So what cha want (?:to sig)?
in Suri 4.0

Travis Green, Francis Trudeau, Jack Mott, Jason Williams

A drinking team that sometimes writes IDS sigs
Gonna talk about this stuff

- 4.0 ET Fork
- ET Rule Writing Guidelines
- Xbits
- http_* buffers / tls_* buffers
- Prefilter
- Dns_query
- Base64
- SMTP
- Fun Stuff
- False Positive Communications
- Questions
ET 4.0 FORK!

---

Tuesday, Nov 21st

New keywords! (tks devs)

Current set 4.0 compatible

Retroactive updating of rules
Writing sigs the ET way™

---

- Unsolicited guidance on rule writing the way we s/prefer/try/ (and works for us)
- Continuity throughout rules makes for happy analysts
- Disclaimer: we’re not always right :)

---
Rule Format

```
alert proto any any -> any any (msg; flow; content; pcre; reference; classtype; sid; rev; metadata;)
```

- This is our preferred format for structuring rules
Content things

- Order the buffers and location modifiers strategically:

```content:"SuriCon"; <buffer>; <content modifier>; <fast_pattern>; <nocase>; <isdataat>;
```
Rule msg things

---

- msg:"<RULESET>  <CATEGORY>  <malware/product/thing>  <verb>  
  <date>";
  - If using a date, we like ISO-8601 format: YYYY-MM-DD

msg:"ET CURRENT_EVENTS Successful BankAustria Phish 2017-11-04";

- Include file type/architecture in name
  - E.g. Win32/Locky
- Defang urls in msg
  - E.g. msg:"Observed Malicious DNS Query to CnC (evil .com)";
- Define several types of checkins-- use Method[0-9]
  - E.g. msg:"Win32/Locky CnC Checkin M1";
fast_pattern; things

- Not necessary for a rule with one content
- Don’t use fast_pattern:only;
- No need to use fast_pattern; “chop”
  - E.g. content:”User-Agent: Mozilla/4.0 SuperEvilMaliciousness”; fast_pattern:25,20;
Metadata

- ET/PRO Ruleset uses metadata to include contextual information
- This is a freeform field which can contain anything
- Ideas could be:
  - Analyst name
  - Creation date
  - Modification date
  - Category

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO TROJAN Chthonic CnC Beacon 10"; flow:established,to_server; content:"POST"; http_method; urilen:3; content:"/p/"; http_uri; fast_pattern; http_header_names; content:"User-Agent"; content:"Accept"; content:"Referer"; pcre:"\^Host[^\r\n]+\r\nContent-Length\x3a\x20\d\{3\}\r\n\$/i"; metadata: former category TROJAN; reference:md5,b2cb9d1ecf41982f3bc54e03e516d7cd; classtype:trojan-activity; sid:2828440; rev:1; metadata:affected_product Windows_XP_Vista_7_8_10_Server_32_64_Bit, attack_target Client_Endpoint, deployment Perimeter, signature_severity Major, created_at 2017_10_26, malware_family Chthonic, performance_impact Low, updated_at 2017_10_26;)
Working with hex

- Use hex for common characters
- Use lowercase for hex
- Use hex for spaces
  - `content:”something\20\Evil\20\ya”`;
- Spaces between hex characters
  - `content:”\c0 \ff \ee”`

<table>
<thead>
<tr>
<th>Hex</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td>3a</td>
</tr>
<tr>
<td>;</td>
<td>3b</td>
</tr>
<tr>
<td>[space]</td>
<td>20</td>
</tr>
<tr>
<td>“</td>
<td>22</td>
</tr>
<tr>
<td>‘</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>7c</td>
</tr>
<tr>
<td>\</td>
<td>5c</td>
</tr>
</tbody>
</table>
Working with PCRE

- Easy to end up in PCRE jail
- Anchor, anchor, anchor!
- Use non-capture groups! -> (?:something|else)
- Stick PCREs last
- PCRE comes after content
  - content:"Evil"; nocase; pcre:"/^\s*something\s*else/Rsi";
- Use named capture groups, not back references
  - (?P<var>[\w-]{3})
- No need to use flags with sticky buffers :D
  - http_content_len; pcre:"/^3\d{2}$/R";
Xbits

- We’ve been able to make use of flowbits for a long time
- We’ll soon be able to make use of xbits for triggering alerts on host/ip pair.
- Similar to flowbit notation
- Great if you’re running in a mode that doesn’t utilize workers. (for now)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT_EVENTS Office 365 Phishing Landing 2017-10-10"; flow:established,to_client; file_data; content:"<title>Sign in to your account"; nocase; content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

---

Xbits
POST /var/
sessID=59f9421a0935d_3634-7eb7eabbe9bd03c2fc99881d04da9cbd-33213b5dca501ee1e6d8cd7b905f4e1bf723/offic...
Xbits

---

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT_EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established,to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established,to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT_EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established,tot_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established,to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT_EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT_EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT_EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ETPRO CURRENT EVENTS Office 365 Phishing Landing 2017-10-10 (set)"); flow:established, to_client;

xbits:set,o365.phish, track ip_src, expire 120; file_data;
content:"<title>Sign in to your account"; nocase;
content:"placeholder=|22|someone@example.com"; nocase; distance:0; content:"Office 365_EN_v1.0_AYB"; nocase; distance:0; fast_pattern; classtype:trojan-activity; sid:2828202; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO CURRENT EVENTS Successful Office 365 Phish"); flow:established, to_server; content:"POST"; http_method;
xbits:isset,o365.phish, track ip_dst;
classtype:trojan-activity; sid:2828202; rev:1;)
All the HTTP buffers

---

- We’ll go through examples of the new buffer usage
- Suri 1.3+ (yes we still wrote for this recently) vs. Suri 4.0
- Speed / Efficiency Gains
- Not parsing through http_header for every little thing
- Using less pcre (yay)
- Most new ones are sticky, as opposed to current ones that are modifiers
Things we won’t do anymore in 4.0 fork

---

content: "Content-Type|3a 20|text/html|0d 0a|"; http_header;

content: !"Referer|3a|"; http_header; content: !"Accept"; http_header;

pcre: "/^Host\x3a\x20(\d{1,3}\.){3}\d{1,3}\r\nContent-Length\x3a\d{3}\r\nUser-Agent\x3a [^\r\n]+?\r\n$/Hi";

content: ".php?cmd=login_submit"; http_header; (this is referer)

content: "Content-Length|3a 20|"; http_header; content: "|0d 0a|"; distance:2; within:3; http_header;
http_accept_enc / http_connection / http_header_names

---

**Suricata 1.3+**

```
content:"GET"; http_method;
content:"/app-ui/Installer.html?has
h="; http_uri; fast_pattern;
pcre:"/^\[0-9\]{10}/UR";
content:"Accept-Encoding|3
a 20|gzip, deflate";
http_header;
content:"Connection|3a
20|Keep-Alive";
http_header;
content:"!"Referer|3a 20|";
http_header;
```

**Suricata 4.0+**

```
content:"GET"; http_method;
content:"/app-ui/Installer.html?has
h="; http_uri; fast_pattern;
pcre:"/^\[0-9\]{10}/UR";

http_accept_enc;
content:"gzip, deflate";

http_connection;
content:"Keep-Alive";

http_header_names;
content:"!"Referer";
```
http_header_names / http_referer

---

Suricata 1.3+

```
pcre:"/^\nConnection\x3a[^\r\n]+\r\nContent-Type\x3a[^\r\n]+\r\nAccept\x3a[^\r\n]+\r\nAccept-Encoding\x3a[^\r\n]+\r\nAccept-Language\x3a[^\r\n]+\r\nReferer\x3a[^\r\n]+\.php?id=[^\r\n]*\r\nUser-Agent\x3a[^\r\n]+(?:\sMSIE|\rv\x3a11)\x27/Hi";
```

Suricata 4.0+

```
pcre"/(?:\x20MSIE|\rv\x3a11)/Vi";

http_header_names;
content:"|0d 0a 0d 0a|Connection|0d 0a|Content-Type|0d 0a|Accept|0d 0a|Accept-Encoding|0d 0a|Accept-Language|0d 0a|Referer|0d 0a|User-Agent|0d 0a|"; nocase;

http_referer; content:"\php?id="; nocase;
```
OR http_protocol

---

**Suricata 1.3+**

```plaintext
urilen:1; content:"POST"; http_method;
content:"POST / HTTP/1.1|0d
0a|Host|3a|"; fast_pattern:only;
content:"!"User-Agent|3a|";
http_header; content:"!"Accept";
http_header; content:"!"Referer|3a|";
http_header;
content:"!"Content-Type|3a|";
http_header;
```

---

**Suricata 4.0+**

```plaintext
urilen:1; content:"POST"; http_method;

http_protocol; content:"HTTP/1.1";

http_header_names; content:"|0d 0a 0d
0a|Host"; fast_pattern;
content:"Accept"; content:"!"Referer";
content:"!"User-Agent";
content:"!"Content-Type";
```
http_response_line

Suricata 1.3+

```
content:"HTTP/1.1 200 OK\r\nServer:nginx"
```

Suricata 4.0+

```
pcre:"/^\r\nDate\r\nContent-Type: text/html\r\nContent-Length\r\nConnection: keep-alive$/Hmi";
```

```
http_response_line; content:"HTTP/1.1 200 OK"
```

```
http_connection; content:"keep-alive"
```

```
http_content_type; content:"text/html"
```

```
http_header_names; content:"Server 0d 0a 0d 0a|Date 0d 0a|Content-Type 0d 0a|Content-Length 0d 0a|Connection 0d 0a|
```
OR http_start w/ http_content_len byte_test

---

**Suricata 1.3+**

```
content:"HTTP/1.1 200 OK|0d 0a|Server|3a 20|nginx";

Content-Length\x3a\x20(?:[2-9]\d{4}|1\d{5})

Above 20000 and below 199999
```

---

**Suricata 4.0+**

```
http_start; content:"HTTP/1.1 200 OK|0d 0a|Server|3a 20|nginx";
depth:29;

http_content_len;
byte_test:0,<=,199999,0,string,dec;
byte_test:0,>=,20000,0,string,dec;
```
http_accept / http_accept_lang

---

Suricata 1.3+

http_accept / http_accept_lang

---

Suricata 4.0+

http_accept / http_accept_lang

---

Suricata 4.0+

http_accept / http_accept_lang

---
All the SSL/TLS buffers

---

- Similar to the http buffer slides, let's see how things are changing with the move to 4.0
- We have cert based buffers where we had none before
- Less raw content matches is better (for rule writers)
- Most are sticky buffers
alert tls $EXTERNAL_NET any -> $HOME_NET any
(msg:"ET TROJAN Send-Safe Bulk Mailer SSL Cert";
flow:established,from_server;
content:"|55 04 06|"; content:"|07|Unknown";
distance:1; within:8; content:"|55 04 08|";
distance:0; content:"|07|Unknown";
distance:1; within:8; content:"|55 04 07|";
distance:0; content:"|07|Unknown";
distance:1; within:8; content:"|55 04 0a|";
distance:0; content:"|09|Send-Safe";
distance:1; within:10; content:"|55 04 0b|";
distance:0; content:"|07|Unknown";
distance:1; within:8; content:"|55 04 03|";
distance:0; content:"|09|Send-Safe";
distance:1; within:10; fast_pattern;
class:type:trojan-activity; sid:3434; rev:1;)

Suricata 1.3+

Suricata 4.0+
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET TROJAN ABUSE.CH SSL Fingerprint Blacklist Malicious SSL Certificate Detected (Cobalt CnC)"; flow:established,from_server; content:"|55 04 03|"; content:"|13|webmail.gr3atest.kz"; distance:1; within:20; fast_pattern; content:"Let's Encrypt"; distance:0; reference:url,sslbl.abuse.ch; classtype:trojan-activity; sid:1001; rev:1;)

Subject Common Name: CN=webmail.gr3atest.kz
Subject: CN=webmail.gr3atest.kz
Issuer Common Name: Let's Encrypt Authority X3
Issuer: C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
Fingerprint (SHA1): 127c20e7ad6a12fc275ee12ecfa290902a597936
Status: Blacklisted (Reason: Cobalt CnC, Listing date: 2022-03-30)

Suricata 1.3+
https://sslbl.abuse.ch/
### SSL Certificate Information

<table>
<thead>
<tr>
<th>Subject Common Name:</th>
<th>CN=webmail.gr3atest.kz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject:</td>
<td>CN=webmail.gr3atest.kz</td>
</tr>
<tr>
<td>Issuer Common Name:</td>
<td>Let's Encrypt Authority X3</td>
</tr>
<tr>
<td>Issuer:</td>
<td>C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3</td>
</tr>
<tr>
<td>Fingerprint (SHA1):</td>
<td>127c20e7ad6a12fc275ee12ecfa290902a597936</td>
</tr>
<tr>
<td>Status:</td>
<td>Blacklisted (Reason: Cobalt C&amp;C, Listing date: 2023-01-20)</td>
</tr>
</tbody>
</table>

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET TROJAN ABUSE.CH SSL Fingerprint Blacklist Malicious SSL Certificate Detected (Cobalt CnC)"; flow:established,from_server;
tls_cert_subject; content:"CN=webmail.gr3atest.kz";
isdataat:!1,relative; tls_certIssuer; content:"CN=Let's Encrypt"; reference:url,sslbl.abuse.ch;
classType:trojan-activity; sid:1002; rev:1;)

Suricata 4.0+
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET TROJAN Zeus OPENSSL Banker Malicious SSL Certificate Detected";
flow:established,from_server;
content:"|16|"; content:"|0b|"; within:8; content:"|09 00 a0 c9 ee 35 a4 1c 6f 74|"; distance:0;
content:"|55 04 06|"; distance:0;
content:"|02|AU"; distance:1;
within:3;
reference:md5,a8139f8c2547f11522c3d7d58b90c422; classtype:trojan-activity; sid:2023590; rev:2;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET TROJAN Zeus OPENSSL Banker Malicious SSL Certificate Detected";
flow:established,from_server;
tls_cert_serial;
content:"00:A0:C9:EE:35:A4:1C:6F:74";
tls_cert_subject; content:"C=AU";
reference:md5,a8139f8c2547f11522c3d7d58b90c422; classtype:trojan-activity; sid:2023590; rev:2;)

---

Suricata 1.3+

Suricata 4.0+


---

Suricata 1.3+

Suricata 4.0+
Prefilter keyword

- We can bring fast_pattern style traffic prefiltering to things other than content, things like flow, flags, dsize

```
alert tcp any any -> $HOME_NET 20000 (msg:"ETPRO SCADA DNP3 Cold Restart"; dsize:14<>26; byte_test:1,&,0x04,3; content:"|@d|"; offset:12; depth:1; classtype:protocol-command-decode; sid:2821683; rev:1;)
```

```
alert tcp any any -> $HOME_NET 20000 (msg:"ETPRO SCADA DNP3 Cold Restart"; dsize:14<>26; prefilter; byte_test:1,&,0x04,3; content:"|@d|"; offset:12; depth:1; classtype:protocol-command-decode; sid:28216831; rev:1;)
```
Prefilter example

---

```
# SCADA DNP3 Cold Restart
alert tcp any any -> $HOME_NET 20000 (msg:"ETPRO SCADA DNP3 Cold Restart"; dsize:14<>26; byte_test:1,&,0x04,3; content:"[0d]"; offset:12; depth:1; classtype:protocol-command-decode; sid:2821683; rev:1;)

alert tcp any any -> $HOME_NET 20000 (msg:"ETPRO SCADA DNP3 Cold Restart"; dsize:14<>26; prefilter; byte_test:1,&,0x04,3; content:"[0d]"; offset:12; depth:1; classtype:protocol-command-decode; sid:2821683; rev:1;)
```


<table>
<thead>
<tr>
<th>Num</th>
<th>Rule</th>
<th>Gid</th>
<th>Rev</th>
<th>Ticks</th>
<th>%</th>
<th>Checks</th>
<th>Matches</th>
<th>Max Ticks</th>
<th>Avg Ticks</th>
<th>Avg Match</th>
<th>Avg No Match</th>
</tr>
</thead>
</table>
Prefilter engine setting

---

- You can enable “auto” it in suricata.yaml

```yaml
prefilter:
  # default prefiltering setting. "mpm" only creates MPM/fast_pattern engines. "auto" also sets up prefilter engines for other keywords. # Use --list-keywords=all to see which keywords support prefiltering.
  default: auto
```

- Setting this to auto allows Suricata to decide at startup time (-vvv)
DNS sigs - background

- ~4300 ET rules are like - content:"|01 00 00 01 00 00 00 00 00 00 00|"; depth:10; offset:2;
  content:"|05|nceba|03|org|00|"; fast_pattern;
  ○ This has been working less well as dns resolvers get away from RFC defined flags
  ○ dns_query makes everything better
DNS resolvers misbehaving
Use `dns_query; buffer!`

- `dns_query;` is a fast_pattern-able, normalized, sticky_buffer (hooray!)
- `dns_query; content:"nceba.org"; isdataat:!1,relative;`
- There is a caveat regarding suri 4.0.0 - 4.0.1 - content matches + `dns_query;` in UDP ([fixed in 4.0.2](#))
- Is there a performance cost for this awesomeness?
alert udp $HOME_NET any -> any 53 (msg:"DNS Test - content matching some flags"; content:"01"; offset:2; depth:1; content:"00 01 00 00 00 00 00 00"; distance:1; within:7; content:"06|google|03|com|00"; distance:0; fast_pattern; classtype:misc-activity; sid:2; rev:1;)

alert dns $HOME_NET any -> any 53 (msg:"DNS Test - dns_query"; dns_query; content:"google.com"; isdataat:!1,relative; classtype:misc-activity; sid:3; rev:1;)


<table>
<thead>
<tr>
<th>Num</th>
<th>Rule</th>
<th>Gid</th>
<th>Rev</th>
<th>Ticks</th>
<th>%</th>
<th>Checks</th>
<th>Matches</th>
<th>Max Ticks</th>
<th>Avg Ticks</th>
<th>Avg Match</th>
<th>Avg No Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14050980</td>
<td>65.64</td>
<td>2675</td>
<td>1291</td>
<td>64646</td>
<td>5252.70</td>
<td>10116.34</td>
<td>715.89</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>7356698</td>
<td>34.36</td>
<td>1293</td>
<td>1293</td>
<td>61262</td>
<td>5689.63</td>
<td>5689.63</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Base64 Signatures

- Base64 creates a unique case that requires tricks on older Suricata (more on next slide)
- Available beginning in 3.0rc1
Base64 Signatures

Base64 on the wire can look three ways depending on surrounding data:
Base64 Old and New

Older Suricata used three signatures for one content:

```plaintext
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ET CURRENT_EVENTS Possible Internet Explorer CVE-2014-6332 Common Construct b64 1 (Observed in Archie EK)"; flow:established,from_server; file_data; content:"Y2hydygwMSknY2hydygwMTc2KS2jaH3KDAxKSZjaH3KDAwK"; reference:cve,2014-6332; classtype:attempted-user; sid:2819773; rev:2;)
```

```plaintext
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ET CURRENT_EVENTS Possible Internet Explorer CVE-2014-6332 Common Construct b64 2 (Observed in Archie EK)"; flow:established,from_server; file_data; content:"NocncoPDEpJnNocncoMjE3Nikny2hydygwMSkmY2hydygwMC"; reference:cve,2014-6332; classtype:attempted-user; sid:2819774; rev:3;)
```

```plaintext
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ET CURRENT_EVENTS Possible Internet Explorer CVE-2014-6332 Common Construct b64 3 (Observed in Archie EK)"; flow:established,from_server; file_data; content:"jaH3KDAxKSZjaH3KDAxKSZjaH3KDAwK"; reference:cve,2014-6332; classtype:attempted-user; sid:2819775; rev:2;)
```

Beginning with Suricata 3.0RC1 we can use one content:

```plaintext
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"B64 TEST"; flow:established,from_server; file_data; content:"overlord>"; base64_decode:relative; base64_data; content:"chrw(01)&chrw(2176)&chrw(01)&chrw(00)"; classtype:trojan-activity; sid:303333; rev:1;)
```
Base64 Basic Auth Example

---

dolemite@patpoopy:~$ echo -n Y3ZjemN2Y0B5YW5kZXgucnVfdjo3Nzc= | base64 -d & & echo
cvczcv@yandex.ru_v:777

|alert http $HOME_NET any $EXTERNAL_NET any (msg:"B64 TEST 2"; flow:established,to_server; content:"Authorization:Basic", msg:"yandex.ru"); http_header; base64_decode: relative; base64_data; content:"yandex.ru"; classtype:trojan-activity; sid:3030303; rev:1;|

08/13/2017-12:20:49.750001 [**] [1:3030303:1] B64 TEST 2 [**] [Classification: A Network Trojan was detected] [Priority: 1] {TCP} 172.16.35.19:1053 -> 172.16.1.1:8332
SMTP Signatures

- Uses 'smtp' keyword
- file_data applies to SMTP attached data
- Must be enabled in Suricata yamal app-layer section:
  - smtp:
    - enabled: yes
    - Mime:
      - decode-mime: yes
      - decode-base64: yes
- Available starting in 2.1beta4
- We never wrote signatures for data in attachments, but we plan on it now.
SMTP Signature Example

alert smtp any any -> any any (msg:"SMTP TEST"; flow:established; file_data; content:"Robust"; classtype:trojan-activity; sid:303011; rev:1;)

Follow TCP Stream (tcp.stream eq 0)

Stream Content

Content-Type: application/x-as400attachment;
name="/tmp/smtp.txt"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="/tmp/smtp.txt"

Um9idXNOcg==

Fun Crap

Useful Oneliner scripts:

Orders the rules by SID:

$ cat etpro.rules | grep -Po 'sid:\d+;' | sort -n | fgrep -f - etpro.rules

Eliminates metadata fields for easier reading (in place):

$ perl -pi -e 's/ metadata[^;].+?\;//g' /path/to/etpro.rules
Some Color:

```bash
$ alias rulesgrep="cat ~/testids/etpro.rules | GREP_COLOR='01;33' egrep --color=always '^alert' | GREP_COLOR='01;36' grep -P --color=always 'sid:\[^;]+' | GREP_COLOR='01;37' grep -P --color=always 'msg:".+?"' | GREP_COLOR='01;32' grep -P --color=always ' [a-zA-Z]+?:' | GREP_COLOR='01;31' egrep --color=always"
```

```bash
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ETPRO TROJAN Win32/Patpoopy CnC Beacon"; flow:established,to_server; content:"GET/20/index.php?d="; depth:17; fast_pattern; pcre:"/^(?:[A-Za-z0-9+\-/]{2})\[\[(?:[A-Za-z0-9+\-/]{3}\[\[(?:[A-Za-z0-9+\-/]{4})\]x20/R"; content:"HTTP/1."; within:7; content:"[0d 0a|Referer|3a|"; distance:0; content:"[0d 0a>User-Agent|3a 20|Mozilla/5.0|20|(Windows NT 6.1)|20|AppleWebKit/537.36 (KHTML, like Gecko)|20|Chrome/41.0.2228.0|20|Safari/537.36"; distance:0; threshold: type limit, track by_src, count 1, seconds 120; reference:md5,02ad367bb56ac5b2aa18df71056fb0e9; classtype:trojan-activity; sid:2826683; rev:1;)
```
WE DON'T KNOW WHAT WE DON'T KNOW

Tell us when you think you are seeing a rule False Positive

Can’t share full PCAP? No problem.

Mailing list:

https://lists.emergingthreats.net/mailman/listinfo/emerging-sigs

#emerging-threats on irc. freenode.net

@ET_Labs on Twitter

support@emergingthreats.net
Come learn

OISF offers a 2-day Suricata rule writing course

2018 dates coming soon:

- [http://oisf.net/training-events/](http://oisf.net/training-events/)
- info@oisf.net
QA / TKS!
@ET_Labs

Travis Green
@travisbgreen

Francis Trudeau
ftrudeau@emergingthreats.net

Jack Mott
@malwareforme

Jason Williams
@switchingtoguns

---