ringzer0team – Coding – Execute me if you can

– Kileak

Some shellcode execution, shouldn't be that hard...

Started with retrieving the page, extracting the shellcode and then just pass it to some shellcode execution binary.

Common shellcode testing code should suffice for this:

shellexec.c

```
void main(int argc, char *argv[]) {
      (*(void(*)()) argv[1])();
}
```

This program just takes a shellcode as a parameter and executes it right away.

```
gcc shellexec.c -o shellexec -fno-stack-protector -z execstack
```

and we should be ready to go.

So we'll download the shellcode and execute it.

```
# Retrieve the challenge text
code = rz.GetChallenge(121)
soup = BeautifulSoup(code)
soup = soup.find ('div', {'class':'message'})
soup = soup.get_text()
soup = soup.replace ('----- BEGIN SHELLCODE -----', '').replace ('----- END
SHELLCODE -----', '').strip()
```

At first I tried to just call my shellcode-executer and read the output from it

```
solution = subprocess.check_output(['./shellexec', soup.decode(„string-
escape“)])
```

Well, just sending it to the challenge page now, and... uhm, the passphrase gets printed out but doesn't land in the solution variable. What evil magic is that?

After reading europa's write-up afterwards, which analyzes the shellcode in depth, it got clear, that Mr. Un1k0d3r switched our output from stdout to stdin, but at that point I was quite stumped :)

If we don't understand something, google is always there to help out. And after several failures I managed to get the shellcode print out its result at least to a temporary file. Should be enough to get the task done

```
# execute the shellcode placing it into a temp file
rz.ShowAction("Executing shellcode")
os.system('./shellexec "%s" > shellout 0>&1' % soup.decode("string-escape"))
```

Then just open the file, read the solution from it and send it back

```
# Send it back to ringzer0
response = rz.SendSolution(121, solution)
flag = rz.GetFlag(response)

rz.ShowAction("Received flag : %s\n" % flag)
```

Let's give it a try and

```
$ python code121.py

ringzer0team - Coding Challenge 121 Solver
-------------------------------------------
 [+] Sending request: http://ringzer0team.com/challenges/121
 [+] Executing shellcode
 [+] Read shellcode result
 [+] Sending solution: http://ringzer0team.com/challenges/121/JStQndSVJndk
 [+] Received flag : FLAG-XXXXXXXXXXXXXXXXXXXXXXXXXX
```

code121.py

```
import rz
import os
from bs4 import BeautifulSoup

print("\nringzer0team - Coding Challenge 121 Solver")
print("-------------------------------------------")

# Retrieve the challenge text
code = rz.GetChallenge(121)
soup = BeautifulSoup(code)
soup = soup.find ('div', {'class':'message'})
soup = soup.get_text()
soup = soup.replace ('----- BEGIN SHELLCODE -----', '').replace ('----- END
SHELLCODE -----', '').strip()

# execute the shellcode placing it into a temp file
rz.ShowAction("Executing shellcode")
os.system('./shellexec "%s" > shellout 0>&1' % soup.decode("string-escape"))

# read the result
rz.ShowAction("Read shellcode result")
with open("shellout", 'r') as f:
      solution = f.read()
      f.close()

# Send it back to ringzer0
response = rz.SendSolution(121, solution)
flag = rz.GetFlag(response)

rz.ShowAction("Received flag : %s\n" % flag)
```

rz.py (my base library for the ringzer0 challenges)

```
import urllib2
from bs4 import BeautifulSoup

# Common ringzer0 variables
RZ_BASE_URL = 'http://ringzer0team.com'
RZ_AUTH_COOKIE = 'PHPSESSID=XXXXXXXXXXXXXXXXXXXXXXXX'

def ShowAction(msg):
      print(" [+] %s" % msg)

def GetChallenge(challID):
      request = urllib2.build_opener()
      request.addheaders.append(('Cookie', RZ_AUTH_COOKIE))
      requestURL = '%s/challenges/%s' % (RZ_BASE_URL, challID)
      ShowAction('Sending request: %s' % requestURL)
      response = request.open(requestURL)
      return response.read()

def SendSolution(challID, sol):
      request = urllib2.build_opener()
      request.addheaders.append(('Cookie', RZ_AUTH_COOKIE))
      requestURL = '%s/challenges/%s/%s' % (RZ_BASE_URL, challID, sol)
      ShowAction('Sending solution: %s' % requestURL)
      response = request.open(requestURL)
      return response.read()

def GetFlag(response):
      soup = BeautifulSoup(response)
      flag = soup.find ('div', { 'class':'alert alert-info'})

      if flag != None:
            flag = flag.get_text()
            return flag

      return "Fail"
```