SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

SANS DFIR Linux Distributions:

SIFT Workstation & REMnux

POSTER

digital-forensics.sans.org

$25.00
DFPS_FOR610_v2.2_01-21

Poster Created by Lenny Zeltser and Rob Lee
with support of the SANS DFIR Faculty

SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

@sansforensics

sansforensics

dfir.to/MAIL-LIST

OPERATING SYSTEM & DEVICE IN-DEPTH

- FOR308 **Digital Forensics Essentials**
- FOR498 **Battlefield Forensics & Data Acquisition** GBFA
- FOR500 **Windows Forensic Analysis** GCFE
- FOR518 **Mac and iOS Forensic Analysis & Incident Response**
- FOR585 **Smartphone Forensic Analysis In-Depth** GASF

INCIDENT RESPONSE & THREAT HUNTING

- FOR508 **Advanced Incident Response, Threat Hunting & Digital Forensics** GCFA
- FOR572 **Advanced Network Forensics: Threat Hunting, Analysis & Incident Response** GNFA
- FOR578 **Cyber Threat Intelligence** GCTI
- FOR610 **REM: Malware Analysis Tools & Techniques** GREM
- SEC504 **Hacker Tools, Techniques, Exploits & Incident Handling** GCIH

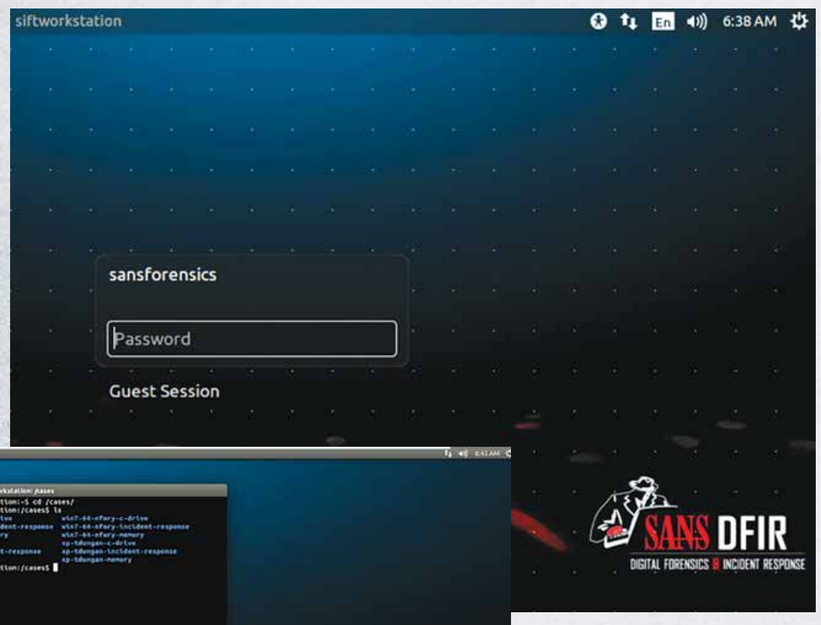
SANS DFIR Linux Distributions:

SIFT Workstation & REMnux

SANS faculty members maintain two popular Linux distributions for performing digital forensics and incident response (DFIR) work. SIFT Workstation,™ created by Rob Lee, is a powerful toolkit for examining forensic artifacts related to file system, registry, memory, and network investigations. REMnux,® created by Lenny Zeltser, focuses on malware analysis and reverse-engineering tasks. These freely available toolkits can be combined on a single host to create the ultimate forensication machine.

SIFT Workstation

An international team of forensics experts created the SIFT Workstation™ for incident response and digital forensics-use and made it available to the community as a public service. The free SIFT toolkit can match any modern incident response and forensic tool suite. It demonstrates that advanced incident response capabilities and deep-dive digital forensic techniques can be accomplished using cutting-edge open-source tools that are freely available and frequently updated.



The SIFT workstation contains hundreds of free and open-source tools that can be used for digital forensics and incident response. Many of the tools and associated analysis techniques are taught in the following courses at SANS:

FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting

FOR572: Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response

FOR578: Cyber Threat Intelligence

REMnux

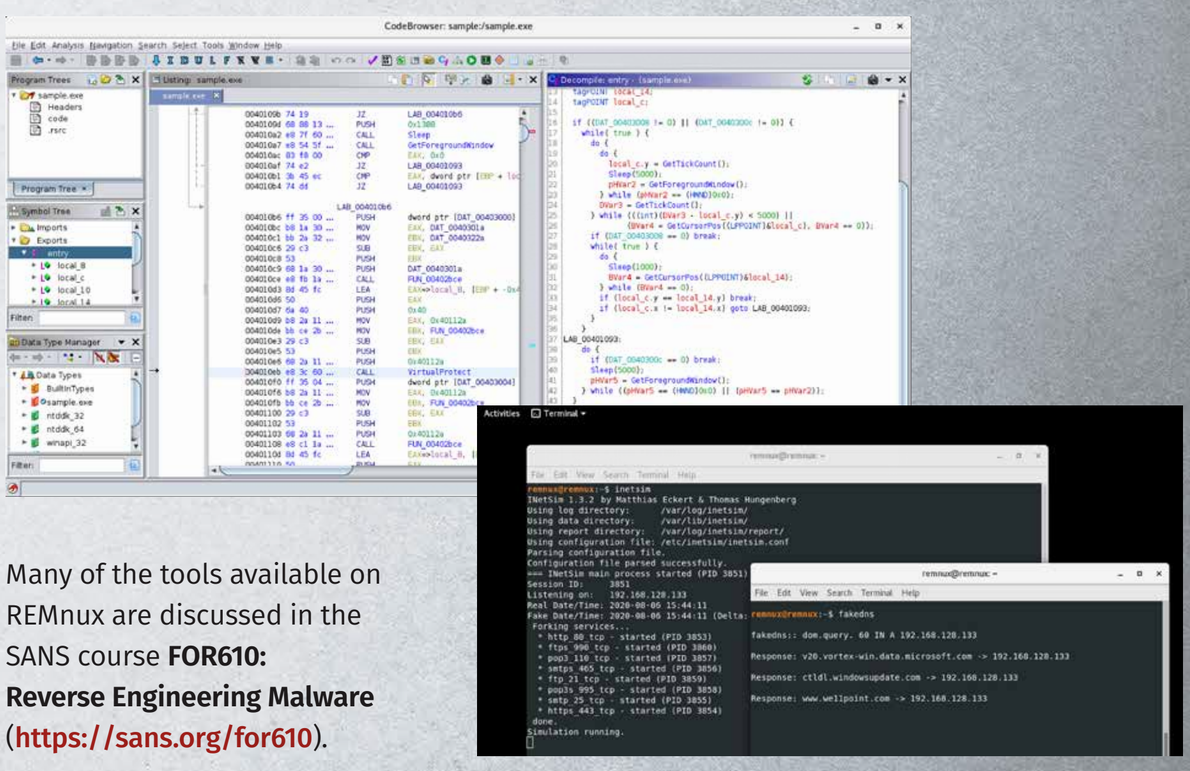
REMnux® is a Linux toolkit for reverse-engineering and analyzing malicious software. REMnux provides a curated collection of free tools created by the community. Analysts can use it to investigate malware without having to find, install, and configure the tools.

The heart of the project is the REMnux Linux distribution based on Ubuntu, which incorporates many tools that malware analysts use to:

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

The REMnux project also offers Docker images of popular malware analysis tools, making it possible to run them as containers without having to install the tools directly on the system.

For details about the tools included with REMnux, see **https://docs.remnux.org**.



Many of the tools available on REMnux are discussed in the SANS course **FOR610: Reverse Engineering Malware** (**https://sans.org/for610**).

The easiest way to get the SIFT Workstation and REMnux distros is to download the corresponding virtual appliance from:

SIFT Workstation: https://sansurl.com/sift-download | REMnux: https://remnux.org

Alternatively, you can set up SIFT and REMnux systems from scratch using the tools' installers, as described in their documentation.

You can even install SIFT Workstation and REMnux on a single system to create a forensics and malware analysis super-toolkit, as described here:

https://sansurl.com/sift-remnux-combo

REMnux Usage Tips for Malware Analysis on Linux

This reference outlines the tools and commands for analyzing malware using the REMnux Linux distribution.
To print a small one-page version; visit:
https://zeltser.com/remnux-malware-analysis-tips

Operate Your REMnux System

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

Analyze Windows Executables

Static Properties:

- **manalyze** https://github.com/JusticeRage/Manalyze
- **peframe** https://github.com/guelfoweb/peframe
- **pefile** https://github.com/erocarrera/pefile
- **pyew** https://github.com/joxeankoret/pyew
- **exiftool** https://exiftool.org
- **clamscan** https://www.clamav.net
- **pescan** http://pev.sourceforge.net
- **portex** https://github.com/katjahahn/PortEx
- **bearcommander** https://github.com/hasherezade/bearparser/wiki
- **pecheck** https://docs.remnux.org/discover-the-tools/examine+static+properties/pe+files#pecheck

Strings and Deobfuscation:

- **pestr** http://pev.sourceforge.net
- **bbcrack** https://github.com/decalage2/balbuzard
- **brxor.py** https://github.com/REMnux/distro/blob/master/files/brxor.py
- **base64dump** https://blog.didierstevens.com/2020/07/03/update-base64dump-py-version-0-0-12/
- **xorsearch** https://blog.didierstevens.com/programs/xorsearch/
- **flarestrings** https://github.com/fireeye/stringsifter
- **floss** https://github.com/fireeye/flare-floss
- **cyberchef** https://github.com/gchq/CyberChef/

Code Emulation:

- **binee** https://github.com/carbonblack/binee
- **capa** https://github.com/fireeye/capa
- **vivbin** https://github.com/vivisect/vivisect

Disassemble/Decompile:

- **ghidra** https://ghidra-sre.org
- **cutter** https://cutter.re
- **objdump** https://en.wikipedia.org/wiki/Objdump
- **r2** https://www.radare.org/n/radare2.html

Unpacking:

- **bytehist** https://www.cert.at/en/downloads/software/software-bytehist
- **de4dot** https://github.com/0xd4d/de4dot
- **upx** https://upx.github.io

Investigate Other Forms of Malicious Code

Android:

- **apktool** https://ibotpeaches.github.io/Apktool/
- **droidlysis** https://github.com/cryptax/droidlysis
- **androgui.py** https://github.com/androguard/androguard
- **baksmali** https://bitbucket.org/JesusFreke/smali/src/master/
- **dex2jar** https://github.com/pxb1988/dex2jar

Java:

- **cfr** https://www.benf.org/other/cfr/
- **procyon**
- **jad**
- **jd-gui** https://java-decompiler.github.io
- **idx_parser.py** https://github.com/digitalsleuth/Java_IDX_Parser

Python:

- **pyinstxtractor.py** https://github.com/extremecoders-re/pyinstxtractor
- **pycdc** https://github.com/zrax/pycdc

JavaScript:

- **js** https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey
- **js-file** https://blog.didierstevens.com/2018/04/19/update-patched-spidermonkey/
- **objects.js** https://github.com/REMnux/salt-states/blob/master/remnux/config/objects/objects.js
- **box-js** https://github.com/CapacitorSet/box-js

Shellcode:

- **shellcode2exe.bat** https://github.com/repnz/shellcode2exe
- **scdbg** http://sandsprite.com/blogs/index.php?uid=7&pid=152
- **xorsearch** https://blog.didierstevens.com/programs/xorsearch/

PowerShell:

- **pwsh** https://github.com/powershell/powershell
- **base64dump** https://blog.didierstevens.com/2020/07/03/update-base64dump-py-version-0-0-12/

Flash:

- **swfdump** http://swftools.org
- **flare** http://www.nowrap.de/flare.html
- **flasm** http://www.nowrap.de/flasm.html
- **swf_mastah.py** https://github.com/9b/pdfxray_lite
- **xxxswf** https://github.com/viper-framework/xxxswf

Examine Suspicious Documents

Microsoft Office Files:

- **vmonkey** https://www.decalage.info/en/vba_emulation
- **pcodedmp** https://github.com/bontchev/pcodedmp
- **olevba** http://www.decalage.info/python/oletools
- **xlmdeobfuscator** https://github.com/DissectMalware/XLMMacroDeobfuscator
- **oledump.py** https://blog.didierstevens.com/programs/oledump-py/
- **msoffice-crypt** https://github.com/herumi/msoffice
- **ssview** https://www.mitec.cz/ssv.html

RTF Files:

- **rtfobj** http://www.decalage.info/python/oletools
- **rtfdump** https://blog.didierstevens.com/2018/12/10/update-rtfdump-py-version-0-0-9/

Email Messages:

- **emldump** https://blog.didierstevens.com/2017/07/21/update-emldump-py-version-0-0-10/
- **msgconvert** https://www.matijs.net/software/msgconv/

PDF Files:

- **pdfid** https://blog.didierstevens.com/programs/pdf-tools/
- **pdfparser** https://blog.didierstevens.com/programs/pdf-tools/
- **pdfextract** https://github.com/gdelugre/origami
- **pdfdecrypt** https://github.com/jancschaefer/PDFDecrypt
- **peepdf** https://eternal-todo.com/tools/peepdf-pdf-analysis-tool
- **pdftk** https://gitlab.com/pdftk-java/pdftk
- **pdfresurrect** https://github.com/enferex/pdfresurrect
- **qpdf** http://qpdf.sourceforge.net
- **pdfobjflow** https://bitbucket.org/sebastiendamaye/pdfobjflow/src/master/

General:

- **base64dump** https://blog.didierstevens.com/2020/07/03/update-base64dump-py-version-0-0-12/
- **tesseract** https://github.com/tesseract-ocr/tesseract
- **exiftool** https://exiftool.org

Reverse-Engineer Linux Binaries

Static Properties:

- **trid** https://mark0.net/soft-trid-e.html
- **exiftool** https://exiftool.org
- **pyew** https://github.com/joxeankoret/pyew
- **readelf.py** https://github.com/eliben/pyelftools

Disassemble/Decompile:

- **ghidra** https://ghidra-sre.org
- **cutter** https://cutter.re
- **objdump** https://en.wikipedia.org/wiki/Objdump
- **r2** https://www.radare.org/n/radare2.html

Debugging:

- **edb** https://github.com/eteran/edb-debugger
- **gdb** https://www.sourceware.org/gdb/

Behavior Analysis:

- **ltrace** https://ltrace.org
- **strace** https://strace.io
- **frida** https://frida.re
- **sysdig** https://github.com/draios/sysdig
- **unhide** http://www.unhide-forensics.info

Explore Network Interactions

Monitoring:

- **burpsuite** https://portswigger.net
- **networkminer** https://www.netresec.com
- **polarproxy** https://www.netresec.com/?page=PolarProxy
- **mitmproxy** https://mitmproxy.org
- **wireshark** https://www.wireshark.org
- **tshark** https://linux.die.net/man/1/tshark
- **ngrep** https://github.com/jpr5/ngrep
- **tcpxtract** http://tcpxtract.sourceforge.net
- **tcpick** http://tcpick.sourceforge.net

Connecting:

- **thug** https://github.com/buffer/thug
- **nc** https://nc110.sourceforge.io
- **tor** https://www.torproject.org
- **wget** https://www.gnu.org/software/wget/
- **curl** https://curl.haxx.se
- **irc** http://www.epicsol.org
- **ssh** https://man.openbsd.org/ssh.1
- **unfurl** https://github.com/obsidianforensics/unfurl

Services:

- **fakedns** https://code.activestate.com/recipes/491264-mini-fake-dns-server/
- **fakemail** https://hg.sr.ht/~olly/fakemail
- **accept-all-ips** https://github.com/REMnux/distro/blob/master/files/accept-all-ips
- **nc** https://nc110.sourceforge.io
- **httpd** https://nginx.org
- **inetsim** https://www.inetsim.org
- **fakenet** https://github.com/fireeye/flare-fakenet-ng
- **sshd** https://man.openbsd.org/sshd.8
- **myip** https://github.com/REMnux/distro/blob/master/files/myip

Gather and Analyze Data

Network:

- **Automater.py** http://www.tekdefense.com/automater/
- **shodan** https://github.com/achillean/shodan-python/
- **ipwhois_cli.py** https://github.com/secynic/ipwhois
- **pdnstool** https://github.com/chrislee35/passivedns-client

Hashes:

- **malwoverview.py** https://github.com/digitalsleuth/malwoverview
- **nsrllookup** https://github.com/rjhansen/nsrllookup
- **Automater.py** http://www.tekdefense.com/automater/
- **vt** https://github.com/doomedraven/VirusTotalApi
- **virustotal-search.py** https://blog.didierstevens.com/programs/virustotal-tools

Files:

- **yara** https://virustotal.github.io/yara/
- **scalpel** https://github.com/sleuthkit/scalpel
- **bulk_extractor** https://github.com/simsong/bulk_extractor/
- **ioc_writer** https://github.com/mandiant/ioc_writer

Other:

- **dexray** http://www.hexacorn.com/blog/category/software-releases/dexray/
- **viper** https://github.com/viper-framework/viper
- **time-decode.py** https://github.com/digitalsleuth/time_decode

Other Analysis Tasks

Memory Forensics:

- **vol.py**
- **vol3** https://github.com/volatilityfoundation/volatility3
- **linux_mem_diff.py** https://github.com/monnappa22/linux_mem_diff_tool
- **aeskeyfind** https://github.com/makomk/aeskeyfind
- **rsakeyfind** https://packages.debian.org/jessie/utils/rsakeyfind
- **bulk_extractor** https://github.com/simsong/bulk_extractor/

File Editing:

- **wxHexEditor** https://sourceforge.net/projects/wxhexeditor/
- **scite** https://www.scintilla.org/SciTE.html
- **code** https://code.visualstudio.com
- **xpdf** http://www.xpdfreader.com
- **convert** https://imagemagick.org

File Extraction:

- **7z** https://www.7-zip.org
- **unzip** http://infozip.sourceforge.net
- **unrar** https://www.rarlab.com
- **cabextract** https://www.cabextract.org.uk

Use Docker Containers for Analysis

- **Thug Honeyclient:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#thug
- **JSDetox JavaScript Analysis:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#jsdetox
- **Rekall Memory Forensics:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#rekall
- **RetDec Decompiler:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#retdec
- **Radare2 Reversing Framework:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#radare2
- **Ciphey Automatic Decrypter:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#ciphey
- **Viper Binary Analysis Framework:** https://docs.remnux.org/run-tools-in-containers/remnux-containers#viper-binary-analysis-and-management-framework
- **REMnux distro in a Container:** https://docs.remnux.org/install-distro/remnux-as-a-container

Interact with Docker Images

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

You can learn the malware analysis techniques that make use of the tools installed and pre-configured on REMnux by taking reverse-engineering malware training at SANS Institute.
https://www.sans.org/for610

SANS DFIR
digital-forensics.sans.org