SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

REMNUX Usage Tips for Malware Analysis on Linux

This cheat sheet outlines some of the commands and tools for analyzing malware using the REMnux

Get Started with REMnux

Get REMnux as a virtual appliance, install the distro on a dedicated system, or add it to an existing one.

Review REMnux documentation at docs.remnux.org.

Keep your system up to date by periodically running "`remnux upgrade`" and "`remnux update`".

Become familiar with REMnux malware analysis tools available as Docker images.

Know default logon credentials: `remnux/malware`

Operate Your REMnux System

Shut down the system	`shutdown`
Reboot the system	`reboot`
Switch to a root shell	`sudo -s`
Renew DHCP lease	`renew-dhcp`
See current IP address	`myip`
Edit a text file	`code` *`file`*
View an image file	`feh` *`file`*
Start web server	`httpd start`
Start SSH server	`sshd start`

Analyze Windows Executables

Static Properties: manalyze, peframe, pefile, pyew, exiftool, clamscan, pescan, portex, bearcommander

Strings and Deobfuscation: pestr, bbcrack, brxor.py, base64dump, xorsearch, flarestrings, floss, cyberchef

Code Emulation: binee, capa, vivbin

Disassemble/Decompile: ghidra, cutter, objdump, r2

Unpacking: bytehist, de4dot, upx

Reverse-Engineer Linux Binaries

Static Properties: trid, exiftool, pyew, readelf.py

Disassemble/Decompile: ghidra, cutter, objdump, r2

Debugging: edb, gdb

Behavior Analysis: ltrace, strace, frida, sysdig, unhide

Investigate Other Forms of Malicious Code

Android: apktool, droidlysis3.py, androgui.py, baksmali, dex2jar

Java: cfr, procyon, jad, jd-gui, idx_parser.py

Python: pyinstxtractor.py, pycdc

JavaScript: js, js-file, objects.js, box-js

Shellcode: shellcode2exe.bat, scdbg, xorsearch

PowerShell: pwsh, base64dump

Flash: swfdump, flare, flasm, swf_mastah.py, xxxswf

Examine Suspicious Documents

Microsoft Office Files: vmonkey, pcodedmp, olevba, xlmdeobfuscator, oledump.py, msoffice-crypt, ssview

RTF Files: rtfobj, rtfdump

Email Messages: emldump, msgconvert

PDF Files: pdfid, pdfparser, pdfextract, pdfdecrypt, peepdf, pdftk, pdfresurrect, qpdf, pdfobjflow

General: base64dump, tesseract, exiftool

Explore Network Interactions

Monitoring: burpsuite, networkminer, polarproxy, mitmproxy, wireshark, tshark, ngrep, tcpxtract, tcpick

Connecting: thug, nc, tor, wget, curl, irc, ssh

Services: fakedns, fakemail, accept-all-ips, nc, httpd, inetsim, fakenet, sshd, myip

Gather and Analyze Data

Network: Automater.py, shodan, ipwhois_cli.py, pdnstool

Hashes: malwoverview.py, nsrllookup, Automater.py, vt, virustotal-search.py

Files: yara, scalpel, bulk_extractor, ioc_writer

Other: dexray, viper, time-decode.py

Other Analysis Tasks

Memory Forensics: vol.py, vol3, linux_mem_diff.py, aeskeyfind, rsakeyfind, bulk_extractor

File Editing: wxHexEditor, scite, code, xpdf, convert

File Extraction: 7z, unzip, unrar, cabextract

Use Docker Containers for Analysis

Thug Honeyclient: remnux/thug

JSDetox JavaScript Analysis: remnux/jsdetox

Rekall Memory Forensics: remnux/recall

RetDec Decompiler: remnux/retdec

Radare2 Reversing Framework: remnux/radare2

REMnux in a Container: remnux/remnux-distro

Interact with Docker Images

List local images	`docker images`
Update local image	`docker pull` *`image`*
Delete local image	`docker rmi` *`imageid`*
Delete unused resources	`docker system prune`
Open a shell inside a transient container	`docker run --rm -it` *`image`* `bash`
Map a local TCP port 80 to container's port 80	`docker run --rm -it -p 80:80` *`image`* `bash`
Map your current directory into container	`docker run --rm -it -v .:`*`dir image`* `bash`

Authored by Lenny Zeltser for REMnux v7. Lenny writes a security blog at zeltser.com and is active on Twitter as @lennyzeltser. Many REMnux tools and techniques are discussed in the Reverse-Engineering Malware course at SANS Institute, which Lenny co-authored.