Prewriting and free writing for ideas for this project

We aim to demonstrate and implement flaws in popular open source software and scripts that runs on a Linux environment via networking client and software relationship. We show the difficultly of XSS attacks, their practicality, and how they can may be prevented.

We will address various methods advantages or disadvantages in defense of such attacks. This may include costs, technical aspects and limitations, and any issues regarding to interference of quality service to customer/clients. We will cover methods of attack and their defenses and how they help to limit or prevent damage.

Cross Site Scripting gets a score of 9 for popularity, 3 for simplicity, and 5 for impact from book Hacking Exposed 7 (Stuart McClure, 2012, p. 557)

Free writing:

Cross site scripting a failing to manage user input. This allows for code to be injected into the clients end of the system and run with permission that of the user.

Implementation requires using JavaScript code. It also requires having a way to find capture the HTTP headers to find details on cookies and other details about how the site is coded.

How to defend from it:

System configuration review

Coding reviews

Using standard libraries instead of using custom coded libraries.

Updating systems or stacks, such as the LAMP stack

QA on the coding cycle

Pen-testing

Fuzzing code before release

Hardening other part of the system to prevent using other volutenriles to allow for an opening for XSS

Firewalls don’t work because Webservers are normally in DMZ or firewalls alone can’t detect the activity happening at higher levels than 4 on the OSI model. Many of these attacks happen in application layer.

Implementations:

About the system:

The system is SEED Lab. It’s a VM of Ubuntu 12.04 running various OSS code to allow training for security students. The stack is a LAMP Stack, Linux, apache2, MySQL, PHP. On top of the stack, Elgg, an OSS social media forum board is the target application. These implementations are by default set to be insecure. Any normal security can be turned on as needed.

We failed on our first try to implement a XSS on our demonstration system to run JavaScript code. None of us have much experience with JavaScript code. However, after 4 tries

We forgot to include the object of window, after updating the code to window.alert(“Code has been Run”); it ever worked. After many attempts and failures, nothing worked. Placing it different boxes within the profile can have different results. Upon further inspection, the XSS protection was turned on when it didn’t work. Also script HTML tags have to be included. These have been left out on many of the attempts.

Second attempt:

While attacking, the site would time out while connect to an external script, allowed for the creation of script stored somewhere else the Elgg script could run.

Code:

<script type="text/javascript"

src="http://127.0.0.1/alert.js">

</script>

Alert.js

window.alert(“Code has been Run”);

On a second attempt to implement the same thing, it would not work.