OSQuery on Debian 10

Installation guide & Security Operations

By Sanil Almeida (CPTE | CEH)

TABLE OF CONTENTS

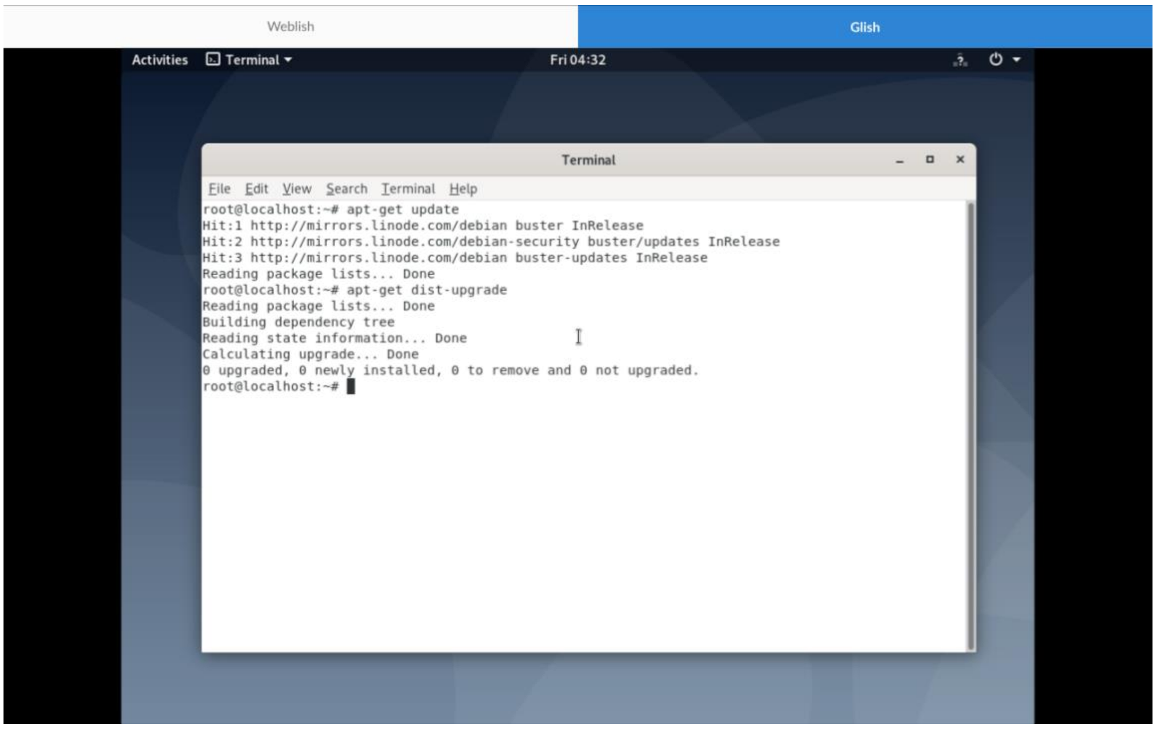
STEPS TO FOLLOW:

Step 1

On your Debian 10 machine open terminal window and type in the following commands to update and upgrade existing packages.

# sudo apt-get update

# sudo apt-get dist-upgrade



```
root@localhost:~# apt-get update
Hit:1 http://mirrors.linode.com/debian buster InRelease
Hit:2 http://mirrors.linode.com/debian-security buster/updates InRelease
Hit:3 http://mirrors.linode.com/debian buster-updates InRelease
Reading package lists... Done
root@localhost:~# apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@localhost:~#
```

Step 2

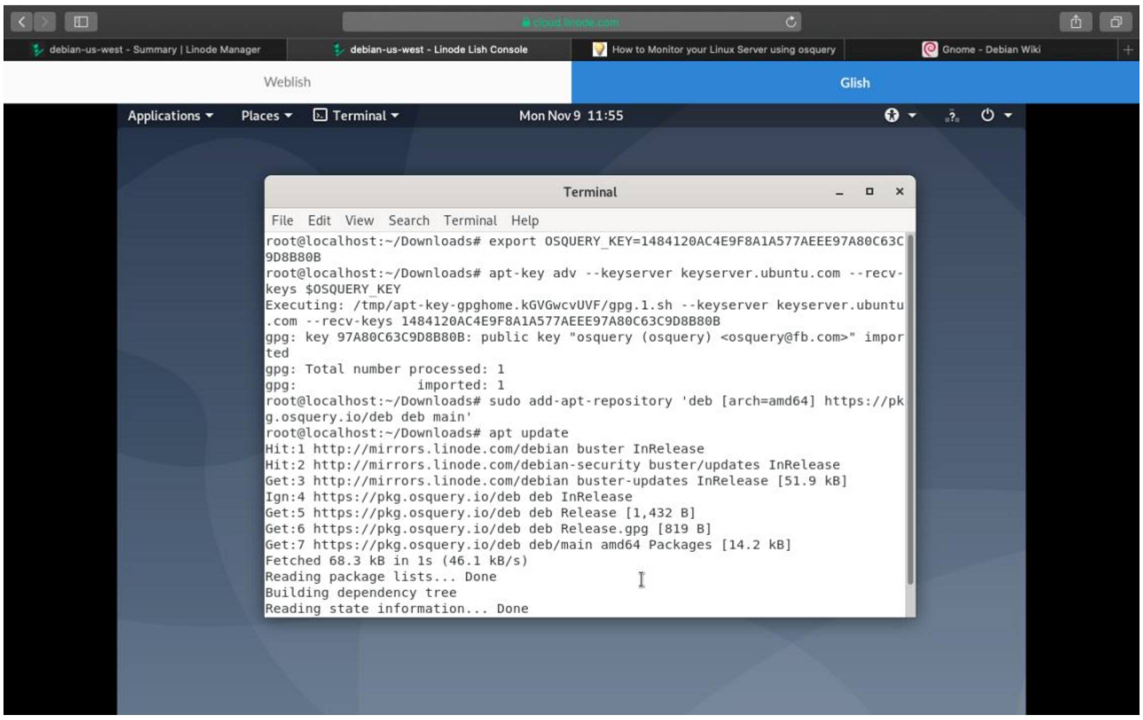
After installing the packages it is time to **download** OSQuery. Run the following commands in the terminal.

# export OSQUERY_KEY= 1484120AC4E9F8A1A577AEEE97A80C63C9D8B80B

# sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys $OSQUERY_KEY

# sudo add-apt-repository 'deb [arch=amd64] https://pkg.osquery.io/deb deb main'

# sudo apt-get update

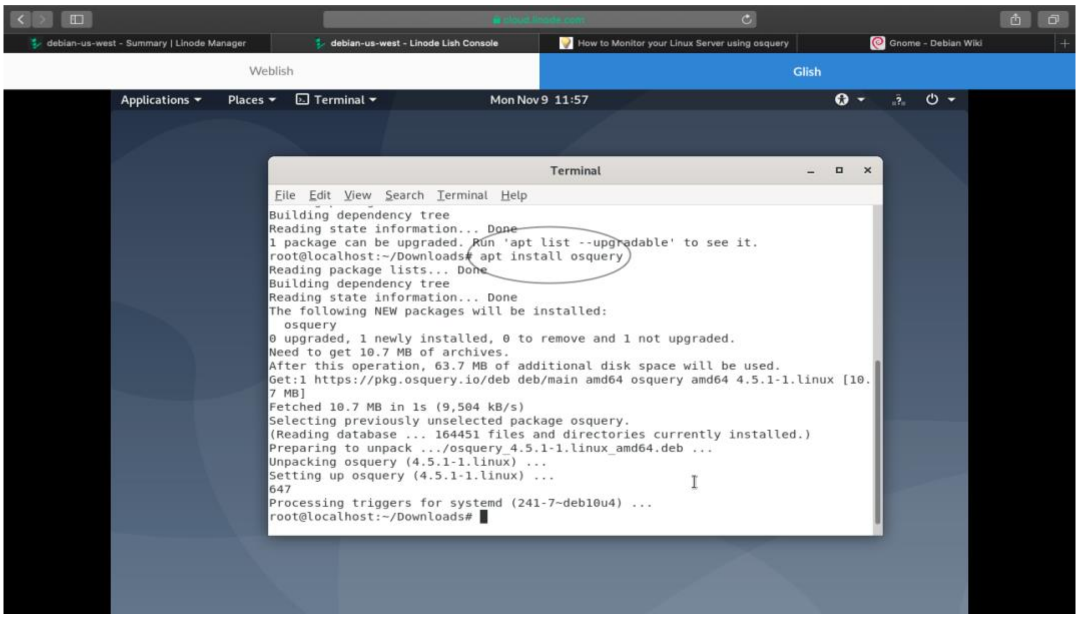

debian-us-west - Summary | Linode Manager
debian-us-west - Linode Lish Console
How to Monitor your Linux Server using osquery
Gnome - Debian Wiki
Weblish
Glish
Applications
Places
Terminal
Mon Nov 9 11:55
Terminal
File Edit View Search Terminal Help

```
root@localhost:~/Downloads# export OSQUERY_KEY=1484120AC4E9F8A1A577AEEE97A80C63C
9D8B80B
root@localhost:~/Downloads# apt-key adv --keyserver keyserver.ubuntu.com --recv-
keys $OSQUERY_KEY
Executing: /tmp/apt-key-gpghome.kGVGwcvUVF/gpg.1.sh --keyserver keyserver.ubuntu
.com --recv-keys 1484120AC4E9F8A1A577AEEE97A80C63C9D8B80B
gpg: key 97A80C63C9D8B80B: public key "osquery (osquery) <osquery@fb.com>" impor
ted
gpg: Total number processed: 1
gpg:               imported: 1
root@localhost:~/Downloads# sudo add-apt-repository 'deb [arch=amd64] https://pk
g.osquery.io/deb deb main'
root@localhost:~/Downloads# apt update
Hit:1 http://mirrors.linode.com/debian buster InRelease
Hit:2 http://mirrors.linode.com/debian-security buster/updates InRelease
Get:3 http://mirrors.linode.com/debian buster-updates InRelease [51.9 kB]
Ign:4 https://pkg.osquery.io/deb deb InRelease
Get:5 https://pkg.osquery.io/deb deb Release [1,432 B]
Get:6 https://pkg.osquery.io/deb deb Release.gpg [819 B]
Get:7 https://pkg.osquery.io/deb deb/main amd64 Packages [14.2 kB]
Fetched 68.3 kB in 1s (46.1 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Step 3

Now, to **install** OSQuery, execute following command in the terminal.

# sudo apt-get install osquery


debian-us-west - Summary | Linode Manager
debian-us-west - Linode Lish Console
How to Monitor your Linux Server using osquery
Gnome - Debian Wiki
Weblish
Glish
Applications
Places
Terminal
Mon Nov 9 11:57
Terminal
File Edit View Search Terminal Help

```
Building dependency tree
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@localhost:~/Downloads# apt install osquery
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  osquery
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 10.7 MB of archives.
After this operation, 63.7 MB of additional disk space will be used.
Get:1 https://pkg.osquery.io/deb deb/main amd64 osquery amd64 4.5.1-1.linux [10.
7 MB]
Fetched 10.7 MB in 1s (9,504 kB/s)
Selecting previously unselected package osquery.
(Reading database ... 164451 files and directories currently installed.)
Preparing to unpack .../osquery_4.5.1-1.linux_amd64.deb ...
Unpacking osquery (4.5.1-1.linux) ...
Setting up osquery (4.5.1-1.linux) ...
647
Processing triggers for systemd (241-7~deb10u4) ...
root@localhost:~/Downloads#
```

Step 4

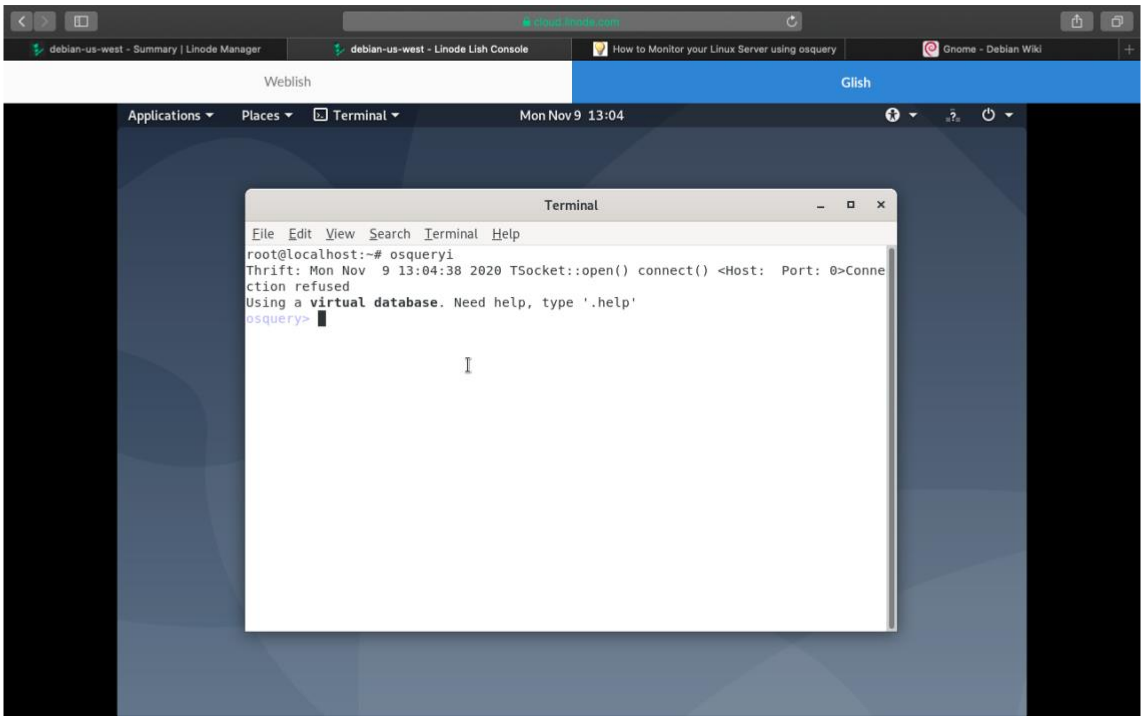
Once the OSQuery has been installed we can launch it as a daemon or a standalone service.

To launch it as a daemon use the command,

# osqueryd

To launch it as a standalone service use the command,

osqueryi

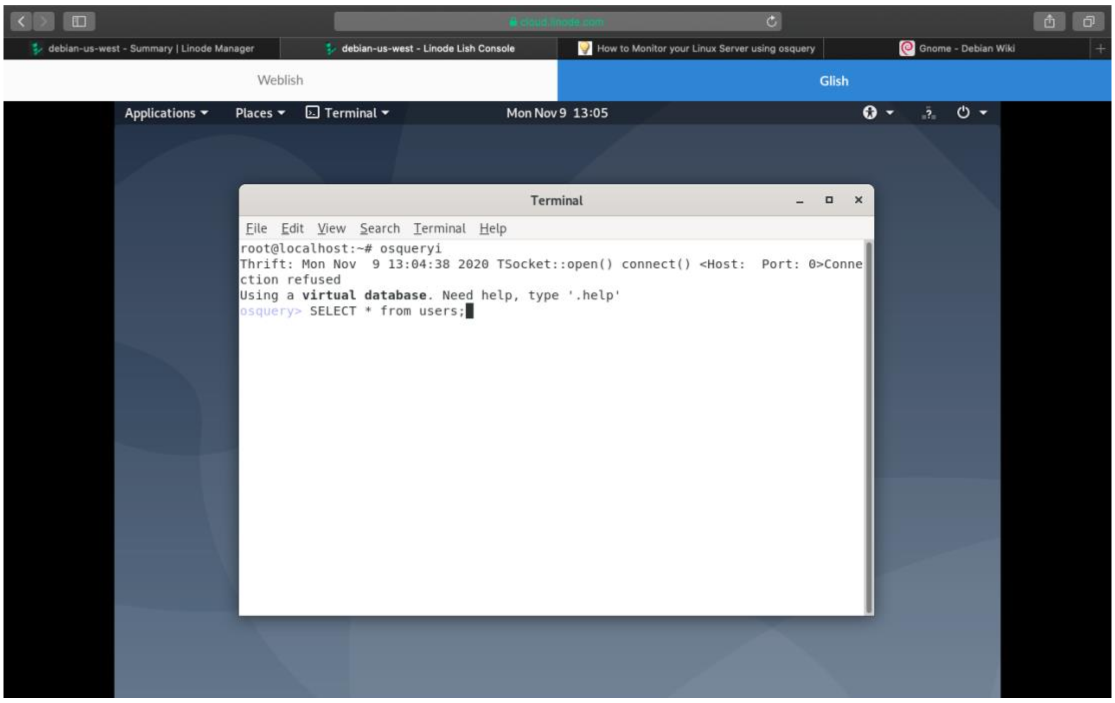

debian-us-west - Summary | Linode Manager
debian-us-west - Linode Lish Console
How to Monitor your Linux Server using osquery
Gnome - Debian Wiki
Weblish
Glish
Applications
Places
Terminal
Mon Nov 9 13:04
Terminal
File Edit View Search Terminal Help

```
root@localhost:~# osqueryi
Thrift: Mon Nov  9 13:04:38 2020 TSocket::open() connect() <Host:  Port: 0>Conne
ction refused
Using a virtual database. Need help, type '.help'
osquery> 
```

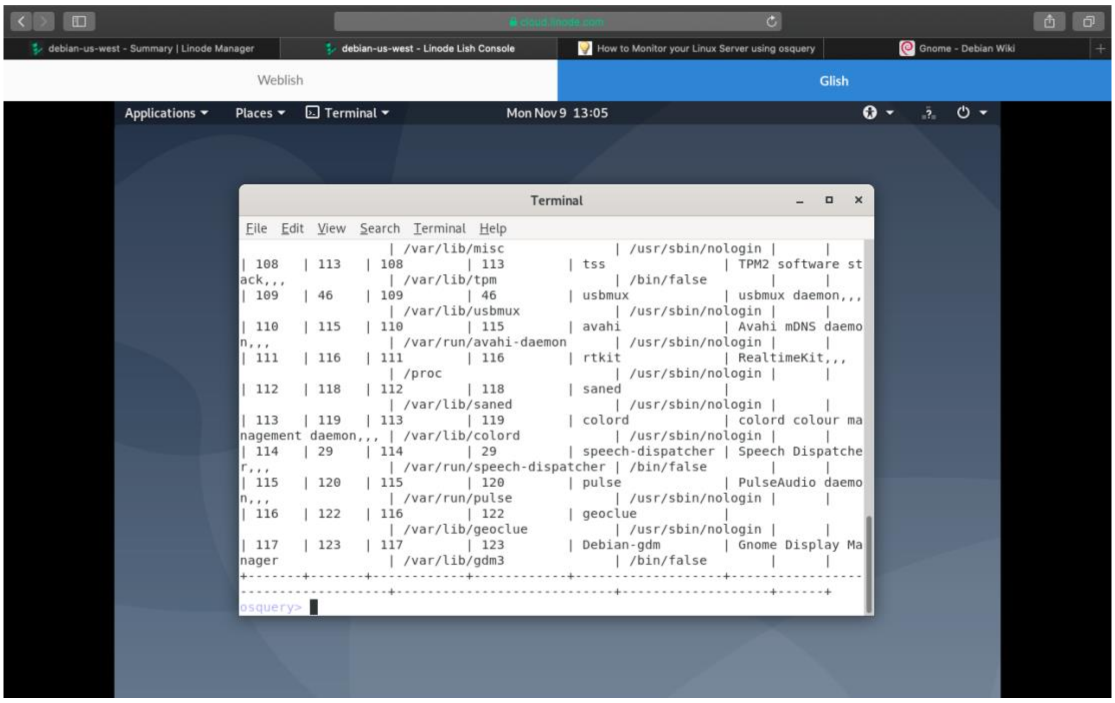
Step 5

After starting the service in the CLI, you can fire SQL queries. As shown in the example below to retrieve user information from the system.

osquery> SELECT * from users;


debian-us-west - Summary | Linode Manager
debian-us-west - Linode Lish Console
How to Monitor your Linux Server using osquery
Gnome - Debian Wiki
Weblish
Glish
Applications
Places
Terminal
Mon Nov 9 13:05
Terminal
File Edit View Search Terminal Help

```
root@localhost:~# osqueryi
Thrift: Mon Nov  9 13:04:38 2020 TSocket::open() connect() <Host:  Port: 0>Conne
ction refused
Using a virtual database. Need help, type '.help'
osquery> SELECT * from users;
```

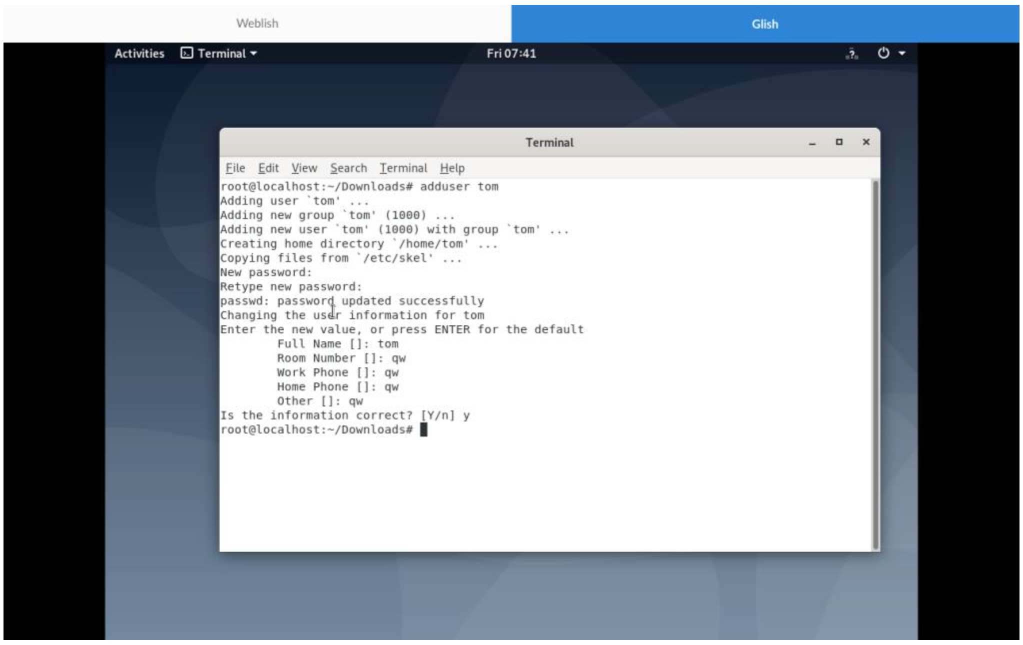

debian-us-west - Summary | Linode Manager
debian-us-west - Linode Lish Console
How to Monitor your Linux Server using osquery
Gnome - Debian Wiki
Weblish
Glish
Applications
Places
Terminal
Mon Nov 9 13:05
Terminal
File Edit View Search Terminal Help

```
                   | /var/lib/misc             | /usr/sbin/nologin |      |
| 108   | 113   | 108          | 113          | tss                   | TPM2 software st
ack,,,             | /var/lib/tpm              | /bin/false        |      |
| 109   | 46    | 109          | 46           | usbmux                | usbmux daemon,,,
                   | /var/lib/usbmux           | /usr/sbin/nologin |      |
| 110   | 115   | 110          | 115          | avahi                 | Avahi mDNS daemo
n,,,               | /var/run/avahi-daemon     | /usr/sbin/nologin |      |
| 111   | 116   | 111          | 116          | rtkit                 | RealtimeKit,,,
                   | /proc                     | /usr/sbin/nologin |      |
| 112   | 118   | 112          | 118          | saned                 |
                   | /var/lib/saned            | /usr/sbin/nologin |      |
| 113   | 119   | 113          | 119          | colord                | colord colour ma
nagement daemon,,, | /var/lib/colord           | /usr/sbin/nologin |      |
| 114   | 29    | 114          | 29           | speech-dispatcher | Speech Dispatche
r,,,               | /var/run/speech-dispatcher | /bin/false        |      |
| 115   | 120   | 115          | 120          | pulse                 | PulseAudio daemo
n,,,               | /var/run/pulse            | /usr/sbin/nologin |      |
| 116   | 122   | 116          | 122          | geoclue               |
                   | /var/lib/geoclue          | /usr/sbin/nologin |      |
| 117   | 123   | 117          | 123          | Debian-gdm            | Gnome Display Ma
nager              | /var/lib/gdm3             | /bin/false        |      |
+-------+-------+------------+------------+-----------------+-----------------
-------------------+----------------------------+-------------------+------+
osquery> 
```

Security Operations

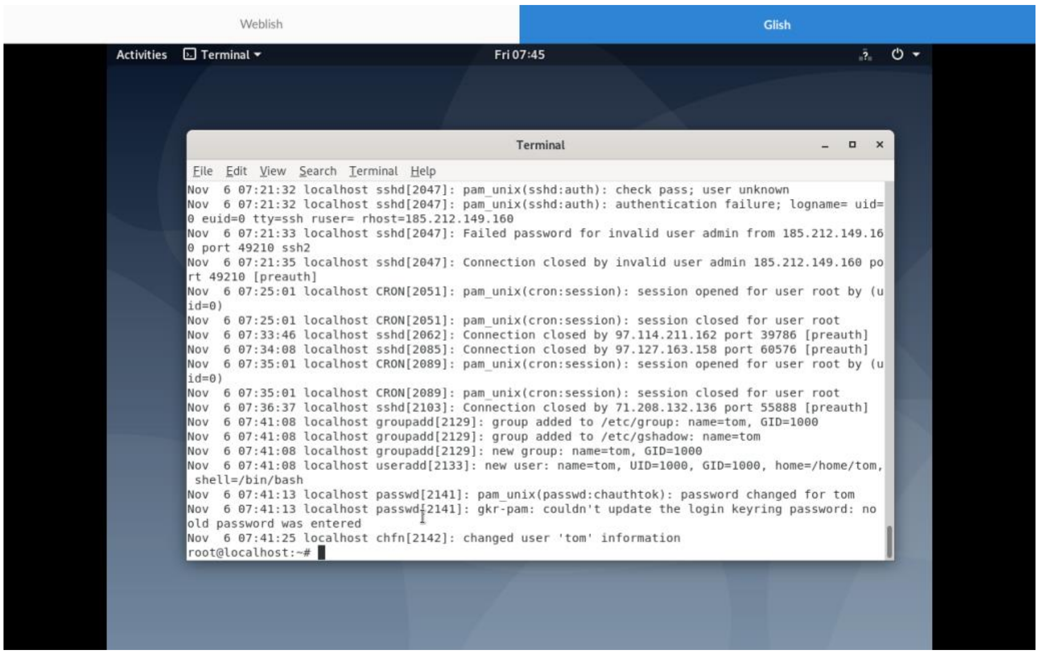
1. Scenario (Inefficient Logging through SIEM)

- We create a user on the system named Tom. Using the command, **# adduser tom**



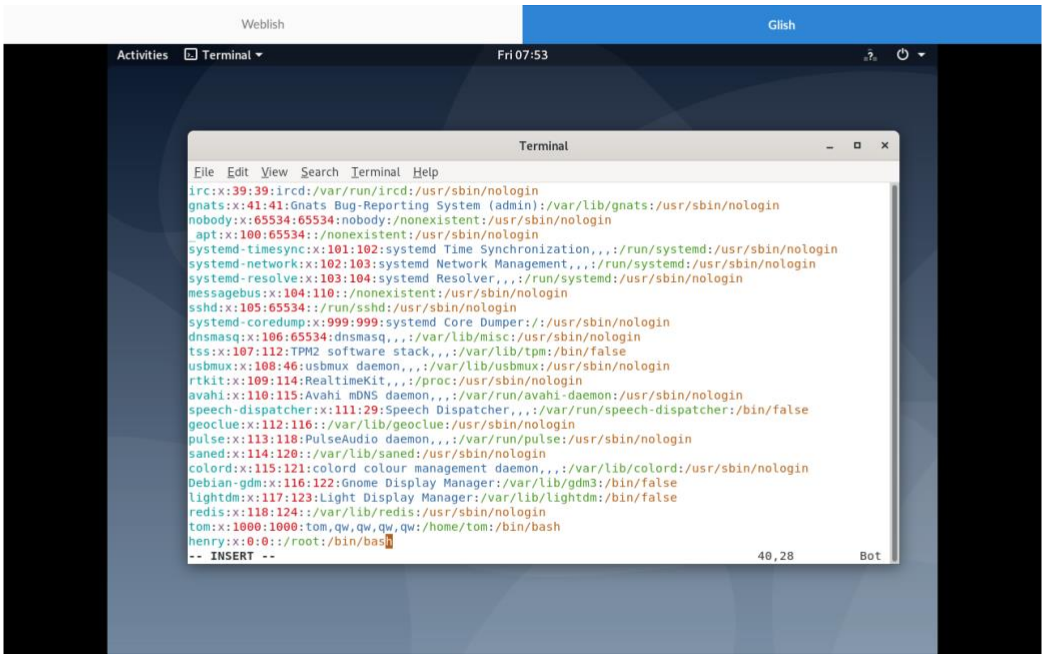
```
root@localhost:~/Downloads# adduser tom
Adding user `tom' ...
Adding new group `tom' (1000) ...
Adding new user `tom' (1000) with group `tom' ...
Creating home directory `/home/tom' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for tom
Enter the new value, or press ENTER for the default
        Full Name []: tom
        Room Number []: qw
        Work Phone []: qw
        Home Phone []: qw
        Other []: qw
Is the information correct? [Y/n] y
root@localhost:~/Downloads#
```

- Tom has been created as a normal user of which logs are maintained and stored in the **/var/log/auth.log** file. To check the logs we run the command,
  # cat var/log/auth.log



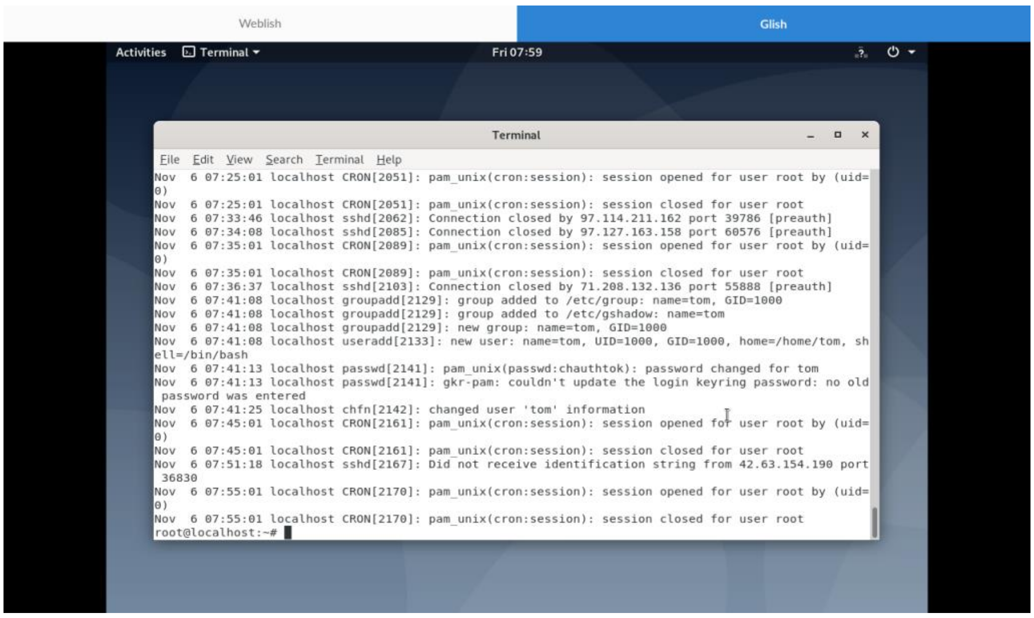
```
Nov  6 07:21:32 localhost sshd[2047]: pam_unix(sshd:auth): check pass; user unknown
Nov  6 07:21:32 localhost sshd[2047]: pam_unix(sshd:auth): authentication failure; logname= uid=
0 euid=0 tty=ssh ruser= rhost=185.212.149.160
Nov  6 07:21:33 localhost sshd[2047]: Failed password for invalid user admin from 185.212.149.16
0 port 49210 ssh2
Nov  6 07:21:35 localhost sshd[2047]: Connection closed by invalid user admin 185.212.149.160 po
rt 49210 [preauth]
Nov  6 07:25:01 localhost CRON[2051]: pam_unix(cron:session): session opened for user root by (u
id=0)
Nov  6 07:25:01 localhost CRON[2051]: pam_unix(cron:session): session closed for user root
Nov  6 07:33:46 localhost sshd[2062]: Connection closed by 97.114.211.162 port 39786 [preauth]
Nov  6 07:34:08 localhost sshd[2085]: Connection closed by 97.127.163.158 port 60576 [preauth]
Nov  6 07:35:01 localhost CRON[2089]: pam_unix(cron:session): session opened for user root by (u
id=0)
Nov  6 07:35:01 localhost CRON[2089]: pam_unix(cron:session): session closed for user root
Nov  6 07:36:37 localhost sshd[2103]: Connection closed by 71.208.132.136 port 55888 [preauth]
Nov  6 07:41:08 localhost groupadd[2129]: group added to /etc/group: name=tom, GID=1000
Nov  6 07:41:08 localhost groupadd[2129]: group added to /etc/gshadow: name=tom
Nov  6 07:41:08 localhost groupadd[2129]: new group: name=tom, GID=1000
Nov  6 07:41:08 localhost useradd[2133]: new user: name=tom, UID=1000, GID=1000, home=/home/tom,
 shell=/bin/bash
Nov  6 07:41:13 localhost passwd[2141]: pam_unix(passwd:chauthtok): password changed for tom
Nov  6 07:41:13 localhost passwd[2141]: gkr-pam: couldn't update the login keyring password: no
old password was entered
Nov  6 07:41:25 localhost chfn[2142]: changed user 'tom' information
root@localhost:~#
```

- This log is then fetched by SIEMs like IBM Qradar. There is a potential flaw in such logging. We will explore it as we break down this scenario.
- Next, using the **# vim /etc/passwd** command we add another user like "**tom**" and name that user as "**henry**". The **privileges** for henry can be changed by editing his (**UID**) user identifier and his (**GID**) group identifier to **root** by **0**. We also change henry's directory to **root**.



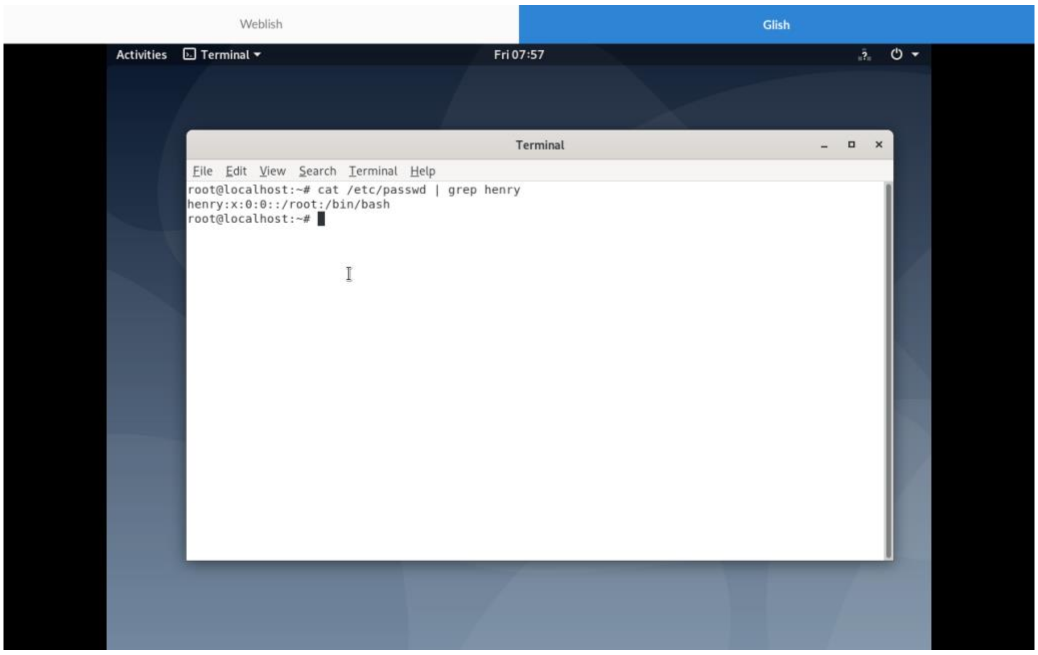
```
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
tss:x:107:112:TPM2 software stack,,,:/var/lib/tpm:/bin/false
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi:x:110:115:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
geoclue:x:112:116::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:113:118:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
saned:x:114:120::/var/lib/saned:/usr/sbin/nologin
colord:x:115:121:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
Debian-gdm:x:116:122:Gnome Display Manager:/var/lib/gdm3:/bin/false
lightdm:x:117:123:Light Display Manager:/var/lib/lightdm:/bin/false
redis:x:118:124::/var/lib/redis:/usr/sbin/nologin
tom:x:1000:1000:tom,qw,qw,qw,qw:/home/tom:/bin/bash
henry:x:0:0::/root:/bin/bash
-- INSERT --                                                              40,28         Bot
```

- After creating henry we again view the **log file** to check if creation of user "**henry**" is logged by the system.



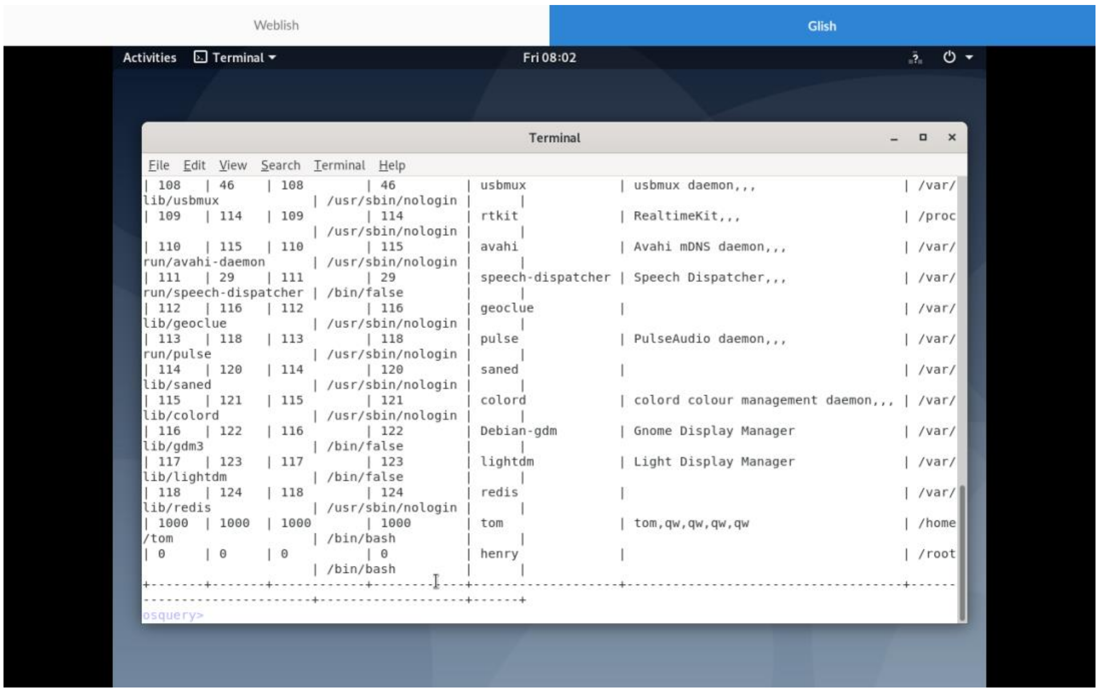
```
Nov  6 07:25:01 localhost CRON[2051]: pam_unix(cron:session): session opened for user root by (uid=
0)
Nov  6 07:25:01 localhost CRON[2051]: pam_unix(cron:session): session closed for user root
Nov  6 07:33:46 localhost sshd[2062]: Connection closed by 97.114.211.162 port 39786 [preauth]
Nov  6 07:34:08 localhost sshd[2085]: Connection closed by 97.127.163.158 port 60576 [preauth]
Nov  6 07:35:01 localhost CRON[2089]: pam_unix(cron:session): session opened for user root by (uid=
0)
Nov  6 07:35:01 localhost CRON[2089]: pam_unix(cron:session): session closed for user root
Nov  6 07:36:37 localhost sshd[2103]: Connection closed by 71.208.132.136 port 55888 [preauth]
Nov  6 07:41:08 localhost groupadd[2129]: group added to /etc/group: name=tom, GID=1000
Nov  6 07:41:08 localhost groupadd[2129]: group added to /etc/gshadow: name=tom
Nov  6 07:41:08 localhost groupadd[2129]: new group: name=tom, GID=1000
Nov  6 07:41:08 localhost useradd[2133]: new user: name=tom, UID=1000, GID=1000, home=/home/tom, sh
ell=/bin/bash
Nov  6 07:41:13 localhost passwd[2141]: pam_unix(passwd:chauthtok): password changed for tom
Nov  6 07:41:13 localhost passwd[2141]: gkr-pam: couldn't update the login keyring password: no old
 password was entered
Nov  6 07:41:25 localhost chfn[2142]: changed user 'tom' information
Nov  6 07:45:01 localhost CRON[2161]: pam_unix(cron:session): session opened for user root by (uid=
0)
Nov  6 07:45:01 localhost CRON[2161]: pam_unix(cron:session): session closed for user root
Nov  6 07:51:18 localhost sshd[2167]: Did not receive identification string from 42.63.154.190 port
 36830
Nov  6 07:55:01 localhost CRON[2170]: pam_unix(cron:session): session opened for user root by (uid=
0)
Nov  6 07:55:01 localhost CRON[2170]: pam_unix(cron:session): session closed for user root
root@localhost:~#
```

- System doesn’t log the creation of **root** user henry. And therefore **auth.log** file doesn’t have that data and **SIEM Qradar** is unaware of this user creation.



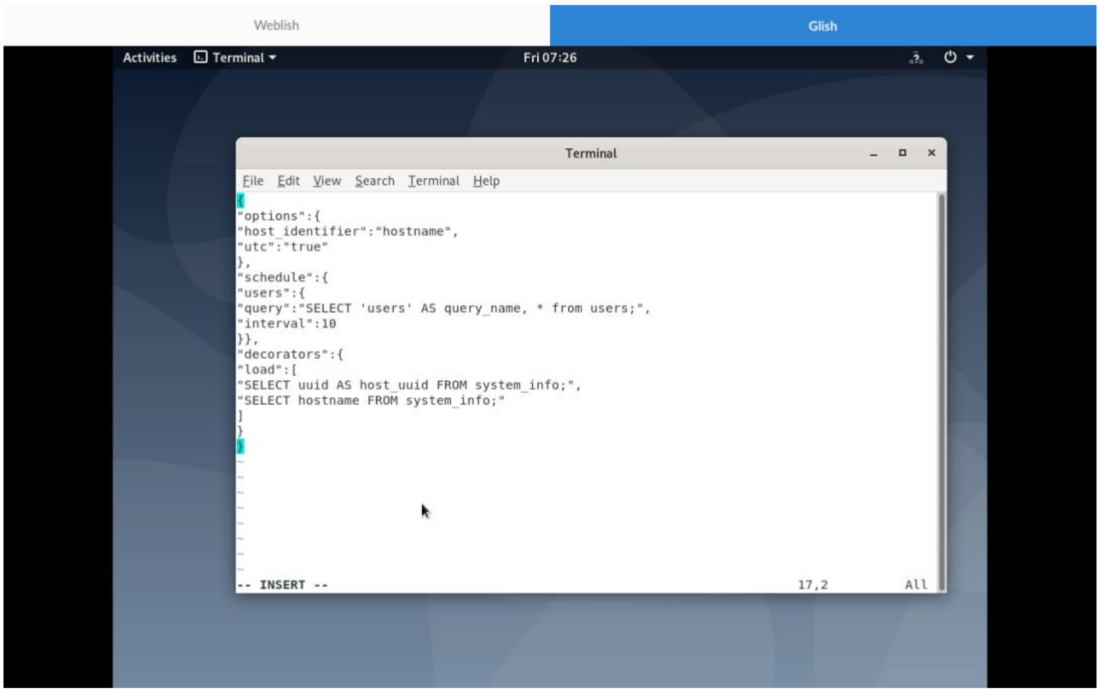
```
root@localhost:~# cat /etc/passwd | grep henry
henry:x:0:0::/root:/bin/bash
root@localhost:~# 
```

- Now, after installing OSQuery we can check the users table and it gives us the information on user “**henry**”.



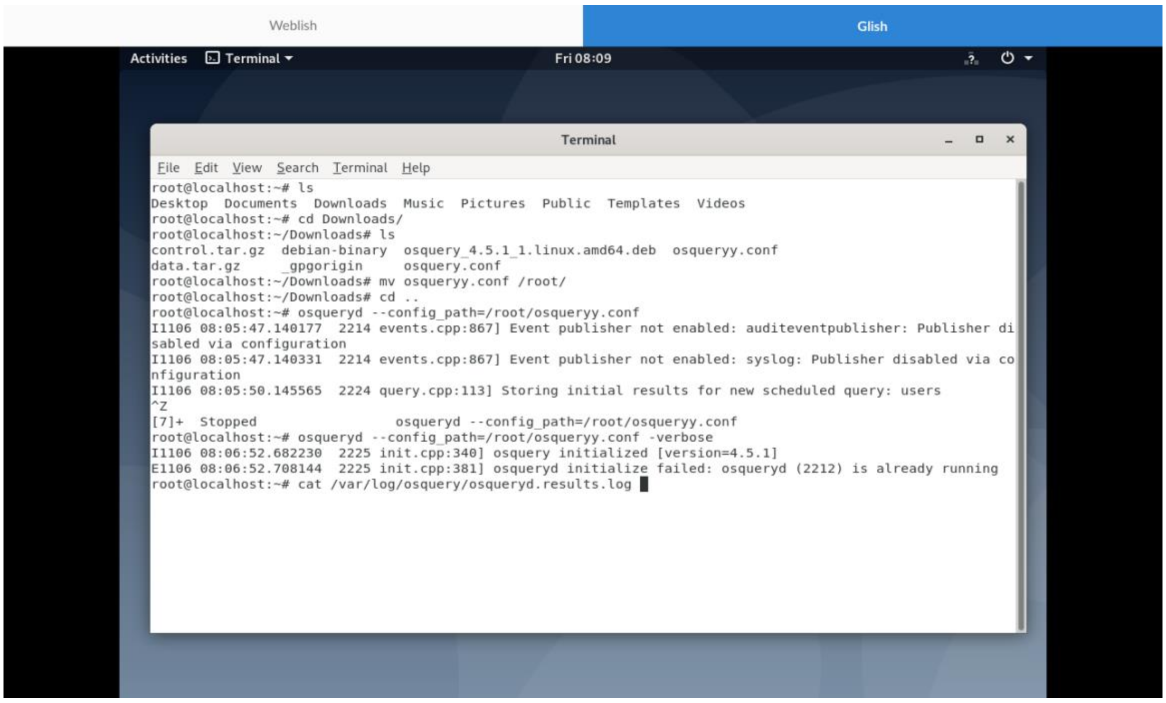
```
| 108    | 46     | 108          | 46      | usbmux            | usbmux daemon,,,                  | /var/
lib/usbmux            | /usr/sbin/nologin |       |
| 109    | 114    | 109          | 114     | rtkit             | RealtimeKit,,,                    | /proc
                      | /usr/sbin/nologin |       |
| 110    | 115    | 110          | 115     | avahi             | Avahi mDNS daemon,,,              | /var/
run/avahi-daemon      | /usr/sbin/nologin |       |
| 111    | 29     | 111          | 29      | speech-dispatcher | Speech Dispatcher,,,              | /var/
run/speech-dispatcher | /bin/false        |       |
| 112    | 116    | 112          | 116     | geoclue           |                                   | /var/
lib/geoclue           | /usr/sbin/nologin |       |
| 113    | 118    | 113          | 118     | pulse             | PulseAudio daemon,,,              | /var/
run/pulse             | /usr/sbin/nologin |       |
| 114    | 120    | 114          | 120     | saned             |                                   | /var/
lib/saned             | /usr/sbin/nologin |       |
| 115    | 121    | 115          | 121     | colord            | colord colour management daemon,,, | /var/
lib/colord            | /usr/sbin/nologin |       |
| 116    | 122    | 116          | 122     | Debian-gdm        | Gnome Display Manager             | /var/
lib/gdm3              | /bin/false        |       |
| 117    | 123    | 117          | 123     | lightdm           | Light Display Manager             | /var/
lib/lightdm           | /bin/false        |       |
| 118    | 124    | 118          | 124     | redis             |                                   | /var/
lib/redis             | /usr/sbin/nologin |       |
| 1000   | 1000   | 1000         | 1000    | tom               | tom,qw,qw,qw,qw                   | /home
/tom                  | /bin/bash         |       |
| 0      | 0      | 0            | 0       | henry             |                                   | /root
                      | /bin/bash         |       |
+--------+--------+-------------+---------+-------------------+-----------------------------------+------
----------------------+-------------------+-------+
osquery>
```

- To log the creation of users through OSQuery we can create a **.conf** file which will log the user data. Following is the script for that configuration file.



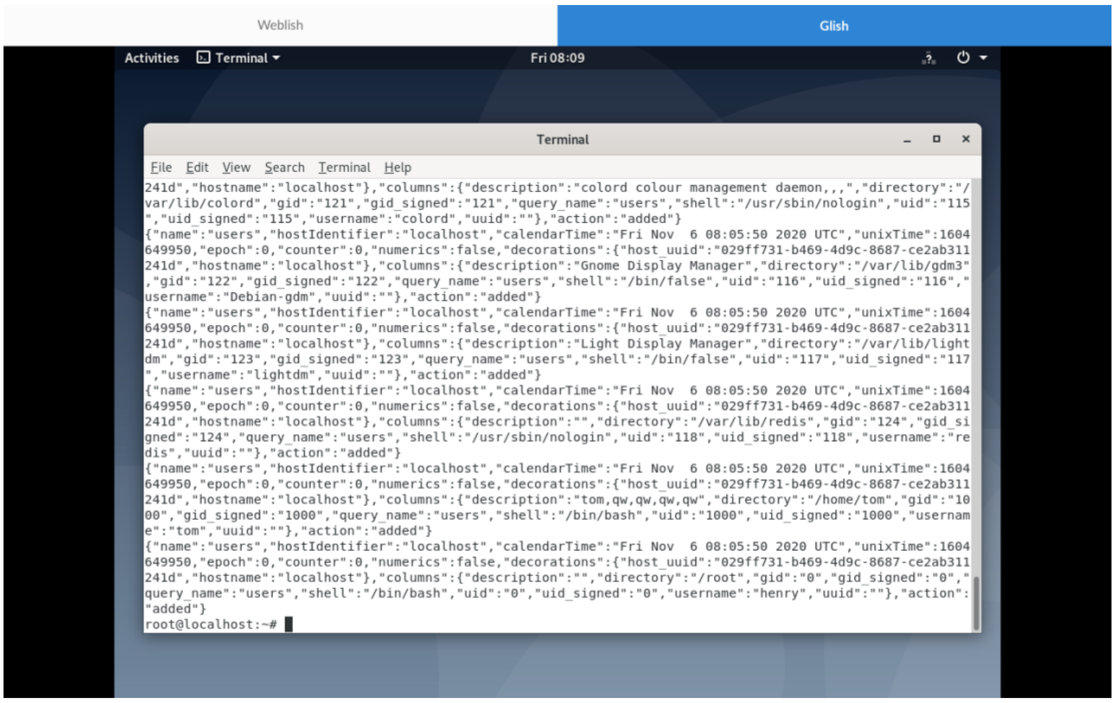
```
{
"options":{
"host_identifier":"hostname",
"utc":"true"
},
"schedule":{
"users":{
"query":"SELECT 'users' AS query_name, * from users;",
"interval":10
}},
"decorators":{
"load":[
"SELECT uuid AS host_uuid FROM system_info;",
"SELECT hostname FROM system_info;"
]
}
}
~
~
~
~
~
~
~
~
-- INSERT --                                                    17,2          All
```

- To display the results of that configuration file. We run the following command,
  # cat /var/log/osquery/osqueryd.results.log



```
root@localhost:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
root@localhost:~# cd Downloads/
root@localhost:~/Downloads# ls
control.tar.gz  debian-binary  osquery_4.5.1_1.linux.amd64.deb  osqueryy.conf
data.tar.gz     _gpgorigin     osquery.conf
root@localhost:~/Downloads# mv osqueryy.conf /root/
root@localhost:~/Downloads# cd ..
root@localhost:~# osqueryd --config_path=/root/osqueryy.conf
I1106 08:05:47.140177  2214 events.cpp:867] Event publisher not enabled: auditeventpublisher: Publisher di
sabled via configuration
I1106 08:05:47.140331  2214 events.cpp:867] Event publisher not enabled: syslog: Publisher disabled via co
nfiguration
I1106 08:05:50.145565  2224 query.cpp:113] Storing initial results for new scheduled query: users
^Z
[7]+  Stopped                 osqueryd --config_path=/root/osqueryy.conf
root@localhost:~# osqueryd --config_path=/root/osqueryy.conf -verbose
I1106 08:06:52.682230  2225 init.cpp:340] osquery initialized [version=4.5.1]
E1106 08:06:52.708144  2225 init.cpp:381] osqueryd initialize failed: osqueryd (2212) is already running
root@localhost:~# cat /var/log/osquery/osqueryd.results.log
```

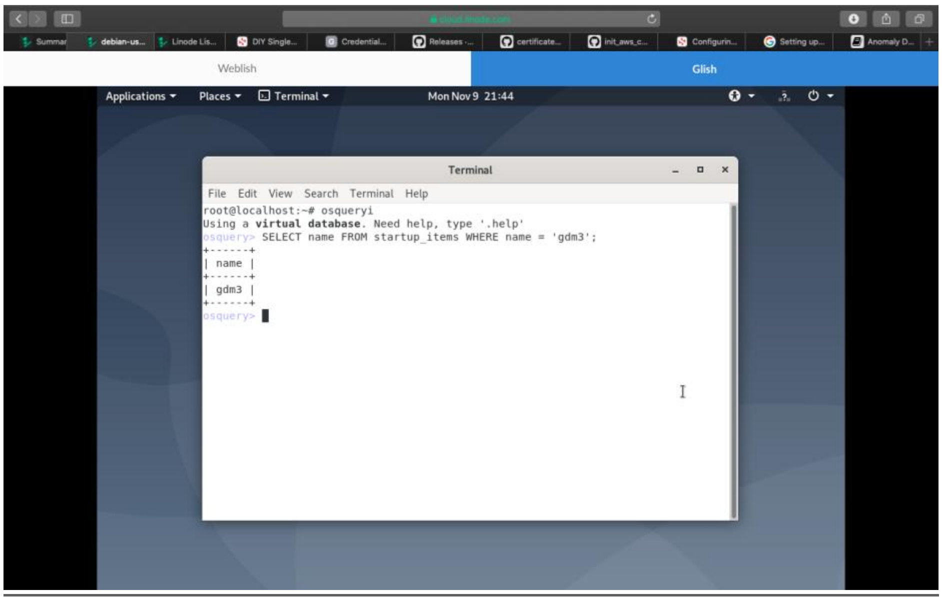
- Results of the OSQuery log file are as shown below. Which consists of the user “**henry**”. Thus, **SIEMs** can fetch this **log** files for better results.



```
241d","hostname":"localhost"},"columns":{"description":"colord colour management daemon,,,","directory":"/
var/lib/colord","gid":"121","gid_signed":"121","query_name":"users","shell":"/usr/sbin/nologin","uid":"115
","uid_signed":"115","username":"colord","uuid":""},"action":"added"}
{"name":"users","hostIdentifier":"localhost","calendarTime":"Fri Nov  6 08:05:50 2020 UTC","unixTime":1604
649950,"epoch":0,"counter":0,"numerics":false,"decorations":{"host_uuid":"029ff731-b469-4d9c-8687-ce2ab311
241d","hostname":"localhost"},"columns":{"description":"Gnome Display Manager","directory":"/var/lib/gdm3"
,"gid":"122","gid_signed":"122","query_name":"users","shell":"/bin/false","uid":"116","uid_signed":"116","
username":"Debian-gdm","uuid":""},"action":"added"}
{"name":"users","hostIdentifier":"localhost","calendarTime":"Fri Nov  6 08:05:50 2020 UTC","unixTime":1604
649950,"epoch":0,"counter":0,"numerics":false,"decorations":{"host_uuid":"029ff731-b469-4d9c-8687-ce2ab311
241d","hostname":"localhost"},"columns":{"description":"Light Display Manager","directory":"/var/lib/light
dm","gid":"123","gid_signed":"123","query_name":"users","shell":"/bin/false","uid":"117","uid_signed":"117
","username":"lightdm","uuid":""},"action":"added"}
{"name":"users","hostIdentifier":"localhost","calendarTime":"Fri Nov  6 08:05:50 2020 UTC","unixTime":1604
649950,"epoch":0,"counter":0,"numerics":false,"decorations":{"host_uuid":"029ff731-b469-4d9c-8687-ce2ab311
241d","hostname":"localhost"},"columns":{"description":"","directory":"/var/lib/redis","gid":"124","gid_si
gned":"124","query_name":"users","shell":"/usr/sbin/nologin","uid":"118","uid_signed":"118","username":"re
dis","uuid":""},"action":"added"}
{"name":"users","hostIdentifier":"localhost","calendarTime":"Fri Nov  6 08:05:50 2020 UTC","unixTime":1604
649950,"epoch":0,"counter":0,"numerics":false,"decorations":{"host_uuid":"029ff731-b469-4d9c-8687-ce2ab311
241d","hostname":"localhost"},"columns":{"description":"tom,qw,qw,qw,qw","directory":"/home/tom","gid":"10
00","gid_signed":"1000","query_name":"users","shell":"/bin/bash","uid":"1000","uid_signed":"1000","usernam
e":"tom","uuid":""},"action":"added"}
{"name":"users","hostIdentifier":"localhost","calendarTime":"Fri Nov  6 08:05:50 2020 UTC","unixTime":1604
649950,"epoch":0,"counter":0,"numerics":false,"decorations":{"host_uuid":"029ff731-b469-4d9c-8687-ce2ab311
241d","hostname":"localhost"},"columns":{"description":"","directory":"/root","gid":"0","gid_signed":"0","
query_name":"users","shell":"/bin/bash","uid":"0","uid_signed":"0","username":"henry","uuid":""},"action":
"added"}
root@localhost:~#
```

2. <u>Scenario (Anomaly detection with OSQuery)</u>

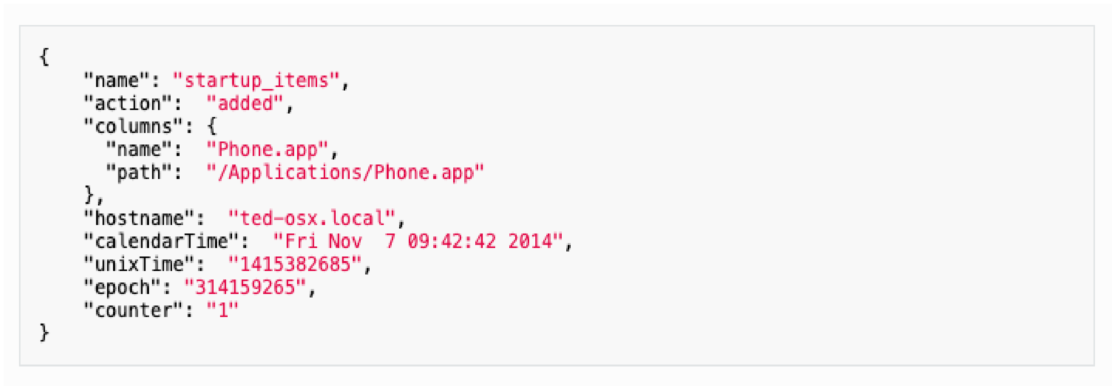
- OSQuery deployment will help you set up an infrastructural basis that helps you to identify **malicious** behavior using **scheduled** queries.



```
root@localhost:~# osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT name FROM startup_items WHERE name = 'gdm3';
+------+
| name |
+------+
| gdm3 |
+------+
osquery>
```

- We can see **services** like **gdm3** run at boot time. Now, considering the system is compromised at a later date or time.
  So, we can tackle such situations by utilizing the **log aggregation** capabilities of OSQuery to quickly determine when the **incident** happened and what was installed or added.

- Using the log aggregation guide we can receive following log lines in our SIEM's datastore.



```
{
    "name": "startup_items",
    "action":  "added",
    "columns": {
      "name":  "Phone.app",
      "path":  "/Applications/Phone.app"
    },
    "hostname":  "ted-osx.local",
    "calendarTime":  "Fri Nov  7 09:42:42 2014",
    "unixTime":  "1415382685",
    "epoch": "314159265",
    "counter": "1"
}
```

- It becomes evident and clear that a **potentially harmful** application termed "Phone" was added to the **startup_items** at 09:42:42 on Friday November 7, 2014.

REFERENCES & USEFUL LINKS:

- *OSQuery Deployment Docs - https://osquery.readthedocs.io/en/stable/deployment/anomaly-detection/*
- *Video playlist on OSQuery integration with Uptycs – https://www.youtube.com/playlist?list=PL6-FgoWOoK2a-aLNvJ3YbR48ra9K98Yno*
- *Video playlist on Logging with OSQuery - https://www.youtube.com/playlist?list=PLHh9jhztlMyp4B7cbTanmCj2DYU6Qc3On*
- *OSQuery Repository on Git – https://github.com/osquery*