A day in the life doing incident response without Bro. And how it could be so much better!

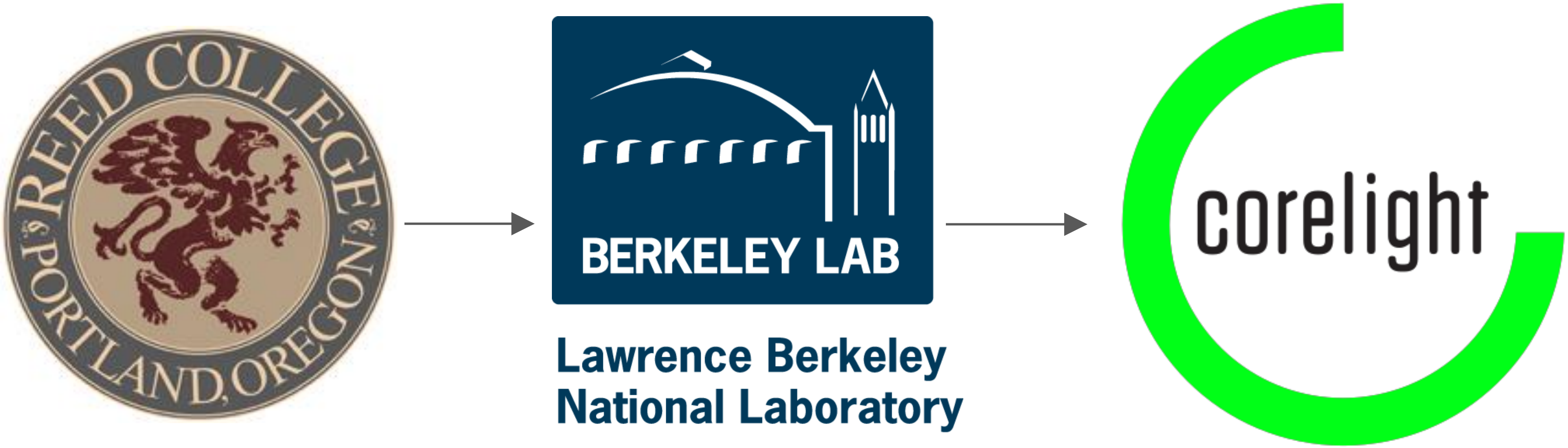
Vincent Stoffer - Corelight
Matt Bromiley – SANS

November 14, 2017

Outline

- Intro, backgrounds
- My life before Bro
- Discovering Bro
- Life after Bro
- Corelight
- Questions

Vince background



GIAC CERTIFIED INTRUSION ANALYST
GCIA
GIAC CERTIFIED INCIDENT HANDLER
GCIH
GIAC
ADVISORY BOARD
CISSP®

Matt Bromiley

- Incident responder/forensicator
  - Disk, network, memory forensics
  - A little bit of malware
- SANS instructor
  - FOR508
  - FOR572
- A lover of making network analysis easier - aka Bro!

Life before Bro

Control
Identify
Analyze
Plan
Track
ARGUS

IDS Alerts

Netflow

Full packet

syslog / other logs

ARGUS
Control
Identify
Analyze
Plan
Track

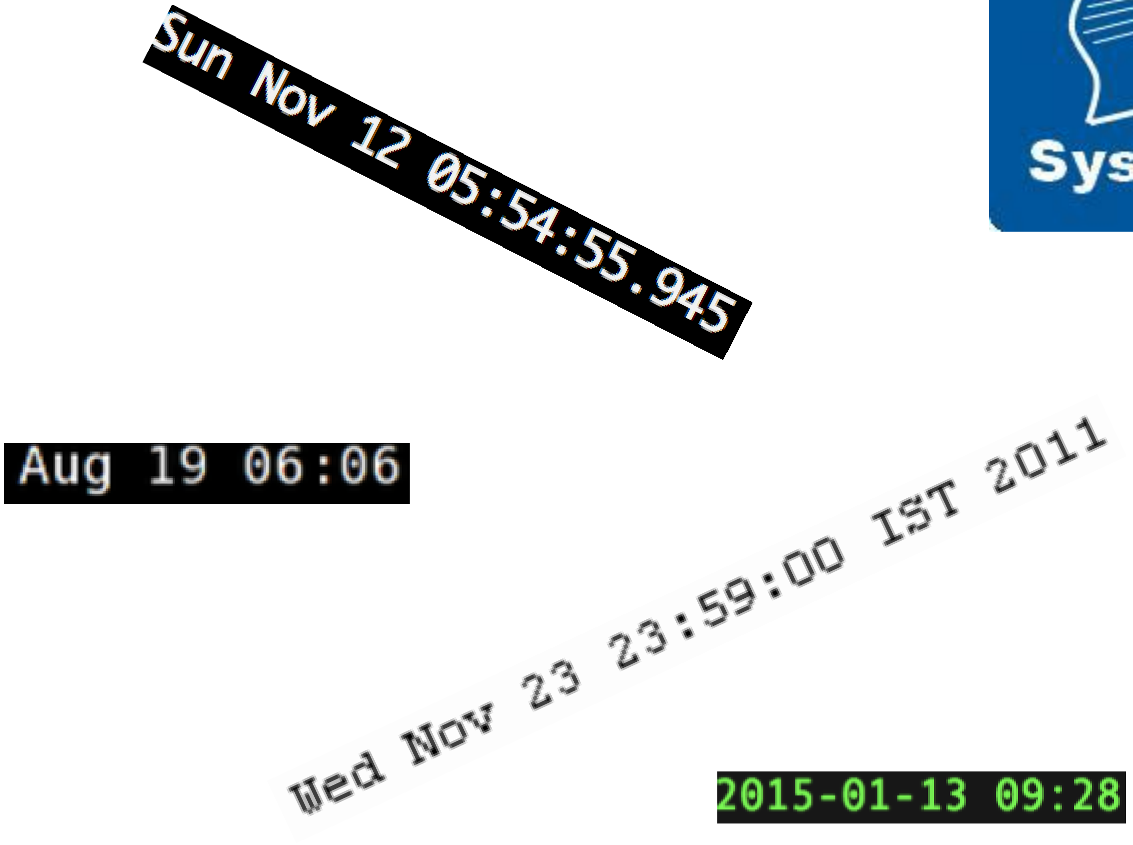
- Software flow generator, netflow replacement
- Easy to deploy
- Suite of command line tools for analysis and graphing
- High level metadata (no protocol analysis)
- Search for bad IPs, check connection details, etc.

- Great for matching packet signatures
- LOTS of tuning
- After a year or so I had a good system running
- Alerts + PCAP snippets

But that left me in the dark a lot!

Other logs

- DHCP
- DNS
- SMTP
- AAA/LDAP
- RADIUS
- etc.

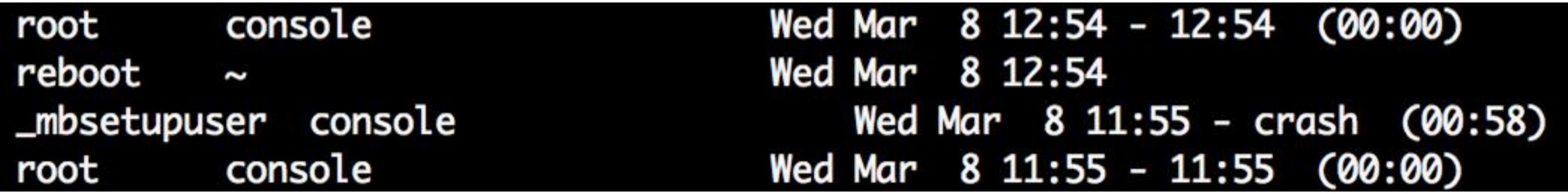


Sun Nov 12 05:54:55.945

Aug 19 06:06

Wed Nov 23 23:59:00 IST 2011

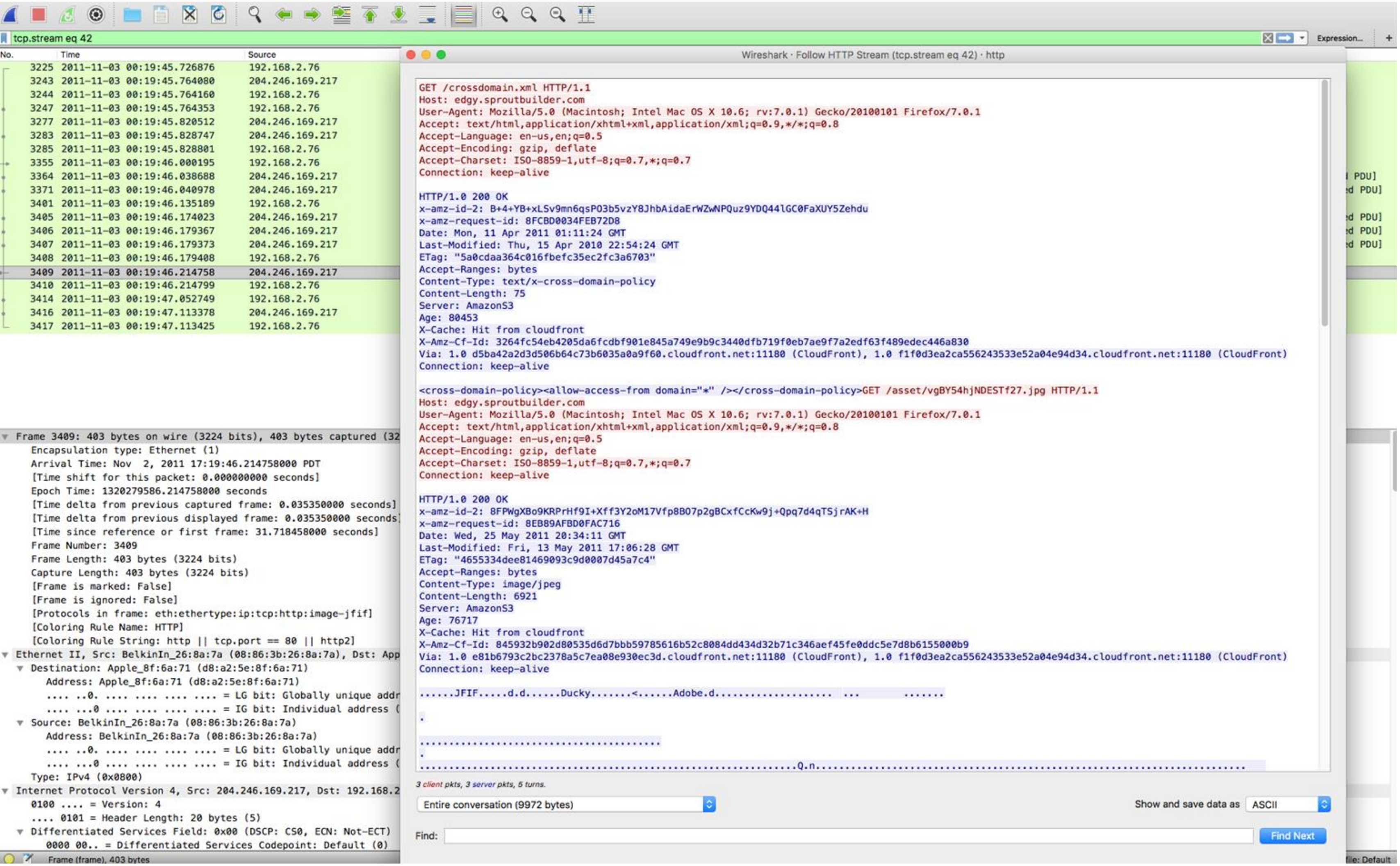
2015-01-13 09:28



```
root          console                      Wed Mar  8 12:54 - 12:54  (00:00)
reboot        ~                            Wed Mar  8 12:54
_mbsetupuser  console                      Wed Mar  8 11:55 - crash  (00:58)
root          console                      Wed Mar  8 11:55 - 11:55  (00:00)
```

Copyright 2017, Corelight

- Possible privacy concerns
- Storage problems
- How to do large-scale search and analysis?



```
No.   Time                          Source
3225  2011-11-03 00:19:45.726876    192.168.2.76
3243  2011-11-03 00:19:45.764080    204.246.169.217
3244  2011-11-03 00:19:45.764160    192.168.2.76
3247  2011-11-03 00:19:45.764353    192.168.2.76
3277  2011-11-03 00:19:45.820512    204.246.169.217
3283  2011-11-03 00:19:45.828747    204.246.169.217
3285  2011-11-03 00:19:45.828801    192.168.2.76
3355  2011-11-03 00:19:46.000195    192.168.2.76
3364  2011-11-03 00:19:46.038688    204.246.169.217
3371  2011-11-03 00:19:46.040978    204.246.169.217
3401  2011-11-03 00:19:46.135189    192.168.2.76
3405  2011-11-03 00:19:46.174023    204.246.169.217
3406  2011-11-03 00:19:46.179367    204.246.169.217
3407  2011-11-03 00:19:46.179373    204.246.169.217
3408  2011-11-03 00:19:46.179408    192.168.2.76
3409  2011-11-03 00:19:46.214758    204.246.169.217
3410  2011-11-03 00:19:46.214799    192.168.2.76
3414  2011-11-03 00:19:47.052749    192.168.2.76
3416  2011-11-03 00:19:47.113378    204.246.169.217
3417  2011-11-03 00:19:47.113425    192.168.2.76
```

```
▼ Frame 3409: 403 bytes on wire (3224 bits), 403 bytes captured (32
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov  2, 2011 17:19:46.214758000 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1320279586.214758000 seconds
    [Time delta from previous captured frame: 0.035350000 seconds]
    [Time delta from previous displayed frame: 0.035350000 seconds]
    [Time since reference or first frame: 31.718458000 seconds]
    Frame Number: 3409
    Frame Length: 403 bytes (3224 bits)
    Capture Length: 403 bytes (3224 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:image-jfif]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
▼ Ethernet II, Src: BelkinIn_26:8a:7a (08:86:3b:26:8a:7a), Dst: App
  ▼ Destination: Apple_8f:6a:71 (d8:a2:5e:8f:6a:71)
      Address: Apple_8f:6a:71 (d8:a2:5e:8f:6a:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique addr
      .... ...0 .... .... .... .... = IG bit: Individual address (
  ▼ Source: BelkinIn_26:8a:7a (08:86:3b:26:8a:7a)
      Address: BelkinIn_26:8a:7a (08:86:3b:26:8a:7a)
      .... ..0. .... .... .... .... = LG bit: Globally unique addr
      .... ...0 .... .... .... .... = IG bit: Individual address (
    Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 204.246.169.217, Dst: 192.168.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
```

Frame (frame), 403 bytes

```
GET /crossdomain.xml HTTP/1.1
Host: edgy.sproutbuilder.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive

HTTP/1.0 200 OK
x-amz-id-2: B+4+YB+xLSv9mn6qsPO3b5vzY8JhbAidaErWZwNPQuz9YDQ44lGC0FaXUY5Zehdu
x-amz-request-id: 8FCBD0034FEB72D8
Date: Mon, 11 Apr 2011 01:11:24 GMT
Last-Modified: Thu, 15 Apr 2010 22:54:24 GMT
ETag: "5a0cdaa364c016fbefc35ec2fc3a6703"
Accept-Ranges: bytes
Content-Type: text/x-cross-domain-policy
Content-Length: 75
Server: AmazonS3
Age: 80453
X-Cache: Hit from cloudfront
X-Amz-Cf-Id: 3264fc54eb4205da6fcdbf901e845a749e9b9c3440dfb719f0eb7ae9f7a2edf63f489edec446a830
Via: 1.0 d5ba42a2d3d506b64c73b6035a0a9f60.cloudfront.net:11180 (CloudFront), 1.0 f1f0d3ea2ca556243533e52a04e94d34.cloudfront.net:11180 (CloudFront)
Connection: keep-alive

<cross-domain-policy><allow-access-from domain="*" /></cross-domain-policy>GET /asset/vgBY54hjNDESTf27.jpg HTTP/1.1
Host: edgy.sproutbuilder.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive

HTTP/1.0 200 OK
x-amz-id-2: 8FPWgXBo9KRPrHf9I+Xff3Y2oM17Vfp8BO7p2gBCxfCcKw9j+Qpq7d4qTSjrAK+H
x-amz-request-id: 8EB89AFBD0FAC716
Date: Wed, 25 May 2011 20:34:11 GMT
Last-Modified: Fri, 13 May 2011 17:06:28 GMT
ETag: "4655334dee81469093c9d0007d45a7c4"
Accept-Ranges: bytes
Content-Type: image/jpeg
Content-Length: 6921
Server: AmazonS3
Age: 76717
X-Cache: Hit from cloudfront
X-Amz-Cf-Id: 845932b902d80535d6d7bbb59785616b52c8084dd434d32b71c346aef45fe0ddc5e7d8b6155000b9
Via: 1.0 e81b6793c2bc2378a5c7ea08e930ec3d.cloudfront.net:11180 (CloudFront), 1.0 f1f0d3ea2ca556243533e52a04e94d34.cloudfront.net:11180 (CloudFront)
Connection: keep-alive

......JFIF.....d.d......Ducky.......<......Adobe.d....................  ...      .......

.

.......................................
.
..............................................................Q.n...................................................................
```

3 client pkts, 3 server pkts, 5 turns.

Entire conversation (9972 bytes)

Show and save data as: ASCII

Find:

Discovery!

Wait….what *is* Bro?

BRO

Open-source network monitoring project created more than 22 years ago

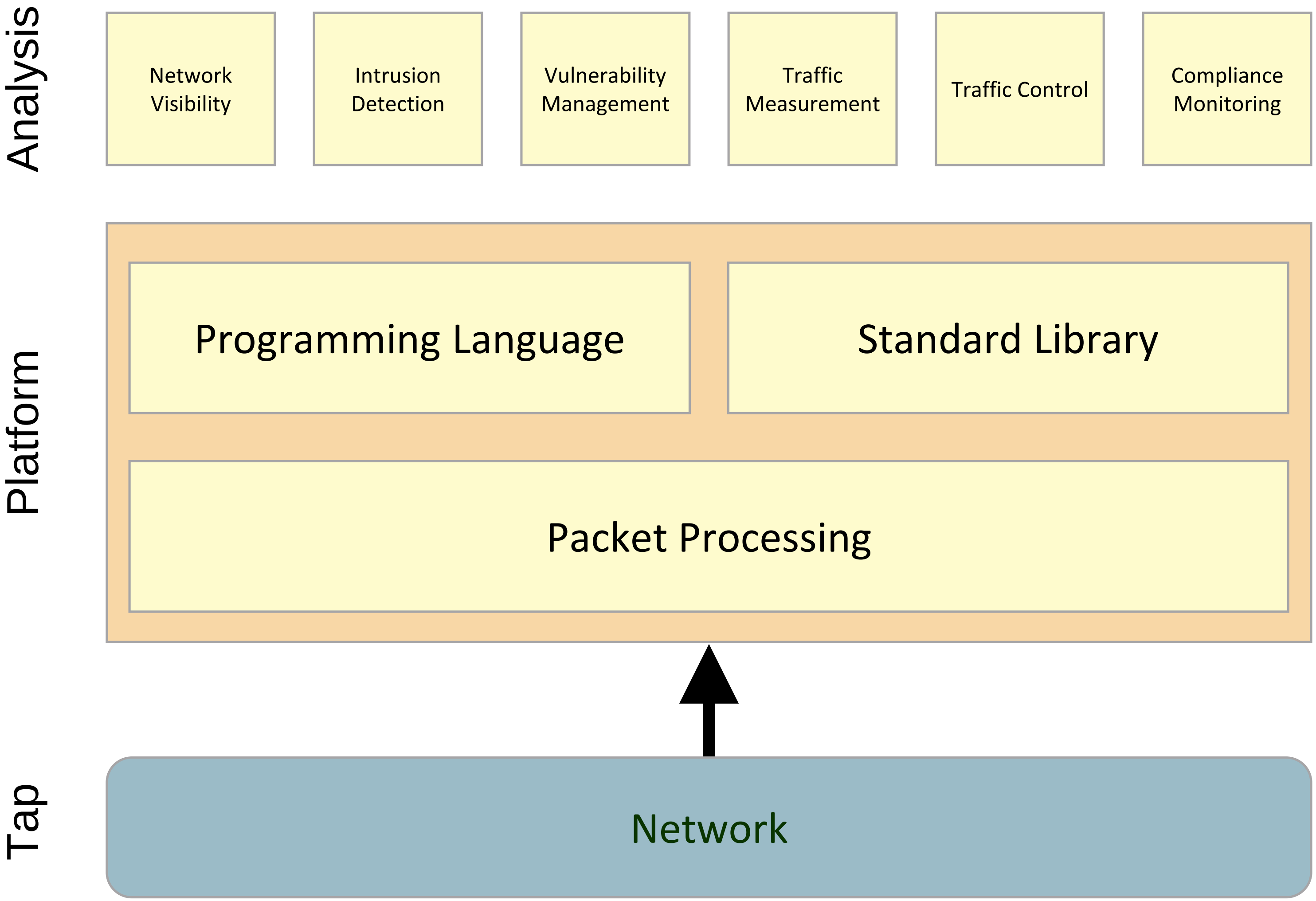
- A standalone network monitor
- A programmable framework
- A community of operators and users

bro.org

corelight

The Bro Platform

Open Source
BSD License



Analysis

Network Visibility

Intrusion Detection

Vulnerability Management

Traffic Measurement

Traffic Control

Compliance Monitoring

Platform

Programming Language

Standard Library

Packet Processing

Tap

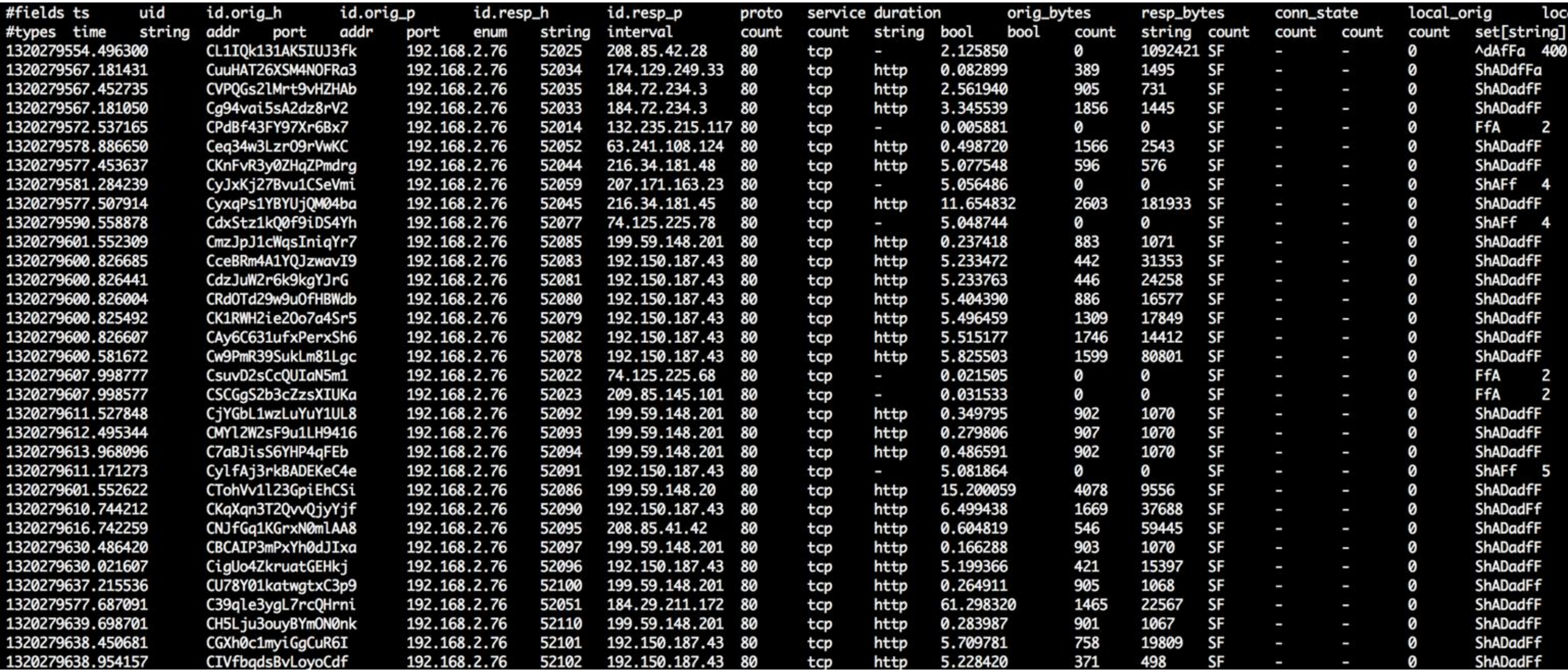
Network

Life after Bro

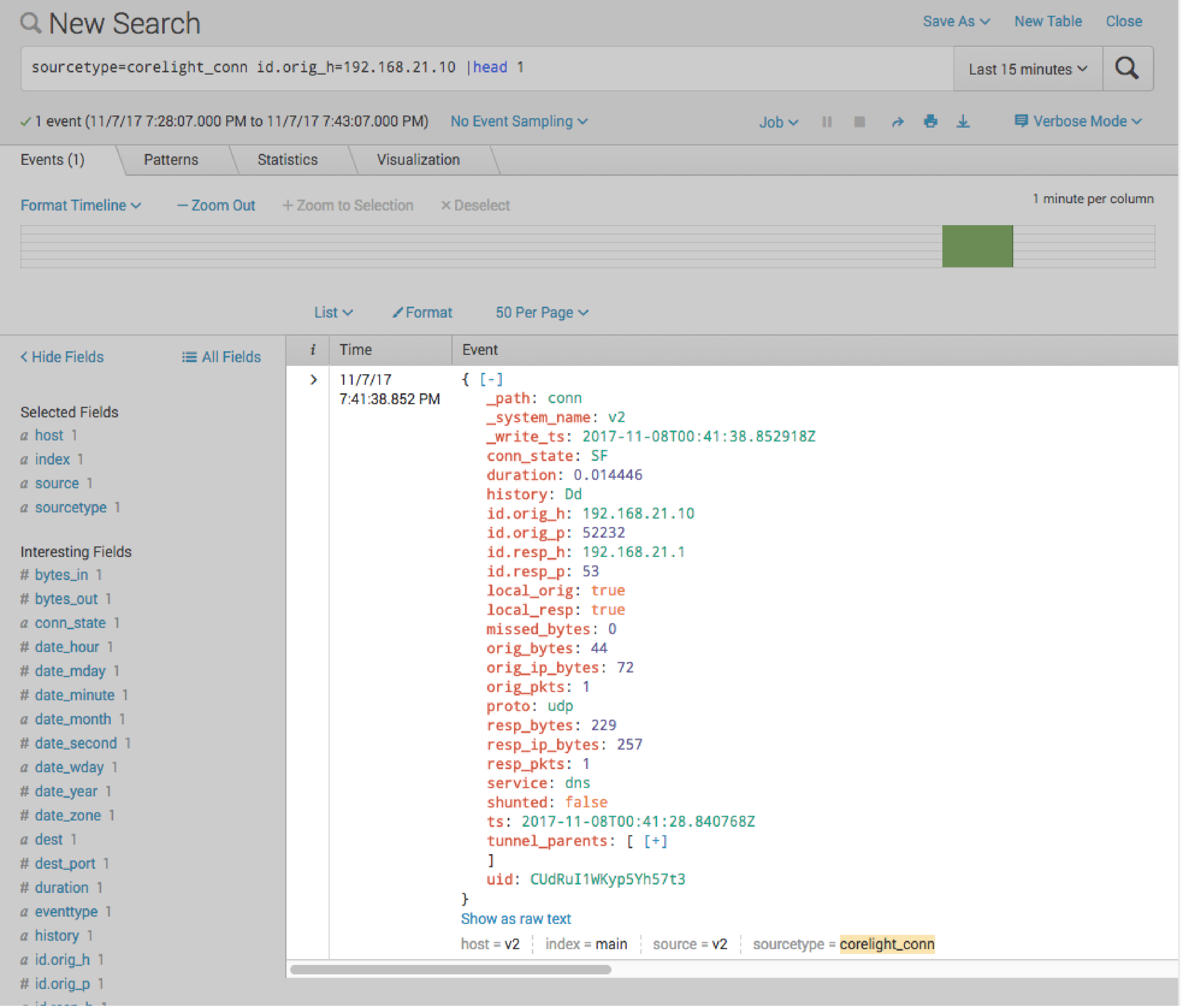
```
> bro -i eth0
[ … wait … ]
> ls *.log
app_stats.log                  ntlm.log
communication.log              rdp.log
conn.log                       reporter.log
dce_rpc.log                    signatures.log
dhcp.log                       smb_files.log
dns.log                        smb_mapping.log
dpd.log                        smtp.log
files.log                      socks.log
ftp.log                        software.log
http.log                       ssh.log
irc.log                        ssl.log
known_hosts.log                syslog.log
known_services.log             tunnel.log
modbus.log                     x509.log
notice.log                     weird.log
```

Bro’s Log Files

Rich, structured, protocol specific real-time activity streams that are policy neutral.



```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   service duration        orig_bytes      resp_bytes      conn_state      local_orig      loc
#types  time    string  addr    port    addr    port    enum    string  interval        count   count   string  bool    bool    count   string  count   count   count   count   set[string]
1320279554.496300       CL1IQk131AK5IUJ3fk      192.168.2.76    52025   208.85.42.28    80      tcp     -       2.125850        0       1092421 SF      -       -       0       ^dAfFa  400
1320279567.181431       CuuHAT26XSM4NOFRa3      192.168.2.76    52034   174.129.249.33  80      tcp     http    0.082899        389     1495    SF      -       -       0       ShADdfFa
1320279567.452735       CVPQGs2lMrt9vHZHAb      192.168.2.76    52035   184.72.234.3    80      tcp     http    2.561940        905     731     SF      -       -       0       ShADadfF
1320279567.181050       Cg94vai5sA2dz8rV2       192.168.2.76    52033   184.72.234.3    80      tcp     http    3.345539        1856    1445    SF      -       -       0       ShADadfF
1320279572.537165       CPdBf43FY97Xr6Bx7       192.168.2.76    52014   132.235.215.117 80      tcp     -       0.005881        0       0       SF      -       -       0       FfA     2
1320279578.886650       Ceq34w3Lzr09rVwKC       192.168.2.76    52052   63.241.108.124  80      tcp     http    0.498720        1566    2543    SF      -       -       0       ShADadfF
1320279577.453637       CKnFvR3y0ZHqZPmdrg      192.168.2.76    52044   216.34.181.48   80      tcp     http    5.077548        596     576     SF      -       -       0       ShADadfF
1320279581.284239       CyJxKj27Bvu1CSeVmi      192.168.2.76    52059   207.171.163.23  80      tcp     -       5.056486        0       0       SF      -       -       0       ShAFf   4
1320279577.507914       CyxqPs1YBYUjQM04ba      192.168.2.76    52045   216.34.181.45   80      tcp     http    11.654832       2603    181933  SF      -       -       0       ShADadfF
1320279590.558878       CdxStz1kQ0f9iDS4Yh      192.168.2.76    52077   74.125.225.78   80      tcp     -       5.048744        0       0       SF      -       -       0       ShAFf   4
1320279601.552309       CmzJpJ1cWqsIniqYr7      192.168.2.76    52085   199.59.148.201  80      tcp     http    0.237418        883     1071    SF      -       -       0       ShADadfF
1320279600.826685       CceBRm4A1YQJzwavI9      192.168.2.76    52083   192.150.187.43  80      tcp     http    5.233472        442     31353   SF      -       -       0       ShADadfF
1320279600.826441       CdzJuW2r6k9kgYJrG       192.168.2.76    52081   192.150.187.43  80      tcp     http    5.233763        446     24258   SF      -       -       0       ShADadfF
1320279600.826004       CRdOTd29w9uOfHBWdb      192.168.2.76    52080   192.150.187.43  80      tcp     http    5.404390        886     16577   SF      -       -       0       ShADadfF
1320279600.825492       CK1RWH2ie2Oo7a4Sr5      192.168.2.76    52079   192.150.187.43  80      tcp     http    5.496459        1309    17849   SF      -       -       0       ShADadfF
1320279600.826607       CAy6C631ufxPerxSh6      192.168.2.76    52082   192.150.187.43  80      tcp     http    5.515177        1746    14412   SF      -       -       0       ShADadfF
1320279600.581672       Cw9PmR39SukLm81Lgc      192.168.2.76    52078   192.150.187.43  80      tcp     http    5.825503        1599    80801   SF      -       -       0       ShADadfF
1320279607.998777       CsuvD2sCcQUIaN5m1       192.168.2.76    52022   74.125.225.68   80      tcp     -       0.021505        0       0       SF      -       -       0       FfA     2
1320279607.998577       CSCGgS2b3cZzsXIUKa      192.168.2.76    52023   209.85.145.101  80      tcp     -       0.031533        0       0       SF      -       -       0       FfA     2
1320279611.527848       CjYGbL1wzLuYuY1UL8      192.168.2.76    52092   199.59.148.201  80      tcp     http    0.349795        902     1070    SF      -       -       0       ShADadfF
1320279612.495344       CMYl2W2sF9u1LH9416      192.168.2.76    52093   199.59.148.201  80      tcp     http    0.279806        907     1070    SF      -       -       0       ShADadfF
1320279613.968096       C7aBJisS6YHP4qFEb       192.168.2.76    52094   199.59.148.201  80      tcp     http    0.486591        902     1070    SF      -       -       0       ShADadfF
1320279611.171273       CylfAj3rkBADEKeC4e      192.168.2.76    52091   192.150.187.43  80      tcp     -       5.081864        0       0       SF      -       -       0       ShAFf   5
1320279601.552622       CTohVv1l23GpiEhCSi      192.168.2.76    52086   199.59.148.20   80      tcp     http    15.200059       4078    9556    SF      -       -       0       ShADadfF
1320279610.744212       CKqXqn3T2QvvQjyYjf      192.168.2.76    52090   192.150.187.43  80      tcp     http    6.499438        1669    37688   SF      -       -       0       ShADadFf
1320279616.742259       CNJfGq1KGrxN0mlAA8      192.168.2.76    52095   208.85.41.42    80      tcp     http    0.604819        546     59445   SF      -       -       0       ShADadfF
1320279630.486420       CBCAIP3mPxYh0dJIxa      192.168.2.76    52097   199.59.148.201  80      tcp     http    0.166288        903     1070    SF      -       -       0       ShADadfF
1320279630.021607       CigUo4ZkruatGEHkj       192.168.2.76    52096   192.150.187.43  80      tcp     http    5.199366        421     15397   SF      -       -       0       ShADadfF
1320279637.215536       CU78Y01katwgtxC3p9      192.168.2.76    52100   199.59.148.201  80      tcp     http    0.264911        905     1068    SF      -       -       0       ShADadFf
1320279577.687091       C39qle3ygL7rcQHrni      192.168.2.76    52051   184.29.211.172  80      tcp     http    61.298320       1465    22567   SF      -       -       0       ShADadfF
1320279639.698701       CH5Lju3ouyBYmON0nk      192.168.2.76    52110   199.59.148.201  80      tcp     http    0.283987        901     1067    SF      -       -       0       ShADadfF
1320279638.450681       CGXh0c1myiGgCuR6I       192.168.2.76    52101   192.150.187.43  80      tcp     http    5.709781        758     19809   SF      -       -       0       ShADadFf
1320279638.954157       CIVfbqdsBvLoyoCdf       192.168.2.76    52102   192.150.187.43  80      tcp     http    5.228420        371     498     SF      -       -       0       ShADadFf
```



```
sourcetype=corelight_conn id.orig_h=192.168.21.10 |head 1
```

✓ 1 event (11/7/17 7:28:07.000 PM to 11/7/17 7:43:07.000 PM)

Selected Fields
- a host 1
- a index 1
- a source 1
- a sourcetype 1

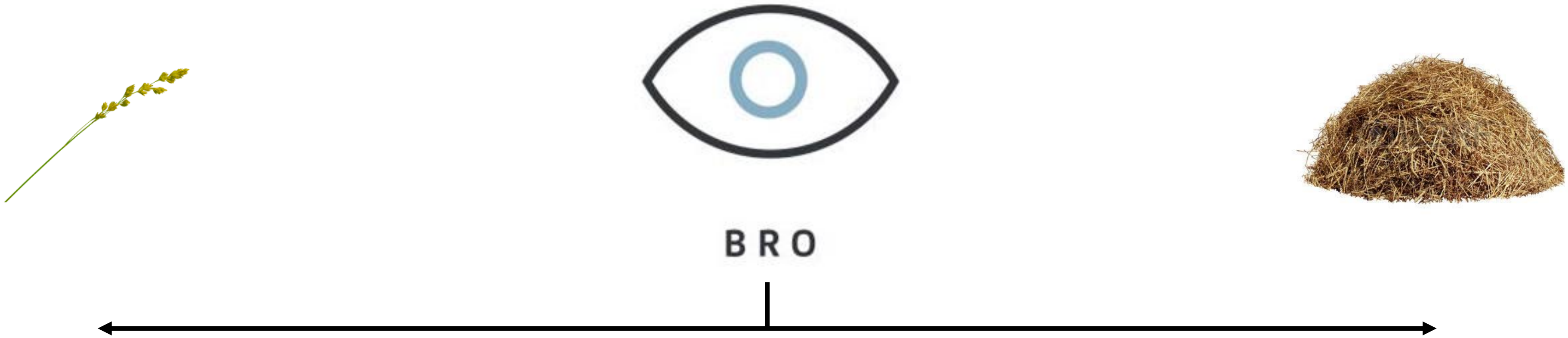
Interesting Fields
- # bytes_in 1
- # bytes_out 1
- a conn_state 1
- # date_hour 1
- # date_mday 1
- # date_minute 1
- a date_month 1
- # date_second 1
- a date_wday 1
- # date_year 1
- # date_zone 1
- a dest 1
- # dest_port 1
- # duration 1
- a eventtype 1
- a history 1
- a id.orig_h 1
- # id.orig_p 1

i	Time	Event
>	11/7/17 7:41:38.852 PM	see below

```
{ [-]
   _path: conn
   _system_name: v2
   _write_ts: 2017-11-08T00:41:38.852918Z
   conn_state: SF
   duration: 0.014446
   history: Dd
   id.orig_h: 192.168.21.10
   id.orig_p: 52232
   id.resp_h: 192.168.21.1
   id.resp_p: 53
   local_orig: true
   local_resp: true
   missed_bytes: 0
   orig_bytes: 44
   orig_ip_bytes: 72
   orig_pkts: 1
   proto: udp
   resp_bytes: 229
   resp_ip_bytes: 257
   resp_pkts: 1
   service: dns
   shunted: false
   ts: 2017-11-08T00:41:28.840768Z
   tunnel_parents: [ [+]
   ]
   uid: CUdRuI1WKyp5Yh57t3
}
```

host = v2 | index = main | source = v2 | sourcetype = corelight_conn

Just right…



NetFlow

Full PCAP

Connection Log (selected fields)

ts	1393099415.790834	Timestamp
uid	CSoqsg12YRTsWjYbZc	Unique ID
id.orig_h	2004:b9e5:6596:9876:[…]	Originator IP
id.orig_p	59258	Originator Port
id.resp_h	2b02:178:2fde:bff:[…]	Responder IP
id.resp_p	80	Responder Port
proto	tcp	IP Protocol
service	http	App-layer Protocol
duration	2.105488	Duration
orig_bytes	416	Bytes by Originator
resp_bytes	858	Bytes by Responder
conn_state	SF	TCP state
local_orig	F	Local Originator?
missed_bytes	0	Gaps
history	ShADafF	State History
tunnel_parents	Cneap78AnVWoA1yml	Outer Tunnels

DNS Log (normalized)

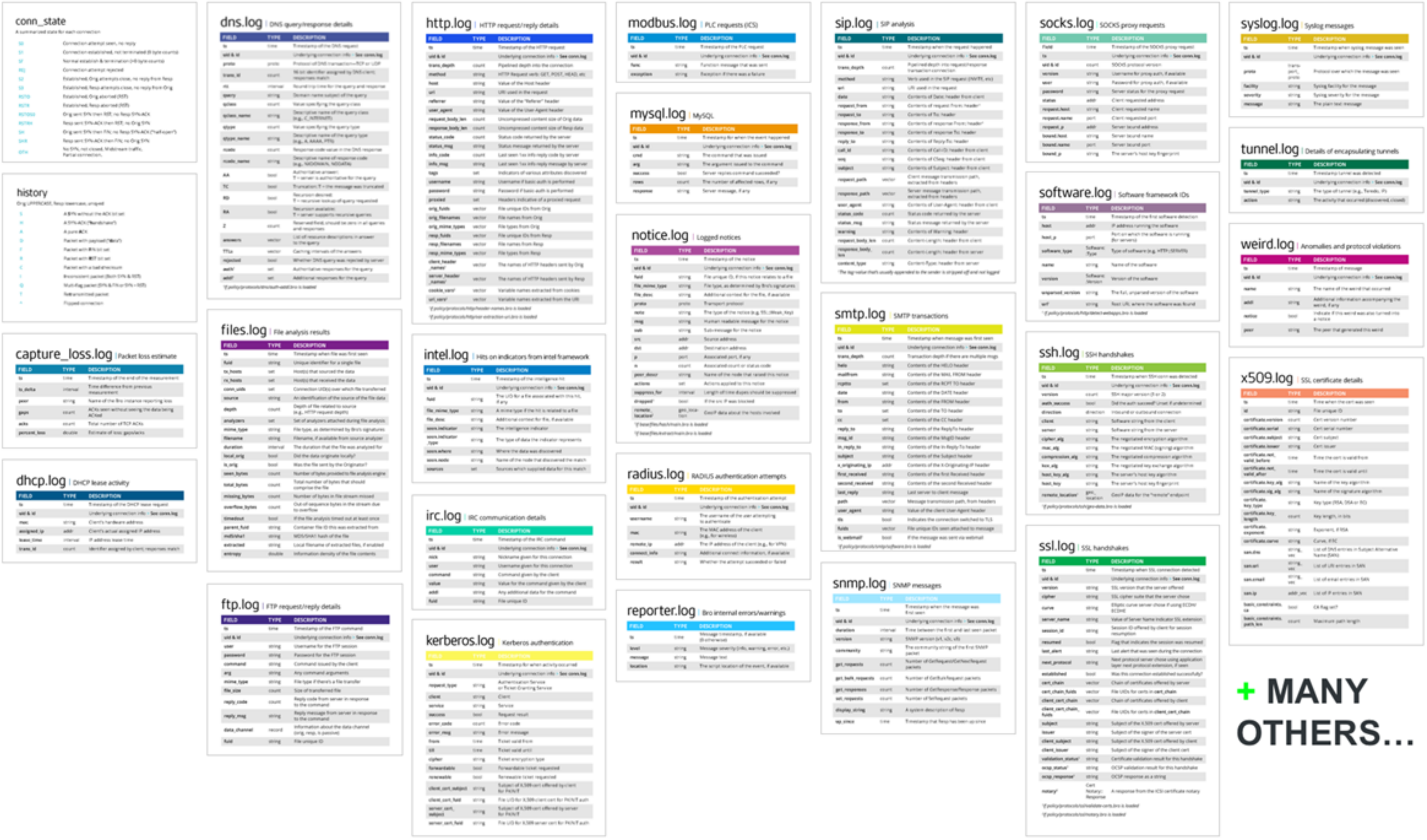
ts	2017-10-27T20:26:04.156295Z	Timestamp
uid	CSoqsg12YRTsWjYbZc	Unique ID
id.orig_h	192.168.1.108	Originator IP
id.orig_p	59258	Originator Port
id.resp_h	192.168.1.1	Responder IP
id.resp_p	53	Responder Port
trans_id	62789	Transaction ID
query	www.test.com	Query
qclass and qclass_name	(1)C_INTERNET	Query class and name
qtype and qtype_name	(1)A	Query type and name
rcode and rcode_name	(0) NOERROR	Response code and name
answers	69.172.200.235	Answers
TTLs	977.0	TTL for answers
rejected	FALSE	Rejected?
flags	"AA":false,"TC":false,"RD":true,"RA":true,"Z":0	DNS flags

HTTP log (selected fields)

ts	1393099291.589208
uid	CKFUW73bIADw0r9pl
id.orig_h	2a07:f2c0:90:402:41e:c13:6cb:99c
id.orig_p	54352
id.resp_h	2406:fe60:f47::aaeb:98c
id.resp_p	80
method	POST
host	com-services.pandonetworks.com
uri	/soapservices/services/SessionStart
referrer	-
user_agent	Mozilla/4.0 (Windows; U) Pando/2.6.0.8
status_code	200
username	anonymous
password	-
orig_mime_types	application/xml
resp_mime_types	application/xml

conn.log | IP, TCP, UDP, ICMP connection details

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the first packet
uid	string	Unique ID of the connection
id.orig_h	addr	Originating endpoint's IP address (Orig)
id.orig_p	port	Originating endpoint's TCP/UDP port (or ICMP code)
id.resp_h	addr	Responding endpoint's IP address (Resp)
id.resp_p	port	Responding endpoint's TCP/UDP port (or ICMP code)
proto	proto	Transport layer protocol of connection
service	string	Detected application protocol, if any
duration	interval	Connection length
orig_bytes	count	Orig payload bytes; from sequence numbers if TCP
resp_bytes	count	Resp payload bytes; from sequence numbers if TCP
conn_state	string	Connection state (see **conn.log** > **conn_state**)
local_orig	bool	Is Orig in Site::local_nets?
local_resp	bool	Is Resp in Site::local_nets?
missed_bytes	count	Number of bytes missing due to content gaps
history	string	Connection state history (see **conn.log** > **history**)
orig_pkts	count	Number of Orig packets
orig_ip_bytes	count	Number of Orig IP bytes (via IP total_length header field)
resp_pkts	count	Number of Resp packets
resp_ip_bytes	count	Number of Resp IP bytes (via IP total_length header field)
tunnel_parents	set	If tunneled, connection UID of encapsulating parent(s)
orig_l2_addr	string	Link-layer address of the originator
resp_l2_addr	string	Link-layer address of the responder
vlan	int	The outer VLAN for this connection
inner_vlan	int	The inner VLAN for this connection

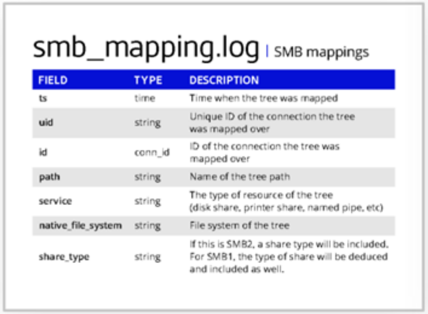

conn_state
history
capture_loss.log | Packet loss estimate
dhcp.log | DHCP lease activity
dns.log | DNS query/response details
files.log | File analysis results
ftp.log | FTP request/reply details
http.log | HTTP request/reply details
intel.log | Hits on indicators from Intel framework
irc.log | IRC communication details
kerberos.log | Kerberos authentication
modbus.log | PLC requests (ICS)
mysql.log | MySQL
notice.log | Logged notices
radius.log | RADIUS authentication attempts
reporter.log | Bro internal errors/warnings
sip.log | SIP analysis
smtp.log | SMTP transactions
snmp.log | SNMP messages
socks.log | SOCKS proxy requests
software.log | Software framework IDs
ssh.log | SSH handshakes
ssl.log | SSL handshakes
syslog.log | Syslog messages
tunnel.log | Details of encapsulating tunnels
weird.log | Anomalies and protocol violations
x509.log | SSL certificate details

+ MANY OTHERS...

corelight

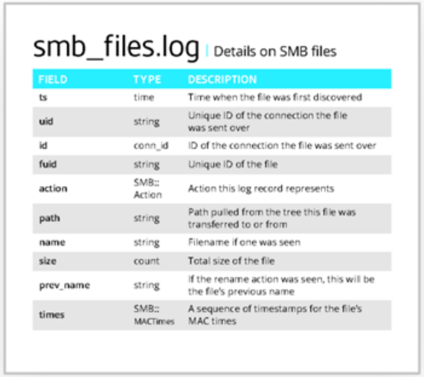
NEW SMB LOGS:

ONE COOL FEATURE AMONG MANY.



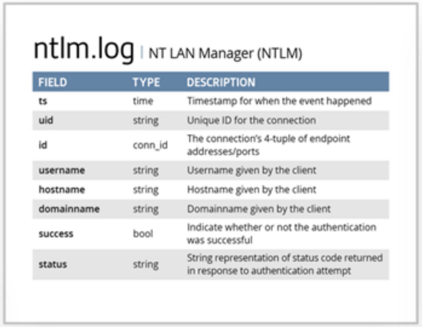
smb_mapping.log | SMB mappings

FIELD	TYPE	DESCRIPTION
ts	time	Time when the tree was mapped
uid	string	Unique ID of the connection the tree was mapped over
id	conn_id	ID of the connection the tree was mapped over
path	string	Name of the tree path
service	string	The type of resource of the tree (disk share, printer share, named pipe, etc)
native_file_system	string	File system of the tree
share_type	string	If this is SMB2, a share type will be included. For SMB1, the type of share will be deduced and included as well.



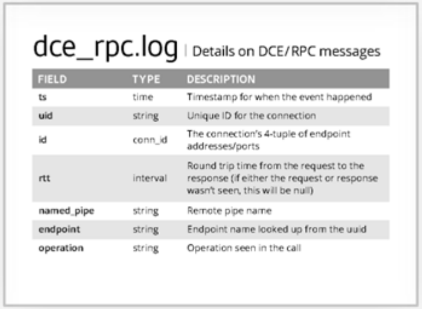
smb_files.log | Details on SMB files

FIELD	TYPE	DESCRIPTION
ts	time	Time when the file was first discovered
uid	string	Unique ID of the connection the file was sent over
id	conn_id	ID of the connection the file was sent over
fuid	string	Unique ID of the file
action	SMB::Action	Action this log record represents
path	string	Path pulled from the tree this file was transferred to or from
name	string	Filename if one was seen
size	count	Total size of the file
prev_name	string	If the rename action was seen, this will be the file's previous name
times	SMB::MACTimes	A sequence of timestamps for the file's MAC times



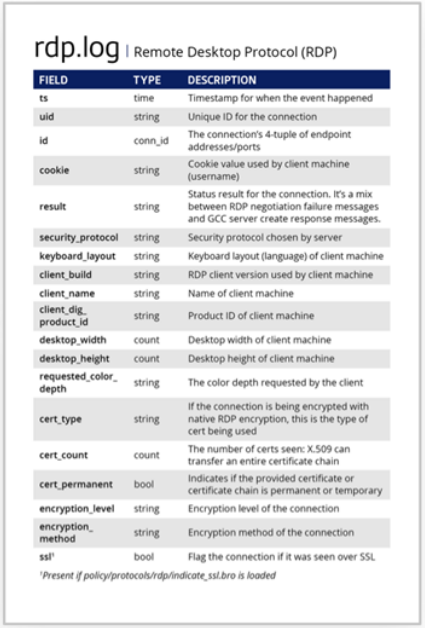
ntlm.log | NT LAN Manager (NTLM)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
username	string	Username given by the client
hostname	string	Hostname given by the client
domainname	string	Domainname given by the client
success	bool	Indicate whether or not the authentication was successful
status	string	String representation of status code returned in response to authentication attempt



dce_rpc.log | Details on DCE/RPC messages

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
rtt	interval	Round trip time from the request to the response (if either the request or response wasn't seen, this will be null)
named_pipe	string	Remote pipe name
endpoint	string	Endpoint name looked up from the uuid
operation	string	Operation seen in the call



rdp.log | Remote Desktop Protocol (RDP)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
cookie	string	Cookie value used by client machine (username)
result	string	Status result for the connection. It's a mix between RDP negotiation failure messages and GCC server create response messages.
security_protocol	string	Security protocol chosen by server
keyboard_layout	string	Keyboard layout (language) of client machine
client_build	string	RDP client version used by client machine
client_name	string	Name of client machine
client_dig_ product_id	string	Product ID of client machine
desktop_width	count	Desktop width of client machine
desktop_height	count	Desktop height of client machine
requested_color_ depth	string	The color depth requested by the client
cert_type	string	If the connection is being encrypted with native RDP encryption, this is the type of cert being used
cert_count	count	The number of certs seen: X.509 can transfer an entire certificate chain
cert_permanent	bool	Indicates if the provided certificate or certificate chain is permanent or temporary
encryption_level	string	Encryption level of the connection
encryption_ method	string	Encryption method of the connection
ssl[1]	bool	Flag the connection if it was seen over SSL

[1] *Present if policy/protocols/rdp/indicate_ssl.bro is loaded*

Ways to use the logs

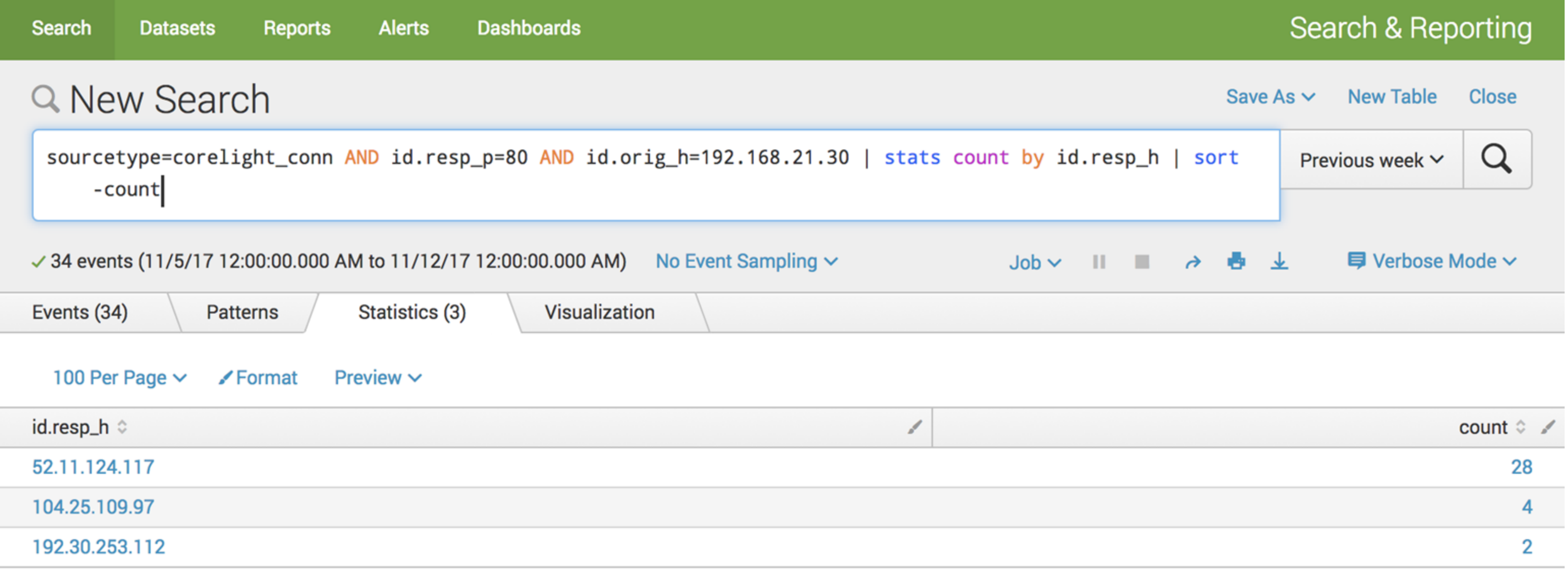
- Incident response
- Forensics
- Threat hunting
- Tracking vulnerable software
- and more...

Bro fundamentally changed the way I did incident response

- Connection log for confirmation
- Protocol logs provide context
- UID to track connections
- Files log is amazing
- SSL log for encrypted traffic

Use case: incident response scenario

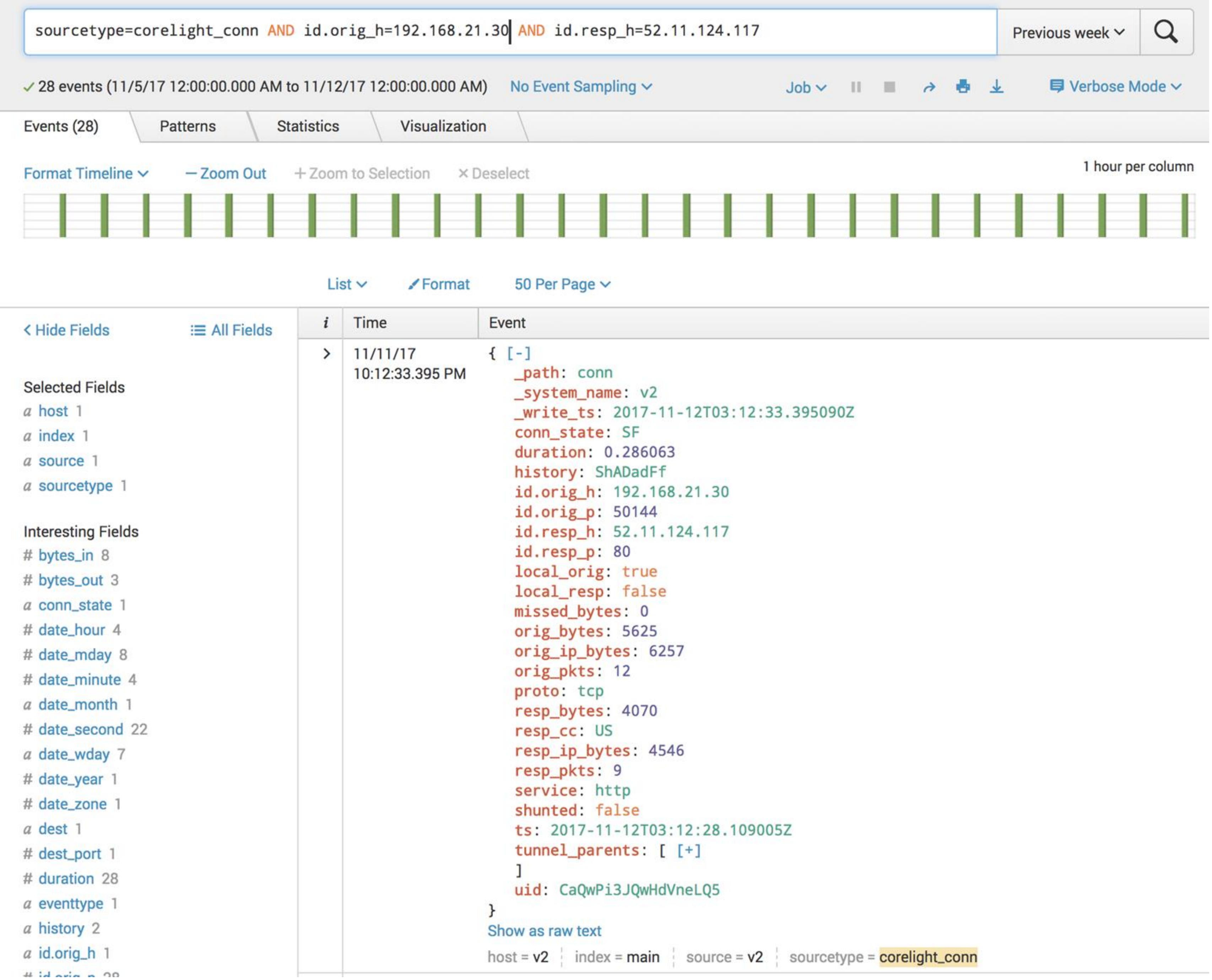
- User came back from a break and saw the machine was logged in as administrator (rarely used)
- saw a run window with a command in it
- called the security team



```
sourcetype=corelight_conn AND id.resp_p=80 AND id.orig_h=192.168.21.30 | stats count by id.resp_h | sort
    -count
```

✓ 34 events (11/5/17 12:00:00.000 AM to 11/12/17 12:00:00.000 AM)

id.resp_h	count
52.11.124.117	28
104.25.109.97	4
192.30.253.112	2



```
sourcetype=corelight_conn AND id.orig_h=192.168.21.30 AND id.resp_h=52.11.124.117
```

Previous week

✓ 28 events (11/5/17 12:00:00.000 AM to 11/12/17 12:00:00.000 AM) No Event Sampling

Job

Verbose Mode

Selected Fields

a host 1
a index 1
a source 1
a sourcetype 1

Interesting Fields

# bytes_in 8
# bytes_out 3
a conn_state 1
# date_hour 4
# date_mday 8
# date_minute 4
a date_month 1
# date_second 22
a date_wday 7
# date_year 1
# date_zone 1
a dest 1
# dest_port 1
# duration 28
a eventtype 1
a history 2
a id.orig_h 1
# id.orig_p 28

i	Time	Event
>	11/11/17 10:12:33.395 PM	see below

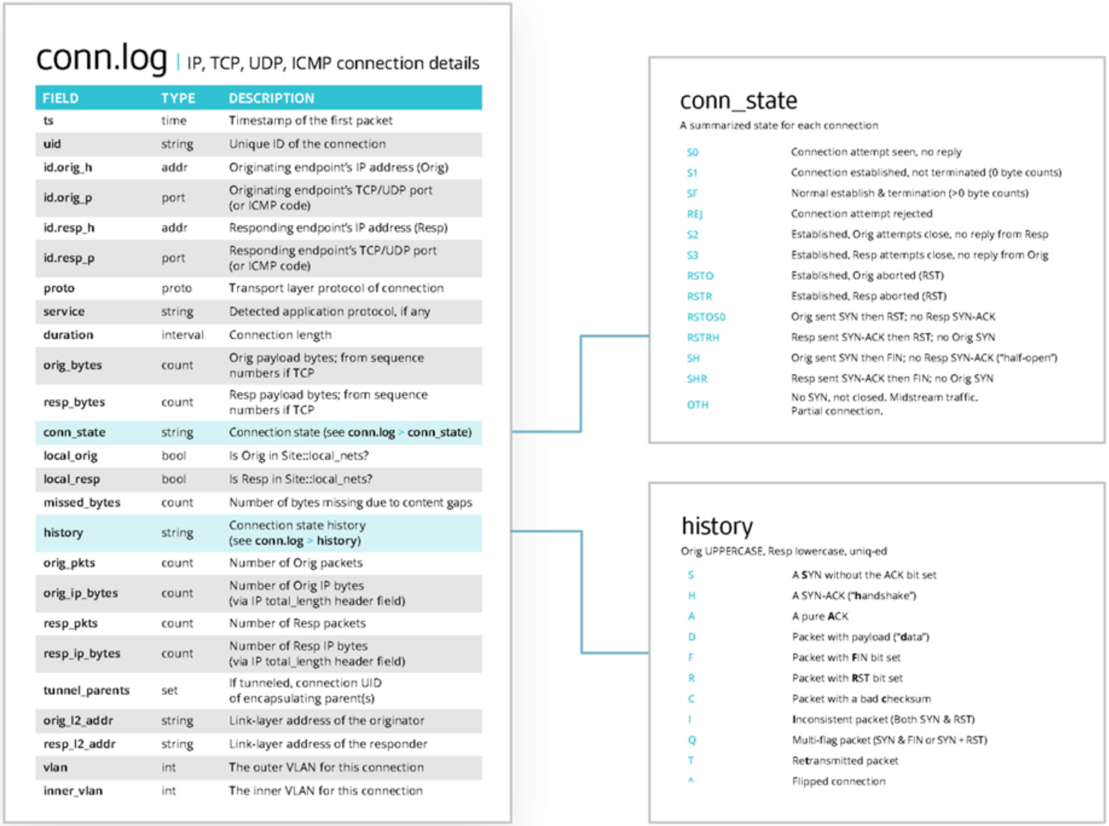
```
{ [-]
   _path: conn
   _system_name: v2
   _write_ts: 2017-11-12T03:12:33.395090Z
   conn_state: SF
   duration: 0.286063
   history: ShADadFf
   id.orig_h: 192.168.21.30
   id.orig_p: 50144
   id.resp_h: 52.11.124.117
   id.resp_p: 80
   local_orig: true
   local_resp: false
   missed_bytes: 0
   orig_bytes: 5625
   orig_ip_bytes: 6257
   orig_pkts: 12
   proto: tcp
   resp_bytes: 4070
   resp_cc: US
   resp_ip_bytes: 4546
   resp_pkts: 9
   service: http
   shunted: false
   ts: 2017-11-12T03:12:28.109005Z
   tunnel_parents: [ [+]
   ]
   uid: CaQwPi3JQwHdVneLQ5
}
```

Show as raw text

host = v2 | index = main | source = v2 | sourcetype = corelight_conn

UID:

ONE COOL FEATURE AMONG MANY.



conn.log | IP, TCP, UDP, ICMP connection details

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the first packet
uid	string	Unique ID of the connection
id.orig_h	addr	Originating endpoint's IP address (Orig)
id.orig_p	port	Originating endpoint's TCP/UDP port (or ICMP code)
id.resp_h	addr	Responding endpoint's IP address (Resp)
id.resp_p	port	Responding endpoint's TCP/UDP port (or ICMP code)
proto	proto	Transport layer protocol of connection
service	string	Detected application protocol, if any
duration	interval	Connection length
orig_bytes	count	Orig payload bytes; from sequence numbers if TCP
resp_bytes	count	Resp payload bytes; from sequence numbers if TCP
conn_state	string	Connection state (see **conn.log** > **conn_state**)
local_orig	bool	Is Orig in Site::local_nets?
local_resp	bool	Is Resp in Site::local_nets?
missed_bytes	count	Number of bytes missing due to content gaps
history	string	Connection state history (see **conn.log** > **history**)
orig_pkts	count	Number of Orig packets
orig_ip_bytes	count	Number of Orig IP bytes (via IP total_length header field)
resp_pkts	count	Number of Resp packets
resp_ip_bytes	count	Number of Resp IP bytes (via IP total_length header field)
tunnel_parents	set	If tunneled, connection UID of encapsulating parent(s)
orig_l2_addr	string	Link-layer address of the originator
resp_l2_addr	string	Link-layer address of the responder
vlan	int	The outer VLAN for this connection
inner_vlan	int	The inner VLAN for this connection

conn_state

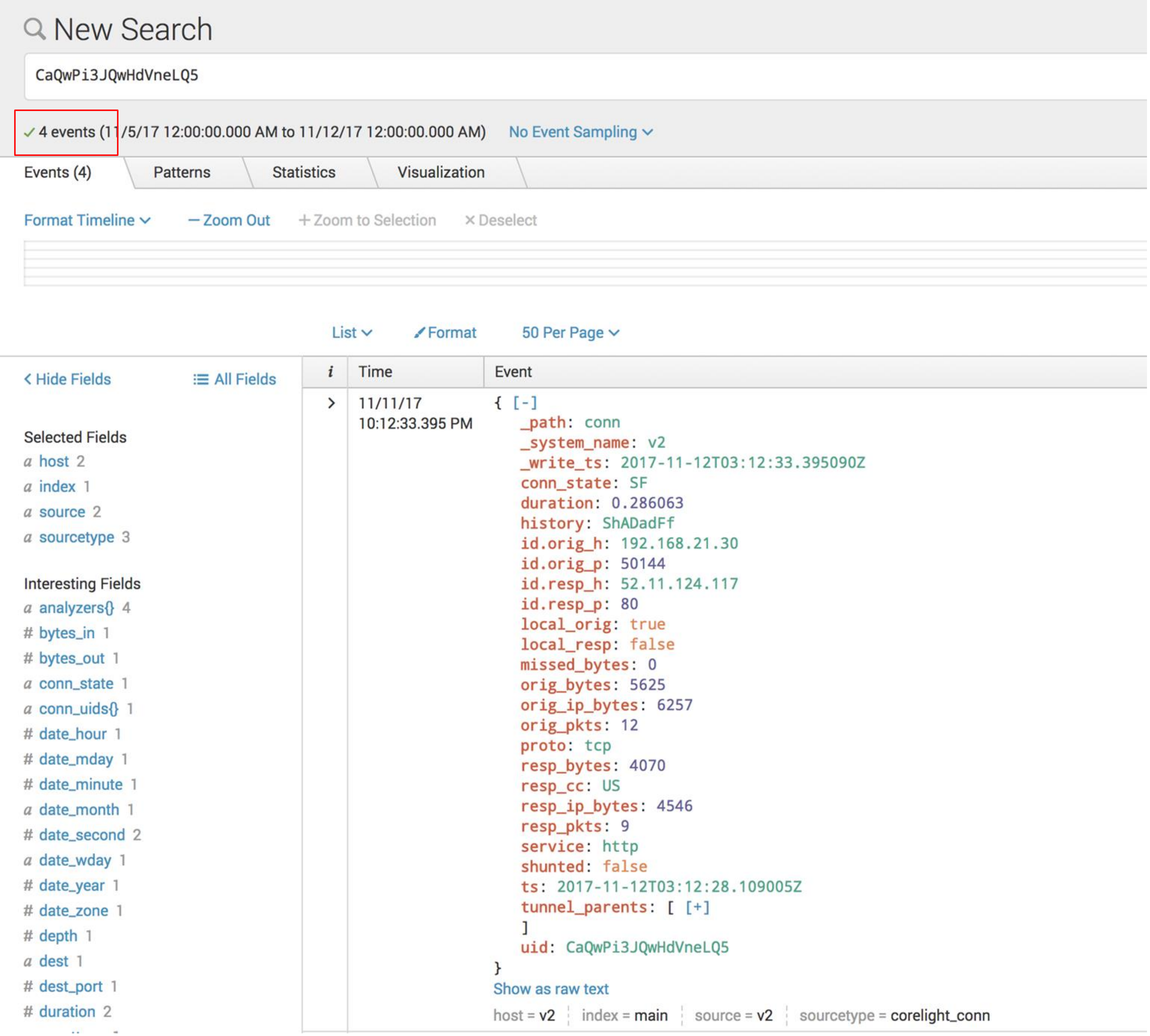
A summarized state for each connection

S0	Connection attempt seen, no reply
S1	Connection established, not terminated (0 byte counts)
SF	Normal establish & termination (>0 byte counts)
REJ	Connection attempt rejected
S2	Established, Orig attempts close, no reply from Resp
S3	Established, Resp attempts close, no reply from Orig
RSTO	Established, Orig aborted (RST)
RSTR	Established, Resp aborted (RST)
RSTOS0	Orig sent SYN then RST; no Resp SYN-ACK
RSTRH	Resp sent SYN-ACK then RST; no Orig SYN
SH	Orig sent SYN then FIN; no Resp SYN-ACK ("half-open")
SHR	Resp sent SYN-ACK then FIN; no Orig SYN
OTH	No SYN, not closed. Midstream traffic. Partial connection.

history

Orig UPPERCASE, Resp lowercase, uniq-ed

S	A **S**YN without the ACK bit set
H	A SYN-ACK ("**h**andshake")
A	A pure **A**CK
D	Packet with payload ("**d**ata")
F	Packet with **F**IN bit set
R	Packet with **R**ST bit set
C	Packet with a bad **c**hecksum
I	**I**nconsistent packet (Both SYN & RST)
Q	Multi-flag packet (SYN & FIN or SYN + RST)
T	Re**t**ransmitted packet
^	Flipped connection



New Search

CaQwPi3JQwHdVneLQ5

✓ 4 events (11/5/17 12:00:00.000 AM to 11/12/17 12:00:00.000 AM) No Event Sampling

Selected Fields

a	host	2
a	index	1
a	source	2
a	sourcetype	3

Interesting Fields

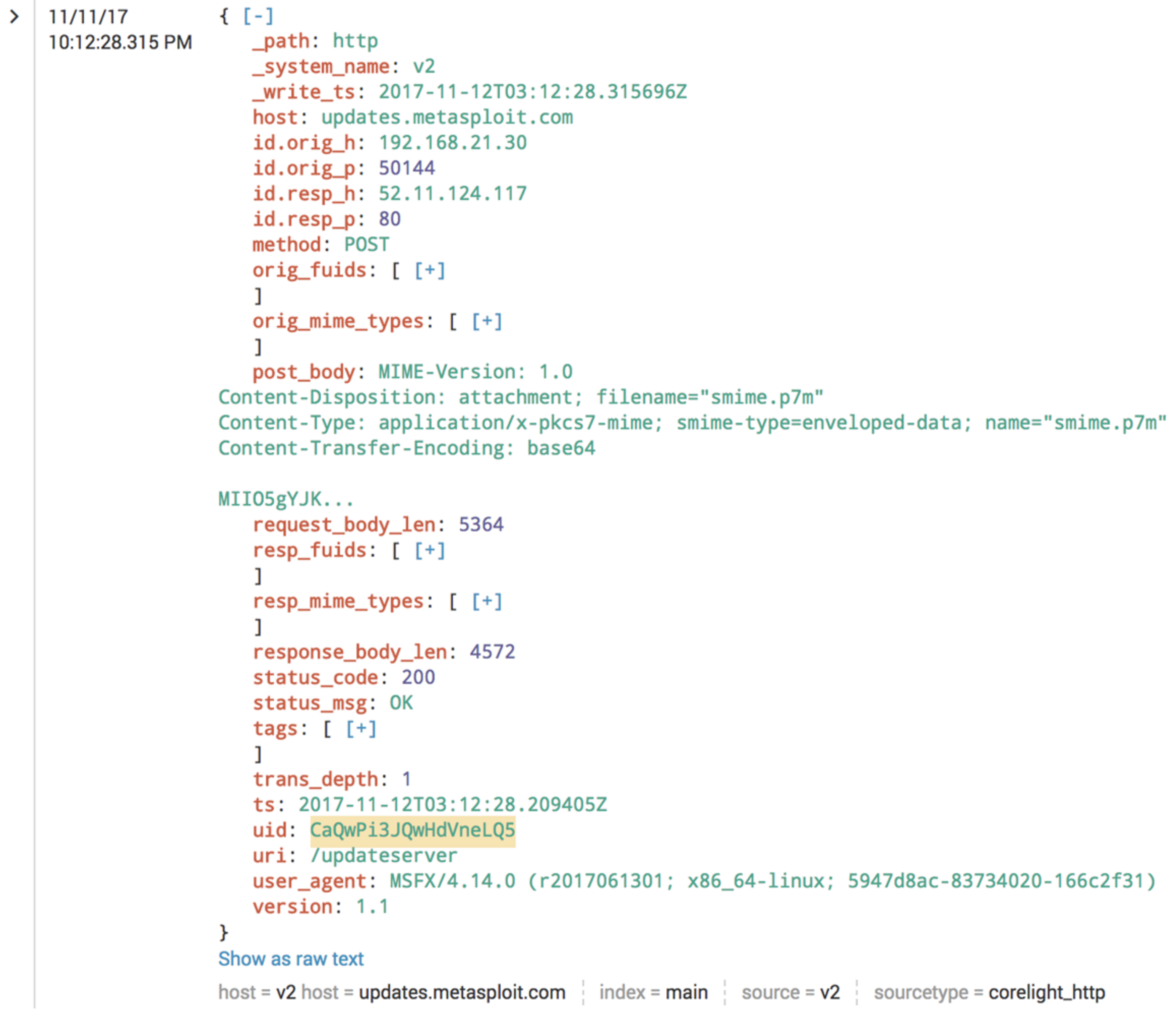
a	analyzers{}	4
#	bytes_in	1
#	bytes_out	1
a	conn_state	1
a	conn_uids{}	1
#	date_hour	1
#	date_mday	1
#	date_minute	1
a	date_month	1
#	date_second	2
a	date_wday	1
#	date_year	1
#	date_zone	1
#	depth	1
a	dest	1
#	dest_port	1
#	duration	2

i	Time	Event
>	11/11/17 10:12:33.395 PM	see below

```
{ [-]
   _path: conn
   _system_name: v2
   _write_ts: 2017-11-12T03:12:33.395090Z
   conn_state: SF
   duration: 0.286063
   history: ShADadFf
   id.orig_h: 192.168.21.30
   id.orig_p: 50144
   id.resp_h: 52.11.124.117
   id.resp_p: 80
   local_orig: true
   local_resp: false
   missed_bytes: 0
   orig_bytes: 5625
   orig_ip_bytes: 6257
   orig_pkts: 12
   proto: tcp
   resp_bytes: 4070
   resp_cc: US
   resp_ip_bytes: 4546
   resp_pkts: 9
   service: http
   shunted: false
   ts: 2017-11-12T03:12:28.109005Z
   tunnel_parents: [ [+]
   ]
   uid: CaQwPi3JQwHdVneLQ5
}
```

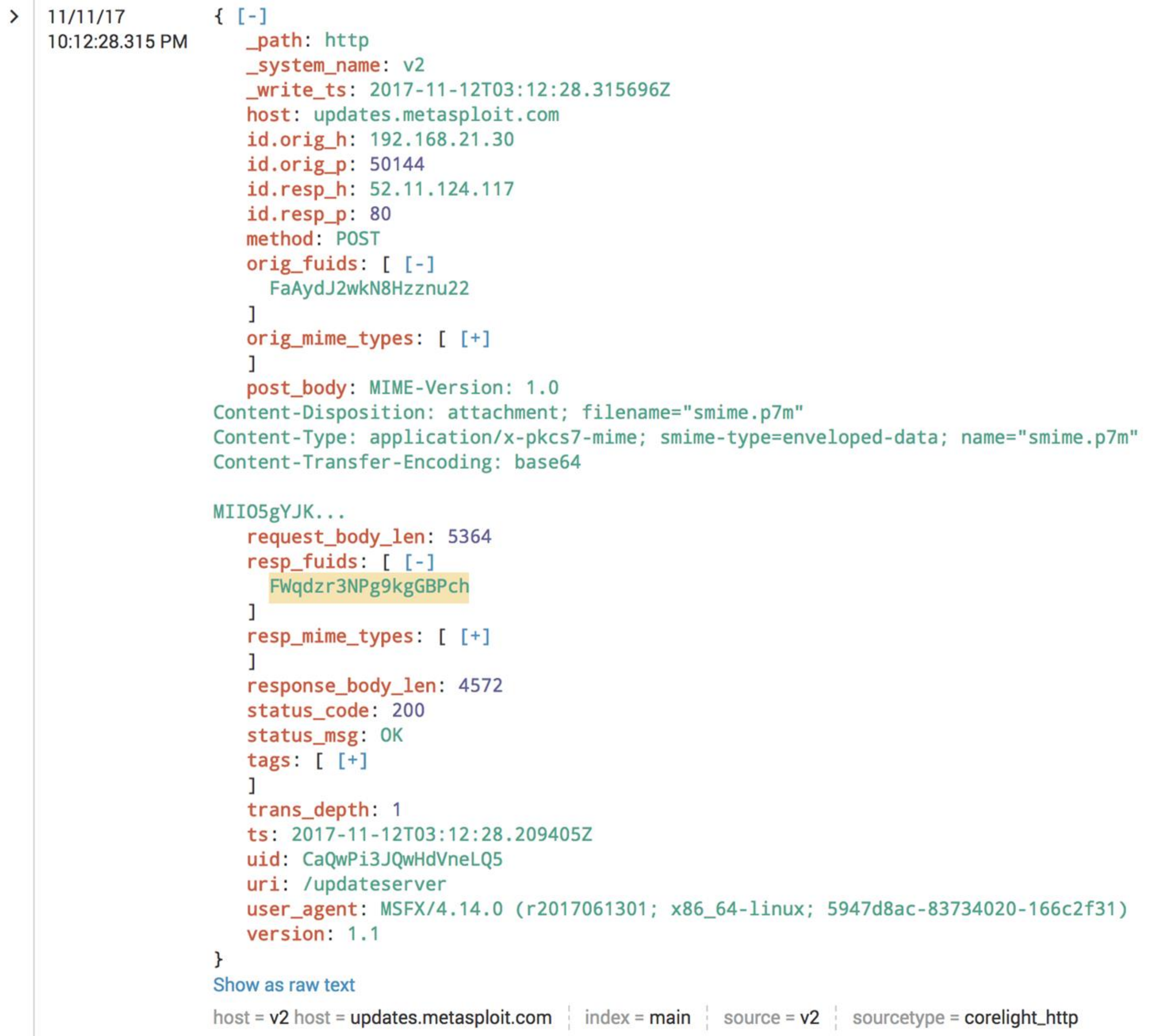
Show as raw text

host = v2 | index = main | source = v2 | sourcetype = corelight_conn



```
> 11/11/17            { [-]
  10:12:28.315 PM        _path: http
                         _system_name: v2
                         _write_ts: 2017-11-12T03:12:28.315696Z
                         host: updates.metasploit.com
                         id.orig_h: 192.168.21.30
                         id.orig_p: 50144
                         id.resp_h: 52.11.124.117
                         id.resp_p: 80
                         method: POST
                         orig_fuids: [ [+]
                         ]
                         orig_mime_types: [ [+]
                         ]
                         post_body: MIME-Version: 1.0
                      Content-Disposition: attachment; filename="smime.p7m"
                      Content-Type: application/x-pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"
                      Content-Transfer-Encoding: base64

                      MIIO5gYJK...
                         request_body_len: 5364
                         resp_fuids: [ [+]
                         ]
                         resp_mime_types: [ [+]
                         ]
                         response_body_len: 4572
                         status_code: 200
                         status_msg: OK
                         tags: [ [+]
                         ]
                         trans_depth: 1
                         ts: 2017-11-12T03:12:28.209405Z
                         uid: CaQwPi3JQwHdVneLQ5
                         uri: /updateserver
                         user_agent: MSFX/4.14.0 (r2017061301; x86_64-linux; 5947d8ac-83734020-166c2f31)
                         version: 1.1
                      }
                      Show as raw text
                      host = v2 host = updates.metasploit.com | index = main | source = v2 | sourcetype = corelight_http
```

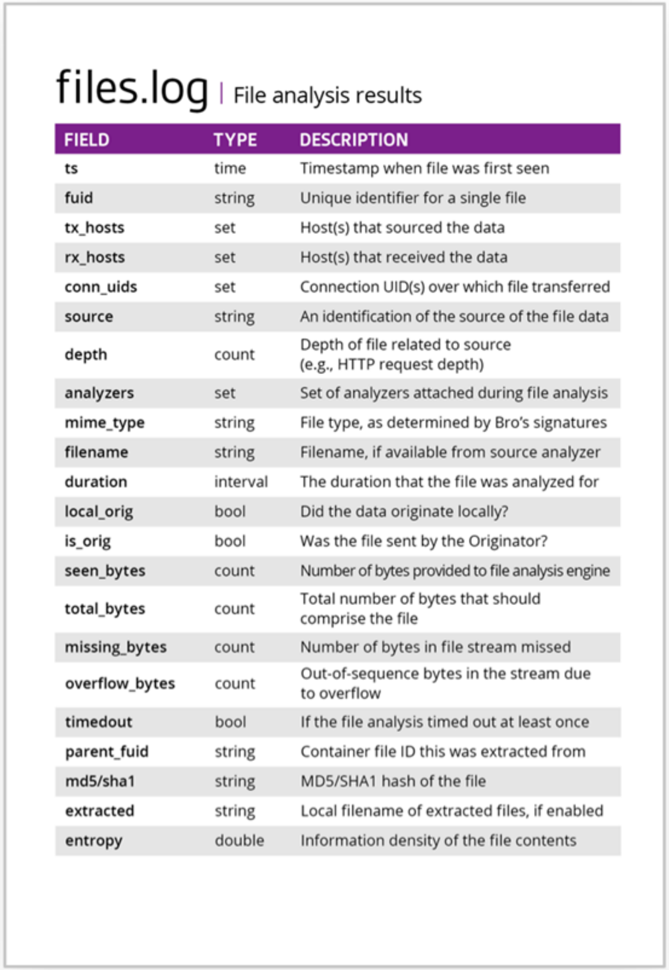


```
>  11/11/17          { [-]
   10:12:28.315 PM      _path: http
                        _system_name: v2
                        _write_ts: 2017-11-12T03:12:28.315696Z
                        host: updates.metasploit.com
                        id.orig_h: 192.168.21.30
                        id.orig_p: 50144
                        id.resp_h: 52.11.124.117
                        id.resp_p: 80
                        method: POST
                        orig_fuids: [ [-]
                          FaAydJ2wkN8Hzznu22
                        ]
                        orig_mime_types: [ [+]
                        ]
                        post_body: MIME-Version: 1.0
                     Content-Disposition: attachment; filename="smime.p7m"
                     Content-Type: application/x-pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"
                     Content-Transfer-Encoding: base64

                     MIIO5gYJK...
                        request_body_len: 5364
                        resp_fuids: [ [-]
                          FWqdzr3NPg9kgGBPch
                        ]
                        resp_mime_types: [ [+]
                        ]
                        response_body_len: 4572
                        status_code: 200
                        status_msg: OK
                        tags: [ [+]
                        ]
                        trans_depth: 1
                        ts: 2017-11-12T03:12:28.209405Z
                        uid: CaQwPi3JQwHdVneLQ5
                        uri: /updateserver
                        user_agent: MSFX/4.14.0 (r2017061301; x86_64-linux; 5947d8ac-83734020-166c2f31)
                        version: 1.1
                     }
                     Show as raw text
                     host = v2 host = updates.metasploit.com | index = main | source = v2 | sourcetype = corelight_http
```

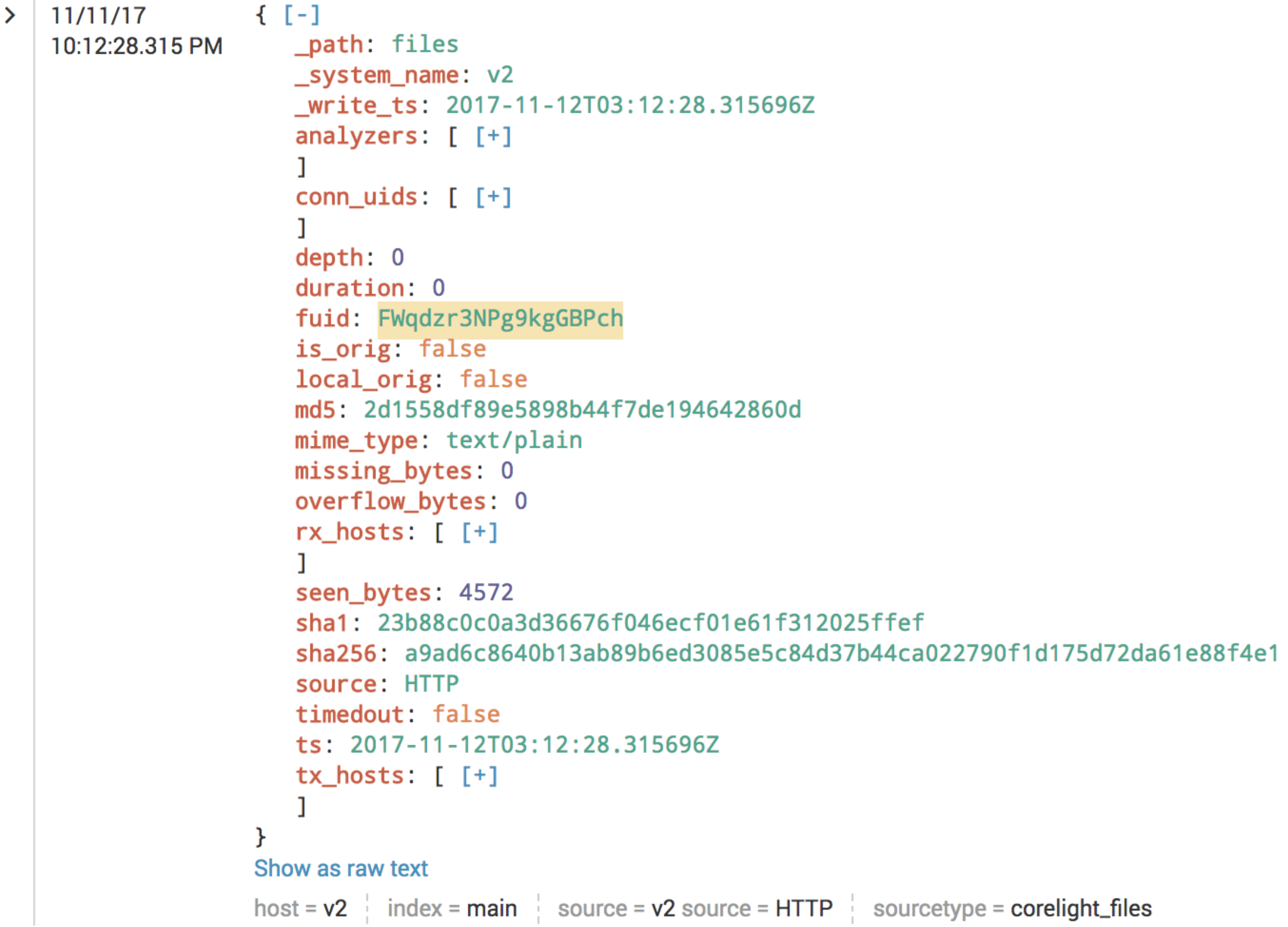
FILE ANALYSIS:

ONE COOL FEATURE AMONG MANY.



files.log | File analysis results

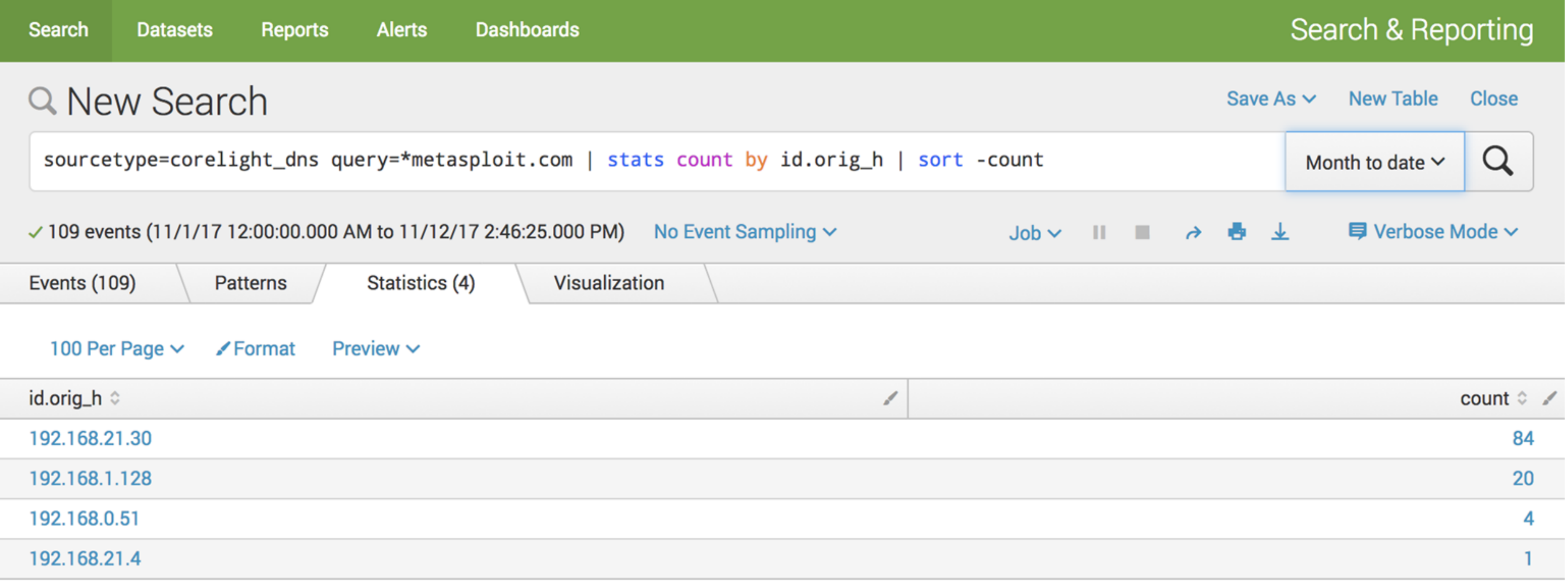
FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when file was first seen
fuid	string	Unique identifier for a single file
tx_hosts	set	Host(s) that sourced the data
rx_hosts	set	Host(s) that received the data
conn_uids	set	Connection UID(s) over which file transferred
source	string	An identification of the source of the file data
depth	count	Depth of file related to source (e.g., HTTP request depth)
analyzers	set	Set of analyzers attached during file analysis
mime_type	string	File type, as determined by Bro's signatures
filename	string	Filename, if available from source analyzer
duration	interval	The duration that the file was analyzed for
local_orig	bool	Did the data originate locally?
is_orig	bool	Was the file sent by the Originator?
seen_bytes	count	Number of bytes provided to file analysis engine
total_bytes	count	Total number of bytes that should comprise the file
missing_bytes	count	Number of bytes in file stream missed
overflow_bytes	count	Out-of-sequence bytes in the stream due to overflow
timedout	bool	If the file analysis timed out at least once
parent_fuid	string	Container file ID this was extracted from
md5/sha1	string	MD5/SHA1 hash of the file
extracted	string	Local filename of extracted files, if enabled
entropy	double	Information density of the file contents



```
>  11/11/17          { [-]
   10:12:28.315 PM      _path: files
                        _system_name: v2
                        _write_ts: 2017-11-12T03:12:28.315696Z
                        analyzers: [ [+]
                        ]
                        conn_uids: [ [+]
                        ]
                        depth: 0
                        duration: 0
                        fuid: FWqdzr3NPg9kgGBPch
                        is_orig: false
                        local_orig: false
                        md5: 2d1558df89e5898b44f7de194642860d
                        mime_type: text/plain
                        missing_bytes: 0
                        overflow_bytes: 0
                        rx_hosts: [ [+]
                        ]
                        seen_bytes: 4572
                        sha1: 23b88c0c0a3d36676f046ecf01e61f312025ffef
                        sha256: a9ad6c8640b13ab89b6ed3085e5c84d37b44ca022790f1d175d72da61e88f4e1
                        source: HTTP
                        timedout: false
                        ts: 2017-11-12T03:12:28.315696Z
                        tx_hosts: [ [+]
                        ]
                     }
                     Show as raw text
                     host = v2 | index = main | source = v2 source = HTTP | sourcetype = corelight_files
```

Use case: Forensics

- Since Bro is not alert based
- Same data available back in time
- ALL of your connections, files, protocols!!!
- Query for a URI, hash, domain name, whatever



```
sourcetype=corelight_dns query=*metasploit.com | stats count by id.orig_h | sort -count
```

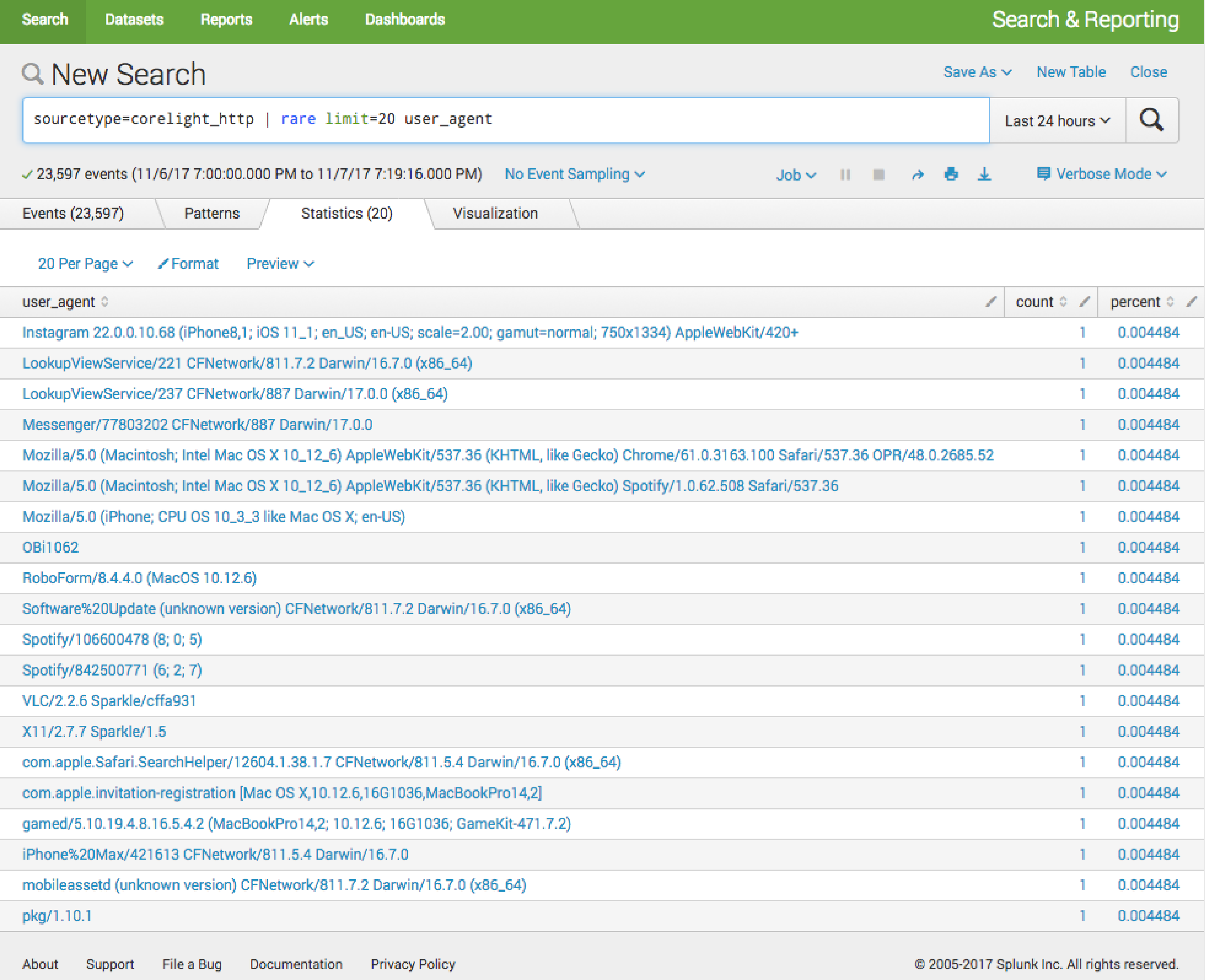
✓ 109 events (11/1/17 12:00:00.000 AM to 11/12/17 2:46:25.000 PM)

id.orig_h	count
192.168.21.30	84
192.168.1.128	20
192.168.0.51	4
192.168.21.4	1

General threat hunting with Bro

Examples:

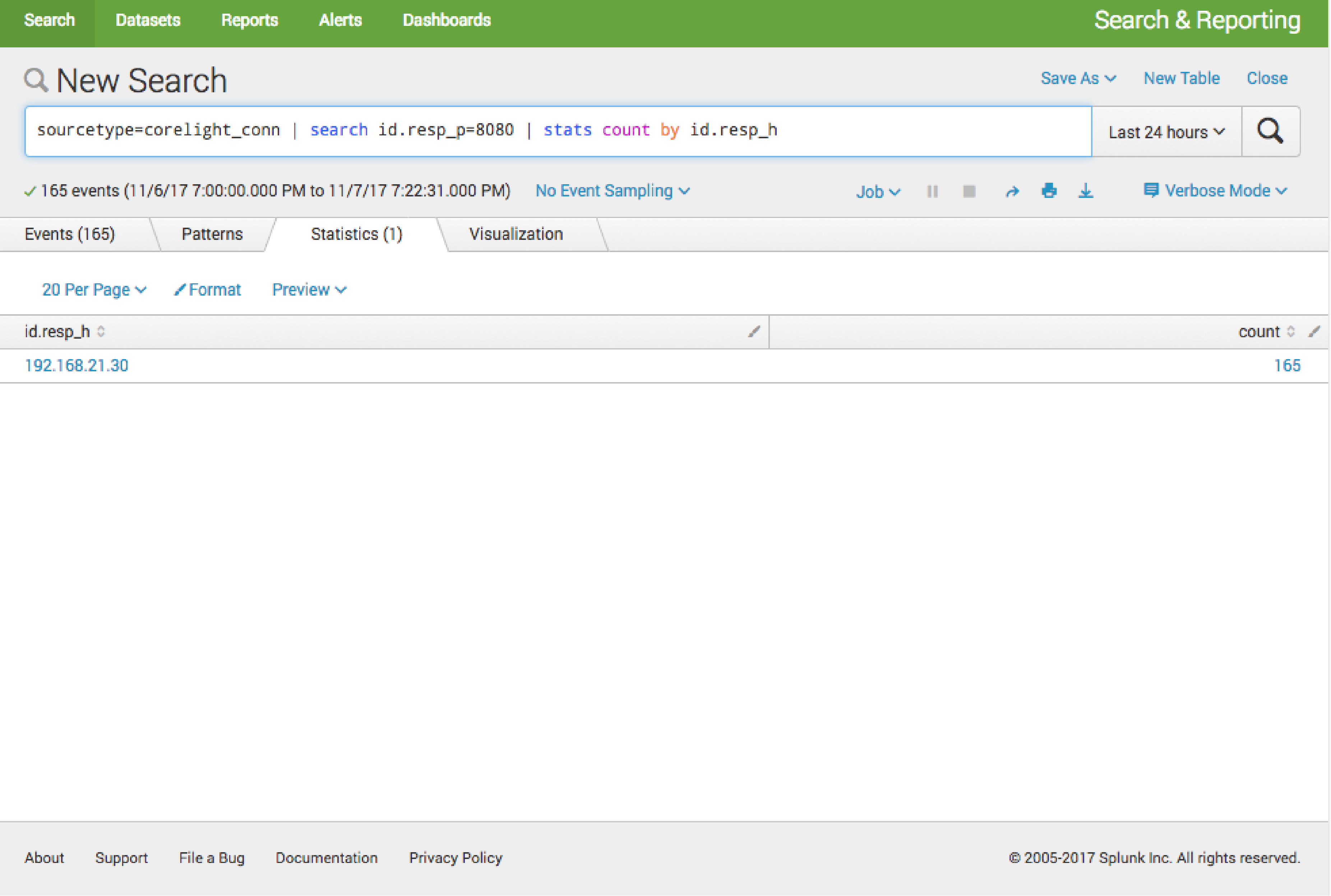
- What are rare user agents?
- How many local servers answering on port 8080?
- How many clients are using TLSv1?



```
sourcetype=corelight_http | rare limit=20 user_agent
```

✓ 23,597 events (11/6/17 7:00:00.000 PM to 11/7/17 7:19:16.000 PM)

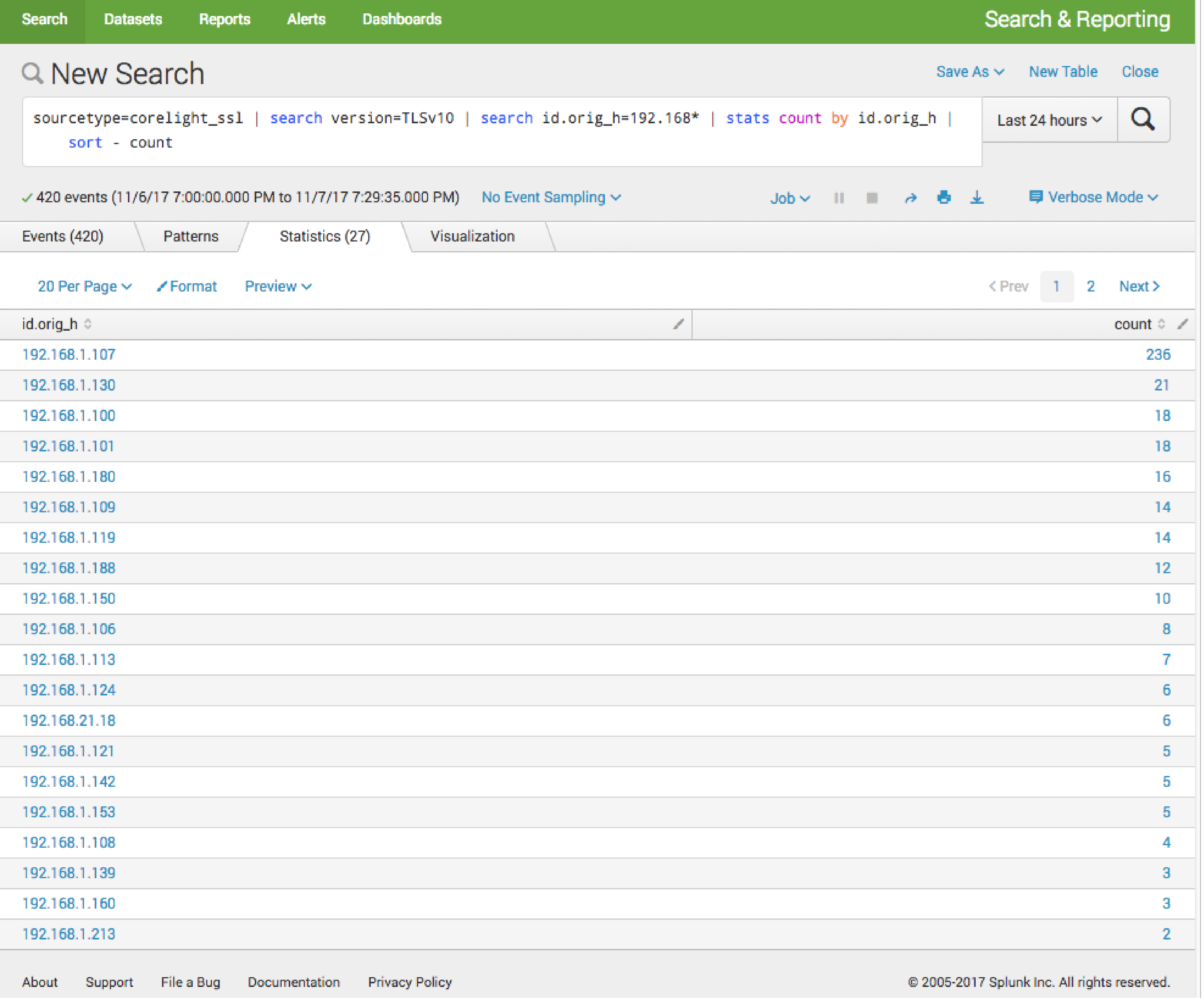
user_agent	count	percent
Instagram 22.0.0.10.68 (iPhone8,1; iOS 11_1; en_US; en-US; scale=2.00; gamut=normal; 750x1334) AppleWebKit/420+	1	0.004484
LookupViewService/221 CFNetwork/811.7.2 Darwin/16.7.0 (x86_64)	1	0.004484
LookupViewService/237 CFNetwork/887 Darwin/17.0.0 (x86_64)	1	0.004484
Messenger/77803202 CFNetwork/887 Darwin/17.0.0	1	0.004484
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36 OPR/48.0.2685.52	1	0.004484
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Spotify/1.0.62.508 Safari/537.36	1	0.004484
Mozilla/5.0 (iPhone; CPU OS 10_3_3 like Mac OS X; en-US)	1	0.004484
OBi1062	1	0.004484
RoboForm/8.4.4.0 (MacOS 10.12.6)	1	0.004484
Software%20Update (unknown version) CFNetwork/811.7.2 Darwin/16.7.0 (x86_64)	1	0.004484
Spotify/106600478 (8; 0; 5)	1	0.004484
Spotify/842500771 (6; 2; 7)	1	0.004484
VLC/2.2.6 Sparkle/cffa931	1	0.004484
X11/2.7.7 Sparkle/1.5	1	0.004484
com.apple.Safari.SearchHelper/12604.1.38.1.7 CFNetwork/811.5.4 Darwin/16.7.0 (x86_64)	1	0.004484
com.apple.invitation-registration [Mac OS X,10.12.6,16G1036,MacBookPro14,2]	1	0.004484
gamed/5.10.19.4.8.16.5.4.2 (MacBookPro14,2; 10.12.6; 16G1036; GameKit-471.7.2)	1	0.004484
iPhone%20Max/421613 CFNetwork/811.5.4 Darwin/16.7.0	1	0.004484
mobileassetd (unknown version) CFNetwork/811.7.2 Darwin/16.7.0 (x86_64)	1	0.004484
pkg/1.10.1	1	0.004484



```
sourcetype=corelight_conn | search id.resp_p=8080 | stats count by id.resp_h
```

✓ 165 events (11/6/17 7:00:00.000 PM to 11/7/17 7:22:31.000 PM)

id.resp_h	count
192.168.21.30	165



```
sourcetype=corelight_ssl | search version=TLSv10 | search id.orig_h=192.168* | stats count by id.orig_h |
    sort - count
```

✓ 420 events (11/6/17 7:00:00.000 PM to 11/7/17 7:29:35.000 PM)

id.orig_h	count
192.168.1.107	236
192.168.1.130	21
192.168.1.100	18
192.168.1.101	18
192.168.1.180	16
192.168.1.109	14
192.168.1.119	14
192.168.1.188	12
192.168.1.150	10
192.168.1.106	8
192.168.1.113	7
192.168.1.124	6
192.168.21.18	6
192.168.1.121	5
192.168.1.142	5
192.168.1.153	5
192.168.1.108	4
192.168.1.139	3
192.168.1.160	3
192.168.1.213	2

Dynamic Protocol Detection (DPD)

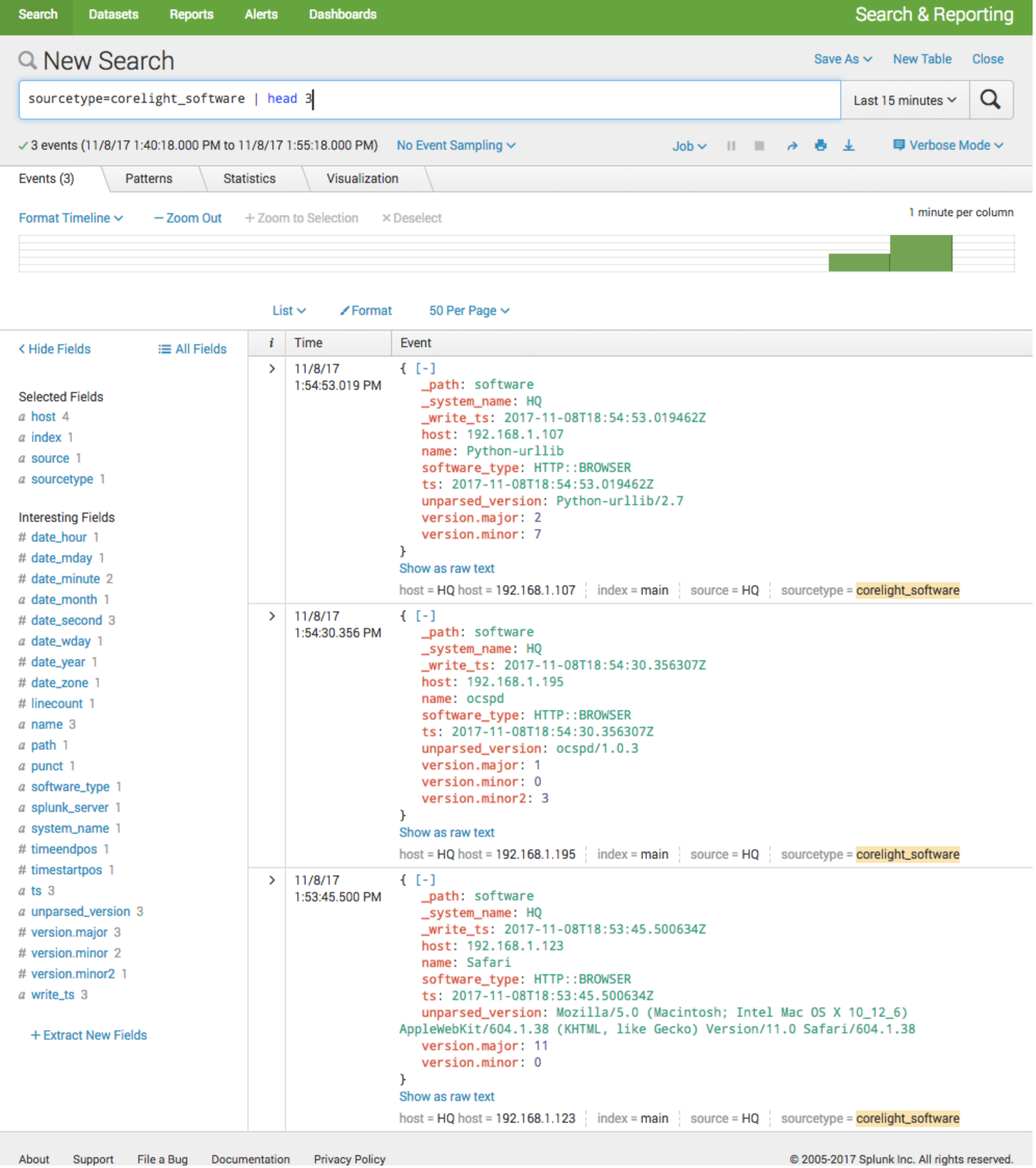
- DPD means that you’ll see the protocol no matter what ports are used
- Don’t need to be limited to searching ports
- Find off port protocol usage easily

File extraction

- Bro can optionally extract all the files it sees
- This can be done for forensics or integration with static or dynamic analysis
- Gets a lot closer to getting what you want out of PCAP

Use case - tracking vulnerable software

- software.log provides rich data about local software seen
- easy to search and script for response
- software log to monitor for strange versions and names



```
sourcetype=corelight_software | head 3
```

✓ 3 events (11/8/17 1:40:18.000 PM to 11/8/17 1:55:18.000 PM)

Selected Fields
- a host 4
- a index 1
- a source 1
- a sourcetype 1

Interesting Fields
- # date_hour 1
- # date_mday 1
- # date_minute 2
- a date_month 1
- # date_second 3
- a date_wday 1
- # date_year 1
- # date_zone 1
- # linecount 1
- a name 3
- a path 1
- a punct 1
- a software_type 1
- a splunk_server 1
- a system_name 1
- # timeendpos 1
- # timestartpos 1
- a ts 3
- a unparsed_version 3
- # version.major 3
- # version.minor 2
- # version.minor2 1
- a write_ts 3

```
i   Time                 Event
>   11/8/17              { [-]
    1:54:53.019 PM          _path: software
                            _system_name: HQ
                            _write_ts: 2017-11-08T18:54:53.019462Z
                            host: 192.168.1.107
                            name: Python-urllib
                            software_type: HTTP::BROWSER
                            ts: 2017-11-08T18:54:53.019462Z
                            unparsed_version: Python-urllib/2.7
                            version.major: 2
                            version.minor: 7
                         }
                         Show as raw text
                         host = HQ host = 192.168.1.107 | index = main | source = HQ | sourcetype = corelight_software
>   11/8/17              { [-]
    1:54:30.356 PM          _path: software
                            _system_name: HQ
                            _write_ts: 2017-11-08T18:54:30.356307Z
                            host: 192.168.1.195
                            name: ocspd
                            software_type: HTTP::BROWSER
                            ts: 2017-11-08T18:54:30.356307Z
                            unparsed_version: ocspd/1.0.3
                            version.major: 1
                            version.minor: 0
                            version.minor2: 3
                         }
                         Show as raw text
                         host = HQ host = 192.168.1.195 | index = main | source = HQ | sourcetype = corelight_software
>   11/8/17              { [-]
    1:53:45.500 PM          _path: software
                            _system_name: HQ
                            _write_ts: 2017-11-08T18:53:45.500634Z
                            host: 192.168.1.123
                            name: Safari
                            software_type: HTTP::BROWSER
                            ts: 2017-11-08T18:53:45.500634Z
                            unparsed_version: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6)
                         AppleWebKit/604.1.38 (KHTML, like Gecko) Version/11.0 Safari/604.1.38
                            version.major: 11
                            version.minor: 0
                         }
                         Show as raw text
                         host = HQ host = 192.168.1.123 | index = main | source = HQ | sourcetype = corelight_software
```

© 2005-2017 Splunk Inc. All rights reserved.

Bro scripting

- Bro is an event engine
- Bro scripting gives you a domain specific language to express simple and complex policies (scripts)
- Bro Package Manager
- So many possibilities - time for another webcast!

Visit try.bro.org for a quick intro to Bro scripting

corelight

All aboard!

- Visit bro.org for docs and training
- Come to Brocon 2018 or other events!
- Most of all, install Bro and use the logs for IR
- Write or edit a Bro script
- Corelight for enterprise Bro deployment

corelight

Corelight
Sensor

SCALABILITY

- 3-5x performance compared to self-engineered Bro
- Optimized file extraction
- Multiple simultaneous exports

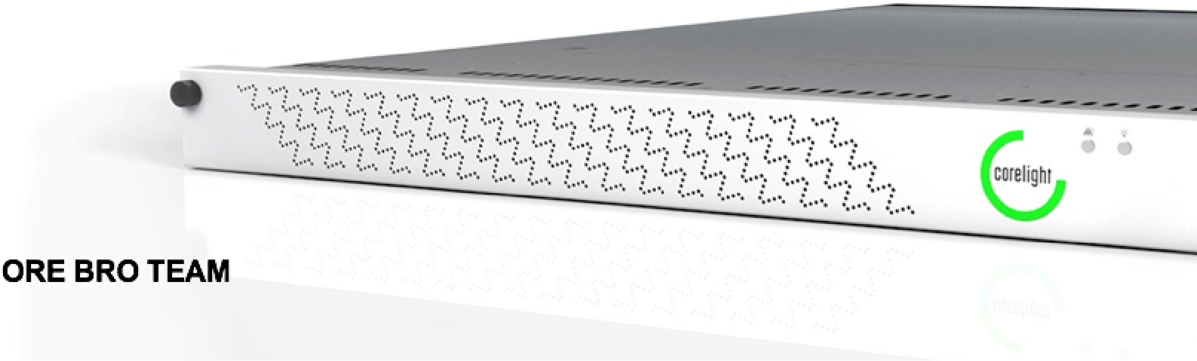
MANAGEMENT

- Comprehensive API
- Python Client

CUSTOM LOGIC AND APPLICATIONS

- Flexible filtering, custom scripts

ENTERPRISE SUPPORT FROM THE CORE BRO TEAM



Questions??

Thank you!

vince@corelight.com

corelight.com

Bro Log Cheat Sheets:

https://github.com/corelight/bro-cheatsheets