Intro to Suricata

Suricata

The term NSM covers a lot of concepts, tools, and techniques. One of these is a traditional

(**I**)ntrusion
(**D**)etection
(**S**)ystem

Suricata

- Open Information Security Foundation
- active community
- truly open source (not tied to corp.)
- fast rule writing / updating cycle
- multithreaded (scaleable)

terminology

NSM: Network Security Monitoring

- gathering network activity
- data about data
- event-based information (bro logs)
- lots of information (PCAP)

terminology

IDS: Intrusion DETECTION
- out-of-band
- passive
  - bruteforcing
  - suspect IP addresses
  - malware detection
  - portscanning

IPS

Intrusion PREVENTION

- in-band
- *active*

The Problem to Solve

- we know what a lot of bad looks like
- network documentation / unknowns
- over-complexity (byod / containers / vms / embedded / etc.)

What is Suricata?

Suricata uses a rule-set to compare against incoming traffic and it fires off alerts if it matches on a rule.

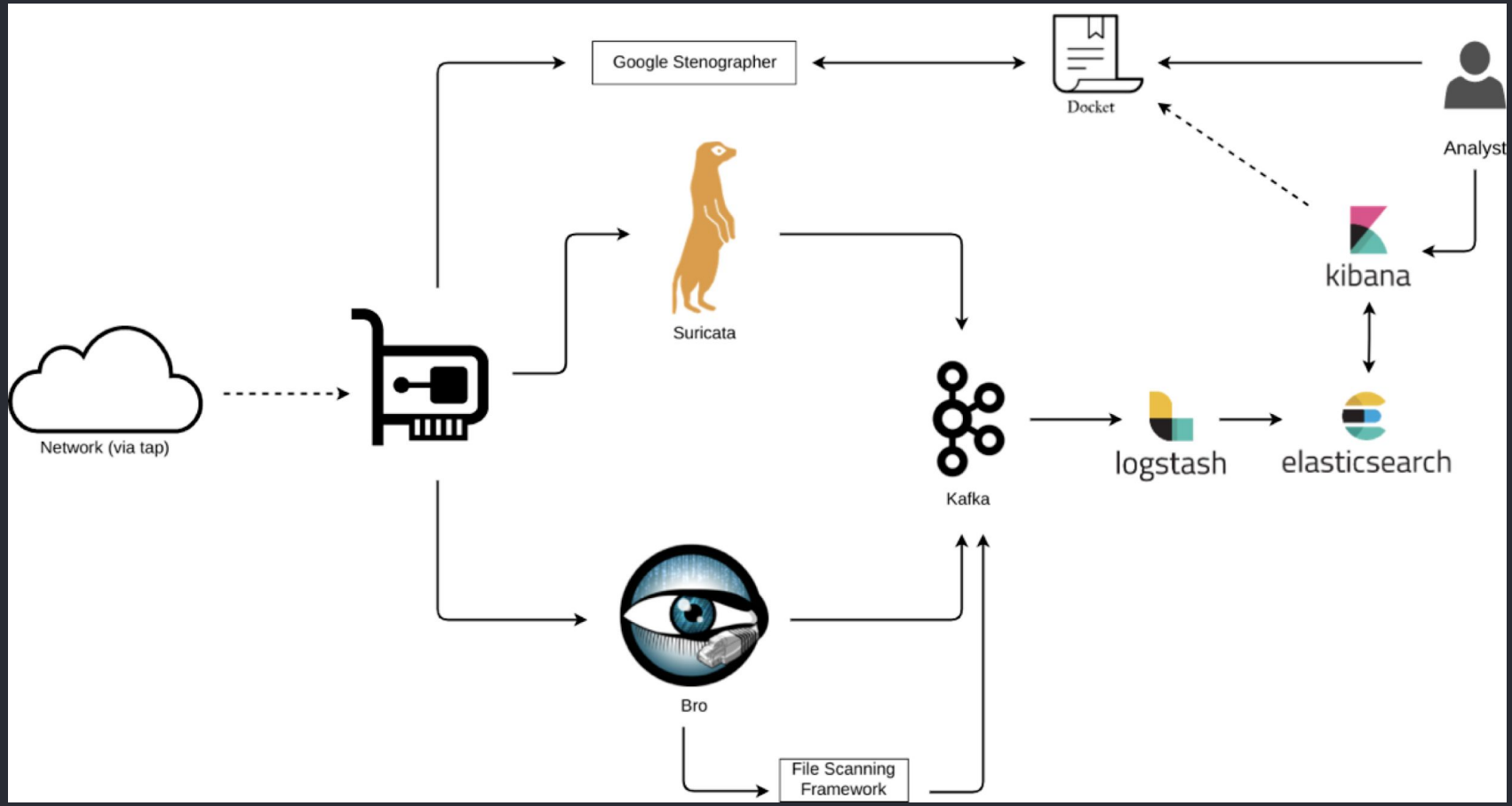
- IDS
- multithreaded! (scalability)
- rules / signatures written to find known bad
- extended Snort language

What it Does

- extracts files (http & smtp)
- ipv4 / v6
- geoip / reputation
- port independent protocol detection

What it Does

- outputs to `eve.json`
- DNS parse / match / log
- "NSM runmode" - just events (no alerting)



Build

requirements:

- `epel-release` yum repository
- emerging threats ruleset
- install `jq` (useful tool for exercises)

Install

add epel repository:

```
sudo yum install -y epel-release
```

install Suricata:

```
sudo yum install -y suricata
sudo yum install -y jq
```


Configure

Primary config file: **/etc/suricata/suricata.yml**

line 15: define local networks
line 55: default-log-dir: /data/suricata/logs
line 59: global stats: `no`
line 75: no # disables a log called fastlog
line 82: eve.log: yes

Configure

/etc/sysconfig/suricata

AFpacket Setting:

add the following line to specify collection interface using AFpacket

```
OPTIONS="--af-packet=interface_name"
```


Configure

create the directory for Suricata to write logs to:

mkdir -p /data/suricata/logs

give the Suricata user permissions to write:

chown -R suricata:suricata /data/suricata

Rules

Suricata alerts are generated by matching behavior defined by **rules**.

We're going to use the Emerging Threats Ruleset:

- crowd-sourced ruleset
- set update interval
- don't run them all!

Rules

```
drop  tcp $HOME_NET any -> $EXTERNAL_NET any
(msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”;
flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “;
pcre:”/NICK .*USA.*[0-9]{3,}/i”;
reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity;
sid:2008124; rev:2;)
```


Actions

Pass

- can be considered a "whitelist"

Drop

- if signature matches it is stopped and drops the packet, generates alert

Reject

- active rejection of the packet, generates alert

Alert - ONLY an alert generated

Header

tcp $HOME_NET any -> $EXTERNAL_NET any

- protocol - tcp / udp / icmp / ip (all) / more
- Source to Destination
- $HOME_NET / EXTERNAL_NET - variable defined in config
- any - port
- direction - `->` and `<>`

Operator Examples

$HOME_NET any -> $EXTERNAL_NET any
$HOME_NET any -> 1.2.3.4 any
$HOME_NET any -> [1.2.3.4,10.11.12.13] any
$HOME_NET any -> !1.2.3.4 any

Source / Dest IPs and Ports

- list of symbols used in describing IPs and ports in the signature header.

Note that the colon (:) is only used with ports and not with source and destination IPs.

!	Negation
[ ]	Describes which parts go together
,	Delineator
:	Range

Port Examples

[21, 69]

[20: 25]

[1024:]

!25

[20: 25,!24]

IP Examples

1.2.3.4

!1.2.3.4

![1.2.3.4, 5.6.7.8]

[192.168.1.100/24, ![192.168.1.102, 192.168.1.103]]

$HOME_NET

Source / Dest IPs and Ports

Note:
Using our signature example, highlight ports and IPs.

Ports example
- tcp $HOME_NET `any` -> any `[21,22]`

IP example
- tcp `$HOME_NET` any -> any `[1.2.3.4,10.11.12.13]`

Rules - Direction

```
source -> destination          # match from source to dest
source <- destination          # match from dest to source
source <> destination       # match on either direction
```

```
tcp $HOME_NET any `->` [1.2.3.4,10.11.12.13] any
```


Rule Options

- information about the signature
- no impact on inspection
- important for data tagging & rule mgmt

Rule Options

- msg: "information about a signature";
- sid:4;
- reference: type, www.website.com;
- priority:1;

Rule Options Example

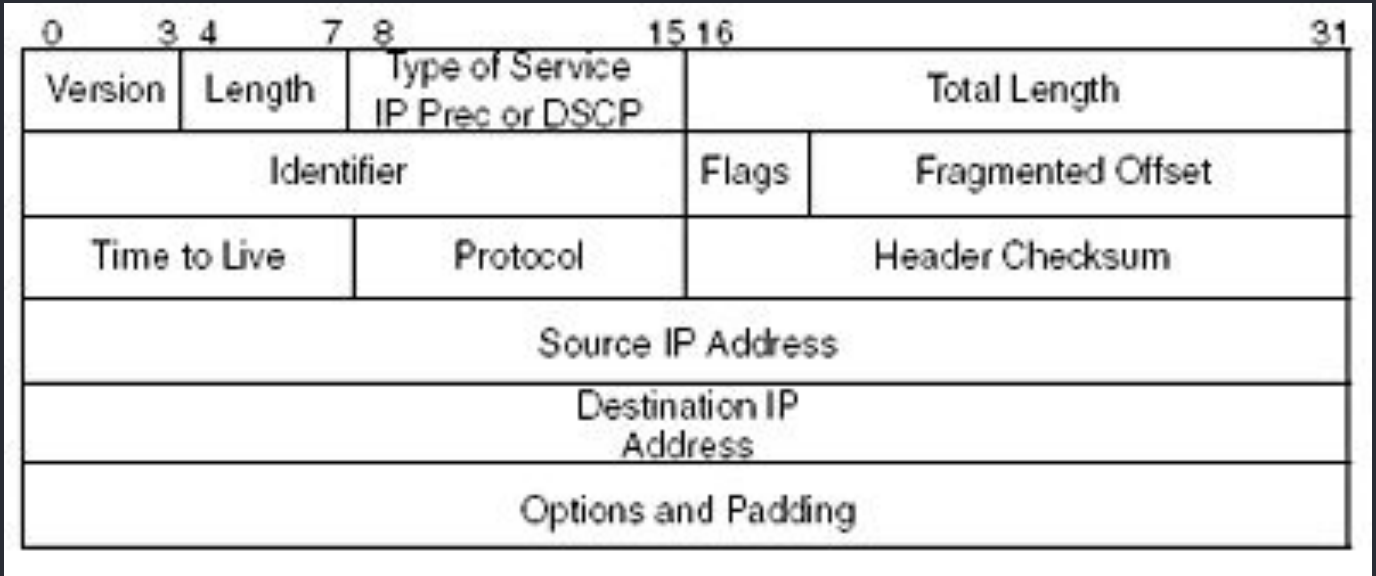
Let's make sense of this:

(msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to_server; \
flowbits:isset,is_proto_irc; content:”NICK “; pcre:”/NICK .*USA.*[0-9]{3,}/i”; \
reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; \
sid:2008124; rev:2;)

Rule Options

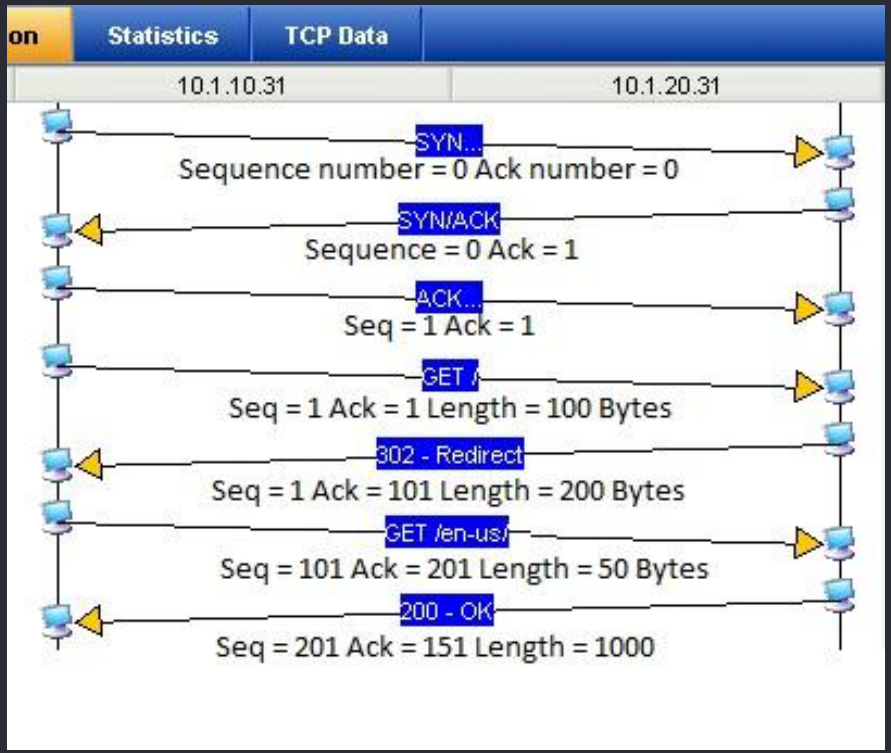
(**msg:** ”ET TROJAN Likely Bot Nick in IRC (USA +..)”;
flow: established,to_server;
flowbits: isset,is_proto_irc;
content: ”NICK “;
pcre: ”/NICK .*USA.*[0-9]{3,}/i”;
reference: url,doc.emergingthreats.net/2008124;
classtype: trojan-activity;
sid: 2008124;
rev: 2;)

IP Header Options

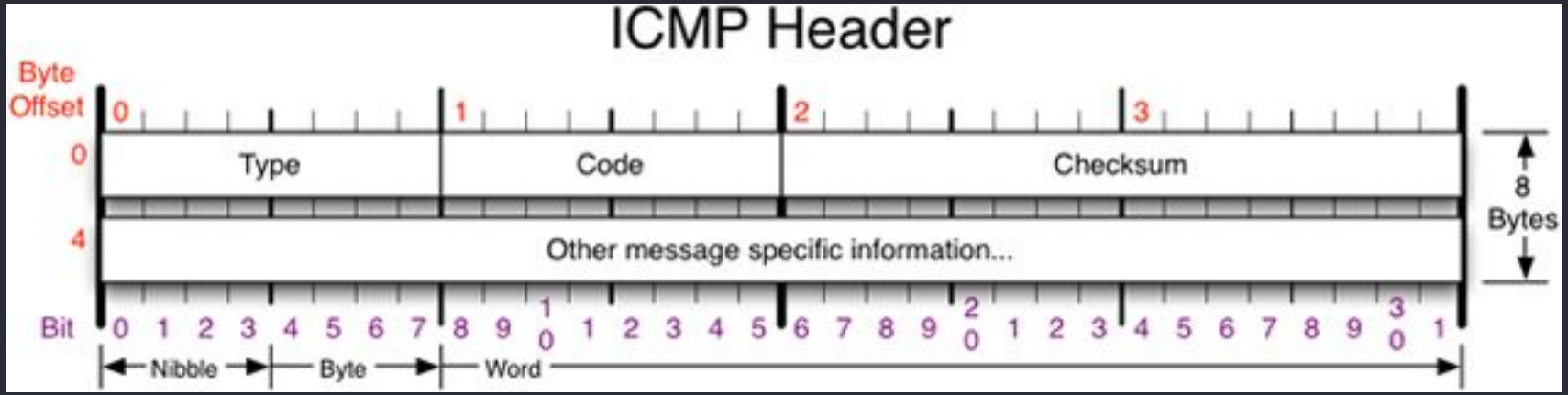


TCP Header

- seq:0;
  - Sequence number
- ack:1;
  - Acknowledgement number
- window:6400;
  - TCP window size



ICMP Header



ICMP Codes

Network Protocol: ICMP: Types

Type	Code	Meaning
0	0	echo reply
3	0	network unreachable
3	1	host is unreachable
3	3	port is unreachable
4	0	source quench
5	0	redirect
8	0	echo request
9/10	0	router discovery/advertisement
11	0	time exceed
12	0	parameter problem
13/14	0	time stamp request
17/18	0	network request/reply

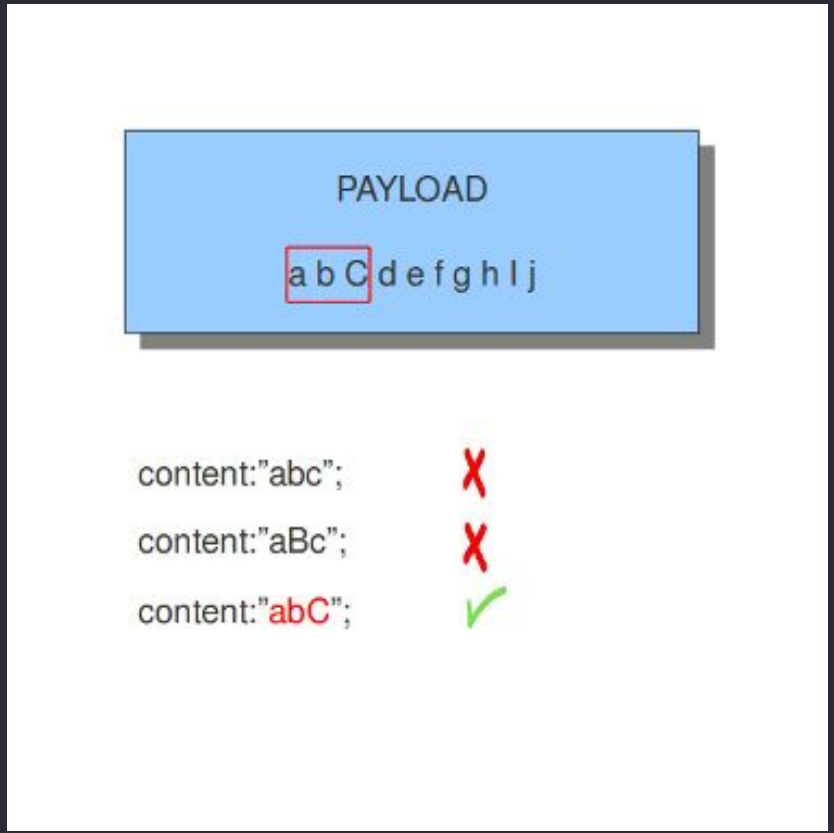
14

Payload Options

Payload keywords inspect the content of the payload of a packet or stream.

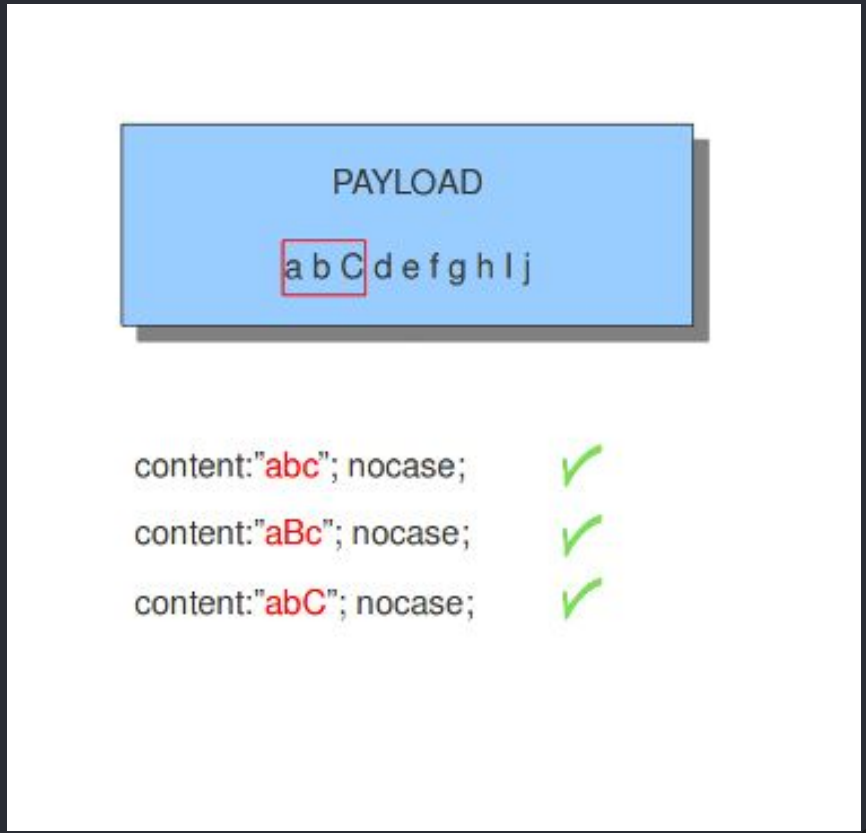
Payload Options

- Content Matching



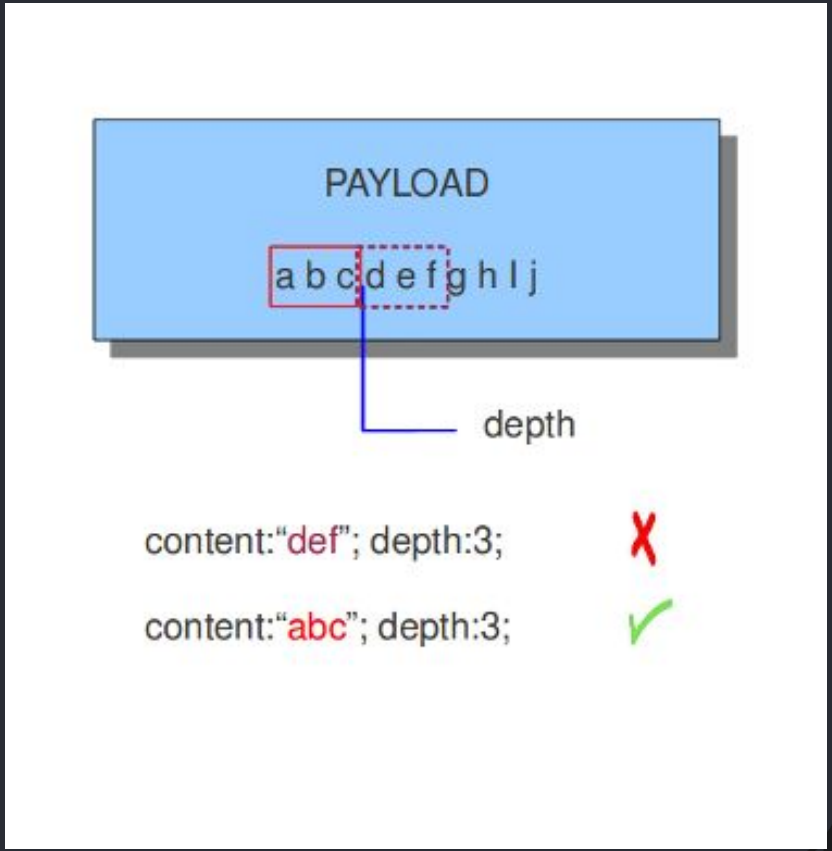
Payload Options

- Nocase



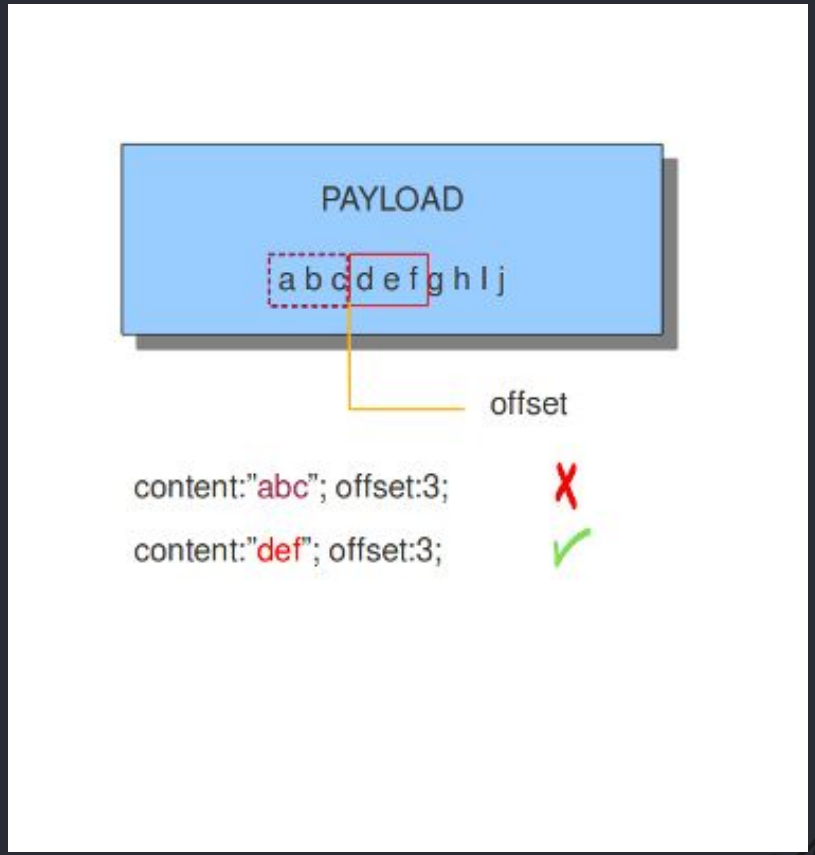
Payload Options

- Depth



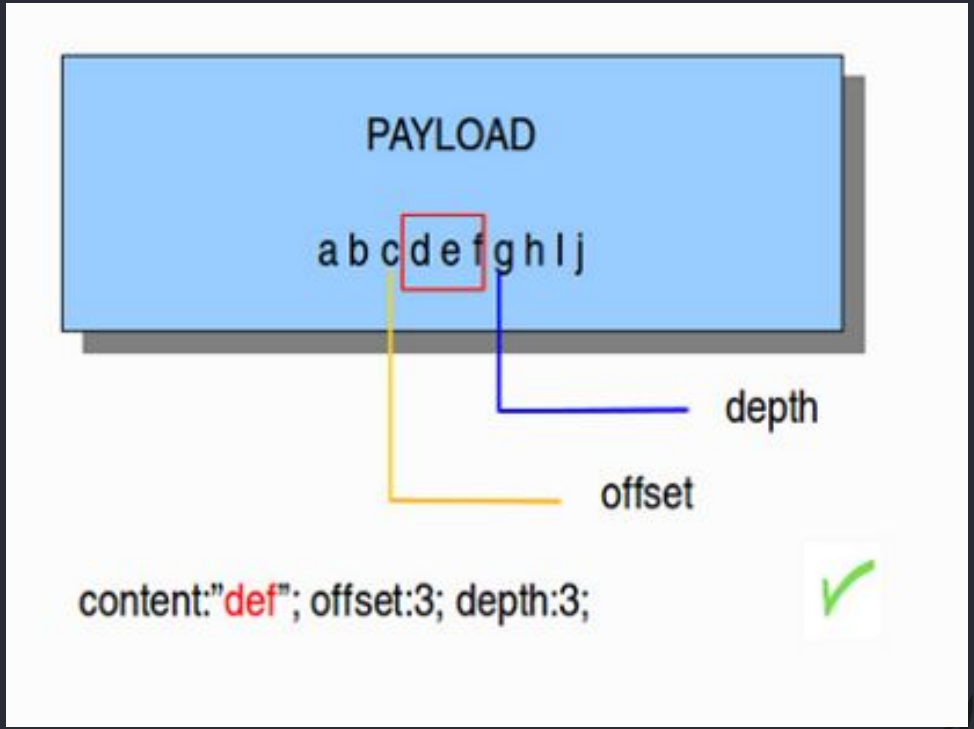
Payload Options

- Offset



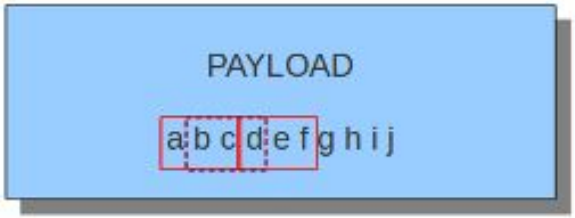
Payload Options

- Offset and Depth



Payload Options

- Distance

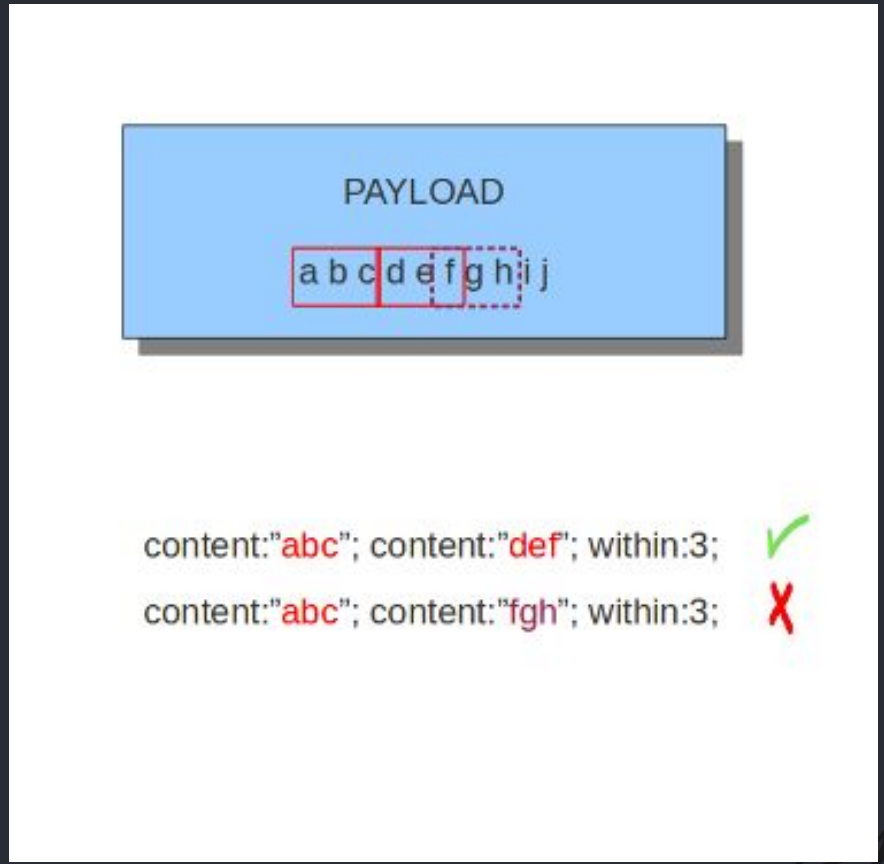


content:"abc"; content:"def"; distance:0; ✓

content:"abc"; content:"bcd"; distance:0; ✗

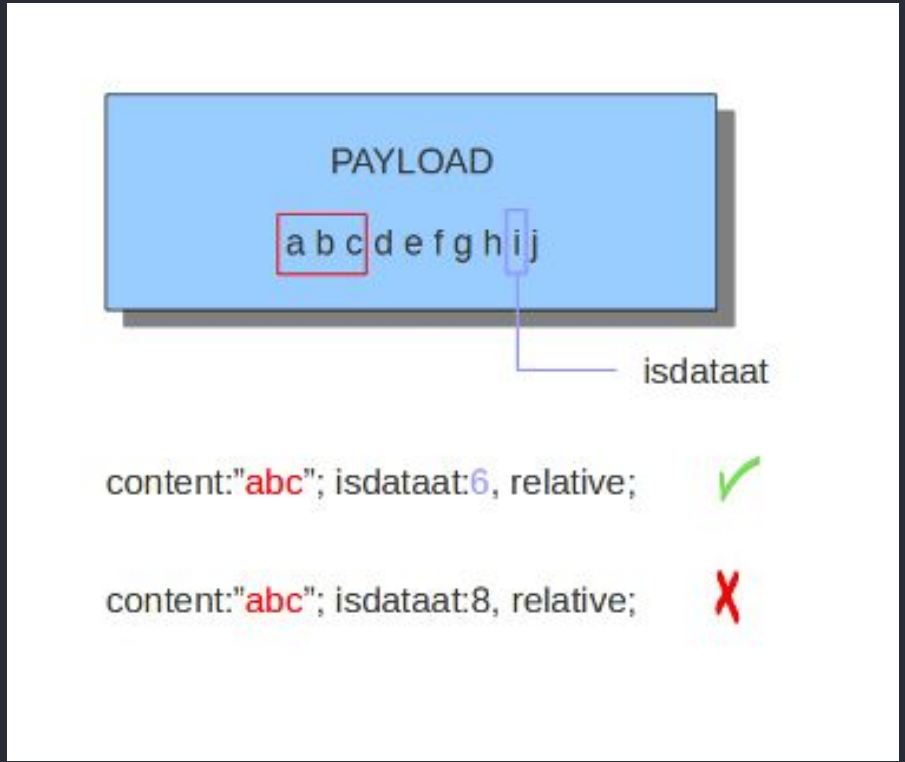
Payload Options

- Within



Payload Options

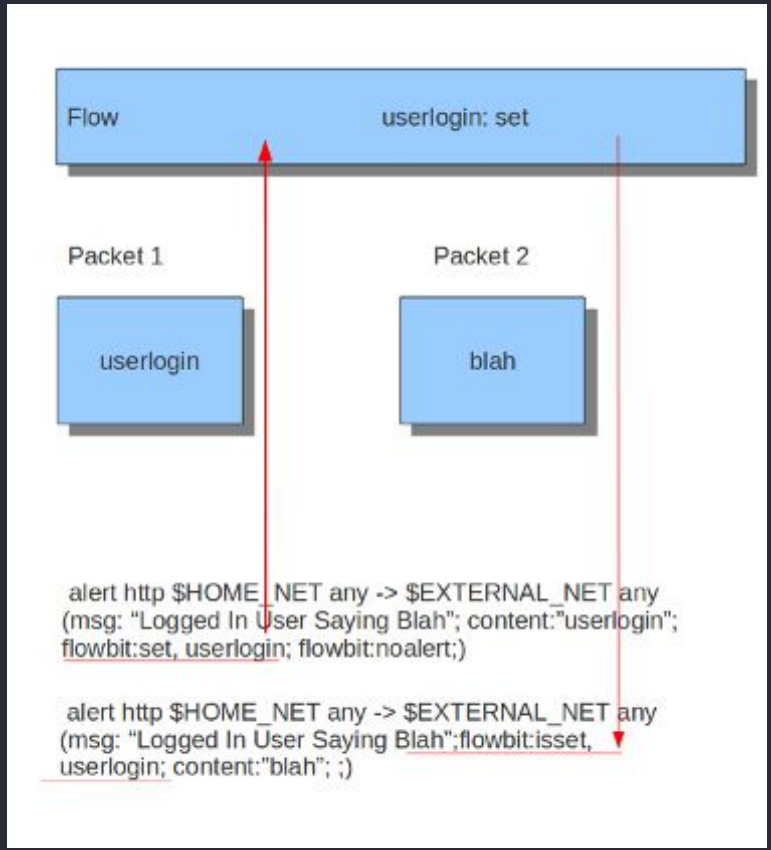
- Isdataat



Flow

- to_client
- to_server
- from_client
- from_server
- established
- not_established
- stateless
- only_stream
- no_stream
- only_frag
- no_frag

Flow Options



HTTP Rule Options

- **content modifiers** which look at the preceding rule options

alert http any any -> any any (content:"index.php"; http_uri; sid:1;)

- **sticky buffer** which applies all following rule options to it

alert http any any -> any any (http_response_line; content:"403 Forbidden"; sid:1;)

HTTP Request Options

GET / HTTP/1.1	HTTP-method, keyword: http_method HTTP-uri, keywords: http_uri or http_raw_uri HTTP-version
Host: www.google.com Connection: keep-alive User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US) AppleWebKit/534.16 (KHTML, like Gecko) Ubuntu/10.10 Chromium/10.0.618.0 Chrome/10.0.618.0 Safari/534.16 Accept-Encoding: gzip,deflate,sdch Accept-Language: en-US,en;q=0.8 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3	HTTP-header, keywords: http_header, http_raw_header User-Agent (part of HTTP-header), keyword: http_user_agent
Cookie: PREF=ID=efe36c63a3bfa6a4:U=aa0cf39996084d7e:TM=1252314621:LM=1292956821:GM=1:S=dYtecyNBioerA47b	HTTP-cookie, keyword: http_cookie

HTTP Response Options

HTTP/1.1 302 Found	HTTP-version HTTP-response code, keyword: http_stat_code HTTP-response message, keyword: http_stat_msg
Location: http://www.google.nl/ Cache-Control: private Content-Type: text/html; charset=UTF-8 Set-Cookie: PREF=ID=efe36c63a3bfa6a4:FF=0:TM=1252314621:LM=1293104406:GM=1:S=xeKyIaBhZkPrZEkN; expires=Sat, 22-Dec-2012 11:40:06 GMT; path=/; domain=.google.com Date: Thu, 23 Dec 2010 11:40:06 GMT Server: gws Content-Length: 218 X-XSS-Protection: 1; mode=block	HTTP-header, keywords: http_header, http_raw_header
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8"> <TITLE>302 Moved</TITLE></HEAD><BODY> <H1>302 Moved</H1> The document has moved <A HREF="http://www.google.nl/">here</A>. </BODY></HTML>	HTTP-response body, keywords: file_data, http_server_body

HTTP URI

HTTP User Agent String

DNS Rule Options

alert dns any any -> any any (msg:”Test dns_query option”; **dns_query;** content:”google”; nocase; sid:1;)

Regex Basics

^[a-zA-Z0-9._]+@[a-zA-Z0-9]+\.[a-zA-z]{2,3}$

What is Regex?

- text-based syntax to define patterns
- used to find / find & replace text

Regex Basics

- most basic regex: matching strings

    Analyst Course

- can add OR with `|`

    Analyst Course|Foundations Course

- simplify this with a group

    (Analyst|Foundations) Course

Regex Basics

- ^
- $
- [ ]
- { }
- +
- [^0-9]

Regex Exercise #1

Write an expression to match the following phone number:

212-867-5309

Regex Exercise #2

Write a regex pattern that will match both of the following email addresses:

bill_lumbergh@initech.com
michael.bolton@penetrode.com

Regex Exercise #3

Write a regex pattern that will match all of the following ip addresses:

192.168.20.1
172.16.9.254
10.0.0.1

Regex Bonus

- https://regexr.com/
- you just leveled up your VIM skillz
- everyday cli usage:

egrep & grep -e

Suricata PCRE Example

alert

http any any -> any any

(msg: "HTTP weird top level domain biz or tk";

pcre:"/((w{3})?[\w-._~?#\[\]@!$&'()*+=]+\.(bit|tk))/si";

sid:7;)

Suricata Operation

suricata --help

```
-h              # help
-v              # verbose
-V              # version
-T              # test config
-D              # daemon mode (bg)
-r <path>       # run pcap offline mode
-i <int>        # specify interface
-S <file>       # specify exclusive .rules file to use
```


Group Lab: your first rule

- create local rules folder:

```
mkdir -p ~/suricata/{rules,alerts}
cd ~/suricata/rules
vim test.rules
```


Group Lab: your first rule

- write functions-test rule content:

```
alert tcp any any <> any any \
(msg:"Testing Suricata functionality"; \
nocase; classtype:not-suspicious,Not Suspicious Traffic,4; \
sid:1; rev:1;)
```


Group Lab: your first rule

- here is the exercise pcap:
  /pcap/2017-10-21-traffic-analysis-exercise.pcap

- run suricata against test rule:
  suricata -c /etc/suricata/suricata.yaml -S ./test.rules \
  -r /mnt/pcap/2017<TAB> -l ~/suricata/alerts/

Lab #1 - Basics

- create ~/suricata/rules/ex1.rules file
- write rules to match the following (include `msg` and `sid`):

1. ALL traffic going to and from all IPs
2. ALL traffic from internal IPs going to Destinations of external IPs
3. ALL traffic with source IPs are internal and communicating to port 80

Lab #1 - Solutions

1. alert ip any any <> any any (msg: "Test rule that fires on all traffic"; sid: 1;)

2. alert ip $HOME_NET any -> $EXTERNAL_NET any (msg: "Test rule that fires on internal to external traffic"; sid:2;)

3. alert ip $HOME_NET any -> any 80 (msg: "Test rule that fires on internal to port 80 traffic"; sid:3;)

Lab #2 - Basics

- create ~/suricata/rules/ex2.rules file
- write rules to match the following (include `msg` and `sid`):

1. DNS lookups going to external IPs only
2. DNS lookups going to external IPs with "bit" in the name

Lab #2 - Solutions

1. alert dns $HOME_NET any -> $EXTERNAL_NET any (msg:"Test rule that fires on external DNS"; sid:1;)

2. alert dns $HOME_NET any -> $EXTERNAL_NET any (msg:"Test rule that fires on external DNS with bit in the name"; dns_query; content:"bit"; sid:2;)

Lab #3 - HTTP

- create ~/suricata/rules/ex3.rules file
- write rules to match the following (include `msg` and `sid`):

1. fire when it detects POST HTTP methods
2. fire when it detects POST HTTP methods and has a successful status code
3. fire when it has the response message and code of 301 Moved Permanently

Lab #3 - HTTP Solutions

1. alert http any any -> any any (msg:"HTTP POST methods seen"; content:"POST"; http_method; sid:4;)

2. alert http any any -> any any (msg:"HTTP POST method seen to known bad host"; content:"POST"; http_method; content:"amellet.bit"; http_host; sid:5;)

3. alert http any any -> any any (msg: "HTTP redirect"; http_response_line; content:"301 Moved Permanently"; nocase; sid:6;)

Lab #4 - PCRE

Create a rule using PCRE that matches websites that end in "bit" or "tk"

Lab #4 - PCRE Solution

```
alert http any any -> any any \
(msg: "HTTP weird top level domain bit or tk"; \
pcre:"/((w{3})?[\w-._~?#\[\]@!$&'()*+=]+\.(bit|tk))/si"; sid:7;)
```


Demo: Add Emerging Threats

1. download tarball from class repo:

```
cd ~
curl -L -O classroom.perched.io/repos/emerging.rules.tar.gz
tar -xzvf emerging.rules.tar.gz
```

2. move all the ".rules" files to `/etc/suricata/rules`

3. add the first 5 of the newly added rules to the suricata.yml listed one per line underneath rules-list: section:

Demo: Let's Script It!

```
sudo vi /etc/suricata/update-rules.sh
```


Maintain - Common Issues

- log maintenance
- maintaining rulesets

/etc/logrotate.d/suricata.conf

```
{{ suricata_data_dir }}/*.log {{ suricata_data_dir }}/*.json
{
    rotate 3
    missingok
    nocompress
    create
    sharedscripts
    postrotate
            /bin/kill -HUP $(cat /var/run/suricata.pid)
    endscript
}
```


Managing Rules

- update rules:

```
sudo suricata-update
```

- look at what is available:

```
sudo suricata-update list-sources
```

- fetch master index:

```
sudo suricata-update update-sources
```