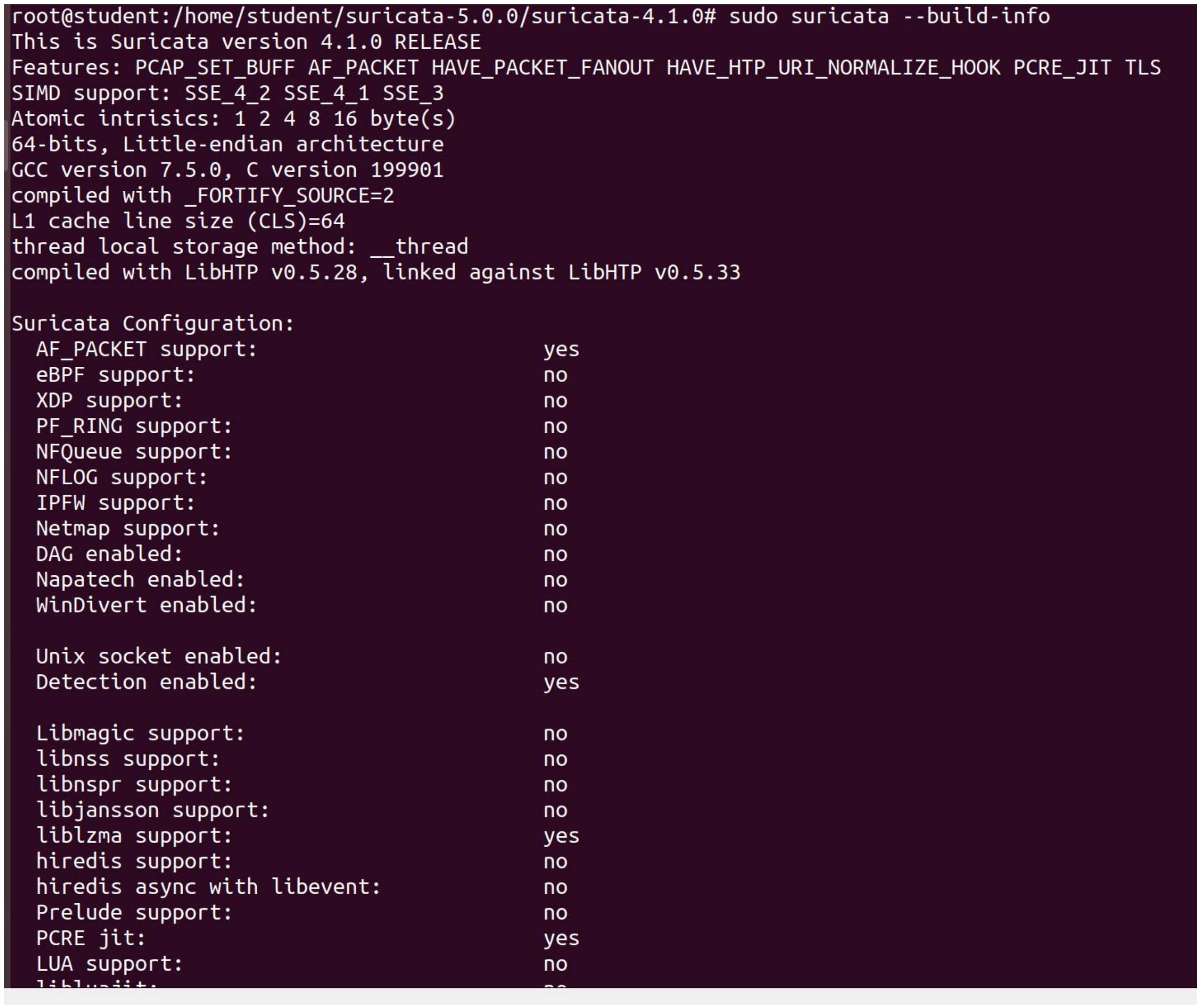
JHU IDS Module 5 Lab Activity
06/27/2020
Audrey Long

1. What is Suricata?
Continuing our deep-dive into network-based IDS's we are going to transition from Snort to another popular, open source NIDS called Suricata. You can read all about Suricata at https://suricata-ids.org/ and at https://suricata.readthedocs.io/en/suricata-4.1.0/index.html.

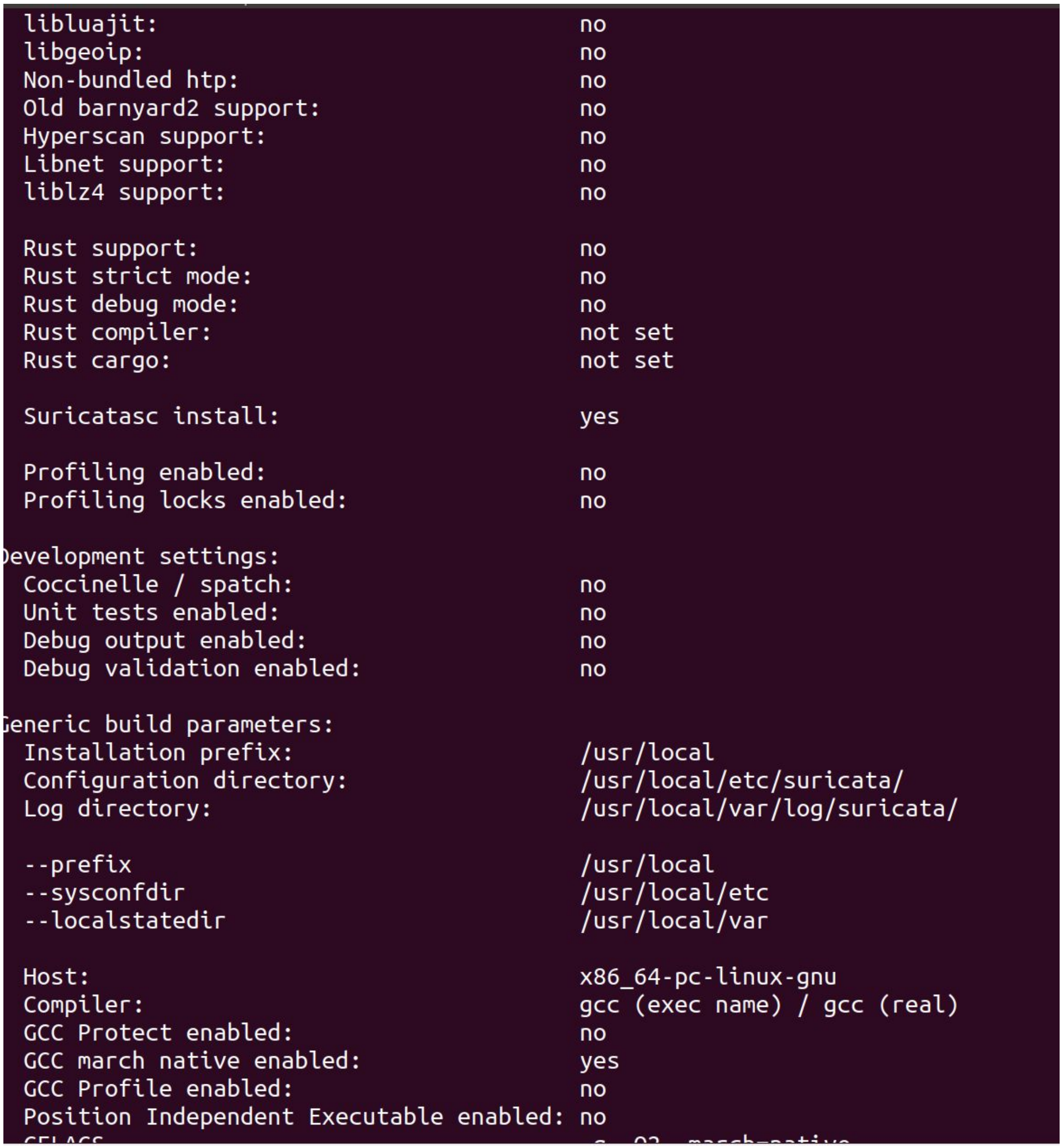
1.) **describe how the tool can be useful for detecting network attacks and how it can be customized to detect new attacks.**

Suricata is an open source network threat detection engine that provides capabilities including intrusion detection (IDS), intrusion prevention (IPS) and network security monitoring. It does extremely well with deep packet inspection and pattern matching which makes it incredibly useful for threat and attack detection. This tool is mainly used for packet analysis on a multi threaded environment so it can analyze packets much quicker, supports application layer protocols, supports hashing and file extraction, and creates complex detailed signature logic which can also be customizable in the rule and configuration files. This tool is highly customizable such as the Snort tool and therefore can easily be reconfigured or have custom rules added with newly emerging cyber threats.

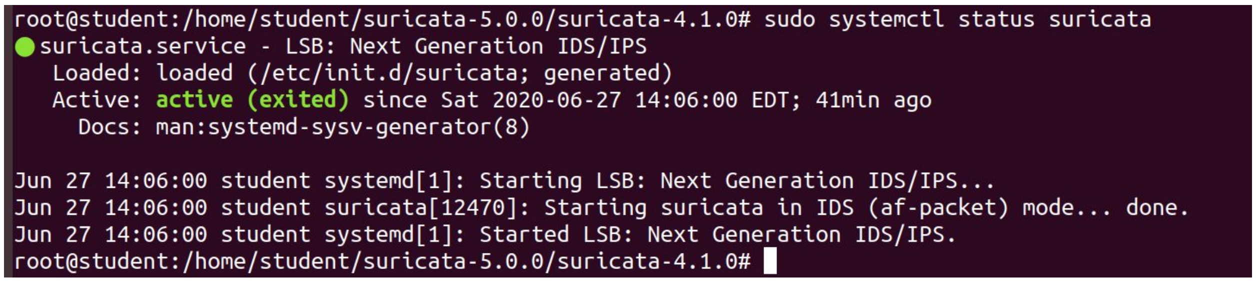
2. Installing Suricata?
Suricata can be installed from source code or using an Ubuntu repo. You are free to try whichever you would like, but the repo option is probably easiest. To install Suricata using a Debian package for Ubuntu, follow https://suricata.readthedocs.io/en/suricata-4.1.0/install.html#install-binary-packages . To do a source-based installation, use https://suricata.readthedocs.io/en/suricata-4.1.0/install.html instead. If you experience issues with either of these methods, additional installation instructions are provided by the Open InfoSec Foundation https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Installation . Provide screenshots of installing Suricata.



```
root@student:/home/student/suricata-5.0.0/suricata-4.1.0# sudo suricata --build-info
This is Suricata version 4.1.0 RELEASE
Features: PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT HAVE_HTP_URI_NORMALIZE_HOOK PCRE_JIT TLS
SIMD support: SSE_4_2 SSE_4_1 SSE_3
Atomic intrisics: 1 2 4 8 16 byte(s)
64-bits, Little-endian architecture
GCC version 7.5.0, C version 199901
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: __thread
compiled with LibHTP v0.5.28, linked against LibHTP v0.5.33

Suricata Configuration:
  AF_PACKET support:                        yes
  eBPF support:                             no
  XDP support:                              no
  PF_RING support:                          no
  NFQueue support:                          no
  NFLOG support:                            no
  IPFW support:                             no
  Netmap support:                           no
  DAG enabled:                              no
  Napatech enabled:                         no
  WinDivert enabled:                        no

  Unix socket enabled:                      no
  Detection enabled:                        yes

  Libmagic support:                         no
  libnss support:                           no
  libnspr support:                          no
  libjansson support:                       no
  liblzma support:                          yes
  hiredis support:                          no
  hiredis async with libevent:              no
  Prelude support:                          no
  PCRE jit:                                 yes
  LUA support:                              no
```



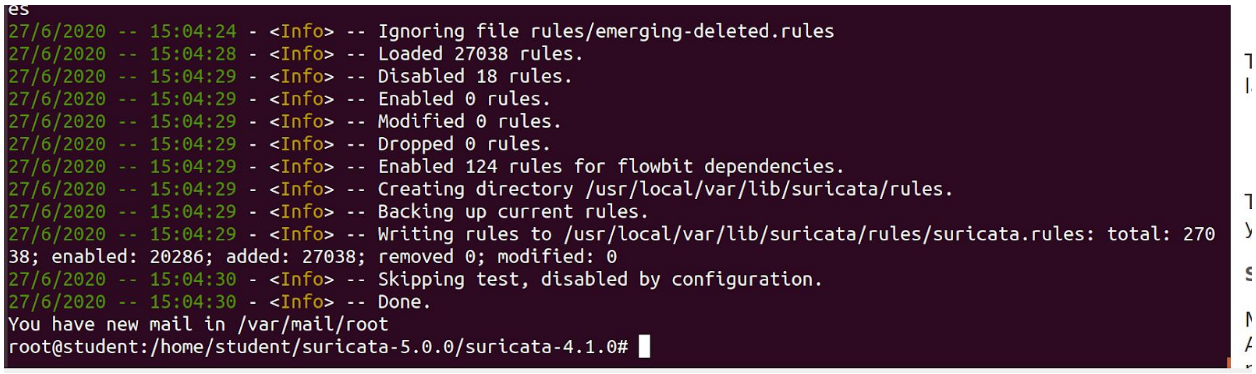
```
  libluajit:                               no
  libgeoip:                                no
  Non-bundled htp:                         no
  Old barnyard2 support:                   no
  Hyperscan support:                       no
  Libnet support:                          no
  liblz4 support:                          no

  Rust support:                            no
  Rust strict mode:                        no
  Rust debug mode:                         no
  Rust compiler:                           not set
  Rust cargo:                              not set

  Suricatasc install:                      yes

  Profiling enabled:                       no
  Profiling locks enabled:                 no

Development settings:
  Coccinelle / spatch:                     no
  Unit tests enabled:                      no
  Debug output enabled:                    no
  Debug validation enabled:                no

Generic build parameters:
  Installation prefix:                     /usr/local
  Configuration directory:                 /usr/local/etc/suricata/
  Log directory:                           /usr/local/var/log/suricata/

  --prefix                                 /usr/local
  --sysconfdir                             /usr/local/etc
  --localstatedir                          /usr/local/var

  Host:                                    x86_64-pc-linux-gnu
  Compiler:                                gcc (exec name) / gcc (real)
  GCC Protect enabled:                     no
  GCC march native enabled:                yes
  GCC Profile enabled:                     no
  Position Independent Executable enabled: no
```



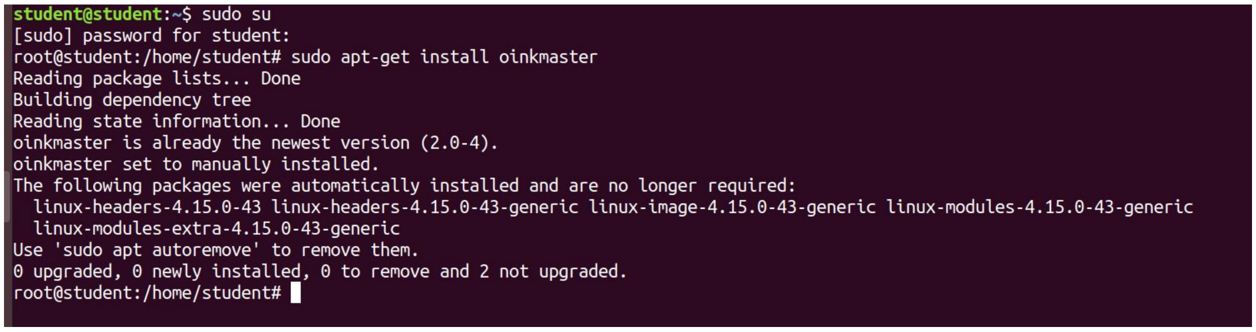
```
root@student:/home/student/suricata-5.0.0/suricata-4.1.0# sudo systemctl status suricata
● suricata.service - LSB: Next Generation IDS/IPS
   Loaded: loaded (/etc/init.d/suricata; generated)
   Active: active (exited) since Sat 2020-06-27 14:06:00 EDT; 41min ago
     Docs: man:systemd-sysv-generator(8)

Jun 27 14:06:00 student systemd[1]: Starting LSB: Next Generation IDS/IPS...
Jun 27 14:06:00 student suricata[12470]: Starting suricata in IDS (af-packet) mode... done.
Jun 27 14:06:00 student systemd[1]: Started LSB: Next Generation IDS/IPS.
root@student:/home/student/suricata-5.0.0/suricata-4.1.0#
```

3. Configuring Suricata

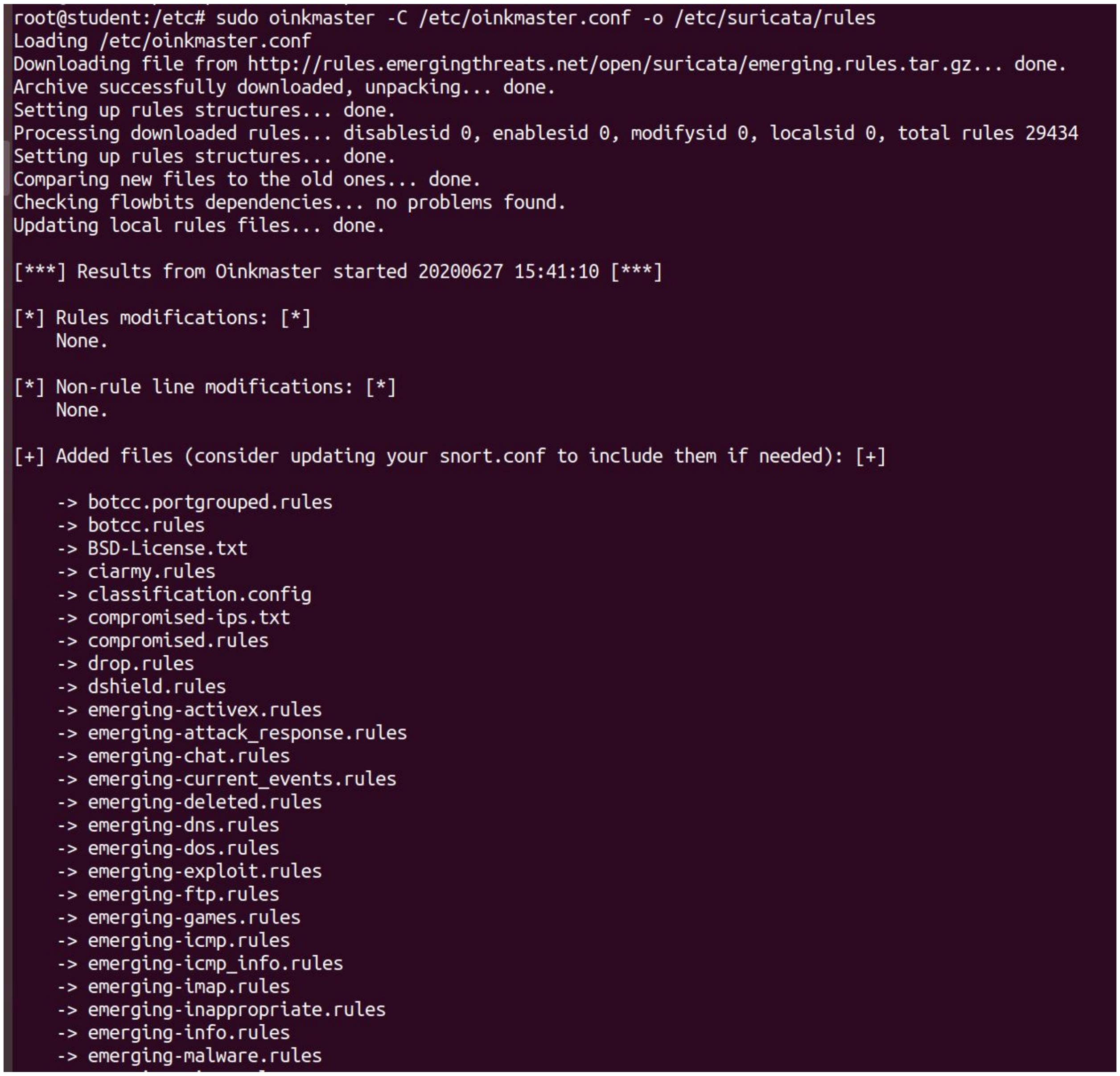
Once Suricata is installed, you will want to complete the basic configuration steps by following the https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Basic_Setup . Next, use Oinkmaster to configure Suricata to use a ruleset called, Rule Management with Oinkmaster, which can be found at https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Rule_Management_with_Oinkmaster. Additional documentation can be found at https://suricata.readthedocs.io/en/suricata-5.0.2/. Note, you will probably have to change the name of your network interface to en0sp1. Another note, you don't have to disable the unavailable rulesets unless you want to; it shouldn't hurt anything. Provide screenshots of configuring Suricata.



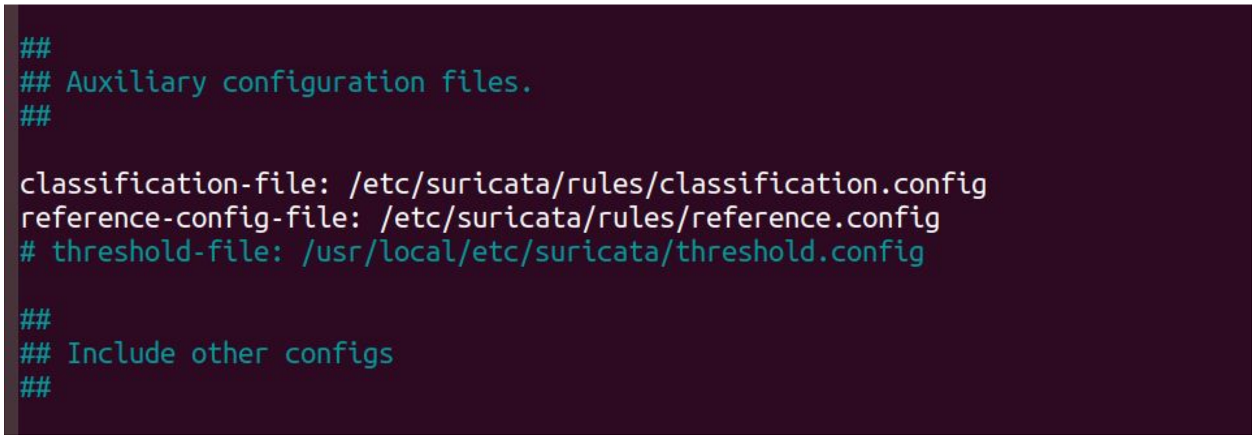
```
es
27/6/2020 -- 15:04:24 - <Info> -- Ignoring file rules/emerging-deleted.rules
27/6/2020 -- 15:04:28 - <Info> -- Loaded 27038 rules.
27/6/2020 -- 15:04:29 - <Info> -- Disabled 18 rules.
27/6/2020 -- 15:04:29 - <Info> -- Enabled 0 rules.
27/6/2020 -- 15:04:29 - <Info> -- Modified 0 rules.
27/6/2020 -- 15:04:29 - <Info> -- Dropped 0 rules.
27/6/2020 -- 15:04:29 - <Info> -- Enabled 124 rules for flowbit dependencies.
27/6/2020 -- 15:04:29 - <Info> -- Creating directory /usr/local/var/lib/suricata/rules.
27/6/2020 -- 15:04:29 - <Info> -- Backing up current rules.
27/6/2020 -- 15:04:29 - <Info> -- Writing rules to /usr/local/var/lib/suricata/rules/suricata.rules: total: 270
38; enabled: 20286; added: 27038; removed 0; modified: 0
27/6/2020 -- 15:04:30 - <Info> -- Skipping test, disabled by configuration.
27/6/2020 -- 15:04:30 - <Info> -- Done.
You have new mail in /var/mail/root
root@student:/home/student/suricata-5.0.0/suricata-4.1.0#
```



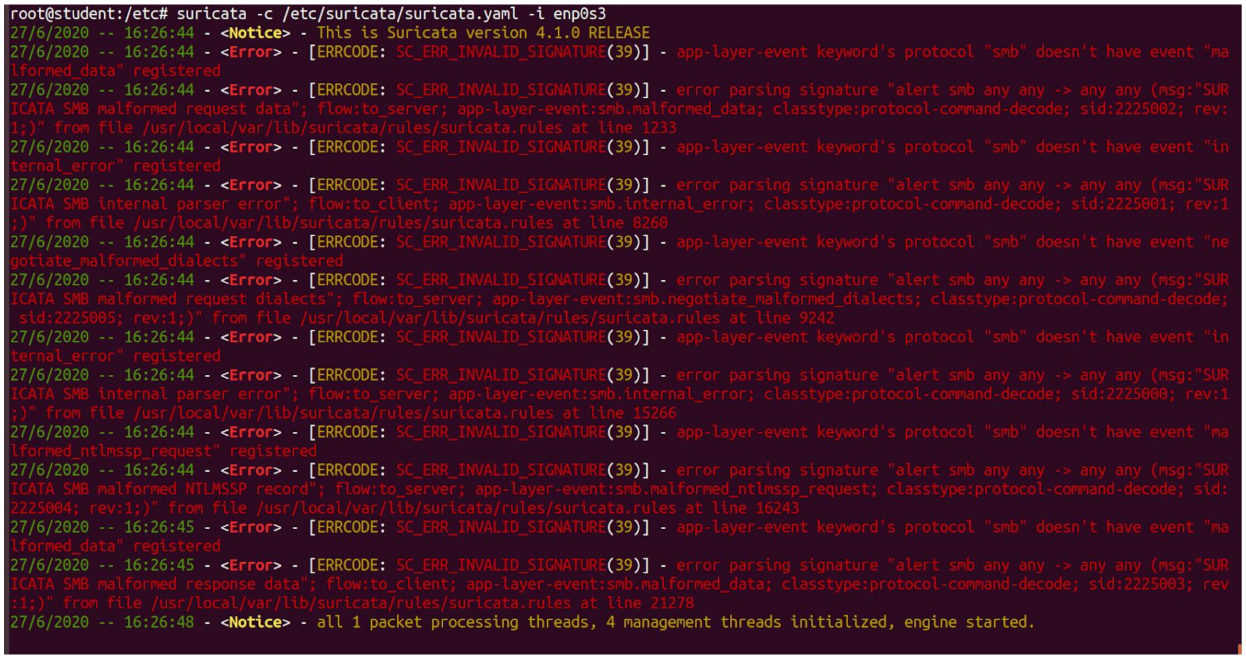
```
student@student:~$ sudo su
[sudo] password for student:
root@student:/home/student# sudo apt-get install oinkmaster
Reading package lists... Done
Building dependency tree
Reading state information... Done
oinkmaster is already the newest version (2.0-4).
oinkmaster set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-43 linux-headers-4.15.0-43-generic linux-image-4.15.0-43-generic linux-modules-4.15.0-43-generic
  linux-modules-extra-4.15.0-43-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@student:/home/student#
```



```
root@student:/etc# sudo oinkmaster -C /etc/oinkmaster.conf -o /etc/suricata/rules
Loading /etc/oinkmaster.conf
Downloading file from http://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz... done.
Archive successfully downloaded, unpacking... done.
Setting up rules structures... done.
Processing downloaded rules... disablesid 0, enablesid 0, modifysid 0, localsid 0, total rules 29434
Setting up rules structures... done.
Comparing new files to the old ones... done.
Checking flowbits dependencies... no problems found.
Updating local rules files... done.

[***] Results from Oinkmaster started 20200627 15:41:10 [***]

[*] Rules modifications: [*]
    None.

[*] Non-rule line modifications: [*]
    None.

[+] Added files (consider updating your snort.conf to include them if needed): [+]

    -> botcc.portgrouped.rules
    -> botcc.rules
    -> BSD-License.txt
    -> ciarmy.rules
    -> classification.config
    -> compromised-ips.txt
    -> compromised.rules
    -> drop.rules
    -> dshield.rules
    -> emerging-activex.rules
    -> emerging-attack_response.rules
    -> emerging-chat.rules
    -> emerging-current_events.rules
    -> emerging-deleted.rules
    -> emerging-dns.rules
    -> emerging-dos.rules
    -> emerging-exploit.rules
    -> emerging-ftp.rules
    -> emerging-games.rules
    -> emerging-icmp.rules
    -> emerging-icmp_info.rules
    -> emerging-imap.rules
    -> emerging-inappropriate.rules
    -> emerging-info.rules
    -> emerging-malware.rules
```



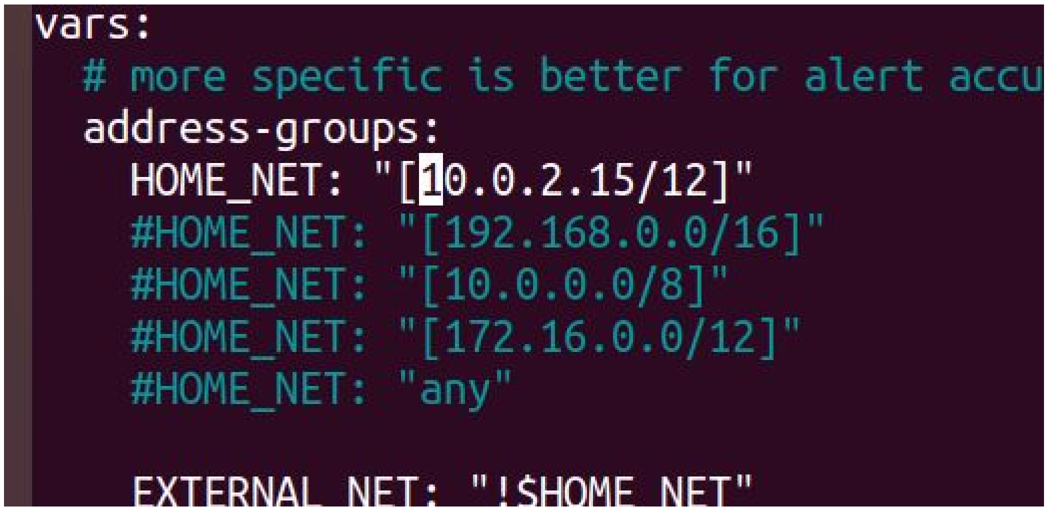
```
##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/rules/classification.config
reference-config-file: /etc/suricata/rules/reference.config
# threshold-file: /usr/local/etc/suricata/threshold.config

##
## Include other configs
##
```



```
root@student:/etc# suricata -c /etc/suricata/suricata.yaml -i enp0s3
27/6/2020 -- 16:26:44 - <Notice> - This is Suricata version 4.1.0 RELEASE
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - app-layer-event keyword's protocol "smb" doesn't have event "ma
lformed_data" registered
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - error parsing signature "alert smb any any -> any any (msg:"SUR
ICATA SMB malformed request data"; flow:to_server; app-layer-event:smb.malformed_data; classtype:protocol-command-decode; sid:2225002; rev:
1;)" from file /usr/local/var/lib/suricata/rules/suricata.rules at line 1233
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - app-layer-event keyword's protocol "smb" doesn't have event "in
ternal_error" registered
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - error parsing signature "alert smb any any -> any any (msg:"SUR
ICATA SMB internal parser error"; flow:to_client; app-layer-event:smb.internal_error; classtype:protocol-command-decode; sid:2225001; rev:1
;)" from file /usr/local/var/lib/suricata/rules/suricata.rules at line 8260
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - app-layer-event keyword's protocol "smb" doesn't have event "ne
gotiate_malformed_dialects" registered
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - error parsing signature "alert smb any any -> any any (msg:"SUR
ICATA SMB malformed request dialects"; flow:to_server; app-layer-event:smb.negotiate_malformed_dialects; classtype:protocol-command-decode;
 sid:2225005; rev:1;)" from file /usr/local/var/lib/suricata/rules/suricata.rules at line 9242
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - app-layer-event keyword's protocol "smb" doesn't have event "in
ternal_error" registered
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - error parsing signature "alert smb any any -> any any (msg:"SUR
ICATA SMB internal parser error"; flow:to_server; app-layer-event:smb.internal_error; classtype:protocol-command-decode; sid:2225000; rev:1
;)" from file /usr/local/var/lib/suricata/rules/suricata.rules at line 15266
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - app-layer-event keyword's protocol "smb" doesn't have event "ma
lformed_ntlmssp_request" registered
27/6/2020 -- 16:26:44 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - error parsing signature "alert smb any any -> any any (msg:"SUR
ICATA SMB malformed NTLMSSP record"; flow:to_server; app-layer-event:smb.malformed_ntlmssp_request; classtype:protocol-command-decode; sid:
2225004; rev:1;)" from file /usr/local/var/lib/suricata/rules/suricata.rules at line 16243
27/6/2020 -- 16:26:45 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - app-layer-event keyword's protocol "smb" doesn't have event "ma
lformed_data" registered
27/6/2020 -- 16:26:45 - <Error> - [ERRCODE: SC_ERR_INVALID_SIGNATURE(39)] - error parsing signature "alert smb any any -> any any (msg:"SUR
ICATA SMB malformed response data"; flow:to_client; app-layer-event:smb.malformed_data; classtype:protocol-command-decode; sid:2225003; rev
:1;)" from file /usr/local/var/lib/suricata/rules/suricata.rules at line 21278
27/6/2020 -- 16:26:48 - <Notice> - all 1 packet processing threads, 4 management threads initialized, engine started.
```

4. Creating Custom Suricata Rules

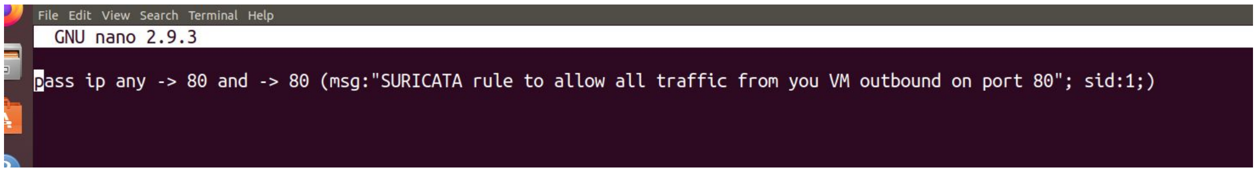
Start by editing your Suricata config.yaml file to point the $HOME_NET variable to the IP of your VM. For most of you, the IP address should be 10.0.2.15, but you can use the ipconfig command to find your VM if you are not sure. You may also find it helpful to define some other variables, but that choice is up to you.



```
vars:
  # more specific is better for alert accu
  address-groups:
    HOME_NET: "[10.0.2.15/12]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
```

Next, please create the following custom rules:

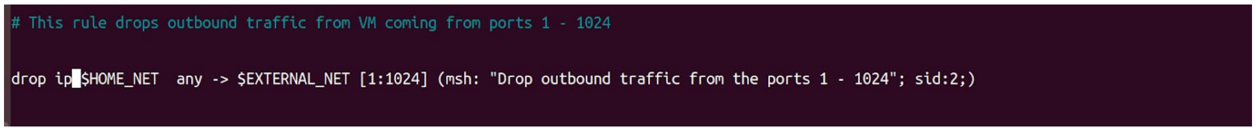
1. Allow all traffic from your VM outbound on port 80



```
GNU nano 2.9.3

pass ip any -> 80 and -> 80 (msg:"SURICATA rule to allow all traffic from you VM outbound on port 80"; sid:1;)
```

This rule above has the pass flag which indicates to suricata to pass if matching any ip, and in our case, the ports are the only things specified. Where the sid is the signature of the rule.

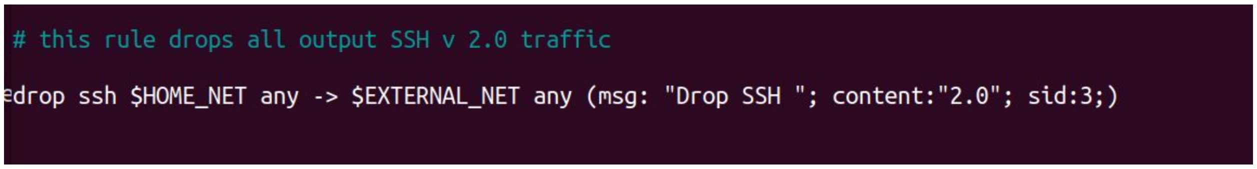
2. Drop any traffic outbound from your VM coming from ports 1-1024



```
# This rule drops outbound traffic from VM coming from ports 1 - 1024

drop ip $HOME_NET  any -> $EXTERNAL_NET [1:1024] (msh: "Drop outbound traffic from the ports 1 - 1024"; sid:2;)
```

The rule above is different from the rule above for the following reasons: first it states to drop if any ip and port match the following array of ports ranging from 1 - 1024 defined in the array syntax.

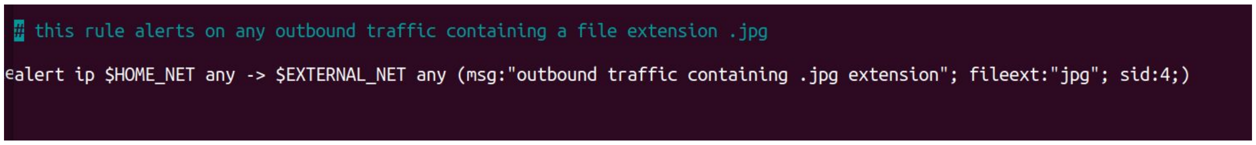
3. Drop all outbound SSH v2.0 traffic coming from your VM



```
# this rule drops all output SSH v 2.0 traffic

drop ssh $HOME_NET any -> $EXTERNAL_NET any (msg: "Drop SSH "; content:"2.0"; sid:3;)
```

This rule says to drop any ssh connection with the content of version 2.0

4. Alert on any outbound traffic containing a file with a .jpg extension that originates from your VM.



```
# this rule alerts on any outbound traffic containing a file extension .jpg

alert ip $HOME_NET any -> $EXTERNAL_NET any (msg:"outbound traffic containing .jpg extension"; fileext:"jpg"; sid:4;)
```

This rule says to alert with the following message from any io external net with any port. The magic for this rule comes with the fileext flag which tells suricata to look for the jpg file extension.

5. Create and explain 3 custom rules of your own choosing. Please be sure to include at least 3 rule options (of any type) in each of your 3 rules. The https://suricata.readthedocs.io/en/suricata-4.1.0/rules/intro.html# should be very helpful.

1.) This is a tcp alert that looks at any inbound traffic on any port for an exploit. The content here will look for the signature in between the quotes to match within the payload. The depth says that it looks 200 bytes from the beginning of the payload. The content here will look for the signature in between the quotes to match within the payload. The distance parameter here is a relative content modifier and basically says that there is no distance between the content keyword and the content preceding it. Classtype is a keyword that gives info about the kind of rule. Sid is a signature, and rev is a revision number.

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "CobaltStrike login server"; flow:established; content:"Cyberspace"; depth:200; content:"Somewhere"; distance:0; content:"cobaltstrike"; distance:0; content:"AdvancedPenTesting";distance:0; classtype:exploit-kit; sid:3016001; rev:1;)

2.) This is a tcp alert that looks at outbound traffic on port 3305 for possible trojan activity. The depth says that it looks 5 bytes from the beginning of the payload. The content here will look for the signature in between the quotes to match within the payload. The distance parameter here is a relative content modifier and basically says that there is no distance between the content keyword and the content preceding it. This is very similar to the previous rule, but looks at it the other direction essentially.

alert tcp $HOME_NET any -> any 3306 (msg: "mysql general_log write file"; flow: established; content:"|03|"; depth: 5; content:"|67 65 6e 65 72 61 6c 5f 6c 6f 67 5f 66 69 6c 65|"; distance:0; classtype:trojan-activity; sid: 3013005; rev: 1;)

3.) The depth is again how many bytes from the beginning of the payload will be checked. PCRE is a Perl Compatible Regular Expression, basically it's a search function that will search the payload from those specific things. The dsize allows for matching of the size of the packet payload. 53 means we are looking at port 53. This whole rule basically is checking for trojan horse activity over udp.

alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"Suspicious dns request"; flow:established,to_server; content:"|01 00|"; depth:4;

```
pcre:"/\x00\x10\x00\x01|\x00\x0f\x00\x01|\x00\x05\x00\x01/"; dsize:>200;
classtype:trojan-activity; sid:3011001; rev:1;)
```

5. Deliverables
Please submit a PDF or word document containing the 7 rules along with an explanation of why you structured each of the rules the way you did as this will help me to be able to award partial credit.

Note: For the first four rules, I asked you to implement all outbound rules so that you have the ability to test them without spinning up another VM should you choose too.

Testing is optional, but the option exists if you'd like to try your rules in action.

References
https://bricata.com/blog/what-is-suricata-ids/

https://blog.inliniac.net/2011/11/29/file-extraction-in-suricata/