Snort Fundamentals

Snort is an open-source tool, which serves as both an Intrusion Detection System (IDS) and Intrusion Prevention System (IPS), but can also function as a packet logger or sniffer, akin to Suricata. By thoroughly inspecting all network traffic, Snort has the capability to identify and log all activity within that traffic, providing a comprehensive view of the situation and detailed logs of all application layer transactions. We require specific rule sets to instruct Snort on how to perform its inspection and what exactly it needs to identify. Snort was created to operate efficiently on both general-purpose and custom hardware.

Snort Operation Modes

Snort typically operates in the following modes:

- Inline IDS/IPS
- Passive IDS
- Network-based IDS
- Host-based IDS (however, Snort is not ideally a host-based IDS. We would recommend opting for more specialized tools for this.)

According to Snort's documentation:

"With certain DAQ modules, Snort is able to utilize two different modes of operation: `passive` and `inline`. `Passive mode` gives Snort the ability to observe and detect traffic on a network interface, but it prevents outright blocking of traffic. `Inline mode` on the other hand, does give Snort the ability to block traffic if a particular packet warrants such an event.

Snort will infer the particular mode of operation based on the options used at the command line. For example, reading from a pcap file with the `-r` option or listening on an interface with `-i` will cause Snort to run in passive mode by default. If the DAQ supports inline, however, then users can specify the `-Q` flag to run Snort inline.

One DAQ module that supports inline mode is `afpacket`, which is a module that gives Snort access to packets received on Linux network devices."

Snort Architecture

In order for Snort to transition from a simple packet sniffer to a robust IDS, several key components were added: `Preprocessor`, `Detection Engine`, `Logging and Alerting System`, and various `Output modules`.

- The packet sniffer (which includes the Packet Decoder) extracts network traffic, recognizing the structure of each packet. The raw packets that are collected are subsequently forwarded to the `Preprocessors`.
- `Preprocessors` within Snort identify the type or behaviour of the forwarded packets. Snort has an array of `Preprocessor` plugins, like the HTTP plugin that distinguishes HTTP-related packets or the `port_scan Preprocessor` which identifies potential port scanning attempts based on predefined protocols, types of scans, and thresholds. After the `Preprocessors` have completed their task, information is passed to the `Detection Engine`. The configuration of these `Preprocessors` can be found within the Snort configuration file, `snort.lua`.
- The `Detection Engine` compares each packet with a predefined set of Snort rules. If a match is found, information is forwarded to the `Logging and Alerting System`.
- The `Logging and Alerting System` and `Output modules` are in charge of recording or triggering alerts as determined by each rule action. Logs are generally stored in `syslog` or `unified2` formats or directly in a database. The `Output modules` are configured within the Snort

? Go to Questions

Table of Contents

Introduction To IDS/IPS

Suricata

Suricata Fundamentals

Suricata Rule Development Part 1

Suricata Rule Development Part 2 (Encrypted Traffic)

Snort

Snort Fundamentals

Snort Rule Development

Zeek

Zeek Fundamentals

Intrusion Detection With Zeek

Skills Assessment

Skills Assessment - Suricata

Skills Assessment - Snort

Skills Assessment - Zeek

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

configuration file, `snort.lua`.

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, let's SSH into the Target IP using the provided credentials. The vast majority of the commands covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

Snort Configuration & Validating Snort's Configuration

Snort offers a wide range of configuration options, and fortunately, the open-source Snort 3 provides users with pre-configured files to facilitate a quick start. These default configuration files, namely `snort.lua` and `snort_defaults.lua`, serve as the foundation for setting up Snort and getting it operational in no time. They provide a standard configuration framework for Snort users.

The `snort.lua` file serves as the principal configuration file for Snort. This file contains the following sections:

- Network variables
- Decoder configuration
- Base detection engine configuration
- Dynamic library configuration
- Preprocessor configuration
- Output plugin configuration
- Rule set customization
- Preprocessor and decoder rule set customization
- Shared object rule set customization

Let's browse the `snort.lua` file residing in this section's target as follows.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ sudo more /root/snorty/etc/snort/snort.lua
--------------------------------------------------------------------------
-- Snort++ configuration
--------------------------------------------------------------------------

-- there are over 200 modules available to tune your policy.
-- many can be used with defaults w/o any explicit configuration.
-- use this conf as a template for your specific configuration.

-- 1. configure defaults
-- 2. configure inspection
-- 3. configure bindings
-- 4. configure performance
-- 5. configure detection
-- 6. configure filters
-- 7. configure outputs
-- 8. configure tweaks

--------------------------------------------------------------------------
-- 1. configure defaults
--------------------------------------------------------------------------

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = 'any'

-- set up the external network addresses.
-- (leave as "any" in most situations)
EXTERNAL_NET = 'any'

include 'snort_defaults.lua'

--------------------------------------------------------------------------
-- 2. configure inspection
--------------------------------------------------------------------------

-- mod = { } uses internal defaults
-- you can see them with snort --help-module mod
```

```
-- mod = default_mod uses external defaults
-- you can see them in snort_defaults.lua

-- the following are quite capable with defaults:

stream = { }
stream_ip = { }
stream_icmp = { }
stream_tcp = { }
stream_udp = { }
stream_user = { }
stream_file = { }
---SNIP---
```

Enabling and fine-tuning Snort `modules` is a significant aspect of the configuration process. To explore the complete list and get a brief description of all Snort 3 modules, you can use the following command.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ snort --help-modules
ack (ips_option): rule option to match on TCP ack numbers
active (basic): configure responses
address_space_selector (policy_selector): configure traffic processing based on address space
alert_csv (logger): output event in csv format
alert_fast (logger): output event with brief text format
alert_full (logger): output event with full packet dump
alert_json (logger): output event in json format
alert_syslog (logger): output event to syslog
alert_talos (logger): output event in Talos alert format
alert_unixsock (logger): output event over unix socket
alerts (basic): configure alerts
appid (inspector): application and service identification
appids (ips_option): detection option for application ids
arp (codec): support for address resolution protocol
arp_spoof (inspector): detect ARP attacks and anomalies
---SNIP---
```

These modules are enabled and configured within the `snort.lua` configuration file as Lua table literals. If a module is initialized as an empty table, it implies that it is utilizing its predefined "default" settings. To view these default settings, you can utilize the following command.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ snort --help-config arp_spoof
ip4 arp_spoof.hosts[].ip: host ip address
mac arp_spoof.hosts[].mac: host mac address
```

Passing (and validating) configuration files to Snort can be done as follows.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ snort -c /root/snorty/etc/snort/snort.lua --daq-dir /usr/local/lib/daq
--------------------------------------------------
o")~   Snort++ 3.1.64.0
--------------------------------------------------
Loading /root/snorty/etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        output
        ips
        classifications
        references
        binder
        file_id
        ftp_server
        smtp
        port_scan
        gtp_inspect
        dce_smb
        s7commplus
        modbus
        ssh
        active
        alerts
        daq
```

```
        daq
        decode
        host_cache
        host_tracker
        hosts
        network
        packets
        process
        search_engine
        so_proxy
        stream_icmp
        normalizer
        stream
        stream_ip
        stream_tcp
        stream_udp
        stream_user
        stream_file
        arp_spoof
        back_orifice
        dns
        imap
        netflow
        pop
        rpc_decode
        sip
        ssl
        telnet
        cip
        dnp3
        iec104
        mms
        dce_tcp
        dce_udp
        dce_http_proxy
        dce_http_server
        ftp_client
        ftp_data
        http_inspect
        http2_inspect
        file_policy
        js_norm
        appid
        wizard
        trace
Finished /root/snorty/etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
--------------------------------------------------
ips policies rule stats
             id  loaded  shared enabled    file
              0     208       0     208    /root/snorty/etc/snort/snort.lua
--------------------------------------------------
rule counts
       total rules loaded: 208
              text rules: 208
            option chains: 208
            chain headers: 1
--------------------------------------------------
service rule counts          to-srv  to-cli
                  file_id:      208     208
                    total:      208     208
--------------------------------------------------
fast pattern groups
               to_server: 1
               to_client: 1
--------------------------------------------------
search engine (ac_bnfa)
                instances: 2
                 patterns: 416
            pattern chars: 2508
               num states: 1778
         num match states: 370
             memory scale: KB
             total memory: 68.5879
          pattern memory: 18.6973
        match list memory: 27.3281
        transition memory: 22.3125
appid: MaxRss diff: 3084
appid: patterns loaded: 300
--------------------------------------------------
pcap DAQ configured to passive.
```

```
Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

Note: `--daq-dir /usr/local/lib/daq` is not required to pass and validate a configuration file. It is added so that we can replicate the command in this section's target.

Since we mentioned `DAQ`, Snort 3 should know where to find the appropriate `LibDAQ`. `LibDAQ` is the "Data Acquisition Library", and at a high-level, it's an abstraction layer used by `modules` to communicate with both hardware and software network data sources.

We highly recommend taking the time to read the comments inside the `snort.lua` file as they provide valuable insights.

Snort Inputs

To observe Snort in action, the easiest method is to execute it against a packet capture file. By providing the name of the pcap file as an argument to the `-r` option in the command line, Snort will process the file accordingly.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ sudo snort -c /root/snorty/etc/snort/snort.lua --daq-dir /usr/local/lib/daq
[sudo] password for htb-student:
--------------------------------------------------
o")~   Snort++ 3.1.64.0
--------------------------------------------------
Loading /root/snorty/etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
---SNIP---
Finished /root/snorty/etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
--------------------------------------------------
ips policies rule stats
              id  loaded  shared enabled    file
               0     208       0     208    /root/snorty/etc/snort/snort.lua
--------------------------------------------------
rule counts
       total rules loaded: 208
              text rules: 208
            option chains: 208
            chain headers: 1
--------------------------------------------------
service rule counts          to-srv  to-cli
                  file_id:      208     208
                    total:      208     208
--------------------------------------------------
fast pattern groups
                to_server: 1
                to_client: 1
--------------------------------------------------
search engine (ac_bnfa)
                instances: 2
                 patterns: 416
            pattern chars: 2508
               num states: 1778
         num match states: 370
             memory scale: KB
             total memory: 68.5879
           pattern memory: 18.6973
        match list memory: 27.3281
        transition memory: 22.3125
appid: MaxRss diff: 3024
appid: patterns loaded: 300
--------------------------------------------------
pcap DAQ configured to read-file.
Commencing packet processing
++ [0] /home/htb-student/pcaps/icmp.pcap
-- [0] /home/htb-student/pcaps/icmp.pcap
--------------------------------------------------
Packet Statistics
--------------------------------------------------
daq
                    pcaps: 1
                 received: 8
                 analyzed: 8
```

```
                   allow: 8
                rx_bytes: 592
-------------------------------------------------
codec
                   total: 8           (100.000%)
                     eth: 8           (100.000%)
                   icmp4: 8           (100.000%)
                    ipv4: 8           (100.000%)
-------------------------------------------------
Module Statistics
-------------------------------------------------
appid
                 packets: 8
       processed_packets: 8
          total_sessions: 1
-------------------------------------------------
binder
               new_flows: 1
                inspects: 1
-------------------------------------------------
detection
                analyzed: 8
-------------------------------------------------
port_scan
                 packets: 8
                trackers: 2
-------------------------------------------------
stream
                   flows: 1
-------------------------------------------------
stream_icmp
                sessions: 1
                     max: 1
                 created: 1
                released: 1
-------------------------------------------------
Summary Statistics
-------------------------------------------------
timing
                 runtime: 00:00:00
                 seconds: 0.033229
                pkts/sec: 241
o")~   Snort exiting
```

Snort also has the capability to listen on active network interfaces. To specify this behavior, you can utilize the `-i` option followed by the names of the interfaces on which Snort should run.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ sudo snort -c /root/snorty/etc/snort/snort.lua --daq-dir /usr/local/lib/daq
-------------------------------------------------
o")~   Snort++ 3.1.64.0
-------------------------------------------------
Loading /root/snorty/etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
---SNIP---
Finished /root/snorty/etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
-------------------------------------------------
ips policies rule stats
             id  loaded  shared enabled    file
              0     208       0     208    /root/snorty/etc/snort/snort.lua
-------------------------------------------------
rule counts
       total rules loaded: 208
              text rules: 208
            option chains: 208
            chain headers: 1
-------------------------------------------------
service rule counts          to-srv  to-cli
                  file_id:      208     208
                    total:      208     208
-------------------------------------------------
fast pattern groups
                to_server: 1
                to_client: 1
```

```
--------------------------------------------------
search engine (ac_bnfa)
                instances: 2
                 patterns: 416
            pattern chars: 2508
               num states: 1778
         num match states: 370
             memory scale: KB
             total memory: 68.5879
            pattern memory: 18.6973
        match list memory: 27.3281
        transition memory: 22.3125
appid: MaxRss diff: 2820
appid: patterns loaded: 300
--------------------------------------------------
pcap DAQ configured to passive.
Commencing packet processing
++ [0] ens160
^C** caught int signal
== stopping
-- [0] ens160
--------------------------------------------------
Packet Statistics
--------------------------------------------------
daq
                 received: 33
                 analyzed: 33
                    allow: 33
                     idle: 9
                 rx_bytes: 2756
--------------------------------------------------
codec
                    total: 33           (100.000%)
                 discards: 2            (  6.061%)
                      arp: 13           ( 39.394%)
                      eth: 33           (100.000%)
                    icmp4: 12           ( 36.364%)
                    icmp6: 3            (  9.091%)
                     ipv4: 17           ( 51.515%)
                     ipv6: 3            (  9.091%)
                      tcp: 5            ( 15.152%)
--------------------------------------------------
Module Statistics
--------------------------------------------------
appid
                 packets: 18
        processed_packets: 16
          ignored_packets: 2
           total_sessions: 3
--------------------------------------------------
arp_spoof
                 packets: 13
--------------------------------------------------
binder
              raw_packets: 15
                new_flows: 3
                 inspects: 18
--------------------------------------------------
detection
                 analyzed: 33
--------------------------------------------------
port_scan
                 packets: 20
                trackers: 8
--------------------------------------------------
stream
                    flows: 3
--------------------------------------------------
stream_icmp
                 sessions: 2
                      max: 2
                  created: 2
                 released: 2
--------------------------------------------------
stream_tcp
                 sessions: 1
                      max: 1
                  created: 1
                 released: 1
             instantiated: 1
                   setups: 1
            data_trackers: 1
              segs_queued: 1
            segs_released: 1
```

```
                max_segs: 1
               max_bytes: 64
-------------------------------------------------
tcp
        bad_tcp4_checksum: 2
-------------------------------------------------
Appid Statistics
-------------------------------------------------
detected apps and services
             Application: Services   Clients    Users      Payloads   Misc       Referred
                 unknown: 1          0          0          0          0          0
-------------------------------------------------
Summary Statistics
-------------------------------------------------
process
                 signals: 1
-------------------------------------------------
timing
                 runtime: 00:00:25
                 seconds: 25.182182
                pkts/sec: 1
o")~   Snort exiting
```

Snort Rules

Snort rules, which resemble Suricata rules, are composed of a `rule header` and `rule options`. Even though Snort rules share similarities with Suricata rules, we strongly suggest studying Snort rule writing from the following resources: https://docs.snort.org/, https://docs.suricata.io/en/latest/rules/differences-from-snort.html. The most recent Snort rules can be obtained from the Snort website or the Emerging Threats website.

In Snort deployments, we have flexibility in managing rules. It's possible to place rules (for example, `local.rules` residing at `/home/htb-student`) directly within the `snort.lua` configuration file using the `ips` module as follows.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ sudo vim /root/snorty/etc/snort/snort.lua
```

Snort Fundamentals

```
---SNIP---
ips =
{
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)

    { variables = default_variables, include = '/home/htb-student/local.rules' }
}
---SNIP---
```

Then, the "included" rules will be automatically loaded.

In our Snort deployment, we have an alternative approach to incorporate rules directly from the command line. We can pass either a single rules file or a path to a directory containing rules files directly to Snort. This can be achieved using two options:

- For a single rules file, we can use the `-R` option followed by the path to the rules file. This allows us to specify a specific rules file to be utilized by Snort.
- To include an entire directory of rules files, we can use the `--rule-path` option followed by the path to the rules directory. This enables us to provide Snort with a directory containing multiple rules files.

Snort Outputs

In our Snort deployment, we may encounter a significant amount of data. To provide a summary of the core output

types, let's explore the key aspects:

- `Basic Statistics`: Upon shutdown, Snort generates various counts based on the configuration and processed traffic. This includes:
  - `Packet Statistics`: It includes information from the DAQ and decoders, such as the number of received packets and UDP packets.
  - `Module Statistics`: Each module keeps track of activity through peg counts, indicating the frequency of observed or performed actions. Examples include the count of processed HTTP GET requests and trimmed TCP reset packets.
  - `File Statistics`: This section provides a breakdown of file types, bytes, and signatures.
  - `Summary Statistics`: It encompasses the total runtime for packet processing, packets per second, and, if configured, profiling data.
- `Alerts`: When rules are configured, it is necessary to enable alerting (using the `-A` option) to view the details of detection events. There are multiple types of alert outputs available, including:
  - `-A cmg`: This option combines `-A fast -d -e` and displays alert information along with packet headers and payload.
  - `-A u2`: This option is equivalent to `-A unified2` and logs events and triggering packets in a binary file, which can be used for post-processing with other tools.
  - `-A csv`: This option outputs fields in comma-separated value format, providing customization options and facilitating pcap analysis.

  To discover the available alert types, we can execute the following command.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ snort --list-plugins | grep logger
logger::alert_csv v0 static
logger::alert_fast v0 static
logger::alert_full v0 static
logger::alert_json v0 static
logger::alert_syslog v0 static
logger::alert_talos v0 static
logger::alert_unixsock v0 static
logger::log_codecs v0 static
logger::log_hext v0 static
logger::log_pcap v0 static
logger::unified2 v0 static
```

- `Performance Statistics`: Beyond the aforementioned outputs, additional data can be obtained. By configuring the `perf_monitor` module, we can capture a customizable set of `peg` counts during runtime. This data can be fed into an external program to monitor Snort's activity without interrupting its operation. The `profiler` module allows tracking of time and space usage by modules and rules. This information is valuable for optimizing system performance. The profiler output appears under `Summary Statistics` during shutdown.

Let's see an example of the `cmg` output.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ sudo snort -c /root/snorty/etc/snort/snort.lua --daq-dir /usr/local/lib/daq
--------------------------------------------------
o")~   Snort++ 3.1.64.0
--------------------------------------------------
Loading /root/snorty/etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
--SNIP---
Finished /root/snorty/etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
Loading /home/htb-student/local.rules:
```

```
Finished /home/htb-student/local.rules:
---------------------------------------------------
ips policies rule stats
             id  loaded  shared enabled    file
              0     209       0     209    /root/snorty/etc/snort/snort.lua
---------------------------------------------------
rule counts
       total rules loaded: 209
              text rules: 209
            option chains: 209
            chain headers: 2
---------------------------------------------------
port rule counts
            tcp     udp    icmp      ip
     any      0       0       1       0
   total      0       0       1       0
---------------------------------------------------
service rule counts          to-srv  to-cli
                  file_id:      208     208
                    total:      208     208
---------------------------------------------------
fast pattern groups
               to_server: 1
               to_client: 1
---------------------------------------------------
search engine (ac_bnfa)
                instances: 2
                 patterns: 416
             pattern chars: 2508
                num states: 1778
         num match states: 370
             memory scale: KB
             total memory: 68.5879
          pattern memory: 18.6973
        match list memory: 27.3281
        transition memory: 22.3125
appid: MaxRss diff: 3024
appid: patterns loaded: 300
---------------------------------------------------
pcap DAQ configured to read-file.
Commencing packet processing
++ [0] /home/htb-student/pcaps/icmp.pcap
06/19-08:45:56.838904 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55107 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:8448  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:57.055699 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30433 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:8448  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:57.840049 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55110 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:8704  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:58.044196 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30436 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:8704  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

06/19-08:45:58.841168 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55113 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:8960  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:59.085428 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30448 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:8960  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:59.841775 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55118 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:9216  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:46:00.042354 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30453 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:9216  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

-- [0] /home/htb-student/pcaps/icmp.pcap
-------------------------------------------------
Packet Statistics
-------------------------------------------------
daq
                   pcaps: 1
                received: 8
                analyzed: 8
                   allow: 8
                rx_bytes: 592
-------------------------------------------------
codec
                   total: 8           (100.000%)
                     eth: 8           (100.000%)
                   icmp4: 8           (100.000%)
                    ipv4: 8           (100.000%)
-------------------------------------------------
Module Statistics
-------------------------------------------------
appid
                 packets: 8
        processed_packets: 8
           total_sessions: 1
-------------------------------------------------
binder
               new_flows: 1
                inspects: 1
-------------------------------------------------
detection
                analyzed: 8
              hard_evals: 8
                  alerts: 8
             total_alerts: 8
                  logged: 8
-------------------------------------------------
port_scan
                 packets: 8
                trackers: 2
```

```
search_engine
        qualified_events: 8
-------------------------------------------------
stream
                   flows: 1
-------------------------------------------------
stream_icmp
                sessions: 1
                     max: 1
                 created: 1
                released: 1
-------------------------------------------------
Summary Statistics
-------------------------------------------------
timing
                 runtime: 00:00:00
                 seconds: 0.042473
                pkts/sec: 188
o")~   Snort exiting
```

The same command but using a `.rules` files that may not be "included" in `snort.lua` is the following.

Snort Fundamentals

```
MisaelMacias@htb[/htb]$ sudo snort -c /root/snorty/etc/snort/snort.lua --daq-dir /usr/local/lib/daq
-------------------------------------------------
o")~   Snort++ 3.1.64.0
-------------------------------------------------
Loading /root/snorty/etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
--SNIP---
Finished /root/snorty/etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
Loading /home/htb-student/local.rules:
Finished /home/htb-student/local.rules:
-------------------------------------------------
ips policies rule stats
             id  loaded  shared enabled    file
              0     209       0     209    /root/snorty/etc/snort/snort.lua
-------------------------------------------------
rule counts
      total rules loaded: 209
             text rules: 209
           option chains: 209
           chain headers: 2
-------------------------------------------------
port rule counts
            tcp     udp    icmp      ip
    any       0       0       1       0
  total       0       0       1       0
-------------------------------------------------
service rule counts          to-srv  to-cli
                 file_id:      208     208
                   total:      208     208
-------------------------------------------------
fast pattern groups
               to_server: 1
               to_client: 1
-------------------------------------------------
search engine (ac_bnfa)
                instances: 2
                 patterns: 416
            pattern chars: 2508
               num states: 1778
         num match states: 370
             memory scale: KB
             total memory: 68.5879
          pattern memory: 18.6973
       match list memory: 27.3281
       transition memory: 22.3125
appid: MaxRss diff: 3024
appid: patterns loaded: 300
-------------------------------------------------
pcap DAQ configured to read-file.
Commencing packet processing
++ [0] /home/htb-student/pcaps/icmp.pcap
```

```
06/19-08:45:56.858904 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55107 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:8448  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:57.055699 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30433 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:8448  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:57.840049 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55110 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:8704  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:58.044196 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30436 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:8704  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:58.841168 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55113 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:8960  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:59.085428 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30448 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:8960  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:45:59.841775 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:0C:29:34:0B:DE -> 00:50:56:E0:14:49 type:0x800 len:0x4A
192.168.158.139 -> 174.137.42.77 ICMP TTL:128 TOS:0x0 ID:55118 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:9216  ECHO

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -

06/19-08:46:00.042354 [**] [1:1000001:1] "ICMP test" [**] [Classification: Generic ICMP event] [Pri
00:50:56:E0:14:49 -> 00:0C:29:34:0B:DE type:0x800 len:0x4A
174.137.42.77 -> 192.168.158.139 ICMP TTL:128 TOS:0x0 ID:30453 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:9216  ECHO REPLY

snort.raw[32]:
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - - -
61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefgh ijklmnop
```

```
71 72 73 74 75 76 77 61  62 63 64 65 66 67 68 69  qrstuvwa bcdefghi
- - - - - - - - - - - -  - - - - - - - - - - - -  - - - - - - - -

-- [0] /home/htb-student/pcaps/icmp.pcap
------------------------------------------------
Packet Statistics
------------------------------------------------
daq
                   pcaps: 1
                received: 8
                analyzed: 8
                   allow: 8
                rx_bytes: 592
------------------------------------------------
codec
                   total: 8            (100.000%)
                     eth: 8            (100.000%)
                   icmp4: 8            (100.000%)
                    ipv4: 8            (100.000%)
------------------------------------------------
Module Statistics
------------------------------------------------
appid
                 packets: 8
       processed_packets: 8
          total_sessions: 1
------------------------------------------------
binder
               new_flows: 1
                inspects: 1
------------------------------------------------
detection
                analyzed: 8
              hard_evals: 8
                  alerts: 8
            total_alerts: 8
                 logged: 8
------------------------------------------------
port_scan
                 packets: 8
                trackers: 2
------------------------------------------------
search_engine
        qualified_events: 8
------------------------------------------------
stream
                   flows: 1
------------------------------------------------
stream_icmp
                sessions: 1
                     max: 1
                 created: 1
                released: 1
------------------------------------------------
Summary Statistics
------------------------------------------------
timing
                 runtime: 00:00:00
                 seconds: 0.042473
                pkts/sec: 188
o")~   Snort exiting
```

Snort Key Features

Key features that bolster Snort's effectiveness include:

- Deep packet inspection, packet capture, and logging
- Intrusion detection
- Network Security Monitoring
- Anomaly detection
- Support for multiple tenants
- Both IPv6 and IPv4 are supported

VPN Servers

⚠ **Warning:** Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Existing PwnBox instances will automatically switch to the new VPN server.

US Academy 3 — Medium Load

PROTOCOL

UDP 1337 | TCP 443

DOWNLOAD VPN CONNECTION FILE

Connect to Pwnbox

Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK — 139ms

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

Download VPN Connection File

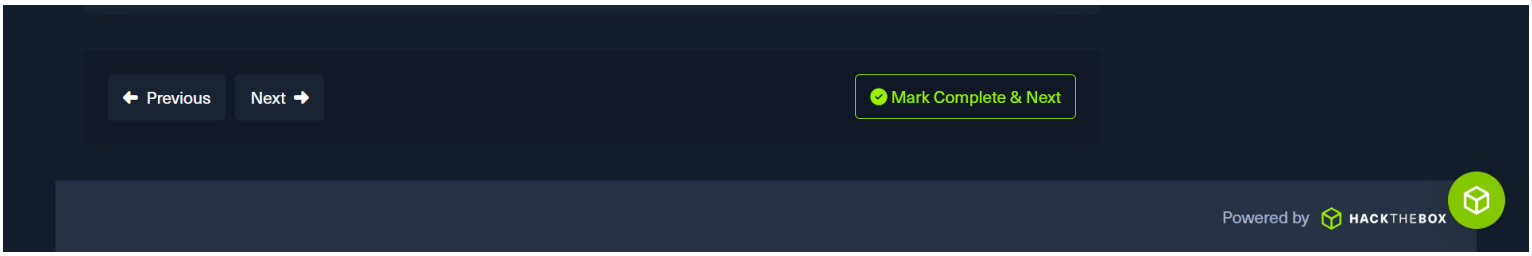
Target(s): Click here to spawn the target system!

SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 1 There is a file named wannamine.pcap in the /home/htb-student/pcaps directory. Run Snort on this PCAP file and enter how many times the rule with sid 1000001 was triggered as your answer.

234

Submit


Previous
Next
Mark Complete & Next
Powered by HACKTHEBOX