YGN ETHICAL HACKER GROUP

Whitepaper

WHAT MALICIOUS THINGS XSS CAN DO

By

d0ubl3_h3lix

Wed Jan 2, 2008

$Revision 0.1$

Next-Generation XSS

XSS stands for Cross Site Scripting. It was first called as CSS but renamed to XSS because of confusion with Cascading StyleSheet. XSS is used to execute alien JavaScript codes that are hosted in malicious web servers. Simply speaking, XSS is running malicious scripts on trusted web sites.

XSS has long existed since JavaScript was born. One of the main reasons that XSS was not prevalent is that JavaScript had dozens of cross-browser issues and capabilities – no actual standard supports for various browsers.

Now that JavaScript has been standardized as ECMA, and almost all modern browsers support far more complex calculations than early-day browsers, we can make sure XSS attacks work perfectly across such browsers along with the advent of AJAX techniques.

XSS has been looked down on and overlooked by a large part of web developers until now. XSS threats are hard to estimate. XSS is not a limited attack vector. How malicious it is depends only on how creative and wicked attacker is.

Benefits

From the attacker's standpoint, the following benefits can be derived from XSS attacks.

1. Sensitive Information Theft

   a) Credential Theft
   We can steal cookie and session data of currently logged-in users. Then we can bypass authentication by session riding or poisoning.

   b) File Theft
   With Ajax techniques, XSS can upload files to our servers from user's computer or user' network computers secretly and anonymously.

2. Intranet Scanning

   We can scan user's Intranet for sensitive information. With the aid of JavaScript port scanning method, we can also gather what network services are running.

3. Attacking Users

   Information theft is one kind of attacks. What more? Oh, yes. We can secretly

control his browser. We can steal his browser history for blackhat marketing. We can sneak his browser add-ons or plugs-in information, e.g., if his browser has vulnerable versions of Adobe plug-ins, we can take advantage of this hole to launch for other attacks. We can make his browser and computer crash – losing his currently unsaved works. Low risks? What's more, we can 0wn users' computers ultimately using JavaScript Buffer Overflow – downloading and executing zero-day viruses that infect all users' networks and from there, we can own the entire network. Combining with XSS payloads, every possible exploit (browser inherent flaws, plugs-in flaws …etc) can be launched. This stage of terminology is called 'Remote Code Execution' in hacking terms.

4. Proxifying XSS

   At the time we are interactively connecting with users via XSS, we can establish users' computers as our proxies to launch attacks to other hosts using JavaScript Malware payloads. Thus Intrusion Detection/Prevent System (IDS/IPS) will log only users' IP addresses not ours. Eventually we can launch the grandson version of Distributed Denial Of Service (DDOS) to a targeted system.

Countermeasures:

It is not that easy to protect XSS because there are a number of factors to be considered for protection. The main problem is that we have to deal with a wide variety of inputs and their valid strings. Definitely, effective XSS protection solution causes some kind of Denial of Service to users.

We can bypass most XSS filters by encoding our XSS with different character sets and different character representations and even with media file types such as JPEG, MP3, MOV ...etc. Although we can scan contents of those media files for searching XSS payloads, it will be resource overhead even for 3MB file size and attackers can cause DOS to our server by opening hundreds of connection. So I wish web browsers and their associated plugins or Antivirus wares should check those media files for malwares codes injection. My sound advice is on what you should do is watching proactively for XSS code injection while implementing a baseline protection for all inputs and outputs of your web applications. A good habit is to check xssed.com whether your site exists there.