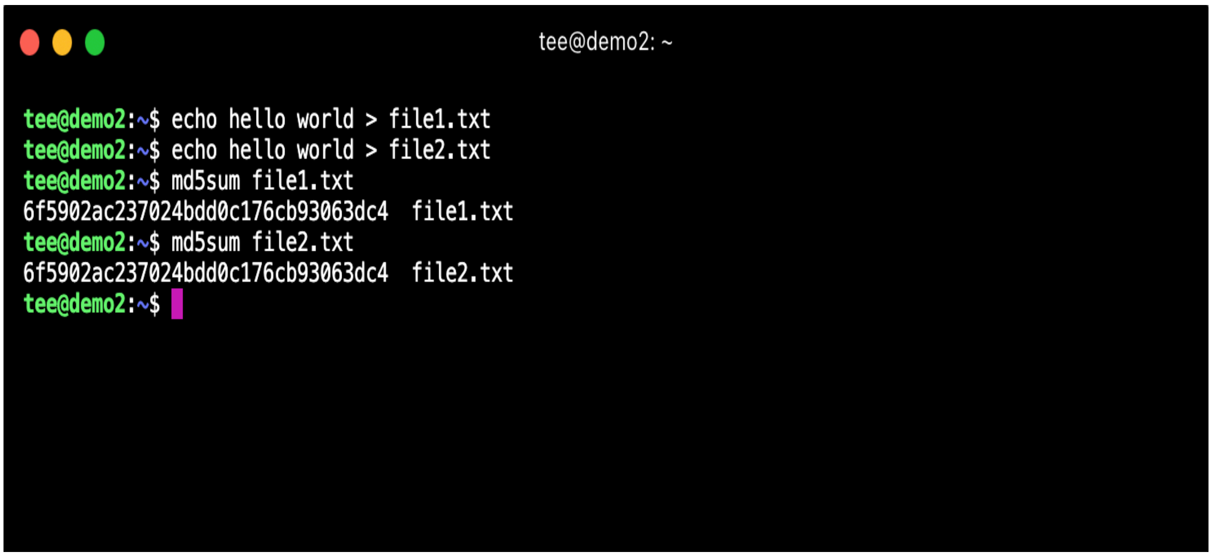
Encoding v Hashing v Encryption

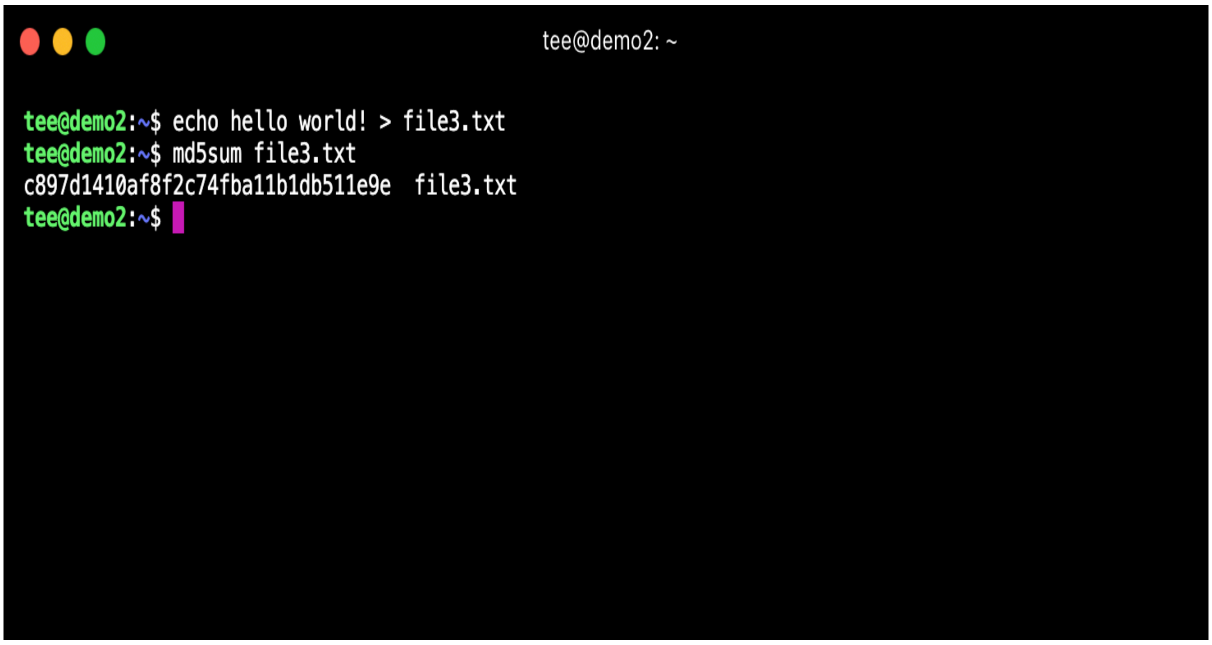
Exercise 2 - Hashing

Task 1 in this exercise is to create three files; file1.txt and file2.txt. The content of these files will be “hello world”. When these files have been created, both will be hashed using the ‘md5sum’ command to establish that they have the same hash because they contain the exact same content.



```
tee@demo2: ~

tee@demo2:~$ echo hello world > file1.txt
tee@demo2:~$ echo hello world > file2.txt
tee@demo2:~$ md5sum file1.txt
6f5902ac237024bdd0c176cb93063dc4  file1.txt
tee@demo2:~$ md5sum file2.txt
6f5902ac237024bdd0c176cb93063dc4  file2.txt
tee@demo2:~$
```

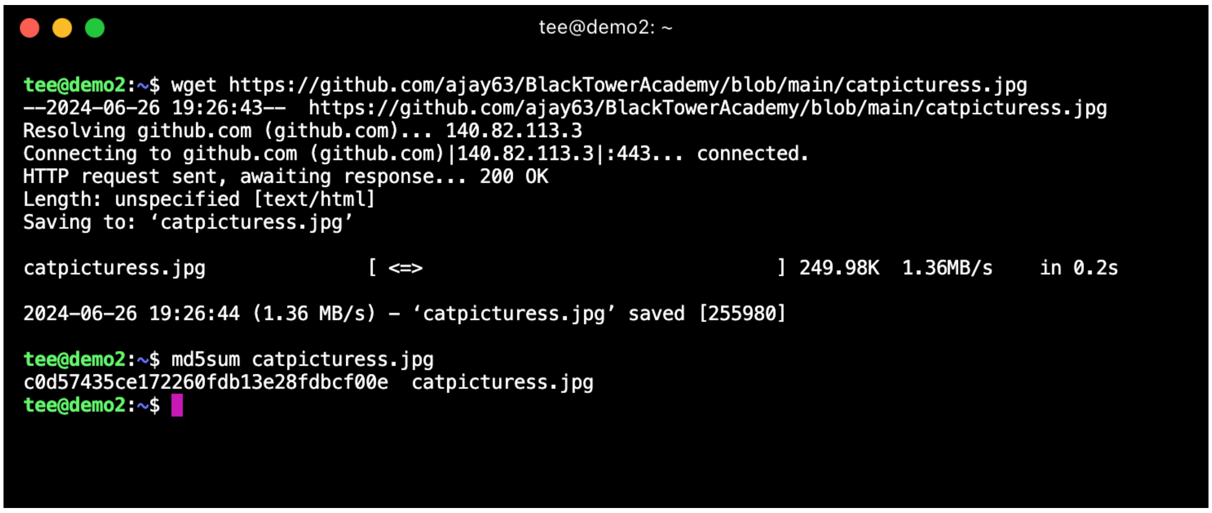
Task 2 is to create a third file; file3.txt. The content of this file will be “hello world!’ The only difference between the content of this file compared to file1.txt and file2.txt is the inclusion of ‘!’ in the text line. After this file has been created, it’ll be hashed. The avalanche effect of the hashing technology ensures that regardless of how minute the difference in the content of files are, there is a significant change in the hash of the files. This is illustrated below:



```
tee@demo2: ~

tee@demo2:~$ echo hello world! > file3.txt
tee@demo2:~$ md5sum file3.txt
c897d1410af8f2c74fba11b1db511e9e  file3.txt
tee@demo2:~$
```

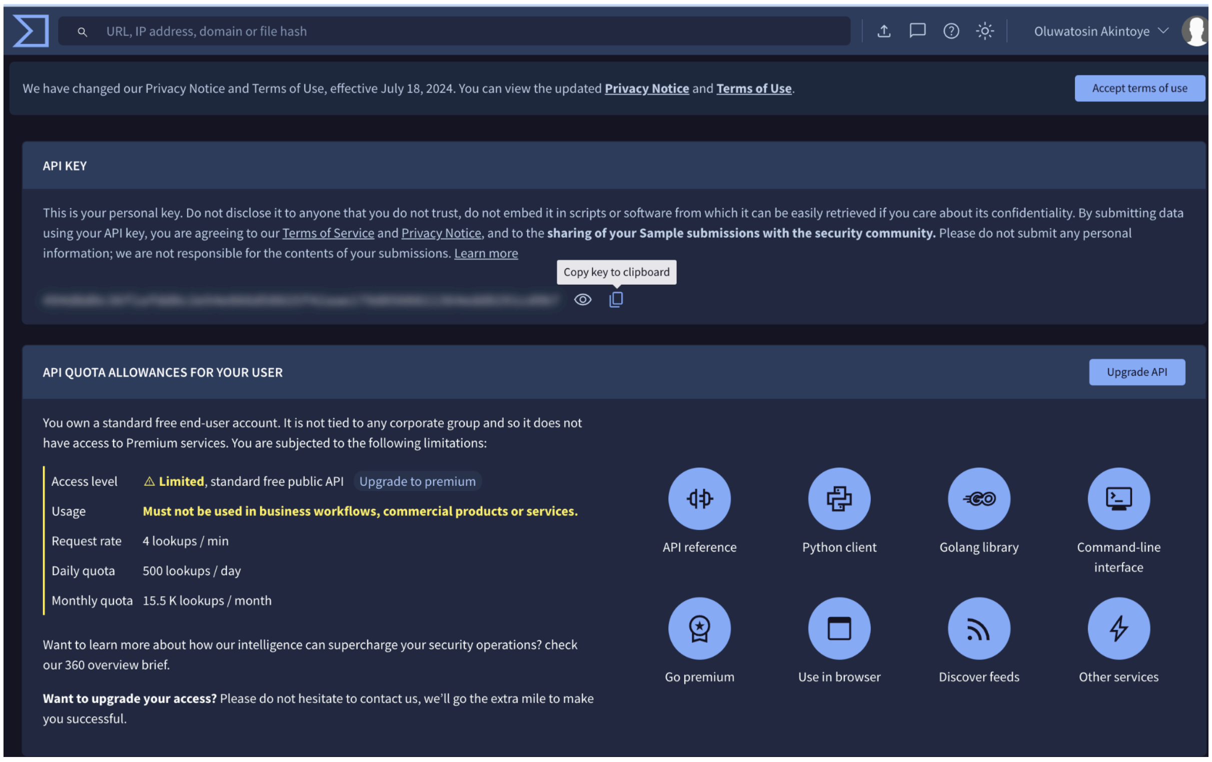
Hence, it can be concluded that a slight change in the content of two similar files will typically result in a drastic change in the hashes.

Task 3 is to download a cat picture in jpg format using the ‘wget’ command. When the picture has been downloaded successfully, it’ll be hashed. This is illustrated below:



```
tee@demo2: ~

tee@demo2:~$ wget https://github.com/ajay63/BlackTowerAcademy/blob/main/catpicturess.jpg
--2024-06-26 19:26:43--  https://github.com/ajay63/BlackTowerAcademy/blob/main/catpicturess.jpg
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘catpicturess.jpg’

catpicturess.jpg             [ <=>                              ] 249.98K  1.36MB/s    in 0.2s

2024-06-26 19:26:44 (1.36 MB/s) - ‘catpicturess.jpg’ saved [255980]

tee@demo2:~$ md5sum catpicturess.jpg
c0d57435ce172260fdb13e28fdbcf00e  catpicturess.jpg
tee@demo2:~$
```

Task4 is to set up a virustotal account going to the link; https://www.virustotal.com. Once this is done, the API key linked to the newly created account should be copied to the clipboard from the virustotal profile. This is illustrated below:



We have changed our Privacy Notice and Terms of Use, effective July 18, 2024. You can view the updated **Privacy Notice** and **Terms of Use**.

API KEY

This is your personal key. Do not disclose it to anyone that you do not trust, do not embed it in scripts or software from which it can be easily retrieved if you care about its confidentiality. By submitting data using your API key, you are agreeing to our Terms of Service and Privacy Notice, and to the **sharing of your Sample submissions with the security community.** Please do not submit any personal information; we are not responsible for the contents of your submissions. Learn more

[redacted]

API QUOTA ALLOWANCES FOR YOUR USER

You own a standard free end-user account. It is not tied to any corporate group and so it does not have access to Premium services. You are subjected to the following limitations:

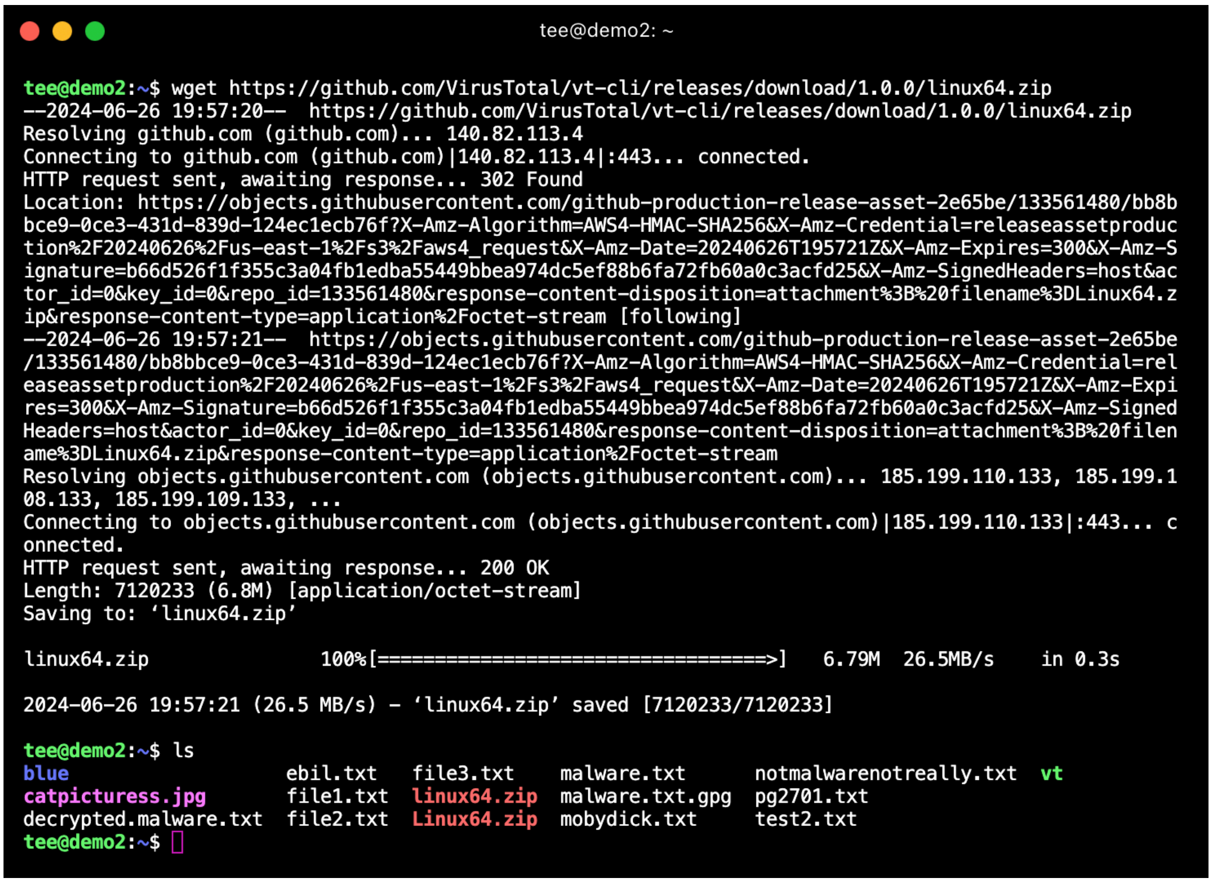
Access level	⚠ **Limited**, standard free public API Upgrade to premium
Usage	**Must not be used in business workflows, commercial products or services.**
Request rate	4 lookups / min
Daily quota	500 lookups / day
Monthly quota	15.5 K lookups / month

Want to learn more about how our intelligence can supercharge your security operations? check our 360 overview brief.

Want to upgrade your access? Please do not hesitate to contact us, we'll go the extra mile to make you successful.

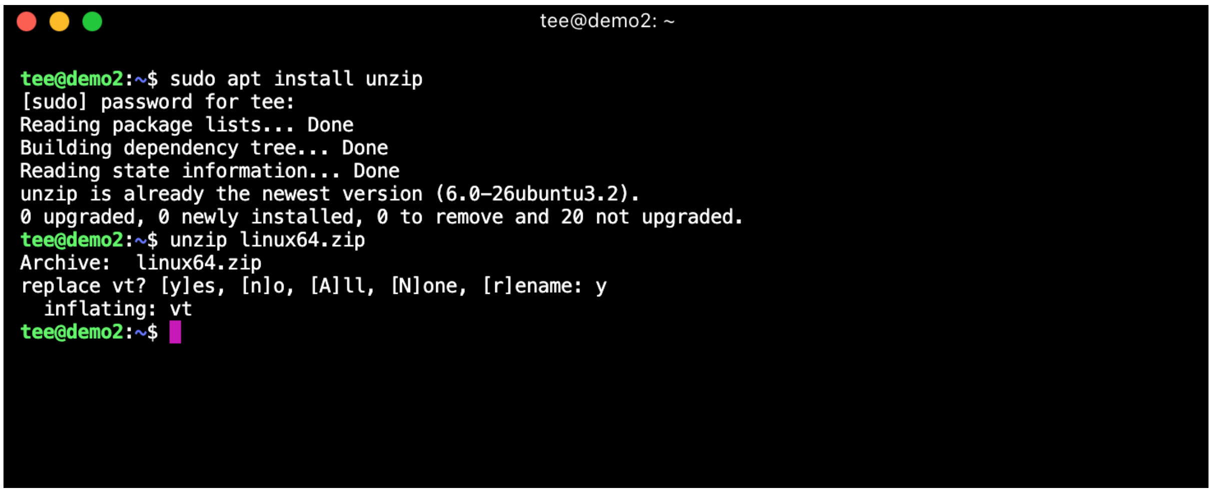
Once the API key has been copied to the clipboard, the next step is to integrate virustotal into the Linux command line. To achieve this, the following steps should be carried out:

Task 1 involves downloading the pre-compiled Virustotal Application using the ‘wget’ command. This is illustrated below:



```
tee@demo2: ~

tee@demo2:~$ wget https://github.com/VirusTotal/vt-cli/releases/download/1.0.0/linux64.zip
--2024-06-26 19:57:20--  https://github.com/VirusTotal/vt-cli/releases/download/1.0.0/linux64.zip
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/133561480/bb8b
bce9-0ce3-431d-839d-124ec1ecb76f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduc
tion%2F20240626%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240626T195721Z&X-Amz-Expires=300&X-Amz-S
ignature=b66d526f1f355c3a04fb1edba55449bbea974dc5ef88b6fa72fb60a0c3acfd25&X-Amz-SignedHeaders=host&ac
tor_id=0&key_id=0&repo_id=133561480&response-content-disposition=attachment%3B%20filename%3DLinux64.z
ip&response-content-type=application%2Foctet-stream [following]
--2024-06-26 19:57:21--  https://objects.githubusercontent.com/github-production-release-asset-2e65be
/133561480/bb8bbce9-0ce3-431d-839d-124ec1ecb76f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=rel
easeassetproduction%2F20240626%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240626T195721Z&X-Amz-Expi
res=300&X-Amz-Signature=b66d526f1f355c3a04fb1edba55449bbea974dc5ef88b6fa72fb60a0c3acfd25&X-Amz-Signed
Headers=host&actor_id=0&key_id=0&repo_id=133561480&response-content-disposition=attachment%3B%20filen
ame%3DLinux64.zip&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.1
08.133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 7120233 (6.8M) [application/octet-stream]
Saving to: ‘linux64.zip’

linux64.zip              100%[===================================>]   6.79M  26.5MB/s    in 0.3s

2024-06-26 19:57:21 (26.5 MB/s) - ‘linux64.zip’ saved [7120233/7120233]

tee@demo2:~$ ls
blue                  ebil.txt   file3.txt    malware.txt      notmalwarenotreally.txt  vt
catpicturess.jpg      file1.txt  linux64.zip  malware.txt.gpg  pg2701.txt
decrypted.malware.txt file2.txt  Linux64.zip  mobydick.txt     test2.txt
tee@demo2:~$ 
```

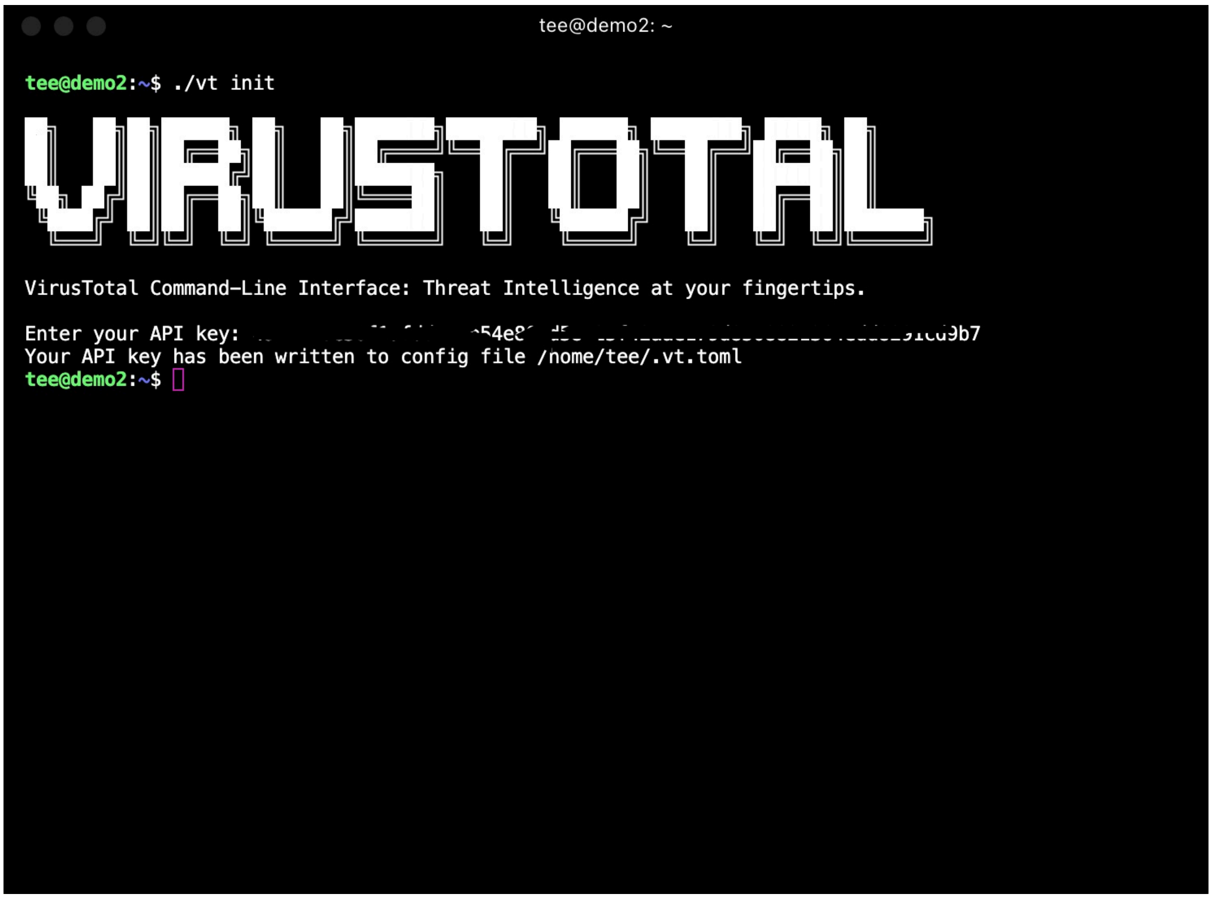
To unzip this zip file, use the following command ‘sudo apt install unzip -y’ to install unzip, then run the following command to unzip the downloaded file. This is illustrated below:



```
tee@demo2: ~

tee@demo2:~$ sudo apt install unzip
[sudo] password for tee:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unzip is already the newest version (6.0-26ubuntu3.2).
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
tee@demo2:~$ unzip linux64.zip
Archive:  linux64.zip
replace vt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: vt
tee@demo2:~$ 
```

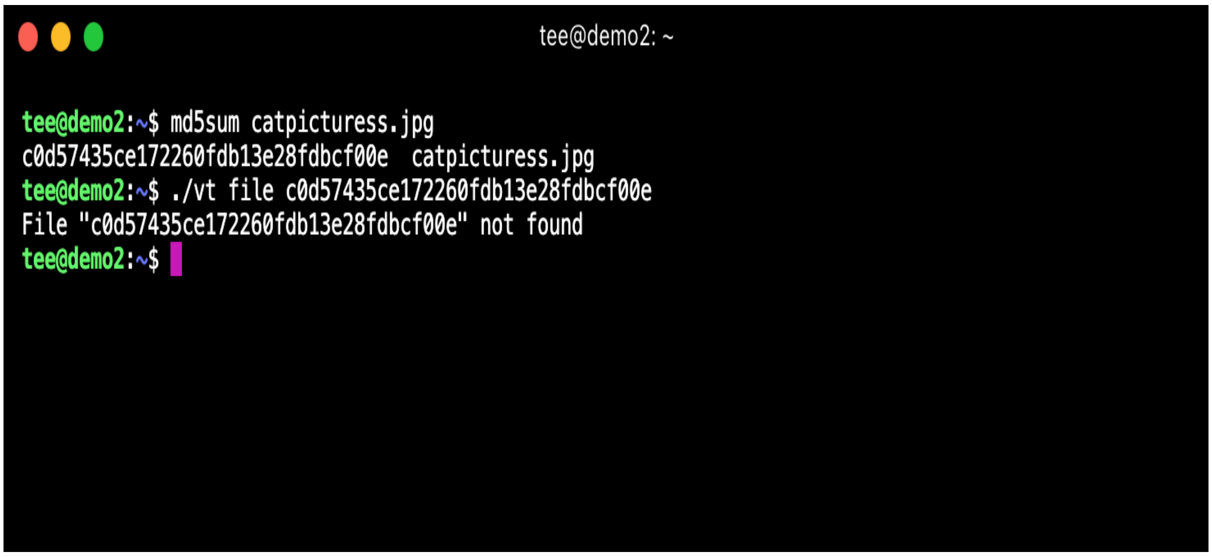
Next, initiate the virustotal application (vt). To do this use the following command: ./vt init.

The result is illustrated below:



```
tee@demo2: ~

tee@demo2:~$ ./vt init
VIRUSTOTAL

VirusTotal Command-Line Interface: Threat Intelligence at your fingertips.

Enter your API key: [redacted]
Your API key has been written to config file /nome/tee/.vt.toml
tee@demo2:~$
```

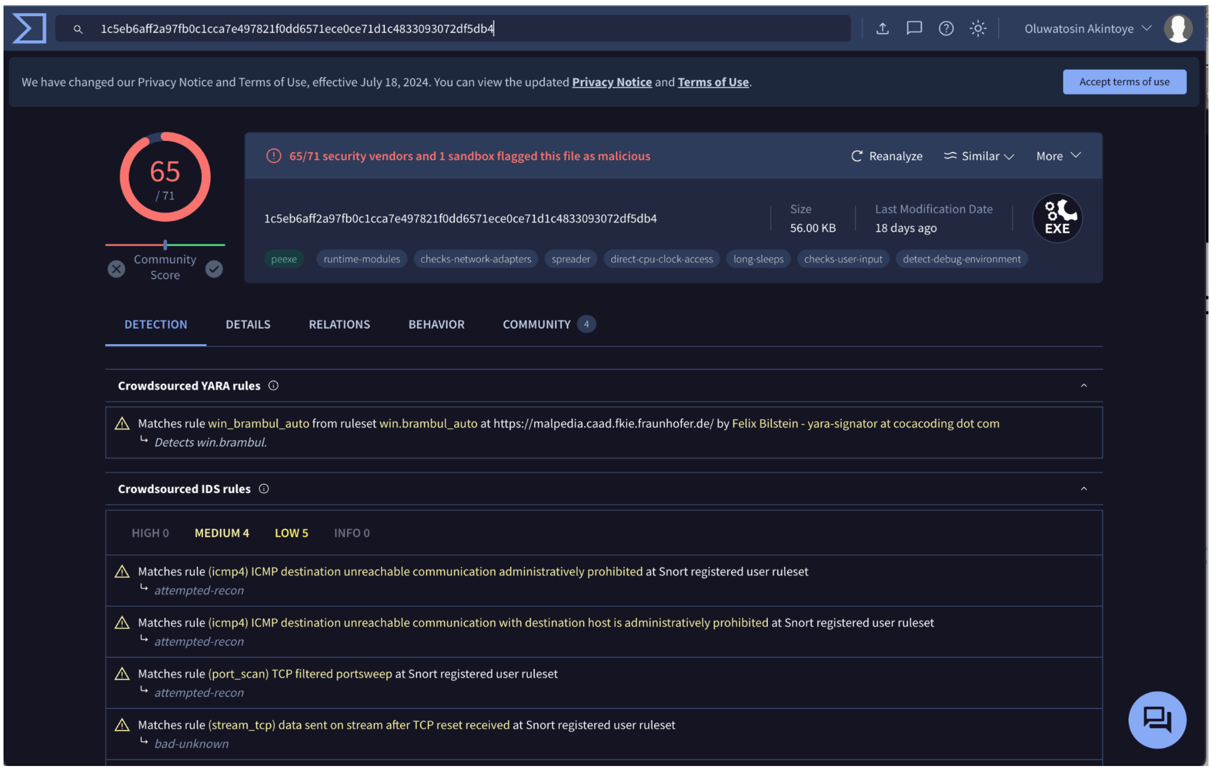
Now that the vt application is installed, it is ready to be used. The next step is to hash the cat picture file; catpicturess.jpg, using the md5sum command. This is shown below:



```
tee@demo2: ~

tee@demo2:~$ md5sum catpicturess.jpg
c0d57435ce172260fdb13e28fdbcf00e  catpicturess.jpg
tee@demo2:~$ ./vt file c0d57435ce172260fdb13e28fdbcf00e
File "c0d57435ce172260fdb13e28fdbcf00e" not found
tee@demo2:~$
```

The output shows that the catpicturess.jpg file is not a virus according to Virus Total.

The final step is to search for a known bad hash on the VirusTotal webpage and search another hash through the CLI. These two steps are illustrated below:



1c5eb6aff2a97fb0c1cca7e497821f0dd6571ece0ce71d1c4833093072df5db4

Oluwatosin Akintoye

We have changed our Privacy Notice and Terms of Use, effective July 18, 2024. You can view the updated **Privacy Notice** and **Terms of Use**.

Accept terms of use

65 / 71

Community Score

65/71 security vendors and 1 sandbox flagged this file as malicious

Reanalyze · Similar · More

1c5eb6aff2a97fb0c1cca7e497821f0dd6571ece0ce71d1c4833093072df5db4

Size: 56.00 KB

Last Modification Date: 18 days ago

EXE

peexe · runtime-modules · checks-network-adapters · spreader · direct-cpu-clock-access · long-sleeps · checks-user-input · detect-debug-environment

DETECTION · DETAILS · RELATIONS · BEHAVIOR · COMMUNITY 4

Crowdsourced YARA rules

Matches rule win_brambul_auto from ruleset win.brambul_auto at https://malpedia.caad.fkie.fraunhofer.de/ by Felix Bilstein - yara-signator at cocacoding dot com
↳ *Detects win.brambul.*

Crowdsourced IDS rules

HIGH 0 · MEDIUM 4 · LOW 5 · INFO 0

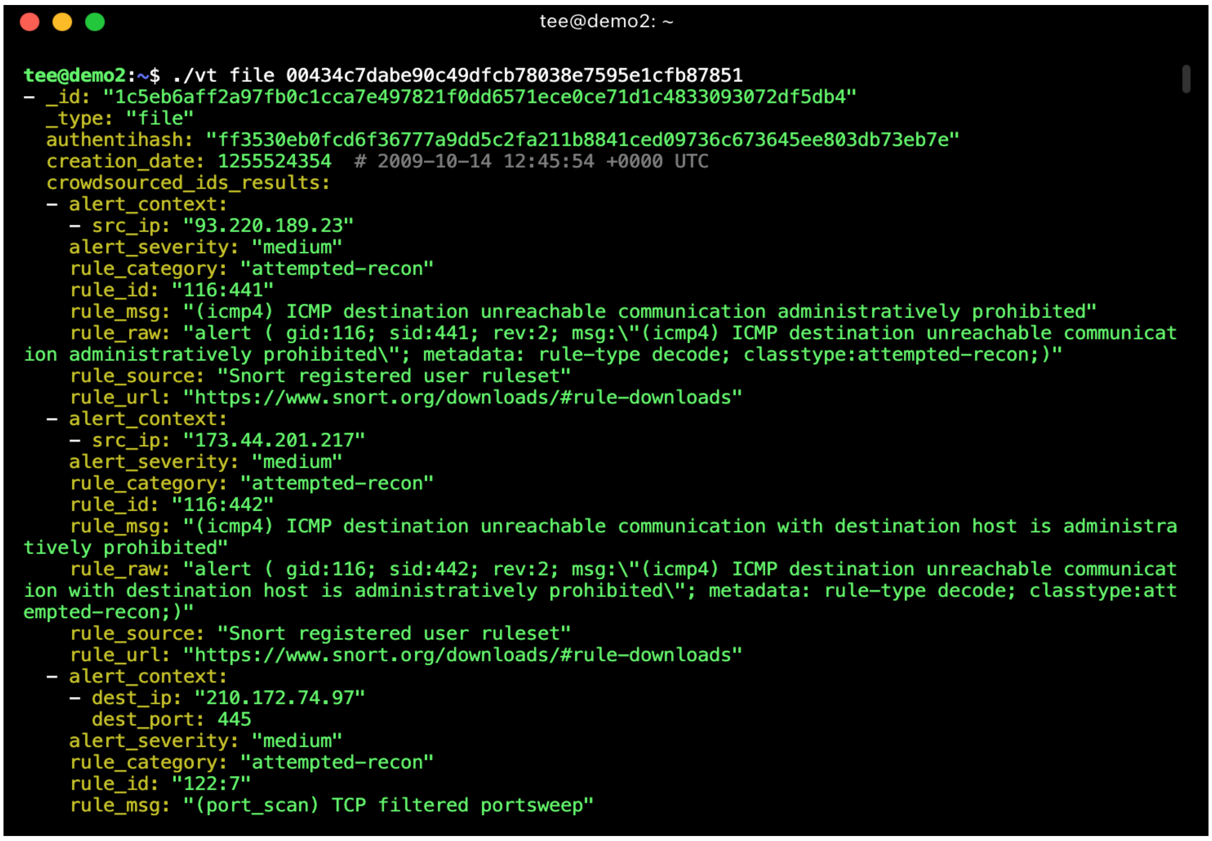
Matches rule (icmp4) ICMP destination unreachable communication administratively prohibited at Snort registered user ruleset
↳ *attempted-recon*

Matches rule (icmp4) ICMP destination unreachable communication with destination host is administratively prohibited at Snort registered user ruleset
↳ *attempted-recon*

Matches rule (port_scan) TCP filtered portsweep at Snort registered user ruleset
↳ *attempted-recon*

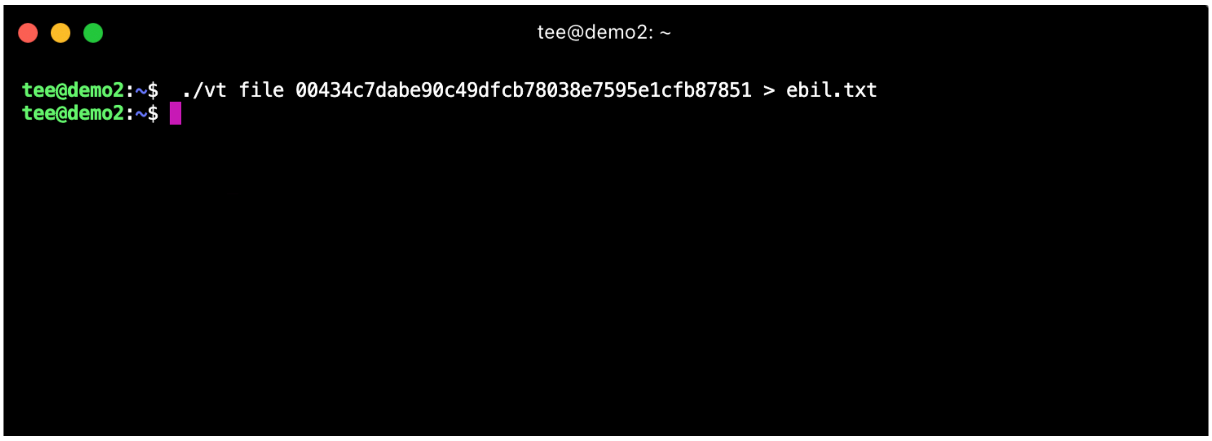
Matches rule (stream_tcp) data sent on stream after TCP reset received at Snort registered user ruleset
↳ *bad-unknown*

As shown above, just searching for a hash in the VirusTotal webpage can show if a file is malicious or not. To do this in the CLI, use the following command: ./vt file 00434c7dabe90c49dfcb78038e7595e1cfb87851. This is illustrated below:



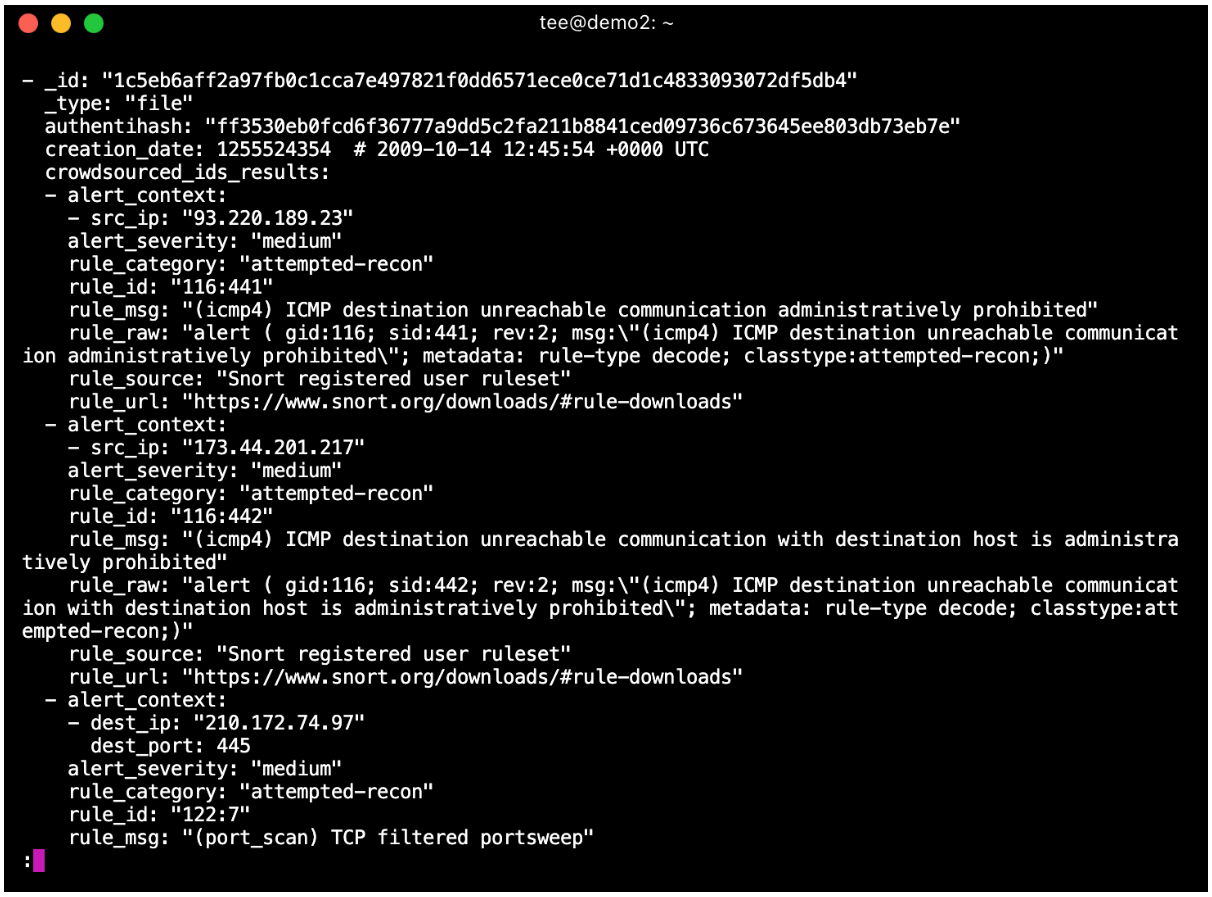
```
tee@demo2: ~

tee@demo2:~$ ./vt file 00434c7dabe90c49dfcb78038e7595e1cfb87851
- _id: "1c5eb6aff2a97fb0c1cca7e497821f0dd6571ece0ce71d1c4833093072df5db4"
  _type: "file"
  authentihash: "ff3530eb0fcd6f36777a9dd5c2fa211b8841ced09736c673645ee803db73eb7e"
  creation_date: 1255524354  # 2009-10-14 12:45:54 +0000 UTC
  crowdsourced_ids_results:
  - alert_context:
    - src_ip: "93.220.189.23"
    alert_severity: "medium"
    rule_category: "attempted-recon"
    rule_id: "116:441"
    rule_msg: "(icmp4) ICMP destination unreachable communication administratively prohibited"
    rule_raw: "alert ( gid:116; sid:441; rev:2; msg:\"(icmp4) ICMP destination unreachable communicat
ion administratively prohibited\"; metadata: rule-type decode; classtype:attempted-recon;)"
    rule_source: "Snort registered user ruleset"
    rule_url: "https://www.snort.org/downloads/#rule-downloads"
  - alert_context:
    - src_ip: "173.44.201.217"
    alert_severity: "medium"
    rule_category: "attempted-recon"
    rule_id: "116:442"
    rule_msg: "(icmp4) ICMP destination unreachable communication with destination host is administra
tively prohibited"
    rule_raw: "alert ( gid:116; sid:442; rev:2; msg:\"(icmp4) ICMP destination unreachable communicat
ion with destination host is administratively prohibited\"; metadata: rule-type decode; classtype:att
empted-recon;)"
    rule_source: "Snort registered user ruleset"
    rule_url: "https://www.snort.org/downloads/#rule-downloads"
  - alert_context:
    - dest_ip: "210.172.74.97"
      dest_port: 445
    alert_severity: "medium"
    rule_category: "attempted-recon"
    rule_id: "122:7"
    rule_msg: "(port_scan) TCP filtered portsweep"
```

The output shown in the illustration above indicates that the file is malicious. To redirect the voluminous output of this scanned file to a file, use the following command: ./vt file 00434c7dabe90c49dfcb78038e7595e1cfb87851 > ebil.txt. This is illustrated below:



```
tee@demo2: ~

tee@demo2:~$  ./vt file 00434c7dabe90c49dfcb78038e7595e1cfb87851 > ebil.txt
tee@demo2:~$ 
```

To check the content of this ebil.txt file, any of the Linux text reader commands can be used; Less, More, Cat, Head, or Tail. It is most reasonable to use the 'less' command here as illustrated below:



```
tee@demo2: ~

- _id: "1c5eb6aff2a97fb0c1cca7e497821f0dd6571ece0ce71d1c4833093072df5db4"
  _type: "file"
  authentihash: "ff3530eb0fcd6f36777a9dd5c2fa211b8841ced09736c673645ee803db73eb7e"
  creation_date: 1255524354  # 2009-10-14 12:45:54 +0000 UTC
  crowdsourced_ids_results:
  - alert_context:
    - src_ip: "93.220.189.23"
    alert_severity: "medium"
    rule_category: "attempted-recon"
    rule_id: "116:441"
    rule_msg: "(icmp4) ICMP destination unreachable communication administratively prohibited"
    rule_raw: "alert ( gid:116; sid:441; rev:2; msg:\"(icmp4) ICMP destination unreachable communicat
ion administratively prohibited\"; metadata: rule-type decode; classtype:attempted-recon;)"
    rule_source: "Snort registered user ruleset"
    rule_url: "https://www.snort.org/downloads/#rule-downloads"
  - alert_context:
    - src_ip: "173.44.201.217"
    alert_severity: "medium"
    rule_category: "attempted-recon"
    rule_id: "116:442"
    rule_msg: "(icmp4) ICMP destination unreachable communication with destination host is administra
tively prohibited"
    rule_raw: "alert ( gid:116; sid:442; rev:2; msg:\"(icmp4) ICMP destination unreachable communicat
ion with destination host is administratively prohibited\"; metadata: rule-type decode; classtype:att
empted-recon;)"
    rule_source: "Snort registered user ruleset"
    rule_url: "https://www.snort.org/downloads/#rule-downloads"
  - alert_context:
    - dest_ip: "210.172.74.97"
      dest_port: 445
    alert_severity: "medium"
    rule_category: "attempted-recon"
    rule_id: "122:7"
    rule_msg: "(port_scan) TCP filtered portsweep"
:
```

This concludes the process of creating a file, checking the hash, confirming the avalanche effect, integrating VirusTotal in the Linux CLI, and scanning for malicious content using VT.