Forensic Identification of Unique Kali Systems Through the Use of File Hashes and Names

Troy Ward

January 25, 2021

Table of Contents

Table of Tables

Table of Figures

Introduction

The act of attributing a cyber attack is a long sought after, but rarely attained goal. From a legal perspective, the ability to attribute a cyberattack to an individual or a group is a vital step to lead to its ultimate prosecution in a court of law. In terms of national defense, attributing an attack accurately can be a step towards reprisal or other military responses.

While things such as techniques, IP addresses, and other artifacts can plan an important role in attributing an attack, being able to attribute actions to a specific computer that has been obtained can help provide overwhelming proof of an act.

A primary task of a forensic investigator is identifying information that can be used to tie something to a specific individual computer. As we use a computer, it becomes more and more individualized to its user. This occurs through the use and configuration of software, the creation of documents, and the browsing of websites among many other ways. Many of these actions leave artifacts that can uniquely be tied to a single system.

One way to minimize the ability to find artifacts that are unique to a system is to use a freshly installed operating system. This can be done through the use of a live-boot operating system, or just creating a freshly installed system. In theory, there is nothing that differentiates two copies of an operating system if they are installed the same way. In reality, though, there are a small number of artifacts that uniquely identify a system as soon as the operating system is installed.

This paper documents the first of what I expect will be several experiments designed to identify a method to uniquely identify a particular system. The hope is to find artifacts that are resistant to reboot, cloning of virtual machines, or intentional/unintentional modification.

Kali Linux

Kali Linux is a highly popular Linux distribution known for its use by “hackers”. Under the hood, Kali is a highly modified and customized version of Debian which is maintained by Offensive Security[1]. The operating system comes with a large number of tools already installed and configured to allow for use from “out of the box”.

Methodology

This first set of experiments is designed to look for identifiable unique signatures that can be observed by a user with root access to the system. While advanced forensic tools were used during the examination of the system disks, this was only done for ease of use. The techniques used to identify “interesting” files only required tools that are inherent on most versions of Linux to a user with root access.

The factors that I looked at include:

- The presence of uniquely named files

[1] https://www.kali.org/

- The presence of commonly named files with unique MD5 hashes[2]

The experiment made use of four identically installed and configured Kali systems. Each system was initially created on 7/22/2020. The same installation media and procedures were used to create each image. All systems were built using a copy of Kali 2020.2 AMD64 iso with a MD5 hash of 9efa96f1069aaa2fcacd56bcd55330824f290e21. All systems were installed on a virtual machine being hosted on VMWare Player 15.5.6. Each system had 20 GB hard drives, 2 GB of RAM, and 1 CPU core. All default installation values were accepted for each install. Upon the completion of the installation, the system was rebooted, and then immediately turned off once it arrived at the login screen (no systems were logged into).

The system's virtual hard drive was mounted in a read-only configuration to the analysis operating system to allow for a view of the disk without the risk of modifying any data present on it. Once mounted, a script was run that traversed across the entire mounted disk and performed an MD5 hash of all files encountered as well as recording other vital information such as owner/group of the file, permissions, file access, modification, and creation times, and other information. This is all information that any person with root access to the actual device would be able to determine themselves.

The results of the MD5 hashes and file metadata were fed into a Splunk instance which was then used to do the data analysis and identify "interesting" files. To aid in the analysis of individual files, the virtual hard drives were ingested into Autopsy to allow for further indexing and hex level examination of individual files.

Analysis

The initial examination of each system image found a total of 284,996 files on the disk. The files were spread out across six directories (see Table 1). What I find interesting in the table is what isn't here. If you notice, there were no files found within /bin, /sbin, or /lib. It's because if you do a full directory listing of /, you'll find that they all symbolic links going back to /usr/bin, /usr/sbin, and /usr/lib respectively.

Path	File Count
/boot	305
/etc	1362
/home	6
/root	3
/usr	272,819
/var	10,501

Table 1 Count of Files by Directory

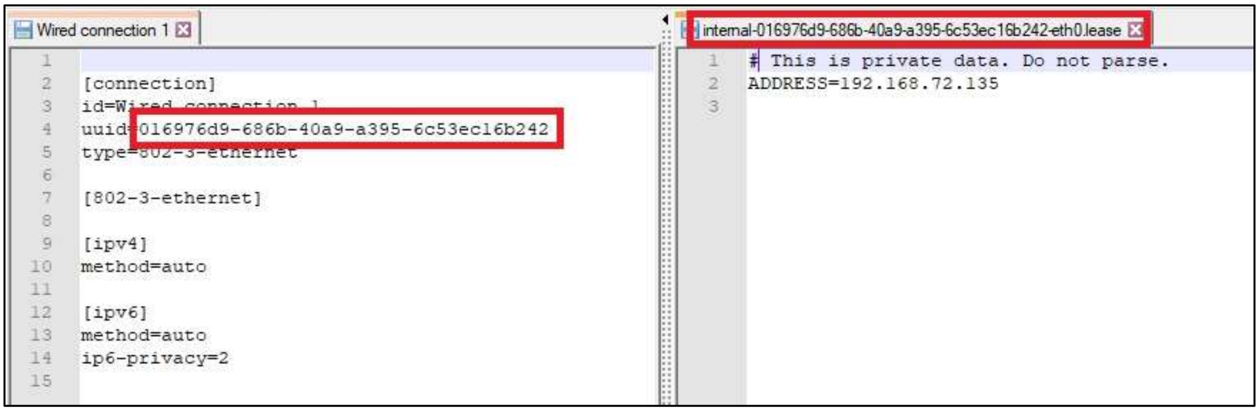
[2] I realize that MD5 is a long since insecure hash in a variety of situations, including in some instances, digital forensics, but given the fact that it is faster to run on my test systems vs. SHA1 and SHA256, and the fact that there is little chance of a file purposely being manipulated to "appear" to be common, I think that this is an acceptable trade-off.

Unique Names

An initial examination of the nearly 285K files found that only 45 files per image (0.01%) had a unique filename[3]. In addition to the 45 unique file names, a single directory on each system image was also found to have a unique name

Eth0.lease

Located within the /var/lib/NetworkManager directory is a file named internal-<UUID>-<Interface ID>.lease (i.e. internal-bdf3e5cc-852d-47f8-b8c3-aaaa604e78d1-eth0.lease). The UID in the name refers to the UID that is assigned to that particular interface in "/etc/NetworkManager/system-connections/Wired connection 1" (see Figure 1).

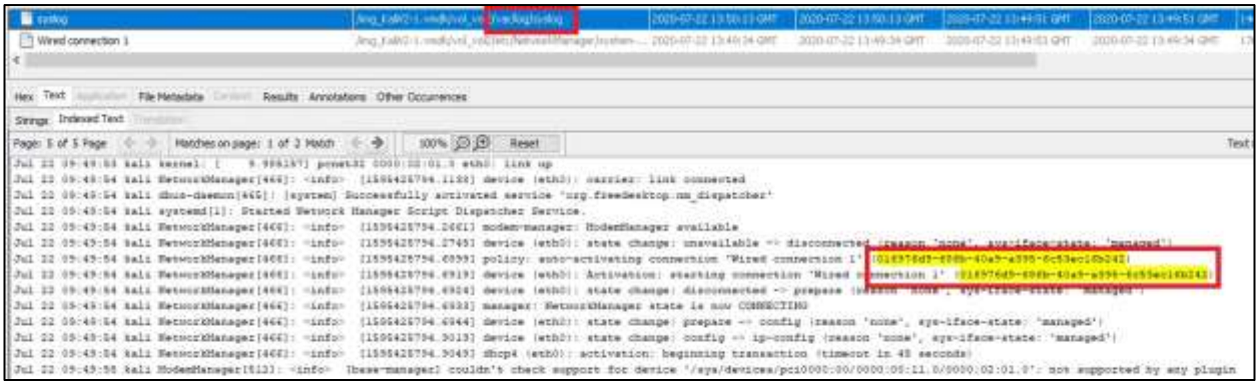


```

[connection]
id=Wired connection 1
uuid=016976d9-686b-40a9-a395-6c53ec16b242
type=802-3-ethernet

[802-3-ethernet]

[ipv4]
method=auto

[ipv6]
method=auto
ip6-privacy=2

```

```
# This is private data. Do not parse.
ADDRESS=192.168.72.135

```

Figure 1 Network Interface Card UID/Lease Comparison

The UID for the NIC makes numerous appearances within /var/log/syslog, as the system boots up (see Figure 2).



```
Jul 22 09:49:53 kali kernel: [    9.998167] pcnet32 0000:02:01.0 eth0: link up
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.1122] device (eth0): carrier: link connected
Jul 22 09:49:54 kali dbus-daemon[465]: [system] Successfully activated service 'org.freedesktop.nm_dispatcher'
Jul 22 09:49:54 kali systemd[1]: Started Network Manager Script Dispatcher Service.
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.2661] modem-manager: ModemManager available
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.2745] device (eth0): state change: unavailable -> disconnected (reason 'none', sys-iface-state: 'managed')
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.6839] policy: auto-activating connection 'Wired connection 1' (016976d9-686b-40a9-a395-6c53ec16b242)
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.6919] device (eth0): Activation: starting connection 'Wired connection 1' (016976d9-686b-40a9-a395-6c53ec16b242)
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.6924] device (eth0): state change: disconnected -> prepare (reason 'none', sys-iface-state: 'managed')
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.6933] manager: NetworkManager state is now CONNECTING
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.6944] device (eth0): state change: prepare -> config (reason 'none', sys-iface-state: 'managed')
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.9013] device (eth0): state change: config -> ip-config (reason 'none', sys-iface-state: 'managed')
Jul 22 09:49:54 kali NetworkManager[466]: <info>  [1595428794.9049] dhcp4 (eth0): activation: beginning transaction (timeout in 45 seconds)
Jul 22 09:49:58 kali ModemManager[513]: <info>  [base-manager] couldn't check support for device '/sys/devices/pci0000:00/0000:00:11.0/0000:02:01.0': not supported by any plugin
```

Figure 2 NIC UID Appearances in Logs

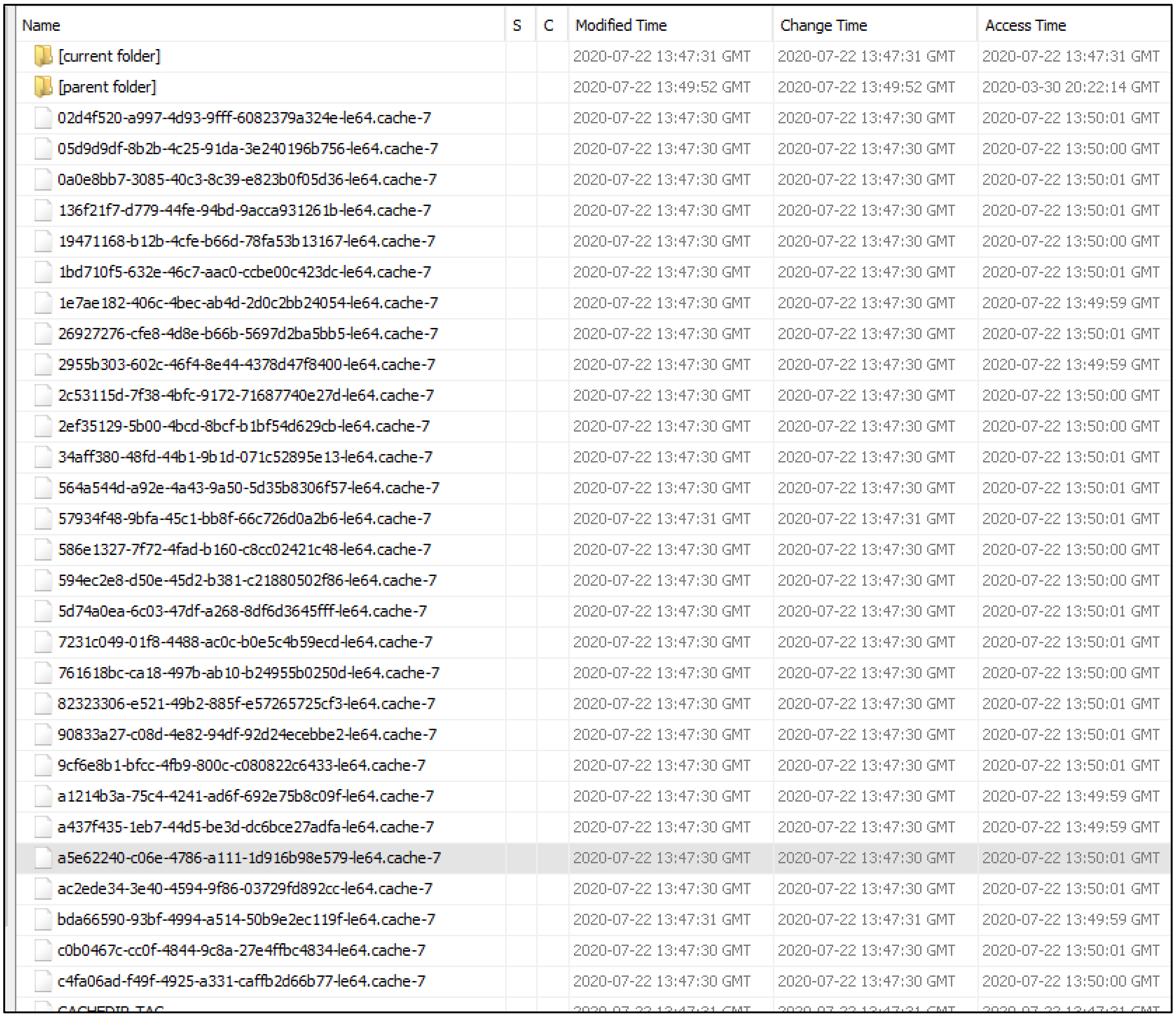
The fact that the name is made up of a UUID that is assigned at the time of creation means that it can be tied to a single system. While it is likely possible to change the UUID present within both the file name and the interface configuration, it would be important to identify all instances where the same UID

[3] For this we consider the combination of path and file name to be a file name

appeared within logs on the system. On a system without remote logging, this would be difficult but possible. Because of this, I consider this to be a moderately confident indicator of a particular system.

Font Config

Of the 45 uniquely named files, 38 of them (84%) were located in the /var/cache/fontconfig/ directory. Each of these files had a name consisting of a 36 character Universally Unique Identifier (UUID) followed by "-le64.cache-7" (i.e. a185396b-1e43-43cd-8ca0-31de187f9b4e-le64.cache-7). The UUID in each file name was different from the other files on the same system (see Figure 3).



Name	S	C	Modified Time	Change Time	Access Time
[current folder]			2020-07-22 13:47:31 GMT	2020-07-22 13:47:31 GMT	2020-07-22 13:47:31 GMT
[parent folder]			2020-07-22 13:49:52 GMT	2020-07-22 13:49:52 GMT	2020-03-30 20:22:14 GMT
02d4f520-a997-4d93-9fff-6082379a324e-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
05d9d9df-8b2b-4c25-91da-3e240196b756-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
0a0e8bb7-3085-40c3-8c39-e823b0f05d36-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
136f21f7-d779-44fe-94bd-9acca931261b-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
19471168-b12b-4cfe-b66d-78fa53b13167-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
1bd710f5-632e-46c7-aac0-ccbe00c423dc-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
1e7ae182-406c-4bec-ab4d-2d0c2bb24054-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:49:59 GMT
26927276-cfe8-4d8e-b66b-5697d2ba5bb5-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
2955b303-602c-46f4-8e44-4378d47f8400-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:49:59 GMT
2c53115d-7f38-4bfc-9172-71687740e27d-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
2ef35129-5b00-4bcd-8bcf-b1bf54d629cb-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
34aff380-48fd-44b1-9b1d-071c52895e13-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
564a544d-a92e-4a43-9a50-5d35b8306f57-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
57934f48-9bfa-45c1-bb8f-66c726d0a2b6-le64.cache-7			2020-07-22 13:47:31 GMT	2020-07-22 13:47:31 GMT	2020-07-22 13:50:01 GMT
586e1327-7f72-4fad-b160-c8cc02421c48-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
594ec2e8-d50e-45d2-b381-c21880502f86-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
5d74a0ea-6c03-47df-a268-8df6d3645fff-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
7231c049-01f8-4488-ac0c-b0e5c4b59ecd-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
761618bc-ca18-497b-ab10-b24955b0250d-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT
82323306-e521-49b2-885f-e57265725cf3-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
90833a27-c08d-4e82-94df-92d24ecebbe2-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
9cf6e8b1-bfcc-4fb9-800c-c080822c6433-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
a1214b3a-75c4-4241-ad6f-692e75b8c09f-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:49:59 GMT
a437f435-1eb7-44d5-be3d-dc6bce27adfa-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:49:59 GMT
a5e62240-c06e-4786-a111-1d916b98e579-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
ac2ede34-3e40-4594-9f86-03729fd892cc-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
bda66590-93bf-4994-a514-50b9e2ec119f-le64.cache-7			2020-07-22 13:47:31 GMT	2020-07-22 13:47:31 GMT	2020-07-22 13:49:59 GMT
c0b0467c-cc0f-4844-9c8a-27e4ffbc4834-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:01 GMT
c4fa06ad-f49f-4925-a331-caffb2d66b77-le64.cache-7			2020-07-22 13:47:30 GMT	2020-07-22 13:47:30 GMT	2020-07-22 13:50:00 GMT

Figure 3 Uniquely named files located in /var/cache/fontfig

Font Config is a library designed to configure and manage various fonts on the system. When an application needs to make use of a font that has been installed on the system, Font Config provides access to the requested font. If the requested font isn't available, Font Config attempts to find a similar one[4]. To speed up system operation, Font Config creates a cache of all fonts installed on the system. These *-le64.cache-7 files are those font caches.

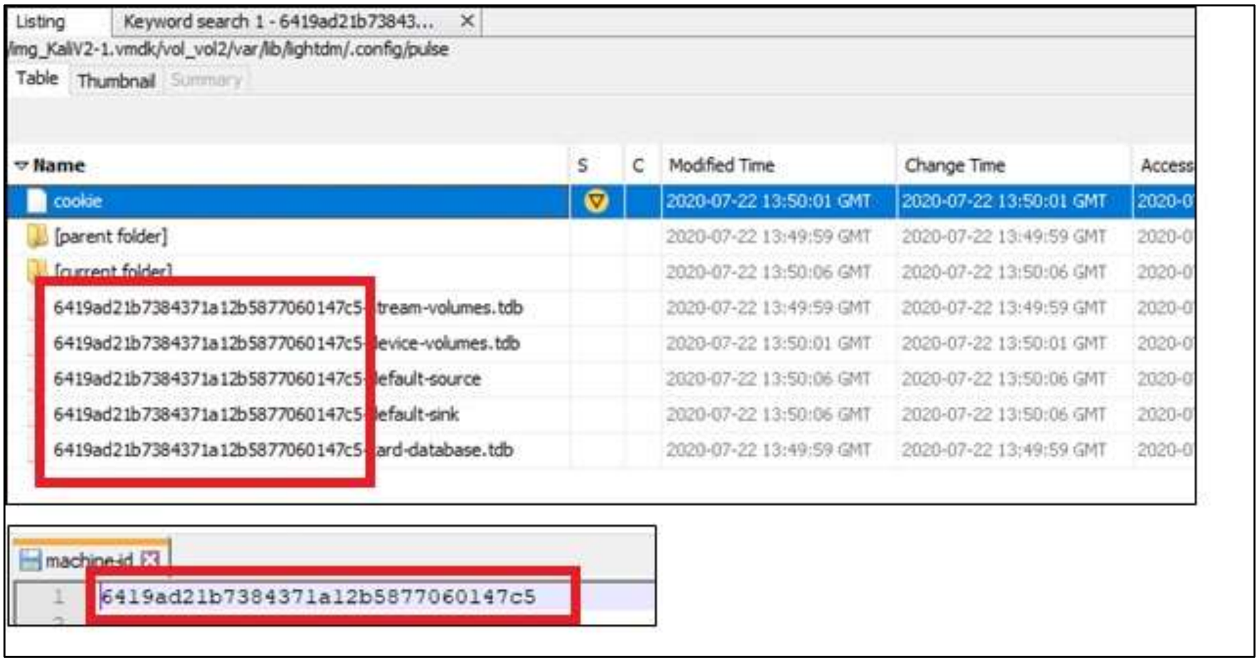
Because these file names are made up of actual UUIDs that are generated at the time of creation, they are unique to that particular system. That being said though because it is a cache, the files can safely be deleted but it will impact system performance. The cache can be regenerated through the fc-cache -f command which will force a cache regeneration[5].

LightDM

Another five of the uniquely named files are located in /var/lib/lightdm/.config/pulse/. These files all consisted of a single 32-character ID[6], followed by one of the following:

- -stream-volumes.tdb
- -device-volumes.tdb
- -default-source
- -default-sink
- -card-database.tdb

An example of this is /var/lib/lightdm/.config/pulse/d94a20e2919f456db65752d2fa3df4d8-stream-volumes.tdb. It is important to note that all five of the files exist on each system. The ID string at the start of each file name is actually the same value found within /etc/machine-id (see Figure 4).



/img_KaliV2-1.vmdk/vol_vol2/var/lib/lightdm/.config/pulse

Name	S	C	Modified Time	Change Time	Access
cookie	▽		2020-07-22 13:50:01 GMT	2020-07-22 13:50:01 GMT	2020-0
[parent folder]			2020-07-22 13:49:59 GMT	2020-07-22 13:49:59 GMT	2020-0
[current folder]			2020-07-22 13:50:06 GMT	2020-07-22 13:50:06 GMT	2020-0
6419ad21b7384371a12b5877060147c5-tream-volumes.tdb			2020-07-22 13:49:59 GMT	2020-07-22 13:49:59 GMT	2020-0
6419ad21b7384371a12b5877060147c5-levice-volumes.tdb			2020-07-22 13:50:01 GMT	2020-07-22 13:50:01 GMT	2020-0
6419ad21b7384371a12b5877060147c5-lefault-source			2020-07-22 13:50:06 GMT	2020-07-22 13:50:06 GMT	2020-0
6419ad21b7384371a12b5877060147c5-lefault-sink			2020-07-22 13:50:06 GMT	2020-07-22 13:50:06 GMT	2020-0
6419ad21b7384371a12b5877060147c5-ard-database.tdb			2020-07-22 13:49:59 GMT	2020-07-22 13:49:59 GMT	2020-0

```
6419ad21b7384371a12b5877060147c5
```

Figure 4 /etc/machine-id and /var/lib/lightdm/.configure/pulse comparison

[4] https://en.wikipedia.org/wiki/Fontconfig

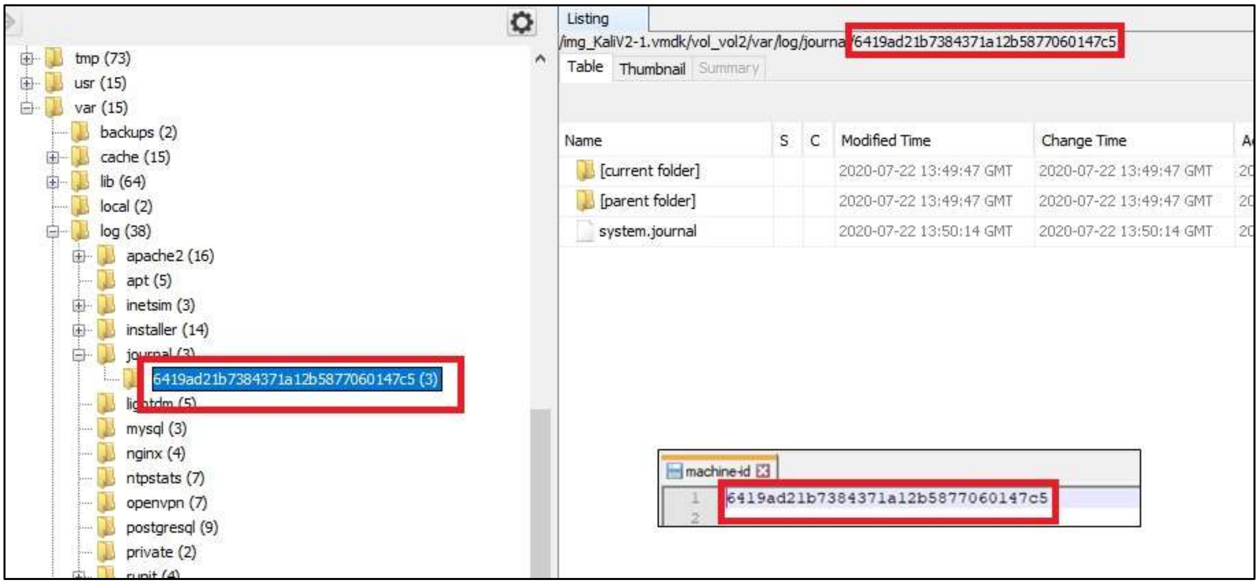
[5] https://linux.die.net/man/1/fc-cache

[6] Note: The ID string presented is similar to a UUID but does not meet the exact technical requirements to be considered on. See https://www.itu.int/en/ITU-T/asn1/Pages/UUID/uuids.aspx

LightDM is a cross-platform display manager that provides the initial login and then the desktop for the Kali workstation[7]. LightDM is one of several choices but is currently the default installed. The pulse in the pathname refers to the Pulse Audio capability that it offers

System.Journal

Another uniquely named file is actually a unique folder. Located at /var/log/journal/<UID> is a folder that contains the system journal. Once again, the UID found in the folder name is the same UID found in /etc/machine-id (see Figure 5). Within the folder is the system journal which records all significant events related to the system.



/img_KaliV2-1.vmdk/vol_vol2/var/log/journa/6419ad21b7384371a12b5877060147c5

Name	S	C	Modified Time	Change Time	A
[current folder]			2020-07-22 13:49:47 GMT	2020-07-22 13:49:47 GMT	20
[parent folder]			2020-07-22 13:49:47 GMT	2020-07-22 13:49:47 GMT	20
system.journal			2020-07-22 13:50:14 GMT	2020-07-22 13:50:14 GMT	20

```
6419ad21b7384371a12b5877060147c5
```

Figure 5 /etc/machine-id vs /var/log/journal/<UID> Comparison

While it is possible to delete the folder and it's contents, the folder will automatically be regenerated (with the same UID) either immediately or upon the restart of Systemd service[8].

Hashes

As noted above, each system image contained 284,996 files, each with its own MD5 hash. Of those, there were a total of 267,192 unique hashes. Of those, 7,880 hashes were repeated two or more times meaning that approximately 90% of the files were unique across the disk. Table 2 shows the 20 most repeated hashes that were found on the drive image, along with one of the files that used that hash.

[7] https://wiki.debian.org/LightDM
[8] https://www.freedesktop.org/software/systemd/man/systemd-journald.service.html

File Hash	Total Files	Example File
d41d8cd98f00b204e9800998ecf8427e	1386	/usr/lib/llvm-9/build/utils/lit/lit/builtin_commands/__init__.py
68b329da9893e34099c7d8ad5cb9c940	490	/usr/lib/python3/dist-packages/filedepot-0.5.2.egg-info/not-zip-safe
b3c40037edbaf26a9bffb1a3d4cf734e	207	/usr/lib/python3/dist-packages/OpenGL/GL/APPLE/__init__.py
5e1d07d82002f02a87f0f26be2d59c7b	193	/var/lib/dpkg/info/libksba8:amd64.triggers
2922a1ff6ca1e589db20b56a1b856709	122	/usr/lib/python3/dist-packages/plotly/validators/layout/updatemenu/font/__init__.py
1d35517f12a3c14d7ec65e84bc0dde06	106	/usr/share/help/eu/mate-calc/legal.xml
e0d0fd012f6293f533701effcc6256fe	99	/usr/lib/python3/dist-packages/plotly/graph_objs/ohlc/hoverlabel/__init__.py
650ba8a15be160b30bf3bd05b5e29293	93	/var/lib/dpkg/info/libvisual-0.4-0:amd64.triggers
feebc3772e647be7c2fffe255fc99cfa	79	/usr/share/help/kab/atril/legal.xml
2744d2bc4adbbf6a7ccd67d74bd9991f	76	/usr/lib/python3/dist-packages/plotly/validators/scatter3d/hoverlabel/font/__init__.py
93c3413b07bba9d38750255de2755431	66	/var/lib/dpkg/info/libproxy1v5:amd64.triggers
ad93072740c556aa6a8d73084aeff2ca	55	/var/lib/dpkg/info/libprotobuf23:amd64.triggers
ef758398051ffbc2baca272f62f4c6dd	53	/var/lib/dpkg/info/libi2c0:amd64.triggers
ff49aa304d48c6a7c101de3135d66052	51	/usr/lib/python3/dist-packages/faraday/utils/__init__.py
4d17ffa4ebca8d702245a276a1cfe2a0	48	/usr/lib/python3/dist-packages/plotly/validators/layout/xaxis/tickformatstop/__init__.py
3c124a19ee0e1348068465cbac8387b7	47	/usr/lib/python3/dist-packages/plotly/validators/scatter3d/stream/__init__.py
1bd16ddfbfdcc942da8f98f69966e6fe	44	/usr/lib/python3/dist-packages/plotly/graph_objs/choropleth/colorbar/__init__.py
0ec9da0a97e8e2f03542d8acfb4fcc7b	42	/usr/lib/python3/dist-packages/plotly/validators/scatter3d/hoverlabel/__init__.py
37802a7cc55bef61ab6453d144b186c6	39	/usr/share/icons/Flat-Remix-Blue-Dark/panel/weather-clear-night-260.svg
fcbe8b2739c9b6ffae4de75d6f462464	39	/usr/share/icons/Flat-Remix-Blue-Light/panel/weather-clear-night-260.svg

Table 2 Top 20 Repeated Hashes

It stands to reason that one of two things will be true for every file that is included on all four system images (This is excluding the 45 files that had unique names). The first possibility is that the same file will have the same hash on all four systems. If the file is being copied over from the install image or downloaded from a repository, then there is no reason it would change between images (unless that package is updated). The other possibility is that the same file will have a unique hash for each of the four images. This is likely to occur when the file is generated as part of the install instead of being copied onto the system.

Out of the 284,951 common files found on all four system images, only 281 of them did not share a common hash across all four images. This means that each installation is 99.9% the same. Of the 281 files that did not have a consistent hash, 202 of them had a unique hash on each of the drive images. For the remainder of this paper, they will be referred to as "Always Unique" files.

Many of these Always Unique files are not at all surprising. For example, 22 (9%) of the files live in /var/log. The data included in log files for two different systems is almost guaranteed to be different (at the very least the times are likely to be different) which will cause the file to produce a unique hash. As a side note, it is important to remember that even if an image is copied, the chances are that the hash of a log file will change very quickly once the system is in use again.

As I said, I expected for a file either to be the same across all four images or to be unique across all four images. When I began looking at the data though, I found 79 files that had either two or three different hashes instead of one or four as expected. This got my attention. We will call these “semi-unique” hashes. In the following sections, we’ll examine each file or set of files that presents a unique or semi-unique hash.

/boot

Initrd.img

Initrd (initial ramdisk) files are used by the system at the initial boot. They are responsible for loading a temporary file system into memory which is then used to mount to the true root file system so booting can continue[9]. Because of the vast number of different types of hardware that are available for a computer, this image is stripped down to the bare essentials just to get the system to a point where it can load the much large and fully configured kernel.

The initrd.img files are created when the operating system is installed and are statically compiled to that particular system. This means that the initrd.img file is unique to the system. Each system image contained an initrd.img file at both /boot/initrd.img-5.5.0-kali2-amd64 and /boot/initrd.img-5.7.0-kali1-amd64. The two files themselves are also different from each other.

Because these files are compiled specifically to this machine, they will always present a unique signature meaning that they can be tied to a specific machine. There are indications that these files can be regenerated after the system has already been installed, but this is a process that I have not personally tried and do not know if it will even work, let alone generate a new hash for the file[10].

grub.cfg

/boot/grub/grub.cfg is a 245-line file that is generated near the end of the installation of Kali. The file contains the configuration for the Grub bootloader that runs immediately when the computer is started. It is responsible for loading and then transferring control of the system to the desired operating system’s kernel[11]. The file is generated using the **grub-mkconfig** command.

Of the 245 lines within the file, 27 of them are unique. Each of the 27 lines that are different contain the UUID of the file system installed by Kali. This UUID is the same on all 27 lines within the config but different between each file image (see Figure 6). This UUID number can also be found in /etc/fstab as described below.

[9] https://wiki.debian.org/Initrd

[10] https://linoxide.com/linux-how-to/fixing-broken-initrd-image-linux/

[11] https://www.gnu.org/software/grub/manual/grub/grub.html

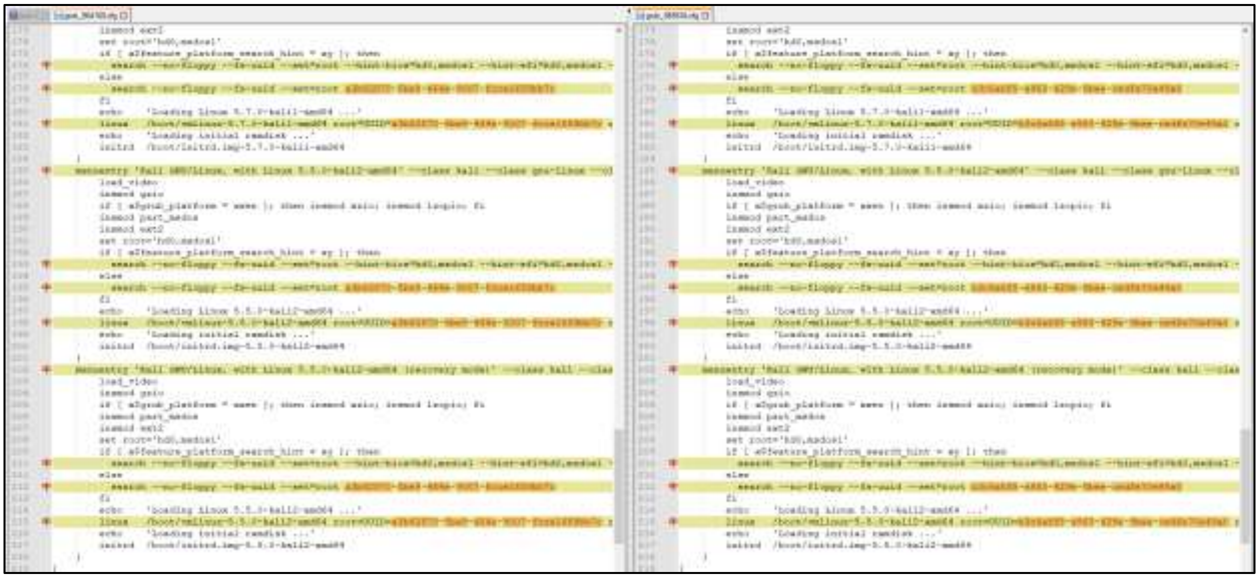


Figure 6 /boot/grub/grub.cfg Differences

Because the UUID of each file system is by definition, unique, it is tied to a specific system. Additionally, because this UUID is tied directly to the UUID of the boot partition, it is found through the disk and in various logs and cannot easily be changed.

/etc

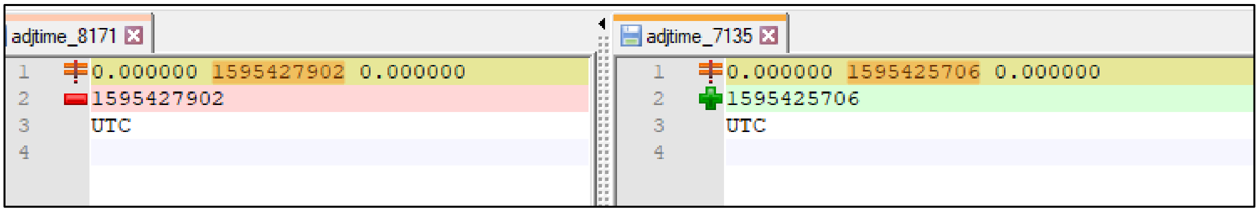
adjtime

The file /etc/adjtime tracks the amount of drift between the hardware clock of the system and the actual time. In theory, the hardware clock gains/loses the same amount of time each day. By recording the amount of draft that occurs each day, and then determining how many days have passed since that time was recorded, the system can figure out the approximate true system time.

The file consists of 3 lines (see Figure 7). The 1st line has three space-separated fields. The first field is the drift rate or the amount of drift the clock has each day. The second field contains the epoch time of the last adjustment. The final field is the last adjustment status which is always set to 0 now for backward compatibility. The 2nd line is the epoch time of the last calibration and should generally be the same as field two on the 1st line. The 3rd line is set to the time zone.

On a physical server, the amount of drift between two systems has a high likelihood of being unique, however, there is a good chance that multiple virtual machines on the same physical server will hold the same drift rate. Even though the drift rate may be the same across virtual machines though, it is highly unlikely that the time of the last adjustment will be the same, even across virtual machines[12].

[12] https://man7.org/linux/man-pages/man5/adjtime.5.html



```
adjtime_8171                              adjtime_7135
1  0.000000 1595427902 0.000000           1  0.000000 1595425706 0.000000
2  1595427902                             2  1595425706
3  UTC                                    3  UTC
4                                         4
```

Figure 7 /etc/adjtime differences

While there is a high likelihood that epoch time of the last time adjustment makes /etc/adjtime unique to a particular system, the fact that this file is simple flat text and a change to any of the fields within the file will likely result in no impact to the system's operation. This means that this file is a poor indicator of system uniqueness.

cacerts

The file /etc/ssl/certs/java/cacerts contains all of the certificate authorities that java trusts by default on the system. This file is separate from what the system's web browser uses. An examination of the file shows that each system's copy is 150,007 bytes in length and that while there are large sections of the file that are almost identical, there are also portions with significant differences.

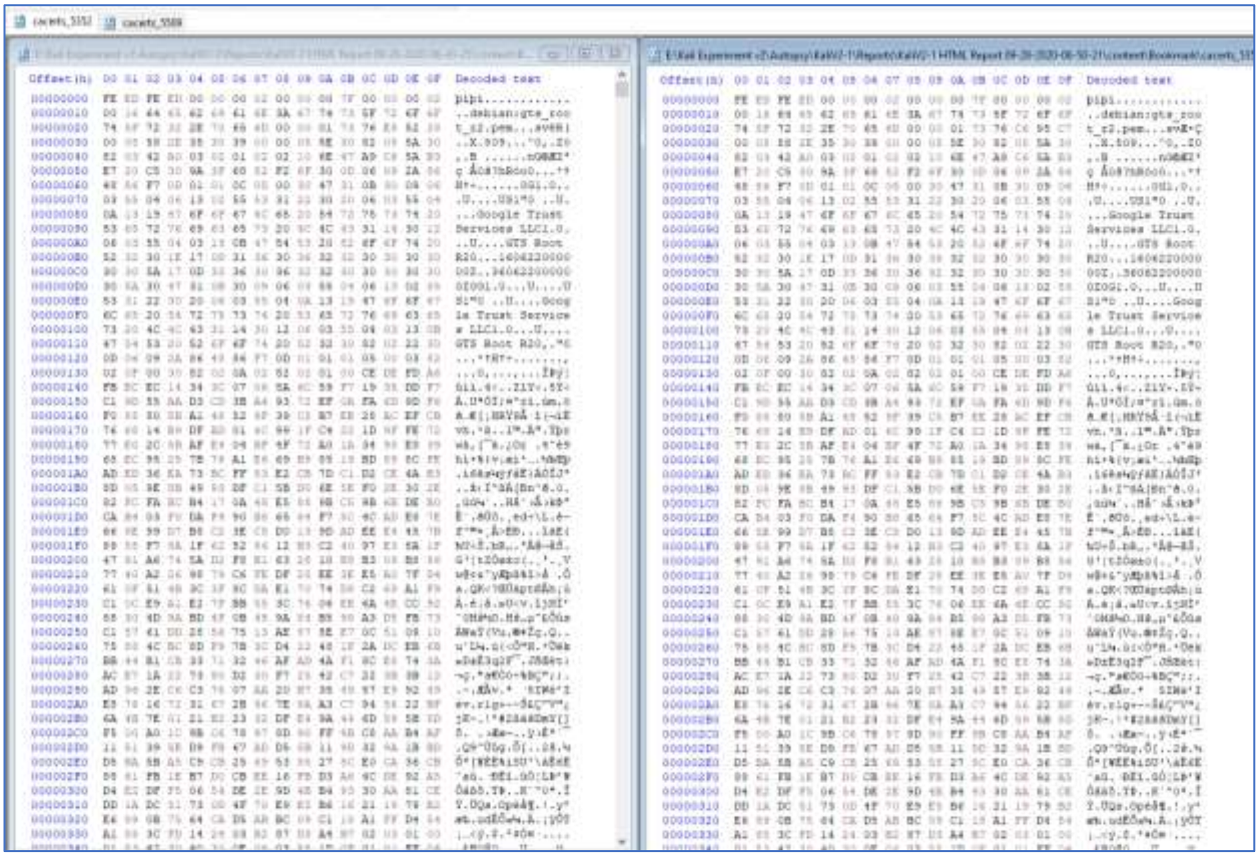


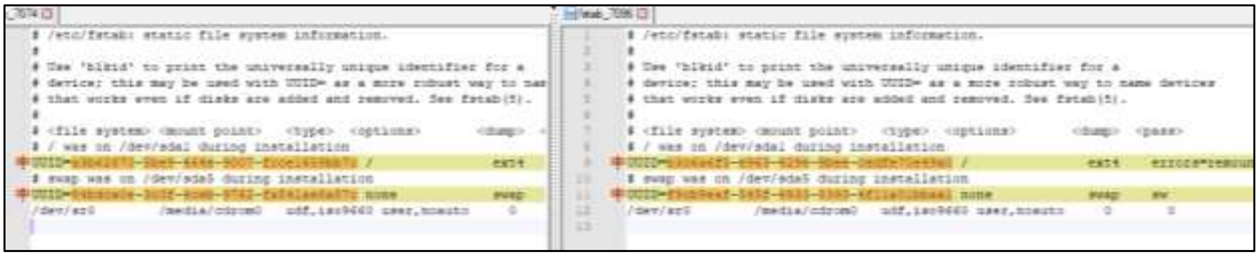
Figure 8 File Comparission of /etc/ssl/certs/java/cacerts

I believe that the reason why each of these is different is that each system has several self-signed certificates that are created as part of the system install. I believe that one or more of these certificates is included within this file which causes each one to be unique.

While this file appears to be unique to a single system, it is easy enough to change this simply by creating a new certificate and then updating the file with the update-ca-trust command[13]. This means that this file is likely a poor indicator of system uniqueness.

fstab

/etc/fstab contains all of the information that the operating system needs to identify the various drive partitions and other information within the file system and mount them. Each line contains multiple fields that identify file systems connected to the system and how they are configured. The first field on each line describes the individual block device. The preferred method to identify these devices is through its UUID, which is how Kali does it by default[14]. There will always be at least one device listed here (the / partition) and in our tests, there was another listed (/dev/sr0) which is the CD-ROM that was used to install the system (see Figure 9). Because the UUID included within the file is by definition, supposed to be unique, this file is tied to a particular system.



```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to nam
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <
# / was on /dev/sda1 during installation
UUID=[illegible] /               ext4
# swap was on /dev/sda5 during installation
UUID=[illegible] none            swap
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0
```

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=[illegible] /               ext4    errors=remoun
# swap was on /dev/sda5 during installation
UUID=[illegible] none            swap    sw
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

Figure 9 /etc/fstab differences

Also within /etc/fstab is a reference to the boot partition (in some cases, this could be / partition noted above). Regardless of what the boot partition is, this same partition UUID is also referenced in /boot/grub as noted above.

Because of the multiple UUIDs within the file, /etc/fstab is unique to the system. Because these UUIDs are tied to the file system partitions they are not very easy to change. Additionally, these UUIDs appear in /boot/grub and multiple logs making this file a very good unique system indicator.

machine-id

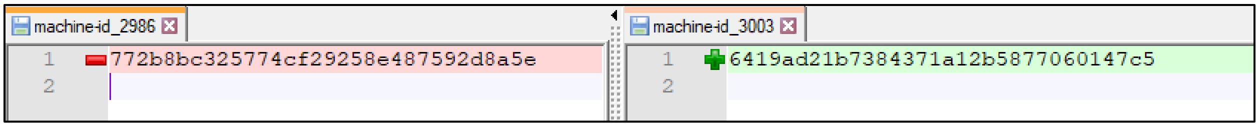
Located at both /etc/machine-id and /var/lib/dbus/machine-id is a file that simply contains a 32-character (128-bit) hexadecimal number that uniquely identifies this particular installation of the operating system (see Figure 10)[15]. This number is unique to this individual installation and does not change if the install is moved to a new physical device or if the IP address or hostname is changed[16].

[13] https://www.systutorials.com/docs/linux/man/8-update-ca-trust/

[14] https://man7.org/linux/man-pages/man5/fstab.5.html

[15] It is important to note that while this number is not a true UUID (by the definition) it serves the same role and is considered to be equivalent to a UUID.

[16] https://wiki.debian.org/MachineId



```
772b8bc325774cf29258e487592d8a5e
```

```
6419ad21b7384371a12b5877060147c5
```

Figure 10 /etc/machine-id Comparison

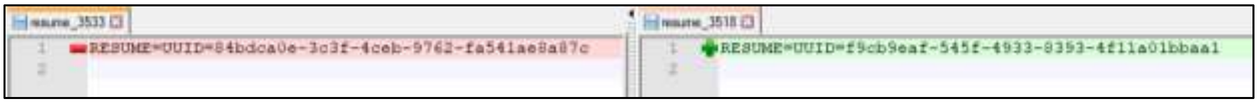
The ID is randomly generated at the time of the OS installation and does not tie directly to any particular information that may be of use, although a variety of other items within the operating system make use of it to uniquely identify the system. The machine-id can be regenerated by using the following commands[17]:

```
rm -f /etc/machine-id /var/lib/dbus/machine-id
dbus-uuidgen --ensure=/etc/machine-id
dbus-uuidgen –ensure
reboot
```

It is unclear what the effects are if the system-id is changed given that several files are named based on its value (see the description of LightDM above). Also, it is important to note that this file exists in both /etc/ and /var/lib/dbus/. These are separate physical files that contain the exact same content (exact hash match) but they are not a link. Because the contents of the file by definition uniquely identifies this system, the file itself can uniquely identify the system. What is currently unclear to me though is how hard it is to change the contents of the file (along with the names of the LightDM files) and what effect it would have on the system.

resume

/etc/initramfs-tools/conf.d/resume is a 1 line file that contains the UUID for the memory image that was made when the system was placed in hibernate (see Figure 11)[18]. This image is saved within the swap file and has a UUID just like the file systems referenced by grub.cnf



```
RESUME=UUID=84bdca0e-3c3f-4ceb-9762-fa541ae8a87c
```

```
RESUME=UUID=f9cb9eaf-545f-4933-8353-4f11a01bbaa1
```

Figure 11 /etc/initramfs-tools/conf.d/resume Differences

Because the UUID of each file system is by definition, unique, and it's included within the resume file, then this file is tied to a particular system. Unless the resume feature is used though, it is possible to remove this file along with the partition and not affect the functionality of the system.

server_config.yml

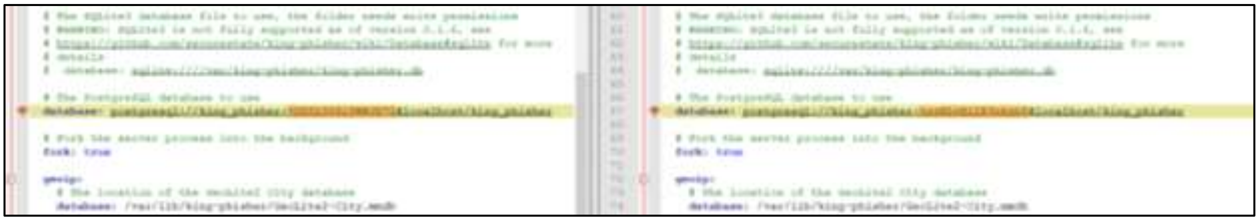
The file /etc/king-phisher/server_config.yml is part of the configuration for the King Phisher tool that is installed by Kali. What is interesting, is that out of all of the tools installed on Kali by default, this is the

[17] https://dbus.freedesktop.org/doc/dbus-uuidgen.1.html

[18] https://askubuntu.com/questions/292878/how-to-set-swap-in-etc-initramfs-tools-conf-d-resume-if-i-have-two-swap-partito

only file that I can find that is unique for them. King Phisher is a tool “for testing and promoting user awareness by simulating real-world phishing attacks.[19]”

The software makes use of the PostgresSQL database server to store the various pieces of data required for each phishing campaign. On line 67 of server_config.yml is the username and password that King Phisher uses to access the database. The username is always “king_phisher”, but the password is 16 characters randomly generated as part of the install script (see Figure 12).



```
# The SQLite3 database file to use, the folder needs write permissions
# WARNING: SQLite3 is not fully supported as of version 0.1.4, see
# https://github.com/securestate/king-phisher/wiki/Database#sqlite for more
# details
#  database: sqlite:////var/king-phisher/king-phisher.db

# The PostgreSQL database to use
database: postgresql://king_phisher:[illegible]@localhost/king_phisher

# Fork the server process into the background
fork: true

geoip:
  # The location of the GeoLite2 City database
  database: /var/lib/king-phisher/GeoLite2-City.mmdb
```

```
# The SQLite3 database file to use, the folder needs write permissions
# WARNING: SQLite3 is not fully supported as of version 0.1.4, see
# https://github.com/securestate/king-phisher/wiki/Database#sqlite for more
# details
#  database: sqlite:////var/king-phisher/king-phisher.db

# The PostgreSQL database to use
database: postgresql://king_phisher:[illegible]@localhost/king_phisher

# Fork the server process into the background
fork: true

geoip:
  # The location of the GeoLite2 City database
  database: /var/lib/king-phisher/GeoLite2-City.mmdb
```

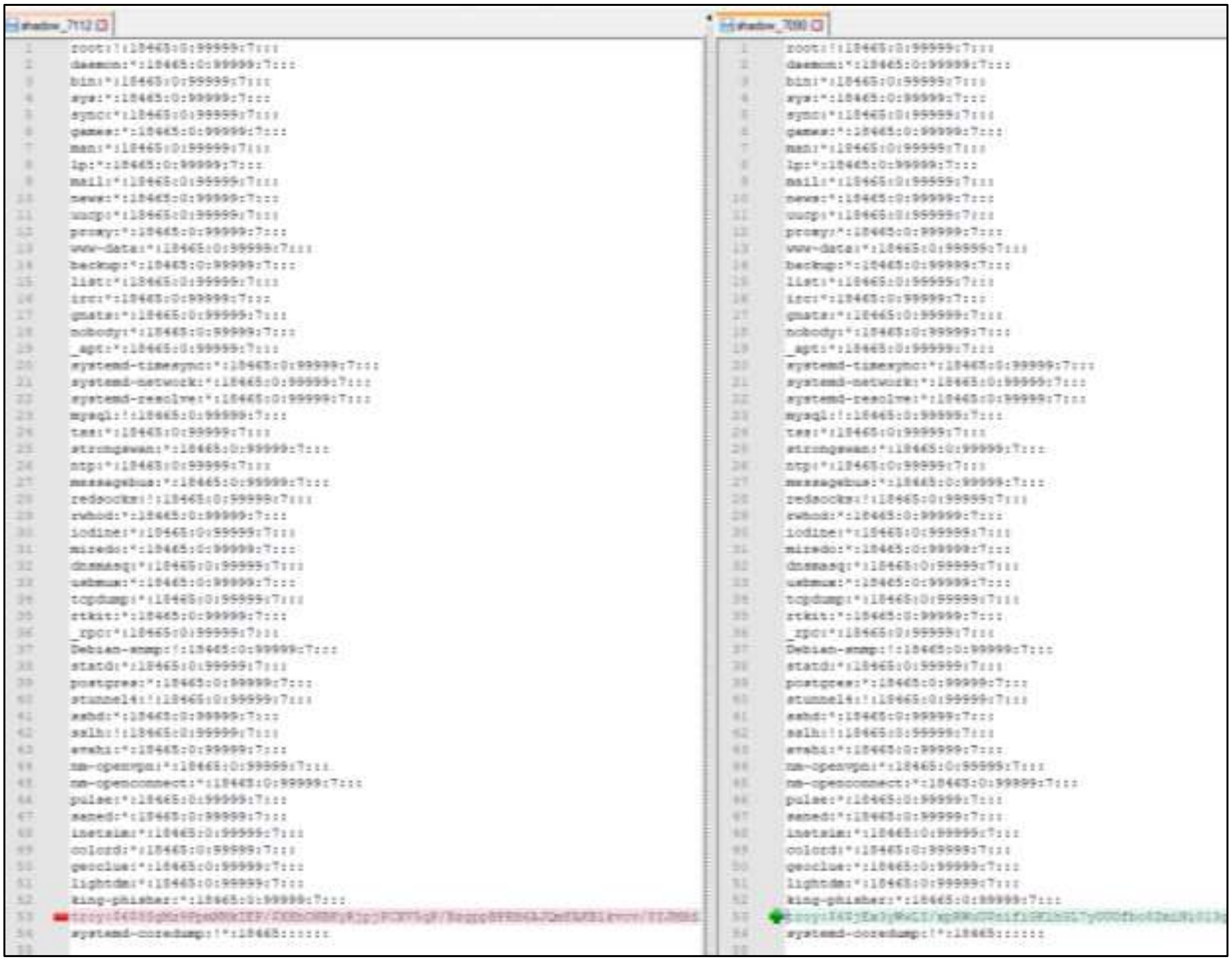
Figure 12 /etc/king-phisher/server_config.yml Comparison

While the password generated by the install is not truly unique to the system, the fact that has 62^16 combinations makes it highly unlikely that the password will be repeated on a random machine. That being said, because this is a simple password to a database, it is very easy for a user to change making this a poor indicator of system uniqueness.

shadow

The encrypted passwords for all local accounts are stored in the /etc/shadow file (a backup copy of the last file is also stored in /etc/shadow-). The file consists of a line for each account (53 by default). The file is the same until you get to the user-created accounts (see Figure 13).

[19] https://github.com/rsmusllp/king-phisher



shadow_7112

```
root:!:18465:0:99999:7:::
daemon:*:18465:0:99999:7:::
bin:*:18465:0:99999:7:::
sys:*:18465:0:99999:7:::
sync:*:18465:0:99999:7:::
games:*:18465:0:99999:7:::
man:*:18465:0:99999:7:::
lp:*:18465:0:99999:7:::
mail:*:18465:0:99999:7:::
news:*:18465:0:99999:7:::
uucp:*:18465:0:99999:7:::
proxy:*:18465:0:99999:7:::
www-data:*:18465:0:99999:7:::
backup:*:18465:0:99999:7:::
list:*:18465:0:99999:7:::
irc:*:18465:0:99999:7:::
gnats:*:18465:0:99999:7:::
nobody:*:18465:0:99999:7:::
_apt:*:18465:0:99999:7:::
systemd-timesync:*:18465:0:99999:7:::
systemd-network:*:18465:0:99999:7:::
systemd-resolve:*:18465:0:99999:7:::
mysql:!:18465:0:99999:7:::
tss:*:18465:0:99999:7:::
strongswan:*:18465:0:99999:7:::
ntp:*:18465:0:99999:7:::
messagebus:*:18465:0:99999:7:::
redsocks:!:18465:0:99999:7:::
rwhod:*:18465:0:99999:7:::
iodine:*:18465:0:99999:7:::
miredo:*:18465:0:99999:7:::
dnsmasq:*:18465:0:99999:7:::
usbmux:*:18465:0:99999:7:::
tcpdump:*:18465:0:99999:7:::
rtkit:*:18465:0:99999:7:::
_rpc:*:18465:0:99999:7:::
Debian-snmp:!:18465:0:99999:7:::
statd:*:18465:0:99999:7:::
postgres:*:18465:0:99999:7:::
stunnel4:!:18465:0:99999:7:::
sshd:*:18465:0:99999:7:::
sslh:!:18465:0:99999:7:::
avahi:*:18465:0:99999:7:::
nm-openvpn:*:18465:0:99999:7:::
nm-openconnect:*:18465:0:99999:7:::
pulse:*:18465:0:99999:7:::
saned:*:18465:0:99999:7:::
inetsim:*:18465:0:99999:7:::
colord:*:18465:0:99999:7:::
geoclue:*:18465:0:99999:7:::
lightdm:*:18465:0:99999:7:::
king-phisher:*:18465:0:99999:7:::
troy:$6$[illegible]
systemd-coredump:!*:18465::::::
```

shadow_7090

```
root:!:18465:0:99999:7:::
daemon:*:18465:0:99999:7:::
bin:*:18465:0:99999:7:::
sys:*:18465:0:99999:7:::
sync:*:18465:0:99999:7:::
games:*:18465:0:99999:7:::
man:*:18465:0:99999:7:::
lp:*:18465:0:99999:7:::
mail:*:18465:0:99999:7:::
news:*:18465:0:99999:7:::
uucp:*:18465:0:99999:7:::
proxy:*:18465:0:99999:7:::
www-data:*:18465:0:99999:7:::
backup:*:18465:0:99999:7:::
list:*:18465:0:99999:7:::
irc:*:18465:0:99999:7:::
gnats:*:18465:0:99999:7:::
nobody:*:18465:0:99999:7:::
_apt:*:18465:0:99999:7:::
systemd-timesync:*:18465:0:99999:7:::
systemd-network:*:18465:0:99999:7:::
systemd-resolve:*:18465:0:99999:7:::
mysql:!:18465:0:99999:7:::
tss:*:18465:0:99999:7:::
strongswan:*:18465:0:99999:7:::
ntp:*:18465:0:99999:7:::
messagebus:*:18465:0:99999:7:::
redsocks:!:18465:0:99999:7:::
rwhod:*:18465:0:99999:7:::
iodine:*:18465:0:99999:7:::
miredo:*:18465:0:99999:7:::
dnsmasq:*:18465:0:99999:7:::
usbmux:*:18465:0:99999:7:::
tcpdump:*:18465:0:99999:7:::
rtkit:*:18465:0:99999:7:::
_rpc:*:18465:0:99999:7:::
Debian-snmp:!:18465:0:99999:7:::
statd:*:18465:0:99999:7:::
postgres:*:18465:0:99999:7:::
stunnel4:!:18465:0:99999:7:::
sshd:*:18465:0:99999:7:::
sslh:!:18465:0:99999:7:::
avahi:*:18465:0:99999:7:::
nm-openvpn:*:18465:0:99999:7:::
nm-openconnect:*:18465:0:99999:7:::
pulse:*:18465:0:99999:7:::
saned:*:18465:0:99999:7:::
inetsim:*:18465:0:99999:7:::
colord:*:18465:0:99999:7:::
geoclue:*:18465:0:99999:7:::
lightdm:*:18465:0:99999:7:::
king-phisher:*:18465:0:99999:7:::
troy:$6$jEx3yWwLY/xpRWuU$nifiGKlhGL7yOUOfbo8ZmiNi013g
systemd-coredump:!*:18465::::::
```

Figure 13 /etc/shadow differences

For the case of this project, I used the username "troy" and the password "password". This created the below entry for the account in the shadow file.

```
troy:$6$jEx3yWwLY/xpRWuU$nifiGKlhGL7yOUOfbo8ZmiNi013g18EeweqIE3M8pSVgdVcfsRLko.vDPspup5agio4A2NHHUz9Hvg84YUD2p.:18465:0:99999:7:::
```

The line is a colon (:) separated string that contains several fields[20].

- The 1st field is the username for the account (in this case "troy")
- The 2nd field contains the hashed password. We'll look at this one closer in a minute.
- The 3rd field indicates the number of days that have passed between Jan 1, 1970, and the last time the password was changed. In this case, the value is 18465 which means the password was set on July 22, 2020. This field is not particularly unique given the fact that all of the systems that are created on a single day (or have a password changed on that day) will all share the same value.

[20] https://linux.die.net/man/5/shadow

- The 4th field is the minimum number of days that must pass between password changes. Here it is 0 which is the default.
- The 5th field is the max number of days that can pass before the user must change their password. Here the value is 99999 which is the default.
- The 6th field indicates the number of days before the max password age that it will warn the user to change their password.
- The 7th and 8th fields are not set by default.

Within the password field noted above is a sub dollar sign ($) separated string[21].

```
$6$jEx3yWwLY/xpRWuU$nifiGKlhGL7yOUOfbo8ZmiNi013g18EeweqIE3M8pSVgdVcfsRLko.vDPspup5agi
o4A2NHHUz9Hvg84YUD2p.
```

- The 1st field with a value of 6 indicates the hash that was used. A value of 6 indicates SHA512 which is the current default for Kali.
- The 2nd field indicates the salt that was used with the password. A salt is an extra piece of data (in this case randomly generated) that is appended to the password to prevent the use of rainbow tables when it comes to cracking an encrypted password. In this case, the salt value is "jEx3yWwLY/xpRWuU"
- The 3rd field is the SHA512 hashed combination of the salt and the password

The fact that the salt is 16 characters long with 64 possible choices for each character gives it 7.9 x 10^{28} combinations. This combined with the fact that virtually the only way to get the same password hash is to have the same salt and password makes it exceedingly likely that if this line matches on two systems, they have a common past.

Fortunately, the regeneration of a password will cause a new salt and in turn a new hash to be generated (even if the same password is used) and will also reset the password change date field to the current date. That makes this file a poor indicator of system uniqueness.

ssh_host_*_key

Within /etc/ssh/ are three pairs of private/public ssh keys that are used by the sshd process. The keys are:

- ssh_host_ecdsa_key / ssh_host_ecdsa_key.pub
- ssh_host_ed25519_key / ssh_host_ed25519_key.pub
- ssh_host_rsa_key / ssh_host_rsa _key.pub

The three keys are referenced in /etc/ssh/sshd_config and are used to authenticate the ssh server itself to clients. These are the keys (the public keys) that are stored in the known_hosts files of clients. These keys are transmitted across the internet.

Because these are ssh keys, they are unique to the system, but they can be easily regenerated by using the following commands[22]:

[21] https://linux.die.net/man/3/crypt

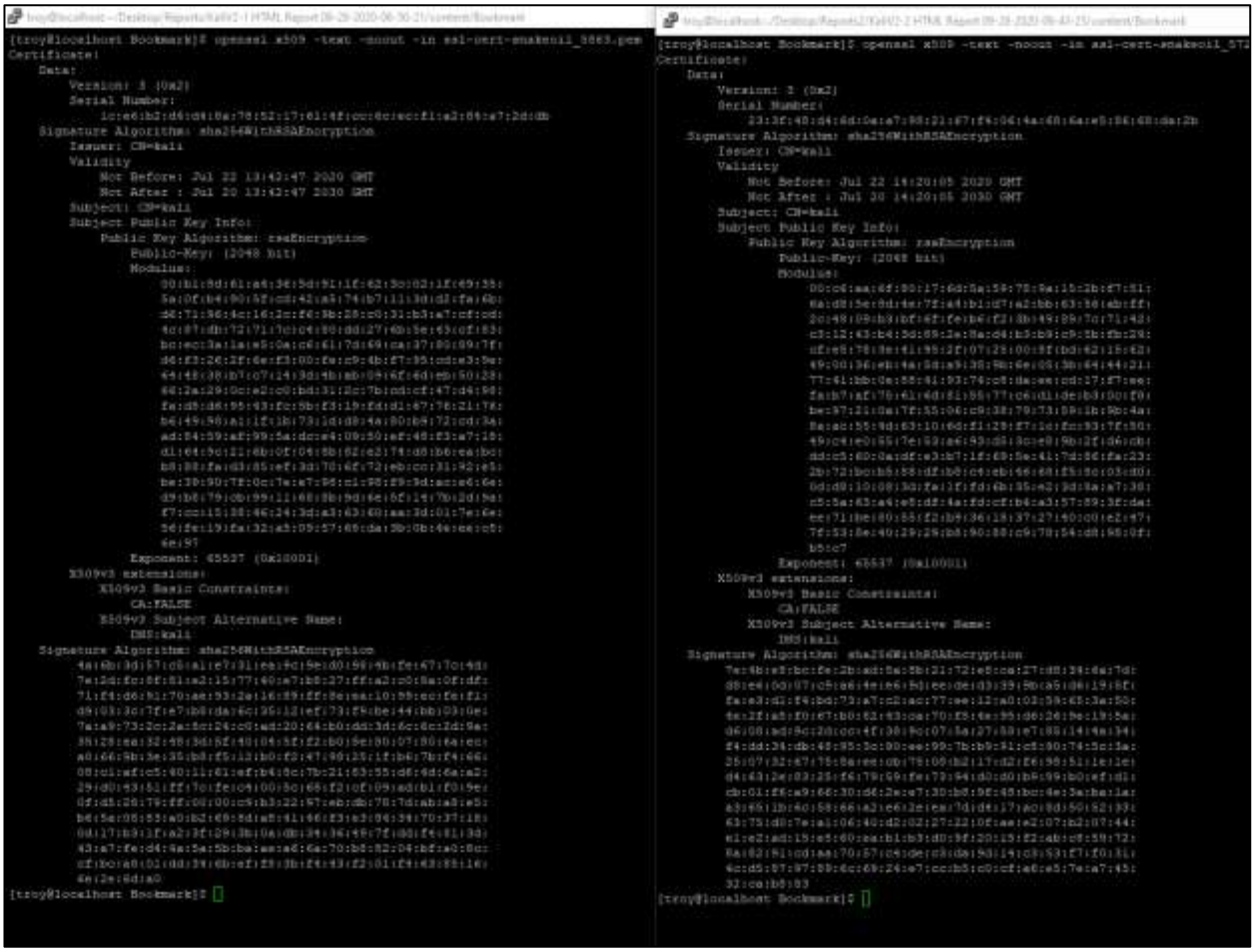
[22] https://www.cyberciti.biz/faq/howto-regenerate-openssh-host-keys/

```
rm -f /etc/ssh_host_*
dpkg-reconfigure openssh-serve
systemctl restart ssh
```

ssl_cert_snakeoil

When a web server starts to run HTTPS, there has to be a certificate there to support it. Generally, the preferred method is to get a signed certificate from a recognized certificate authority so that a web browser will accept it without a problem. When that doesn't happen though, a self-signed certificate can be used.

/etc/ssl/certs/ssl-cert-snakeoil.pem and /etc/ssl/private/ssl-cert-snakeoil.key is a self-signed certificate pair that is generated at the time of installation[23]. This certificate is used for SSL web sessions if another certificate hasn't been explicitly designated.



```
[troy@localhost Bookmark]$ openssl x509 -text -noout -in ssl-cert-snakeoil_5863.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            [illegible]
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=kali
        Validity
            Not Before: Jul 22 13:42:47 2020 GMT
            Not After : Jul 20 13:42:47 2030 GMT
        Subject: CN=kali
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    [illegible]
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Alternative Name:
                DNS:kali
    Signature Algorithm: sha256WithRSAEncryption
         [illegible]
[troy@localhost Bookmark]$
```

```
[troy@localhost Bookmark]$ openssl x509 -text -noout -in ssl-cert-snakeoil_572
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            [illegible]
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=kali
        Validity
            Not Before: Jul 22 14:20:05 2020 GMT
            Not After : Jul 20 14:20:05 2030 GMT
        Subject: CN=kali
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    [illegible]
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Alternative Name:
                DNS:kali
    Signature Algorithm: sha256WithRSAEncryption
         [illegible]
[troy@localhost Bookmark]$
```

Figure 14 /etc/ssl/certs/ssl-cert-snakeoil.pem comparrision

An examination of the keys shows that while the Issuer CN is always kali (based on the system name given at install), everything else is completely different. The validity dates are from the time that the system was originally imaged, until 10 years later.

[23] https://wiki.debian.org/Self-Signed_Certificate

The snakeoil SSL certificate can be regenerated with the following command:

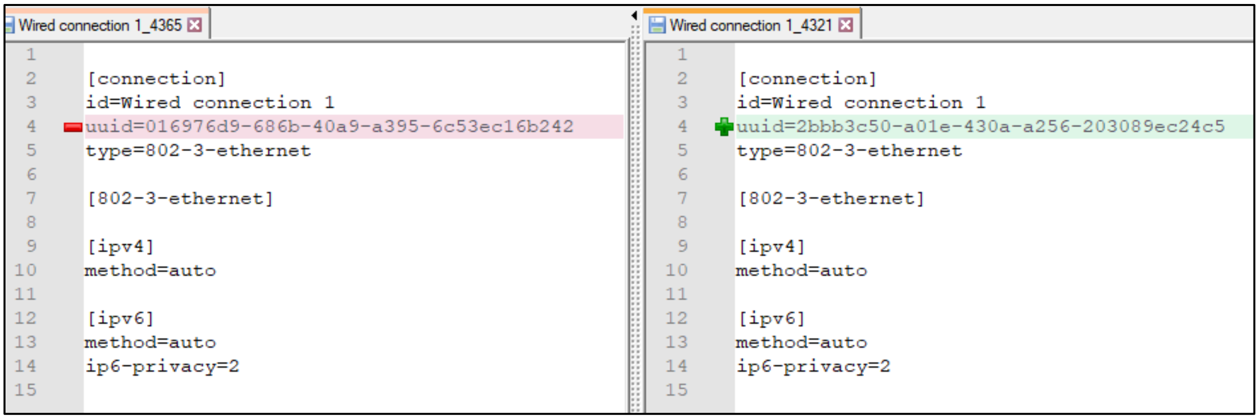
```
make-ssl-cert generate-default-snakeoil --force-overwrite
```

The serial number, modulus, and signatures are always different between each system which makes this file a unique identifier to the system. While there is a risk of this artifact being viewable by the outside world if it was served up as the SSL server certificate when someone tried to browse to it, this would only occur if a web server was installed and it was configured to point to this specific certificate which makes the chances of it being externally visible significantly smaller. While the certificate itself is easy enough to regenerate, the fact that it is unlikely to be noticed in the first place makes it a moderate indicator of system uniqueness.

Wired Connection 1

The file /etc/NetworkManager/system-connections/Wired connection 1 is a file generated by the network-manager to uniquely identify the physical network interface. It is important to note that if additional interfaces are added to the system (wired or wireless) that additional files will be generated[24].

This file contains information about the physical interface itself and is not directly tied to the system's MAC or IP addresses. Each file is 14 lines long and the only difference is the UUID for the interface located on line 4 (see Figure 15).



```
1
2   [connection]
3   id=Wired connection 1
4   uuid=016976d9-686b-40a9-a395-6c53ec16b242
5   type=802-3-ethernet
6
7   [802-3-ethernet]
8
9   [ipv4]
10  method=auto
11
12  [ipv6]
13  method=auto
14  ip6-privacy=2
15
```

```
1
2   [connection]
3   id=Wired connection 1
4   uuid=2bbb3c50-a01e-430a-a256-203089ec24c5
5   type=802-3-ethernet
6
7   [802-3-ethernet]
8
9   [ipv4]
10  method=auto
11
12  [ipv6]
13  method=auto
14  ip6-privacy=2
15
```

Figure 15 /etc/NetworkManager/system-connections/Wired connection 1 differences

The UUID within the file is also contained within /var/lib/NetworkManager and appears in multiple logs. The UUID allows the file to uniquely identify a system but the ID can be changed with minimal effort making this file a poor indicator of uniqueness.

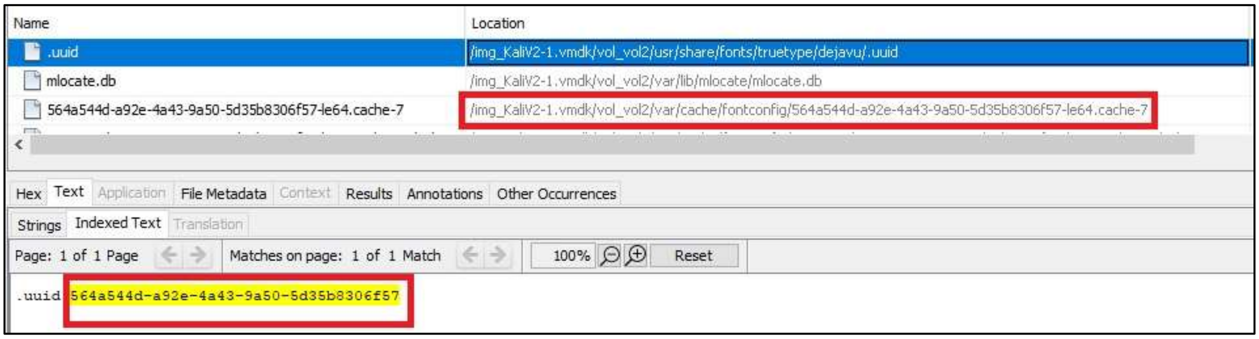
/usr

.uuid

Scattered throughout multiple folders within /usr/share is a file simply called .uuid. The vast majority of these are located within folders related to fonts (33/39) but a few others appear in other folders. Within each file is simply a single UUID and nothing else. When completing a search of the system for each of the UUIDs though, it ties back to one of the uniquely named files found in

[24] https://wiki.archlinux.org/index.php/NetworkManager

/var/cache/fontconfig/ (see Font Config above). This indicates to the system what cache file it can find information about the fonts within that folder.



Name	Location
.uuid	/img_KaliV2-1.vmdk/vol_vol2/usr/share/fonts/truetype/dejavu/.uuid
mlocate.db	/img_KaliV2-1.vmdk/vol_vol2/var/lib/mlocate/mlocate.db
564a544d-a92e-4a43-9a50-5d35b8306f57-le64.cache-7	/img_KaliV2-1.vmdk/vol_vol2/var/cache/fontconfig/564a544d-a92e-4a43-9a50-5d35b8306f57-le64.cache-7

```
.uuid 564a544d-a92e-4a43-9a50-5d35b8306f57
```

Figure 16 /usr/share/.uuid and /var/carche/fontconfig correlation

As I indicated before, the fact that a UUID is present makes this file unique to this particular system but it can also be regenerated simply by deleting the file and the fc-cache -f command making this a poor indicator of system uniqueness.

classes.jsa

Beginning in Java 5.0 the Class Data Sharing feature was provided. The feature was designed to reduce loading time by loading up all frequently used classes once and then mapping the memory structure into a single file. Once this was done, every time java was loaded it would simply reload the memory structure that had already been established[25].

These memory structures are stored in a file called classes.jsa. In Kali, these files exist in two places, /usr/lib/jvm/java-11-openjdk-amd64/lib/server/classes.jsa and /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/classes.jsa

While the classes that get loaded are the same from system to system, because the jsa file is generated on the system and not as part of the downloaded package, it is unique to the system itself however it looks like the file can easily be regenerated on the system.

pyc files

Out of the 79 semi-unique files, 71 of them were .pyc files, mostly located within various cache directories in /usr/lib/python3. If you're not familiar with what a .pyc is, it is just a compiled version of a python file. This is done to speed up the execution of a python file that is frequently used and is done transparently to the user. My first thought is that maybe I got unlucky and perhaps there was an update to one of the source files between the few hours when I started to make images 1 and image 4. But then I noticed that if that was the case, I should find a .py file somewhere where the hash also changed. That wasn't the case. The other thing that I found was that there were cases where images 1 and 3 would have one hash and 2 and 4 would have a new (but same) hash. Again, if a source file had changed, I would expect all of the images that were made after that change to have the same change.

After doing some research, it turns out that this isn't completely unusual. For reasons that I don't completely understand, it's not unusual for a .pyc file to compile with either the same hash or a new

[25] http://sekhar4j.blogspot.com/2014/11/what-is-class-data-sharing.html

hash. The really interesting part is that this will occur even if you recompile on the same system multiple times, let alone a different system. It's just kind of a crapshoot.

I decided to dig into a couple of these .pyc files just to see what the difference was. The first file that I looked at was /usr/lib/python3.8/__pycache__/_pyio.cpython-38.pyc. The file is 74,079 bytes long but between the four images I captured, presented two separate hashes. An examination of the file reveals a total of 14 bytes that are different and 11 of those bytes are actually just a group of 5 bytes and 6 bytes) where the locations were swapped (see Figure 17).

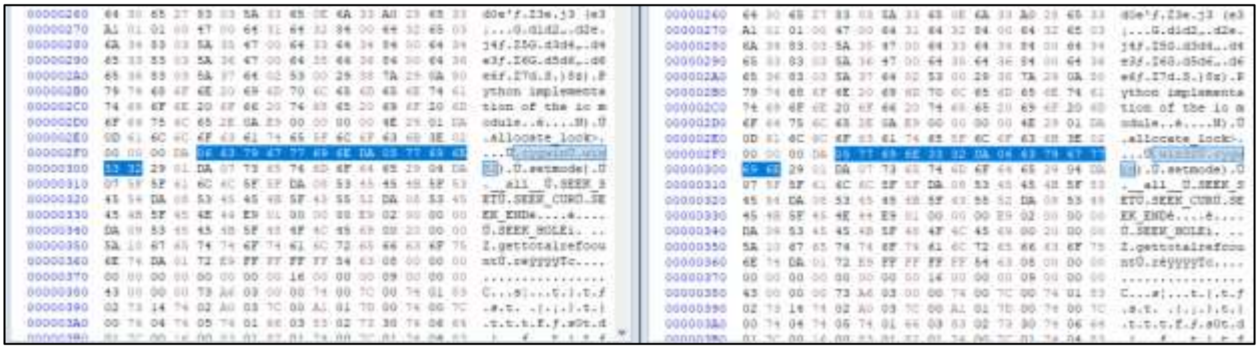


Figure 17 Byte Comparison of /usr/lib/python3.8/__pycache__/_pyio.cpython-38.pyc

A look at another file (/usr/lib/python3/dist-packages/IPython/core/__pycache__/application.cpython-38.pyc) which actually had 3 variations in hashes. In this case, we see something very similar starting at hex offset 0x4E0. In this case the file is 14,126 bytes and depending on which two version you are comparing there are only 7 or 9 bytes different between the two versions (see Table 3).

4E0	4E1	4E2	4E3	4E4	4E5	4E6	4E7	4E8	4E9	4EA	4EB	4EC	4ED	4EE	4EF
00	**DA**	**01**	**31**	**DA**	**04**	**74**	**72**	**75**	**65**	**54**	**3E**	**02**	**00**	**00**	**00**
00	**DA**	**01**	**31**	**DA**	**04**	**74**	**72**	**75**	**65**	**54**	**3E**	**02**	**00**	**00**	**00**
00	**DA**	04	74	72	75	65	DA	01	31	**54**	**3E**	**02**	**00**	**00**	**00**

5E0	5E1	5E2	5E3	5E4	5E5	5E6	5E7	5E8	5E9	5EA	5EB	5EC	5ED	5EE	5EF
DA	**01**	**30**	**DA**	**05**	**66**	**61**	**6C**	**73**	**65**	**46**	**7A**	**8E**	**55**	**6E**	**73**
DA	05	66	61	6C	73	65	DA	01	30	**46**	**7A**	**8E**	**55**	**6E**	**73**
DA	**01**	**30**	**DA**	**05**	**66**	**61**	**6C**	**73**	**65**	**46**	**7A**	**8E**	**55**	**6E**	**73**

Table 3 Byte Comparison of /usr/lib/python3/dist-packages/IPython/core/__pycache__/application.cpython-38.pyc

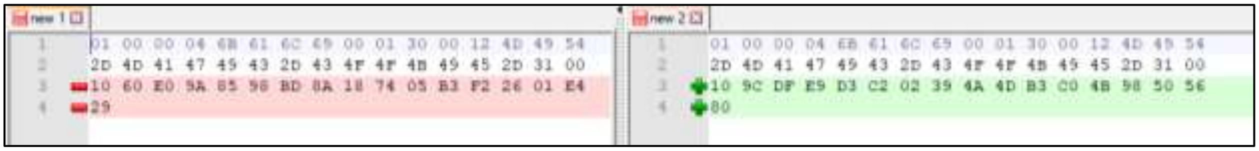
Because of this known situation, I don't believe that having a unique or non-unique hash with .pyc files is indicative one way or another of a particular system. That being said, .pyc files are frequently recompiled by python as different modules are used by the system. Additionally, they can be completely deleted from the system and they will be regenerated as required[26].

[26] https://indianpythonista.wordpress.com/2018/01/05/demystifying-pyc-files/

/var

.Xauthority

LightDM allows for the use of cookies for authentication. Each system stores it's cookie at /var/lib/lightdm/.Xauthority. Each file is 49 bytes, and the first 32 bytes are the same.



```
01 00 00 04 6B 61 6C 69 00 01 30 00 12 4D 49 54
2D 4D 41 47 49 43 2D 43 4F 4F 4B 49 45 2D 31 00
10 60 E0 9A 85 96 BD 8A 18 74 05 B3 F2 26 01 E4
29
```

```
01 00 00 04 6B 61 6C 69 00 01 30 00 12 4D 49 54
2D 4D 41 47 49 43 2D 43 4F 4F 4B 49 45 2D 31 00
10 9C DF E9 D3 C2 02 39 4A 4D B3 C0 4B 98 50 56
80
```

Figure 18 /var/lib/lightdm/.Xauthority differences

Because there are only 17 bytes of entropy in each file, it means that there are only 131 thousand possible combinations. This means that it will repeat itself rather frequently and should not be used to uniquely identify the system even though in our case it had a unique hash each time.

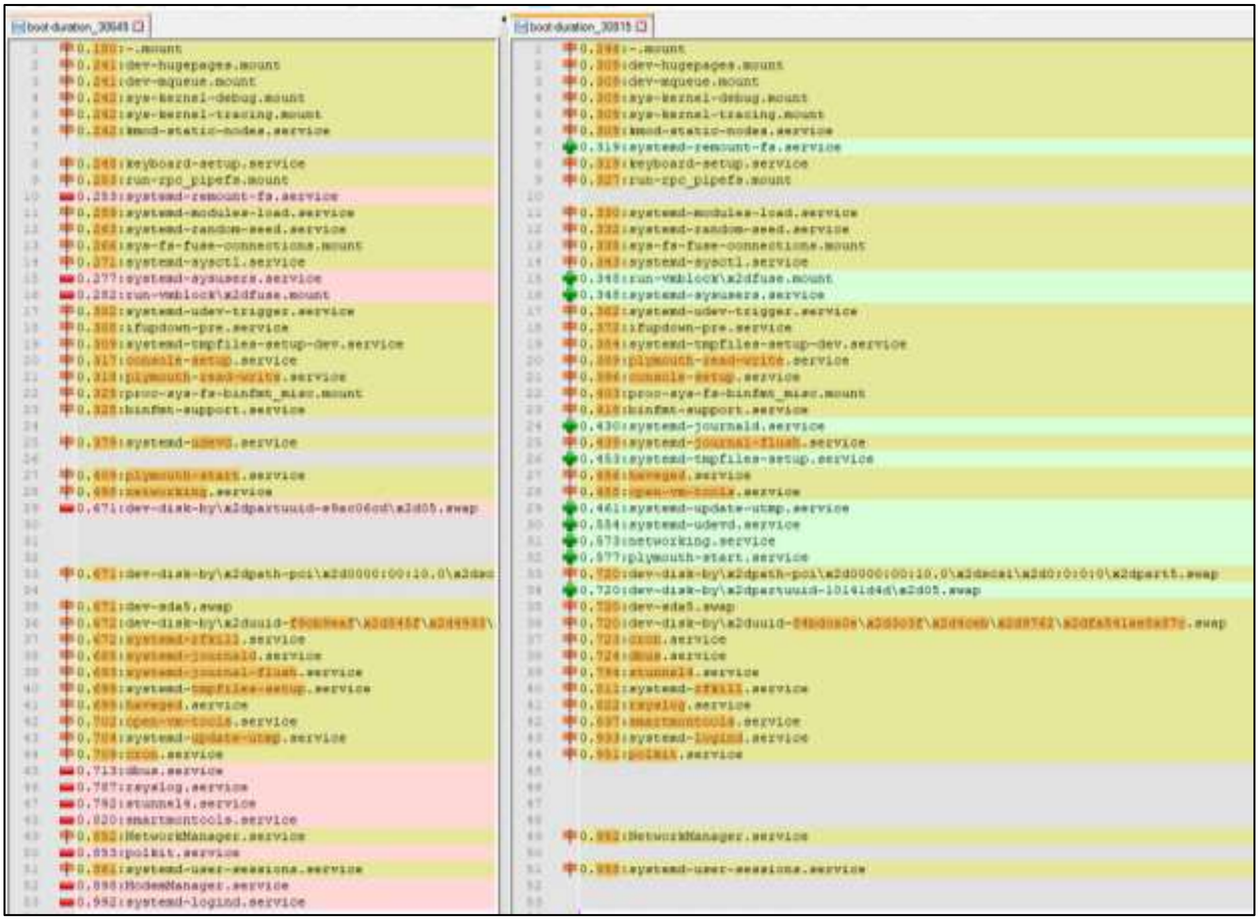
aux-cache

Located at /var/cache/ldconfig/aux-cache is the shared library links and cache file. It is designed to speed up the loading of some libraries, and also allows the dynamic linker to call libraries by their soname[27]. While the same libraries are cached in each installation, because the cache is compiled at run time, they are all different at the hash level. The file can be regenerated by deleting it and then running ldconfig. This cache file can be used to uniquely identify a system but the ease with which it can be regenerated makes it a poor indicator of system uniqueness.

boot_duration

Located at /var/lib/plymouth/ is a file called boot-duration.   Plymouth is the program that shows the splash screen when the system first boots up. While each version of the file is similar, there are differences. First, certain events seem to be missing from some versions as opposed to others. Second, the numbers within each event are also generally different. The one thing within the file that does make it unique is the UUID for the swap partition found on line 36 in the images below. This UUID is located within fstab as well as several log files which makes it unique to the system. While this particular instance of the UUID could be changed relatively easily, it would prove significantly harder to change it in its other locations making this a good indicator of system uniqueness.

[27] https://lists.fedoraproject.org/archives/list/devel@lists.fedoraproject.org/thread/DVBZI2UJRHI5GVMJXIQNLHK4TUFS5ZM2/



boot-duration_30648

```
0.180:-.mount
0.241:dev-hugepages.mount
0.241:dev-mqueue.mount
0.242:sys-kernel-debug.mount
0.242:sys-kernel-tracing.mount
0.242:kmod-static-nodes.service

0.248:keyboard-setup.service
0.253:run-rpc_pipefs.mount
0.253:systemd-remount-fs.service
0.259:systemd-modules-load.service
0.263:systemd-random-seed.service
0.266:sys-fs-fuse-connections.mount
0.271:systemd-sysctl.service
0.277:systemd-sysusers.service
0.282:run-vmblock\x2dfuse.mount
0.302:systemd-udev-trigger.service
0.308:ifupdown-pre.service
0.309:systemd-tmpfiles-setup-dev.service
0.317:console-setup.service
0.318:plymouth-read-write.service
0.323:proc-sys-fs-binfmt_misc.mount
0.328:binfmt-support.service

0.379:systemd-udevd.service

0.409:plymouth-start.service
0.498:networking.service
0.671:dev-disk-by\x2dpartuuid-e9ac06cd\x2d05.swap



0.671:dev-disk-by\x2dpath-pci\x2d0000:00:10.0\x2dsc

0.671:dev-sda5.swap
0.672:dev-disk-by\x2duuid-[illegible]\
0.672:systemd-rfkill.service
0.685:systemd-journald.service
0.693:systemd-journal-flush.service
0.699:systemd-tmpfiles-setup.service
0.699:haveged.service
0.702:open-vm-tools.service
0.704:systemd-update-utmp.service
0.709:cron.service
0.713:dbus.service
0.787:rsyslog.service
0.792:stunnel4.service
0.820:smartmontools.service
0.852:NetworkManager.service
0.853:polkit.service
0.881:systemd-user-sessions.service
0.898:ModemManager.service
0.992:systemd-logind.service
```

boot-duration_30815

```
0.298:-.mount
0.305:dev-hugepages.mount
0.305:dev-mqueue.mount
0.305:sys-kernel-debug.mount
0.305:sys-kernel-tracing.mount
0.305:kmod-static-nodes.service
0.319:systemd-remount-fs.service
0.319:keyboard-setup.service
0.327:run-rpc_pipefs.mount

0.330:systemd-modules-load.service
0.332:systemd-random-seed.service
0.333:sys-fs-fuse-connections.mount
0.343:systemd-sysctl.service
0.348:run-vmblock\x2dfuse.mount
0.348:systemd-sysusers.service
0.362:systemd-udev-trigger.service
0.372:ifupdown-pre.service
0.384:systemd-tmpfiles-setup-dev.service
0.389:plymouth-read-write.service
0.396:console-setup.service
0.403:proc-sys-fs-binfmt_misc.mount
0.418:binfmt-support.service
0.430:systemd-journald.service
0.439:systemd-journal-flush.service
0.453:systemd-tmpfiles-setup.service
0.456:haveged.service
0.458:open-vm-tools.service
0.461:systemd-update-utmp.service
0.554:systemd-udevd.service
0.573:networking.service
0.577:plymouth-start.service
0.720:dev-disk-by\x2dpath-pci\x2d0000:00:10.0\x2dscsi\x2d0:0:0:0\x2dpart5.swap
0.720:dev-disk-by\x2dpartuuid-10141d4d\x2d05.swap
0.720:dev-sda5.swap
0.720:dev-disk-by\x2duuid-[illegible].swap
0.723:cron.service
0.724:dbus.service
0.784:stunnel4.service
0.811:systemd-rfkill.service
0.822:rsyslog.service
0.837:smartmontools.service
0.933:systemd-logind.service
0.951:polkit.service




0.992:NetworkManager.service

0.993:systemd-user-sessions.service
```

cookie

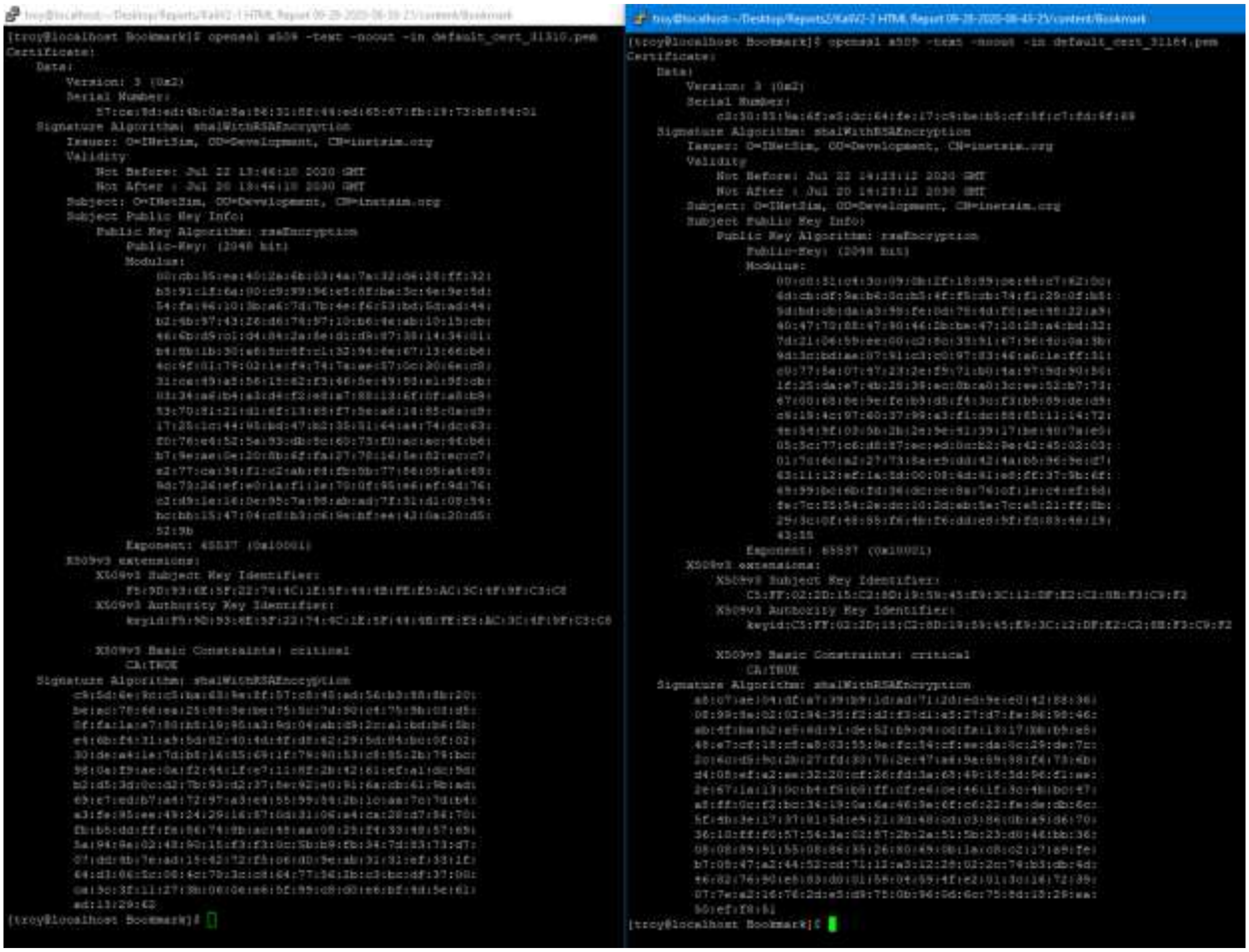
/var/lib/lightdm/.config/pulse/cookie is part of the lightdm display manager that is standard for Kali 1604 and below[28]. The cookie file is used to uniquely identify the system. Because of that, it is of course unique to the system and can be used to identify an individual system.

default_cert.pem

Inetsim is a software suite that can simulate common Internet services within a lab[29]. It comes ready to run many pre-configured services to include an HTTPS server. When run, the certificate located at /var/lib/inetsim/certs/default_cert.pem and key located at /var/lib/inetsim/certs/default_key.pem serves as the SSL certificate for the webserver. Like the /etc/ssl/certs/ssl-cert-snakeoil.pem file, this is a valid SSL certificate. Instead of having a subject CN of kali, it now has a more fully populated subject record that includes an Organization of INetSim, Organization Unit of Development, and a CN of inetsim.org. This key is also valid for 10 years. Also like the snake_oil certificate, the rest of the certificate is unique to this particular installation of the operating system.

[28] https://wiki.ubuntu.com/LightDM
[29] https://www.aldeid.com/wiki/INetSim

Inet by its design is supposed to be used within a lab environment only. While the certificate is technically viewable from the outside if it is served, the chances of this happening accidentally are very low.


troy@localhost:~/Desktop/Reports/KaliV2-1 HTML Report 09-28-2020-08-39-23/content/Bookmark
troy@localhost:~/Desktop/Reports2/KaliV2-2 HTML Report 09-28-2020-08-45-25/content/Bookmark

```
[troy@localhost Bookmark]$ openssl x509 -text -noout -in default_cert_31310.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            57:ce:9d:ed:4b:0a:8a:96:31:8f:44:ed:65:67:fb:19:73:b8:94:01
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: O=INetSim, OU=Development, CN=inetsim.org
        Validity
            Not Before: Jul 22 13:46:10 2020 GMT
            Not After : Jul 20 13:46:10 2030 GMT
        Subject: O=INetSim, OU=Development, CN=inetsim.org
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    [illegible]
                    52:9b
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                F5:9D:93:6E:5F:22:74:4C:1E:5F:44:4B:FE:E5:AC:3C:4F:9F:C3:C6
            X509v3 Authority Key Identifier:
                keyid:F5:9D:93:6E:5F:22:74:4C:1E:5F:44:4B:FE:E5:AC:3C:4F:9F:C3:C6

            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
         [illegible]
         ad:13:29:42
[troy@localhost Bookmark]$
```

```
[troy@localhost Bookmark]$ openssl x509 -text -noout -in default_cert_31164.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            c2:50:83:9a:6f:e5:dc:64:fe:17:c9:be:b5:cf:8f:c7:fd:4f:69
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: O=INetSim, OU=Development, CN=inetsim.org
        Validity
            Not Before: Jul 22 14:23:12 2020 GMT
            Not After : Jul 20 14:23:12 2030 GMT
        Subject: O=INetSim, OU=Development, CN=inetsim.org
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    [illegible]
                    43:15
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                C5:FF:02:2D:15:C2:8D:19:59:45:E9:3C:12:DF:E2:C2:8B:F3:C9:F2
            X509v3 Authority Key Identifier:
                keyid:C5:FF:02:2D:15:C2:8D:19:59:45:E9:3C:12:DF:E2:C2:8B:F3:C9:F2

            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
         [illegible]
         b0:ef:f8:b1
[troy@localhost Bookmark]$
```

Figure 19 /var/lib/inetsim/certs/default_cert.pem comparison

Index.db

Located in /var/cache/man/ are 30 additional directories with cache files for the various man files in each language. Within each directory is a file called index.db which is the cache. The cache is populated by the program mandb[30] and can manually be regenerated on demand. Out of the 79 semi-unique files previously identified, 7 of them belong to the index.db files for various languages.

An examination of the files shows a significant difference between two copies of the same file, but there are still plenty of instances where the same copy is found. For right now, I have no idea why it only affected these 7 languages (vs the other 23) and why some files are the same and others are different. Again, because of this, I don't believe that index.db files for the man files are indicative of a particular system.

[30] https://man7.org/linux/man-pages/man8/mandb.8.html

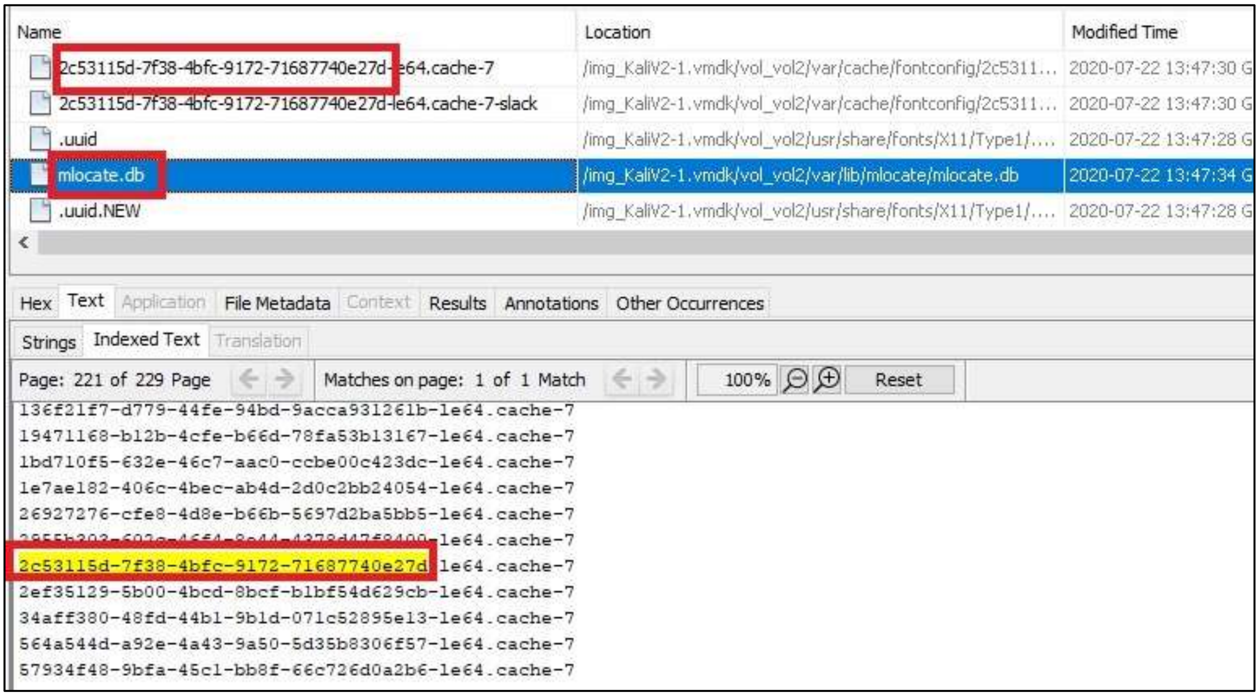
kali-amd64

The folder /var/lib/initramfs-tools/ contains two files, 5.5.0-kali2-amd64 and 5.7.0-kali1-amd64. While I can't figure out exactly what these files are for, initramfs-tools is used to create an initramfs for the kernel headers. These images are loaded up into memory at boot time to allow the rest of the file system to load[31].

Both files contain unique hashes in all four instances and can be used to uniquely identify a system.

mlocate.db

The file /var/lib/mlocate/mlocate.db contains the database that is generated for use with the locate command. It contains among other things, a list of all of the files found on the disk. The reason why the database is different is because of the 45 uniquely named files noted at the start of this paper (see Figure 20). The mlocate.db is continuously changed as files are added/removed from the system. The file can be manually updated through the use of the dbupdate command. This file is not a good indicator of a unique system.



Name	Location	Modified Time
2c53115d-7f38-4bfc-9172-71687740e27d-le64.cache-7	/img_KaliV2-1.vmdk/vol_vol2/var/cache/fontconfig/2c5311...	2020-07-22 13:47:30 G
2c53115d-7f38-4bfc-9172-71687740e27d-le64.cache-7-slack	/img_KaliV2-1.vmdk/vol_vol2/var/cache/fontconfig/2c5311...	2020-07-22 13:47:30 G
.uuid	/img_KaliV2-1.vmdk/vol_vol2/usr/share/fonts/X11/Type1/....	2020-07-22 13:47:28 G
mlocate.db	/img_KaliV2-1.vmdk/vol_vol2/var/lib/mlocate/mlocate.db	2020-07-22 13:47:34 G
.uuid.NEW	/img_KaliV2-1.vmdk/vol_vol2/usr/share/fonts/X11/Type1/....	2020-07-22 13:47:28 G

```
136f21f7-d779-44fe-94bd-9acca931261b-le64.cache-7
19471168-b12b-4cfe-b66d-78fa53b13167-le64.cache-7
1bd710f5-632e-46c7-aac0-ccbe00c423dc-le64.cache-7
1e7ae182-406c-4bec-ab4d-2d0c2bb24054-le64.cache-7
26927276-cfe8-4d8e-b66b-5697d2ba5bb5-le64.cache-7
2955b303-602c-46f4-8e44-4378d47f8400-le64.cache-7
2c53115d-7f38-4bfc-9172-71687740e27d le64.cache-7
2ef35129-5b00-4bcd-8bcf-b1bf54d629cb-le64.cache-7
34aff380-48fd-44b1-9b1d-071c52895e13-le64.cache-7
564a544d-a92e-4a43-9a50-5d35b8306f57-le64.cache-7
57934f48-9bfa-45c1-bb8f-66c726d0a2b6-le64.cache-7
```

Figure 20 Uniquely named files located within mlocate.db

MySQL files

Of the 202 files across the images that always had a unique file hash, 60 of them were located within /var/lib/mysql/mysql as either .frm or .myi files. A .frm file is a table definition file. What is interesting with the .frm files is that the table definitions on each image should match since they are defined the same (columns, etc.). However in the case of column_stats.frm, and others there are only a small

[31] https://wiki.debian.org/initramfs-tools

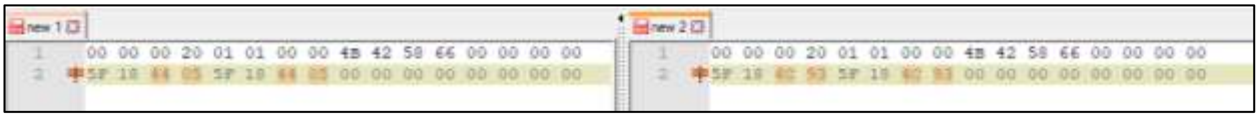
number (11 in that case) of bytes that are different. An examination of the file format shows that these hex offsets should not actually in use so I'm not entirely sure why they are different[32].

The .myi files are the index files for each of the corresponding tables. As with the .frm files, the differences between versions of the same file are minor (only 5 bytes for columns_priv.myi for example) that makes them a poor indicator of a unique system.

pubring.kbx

/var/lib/lightdm/.gnupg/pubring.kbx contains the keyring of public keys that that LightDM recognizes. By default, it doesn't include any keys which means that the keyrings should all have the same hash. But for some reason, they all have a unique hash. The file itself is 32 bytes long and they are the same except for two bytes that are repeated once.

While each file had a different hash here, the fact that there are only two bytes that change means that there are only 65,536 possible combinations so there is a high likelihood that they are repeated. This file should not be relied on to uniquely identify a system.



```
1   00 00 00 20 01 01 00 00 4B 42 58 66 00 00 00 00
2   5F 18 64 05 5F 18 64 05 00 00 00 00 00 00 00 00
```

```
1   00 00 00 20 01 01 00 00 4B 42 58 66 00 00 00 00
2   5F 18 6C 53 5F 18 6C 53 00 00 00 00 00 00 00 00
```

Figure 21 /var/lib/lightdm/.gnupg/pubring.kbx comparison

questions.dat

Located at /var/log/installer/cdebconf/questions.dat, I expected this file to be unique across all images. Instead, though, I got one hash for two images and another hash for the other two images. Questions.dat is a file that is generated when the initial Kali install is occurring. It records the values that are assigned to various variables. The file doesn't contain any date or timestamps, so the chances of getting an exact duplicate (assuming you follow the same build instructions) are significantly higher.

When I examined the files, it took a while, but I finally figured out what the difference was:

```
Name: tasksel/first
Template: tasksel/first
Value: Desktop environment [selecting this item has no effect], ... Xfce (Kali's default desktop environment), Collection of tools [selecting this item has no effect], ... top10 -- the 10 most popular tools, ... default -- recommended tools (available in the live system)
Variables:
CHOICES = Desktop environment [selecting this item has no effect], ... Xfce (Kali's default desktop environment), ... GNOME, ... KDE Plasma, Collection of tools [selecting this item has no effect], ... top10 -- the 10 most popular tools, ... default -- recommended tools (available in the live system), ... large -- default selection plus additional tools
CHOICES_C = desktop, desktop-xfce, desktop-gnome, desktop-kde, meta, meta-top10, meta-default, meta-large
```

[32] https://dev.mysql.com/doc/internals/en/frm-file-format.html

```
Name: tasksel/first
Template: tasksel/first
Value: Desktop environment [selecting this item has no effect], ... Xfce (Kali's default desktop environment), Collection of tools [selecting this item has no effect], ... top10 -- the 10 most popular tools, ... default -- recommended tools (available in the live system)
Variables:
CHOICES_C = desktop, desktop-xfce, desktop-gnome, desktop-kde, meta, meta-top10, meta-default, meta-large
CHOICES = Desktop environment [selecting this item has no effect], ... Xfce (Kali's default desktop environment), ... GNOME, ... KDE Plasma, Collection of tools [selecting this item has no effect], ... top10 -- the 10 most popular tools, ... default -- recommended tools (available in the live system), ... large -- default selection plus additional tools
```

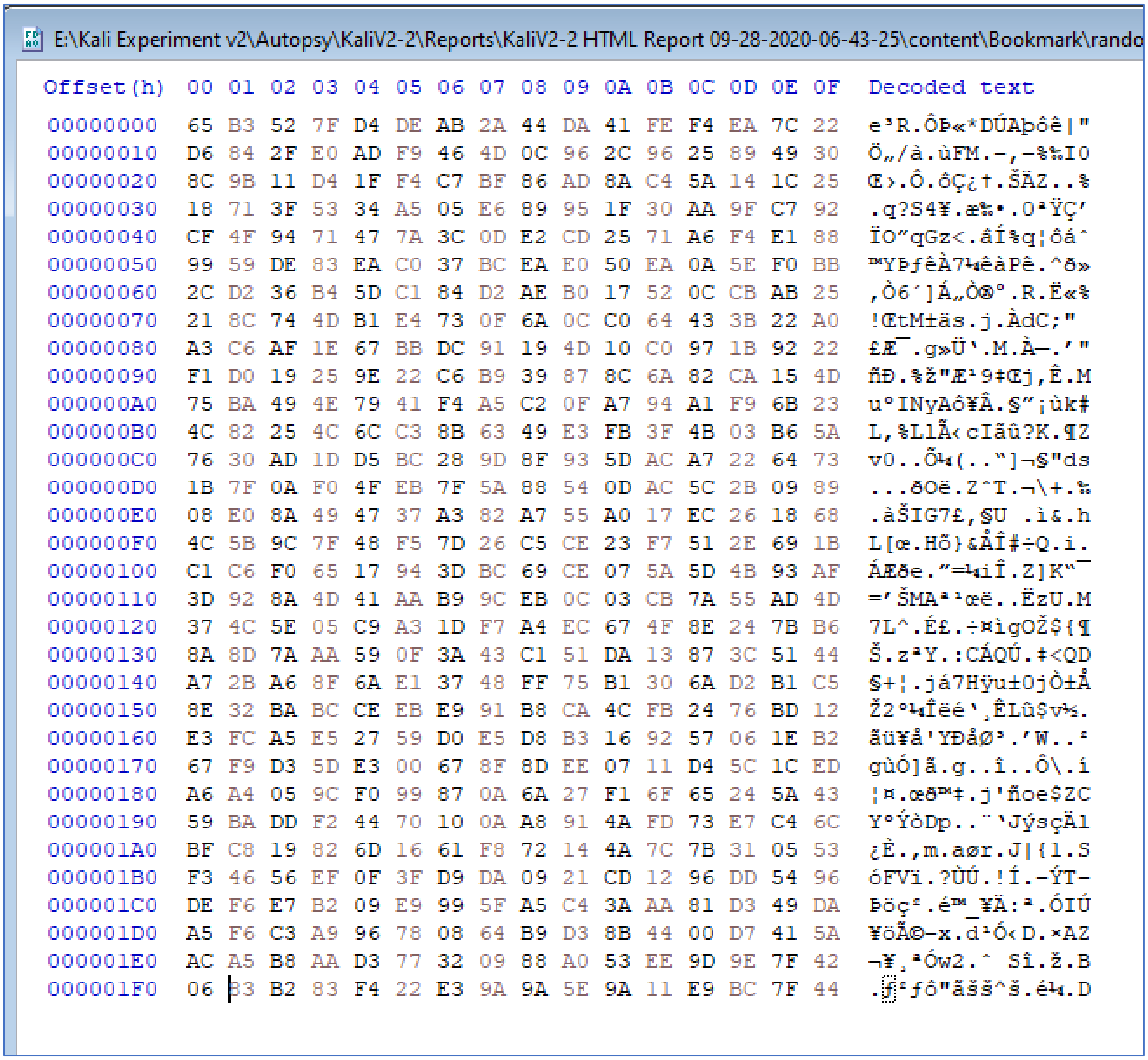
The last and 2nd to last lines contain the same information, but the order is reversed. That is the only difference in a 4k line log file. When I built the images, I did the same steps each time, but for some reason this log entry is different and I'm not entirely sure why. And the fact that it is the same on two of the systems and has the same difference on the other two systems makes it a little unusual. I suspect that there are likely two processes that are going on in parallel and that some of the time the first process finishes first and generates its log entry while other times the second one will complete first.

random-seed

Operating systems make extensive use of various cryptographic functions. For them to operate securely, they must be fed with random numbers. In Linux, random numbers are provided through the /dev/random interface. For this interface to produce truly random numbers though it, must wait for enough entropy to be collected. Unfortunately in the case of booting the system, this can be a lengthy process. To prevent a slow-down, an initial entropy value is saved to disk before shutdown. This value is saved at /var/lib/random-seed[33]. The file is a 4,096-bit file that contains a dump of entropy from the system at shutdown. While it doesn't meet the requirements for true cryptographic use, it is sufficiently random to provide the systemd process what it needs at boot to generate UUIDs and other functions.

While this file is unique to the system, the fact that it has a fairly short lifetime between changes (each system shutdown/reboot) makes it a very poor indicator of system uniqueness.

[33] https://systemd.io/RANDOM_SEEDS/



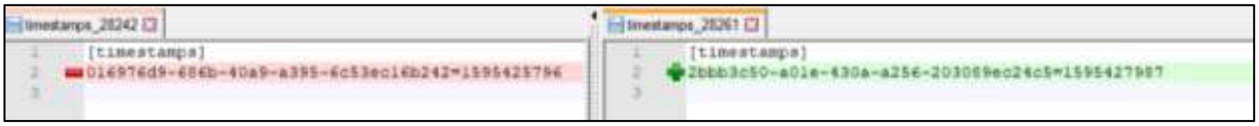
E:\Kali Experiment v2\Autopsy\KaliV2-2\Reports\KaliV2-2 HTML Report 09-28-2020-06-43-25\content\Bookmark\rando

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  65 B3 52 7F D4 DE AB 2A 44 DA 41 FE F4 EA 7C 22  e³R.ÔÞ«*DÚAþôê|"
00000010  D6 84 2F E0 AD F9 46 4D 0C 96 2C 96 25 89 49 30  Ö„/à.ùFM.–,–%‰I0
00000020  8C 9B 11 D4 1F F4 C7 BF 86 AD 8A C4 5A 14 1C 25  Œ›.Ô.ôÇ¿†.ŠÄZ..%
00000030  18 71 3F 53 34 A5 05 E6 89 95 1F 30 AA 9F C7 92  .q?S4¥.æ‰•.0ªŸÇ’
00000040  CF 4F 94 71 47 7A 3C 0D E2 CD 25 71 A6 F4 E1 88  ÏO”qGz<.âÍ%q¦ôáˆ
00000050  99 59 DE 83 EA C0 37 BC EA E0 50 EA 0A 5E F0 BB  ™YÞƒêÀ7¼êàPê.^ð»
00000060  2C D2 36 B4 5D C1 84 D2 AE B0 17 52 0C CB AB 25  ,Ò6´]Á„Ò®°.R.Ë«%
00000070  21 8C 74 4D B1 E4 73 0F 6A 0C C0 64 43 3B 22 A0  !ŒtM±äs.j.ÀdC;" 
00000080  A3 C6 AF 1E 67 BB DC 91 19 4D 10 C0 97 1B 92 22  £Æ¯.g»Ü‘.M.À—.’"
00000090  F1 D0 19 25 9E 22 C6 B9 39 87 8C 6A 82 CA 15 4D  ñÐ.%ž"Æ¹9‡Œj‚Ê.M
000000A0  75 BA 49 4E 79 41 F4 A5 C2 0F A7 94 A1 F9 6B 23  uºINyAô¥Â.§”¡ùk#
000000B0  4C 82 25 4C 6C C3 8B 63 49 E3 FB 3F 4B 03 B6 5A  L‚%LlÃ‹cIãû?K.¶Z
000000C0  76 30 AD 1D D5 BC 28 9D 8F 93 5D AC A7 22 64 73  v0..Õ¼(..“]¬§"ds
000000D0  1B 7F 0A F0 4F EB 7F 5A 88 54 0D AC 5C 2B 09 89  ...ðOë.ZˆT.¬\+.‰
000000E0  08 E0 8A 49 47 37 A3 82 A7 55 A0 17 EC 26 18 68  .àŠIG7£‚§U .ì&.h
000000F0  4C 5B 9C 7F 48 F5 7D 26 C5 CE 23 F7 51 2E 69 1B  L[œ.Hõ}&ÅÎ#÷Q.i.
00000100  C1 C6 F0 65 17 94 3D BC 69 CE 07 5A 5D 4B 93 AF  ÁÆðe.”=¼iÎ.Z]K“¯
00000110  3D 92 8A 4D 41 AA B9 9C EB 0C 03 CB 7A 55 AD 4D  =’ŠMAª¹œë..ËzU.M
00000120  37 4C 5E 05 C9 A3 1D F7 A4 EC 67 4F 8E 24 7B B6  7L^.É£.÷¤ìgOŽ${¶
00000130  8A 8D 7A AA 59 0F 3A 43 C1 51 DA 13 87 3C 51 44  Š.zªY.:CÁQÚ.‡<QD
00000140  A7 2B A6 8F 6A E1 37 48 FF 75 B1 30 6A D2 B1 C5  §+¦.já7Hÿu±0jÒ±Å
00000150  8E 32 BA BC CE EB E9 91 B8 CA 4C FB 24 76 BD 12  Ž2º¼Îëé‘¸ÊLû$v½.
00000160  E3 FC A5 E5 27 59 D0 E5 D8 B3 16 92 57 06 1E B2  ãü¥å'YÐåØ³.’W..²
00000170  67 F9 D3 5D E3 00 67 8F 8D EE 07 11 D4 5C 1C ED  gùÓ]ã.g..î..Ô\.í
00000180  A6 A4 05 9C F0 99 87 0A 6A 27 F1 6F 65 24 5A 43  ¦¤.œð™‡.j'ñoe$ZC
00000190  59 BA DD F2 44 70 10 0A A8 91 4A FD 73 E7 C4 6C  YºÝòDp..¨‘JýsçÄl
000001A0  BF C8 19 82 6D 16 61 F8 72 14 4A 7C 7B 31 05 53  ¿È.‚m.aør.J|{1.S
000001B0  F3 46 56 EF 0F 3F D9 DA 09 21 CD 12 96 DD 54 96  óFVï.?ÙÚ.!Í.–ÝT–
000001C0  DE F6 E7 B2 09 E9 99 5F A5 C4 3A AA 81 D3 49 DA  Þöç².é™_¥Ä:ª.ÓIÚ
000001D0  A5 F6 C3 A9 96 78 08 64 B9 D3 8B 44 00 D7 41 5A  ¥öÃ©–x.d¹Ó‹D.×AZ
000001E0  AC A5 B8 AA D3 77 32 09 88 A0 53 EE 9D 9E 7F 42  ¬¥¸ªÓw2.ˆ Sî.ž.B
000001F0  06 83 B2 83 F4 22 E3 9A 9A 5E 9A 11 E9 BC 7F 44  .ƒ²ƒô"ãšš^š.é¼.D
```

Figure 22 /var/lib/systemd/random-seed

timestamps

When a user uses the Network Manager Connection Editor in the GUI, one item presented to them is the last time that a particular network connection was used. This information is found in the file /var/lib/NetworkManager/timestamps (see Figure 23). The file consists of a single line for each network interface on the system. The line contains the UUID for the connection (see the analysis of file /etc/NetworkManager/system-connections/Wired connection 1 above) and the epoch time of when that connection was last active.



timestamps_28242

```
[timestamps]
016976d9-686b-40a9-a395-6c53ec16b242=1595425796
```

timestamps_28261

```
[timestamps]
2bbb3c50-a01e-430a-a256-203089ec24c5=1595427987
```

Figure 23 /var/lin/NetworkManager/timestamps differences

While the timestamp does not help in the identification of a system (although can help with the building of a timeline in traditional forensics), the fact that the file contains the same UUID as is used in /etc/NetworkManager/system-connections/Wired connection 1 means that this file can uniquely identify a particular system.

Conclusion

In conclusion, the Kali is a fairly complicated, and poorly documented operating system. While the vast majority of it is common to every installation, there are 226 files that are unique to every installation of it (excluding log files). While many of these files are very easy to replace without affecting the system, there are 13 that appear to be extremely difficult to modify without either affecting the operation of the system, or removing all artificats that would indicate that they had been purposely changed.

While it appears that all of these files would require access to the system itself (likely root level), they do present a vulnerability to the complete anonamyzation of a system. In the end, if true annomization of the system is a requirement, the operating system should be installed fresh each time it is used.

Appendix 1: Always Unique Hashes

Path
/boot/grub/grub.cfg
/boot/initrd.img-5.5.0-kali2-amd64
/boot/initrd.img-5.7.0-kali1-amd64
/etc/adjtime
/etc/fstab
/etc/initramfs-tools/conf.d/resume
/etc/king-phisher/server_config.yml
/etc/machine-id
/etc/NetworkManager/system-connections/Wired connection 1
/etc/shadow
/etc/shadow-
/etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_ecdsa_key.pub
/etc/ssh/ssh_host_ed25519_key
/etc/ssh/ssh_host_ed25519_key.pub
/etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_rsa_key.pub
/etc/ssl/certs/java/cacerts
/etc/ssl/certs/ssl-cert-snakeoil.pem
/etc/ssl/private/ssl-cert-snakeoil.key
/usr/lib/jvm/java-11-openjdk-amd64/lib/server/classes.jsa
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/classes.jsa
/usr/lib/python3.8/__pycache__/_markupbase.cpython-38.pyc
/usr/lib/python3.8/__pycache__/ftplib.cpython-38.pyc
/usr/lib/python3.8/__pycache__/hashlib.cpython-38.pyc
/usr/lib/python3.8/__pycache__/pydoc.cpython-38.pyc
/usr/lib/python3.8/__pycache__/rlcompleter.cpython-38.pyc
/usr/lib/python3.8/asyncio/__pycache__/__main__.cpython-38.pyc
/usr/lib/python3/dist-packages/django/db/__pycache__/utils.cpython-38.pyc
/usr/lib/python3/dist-packages/django/test/__pycache__/signals.cpython-38.pyc
/usr/lib/python3/dist-packages/flask/__pycache__/helpers.cpython-38.pyc
/usr/lib/python3/dist-packages/gevent/__pycache__/_socket3.cpython-38.pyc
/usr/lib/python3/dist-packages/gpg/__pycache__/core.cpython-38.pyc
/usr/lib/python3/dist-packages/hupper/__pycache__/worker.cpython-38.pyc
/usr/lib/python3/dist-packages/IPython/core/__pycache__/inputsplitter.cpython-38.pyc
/usr/lib/python3/dist-packages/IPython/core/__pycache__/inputtransformer2.cpython-38.pyc
/usr/lib/python3/dist-packages/matplotlib/__pycache__/dates.cpython-38.pyc
/usr/lib/python3/dist-packages/mitmproxy/net/http/__pycache__/request.cpython-38.pyc
/usr/lib/python3/dist-packages/nbformat/v4/__pycache__/convert.cpython-38.pyc

/usr/lib/python3/dist-packages/packaging/__pycache__/tags.cpython-38.pyc
/usr/lib/python3/dist-packages/pandas/core/__pycache__/resample.cpython-38.pyc
/usr/lib/python3/dist-packages/pandas/tests/indexes/datetimes/__pycache__/test_tools.cpython-38.pyc
/usr/lib/python3/dist-packages/pandas/tests/indexes/interval/__pycache__/test_interval.cpython-38.pyc
/usr/lib/python3/dist-packages/scipy/_lib/__pycache__/_numpy_compat.cpython-38.pyc
/usr/lib/python3/dist-packages/scipy/spatial/tests/__pycache__/test_distance.cpython-38.pyc
/usr/lib/python3/dist-packages/wapitiCore/net/__pycache__/swf.cpython-38.pyc
/usr/local/share/fonts/.uuid
/usr/share/fonts/.uuid
/usr/share/fonts/cMap/.uuid
/usr/share/fonts/cmap/.uuid
/usr/share/fonts/opentype/.uuid
/usr/share/fonts/opentype/cantarell/.uuid
/usr/share/fonts/opentype/font-awesome/.uuid
/usr/share/fonts/opentype/roboto/.uuid
/usr/share/fonts/opentype/roboto/slab/.uuid
/usr/share/fonts/opentype/urw-base35/.uuid
/usr/share/fonts/truetype/.uuid
/usr/share/fonts/truetype/dejavu/.uuid
/usr/share/fonts/truetype/droid/.uuid
/usr/share/fonts/truetype/firacode/.uuid
/usr/share/fonts/truetype/font-awesome/.uuid
/usr/share/fonts/truetype/hack/.uuid
/usr/share/fonts/truetype/lato/.uuid
/usr/share/fonts/truetype/liberation/.uuid
/usr/share/fonts/truetype/lyx/.uuid
/usr/share/fonts/truetype/noto/.uuid
/usr/share/fonts/truetype/quicksand/.uuid
/usr/share/fonts/truetype/ttf-bitstream-vera/.uuid
/usr/share/fonts/type1/.uuid
/usr/share/fonts/type1/urw-base35/.uuid
/usr/share/fonts/X11/.uuid
/usr/share/fonts/X11/100dpi/.uuid
/usr/share/fonts/X11/75dpi/.uuid
/usr/share/fonts/X11/encodings/.uuid
/usr/share/fonts/X11/encodings/large/.uuid
/usr/share/fonts/X11/misc/.uuid
/usr/share/fonts/X11/Type1/.uuid
/usr/share/fonts/X11/util/.uuid
/usr/share/javascript/mathjax/fonts/HTML-CSS/TeX/otf/.uuid

/usr/share/pgcli/pgcli/packages/__pycache__/sqlcompletion.cpython-38.pyc
/usr/share/poppler/cMap/Adobe-CNS1/.uuid
/usr/share/poppler/cMap/Adobe-GB1/.uuid
/usr/share/poppler/cMap/Adobe-Japan1/.uuid
/usr/share/poppler/cMap/Adobe-Japan2/.uuid
/usr/share/poppler/cMap/Adobe-Korea1/.uuid
/var/cache/ldconfig/aux-cache
/var/cache/man/cs/index.db
/var/cache/man/da/index.db
/var/cache/man/de/index.db
/var/cache/man/es/index.db
/var/cache/man/fr/index.db
/var/cache/man/hu/index.db
/var/cache/man/index.db
/var/cache/man/it/index.db
/var/cache/man/ja/index.db
/var/cache/man/nl/index.db
/var/cache/man/pl/index.db
/var/cache/man/pt/index.db
/var/cache/man/pt_BR/index.db
/var/cache/man/ru/index.db
/var/cache/man/sv/index.db
/var/cache/man/zh_CN/index.db
/var/lib/dbus/machine-id
/var/lib/inetsim/certs/default_cert.pem
/var/lib/inetsim/certs/default_key.pem
/var/lib/initramfs-tools/5.5.0-kali2-amd64
/var/lib/initramfs-tools/5.7.0-kali1-amd64
/var/lib/lightdm/.config/pulse/cookie
/var/lib/lightdm/.gnupg/pubring.kbx
/var/lib/lightdm/.Xauthority
/var/lib/mlocate/mlocate.db
/var/lib/mysql/aria_log.00000001
/var/lib/mysql/aria_log_control
/var/lib/mysql/ib_logfile0
/var/lib/mysql/ibdata1
/var/lib/mysql/mysql/column_stats.frm
/var/lib/mysql/mysql/column_stats.MYI
/var/lib/mysql/mysql/columns_priv.frm
/var/lib/mysql/mysql/columns_priv.MYI
/var/lib/mysql/mysql/db.frm

/var/lib/mysql/mysql/db.MYI
/var/lib/mysql/mysql/event.frm
/var/lib/mysql/mysql/event.MYI
/var/lib/mysql/mysql/func.frm
/var/lib/mysql/mysql/func.MYI
/var/lib/mysql/mysql/general_log.frm
/var/lib/mysql/mysql/gtid_slave_pos.frm
/var/lib/mysql/mysql/help_category.frm
/var/lib/mysql/mysql/help_category.MYI
/var/lib/mysql/mysql/help_keyword.frm
/var/lib/mysql/mysql/help_keyword.MYI
/var/lib/mysql/mysql/help_relation.frm
/var/lib/mysql/mysql/help_relation.MYI
/var/lib/mysql/mysql/help_topic.frm
/var/lib/mysql/mysql/help_topic.MYI
/var/lib/mysql/mysql/host.frm
/var/lib/mysql/mysql/host.MYI
/var/lib/mysql/mysql/index_stats.frm
/var/lib/mysql/mysql/index_stats.MYI
/var/lib/mysql/mysql/innodb_index_stats.frm
/var/lib/mysql/mysql/innodb_index_stats.ibd
/var/lib/mysql/mysql/innodb_table_stats.frm
/var/lib/mysql/mysql/innodb_table_stats.ibd
/var/lib/mysql/mysql/plugin.frm
/var/lib/mysql/mysql/plugin.MYI
/var/lib/mysql/mysql/proc.frm
/var/lib/mysql/mysql/proc.MYD
/var/lib/mysql/mysql/proc.MYI
/var/lib/mysql/mysql/procs_priv.frm
/var/lib/mysql/mysql/procs_priv.MYI
/var/lib/mysql/mysql/proxies_priv.frm
/var/lib/mysql/mysql/proxies_priv.MYD
/var/lib/mysql/mysql/proxies_priv.MYI
/var/lib/mysql/mysql/roles_mapping.frm
/var/lib/mysql/mysql/roles_mapping.MYI
/var/lib/mysql/mysql/servers.frm
/var/lib/mysql/mysql/servers.MYI
/var/lib/mysql/mysql/slow_log.frm
/var/lib/mysql/mysql/table_stats.frm
/var/lib/mysql/mysql/table_stats.MYI
/var/lib/mysql/mysql/tables_priv.frm

/var/lib/mysql/mysql/tables_priv.MYI
/var/lib/mysql/mysql/time_zone.frm
/var/lib/mysql/mysql/time_zone.MYI
/var/lib/mysql/mysql/time_zone_leap_second.frm
/var/lib/mysql/mysql/time_zone_leap_second.MYI
/var/lib/mysql/mysql/time_zone_name.frm
/var/lib/mysql/mysql/time_zone_name.MYI
/var/lib/mysql/mysql/time_zone_transition.frm
/var/lib/mysql/mysql/time_zone_transition.MYI
/var/lib/mysql/mysql/time_zone_transition_type.frm
/var/lib/mysql/mysql/time_zone_transition_type.MYI
/var/lib/mysql/mysql/transaction_registry.frm
/var/lib/mysql/mysql/user.frm
/var/lib/mysql/mysql/user.MYI
/var/lib/NetworkManager/timestamps
/var/lib/plymouth/boot-duration
/var/lib/postgresql/12/main/global/pg_control
/var/lib/postgresql/12/main/pg_wal/000000010000000000000001
/var/lib/systemd/random-seed

Table 4 Always Unique Hash Listing

Appendix 2: Uniquely Named Files

Path
/var/lib/NetworkManager/internal-016976d9-686b-40a9-a395-6c53ec16b242-eth0.lease
/var/cache/fontconfig/02d4f520-a997-4d93-9fff-6082379a324e-le64.cache-7
/var/cache/fontconfig/c0b0467c-cc0f-4844-9c8a-27e4ffbc4834-le64.cache-7
/var/cache/fontconfig/a5e62240-c06e-4786-a111-1d916b98e579-le64.cache-7
/var/cache/fontconfig/26927276-cfe8-4d8e-b66b-5697d2ba5bb5-le64.cache-7
/var/cache/fontconfig/7231c049-01f8-4488-ac0c-b0e5c4b59ecd-le64.cache-7
/var/cache/fontconfig/f04028cc-8cf2-4a79-9217-ccb09ab5f7f5-le64.cache-7
/var/cache/fontconfig/ac2ede34-3e40-4594-9f86-03729fd892cc-le64.cache-7
/var/cache/fontconfig/564a544d-a92e-4a43-9a50-5d35b8306f57-le64.cache-7
/var/cache/fontconfig/2955b303-602c-46f4-8e44-4378d47f8400-le64.cache-7
/var/cache/fontconfig/90833a27-c08d-4e82-94df-92d24ecebbe2-le64.cache-7
/var/cache/fontconfig/586e1327-7f72-4fad-b160-c8cc02421c48-le64.cache-7
/var/cache/fontconfig/c4fa06ad-f49f-4925-a331-caffb2d66b77-le64.cache-7
/var/cache/fontconfig/136f21f7-d779-44fe-94bd-9acca931261b-le64.cache-7
/var/cache/fontconfig/0a0e8bb7-3085-40c3-8c39-e823b0f05d36-le64.cache-7
/var/cache/fontconfig/05d9d9df-8b2b-4c25-91da-3e240196b756-le64.cache-7
/var/cache/fontconfig/bda66590-93bf-4994-a514-50b9e2ec119f-le64.cache-7
/var/cache/fontconfig/1e7ae182-406c-4bec-ab4d-2d0c2bb24054-le64.cache-7
/var/cache/fontconfig/da91ae14-4828-40aa-a41a-a78c9d0a88b5-le64.cache-7
/var/cache/fontconfig/d802e684-a3fd-4627-ac3c-8582bb44db61-le64.cache-7
/var/cache/fontconfig/2ef35129-5b00-4bcd-8bcf-b1bf54d629cb-le64.cache-7
/var/cache/fontconfig/19471168-b12b-4cfe-b66d-78fa53b13167-le64.cache-7
/var/cache/fontconfig/82323306-e521-49b2-885f-e57265725cf3-le64.cache-7
/var/cache/fontconfig/2c53115d-7f38-4bfc-9172-71687740e27d-le64.cache-7
/var/cache/fontconfig/594ec2e8-d50e-45d2-b381-c21880502f86-le64.cache-7
/var/cache/fontconfig/d26e4990-f173-4e6f-8e64-a5375eb79e29-le64.cache-7
/var/cache/fontconfig/d46ee887-955a-4e71-84b1-70fcb6868152-le64.cache-7
/var/cache/fontconfig/57934f48-9bfa-45c1-bb8f-66c726d0a2b6-le64.cache-7
/var/cache/fontconfig/9cf6e8b1-bfcc-4fb9-800c-c080822c6433-le64.cache-7
/var/cache/fontconfig/ed6ec1d1-1752-496b-87c3-b9b1cc507b10-le64.cache-7
/var/cache/fontconfig/e55bbb24-eeb9-4478-a0b4-d0a88a3aceb3-le64.cache-7
/var/cache/fontconfig/a437f435-1eb7-44d5-be3d-dc6bce27adfa-le64.cache-7
/var/cache/fontconfig/761618bc-ca18-497b-ab10-b24955b0250d-le64.cache-7
/var/cache/fontconfig/1bd710f5-632e-46c7-aac0-ccbe00c423dc-le64.cache-7
/var/cache/fontconfig/34aff380-48fd-44b1-9b1d-071c52895e13-le64.cache-7
/var/cache/fontconfig/a1214b3a-75c4-4241-ad6f-692e75b8c09f-le64.cache-7
/var/cache/fontconfig/d3646869-ec44-4f6f-8a94-fa0b5b5937fc-le64.cache-7
/var/cache/fontconfig/da8b9f6d-322b-4513-8434-ac674e86cd94-le64.cache-7
/var/cache/fontconfig/5d74a0ea-6c03-47df-a268-8df6d3645fff-le64.cache-7

/var/lib/lightdm/.config/pulse/6419ad21b7384371a12b5877060147c5-device-volumes.tdb
/var/lib/lightdm/.config/pulse/6419ad21b7384371a12b5877060147c5-default-sink
/var/lib/lightdm/.config/pulse/6419ad21b7384371a12b5877060147c5-default-source
/var/lib/lightdm/.config/pulse/6419ad21b7384371a12b5877060147c5-stream-volumes.tdb
/var/log/journal/6419ad21b7384371a12b5877060147c5/system.journal

Table 5 Uniquely Named Files