Security day 1

What is web security, OWASP, Input validation

What is security?

"Preservation of confidentiality, integrity and availability of information” (CIA).

Confidentiality

Confidentiality is the ability to hide information from those people unauthorised to view it.

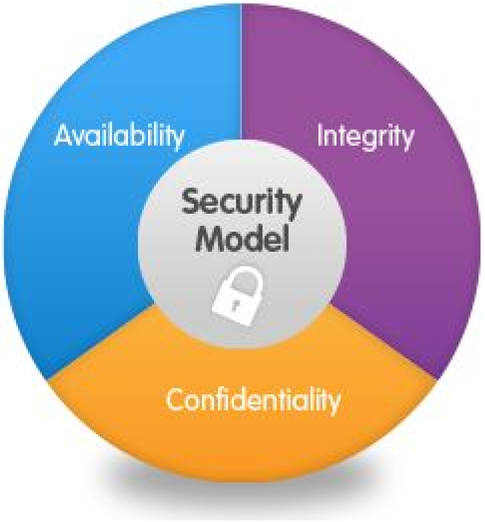
Integrity

The ability to ensure that data is an accurate and unchanged representation of the original secure information.

Availability

Ensuring that the information concerned is readily accessible to the authorised viewer at all times.

CIA security model



Why do we focus on web security?

- Accounts for a large part of all vulnerabilities
- Protection our user
- Protecting our business
- Every “thing” has an webserver (http://www.zdnet.com/article/this-is-the-dishwasher-with-an-unsecured-web-server-we-deserve/)

What do we cover

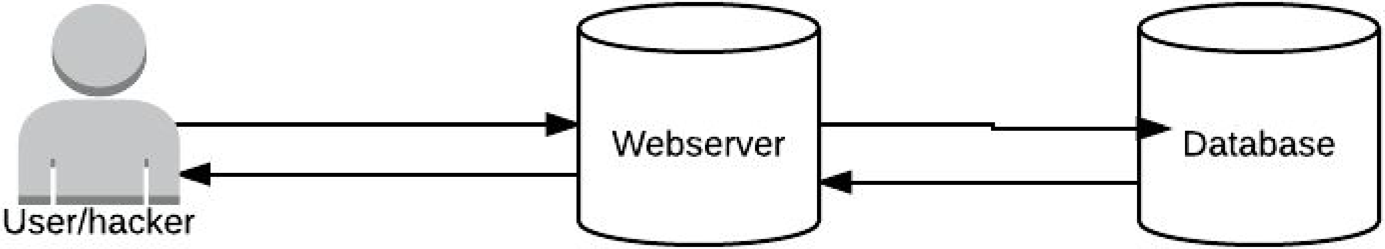
We cover:

- The security of the web application, regardless of infrastructure

We do not cover:

- Network security
- Server security
- Hardware security
- etc

An attack on a web app



Defences against attacks

- Firewalls
- WAF
- Proxys
- Network intrusion detection
- Host intrusion detection
- Containers

Defences against attacks

- Add a firewall in front of web server, block ports,blocks ip
- Add Waf in front webserver, filter, and deny patterns
- Use Network intrusion detection eg SNORT to detect and block attacks
- Use a proxy / load balancer in front of mysql to block sql injections.
- Host intrusion detection on the specific host machine to detect files that have been changed(difficult to manage)
- Containers to block syscals and limit impact (dirty cow)

Usecases

Layered defense

Old untrusted app

IOT devices that are difficult to update

Mitigation of occurring attack

Vocabulary

MiT

Dos / DDOS

SQL injection

Exploit

Xss

Hashing/Encryption

DefCon

OWASP

Buffer overflow

0-day

regex

What is Owasp?

The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on:

- Improving the security of application software.
- Make application security "visible," so that people and organizations can make informed decisions about application security risks.

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)

STRIDE/Dread

STRIDE

- A classification system(groping /characterizing)

Dread

- A system to quantifying, comparing and prioritizing (score)

STRIDE and Dread are not in use any more!

CWE

Targeted to developers and security practitioners, the Common Weakness Enumeration (CWE) is a formal list of software weakness types.

Sql injection:

https://cwe.mitre.org/data/definitions/89.html

CWSS

The Common Weakness Scoring System (CWSS) provides a mechanism for prioritizing software weaknesses in a consistent, flexible, open manner. It is a collaborative, community-based effort that is addressing the needs of its stakeholders across government, academia, and industry.

https://cwe.mitre.org/scoring/index.html

CVSS

ng system as CWSS.

mplex.

used.

vd.nist.gov/vuln-metrics/cvss/v3-calculator

Common Vulnerabilities and Exposures (CVE)

- One name for one vulnerability or exposure
- One standardized description for each vulnerability or exposure
- A dictionary rather than a database

Eg:
https://www.cvedetails.com/vulnerability-list/vendor_id-1367/product_id-2387/Drupal-Drupal.html

Validate, Sanitize and Escape

Validation:

- Validation makes sure that you have the right kind of data.

Sanitization:

- Removes any harmful data.

Escaping:

- Take any harmful data and makes it harmless.

https://www.wordfence.com/learn/how-to-write-secure-php-code/

Whitelisting / Blacklisting

About

- Blacklisting allows all except denied (border control)
- Whitelisting allows non exempt approved (Apple store)

In most cases it is more effective to whitelist than to blacklist

https://www.schneier.com/blog/archives/2011/01/whitelisting_vs.html

Filter

filter_var

- Filters a variable with a specified filter (filter and sanitize)

filter_input

- Gets a specific external variable(POST/GET etc) by name and optionally filters it (filter and sanitize)

Eksample:

filter_var(‘127.0.0.1’), FILTER_VALIDATE_IP);

Regular expression (RegEx)

Pattern matching based on expressions

preg_match

- Perform a regular expression match

Preg_replace

-Perform a regular expression search and replace

Regex cheatsheats

/^ start of line or

$/ end of line

[a-z] match from a-z or 0-9

- + Match additional eg +åæø@

{5,10} Length from 5 to 10

http://www.php.net/manual/en/regexp.reference.meta.php

Regex example

```
preg_match('/^[a-zA-Z0-9+åøæØÅÆ]{5,10}$/', $data)
```

- Match from start of line /^
- The following [a-zA-Z0-9]
- With a length of 5-10 {5,10}
- To end of line $/

Task 1, CVSS Score

Try calculating a score at(8+):

https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

With the following:

Confidentiality Impact: Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact: Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact: Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity: Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)

Authentication: Single system (The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).)

Gained Access :   None

Tasks 2:

You have a form that accepts different input from a user. Create the functions to validate the input sent from the form:

The form sends Name. email, street, postNr, socialsecurityNumber(cpr) and password.

See:

https://github.com/renegulager/sec101/blob/master/task.php

Score

https://www.cvedetails.com/cve/CVE-2016-3168/

Conclusion

Validation of input is a security fundamental

Regex can be hard and complex

(Is !def!x+yz%./a-b_c@example.ninja valid ?)

shopname+mail@my

Use php build in functions

Use whitelisting if possible

Use blacklisting to block specific attacks