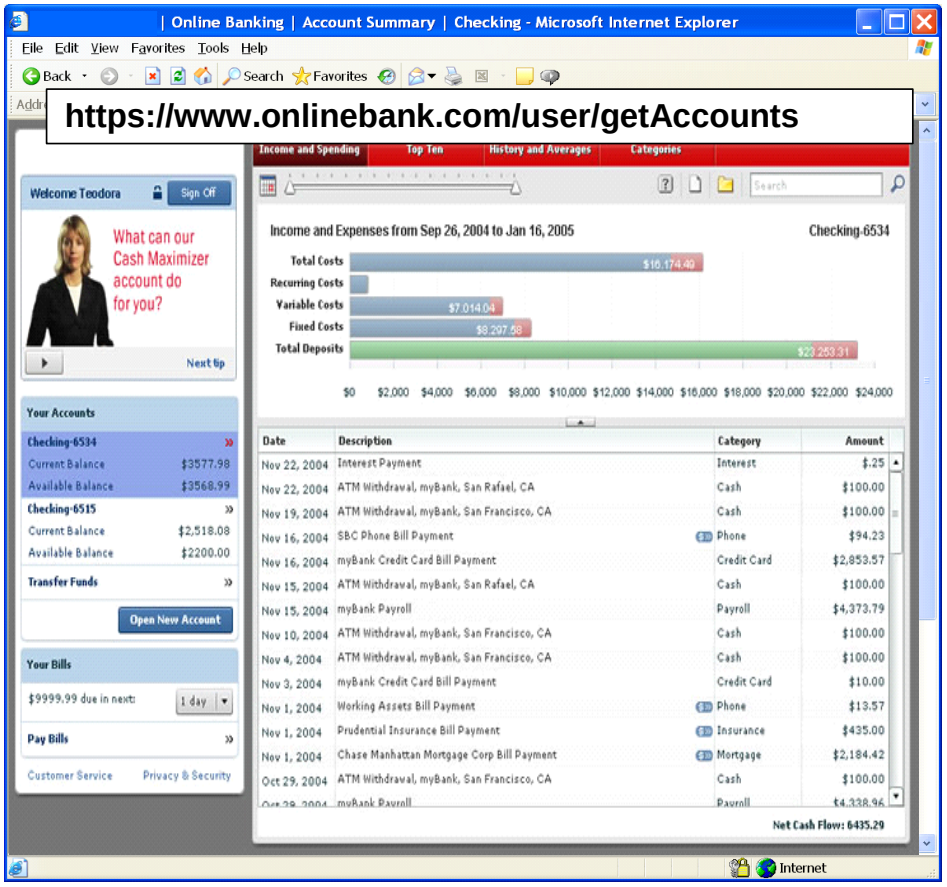
https://www.owasp.org/index.php/Top_10_2010-A7-Insecure_Cryptographic_Storage

Failure to Restrict URL Access

- One way that web app sometimes keep unauthorized users out of certain pages on the site is to hide or display the links to those pages (normal and admin users);
- Attackers invoke functions and services they're not authorized for;
- Access other user's accounts and data;
- Perform privileged actions;

https://www.owasp.org/index.php/Top_10_2010-A8-Failure_to_Restrict_URL_Access

Failure to Restrict URL Access Illustrated



- Attacker notices the URL indicates his role

  /user/getAccounts

- He modifies it to another directory (role)

  /admin/getAccounts, or

  /manager/getAccounts

- Attacker views more accounts than just their own

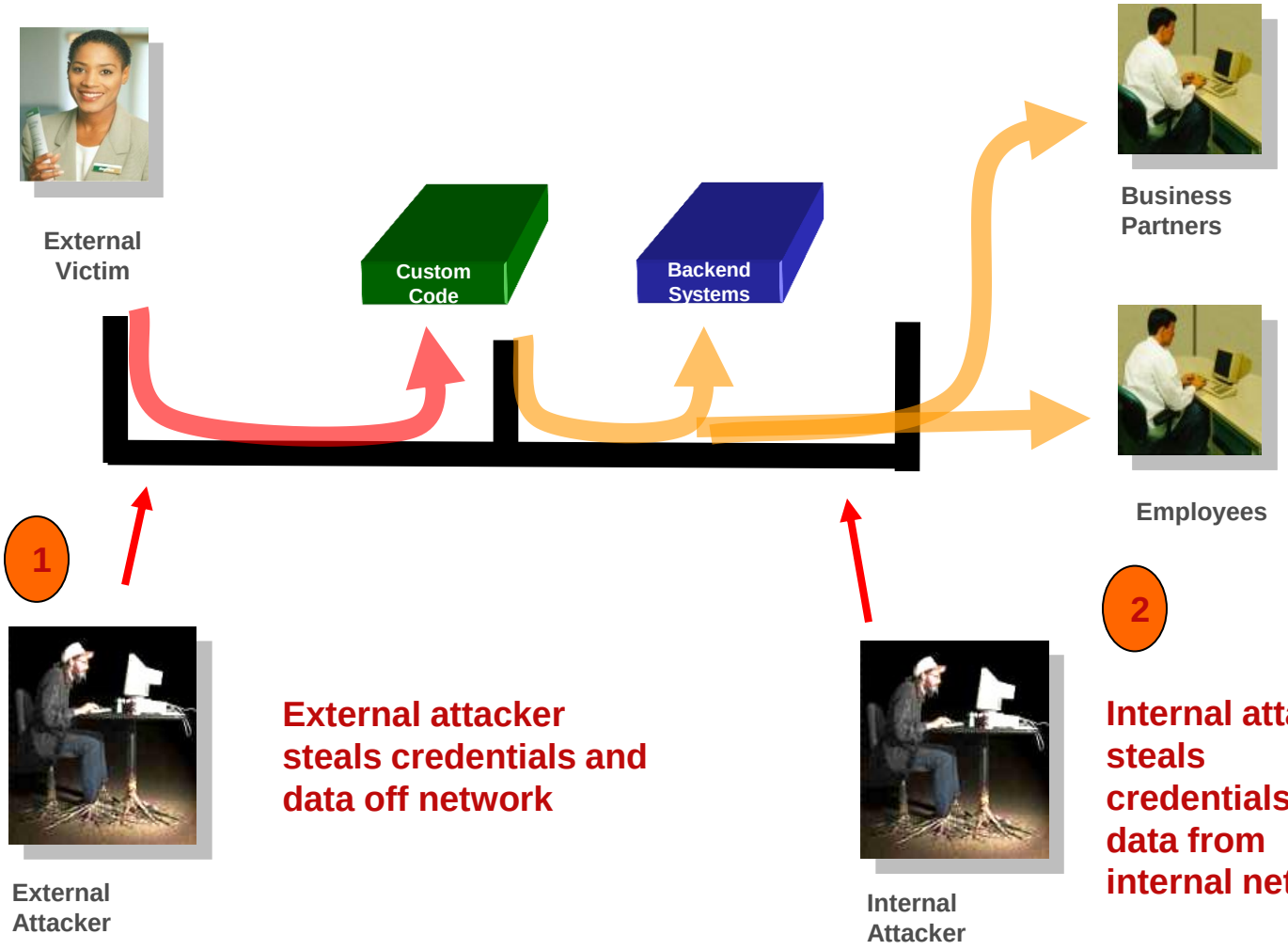
https://www.owasp.org/index.php/Top_10_2010-A8-Failure_to_Restrict_URL_Access

Insufficient Transport Layer Protection

- Because HTTPs is slower than HTTP, many web apps don't use HTTPs as they should;
- Failure to identify all sensitive data;
- Failure to identify all the places that this sensitive data is sent:
  - On the web, to backend databases, to business partners, internal communications
- Failure to properly protect this data in every location.

https://www.owasp.org/index.php/Top_10_2010-A9-Insufficient_Transport_Layer_Protection

Insufficient Transport Layer Protection Illustrated



Unvalidated Redirects and Forwards

Unvalidated Redirects and Forwards

- Applications frequently redirect users to other pages, or use internal forwards in a similar manner. Sometimes the target page is specified in an unvalidated parameter, allowing attackers to choose the destination page;
- Attacker links to unvalidated redirect and tricks victims into clicking it. Victims are more likely to click on it, since the link is to a valid site. Attacker targets unsafe forward to bypass security checks;

https://www.owasp.org/index.php/Top_10_2010-A10-Unvalidated_Redirects_and_Forwards

Unvalidated Redirects and Forwards

Unvalidated Redirects and Forwards

- They internally send the request to a new page in the same application;
- Sometimes parameters define the target page;
- If not validated, attacker may be able to use unvalidated forward to bypass authentication or authorization checks.

https://www.owasp.org/index.php/Top_10_2010-A10-Unvalidated_Redirects_and_Forwards

FURTHER YOUR PASSION

Questions