EXAMINING ALERTS, LOGS, AND RULES WITH SURICATA

A LAB BY DARNELL MAIDEN

```
analyst@824bcdcaee9e:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; fl
ow:established,to_server; content:"GET"; http_method; sid:12345; rev
:3;)
analyst@824bcdcaee9e:~$ ls -l /var/log/suricata
total 0
analyst@824bcdcaee9e:~$ sudo suricata -r sample.pcap -S custom.rules
 -k none
20/2/2026 -- 17:45:32 - <Notice> - This is Suricata version 6.0.1 RE
LEASE running in USER mode
20/2/2026 -- 17:45:33 - <Notice> - all 2 packet processing threads,
4 management threads initialized, engine started.
20/2/2026 -- 17:45:33 - <Notice> - Signal Received.  Stopping engine
.
20/2/2026 -- 17:45:33 - <Notice> - Pcap-file module read 1 files, 20
0 packets, 54238 bytes
analyst@824bcdcaee9e:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1419 Feb 20 17:45 eve.json
-rw-r--r-- 1 root root  292 Feb 20 17:45 fast.log
-rw-r--r-- 1 root root 2846 Feb 20 17:45 stats.log
-rw-r--r-- 1 root root 1512 Feb 20 17:45 suricata.log
analyst@824bcdcaee9e:~$ cat /var/log/suricata/fast.log
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**] [Class
ification: (null)] [Priority: 3] {TCP} 172.21.224.2:49652 -> 142.250
.1.139:80
11/23/2022-12:38:58.958203  [**] [1:12345:3] GET on wire [**] [Class
ification: (null)] [Priority: 3] {TCP} 172.21.224.2:58494 -> 142.250
.1.102:80
analyst@824bcdcaee9e:~$
```

SUMMARY:

AS I CLOCK IN AND SIT AT MY DESK, THE LEAD PUTS THEIR HAND ON MY SHOULDER AND SAYS, “YOU KNOW YOU’RE MY FAVORITE EMPLOYEE RIGHT?”. AHHHH YES. THE MANAGERIAL GUILT TRIP. I LOVE IT. “WHAT DO YOU WANT?” I SAY WITH FAKE ENTHUSIASM AND GRIEF. THEY TELL ME THEY NEED TO EXAMINE SOME LOGS, ALERTS, AND RULES WITH SURICATA. SURICATA SOUNDS PRETTY COOL, LIKE A MOTORCYCLE THAT ATTRACTS DIVORCED PTA MOMS. NEVERTHELESS, I HAVE TO GET TO WORK ON THIS.

- FIRST, I HAVE TO EXPLORE CUSTOM RULES IN SURICATA.
- SECOND, I NEED TO RUN SURICATA WITH A CUSTOM RULE TO TRIGGER IT AND THEN EXAMINE THE OUTPUT LOGS IN THE [FAST.LOG] FILE.
- FINALLY, I HAVE TO EXAMINE THE ADDITIONAL OUTPUT THAT SURICATA GENERATES IN THE STANDARD [EVE.JSON] LOG FILE.

I WAS GIVEN A [SAMPLE.PCAP] FILE AND A [CUSTOM.RULES] FILE THAT RESIDE IN THE HOME FOLDER.

WAIT.....

I SHOULD EXPLAIN THESE THINGS SHOULDN'T I?

MY FAULT, READER.

WHAT I'M WORKING WITH:

THE [SAMPLE.PCAP] FILE IS A PACKET CAPTURE FILE THAT CONTAINS AN EXAMPLE OF NETWORK TRAFFIC DATA, WHICH I'LL BE USING TO TRIGGER THE SURICATA RULES. THIS WILL ALLOW ME TO SIMULATE AND REPEAT THE MONITORING NETWORK TRAFFIC.

THE [CUSTOM.RULES] FILE CONTAINS A CUSTOM RULE. I WILL ADD RULES TO THIS FILE AND RUN THEM AGAINST THE NETWORK TRAFFIC DATA IN THE SAMPLE.PCAP FILE.

THE [FAST.LOG] FILE WILL CONTAIN THE ALERTS THAT SURICATA GENERATES.

THE [EVE.JSON] FILE IS THE MAIN, STANDARD, AND DEFAULT LOG FOR EVENTS GENERATED BY SURICATA. IT CONTAINS DETAILED INFORMATION ABOUT ALERTS TRIGGERED, AS WELL AS OTHER NETWORK TELEMETRY EVENTS, IN JSON FORMAT.

RULES ARE RULES: CUSTOM RULES IN SURICATA:

LET'S HEAD TO THE /HOME/ANALYST DIRECTORY AND USE OUR CUSTOM.RULES FILE THAT DEFINES THE NETWORK TRAFFIC RULES, WHICH SURICATA CAPTURES.

LET'S USE THE CAT COMMAND TO DISPLAY THE RULE IN THE CUSTOM.RULES FILE.
[CAT CUSTOM.RULES]

```
analyst@aa3669282539:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; fl
ow:established,to_server; content:"GET"; http_method; sid:12345; rev
:3;)
```

NOW THAT I HAVE MY [OUTPUT] WHILE EATING CHICKEN WINGS AT MY DESK, I WANT TO BREAK DOWN THE COMPONENTS OF IT.

```
analyst@aa3669282539:~
alert http $HOME_NET a
```

THE [ALERT] SECTION IN THE [OUTPUT] IS THE [ACTION], WHICH DETERMINES THE ACTION TO TAKE IF ALL CONDITIONS ARE MET. THERE ARE DIFFERENT ACTIONS ACROSS ALL NETWORK INTRUSION DETECTION SYSTEM (NIDS) RULE LANGUAGES.

THE MOST COMMON [ACTIONS] ARE:

[ALERT]: INSTRUCTS TO ALERT ON SELECTED NETWORK TRAFFIC.

[DROP]: ALSO GENERATES AN ALERT, BUT DROPS THE TRAFFIC. IT ONLY OCCURS WHEN SURICATA RUNS IN IPS MODE.

[PASS]: ALLOWS TRAFFIC TO PASS THROUGH THE INTERFACE. THIS RULE CAN OVERRIDE OTHER RULES. THE ONE EXCEPTION TO THE [DROP] RULE CAN BE MADE WITH A [PASS] RULE.

[REJECT]: DOES NOT ALLOW TRAFFIC TO PASS WHATSOEVER. IT'LL SEND A TCP RESET PACKET AND SURICATA WILL DROP THE MATCHING PACKET. A TCP PACKET TELLS THE COMPUTERS TO STOP SENDING MESSAGES TO EACH OTHER.

THIS PART OF THE SIGNATURE IS THE [HEADER].

```
http $HOME_NET any -> $EXTERNAL_NET any
```

THIS IS WHAT DEFINES THE SIGNATURE'S NETWORK TRAFFIC, WHICH INCLUDES IMPORTANT INFORMATION SUCH AS:

- PROTOCOL: "HTTP" IS THE PROTOCOL.
- SOURCE/DESTINATION IP ADDRESSES
- SOURCE/DESTINATION PORTS
- TRAFFIC DIRECTION

THE PARAMETERS FOR THE "HTTP" FIELD ARE:

[$HOME_NET ANY -> $EXTERNAL_NET ANY]

THE ARROW INDICATES THE DIRECTION OF THE TRAFFIC COMING FROM THE $HOME_NET AND GOING TO THE DESTINATION IP $EXTERNAL_NET

$HOME_NET

IS A SURICATA VARIABLE DEFINED IN /ETC/SURICATA/SURICATA.YAML THAT YOU CAN USE IN YOUR RULE DEFINITIONS AS A PLACEHOLDER FOR YOUR LOCAL OR HOME NETWORK TO IDENTIFY TRAFFIC THAT CONNECTS TO OR FROM YOUR SYSTEMS WITHIN THE ORGANIZATION. THIS IS ALSO DEFINED AS THE 172.21.224.0/20 SUBNET. THE WORD [ANY] MEANS THAT SURICATA CATCHES TRAFFIC FROM ANY PORT DEFINED IN THE $HOME_NET NETWORK.

THIS SIGNATURE BASICALLY TRIGGERS AN ALERT WHEN IT DETECTS ANY HTTP TRAFFIC LEAVING THE HOME NETWORK AND GOING TO THE EXTERNAL NETWORK. NOW THAT'S GOING TO BE USEFUL IN MY ANALYST JOURNEY. I LOVE IT.

NOW LET'S EXPLAIN THE [RULE].

```
analyst@aa3669282539:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; fl
ow:established,to_server; content:"GET"; http_method; sid:12345; rev
:3;)
```

THE RULE OPTION HELPS YOU NARROW DOWN THE NETWORK TRAFFIC TO GET WHAT YOU'RE LOOKING FOR FAST. IT'S USUALLY IN BETWEEN PARENTHESES AND SEPARATED BY SEMICOLONS. LET'S BREAK IT DOWN FURTHER.

- THE `MSG:` OPTION PROVIDES THE ALERT TEXT. IN THIS CASE, THE ALERT WILL PRINT OUT THE TEXT `"GET ON WIRE"`, WHICH SPECIFIES WHY THE ALERT WAS TRIGGERED.
- THE `FLOW: ESTABLISHED,TO_SERVER` OPTION DETERMINES THAT PACKETS FROM THE CLIENT TO THE SERVER SHOULD BE MATCHED. (IN THIS INSTANCE, A SERVER IS DEFINED AS THE DEVICE RESPONDING TO THE INITIAL SYN PACKET WITH A SYN-ACK PACKET.)
- THE `CONTENT: "GET"` OPTION TELLS SURICATA TO LOOK FOR THE WORD GET IN THE CONTENT OF THE `HTTP.METHOD` PORTION OF THE PACKET.
- THE `SID: 12345 (SIGNATURE ID)` OPTION IS A UNIQUE NUMERICAL VALUE THAT IDENTIFIES THE RULE.
- THE `REV:3` OPTION INDICATES THE SIGNATURE'S REVISION WHICH IS USED TO IDENTIFY THE SIGNATURE'S VERSION. HERE, THE REVISION VERSION IS 3.

TO MAKE THIS PERFECTLY CLEAR...

THIS SIGNATURE TRIGGERS AN ALERT WHENEVER SURICATA OBSERVES THE TEXT GET AS THE HTTP METHOD IN AN HTTP PACKET FROM THE HOME NETWORK GOING TO THE EXTERNAL NETWORK.

PULL THE TRIGGER: TRIGGERING THE RULE.:

NOW THAT WE HAVE A PERFECT UNDERSTANDING OF THE DYNAMICS, I WILL NOW LIST THE FILES IN THE /VAR/LOG/SURICATA FOLDER.

[ls -l /var/log/suricata]

(REMEMBER, LS IS USED TO DISPLAY THE CONTENTS OF THE DIRECTORY AND -L DISPLAYS THE DETAILED INFO)

```
analyst@aa3669282539:~$ ls -l /var/log/suricata
total 0
```

WELL WOULD YOU LOOK AT THAT? THERE'S NOTHING HERE! BUMMER.

LET'S RUN SURICATA USING THE CUSTOM.RULES AND SAMPLE.PCAP FILES:

[sudo suricata -r sample.pcap -S custom.rules -k none]

AND THIS IS WHAT OUR OUTPUT IS. AS PER USUAL, LET'S EXAMINE THIS CLOSELY.

```
analyst@aa3669282539:~$ sudo suricata -r sample.pcap -S custom.rules
 -k none
12/3/2026 -- 06:39:15 - <Notice> - This is Suricata version 6.0.1 RE
LEASE running in USER mode
12/3/2026 -- 06:39:16 - <Notice> - all 2 packet processing threads,
4 management threads initialized, engine started.
12/3/2026 -- 06:39:16 - <Notice> - Signal Received.  Stopping engine
.
12/3/2026 -- 06:39:16 - <Notice> - Pcap-file module read 1 files, 20
0 packets, 54238 bytes
```

THIS COMMAND STARTS THE SURICATA APPLICATION AND PROCESSES THE SAMPLE.PCAP FILE USING THE RULES IN THE CUSTOM.RULES FILE. IT RETURNS AN OUTPUT STATING HOW MANY PACKETS WERE PROCESSED IN SURICATA.

LET'S CHECK THE COMPONENTS OF THAT COMMAND FOR BETTER UNDERSTANDING.

- THE `-R SAMPLE.PCAP` OPTION SPECIFIES AN INPUT FILE TO MIMIC NETWORK TRAFFIC. IN THIS CASE, THE `SAMPLE.PCAP` FILE.
- THE `-S CUSTOM.RULES` OPTION INSTRUCTS SURICATA TO USE THE RULES DEFINED IN THE `CUSTOM.RULES` FILE.
- THE `-K NONE` OPTION INSTRUCTS SURICATA TO DISABLE ALL CHECKSUM CHECKS.

ALSO! BIG NOTE!:

"CHECKSUMS" ARE A WAY TO DETECT IF A PACKET HAS BEEN MODIFIED IN TRANSIT. JUST IN CASE YOU FORGET YOU KNOW? MOVING ON.

SURICATA ADDS A NEW ALERT LINE TO THE `/VAR/LOG/SURICATA/FAST.LOG` FILE WHEN ALL CONDITIONS IN ANY PART OF THE RULES ARE MET.

SO LET'S LIST THE FILES IN THE `/VAR/LOG/SURICATA` FOLDER AGAIN

[`ls -l /var/log/suricata`]

```
analyst@aa3669282539:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1418 Mar 12 06:39 eve.json
-rw-r--r-- 1 root root  292 Mar 12 06:39 fast.log
-rw-r--r-- 1 root root 3239 Mar 12 06:39 stats.log
-rw-r--r-- 1 root root 1512 Mar 12 06:39 suricata.log
```

THERE ARE NOW FOUR FILES IN THE DIRECTORY, SUCH AS THE `FAST.LOG` AND `EVE.JSON` FILES. I STILL HAVE THE FEELING I NEED TO EXAMINE THESE FILES IN MORE DETAIL. AND THUS, I WILL.

I'LL DO THIS BY USING THE CAT COMMAND TO DISPLAY THE FAST.LOG FILE GENERATED BY SURICATA:

[cat /var/log/suricata/fast.log]

```
analyst@aa3669282539:~$ cat /var/log/suricata/fast.log
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**] [Class
ification: (null)] [Priority: 3] {TCP} 172.21.224.2:49652 -> 142.250
.1.139:80
11/23/2022-12:38:58.958203  [**] [1:12345:3] GET on wire [**] [Class
ification: (null)] [Priority: 3] {TCP} 172.21.224.2:58494 -> 142.250
.1.102:80
```

AHHH. THE SIGHT OF ALERT ENTRIES. EACH OF THESE LINES OR ENTRIES CORRESPOND TO AN ALERT GENERATED BY SURICATA WHEN IT PROCESSES A PACKET THAT MEETS THE REQUIREMENTS OF AN ALERT GENERATING RULE. EACH ALERT LINE INCLUDES THE MESSAGE THAT IDENTIFIES THE RULE THAT TRIGGERED THE ALERT ALONG WITH THE SOURCE, DESTINATION, AND DIRECTION OF THE TRAFFIC.

NOW THEN, LET'S EXAMINE THE OUTPUT IN THE EVE.JSON FILE. THIS IS THE MAIN SURICATA LOG FILE AND CONTAINS MORE DATA THAN THE FAST.LOG FILE. THIS DATA IS STORED IN A JSON FORMAT AND THAT MAKES IT VERY USEFUL FOR ANALYSIS AND PROCESSING BY OTHER APPLICATIONS.

PSSSSST... KEEP SCROLLING.

LET'S USE THIS CAT COMMAND TO DISPLAY ENTRIES IN THE EVE.JSON FILE:

[cat /var/log/suricata/eve.json]

```
analyst@aa3669282539:~$ cat /var/log/suricata/eve.json
{"timestamp":"2022-11-23T12:38:34.624866+0000","flow_id":12627572326
70869,"pcap_cnt":70,"event_type":"alert","src_ip":"172.21.224.2","sr
c_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":"cu
rl/7.74.0","http_content_type":"text/html","http_method":"GET","prot
ocol":"HTTP/1.1","status":301,"redirect":"https://opensource.google/
","length":223},"app_proto":"http","flow":{"pkts_toserver":4,"pkts_t
oclient":3,"bytes_toserver":357,"bytes_toclient":788,"start":"2022-1
1-23T12:38:34.620693+0000"}}
{"timestamp":"2022-11-23T12:38:58.958203+0000","flow_id":16838453484
4660,"pcap_cnt":151,"event_type":"alert","src_ip":"172.21.224.2","sr
c_port":58494,"dest_ip":"142.250.1.102","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":"cu
rl/7.74.0","http_content_type":"text/html","http_method":"GET","prot
ocol":"HTTP/1.1","status":301,"redirect":"https://opensource.google/
","length":223},"app_proto":"http","flow":{"pkts_toserver":4,"pkts_t
oclient":3,"bytes_toserver":357,"bytes_toclient":797,"start":"2022-1
1-23T12:38:58.955636+0000"}}
```

NOW THAT... IS A LOT OF DATA. RAW DATA THAT IS. HONESTLY READER, I CANNOT UNDERSTAND THIS, EVEN WITH THE HIGHEST SUBSCRIPTION OF ROSETTA STONE. WAIT... I HAVE AN IDEA. LET'S USE A COMMAND BUT WITH [JQ].

(TO SCHOOL YOU ON SOMETHING, THE COMMAND [JQ] STANDS FOR JSON QUERY AND IS USED TO PROCESS, FILTER, MAP, AND TRANSFORM JSON DATA.)

HERE'S THE COMMAND BY THE WAY:

[jq . /var/log/suricata/eve.json | less]

```
analyst@aa3669282539:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 1262757232670869,
  "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",
```

LOOK AT HOW EASY IT IS TO READ THE DATA NOW OPPOSED TO THE CAT COMMAND! PRETTY NEAT HUH?

NOW, LET'S DO A FEW MORE TASKS AND WE SHOULD BE GOOD.

LET'S USE THE JQ COMMAND TO EXTRACT SPECIFIC EVENT DATA FROM THE EVE.JSON FILE. THIS IS A MUST FOR THE THRIVING AND IMPATIENT SECURITY ANALYST.

[jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]" /var/log/suricata/eve.json]

```
analyst@aa3669282539:~$ jq -c "[.timestamp,.flow_id,.alert.signature
,.proto,.dest_ip]" /var/log/suricata/eve.json
["2022-11-23T12:38:34.624866+0000",1262757232670869,"GET on wire","T
CP","142.250.1.139"]
["2022-11-23T12:38:58.958203+0000",168384534844660,"GET on wire","TC
P","142.250.1.102"]
```

THE JQ COMMAND ABOVE EXTRACTS THE FIELDS SPECIFIED IN THE LIST IN THE SQUARE BRACKETS FROM THE JSON PAYLOAD. THE FIELDS SELECTED ARE THE TIMESTAMP (.TIMESTAMP), THE FLOW ID (.FLOW_ID), THE ALERT SIGNATURE OR MSG (.ALERT.SIGNATURE), THE PROTOCOL (.PROTO), AND THE DESTINATION IP ADDRESS (.DEST_IP).

CONCLUSION:

AFTER A LONG DAY ON CONFIGURING CUSTOM RULES, MY LEAD COMES OVER TO VERIFY MY WORK. AS USUAL, THEY'RE HAPPY WITH THE RESULT AND I CLOCK OUT IN TIME FOR $1 WING TUESDAYS. NOW, ALL JOKES ASIDE, I REFRESHED MY BRAIN ON SOME COOL CONCEPTS TODAY:

THE CREATION OF CUSTOM RULES AND RUNNING THEM IN SURICATA
I MONITORED TRAFFIC CAPTURED IN A PCAP FILE
LASTLY, I EXAMINED THE FAST.LOG AND EVE.JSON OUTPUTS THEY PRODUCE.

OVERALL, I'M HAPPY WITH WHAT I DID TODAY AND I'LL DEFINITELY KEEP IT IN MY ANALYST TOOL BELT FOR NEXT TIME. I PROMISE.

IF YOU ENJOYED THIS LAB, CLICK THE LINK AND SHOOT ME A FOLLOW OR A MESSAGE ON LINKEDIN. I'M HAPPY TO COLLABORATE AND WORK. THANK YOU FOR READING.