

Dr. Mohammed Anbar

Network Security CYB 630
Web Security (part two)

Lecture 9



Learning objectives

- ~~Explain basic web terminology,~~
- ~~Understand HTTP and HTTPS protocols,~~
- Understand web security and web security threats,
- Explain technologies for Web Security,
- Understand Open Web Application Security Project (OWASP),
- Explain attacks on web applications.

Web security

- Web security refers to the safeguards and procedures that businesses use to guard against risks and cyber criminals who use the web channel.
- Web security is essential for ensuring business continuity and reducing risk to users, users' data, and businesses.
- Web security threats are vulnerabilities within websites and applications, or attacks launched by malicious actors.

Web security (cont.)

- Web security threats are designed to breach an organization's security defenses, enabling hackers and cyber criminals to control systems, access data, and steal valuable resources.
- Web security threats involve malicious people and organizations, as well as the tools they use to leverage the internet to infiltrate your network or devices.
- Common web security threats include malware, ransomware, cross-site scripting (XSS), SQL injection, phishing, denial of service, and many others.

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Modification of memory • Modification of message traffic in transit	• Loss of Information • Compromise of machine • Vulnerability to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the network • Theft of data from the client • Info about network configuration • Info about which client talks to the server	• Loss of information • Loss of privacy	Encryption, web proxies

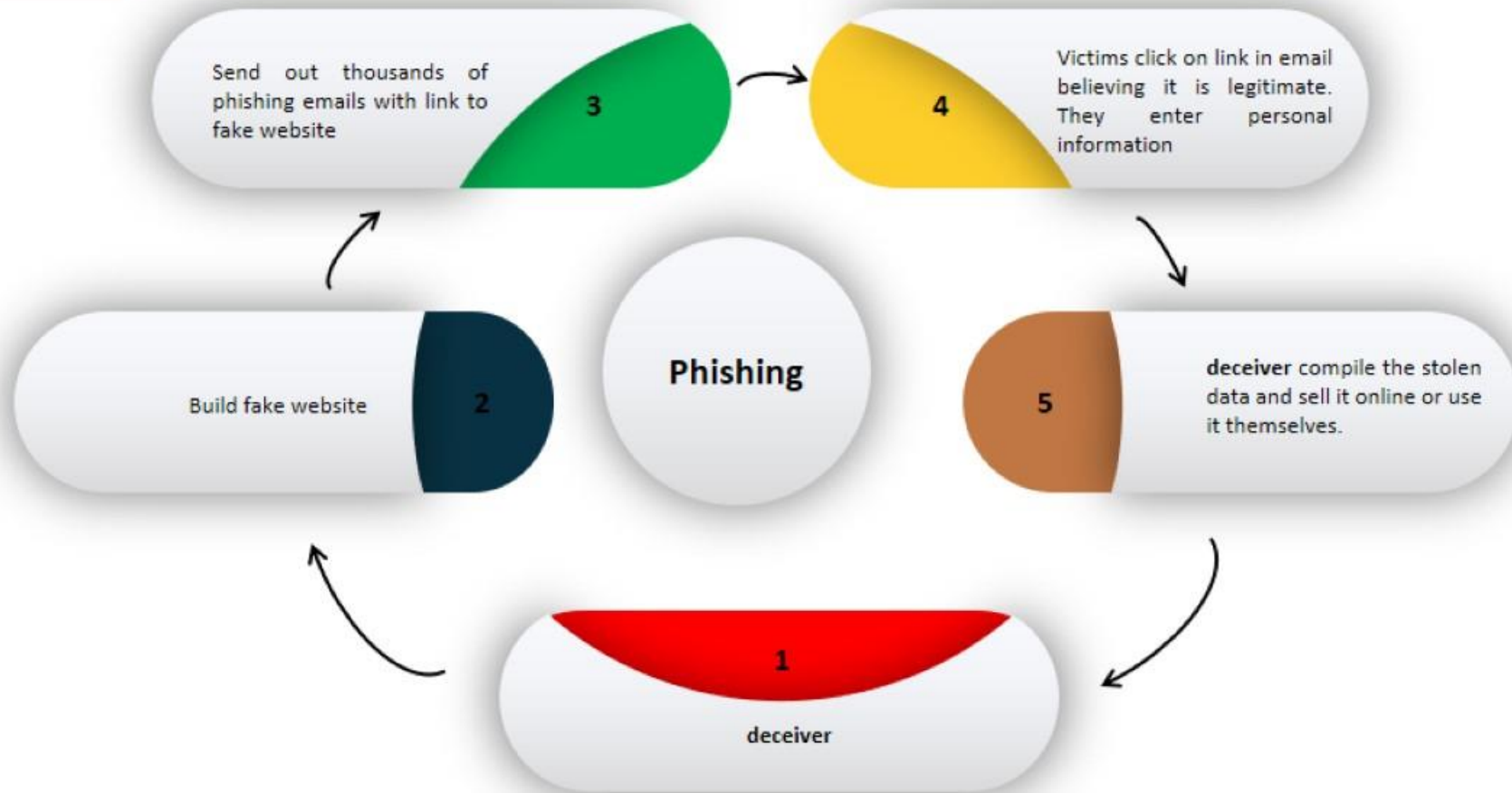
Source: <http://elearningatria.files.wordpress.com/2013/10/ise-viii-information-and-network-security-06is835-notes.pdf>

	Threats	Consequences	Countermeasures
Denial of Service	• Flooding machine with fake requests • Filling up disk or memory • Isolating machine by DNS attacks	Prevent users from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques

Phishing Website - an overview

- Phishing is a type of social engineering in which an attacker sends a fake message/email intended to mislead a person into giving them personal information or to install harmful software, such as ransomware, on the victim's machine.
- Phishing is typically carried out by e-mail spoofing or instant messaging.

Phishing Website (cont.)



Fake web page for Phishing

- A webpage that looks exactly as the original one
- To generate the fake webpage, the attacker usually uses a third-party tool to design the fake page or save the original webpage as an HTML file and view the source code
- An attacker write server-side (PHP, ASP, etc.) script to handle the user request and save the sensitive data such as credit card details or credentials to a file In the HTML
- Send a spoofed email with a link to a spoofed webpage

Fake web page for Phishing (cont.)



➤ **Web Application Firewalls (WAFs)**

- Protect web applications by monitoring and filtering internet traffic that flows between an application and the internet. In this way, a WAF works as a secure web gateway (SWG).
- It provides protection for web applications against attacks, including cross-site scripting, file inclusion, cross-site forgery, Structured Query Language (SQL) injection, and other threats.
- Even though it works against many internet threats, it is not intended to defend against all kinds of threats.

- WAF protects the application layer and is specifically designed to analyze each HTTP/S request at the application layer.
- you can think of a WAF as the intermediate between the user and the app itself, analyzing all communications before they reach the app or the user.

- Traditional WAFs ensure that only allowed actions (based on security policy) can be performed.
- For many organizations, WAFs are a trusted, first line of defense for applications, especially to protect against the OWASP Top 10—the foundational list of the most seen application vulnerabilities.

➤ **Security or Vulnerability Scanners**

- Tools that organizations use to automatically examine their systems, networks, and applications to check for weaknesses in their security.
- Once a vulnerability scanner has finished checking the target system, security teams can use the results to address critical vulnerabilities.

➤ **Password-cracking Tools**

- Can still gain access to your system even if you have lost or forgotten your password. This helps maintain web security for businesses in a couple of different ways.
- First, if you need to reset your password but cannot remember the original one, a password-cracking tool allows you to gain access. Second, if someone has penetrated your system and changed the password, you can use a password-cracking tool to get back in and change the password to something harder.

➤ **White Box Testing Tools**

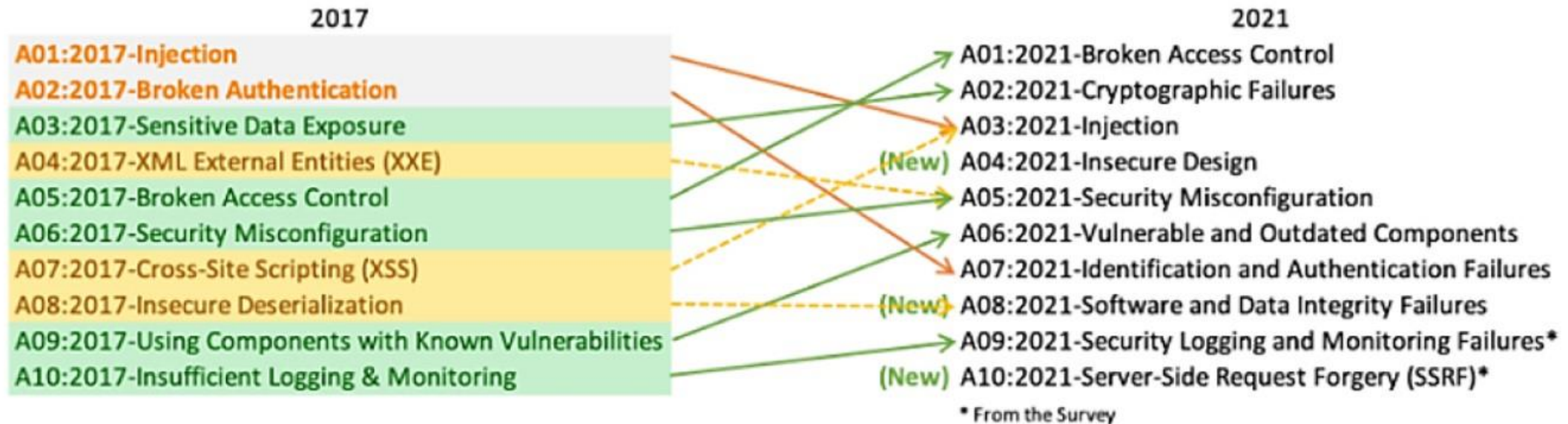
- Test the design, coding, and internal structure of the software to enhance its design, as well as ensure the smooth flow of data into and out of the application.
- During white box testing, you can see the code, so it is sometimes also called clear box testing or transparent box testing

➤ **Black Box Testing Tools**

- Refers to checking a system without any knowledge regarding how it works. The only thing the tester sees is the input they key in and the resulting output.
- Black box testing tools are used to see how the system responds to unexpected actions taken by users. They can help security personnel inspect response times and detect issues in software performance and whether the system is reliable.

Open Web Application Security Project (OWASP)

- OWASP is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.
- OWASP provides free and open resources. It is led by a non-profit called The OWASP Foundation.
- The OWASP Top 10 - 2021 is the published result of recent research based on comprehensive data compiled from over 40 partner organizations.



Source: <https://owasp.org/Top10/>

Broken Access Control	Vulnerable and Outdated Components
Cryptographic Failures	Identification and Authentication Failures
Injection	Software and Data Integrity Failures
Insecure Design	Security Logging and Monitoring Failures
Security Misconfiguration	Server-Side Request Forgery

SQL injection



Cross-site
Scripting
(XSS)



Access control enforces a policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits. Common access control vulnerabilities include:

- Violation of the principle of least privilege or denial by default, where access should only be granted for capabilities, roles, or users, but is available to anyone.
- Bypassing access control checks by modifying the URL (parameter tampering or force browsing), internal application state, or the HTML page, or by using an attack tool modifying API requests.

- Many web applications are connected to a database. The database holds all the information the web application wishes to store and use.
- SQL Injection is a technique that allows attackers to manipulate the SQL ("Structured Query Language") of the web application is using.
- This typically happens because of a lack of data sanitization. SQL is used regularly by developers to access database resources.

An application is vulnerable to attack when:

- ✓ User-supplied data is not validated, filtered, or sanitized by the application.
- ✓ Dynamic queries or non-parameterized calls without context-aware escaping are used directly in the interpreter.

Log In

" or ""="

" or ""="

☐ Remember Me!

Submit

Don't have an account? [Register Here](#)

- When the input the value is " or ""= "
- Because the database interprets the query as always true, the resulting SQL Query returns all entries of the table. The code at the server will create a valid SQL statement like this:

```
select * from userAccountsTable where username=" OR ""= " and password= " or ""= "
```

- The pseudo-code example that contains a SQL Injection vulnerability is presented in the next slide.

Chapter

08

SQL Injection (cont.)

```
$username = getUsername();
$password = getPassword();
$user = mysql_query("SELECT * FROM userAccountsTable WHERE
username = $username AND password = $pw");
if ($user) {
    $loggedIn = True;
} else {
    $loggedIn = False;
}
```

select * from userAccountsTable where username=" OR "=" and password=" or "="

userAccountsTable

Log In

" or ""="

" or ""="

☐ Remember Me!

Submit

Don't have an account? [Register Here](#)

Username	Password
admin	admin
sys	password

- `/login?username=admin&password=password' OR '1'='1`
- The password variable is set to contain the SQL characters, causing the resulting SQL string to return a row, even if the password is unknown to us. The resulting SQL query would be:

```
select * from userAccountsTable where username='admin' and password='password ' OR 1=1
```

```
select * from userAccountsTable where  
username='admin' and password='admin'
```

username	password
admin	admin

Log In

☐ Remember Me!

Don't have an account? [Register Here](#)

```
$username = getUsername();  
$pw = getPassword();  
$parameterizedQuery = prepare_query("SELECT * FROM userTable where username = ? and  
password = ?");  
$parameterizedQuery.setString(1, $username)  
$parameterizedQuery.setString(2, $password)  
$user = parameterizedQuery.execute();  
if ($user) {  
    $loggedIn = True;  
} else {  
    $loggedIn = False;  
}
```

Cross-Site Scripting (XSS)

- XSS uses the server to attack visitors of the server. The attack does not target the server itself, but instead the users.
- The server is simply used to reflect attackers' values, typically JavaScript, against visitors who then run the attacker's data in their own browser.
- The attacker must craft an input that the server does not clean and sanitize, that way when a visitor clicks a link containing the attacker's values or visits a resource on the webpage that the attacker has used in their attack, the user runs code that the attacker-supplied.

- Another kind of XSS is called a Stored XSS attack. In Stored XSS attacks, the attacker can save content on the webpage which is reflected every time someone visits the website.
- It does not require someone to click a link necessarily.

- SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL.
- It allows an attacker to force the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).
- As modern web applications provide end-users with convenient features, fetching a URL becomes a common scenario.
- As a result, the incidence of SSRF is increasing. Also, the severity of SSRF is becoming higher due to cloud services and the complexity of architectures.

OWASP Broken Web Applications Project

- OWASP Broken Web Applications Project (OWASPBWA) provides a collection of purposefully vulnerable applications to safely practice penetration testing.
- OWASPBWA can be downloaded from the link below

<https://sourceforge.net/projects/owaspbwa/files/>

Required Software:

- Virtual box: <https://www.virtualbox.org/wiki/Downloads>
- OWASPBWA: <https://sourceforge.net/projects/owaspbwa/files/>

Stallings, W. (2006). Cryptography and network security principles and practices 4th edition

<https://owasp.org/Top10/>



Thank You