

# Java and JavaScript



Web app and web page development

Java and JavaScript in Web Applications

## JavaScript



Runs in a client-side web browser environment (embedded within HTML documents)

Provides functionality beyond simple HTML

Case-sensitive

01:26



Previous Topic

Next Topic

04:55



## JavaScript Code Example

```
<html><head>
<script type="text/javascript">
function showmsg()
{
    alert ('Hello world');
}
</script>
</head>
<body>
<input type="button" value="OK"
onclick="showmsg()" />
</body></html>
```

01:34



◀ Previous Topic

Next Topic ▶

04:48



## Node.js

- **It is JavaScript, but not limited to running in a web browser embedded within HTML**
- **Normally used server-side**
- **Normally uses a .js file extension**
- **How is it used?**
  - Dynamic web page generation
  - Server-side file management
  - Database reading and writing

<NODE>  
JS

## Node.js Code Example

```
var http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type':
    'text/html'});
  res.end('Hello World!');
}).listen(8081);
```

Java and JavaScript in Web Applications

## Java



Not a scripting language like JavaScript, but instead a compiled language

Cross-platform but requires a Java Runtime Environment (JRE)

Java applets can be embedded within HTML, Java programs can run client-side and server-side

05:13



Previous Topic

Next Topic

01:09



## Java Code Example

```
import java.util.Scanner;
public class HelloWorld {
    public static void main(String[] args) {
        // Read user input from keyboard
        Scanner reader = new
Scanner(System.in);
        System.out.print("Enter your age in
years: ");
        int userage = reader.nextInt();
        System.out.println("Your age in years
is: " + userage);
    }
}
```

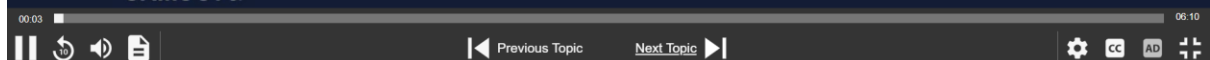


Cross-site Scripting Attacks


## Cross-site Scripting Attacks

Dan Lachance

skillsoft



## Cross-site Scripting (XSS) Attacks



Attacker injects malicious code into vulnerable web site  
JavaScript is commonly used  
Web form, URL, API call  
Results from improper input handling  
All user input must be untrusted

Cross-site Scripting Attacks

## Stored XSS Attack



1. Attacker injects malicious code in web site



2. Web site visitor executes malicious code in their web browser

01:58



Previous Topic

Next Topic

04:15



## XSS Victim



- Web site visitors unknowingly execute malicious code (stored/persistent XSS attack)
- Attackers could then
  - Steal web browser session cookies
  - Impersonation
  - Log keystrokes
  - Turn on webcam

## Injecting JavaScript into a Web Page Form

### Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="Use One"/>
Message *	<input type="text" value="&lt;script&gt;alert('Hello world');&lt;/script&gt;"/>
<input type="button" value="Sign Guestbook"/>	

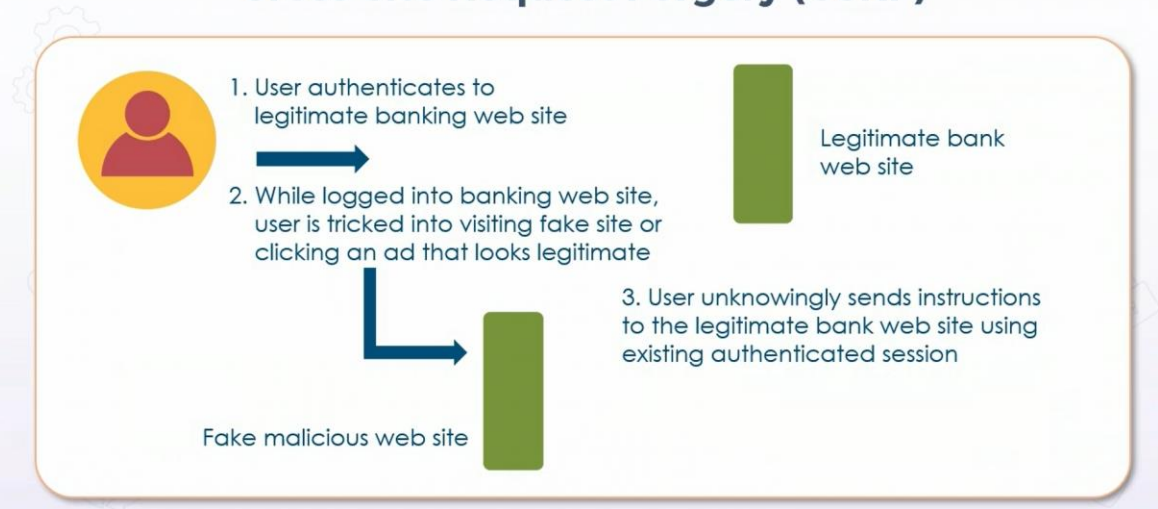
Name: test  
Message: This is a test comment.

Name: Dan  
Message:

#### More info

<http://hacker.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

## Cross-site Request Forgery (CSRF)



05:04



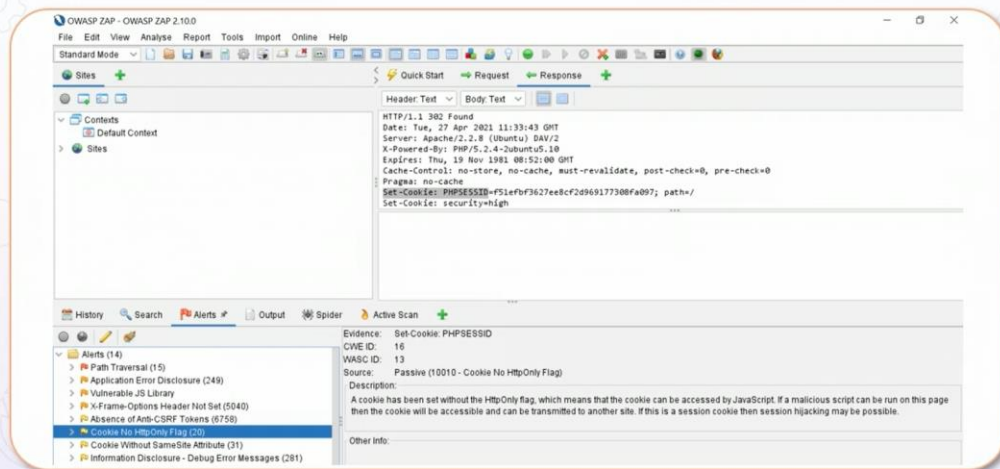
Previous Topic

Next Topic



01:09

# Insecure Cookies



05:18 00:55

Previous Topic Next Topic

Executing XSS through Web Page Forms

Not secure | 192.168.4.124

metasploitable2

Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Dan Lachance

skillsoft

00:05 06:01

Previous Topic Next Topic



Executing XSS through Web Page Forms

← → ↻ ⚠ Not secure | 192.168.4.124/dvwa/vulnerabilities/xss\_s/ ☆ ⓘ ⋮

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected


XSS stored

DVWA Security

PHP Info

About

Logout



## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

cblackwell

Message \*

<script>alert('Hello you');</script>

Sign Guestbook

Name: test

Message: This is a test comment.

### More info

<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

01:40 04:25

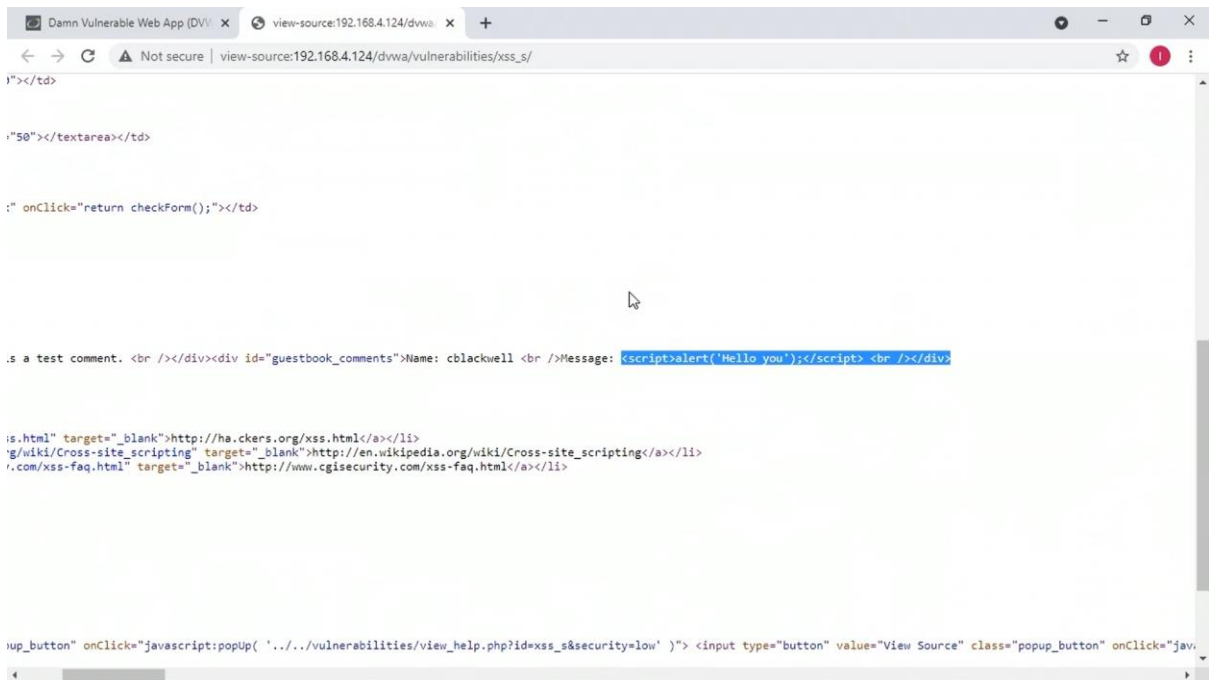
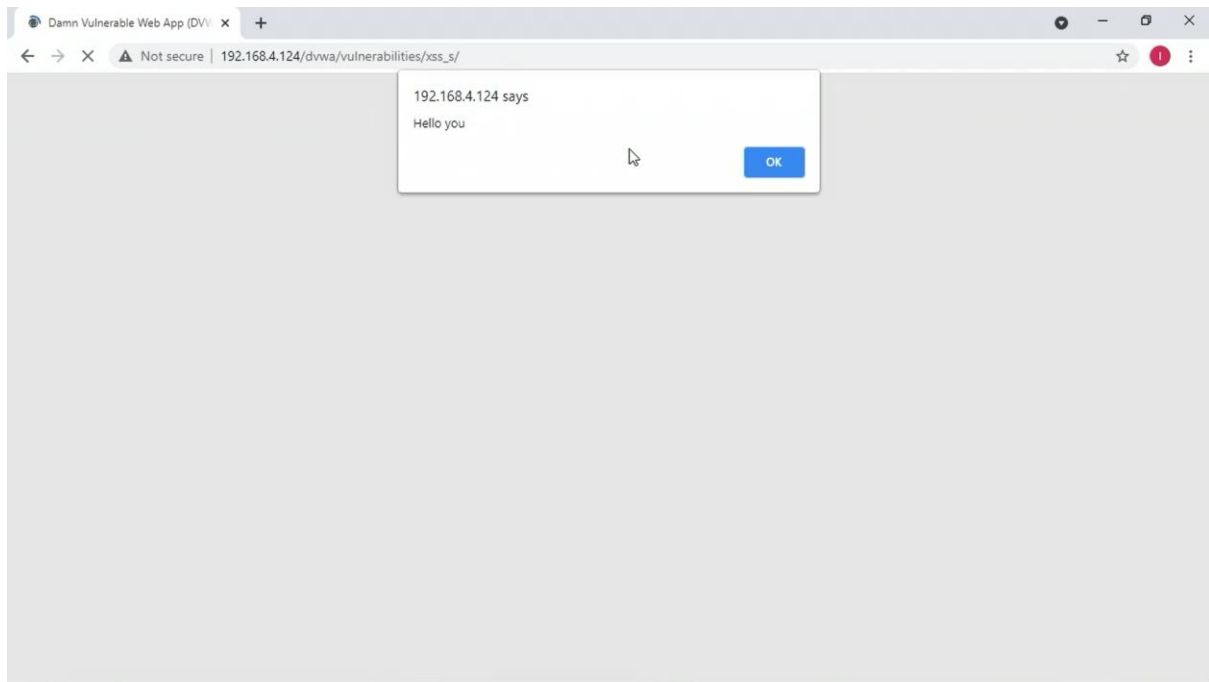
⏮ ⏪ ⏩ ⏭ ⚙ CC AD ⏻

Damn Vulnerable Web App (DVWA) x +

← → × ⚠ Not secure | 192.168.4.124/dvwa/vulnerabilities/xss\_s/ ☆ ⓘ ⋮

192.168.4.124 says  
Hello you

OK



```
192.168.4.124 - PuTTY
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities$ ls
brute  exec  sqli      upload      view_source_all.php  xss_r
csrf  fi    sqli_blind  view_help.php  view_source.php      xss_s
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities$
```

```
192.168.4.124 - PuTTY
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities/xss_s/source$ ls
high.php  low.php  medium.php
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities/xss_s/source$
```

```
192.168.4.124 - PuTTY
GNU nano 2.0.7 File: low.php

<?php

if(isset($_POST['btnSign']))
{

    $message = trim($_POST['mtxMessage']);
    $name     = trim($_POST['txtName']);

    // Sanitize message input
    $message = stripslashes($message);
    $message = mysql_real_escape_string($message);

    // Sanitize name input
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";

    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>' );

}

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

```
192.168.4.124 - PuTTY
GNU nano 2.0.7 File: high.php

<?php

if(isset($_POST['btnSign']))
{

    $message = trim($_POST['mtxMessage']);
    $name     = trim($_POST['txtName']);

    // Sanitize message input
    $message = stripslashes($message);
    $message = mysql_real_escape_string($message);
    $message = htmlspecialchars($message);

    // Sanitize name input
    $name = stripslashes($name);
    $name = mysql_real_escape_string($name);
    $name = htmlspecialchars($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";

}

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Damn Vulnerable Web App (DVWA) x +

Not secure | 192.168.4.124/dvwa/security.php

[Home](#)[Instructions](#)[Setup](#)  
[Brute Force](#)[Command Execution](#)[CSRF](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)[XSS stored](#)  
**[DVWA Security](#)**[PHP Info](#)[About](#)  
[Logout](#)

## DVWA Security

### Script Security

Security Level is currently low.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

high

### PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently disabled. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Username: admin  
Security Level: low

Damn Vulnerable Web App (DVWA) x +

Not secure | 192.168.4.124/dvwa/vulnerabilities/xss\_s/

[Home](#)[Instructions](#)[Setup](#)  
[Brute Force](#)[Command Execution](#)[CSRF](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)**[XSS stored](#)**  
[DVWA Security](#)[PHP Info](#)[About](#)  
[Logout](#)

## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Name: test  
Message: This is a test comment.

Name: cblackwell  
Message: <script>alert('Hello you');</script>

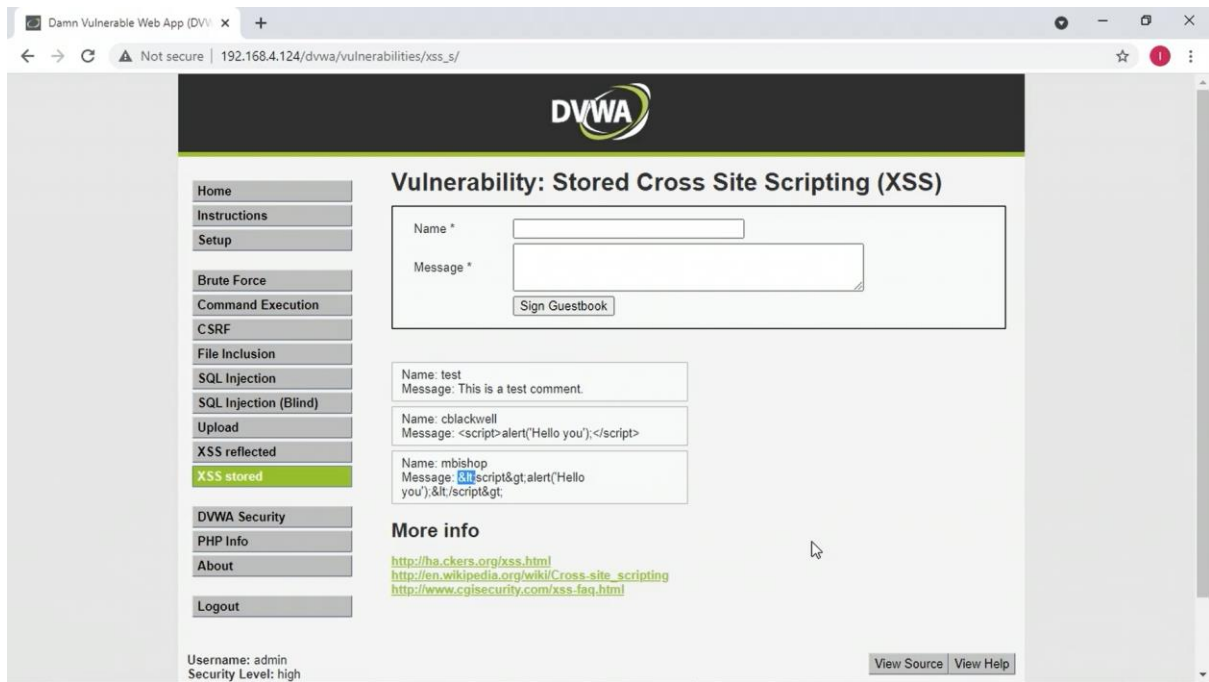
Name: mbishop  
Message: &lt;script>&gt;alert('Hello you');&lt;/script&gt;

### More info

<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

[View Source](#) [View Help](#)

Username: admin  
Security Level: high



# Mitigating XSS Attacks

Dan Lachance

skillsoft

## Mitigating XSS Attacks



Periodic web app vulnerability scans

## Web Application Firewall (WAF)



Designed to look for web application attacks



Can prevent and report on potential web application XSS activity

01:20



Previous Topic

Next Topic



04:15

## Mitigating XSS Attacks



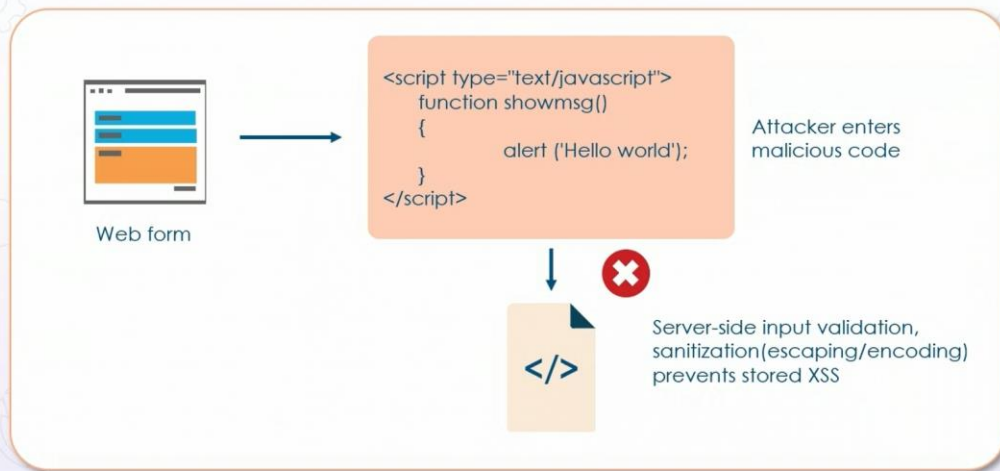
Validate user input



Sanitize user input



## Prevent JavaScript Injection



## Output Encoding



- **Neutralizes the injection of executable statements**
- **Translate special characters to a less dangerous form**
  - E.g. < should be translated to &lt;
- **Encoding is different from escaping**
- **Escaping takes away the meaning of a special character with a prefix**
  - E.g. \" treats the double quote as a literal string

## HTTPOnly Cookie Flag



Optional flag assigned to a Set-Cookie HTTP response header

Prevents client-side code from accessing the cookie