2.1.1 - pyWeb

Overview

Python is a powerful tool in the pen-testing toolbox. In this module, we will be looking at using the Python web-server for hosting content on on our local machine.

pyWeb

How is it used

`pyWeb` is run on the machine where you would like to hos the web server. When run, it will create a web server with a webroot (the highest folder in the web server, the folder which is served) as your current local folder.

> For example, if I ran the pyWeb server in `/opt/web` and then browse to `http://localhost:80`, I could see the contents of `/opt/web` as my webroot.

Why is this important

Web traffic is one of the most common forms of traffic. You will likely find that many of your targets permit traffic (both inbound and outbound) on port 80. Being able to host web-servers easily (on both attacking and target machines) will greatly improve the ease in which you can infill / ex-fill data.

Real-word applications

You would use a pyWeb server to host files for transfer to a target.

> Consider a situation where you needed to transfer an exploit or enumeration script to a target without an internet connection. You could host the files on a `pyWeb` instance and transfer them directly between attacker and target.

Don't be limited to just hosting the pyWeb server on the attacker. Even with a low privilege account on a `target`, you could bind to a port greater than `1023` to ex-filtrate data.

Potential Issues

You can host `pyWeb` on any port that is not currently in use. It will require `sudo` privileges to bind to `privileged` ports (ports less than 1023).

Keep in mind that a `target` machine may not be able to connect back to your `attacker` on certain ports. If port `80` does not work, try a different port.

Exercise

Installation

You do not need to install any additional libraries for the `Python` web server.

Usage

Both Python2 and Python3 are capable of hosting a `pyWeb` server. However, the commands to start it are slightly different.

> Both Python2 and Python3 are capable of binding to any port, though require sudo for `privileged` ports.

Python3

The example below would host a web server on the `privileged port 80` using `Python3`

```
/usr/bin/sudo python3 -m http.server 80
```

Python2

The example below would host a web server on the `un-privileged port 1337` using Python2.

```
python2.7 -m SimpleHTTPServer 1337
```

> Keep in mind `sudo` is required for ≤ port 1023.

Aliases

No one likes remembering lengthy commands when you can use a simple `alias` to start the pyWeb server.

Add the commands to your `.zshrc` file so that you may use `pyweb 80` or `pyweb27 80` to start the pyWeb server.

> `pyweb27` will use Python2.7 and `pyweb` will use Python3.

```
    alias pyweb27="/usr/bin/sudo python2.7 -m SimpleHTTPServer"
```

```
    alias pyweb="/usr/bin/sudo python3 -m http.server"
```

```
alias listen='/usr/bin/sudo rlwrap -r  nc -nlvp'
alias autorecon='/usr/bin/sudo /home/kali/.local/pipx/venvs/AutoRecon/bin/autorecon'
alias pyweb="/usr/bin/sudo python3 -m http.server"
alias pyweb27="/usr/bin/sudo python2.7 -m SimpleHTTPServer"
alias update="sudo apt update"
alias upgrade="sudo apt upgrade"
alias oslab='sudo /usr/sbin/openvpn /home/kali/Documents/OS-PWK.ovpn'
alias htb='sudo /usr/sbin/openvpn /home/kali/Documents/htb.ovpn'
alias htbarena='sudo /usr/sbin/openvpn /home/kali/Documents/htb-arena.ovpn'

alias grep='grep --color=auto'
alias fgrep='fgrep --color=auto'
alias egrep='egrep --color=auto'
alias diff='diff --color=auto'
alias ip='ip --color=auto'

export LESS_TERMCAP_mb=$'\E[1;31m'     # begin blink
export LESS_TERMCAP_md=$'\E[1;36m'     # begin bold
export LESS_TERMCAP_me=$'\E[0m'        # reset bold/blink
export LESS_TERMCAP_so=$'\E[01;33m'    # begin reverse video
export LESS_TERMCAP_se=$'\E[0m'        # reset reverse video
export LESS_TERMCAP_us=$'\E[1;32m'     # begin underline
export LESS_TERMCAP_ue=$'\E[0m'        # reset underline
```

Once you have added the alias, you can either logout / login or `source` the file.

```
source ~/.zshrc
```

Once that is done, you can host the web server uses the created aliases.

```
┌──(kali㉿kali)-[/opt/web]
└─$ pyweb 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Assessment

Nil.