IDS with Suricata

v 4.0.0

Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. Configuration & Setup
4. Actions & Header
5. Rule Options
6. HTTP
7. DNS
8. Regex & PCRE

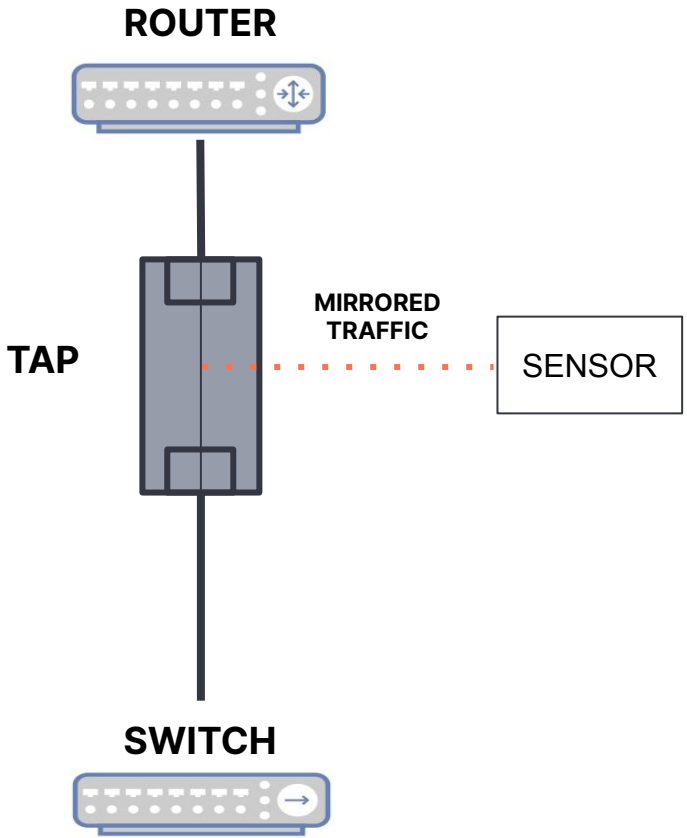
Suricata

1. **Traffic Capture Methods**
2. Suricata Overview
3. Configuration & Setup
4. Actions & Header
5. Rule Options
6. HTTP
7. DNS
8. Regex & PCRE

TAP - Terminal (test) Access Point

- Dedicated hardware device for monitoring networks
- “Forensically” sound (aka. court approved)
- Passive analysis
- Not “networked” (no MAC/IP)
- Designed to be a “listen-in” device

- Expensive
- Can require system interruption to install

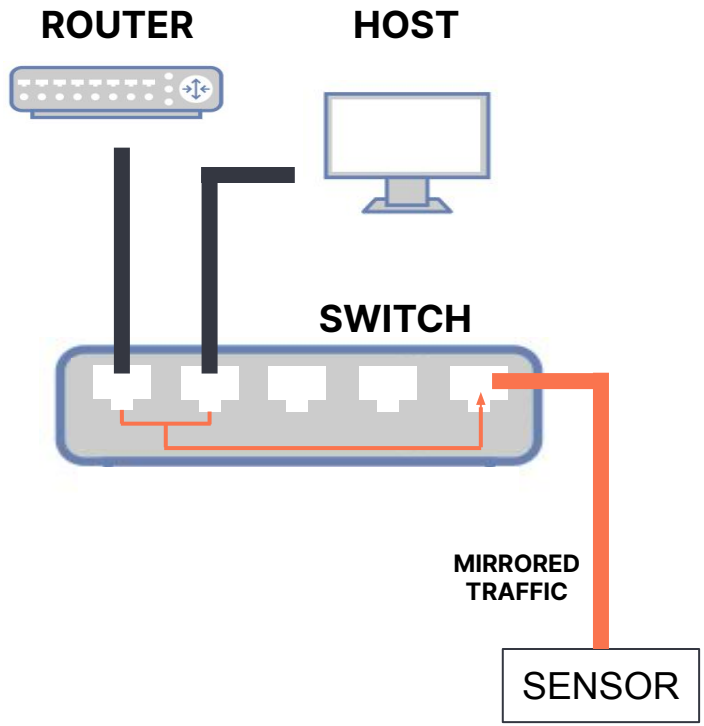


elastic

SPAN / Mirrored Port

- Troubleshooting tool
- Configured interface to monitor ports or VLANs
- No additional hardware
- No impact to implement
- Remotely configurable

- Susceptible to packet loss
- Distortion of real time communication
- “Bad” packets are dropped



Suricata

1. Traffic Capture Methods
2. **Suricata Overview**
3. Configuration & Setup
4. Actions & Header
5. Rule Options
6. HTTP
7. DNS
8. Regex & PCRE

Suricata

- Suricata uses a rule-set to compare against incoming traffic and it fires off alerts if it matches on a rule.
- Open Information Security Foundation
- Active community
- Truly open source (not tied to corp.)
- Fast rule writing / updating cycle
- Extended Snort Language
- Multithreaded (scaleable)

Additional Features

- Extracts files (http & smtp)
- IPv4 / v6
- GeoIP / Reputation
- Port-independent protocol detection
- Outputs to `eve.json`
  - Extensible Event Format
- DNS parse / match / log
- “NSM runmode”
  - just events (no alerting)

Intrusion Detection System (IDS)

- Out-of-band
- Passive
- Able to detect:
  - brute forcing
  - suspect IP addresses
  - malware detection
  - port scanning

Intrusion Prevention System (IPS)

- In-band
- Active
  - Sending an alarm(IDS Mode)
  - Dropping the malicious packets
  - Blocking traffic from the source address
  - Resetting the connection

Suricata vs Snort History

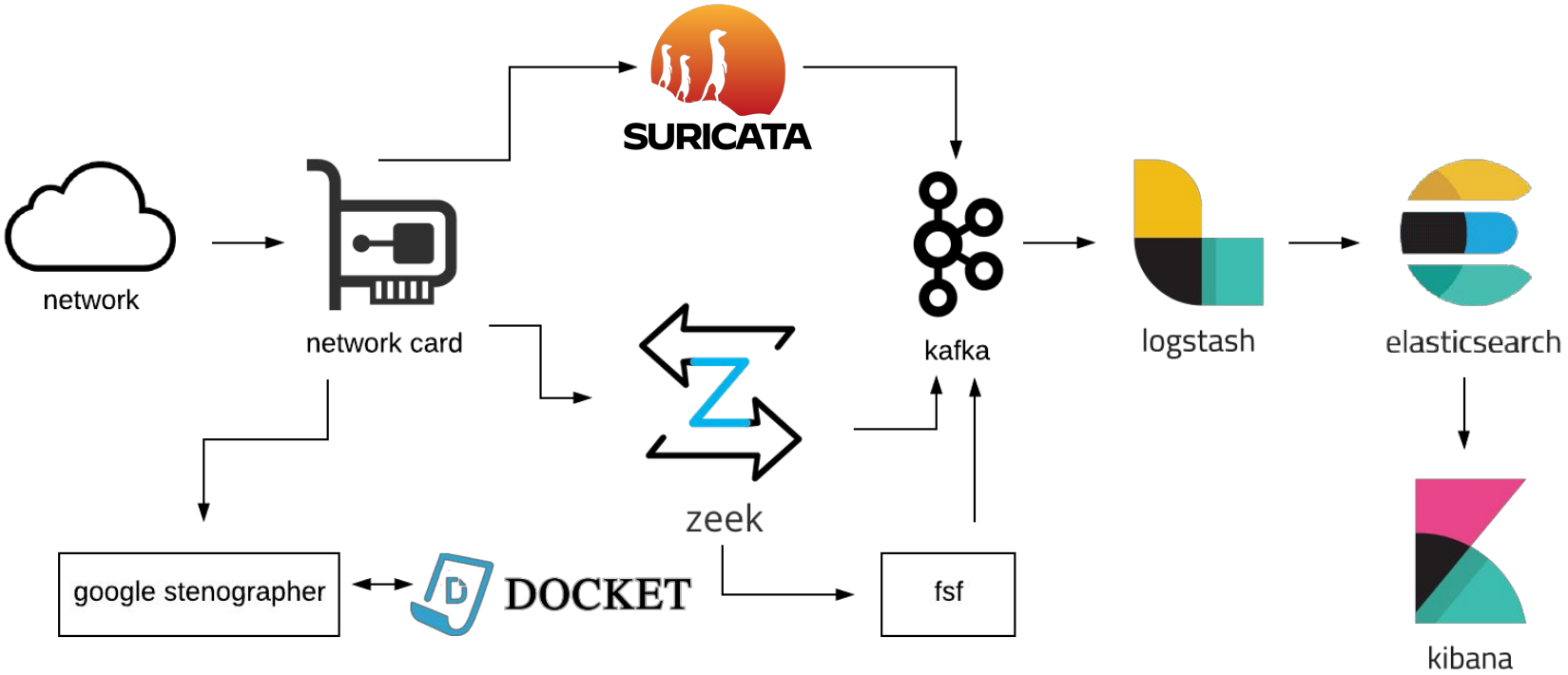
1998

2009

SURICATA

- Port independent protocol analysis
- File extraction for HTTP and SMTP
- Multithreaded (ahead of Snort by a decade)
- Backwards compatible with Snort rules

RockNSM



elastic

The Problem to Solve

- We know what a lot of bad looks like
- Network documentation / unknowns
- Over-complexity
  - bring your own device (BYOD)
  - containers
  - virtual machines (VMs)
  - embedded

CTF: Basics

Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. **Configuration & Setup**
4. Actions & Header
5. Rule Options
6. HTTP
7. DNS
8. Regex & PCRE

Setup Overview

On RHEL family systems, we can install Suricata with yum, but we need to have the EPEL repo.

- Install the “Extra Packages” repo (`epel-release`)
- Install `jq` (additional utility)
- Install some rules… more on this later

Configuration Walkthrough

- Primary config file
  - `/etc/suricata/suricata.yaml`

1. Define variables to define your (internal) network
2. Choose rules to enable or disable
3. Choose your outputs
4. Define capture settings
5. Adjust Application Layer Protocol options (if desired)

Configure

- Configuration files:
  - `/etc/suricata/suricata.yaml`
  - `/etc/sysconfig/suricata`
- Notable entries / variables:
  - network variables
  - default-log-dir
  - **fast.log**
  - **eve.log (eve.json output)**
  - default-rule-path

Reading JSON

```
$ sudo cat eve.json
```


```
{"timestamp":"2017-10-21T04:51:53.314839+0000","flow_id":563194001739223,"pcap_cnt":2,"event_type":"dhcp","src_ip":"10.0.1.254","src_port":67,"dest_ip":"10.0.1.95","dest_port":68,"proto":"UDP","dhcp":{"type":"reply","id":3501026985,"client_mac":"60:a4:4c:6a:b2:1f","assigned_ip":"10.0.1.95","dhcp_type":"ack"}}
```


```
$ sudo cat eve.json | jq .
```


```
{
  "timestamp": "2017-10-21T04:51:53.314839+0000",
  "flow_id": 563194001739223,
  "pcap_cnt": 2,
  "event_type": "dhcp",
  "src_ip": "10.0.1.254",
  "src_port": 67,
  "dest_ip": "10.0.1.95",
  "dest_port": 68,
  "proto": "UDP",
  "dhcp": {
    "type": "reply",
    "id": 3501026985,
    "client_mac": "60:a4:4c:6a:b2:1f",
    "assigned_ip": "10.0.1.95",
    "dhcp_type": "ack"
  }
}
```

Suricata Basic Operation (--help)

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

CTF: Configuration & Execute!

CTF: Setup

Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. Configuration & Setup
4. **Actions & Header**
5. Rule Options
6. HTTP
7. DNS
8. Regex & PCRE

Parts of a Suricata Rule

```
alert http 192.168.1.0/24 any -> !$HOME_NET [80, 8080] (msg: "Test rule"; http.uri; content: "reddit.com"; sid: 1; )
```

- Action
  - what to do on a match
- Header
  - five-tuple info for identifying who is matching
- Rule Options
  - what should the rule match on

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

```
http 192.168.1.0/24 any -> !$HOME_NET [80, 8080]
```

- Protocol
  - ip (all) / tcp / udp / icmp / (many more)
- Source IP & Port
  - `$HOME_NET / $EXTERNAL_NET` (variable defined in config)
  - Any = all
- Direction
  - `->` and `<>`
- Destination IP & Port

IP Header Reserved Characters

Reserved Character	Description
! (exclamation point)	**Negation**
[ ] (square brackets)	**Grouping**
, (comma)	**Delimiter**

IP Header Examples

Example	Description
1.2.3.4	Match on 1.2.3.4
!1.2.3.4	Match all IP addresses except 1.2.3.4
192.168.1.1/24	Match 192.168.1.0 through 192.168.1.255
[1.1.1.1, 2.2.2.2]	Match IP addresses 1.1.1.1 or 2.2.2.2
[1.1.1.0/24, ![1.1.1.2, 1.1.1.3]]	Match IP range 1.1.1.0/24 except 1.1.1.2 or 1.1.1.3
$HOME_NET	Match the IP addresses set by the suricata.yaml file

Port Header Reserved Characters

Reserved Character	Description
! (exclamation point)	**Negation**
[ ] (square brackets)	**Grouping**
: (colon)	**Range**
, (comma)	**Delimiter**

Port Header Examples

Example	Description
25	Match on port 25
!25	Match on all ports except 25
[80, 443]	Match on port 80 or 443
[20:25]	Match on port 20, 21, 22, 23, 24, or 25
[20:25, !24]	Match on port 20, 21, 22, 23, or 25
[1024:]	Match all ports that are equal to and greater than 1024

Direction Header Examples

Example	Description
->	Source to Destination
<>	Any direction

Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. Configuration & Setup
4. Actions & Header
5. **Rule Options**
6. HTTP
7. DNS
8. Regex & PCRE

Rule Options - Meta-Settings

Option	Example
msg	**msg: “This is an example Message”;**
sid	**sid: 1234567;** rev: 9;
reference	reference:url,https://this-page-intentionally-left-blank.org;
priority	priority: 1;
classtype	classtype: suspicious-activity;
metadata	metadata: “this is just ignored and not used in suricata”;
target	target: src_ip;

Rule Options - Reference

reference:cve,**2014-0160**;

```
cat /etc/suricata/reference.config

# config reference: system URL
config reference: bugtraq   http://www.securityfocus.com/bid/
config reference: bid       http://www.securityfocus.com/bid/
config reference: cve       http://cve.mitre.org/cgi-bin/cvename.cgi?name=
config reference: secunia   http://www.secunia.com/advisories/
```

reference:http://cve.mitre.org/cgi-bin/cvename.cgi?name=**2014-0160**;

Rule Options, content option.

- Character matching, **case sensitive**
- Can use multiple content matches
- Searches inside payload/application layer
- Examples:

```
content: “badness”;
```

```
content: “http”; content: “evil”;
```

Rule Options, content option. (Hex)

- Can use hex representation by using pipes
- Some characters must be hex when used:

“ (double quotes)	; (semicolon)	: (colon)	| (pipe)	\ (backslash)
|22|	|3B|	|3A|	|7C|	|5C|

- Example:

```
content: “http|3A|//”;
```

- Looks for “http://”

Rule Options, content option.

- Example:

```
content: “http|3A|//”;
```

- Character matching, case sensitive
- Can use multiple content matches
- Can use hex representation by using pipes
- Some characters must be hex when used:

“ (double quotes)	; (semicolon)	: (colon)	| (pipe)	\ (backslash)
|22|	|3B|	|3A|	|7C|	|5C|

Content Modifier, nocase option

- Example

```
content: “http|3A|//”;
```

- Character matching is case sensitive

Payload	Matched
http://	Yes
HTTP://	No
HtTp://	No

Content Modifier, nocase option

- Example

```
content: “http|3A|//”; nocase;
```

- Character matching no longer case sensitive

Payload	Matched
http://	Yes
HTTP://	Yes
HtTp://	Yes

Content Modifier, depth option

- Example

```
content: “abc”; depth: 6;
```

- How far in from the beginning could the content be
- Will match if abc is in the first 6 bytes

Payload	Matched
abcdefghi	Yes
defghiabc	No
abcdefabc	Yes

Content Modifier, offset option

- Example

```
content: “abc”; offset: 3;
```

- Starts inspection from the specified byte

Payload	Matched
abcdefgh	No
123abc	Yes
123456abc	Yes

Content Modifier, distance option

- Example:

```
content: “abc”; content: “fgh”; distance: 2;
```

- Distance in bytes the content can be from the previous content option

Payload	Matched
abcdefgh0	Yes
abc012fgh	Yes
abcfght12	No

Content Modifier, within option

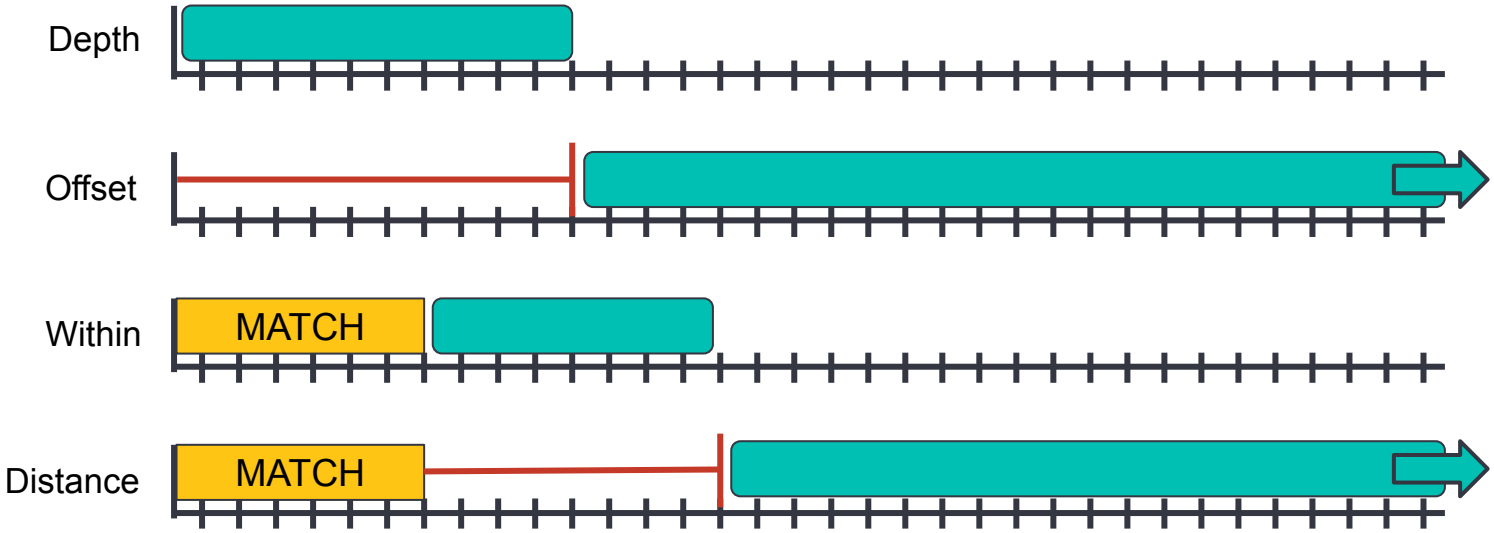
- Example:

```
content: “abc”; content: “def”; within: 6;
```

- Bytes within previous content option
- Cannot be set to zero

Payload	Matched
abc123def	Yes
abcdef	Yes
abc12345def	No

Positioning Modifiers Summary



	How Far to Look	When to Start Looking
From Start of Payload	Depth	Offset
From Previous Match	Within	Distance

Sticky Buffers

- Apply to all payload keywords following the sticky buffer
  - Look forward in the rule
- Use transformations
  - Transformations can modify data in a buffer
- Use “.” to separate protocol from buffer
  - Most keywords become sticky buffers in version 6.x
  - http.stat_msg vs http_stat_msg
- Example
  - `alert http any any -> any any (http.response_line; content: "403 Forbidden"; sid: 1;)`

Content Modifiers, Legacy

- Apply to all payload keywords before the content modifier
  - Look backward in the rule
- Use “_” to separate protocol from the keyword
  - Content modifiers might be seen in rules, as they still function
  - http_stat_msg vs http.stat_msg
- Example
  - `alert http any any -> any any (content: ”403 Forbidden”; http_response_line; sid: 1;)`

Sticky Buffers vs. Keyword Modifiers

- Example
  - Modifier
    - `alert http any any -> any any (content: "403"; http_stat_code; content: "Forbidden"; http_stat_msg; sid: 1)`
  - Sticky Buffer
    - `alert http any any -> any any (http.response_line; content: "403 Forbidden"; sid: 1)`

CTF: Basic Rules

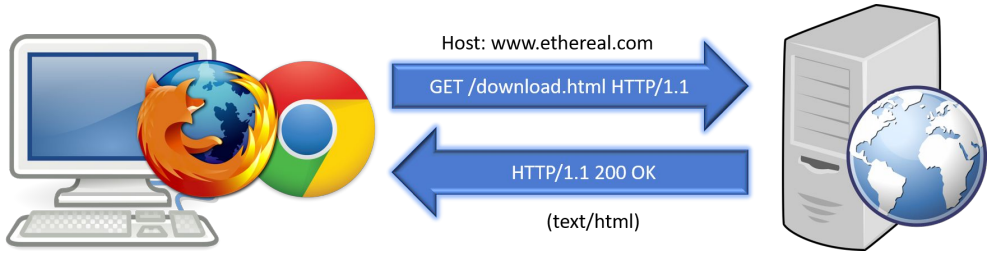
elastic

Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. Configuration & Setup
4. Actions & Header
5. Rule Options
6. **HTTP**
7. DNS
8. Regex & PCRE

Hypertext Transfer Protocol (HTTP)

- Foundation for data communication on the Internet
- Port 80 / 8080 (alt)
- Unencrypted
- Request-Response protocol
  - web browser (client) makes a request
  - website (server) responds with resources
- Stateless protocol
  - some web applications implement states or server side sessions
  - HTTP cookies / hidden variables in web forms



HTTP Request Example

```
GET /mail/15.1.3028.1103/styles/Base/img/thumbnail.jpg HTTP/1.1
Accept: */*
Referer: http://co108w.col108.mail.live.com/mail/InboxLight.aspx?
FolderID=00000000-0000-0000-0000-000000000001&n=1629970862
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Accept-Encoding: gzip, deflate
Host: gfx8.hotmail.com
Connection: Keep-Alive
```

HTTP Request - HTTP Method

http.method

Indicates the desired action to be performed on the identified resource.

GET	requests a specified resource
HEAD	identical to a GET request, but without the response body
POST	appends the enclosed request to a resource
PUT	store enclosed request at the specified URI
DELETE	deletes the specified resource

HTTP Request - Host/URI

HTTP Request - HTTP Referer

```
GET /mail/15.1.3028.1103/styles/Base/img/thumbnail.jpg HTTP/1.1
Accept: */*
Referer: http://co108w.col108.mail.live.com/mail/InboxLight.aspx?
FolderID=00000000-0000-0000-0000-000000000001&n=1629970862
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Accept-Encoding: gzip, deflate
Host: gfx8.hotmail.com
Connection: Keep-Alive
```

http.referer

Identifies the address of the webpage which is linked to the resource being requested

Common Example:
A Google search leads a user to click on one of the results
Google.com is the referer for the new HTTP request to the end site

User Agent

- Software that is acting on behalf of a user
  - web browser
  - email reader (mail user agent)
  - shell application (Powershell, curl, etc.)

Format:

Mozilla/[version] (system/browser info) [platform] (platform details) [extensions]

Example:

Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405

HTTP Response Example

```
HTTP/1.1 200 OK
Content-Length: 757
Content-Type: image/jpeg
Last-Modified: Fri, 30 Oct 2009 02:45:55 GMT
Accept-Ranges: bytes
ETag: "267f2e1ab59ca1:7a69"
Server: Microsoft-IIS/6.0
HMServer: bay0-g1-008
X-Powered-By: ASP.NET
Date: Wed, 18 Nov 2009 17:20:50 GMT
Connection: keep-alive
Cache-Control: public, max-age=31536000, s-maxage=31536000
Expires: Wed, 10 Nov 2010 12:00:00 GMT
```

Status Codes

- Informational - 1XX
- Successful - 2XX
- Redirection - 3XX
- Client Error - 4XX
- Server Error - 5XX

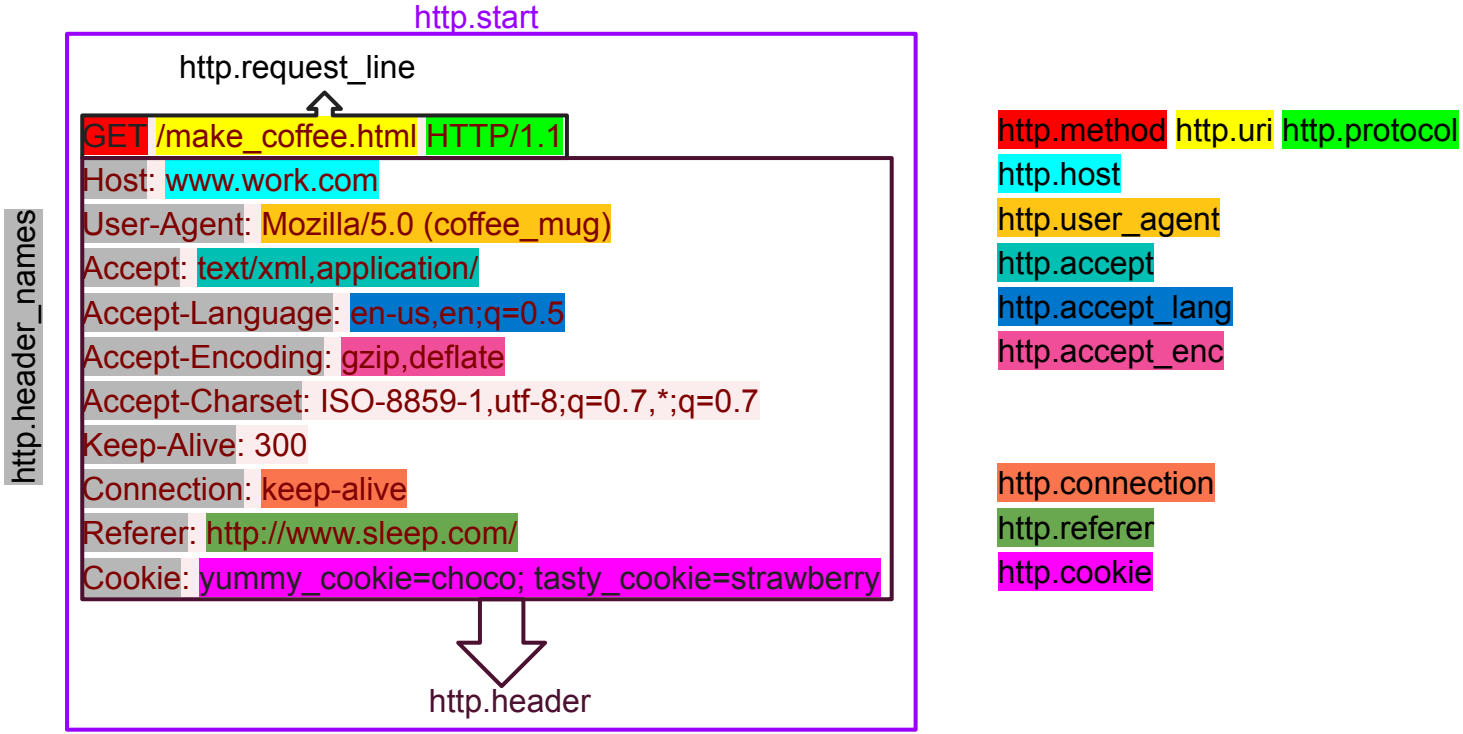
Attack Examples

- HTTP Flood
  - Garbage flood
  - GET flood
- HTTP Fuzzing / Misbehaved fields
  - G3T request
  - HTTP version 1,1
- Low and Slow (Slowloris)
  - G… E… T…
  - hold open connections for as long as possible
- Eavesdropping
- Microsoft Email

Suricata HTTP inspections

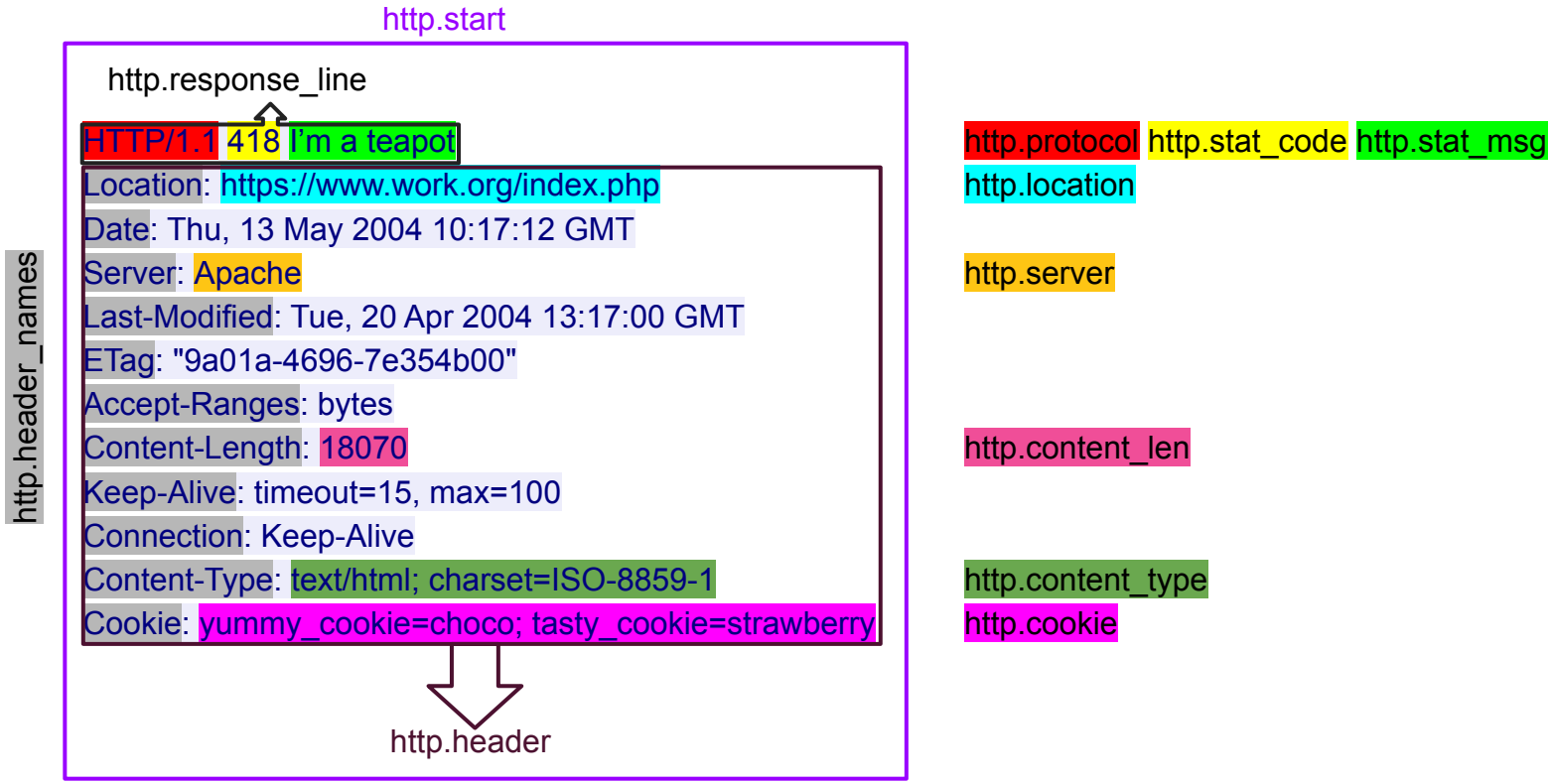
Keyword	Type	Description
http.uri	Sticky buffer	Matches URI after HTTP method (e.g., /index.html)
http.method	Sticky buffer	Matches HTTP method (e.g., GET, POST, etc)
http.request_line	Sticky buffer	Limits matches to request line (e.g., GET /index.html HTTP/1.1)
http.header	Sticky buffer	Matches any HTTP header
http.cookie	Sticky buffer	Matches HTTP cookies
http.user_agent	Sticky buffer	Matches User-Agent header
http.host	Sticky buffer	Matches Host header

Suricata HTTP Request Sticky Buffers



elastic

Suricata HTTP Response Sticky Buffers



elastic

HTTP rules

- When matching HTTP, instead of specifying the port, let Suricata do the work.

```
alert http any any -> any any (msg: "Found HTTP”
sid: 4;)
```

```
alert http any any -> any any (msg: "Found an
iframe!"; content: "iframe"; sid: 5;)
```

CTF: HTTP Rules

HTTP Rules - Review

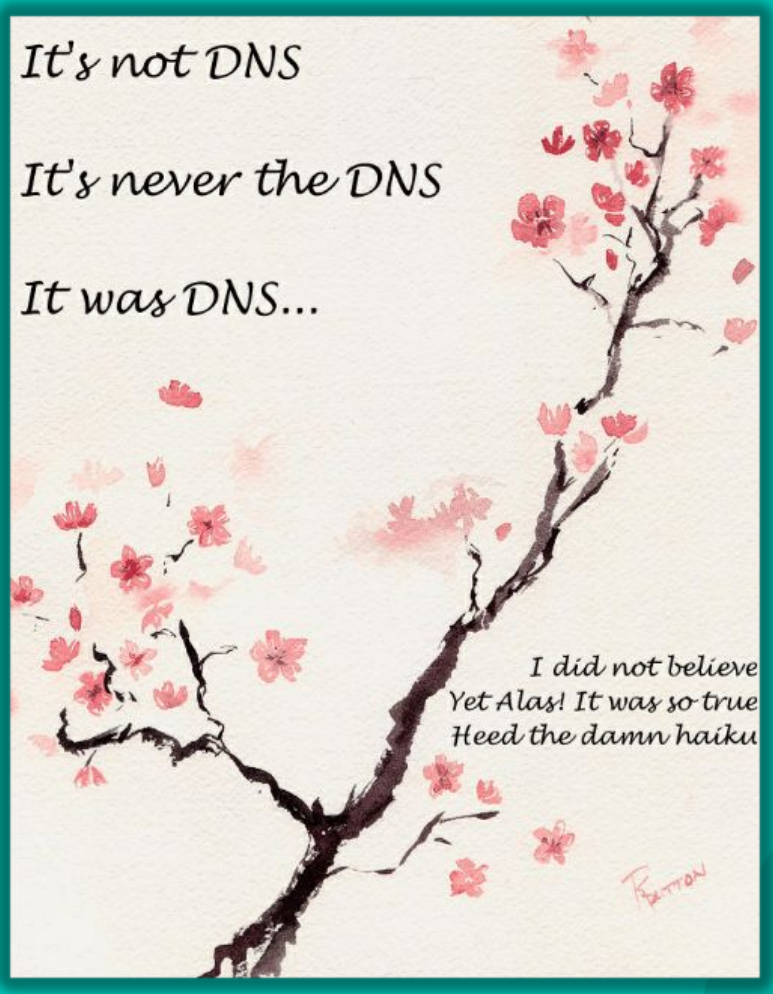
1. Fire when it detects POST HTTP methods

2. Fire when it detects POST HTTP methods to a known bad host "amellet.bit"

3. Fire when it has the response message and code of 301 Moved Permanently

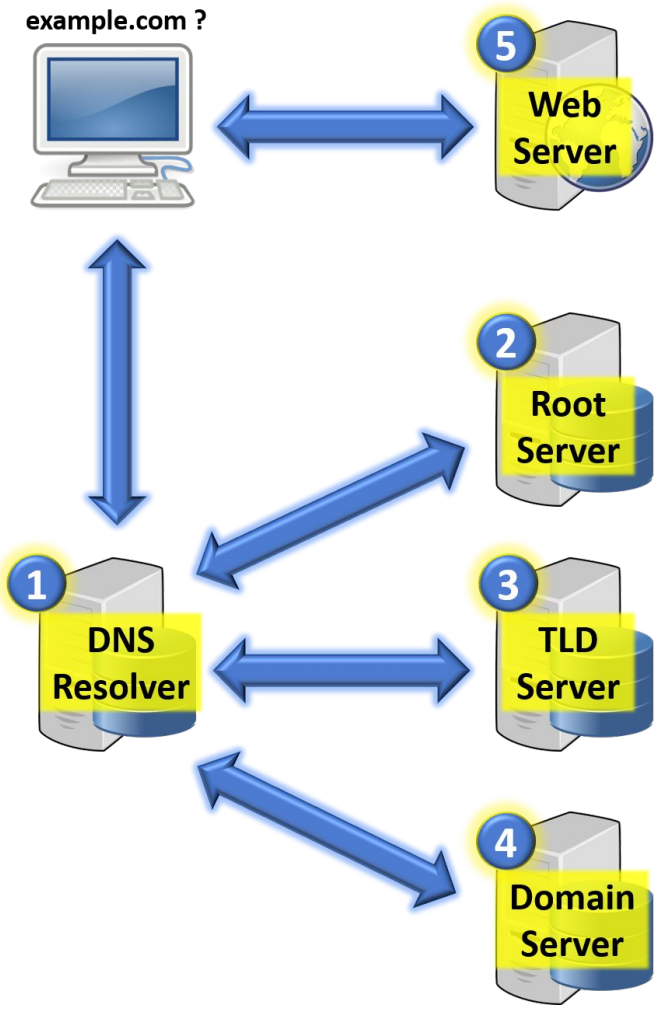
Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. Configuration & Setup
4. Actions & Header
5. Rule Options
6. HTTP
7. **DNS**
8. Regex & PCRE


It's not DNS
It's never the DNS
It was DNS...
I did not believe
Yet Alas! It was so true
Heed the damn haiku

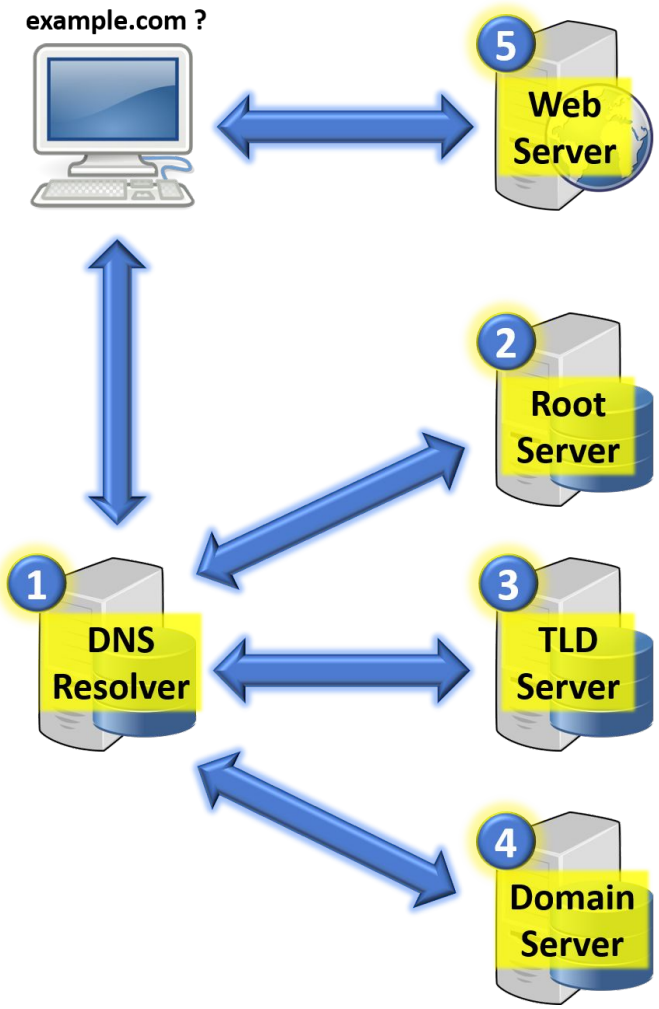
Domain Name System (DNS)

- Phonebook of the Internet
  - www.example.com == 192.168.1.1
- Four servers involved:
  - DNS recursor
  - Root nameserver
  - TLD nameserver
    - ".com", ".net", etc.
  - Authoritative nameserver
    - example.com
- Query & Reply
- UDP Port 53 (5353)
  - netBIOS?

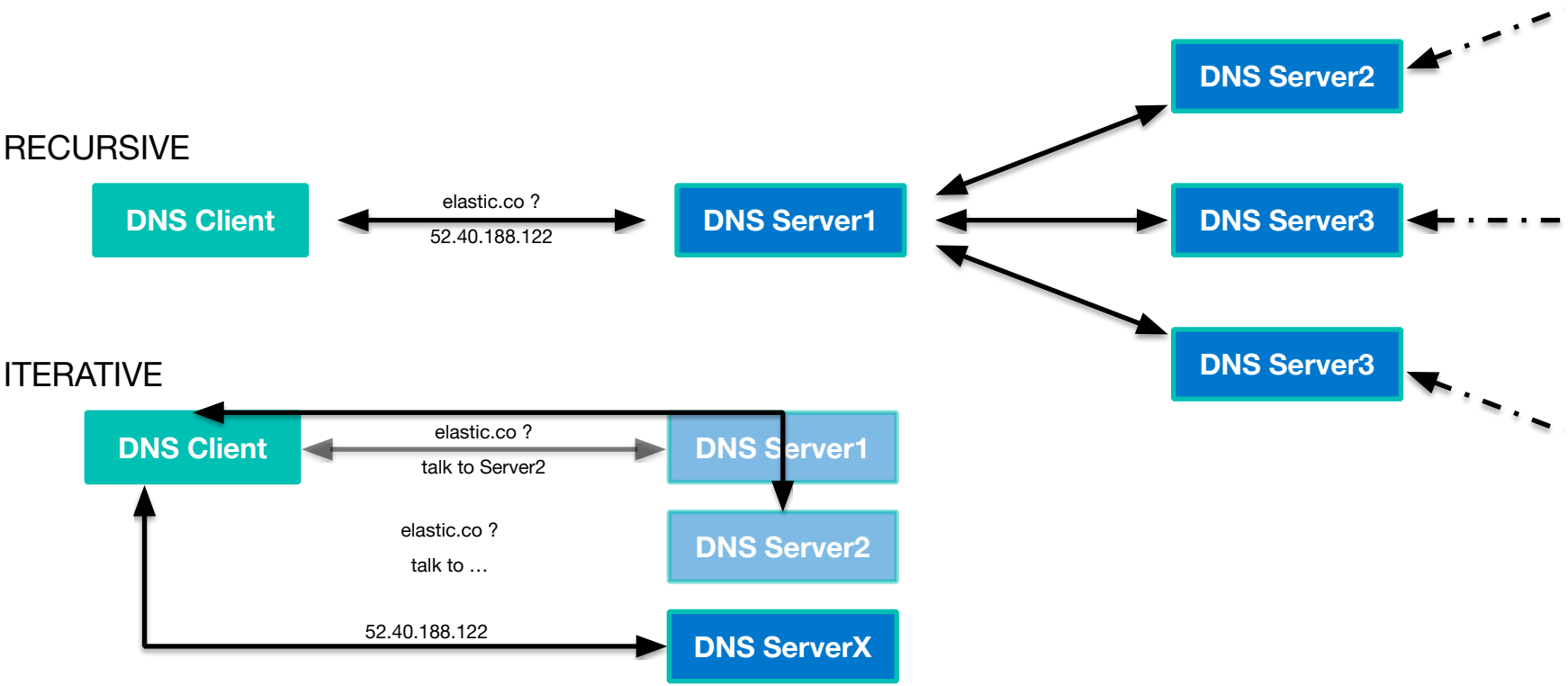


Domain Name System (DNS)

- Phonebook of the Internet
  - www.example.com == 192.168.1.1
- Four servers involved:
  - DNS recursor
  - Root nameserver
  - TLD nameserver
    - ".com", ".net", etc.
  - Authoritative nameserver
    - example.com
- Query & Reply
- UDP Port 53 (5353)
  - netBIOS?



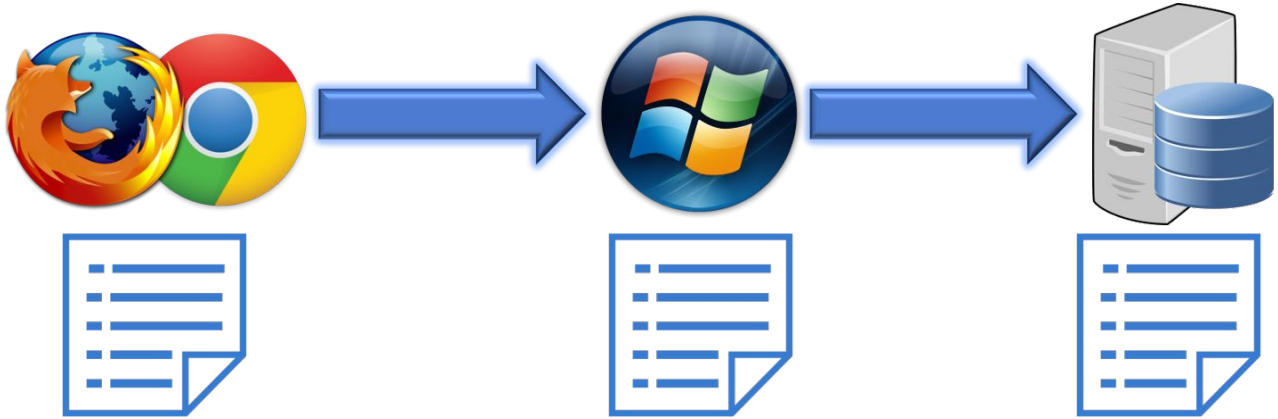
DNS IMPLEMENTATIONS



elastic

DNS Caching

- Browser DNS caching
  - usually for a shorter period of time
- OS level DNS caching
  - will check the local cache for any application request
  - this also pulls from the “hosts” file
- DNS Server caching



Resource Record (RR) Types

- A
  - host address
- AAAA
  - IPv6 host address
- NS
  - name server
- CNAME
  - canonical name for an alias
- SOA
  - start of authority

Attack Examples

- Zone Transfer (legitimate?)
  - Replicate DNS database across a set of DNS servers
  - TCP, client-server transaction
- Open Recursive DNS servers
  - allows anyone on the Internet to use the recursive DNS
- NXDOMAIN attack
  - tons of requests to non-existing domains
- DNS Poisoning / Spoofing
- DNS Tunneling

Suricata DNS Keyword

- dns.query: Provides a normalized string to match against
- DNS query on the wire (snippet)

```
|04|mail|06|google|03|com|00|
```

- dns.query sticky buffer contents: mail.google.com

Suricata DNS Keyword

- Example **without** prepending a “.” to content
  - `alert dns any any -> any any (dns.query; content: “read.com”; sid: 1)`

Payload	Matched
www.**read.com**	Yes
read.com	Yes
b**read.com**	Yes
www.lip**read.com**	Yes

Suricata DNS Keyword

- Example **with** prepending a “.” to content
  - `alert dns any any -> any any (dns.query; content: ".read.com"; sid: 1)`

Payload	Matched
www.**read.com**	Yes
www.lip**read.com**	No
b**read.com**	No
read.com	No

Transformations

- Keywords that modify data in a sticky buffer
- Examples
  - dotprefix; - interpets prepending “.” to enable accurate domain checking
  - strip_whitespace; - removes whitespace from the buffer
  - compress_whitespace; - compresses adjacent whitespace into a single space

Transformations, dotprefix option

- Example **with** dotprefix
  - `alert dns any any -> any any (dns.query; dotprefix; content: ".read.com"; sid: 1)`

Payload	Matched
www.**read.com**	Yes
www.lip**read.com**	No
b**read.com**	No
read.com	Yes

CTF: DNS Rules

Suricata

1. Traffic Capture Methods
2. Suricata Overview
3. Configuration & Setup
4. Actions & Header
5. Rule Options
6. HTTP
7. DNS
8. **Regex & PCRE**

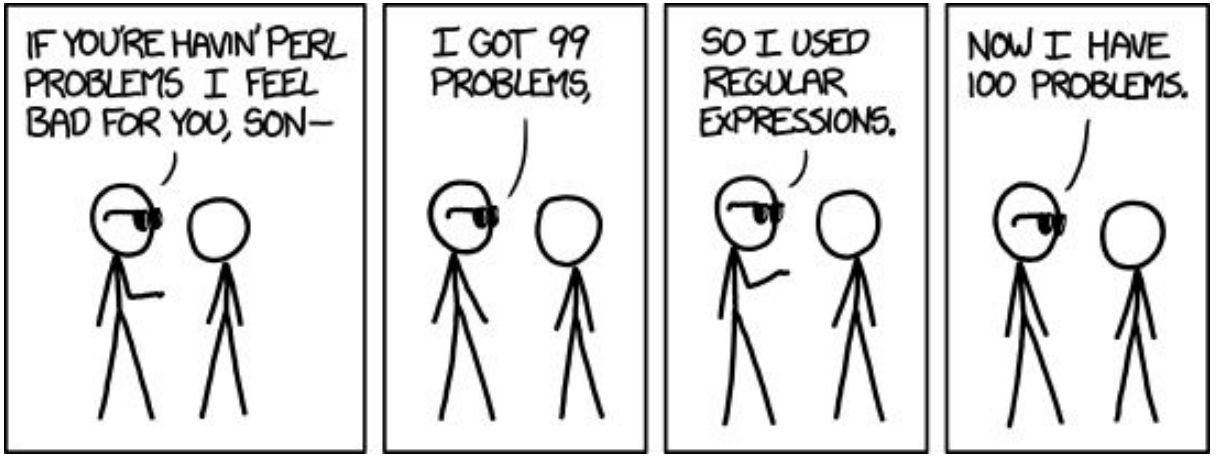
What is Regex?

- Text-based syntax to define patterns
- Used to find / find & replace text
- Resources:
  - https://www.debuggex.com
  - https://www.rexegg.com/regex-quickstart.html
  - https://regexr.com
  - https://regex101.com
  - https://regexone.com

Regex Basics

- Can anyone explain what is happening here?

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



Regex - Basic Character Matching

- Alphanumeric characters behave as expected, standard string match
- Regex is **case sensitive**

- **Regex: o**
- Looks for strings with at least one ‘o’ in them

Payload	Match
book	Yes
Booth	Yes
DOG	No

- **Regex: t711**
- Looks for strings containing ‘t711’

Payload	Match
about7116	Yes
Boot7113	Yes
dog2513	No

Regex - Collections

- [xyz] Brackets allow you to specify multiple characters for matching
- [x-z] Specify ranges using dashes
- Capitalization matters

- **Regex: [db]**
- ‘d’ or ‘b’

Payload	Match
dog	Yes
book	Yes
Booth	No

- **Regex: [A-Z]oo**
- Contains character between capital ‘A’ and ‘Z’ followed by ‘oo’

Payload	Match
Book	Yes
Zoo	Yes
root	No

Regex - Anchors

- ^ Used to specify the beginning of a line
- $ Used to specify the end of a line

- **Regex: ^bot**
- Start with the string ‘bot’

Payload	Match
bottle	Yes
bot	Yes
abbot	No

- **Regex: ^[db]ot$**
- Starts with a ‘b’ or ‘d’ followed by an ‘ot’ at the end of the string

Payload	Match
bot	Yes
dot	Yes
bottom	No

Regex - Quantifiers

- ? Zero or one times
- + One or more
- * Zero or more

- **Regex: ou?t**
- Looks for strings containing ‘out’ or ‘ot’
- ‘u’ can occur zero or one time

Payload	Match
Booth	Yes
about	Yes
dog	No

- **Regex: bo*a?t**
- ‘b’
- ‘o’ can occur zero or more times
- ‘a’ can occur zero or one times
- ‘t’

Payload	Match
boat	Yes
boot	Yes
baat	No

Regex - Quantifiers

- {x} Specifies number of times character repeats
- {x,y} Specifies range for the number of repeats

- **Regex: [a-c]{4}**
- Contains the characters a,b, or c 4

Payload	Match
abba	Yes
tbaba1028509	Yes
cactus	No

- **Regex: b[aeiou]{2,4}**
- Starts with ‘b’, followed by 2 to 4 lowercase vowels

Payload	Match
boot	Yes
beeoos	Yes
bits	No

Regex - Wildcard

. Counts as a wildcard (any character)

- **Regex: ^o.+t$**
- Line starts with ‘o’ followed by one or more of any character, ending with ‘t’

Payload	Match
oAt	Yes
o34@fgA-h24qet	Yes
oats	No

- **Regex: .***
- Looks for any character zero or more times
- Will match every string

Payload	Match
dot	Yes
boot	Yes
Booth	Yes

PCRE escaped characters

- These characters have special functions in regex.

```
.^$*+|?( )[ ]{ }\
```

- The user must “escape” this special function to refer to the literal string characters

```
.           wildcard
\.          literal period character

+           one or more of the previous character
\+          literal plus character

[           offers range of character options ex. [a-z]
\[          literal left bracket character
```

PCRE escaped characters

```
.          wildcard
\.         literal period
```

- **Regex: .com$**
- One or more lowercase letters followed by a wildcard, ending with ‘com’

Payload	Match
spacejam.com	Yes
broadcom	Yes
snort-org	No

- **Regex: \.com$**
- One or more lowercase letters ending with ‘.com’

Payload	Match
spacejam.com	Yes
broadcom	No
snort-org	No

Regex Basics

- ~~Can anyone~~ Everyone can explain what is happening here!

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

^	Start of Line!
[a-zA-Z0-9._]	Any letter (lower or UPPER) number, period or underscore
+	One or more
@	Literal “@” character
\.	Literal “.” character
[a-zA-z]	Any letter (lower or UPPER)
{2,3}	2 or 3 times
$	End of Line!

CTF: Regex

PCRE

- Academic Example
  - `pcre: "/<regex>/<flags>";`
- Expensive to run, can cause negative performance
- Typically combined with the content option

Flags	Description
i	case insensitive
s	. now matches line break characters
m	Will search patterns across multiple lines

PCRE

- Example
  - `pcre: "/^(w{3}\.)?google.*/";`

Payload	Match
www.google.com	Yes
googlestar.edu	Yes
fakegoogle.com	No

CTF: PCRE Rules

PCRE Rules - Review

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

Quiz

Suricata Quiz - Part 1

1. What purpose does an "Intrusion Detection System" like Suricata serve?
    a. Identify things that are normally missed by Analysts
    b. Identify unknown anomalies
    c. Identify known bad
    d. Utilizes signatures to identify known good

2. True or False: On release, a key differentiator Suricata had from Snort was its multi-threaded capabilities.

3. What three parts make up a Suricata signature?

Suricata Quiz - Part 2

4. What protocol options are available for inspection when referring to Suricata?

5. True or False: Meta-settings in a rule affect Suricata's inspection.

6. True or False: These rule options will match against the following traffic:

Rule:
```
(msg: "test"; content: "evil|3A|stuff"; sid: 1;)
```

Traffic:
```
EVIL:stuff
```

Remember: `x003A` is equivalent to a colon ":"

Suricata Quiz - Part 3

7. In the signature header, how can you describe 10.0.0.0/24 and exclude 10.0.0.11 and 10.0.0.12?

8. True or False: These rule options will match against the following traffic:

Rule:

```
(msg: “quiz”; http.uri; content: "/index";
http.uri; content: "html"; within:5; sid: 1;)
```

Traffic:

```
GET /index.html HTTP/1.0\r\n
```

Suricata Quiz - Part 4

9. True or False: These rule options will match against the following traffic:

Rule:
```
(msg: “quiz”; http.uri; content: “GET”; sid: 1;)
```

Traffic:
```
GET /index.html HTTP/1.0\r\n
```

10. True or False: Snort rules can be used in Suricata