XSS Tools Cheat Sheet

by binca via cheatography.com/44948/cs/13443/

Overview

XSSer, xsssniper, XSScrapy are just a few available

Interception proxies afford important capabilities including manual discovery and verification of XSS flaws, facilitating fuzzing and finding less obvious locations for injection including User-Agents, Cookies, HTTP Headers, and Referer.

Burp Intruder

Assists with manual or automated fuzzing for XSS reflection tests.

Use **Battering Ram**, which submits 1 payload at multiple positions simultaneously.

Injection points can include **URL parameters, HTTP User-Agent, HTTP Referer, Cookies and more.**

Grep Payloads is an options in Burp Intruder that searches the applications response for the submitted payload.

Combining Battering Ram and Grep Payloads automates fuzzing multiple injection point per request and resolving whether the payload is in the response.

Sniper can follow the combination to determine which injection point reflected.

XSSsniper

Python-based tool that attempts to automate reflection testing.

Simple and quick, it can perform spidering, scan the target and inject with simple syntax.

xsssniper -u "URI" --crawl --forms

--crawl instructs it to crawl a site for entry points

--forms instructs it to look for injectable forms

--http-proxy [IP: Port #] Runs it through a proxy

XSSer

Python-based XSS discovery tool with command and GTK GUI interface that has a wizard.

--heuristics attempts to determine filtering used by the app

Bypass techniques included:
--hex hexadecimal encoding
--dec decimal
--une unescape()
String.FromCharCode()

Can be pointed at a proxy with **--proxy**

Contains options to discover XSS flaws in HTTP User-Agent, HTTP-Referer, HTTP Cookies and others.

XSSer (cont)

Has a list of potential XSS payloads, though they have no been recently updated.

XSScrapy

Python-based XSS spider.

Uses "magic" reflection string of **9zqjx** to attempt to discover potentially reflected input and location. This string can prepended and appended to payloads as well.

Will employ one of 3 payloads based on location:
'"()=<x>
'"(){}[]
JaVAscRIPT:prompt(99)

The focus is on using innocuous input followed by a crafted payload to determine filtering and injection efficiently.

Typically has reduced false positives.

Reflected POST

Getting users to send a GET request via a malicious link or fetching a resource is fairly simple

Getting a user to send a malicious POST is trickier.

Hosting a malicious HTML form on an attacker controlled site provides an easy strategy, especially if the form is hidden and automatically submits.

Ensure the target supports parameters being passed as query parameters.

HTTP Post does not use URL for parameters but instead sends variables in the payload of the request.

get2post.py is a Python script that you pass the target and PST payloads as URL parameters. It produces a GETified URL that uses the target parameter for redirection and additional query parameters become POST payloads.