Intro to BeEF: Browser Exploitation Framework

Overview

The Browser Exploitation Framework (BeEF) is an open-source penetration testing tool used to test and exploit web application and browser-based vulnerabilities. It uses web app and browser vulnerabilities to assess the security of a target and carry out further intrusions. All of this is done by "hooking" browsers.

The BeEF server is a Ruby on Rails application that communicates with the "hooked browser" through a web-based user interface.

This lab covers the basics of BeEF and will instruct you on how to launch the tool, accessing the interface, and executing a simple intrusion on a localhost browser.

Lab

1) Log into virtual machine (root;Bearcat1)
2) Open Terminal and type the following commands, in order:
   
   cd /usr/share/beef-xss
   ./beef

After you’ve done this, you’ll see the following screen after the tool loads:
After you see that the tool has loaded successfully, open up IceWeasel (Kali’s default browser), copy the localhost (127.0.0.1) UI URL, and navigate to that location in the browser. You’ll be greeted with a login screen that uses the credentials **beef;beef**.

Once logged in, you’ll see your BeEF Control Panel. The first default tab, Getting Started, not only provides two demo pages you can use to hook a browser, but also a great overview of the tool itself and an icon legend. The Logs tab provides a listing of events for all hooked browsers.
In the far left frame of the BeEF Control Panel, you’ll see two folders (Online and Offline Browsers). It’s pretty self-explanatory as to what they are: the Online Browser folder contains the connection(s) for currently hooked and connected browsers, while the Offline Browser folder contains hooked or previously hooked, but not connected browsers.

Click on an IP address that shows up under Online Browsers, and that’ll show you expanded options in the next two panels. The Module Tree is where the exploits and commands can be found to execute on the hooked browsers.
For simplicity’s sake, let’s see if we can try to trick the user into providing some kind of login credentials to us.

To do this, click on the IP address of a currently online browser (for this lab, you should be using localhost, so the IP would just be 127.0.0.1). Then click on the Commands tab, expand the Social Engineering tab, and click on Pretty Theft. Change the dialog type to Generic, and then click Execute.

Since everything is being done via localhost, and is contained within the virtual machine, play around with the various exploits available to get a feel for the tool and envision ways it can be used to gather information from unsuspecting users!