

The project has been funded by the European Commission. The Education, Audiovisual and Culture Executive program (EACEA), TEMPUS IV. The content of this presentation reflects the opinion of the author.

Module 7. Exploitation using client-side attacks

Penetration testing course



BeEF (The Browser Exploitation Framework)

- It is a penetration testing tool that focuses on the web browser.
- BeEF examines exploitability within the context of the one open door: the web browser.
- BeEF will hook one or more web browsers and use them as beachheads for launching directed command modules and further attacks against the system from within the browser context.

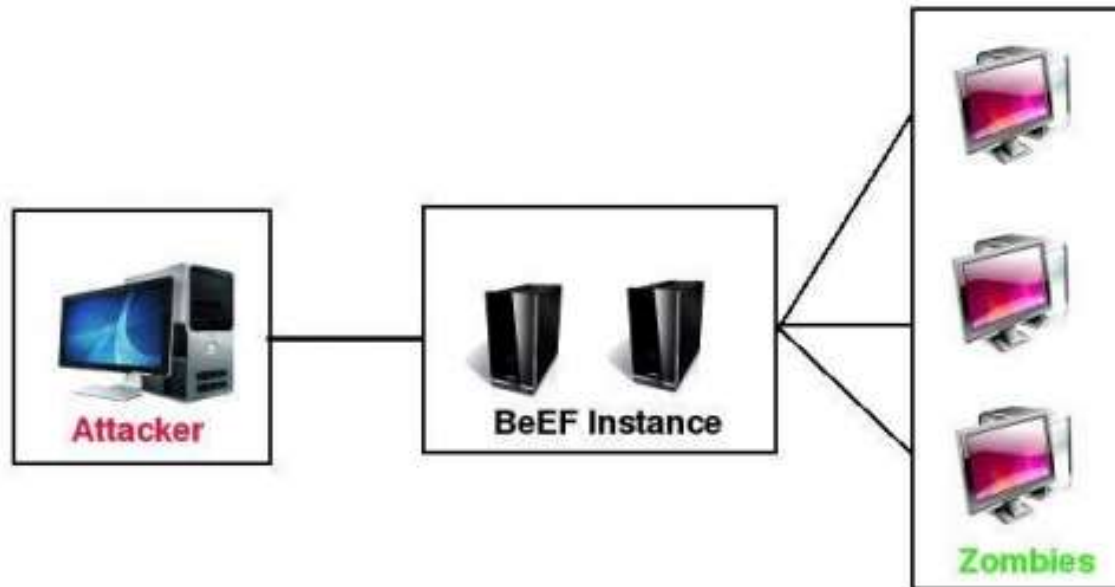


BeEF

- Open source penetration tool
- Tests and exploits Web and Browser-Based vulnerabilities
 - Use client-side attack vectors and leverages XSS
 - Hooks and fingerprints browsers
- BeEF server can issue the following commands:
 - Redirection of browser
 - Changing URLs
 - Dialog box generating on victim computer
 - Uploading of files to victim computer
 - ...
- API integration with other tools
 - Integration with Metasploit and others



Attack scheme



BeeF user interface

The screenshot displays the BeeF user interface. On the left, a sidebar titled "Hooked Browsers" contains two sections: "Online Browsers" and "Offline Browsers". The "Offline Browsers" section is expanded, showing a list of browsers with IP addresses: 10.1.1.2 (repeated five times) and 10.1.1.1. Red arrows point from the text "Hooked browsers" to the "Offline Browsers" section and from "Offline browsers" to the list of IP addresses. The main content area has a tabbed interface with "Getting Started" and "Logs" tabs. Red arrows point from the text "Home tab" to the "Getting Started" tab and from "Log Tab" to the "Logs" tab. The "Getting Started" tab is active, displaying the BeeF logo (a blue bull head) and the text "THE BROWSER EXPLOITATION FRAMEWORK PROJECT". Below the logo, it says "Official website: <http://beefproject.com/>". The main content area contains the following text:

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page [here](#), or the advanced version [here](#).

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

- Main:** Display information about the hooked browser after you've run some command modules.
- Logs:** Displays recent log entries related to this particular hooked browser.
- Commands:** This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript: for example they may gather information about the Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

Each command module has a traffic light icon, which is used to indicate the following:

- The command module works against the target and should be invisible to the user
- The command module works against the target, but may be visible to the user
- The command module is yet to be verified against this target
- The command module does not work against this target

At the bottom left of the interface, there are two buttons: "Basic" and "Requester". The top right corner of the window shows the version "BeEF 0.4.3.7-alpha" and links for "Submit Bug" and "Logout".



BeEF commands for browser

The screenshot displays the BeEF Control Panel interface. The browser's address bar shows the URL `http://192.168.1.101:3000/ui/panel`. The interface includes a sidebar for 'Hooked Browsers' with sub-sections for 'Online Browsers' (containing 192.168.1.101 and 192.168.1.100) and 'Offline Browsers'. The main content area is divided into several panels: 'Getting Started', 'Logs', and 'Current Browser'. The 'Current Browser' panel has tabs for 'Details', 'Logs', 'Commands', 'Rider', 'XaaSays', and 'Ipec'. The 'Commands' tab is active, showing a 'Module Tree' on the left and a 'Module Results History' table on the right. The 'Module Tree' lists various categories and their counts: Browser (43), Chrome Extensions (6), Debug (6), Exploits (48), Host (15), IPEC (6), Metasploit (0), Misc (7), and Network (9). The 'Network' category is expanded, showing sub-items like DNS Enumeration, DOSer, Detect Social Networks, Detect Tor, IRC NAT Pinning, Ping Sweep, Port Scanner, Fingerprint Network, and Ping Sweep (Java). The 'Module Results History' table has columns for 'id', 'data', and 'label', and contains a message: 'The results from executed command modules will be listed here.' To the right of the table is the 'Man-In-The-Browser' section, which includes a 'Description' field with the text: 'This module will use a Man-In-The-Browser attack to ensure that the BeEF hook will stay until the user leaves the domain (manually changing it in the URL bar)'. At the bottom right of the interface is an 'Execute' button. The status bar at the bottom shows 'Basic' and 'Regular' tabs, a 'Ready' indicator, and a 'Done' label.

