EXPLORE SIGNATURES AND LOGS WITH SURICATA

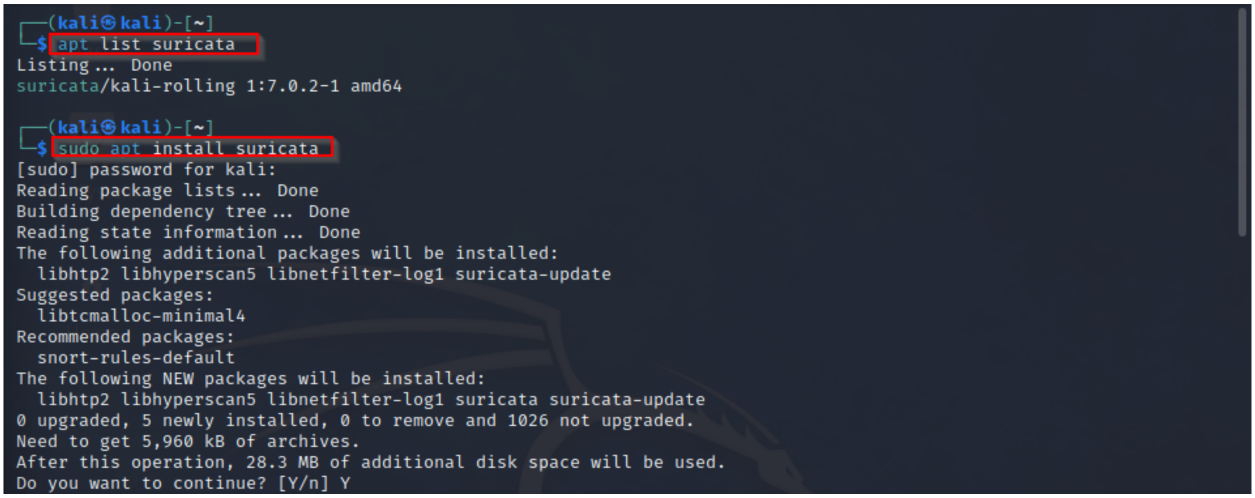
Pedro Osorio Lopez.

1. High-Level Project Description:

*In this lab activity, we'll explore the components of a rule using Suricata. We'll also have an opportunity to trigger a rule and examine the output in Suricata. We'll use the Bash shell to complete these steps.

2. Project Setup and Required Tools:

*We need a Linux Machine and access to the command line interface (CLI). Suricata was already pre-installed in the machine given by Google, but note that the installation and configuration of the tool might be required if using your own device to complete the activity. A screenshot showing how to install Suricata is attached, but a full guide on how to configure the tool is out of scope for this assignment.

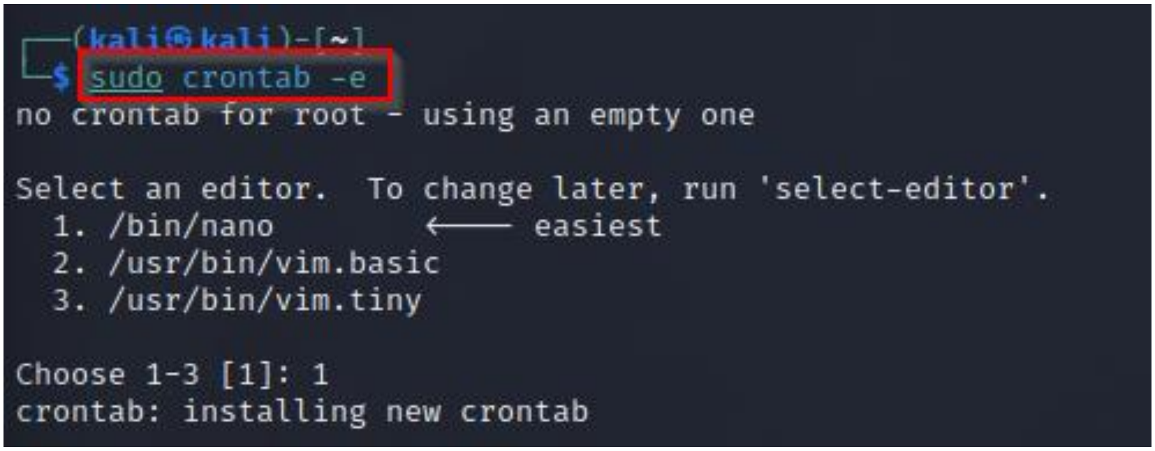


```
┌──(kali㉿kali)-[~]
└─$ apt list suricata
Listing... Done
suricata/kali-rolling 1:7.0.2-1 amd64

┌──(kali㉿kali)-[~]
└─$ sudo apt install suricata
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libhtp2 libhyperscan5 libnetfilter-log1 suricata-update
Suggested packages:
  libtcmalloc-minimal4
Recommended packages:
  snort-rules-default
The following NEW packages will be installed:
  libhtp2 libhyperscan5 libnetfilter-log1 suricata suricata-update
0 upgraded, 5 newly installed, 0 to remove and 1026 not upgraded.
Need to get 5,960 kB of archives.
After this operation, 28.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

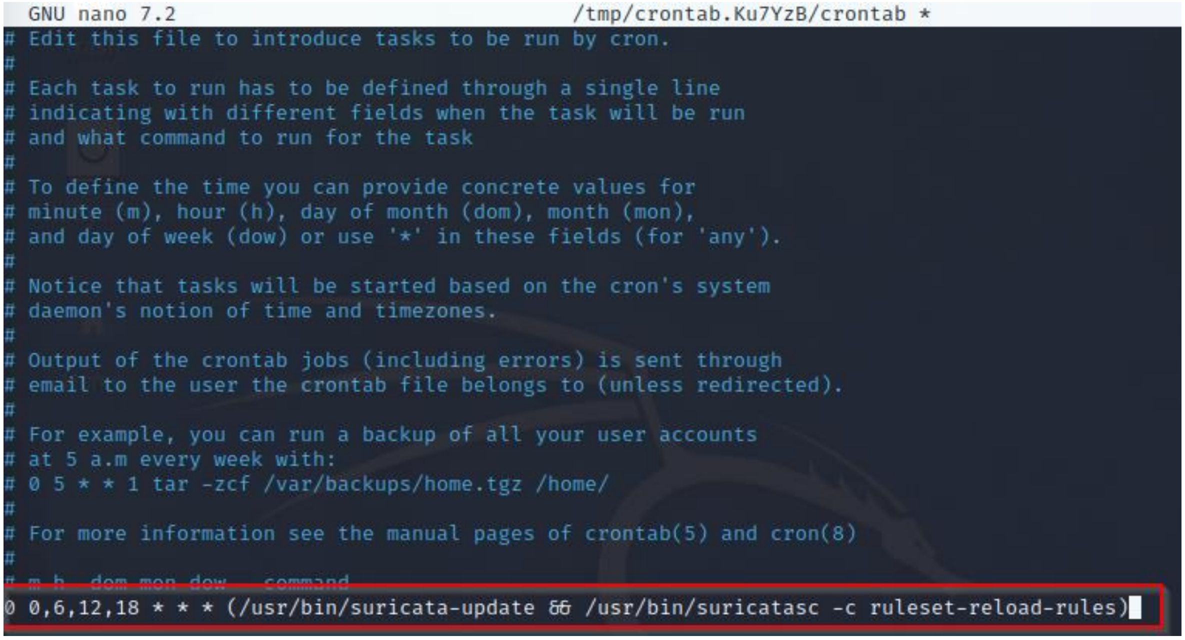
3. Step by Step Project Walkthrough

*In this lab, we will: Examine a rule in Suricata, trigger a rule, review the alert logs, and examine **eve.json** outputs.

*To start, we used crontabs to enable automatic rule updates every 6 hours, which we will use for testing the Intrusion Detection System.

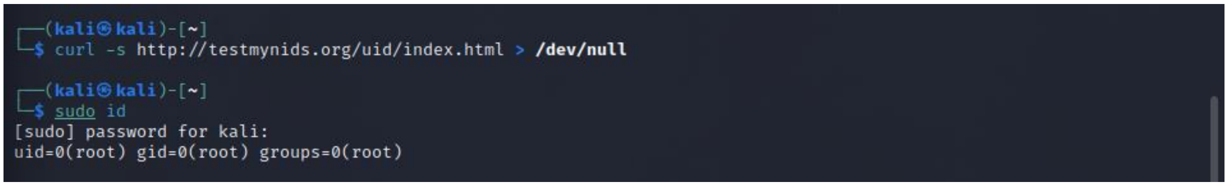


```
┌──(kali㉿kali)-[~]
└─$ sudo crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        ⟵—— easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
crontab: installing new crontab
```



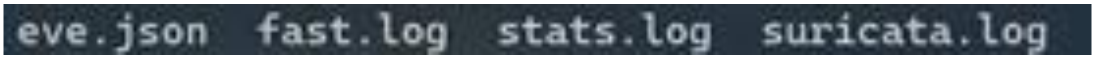
```
  GNU nano 7.2                               /tmp/crontab.Ku7YzB/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 0,6,12,18 * * * (/usr/bin/suricata-update && /usr/bin/suricatasc -c ruleset-reload-rules)
```

*Now that we have the IDS prepared, we will proceed to test it by triggering the rule. We will start by doing an activity that Suricata will consider “suspicious,” such as using the getting a file from testminids and check the sudo id.



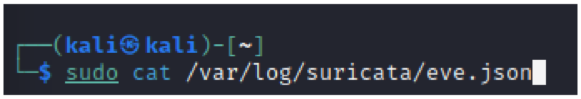
```
┌──(kali㉿kali)-[~]
└─$ curl -s http://testmynids.org/uid/index.html > /dev/null

┌──(kali㉿kali)-[~]
└─$ sudo id
[sudo] password for kali:
uid=0(root) gid=0(root) groups=0(root)
```

*Now, if we type: sudo ls /var/log/Suricata, the following logs should appear.

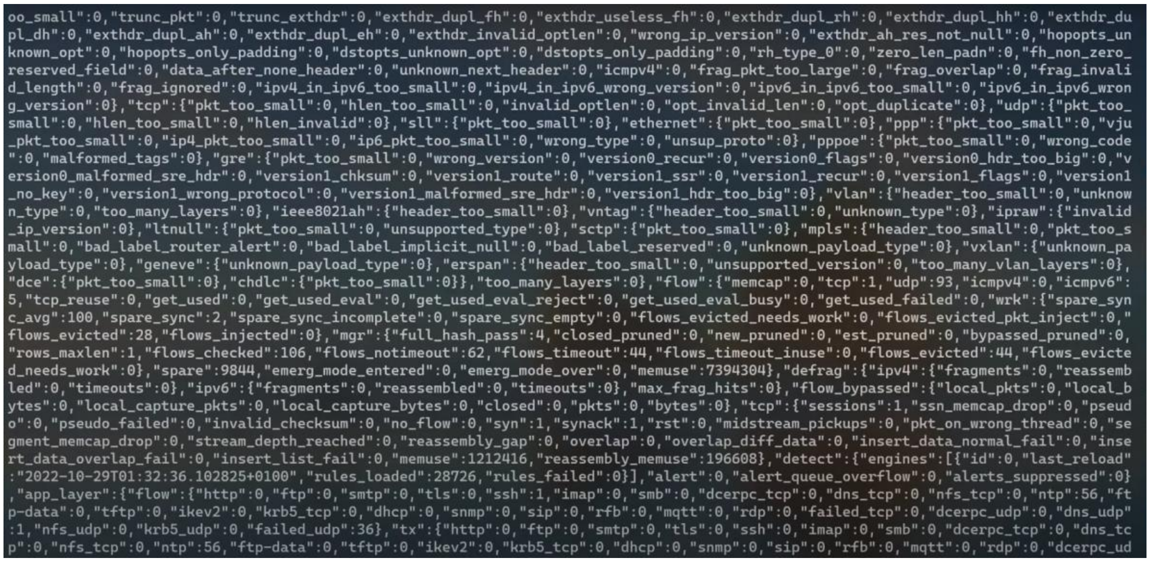


```
eve.json  fast.log  stats.log  suricata.log
```

*Then, we will proceed to take a look at the content inside the eve.json file. Note that in order to analyze its information properly, we will need to apply some filters or use other specialized tools to have a better look.

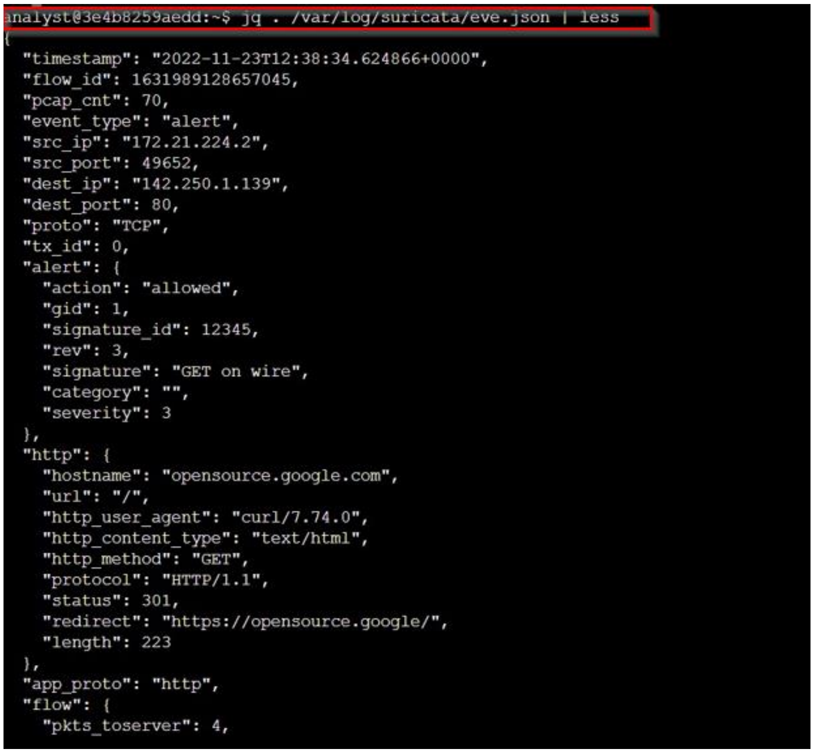


```
┌──(kali㉿kali)-[~]
└─$ sudo cat /var/log/suricata/eve.json
```



```
oo_small":0,"trunc_pkt":0,"trunc_exthdr":0,"exthdr_dupl_fh":0,"exthdr_useless_fh":0,"exthdr_dupl_rh":0,"exthdr_dupl_hh":0,"exthdr_du
pl_dh":0,"exthdr_dupl_ah":0,"exthdr_dupl_eh":0,"exthdr_invalid_optlen":0,"wrong_ip_version":0,"exthdr_ah_res_not_null":0,"hopopts_un
known_opt":0,"hopopts_only_padding":0,"dstopts_unknown_opt":0,"dstopts_only_padding":0,"rh_type_0":0,"zero_len_padn":0,"fh_non_zero_
reserved_field":0,"data_after_none_header":0,"unknown_next_header":0,"icmpv4":0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_invali
d_length":0,"frag_ignored":0,"ipv4_in_ipv6_too_small":0,"ipv4_in_ipv6_wrong_version":0,"ipv6_in_ipv6_too_small":0,"ipv6_in_ipv6_wron
g_version":0},"tcp":{"pkt_too_small":0,"hlen_too_small":0,"invalid_optlen":0,"opt_invalid_len":0,"opt_duplicate":0},"udp":{"pkt_too_
small":0,"hlen_too_small":0,"hlen_invalid":0},"sll":{"pkt_too_small":0},"ethernet":{"pkt_too_small":0},"ppp":{"pkt_too_small":0,"vju
_pkt_too_small":0,"ip4_pkt_too_small":0,"ip6_pkt_too_small":0,"wrong_type":0,"unsup_proto":0},"pppoe":{"pkt_too_small":0,"wrong_code
":0,"malformed_tags":0},"gre":{"pkt_too_small":0,"wrong_version":0,"version0_recur":0,"version0_flags":0,"version0_hdr_too_big":0,"v
ersion0_malformed_sre_hdr":0,"version1_chksum":0,"version1_route":0,"version1_ssr":0,"version1_recur":0,"version1_flags":0,"version1
_no_key":0,"version1_wrong_protocol":0,"version1_malformed_sre_hdr":0,"version1_hdr_too_big":0},"vlan":{"header_too_small":0,"unknow
n_type":0,"too_many_layers":0},"ieee8021ah":{"header_too_small":0},"vntag":{"header_too_small":0,"unknown_type":0},"ipraw":{"invalid
_ip_version":0},"ltnull":{"pkt_too_small":0,"unsupported_type":0},"sctp":{"pkt_too_small":0},"mpls":{"header_too_small":0,"pkt_too_s
mall":0,"bad_label_router_alert":0,"bad_label_implicit_null":0,"bad_label_reserved":0,"unknown_payload_type":0},"vxlan":{"unknown_pa
yload_type":0},"geneve":{"unknown_payload_type":0},"erspan":{"header_too_small":0,"unsupported_version":0,"too_many_vlan_layers":0},
"dce":{"pkt_too_small":0},"chdlc":{"pkt_too_small":0}},"too_many_layers":0},"flow":{"memcap":0,"tcp":1,"udp":93,"icmpv4":0,"icmpv6":
5,"tcp_reuse":0,"get_used":0,"get_used_eval":0,"get_used_eval_reject":0,"get_used_eval_busy":0,"get_used_failed":0,"wrk":{"spare_syn
c_avg":100,"spare_sync":2,"spare_sync_incomplete":0,"spare_sync_empty":0,"flows_evicted_needs_work":0,"flows_evicted_pkt_inject":0,"
flows_evicted":28,"flows_injected":0},"mgr":{"full_hash_pass":4,"closed_pruned":0,"new_pruned":0,"est_pruned":0,"bypassed_pruned":0,
"rows_maxlen":1,"flows_checked":106,"flows_notimeout":62,"flows_timeout":44,"flows_timeout_inuse":0,"flows_evicted":44,"flows_evicte
d_needs_work":0},"spare":9844,"emerg_mode_entered":0,"emerg_mode_over":0,"memuse":7394304},"defrag":{"ipv4":{"fragments":0,"reassemb
led":0,"timeouts":0},"ipv6":{"fragments":0,"reassembled":0,"timeouts":0},"max_frag_hits":0},"flow_bypassed":{"local_pkts":0,"local_b
ytes":0,"local_capture_pkts":0,"local_capture_bytes":0,"closed":0,"pkts":0,"bytes":0},"tcp":{"sessions":1,"ssn_memcap_drop":0,"pseud
o":0,"pseudo_failed":0,"invalid_checksum":0,"no_flow":0,"syn":1,"synack":1,"rst":0,"midstream_pickups":0,"pkt_on_wrong_thread":0,"se
gment_memcap_drop":0,"stream_depth_reached":0,"reassembly_gap":0,"overlap":0,"overlap_diff_data":0,"insert_data_normal_fail":0,"inse
rt_data_overlap_fail":0,"insert_list_fail":0,"memuse":1212416,"reassembly_memuse":196608},"detect":{"engines":[{"id":0,"last_reload"
:"2022-10-29T01:32:36.102825+0100","rules_loaded":28726,"rules_failed":0}],"alert":0,"alert_queue_overflow":0,"alerts_suppressed":0}
,"app_layer":{"flow":{"http":0,"ftp":0,"smtp":0,"tls":0,"ssh":1,"imap":0,"smb":0,"dcerpc_tcp":0,"dns_tcp":0,"nfs_tcp":0,"ntp":56,"ft
p-data":0,"tftp":0,"ikev2":0,"krb5_tcp":0,"dhcp":0,"snmp":0,"sip":0,"rfb":0,"mqtt":0,"rdp":0,"failed_tcp":0,"dcerpc_udp":0,"dns_udp"
:1,"nfs_udp":0,"krb5_udp":0,"failed_udp":36},"tx":{"http":0,"ftp":0,"smtp":0,"tls":0,"ssh":0,"imap":0,"smb":0,"dcerpc_tcp":0,"dns_tc
p":0,"nfs_tcp":0,"ntp":56,"ftp-data":0,"tftp":0,"ikev2":0,"krb5_tcp":0,"dhcp":0,"snmp":0,"sip":0,"rfb":0,"mqtt":0,"rdp":0,"dcerpc_ud
```

*Lastly, we will proceed to display the output of the eve.json file from the Google Cybersecurity lab, we used the command jq to organize the output in a readable manner:



```
analyst@3e4b8259aedd:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 1631989128657045,
  "pcap_cnt": 70,
  "event_type": "alert",
  "src_ip": "172.21.224.2",
  "src_port": 49652,
  "dest_ip": "142.250.1.139",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 12345,
    "rev": 3,
    "signature": "GET on wire",
    "category": "",
    "severity": 3
  },
  "http": {
    "hostname": "opensource.google.com",
    "url": "/",
    "http_user_agent": "curl/7.74.0",
    "http_content_type": "text/html",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "status": 301,
    "redirect": "https://opensource.google/",
    "length": 223
  },
  "app_proto": "http",
  "flow": {
    "pkts_toserver": 4,
```

4. Summary and/or Recommendations:

*In this activity, we learned how to use Suricata, a powerful intrusion detection system which can be used to analyze and trigger rules to examine different types of logs.