

Forum:[Hack Forums](#)

Thread: [\[Tutorial\] Run BEeF on WAN || Control any browser](#)

guardian - December 29, 2020, 5:54 am

Run BEeF on WAN
Control browsers by the click of a link
About BEeF:

Beef stands for Browser exploitation framework
The official website can be found [here](#)
Beef utilizes javascript to control the targets browser and let you perform certain actions inside their browser, which will help you hacking social media accounts, capture credentials or even take control of their complete box

In this tutorial i will showcase, how beef can be installed outside of your LAN
That way you can send the link to anyone and capture information
This tutorial will not show details about every single attack that can be done with BEeF
If you want a tutorial on some specific function or type of function, please let me know with a post

Requirements

For this tutorial a Debian 10 VPS will be used, although you can do the same from within your home network by forwarding the port (which I would not recommend)
A cheap 5\$ VPS will be do the job and keep your home network clean and without any open ports

Installing BeEF

We are cloning the framework from its git repository, therefor make sure that you have git installed

Code `sudo apt -y install git`
When you have git installed, clone into the repository

Code `git clone https://github.com/beefproject/beef/`
A folder named beef will be now be inside your current working directory

Use
Code `cd beef` to change to he beef directory, where further configuration has to be done
Now run
Code `./install` This is an install script that automatically installs BeEF's dependenices
The script will have bugs on some operating systems and not be able to install certain packages, so the rest will be done manually

Beef heavily relies on Ruby and needs Ruby 2.5 <
If your packet manager does not include a version of Ruby that is 2.5 or higher, you can refer to the install.txt within the beef directory and add a repository that supports it
In my case running
Code `apt -y install ruby` Downloaded a version of ruby that is higher than 2.5, so steps about solving errors regarding it, will not be discussed in this tutorial

As the script could not install some packages manually, I know install bundler, which is required to be able to run the ruby code successfully

Code `apt -y install bundler`

After the installation is completed run:

Code `bundle install`

You might run into an error installing the nokogiri gem

In my case this could be solved by manually running the install command

Code `gem install nokogiri -v '1.10.10' --source 'https://rubygems.org/`When the installation succeeded, again run

Code `bundle install`

You might also run into an error installing the curb gem

That's due to the fact, that it uses curl libs to compile natively, so you have run

Code `sudo apt-get install libcurl4-openssl-dev`Code `gem install curb -v '0.9.11' --source 'https://rubygems.org/'`As before, after successful installation of the curb gem, run

Code `bundle install`

The same goes for sqlite3, it needs the packages 'sqlite3' and 'libsqlite3-dev' to function

Code `sudo apt-get install sqlite3 libsqlite3-dev`Code `gem install sqlite3 -v '1.4.2'`Code `bundle install`

In my case I also had to install nodejs manually

Code `sudo apt-get install nodejs`

Now everything should be ready and we can edit the configuration

Inside the config.yaml, you can change the configuration, such as the port the server runs on, your user credentials ...

The first thing you should change are your user credentials, as it is never good to use default credentials (automated scanners etc.)

Code `nano config.yaml`

I changed my password and username for the beef panel

Here is a screenshot of the modified configuration file

You can now save the file by typing CTRL + O then enter

Exit the file then by typing CTRL + X

Now we are ready to start the framework

Code `./beef`

After executing with the command above, you will be provided a few links

I censored the public IP of my server, but the link with the public IP can now be used outside LAN
/ui/panel links will get you to your panel, where you can see the computers your js had been executed on or maybe they are still executing it

/hook.js is a javascript that you could, for instance, embed into a website that does not look suspicious at all, by just linking it as a source from within a script tag

You now know how to install beef and make it run on WAN, so anyone in the world can run your beef script

Look into the panel and test out a few functions, as beef can do a lot of things with devices hooked to it

I will not get into functions that beef provides in this tutorial, but if you would like to see a tutorial on some specific function or other penetration testing stuff, I would appreciate requests
Requests for other tutorials and constructive feedback are also warmly welcome

hdawg12 - *September 5, 2022, 8:18 pm*

do I need ngrok to run outside my LAN?