Cross Site Scripting

COMP620

Goals

- To understand how a cross site scripting attack works
- To know how to protect against XSS

- Some slides in this lecture come from Christopher Lam and Charles Frank

Typical Web Application Design

- Runs on a Web server or application server
- Takes input from Web users (via Web server)
- Interacts with back-end databases and third parties
- Prepares and outputs results for users (via Web server)
  - Dynamically generated HTML pages
  - Contain content from many different sources, often including regular users
    - Blogs, social networks, photo-sharing websites…

slide 3

XSS Example

- Client browser script asks the user for their name
- It sends a message to the web server.

https://example.com/myapp.php?user=Ken

XSS Example

- The name is “Reflected” back from the Web server to the client in a web page.

Hello Ken.

XSS Example

- Instead of entering their name at the prompt, the user could enter

<script>alert(‘xss’);</script>

- This is sent to the web server

https://example.com/myapp.php?user=
<script>alert(‘xss’);</script>

Result of Example

•When the “name” is included in the resulting web page

XSS vulnerability with JavaScript

```
<html>
<head>
<script type="text/javascript">
function showName() {
  var userin = document.getElementById("instuff").value;
  document.write("Hello " + userin);
}
</script>
</head>
<body>
<p>Example of XSS</p>
Username: <input type="text" onchange="showName();" id="instuff"
    maxlength="80" size="50" />
Password: <input type="text"/>
</body>
</html>
```

Top 10 Vulnerabilities

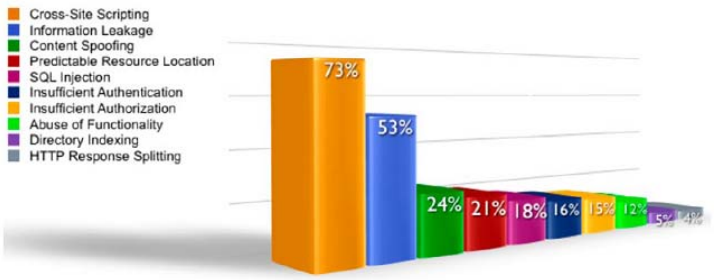
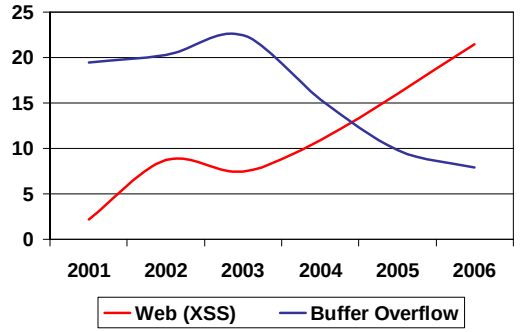


Figure 2. Top 10 vulnerability classes by percentage likelihood.

Vulnerability Stats

Source: MITRE CVE trends

Majority of vulnerabilities now found in web software



slide 10

XSS Attacks

MySpace worm (October 2005)
- When someone viewed Samy's profile:
  - Set him as friend of viewer.
  - Incorporated code in viewer's profile.

Paypal (2006)
- XSS redirect used to steal money from Paypal users in a phishing scam.

BBC, CBS (2006)
- By following XSS link from securitylab.ru, you could read an apparently valid story on the BBC or CBS site claiming that Bush appointed a 9-year old as head of the Information Security department.

March 4, 2009 SIGCSE

Attacking Yourself

- If you enter script in an input field that is then displayed in your browser, aren't you attacking yourself?

Non-Persistent (Reflected) XSS Attacks

- Most common type
- With invalidated user-supplied data in a resulting webpage without html encoding, client-side code can be injected into the dynamic page
- Then with some social engineering
  - Manipulating someone to perform actions
- An attacker convinces a user to follow a malicious URL which injects code into the resulting page
- Now the attacker has full access to that pages content

Persistent (Stored) XSS Attacks

- Allows the most powerful kinds of attacks
- First data is stored in a server provided by a web application
- It is later shown to a user on a webpage without any html encoding
  - Ex: Online message board that allows users to post messages for other users to read
- With this method, malicious scripts can be provided more than once
- An attack can affect a large amount of users and the application can also be infected by a XSS Virus or Worm

DOM-Based (Local) XSS Attacks

- Document Object Model
  - Standard object model for representing html or xhtml
- Problem exists within the page's client side script
- If an attacker hosts a malicious site, which contains a vulnerable website on a clients local system, a script can be injected
- Now the attacker can run the privileges of that users browser on their system "Local Zone"
- Can be either persistent or non-persistent

Mitigating XSS

1. Disallow HTML input
2. Allow only safe HTML tags
3. Filter output
   Replace HTML special characters in output
   ex: replace < with < and > with >
   also replace (,), #, &
4. Tagged cookies
   Include IP address in cookie and only allow access to original IP address that cookie was created for.

March 4, 2009 SIGCSE

XSS Problem

- XSS is a complex problem that is not going away anytime soon.
- The browser is insecure by design.
- It understands JavaScript.
- Disabling scripting seriously dampens the user's browsing experience.

March 4, 2009 SIGCSE