Cheatography

BeEF (Browser Exploitation Framework) Cheat Sheet

by binca via cheatography.com/44948/cs/13444/

Overview

Extensive Ruby based framework with interprotocol features that focuses on payload delivery.

Various panels control a victim's browser.

If a zombie is offline when a command is issues it is sent when the browser reconnects.

The far left panel lists zombies, the next panel contains modules and the far right has description and configuration options for the selected module.

Employs JavaSript file hook.js, which is generated on the fly, to hook a browser. This file is injected via a XSS attack. The hook.js file changes based on the issues commands.

Note: hook.js does not exist locally on the file system but can be viewed when running BeEF by downloading it: **wget** **http://192.168.1.8:300/hook.js****

Icon Color Codes

Green	**Works on victim**
Orange	**Works but may be visible**
Grey	**Not confirmed to work**
Red	**Does not work**

Modules

Autorun	
Clipboard Stealing	**Steals contents of clipboard**
JavaScript Injection	
Request Initiation	**Instructs the zombie browser to make HTTP requests as directed. Excellent for CRF attacks or t download software to the victim. Does not return page content to the attacker.**
History Browsing	**Retrieves history via brute forced and can be used to fingerprint victim, map infrastructure, and determine other targets. It requires a word lists, that is only prepopulated with a few terms.**

Other Capabilities

Controlling Zombies	
Port Scanning	**Port scan a network through the zombie, with a distributed network of them there is a low risk of detection.**
Browser Exploitation	**Injects an iframe into the zombie to deliver a browser exploit. This requires a running instance of Metasploit reachable by the BeEF server. While it supports AutoPWN it is not recommended due to instability.**
Interprotocol Exploitation	**Because many protocols are forgiving and ignore junk including HTTP Request headers, BeEF will inject a payload of a service-side exploit into an HTTP request to be delivered to the target server by the hooked browser. A BindShell could be the payload giving the attacker acces throuhg the BeEF controller applicaiton.**