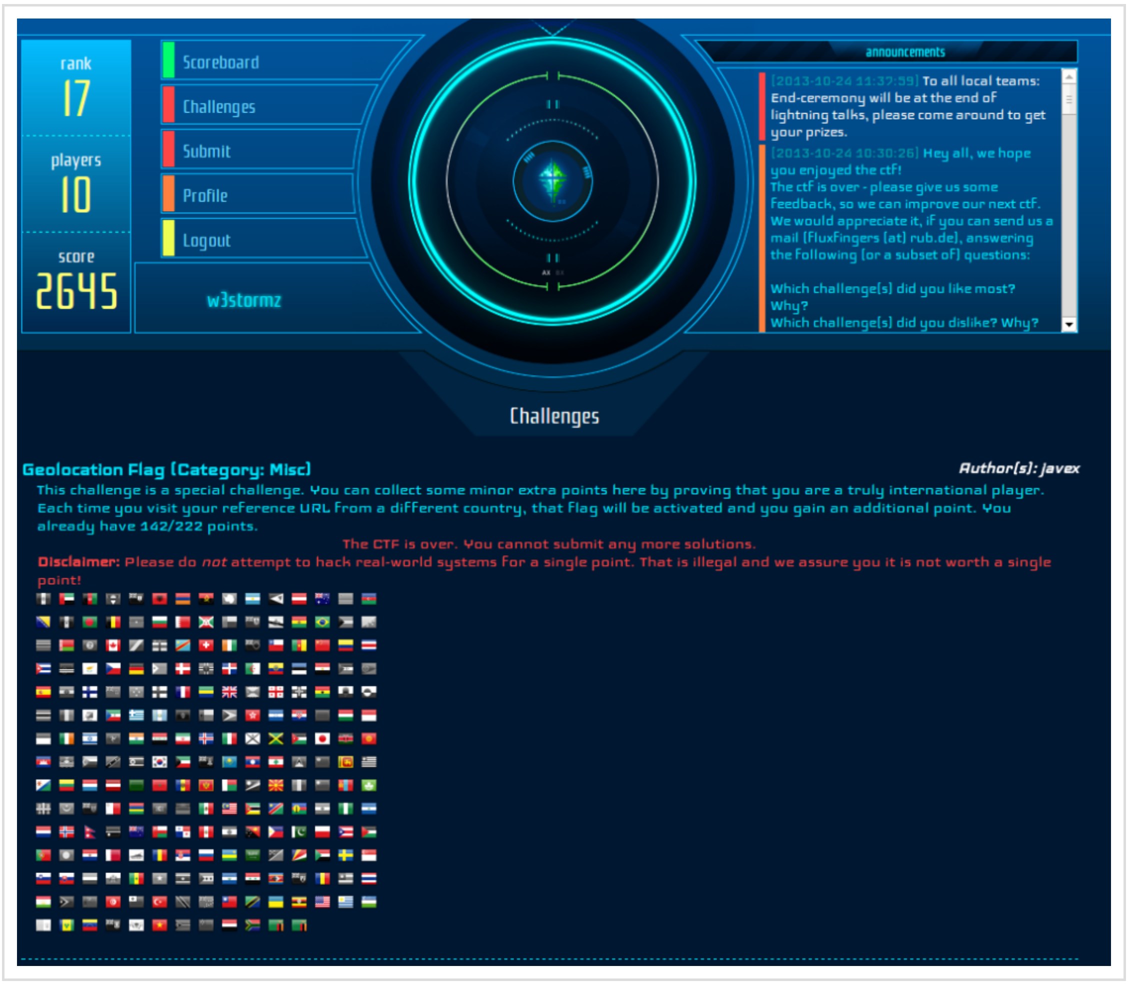
Nikaiw's corner

Scanning the internet for fun and country flags [hack.lu]

Posted on **October 29, 2013**

In hack.lu 2013, there was a not too serious challenge called “Geolocation”. The aim of the challenge was quite simple, you had the HTTPS link to a specific page and you had to reach it from a maximum of different country. Each different country connection was giving a point and activating the matching country flag.



[2013-10-24 11:37:59] To all local teams:
End-ceremony will be at the end of lightning talks, please come around to get your prizes.
[2013-10-24 10:30:26] Hey all, we hope you enjoyed the ctf!
The ctf is over - please give us some feedback, so we can improve our next ctf.
We would appreciate it, if you can send us a mail (fluxfingers (at) rub.de), answering the following (or a subset of) questions:

Which challenge(s) did you like most? Why?
Which challenge(s) did you dislike? Why?

Challenges

Geolocation Flag (Category: Misc) *Author(s): javex*

This challenge is a special challenge. You can collect some minor extra points here by proving that you are a truly international player. Each time you visit your reference URL from a different country, that flag will be activated and you gain an additional point. You already have 142/222 points.

The CTF is over. You cannot submit any more solutions.

Disclaimer: Please do *not* attempt to hack real-world systems for a single point. That is illegal and we assure you it is not worth a single point!

This was a not so fancy challenge but I really found it fun enough to spent time on it, I probably thought the final result would give an interesting overall idea about the world connectivity. For those who did CSAW CTF it was inspired and really close to the first part of the exploit challenge SCP.

I came across this nice write-up of it, making use of Tor network exit nodes. While a friend started making use of proxy and had already validate more than 60 country. I quickly wrote the following dirty script parsing the missing country and making use of tor exits nodes the same way:

```
#!/usr/bin/env python2
import requests
from iso3166 import countries
url="https://ctf.fluxfingers.net/challenges/24"
cookies = dict(session='3f0c74e78d8b526c266c7d3e4ad3988a4a0fd0a
data= requests.get(url,cookies=cookies,verify=False).content
lines = data.split('\n')
flags = []
for i in lines:
  if "inactive" in i:
    flag = i[75:77]
    flags.append(flag)
```

```
print flags
for todo in flags:
  print countries.get(todo)[0]
fd = open("torrc", "w")
fd.write("ExitNodes {" + "},{".join(flags) + "}")
fd.close()
#started via: while true; do ./tor.py; tor -f torrc & ; PID=${!
```

It also made use of the package iso3166 because it's nicer to have the TLD converted in the full name of the missing countries. With both of our scripts we quickly reached more than 100 country. But unfortunately Tor gets quickly limited in term of country exit nodes. I started to hunt for proxy of the missing one, two websites were quite helpful, leaving the ability to provide country name or tld.

- http://spys.ru/free-proxy-list/
- http://gatherproxy.com/proxylist/country/?c=

As well as the following script:
- http://www.cr0security.com/pyproxy-proxy-hunter-and-tester-a-high-level-cross-protocol-proxy-hunter-python-library/

We were over 115 countries but it was really getting hard to increase the score. We were getting struck, some countries had no proxy referenced on any list or none of those listed were working.

But I thought to myself, some of those country may still have open proxy. Open proxy are most of the time misconfigured proxy of enterprise or organisation which accept external connection.

Some month ago a link made the buzz about people who released a tool giving the ability to scan the whole Internet in not less than 45 min Since I didn't need to scan the entire internet, that was looking great.

- Getting the subnets:

I started by downloading the full geoip database using apt-get

```
apt-get source geoip-database
```

It provided me a "GeoIPCountryWhois.csv" with all the registered ranges of IP by countries. But zmap is taking subnets using CIDR notation, not IP ranges so I needed to convert those range.

- Converting range into subnet:

I used this small piece of c code called iprange for this: With the following script I selected the missing iprange from GeoIP

```
#!/usr/bin/env python2
import requests
import csv
from iso3166 import countries
url="https://ctf.fluxfingers.net/challenges/24"
cookies = dict(session='3f0c74e78d8b526c266c7d3e4ad3988a4a0fd0a
data  = requests.get(url,cookies=cookies,verify=False).content
lines = data.split('\n')
flags = []
for i in lines:
  if "inactive" in i:
    flag = i[75:77]
    flags.append(flag)
fd = open('GeoIPCountryWhois.csv','rb')
reader = csv.reader(fd)
fulldata = []
for row in reader:
  if str.lower(row[4]) in flags:
    fulldata.append((row[0],row[1],str.lower(row[4])))
output = open('subnets.tmp', 'w')
for each in fulldata:
  output.write('"' + each[0] + '"' + ", " + '"' + each[1] + '"'
output.close()
```

Now I just had to give it to iprange to get a nice subnet list

```
cat subnet.tmp |./iprange > subnets
```

After compiling zmap, I could start the scan on the most used proxy ports I started a small bash script:

```
#!/bin/bash
./zmap -p 8080 -w /home/nikaiw/tools/subnet -o /home/nikaiw/res
./zmap -p 3128 -w /home/nikaiw/tools/subnet -o /home/nikaiw/res
./zmap -p 1080 -w /home/nikaiw/tools/subnet -o /home/nikaiw/res
./zmap -p 8888 -w /home/nikaiw/tools/subnet -o /home/nikaiw/res
./zmap -p 80 -w /home/nikaiw/tools/subnet -o /home/nikaiw/resul
```

In some minutes I had scanned the whole missing countries including Vatican, North korea and Antarctica !

- Exploiting the results

Now the most important, I had to exploit the result. I wrote this python script loading the data into a workQueue and started 200 threads on it.

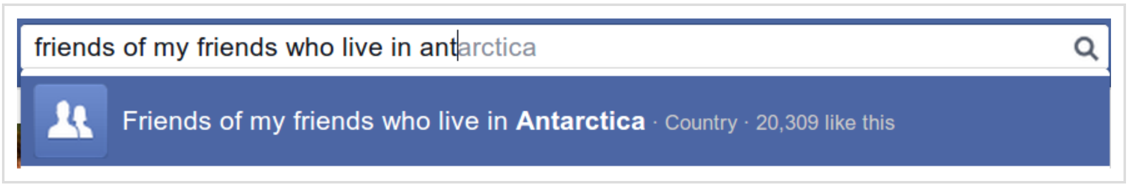
```
#!/usr/bin/python
import Queue
import threading
import time
import requests
import json
import sys

exitFlag = 0

class myThread (threading.Thread):
    def __init__(self, threadID, q):
        threading.Thread.__init__(self)
        self.threadID = threadID
        print "created thread: " + str(threadID)
        self.q = q
        self.killed = False

    def run(self):
      print "started " + str(self.threadID)
      process_data(self.threadID,self.q)

def process_data(tid,q):
  while not exitFlag:
    queueLock.acquire()
    if not workQueue.empty():
      job_id = q.get()
      prox = dict(http="http://" + job_id.rstrip() + ":808°")
      print repr(prox)
      queueLock.release()
      print "%s processing %s" % (tid, job_id)
      try:
        myreq = requests.get('https://149.13.33.74:443/ref/LoH9
        print myreq.status_code
      except:
        pass
    else:
      queueLock.release()

queueLock = threading.Lock()
```

```
workQueue = Queue.Queue(0)
threads = []

# Create new threads
for i in range(200):
  thread = myThread(i+1, workQueue)
  thread.start()
  threads.append(thread)

try:
  print "Filling the queue"
  queueLock.acquire()
  fd = open('result-8080.txt','r')
  for line in fd:
    workQueue.put(line)
    queueLock.release()

  # Wait for queue to empty
  while not workQueue.empty():
    pass

  # Notify threads it's time to exit
  exitFlag = 1

 # Wait for all threads to complete
  for t in threads:
    t.join()
    print "Exiting Main Thread"
    sys.exit(0)
    # Exit nicely if we hit ctrl-c
except KeyboardInterrupt:
  print "Ctrl-c received! Sending kill to threads..."
  for t in threads:
  t.killed = True
  exitFlag = 1
```

Leaving this script working during the night on the differents result of zmap we finally reach **142 countries**, not bad but still behind the Norwegian team HIGIMT who succeeded in reaching 150 countries.

A part I certainly underestimated, that they used a lot was the social engineering and social networking. There is a known theory that you can reach anyone in the world with just 6 degrees of separation.

One axis of research was using facebook graph-search with requests like ”people I may know in <country” or “friend of my friend who live in <country>”.


friends of my friends who live in antarctica

Friends of my friends who live in **Antarctica** · Country · 20,309 like this

The last possibility was simply spamming people email, forum, etc with <img> link pointing to the url of the challenge but I was not too interested in that. However I really want to thanks the few people I bothered and who helped me reaching some more country.

Finally, for the record here is the list of countries we couldn't reach in time:

```
Andorra
Antigua and Barbuda
Anguilla
Antarctica
American Samoa
Aruba
Barbados
Burkina Faso
Benin
Bermuda
Brunei Darussalam
Bahamas
Bhutan
Botswana
Belize
Congo
Central African Republic
Cook Islands
Cape Verde
Djibouti
Dominica
Western Sahara
Eritrea
Ethiopia
Fiji
Micronesia, Federated States of
Faroe Islands
Grenada
Guernsey
Gibraltar
Greenland
Gambia
Guinea
Guadeloupe
Guam
```

```
Guinea-Bissau
Guyana
Haiti
Monaco
Isle of Man
Jersey
Kiribati
Comoros
Saint Kitts and Nevis
North Korea
Cayman Islands
Saint Lucia
Liechtenstein
Liberia
Marshall Islands
Mali
Myanmar
Martinique
Mauritania
Montserrat
Maldives
Malawi
Niger
Nauru
French Polynesia
Palau
Réunion
Solomon Islands
Sierra Leone
San Marino
Somalia
Suriname
Sao Tome and Principe
Turks and Caicos Islands
Togo
Timor-Leste
Turkmenistan
Tonga
Trinidad and Tobago
Tuvalu
Holy See (Vatican City State)
Virgin Islands, British
Virgin Islands, U.S.
Vanuatu
Samoa
```

This entry was posted in **writeup** by **nikaiw**. Bookmark the **permalink [http://nikaiw.io/scanning-the-internet-for-fun-and-country-flags-hack-lu/]** .

0 THOUGHTS ON “SCANNING THE INTERNET FOR FUN AND COUNTRY FLAGS [HACK.LU]”

hellok

on **October 31, 2013 at 2:53 am** said:

Your comment is awaiting moderation.

really cool!