



Splunk App for Stream

David Shpritz, Aplura LLC.

Baltimore Area User Group

3/21/2016

```

32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 nxbob

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR S  PPID  RSSZ  RUSR  RDSZ  RDBR
0.3 splunk  1000    0:02.80 4.3  20  0  135356 43956 8940 S  1  2880 splunk  2880  2880 splunk
0.3 tcpdump  72     0:02.45 0.7  20  0  28588  6872  5464 S  32680  72 tcpdump  72  72 tcpdump
0.3 named    25     7:31.12 1.8  20  0  548272 18872  5584 S  1  25 named  25  25 named
0.3 dan     1028    0:00.08 0.6  20  0  152828  6248  2548 S  38580 1828 dan  25  1828 dan

```



Agenda

- What is Splunk App for Stream?
- Why use Stream?
- Where to use Stream?
- Deploying Stream
- Questions

```
020984 buff/cache 0 used, 615976 avail Mem
```

PID	PPID	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSER	RUID	GROUP
32616		0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	splunkd	splunkd	splunkd
32695		0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	tcpdump
590		0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	named	named	named
1602		0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	vim
1243			python													

What Is Splunk App for Stream?



```
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python
```

%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2880	splunk	2880	2880	2880	2880	2880	2880
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72	72	72	72	72
0.3	named	25	7:31.12	1.8	20	0	548272	18072	5584	S	1	25	named	25	25	25	25	25	25
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1028	dan	1028	1028	1028	1028	1028	1028

0 used, 620984 buff/cache 615976 avail Mem

Some history

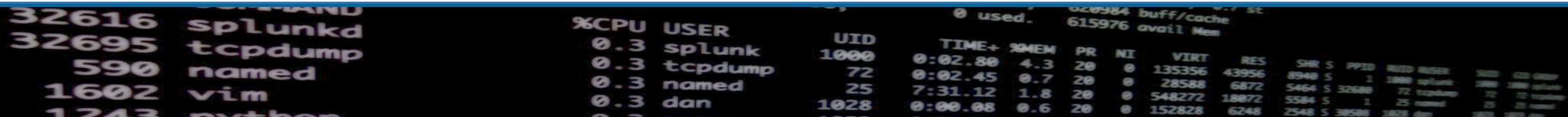
- Splunk acquires Cloudmeter, December 2013
- Renamed Splunk App for Stream
- Released with Splunk 6.0 (August, 2014)
- Now at version 6.4.3 (January, 2016)

```
0 used, 615976 avail Mem
```

	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PID	PPID	USER	MEM	MEM	MEM
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	2888	splunk	2888	2888	splunk
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	72	tcpdump	72	72	tcpdump
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	25	named	25	25	named
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	1602	vim	1602	1602	vim
1243	0.3	python																

Purpose of Stream

- Rapid deployment
- Rapid configuration
- Capture wire data
- Interpret wire data
- Summarize/filter/aggregate
- Index
- Kind of like Bro, but more Splunky, and GUI

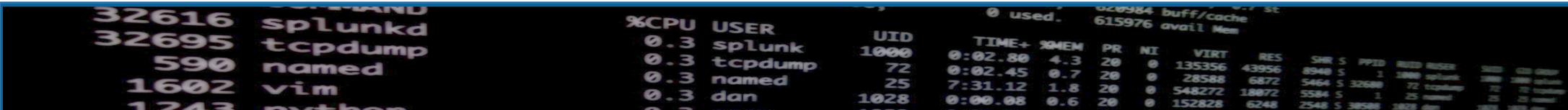


The image shows a terminal window with system statistics and a process list. The top part shows memory usage: 0 used, 615976 avail Mem. Below that is a table of processes with columns for %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, RES, SHR, S, PPID, and others. The processes listed are splunkd, tcpdump, named, vim, and python.

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	...
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	...
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	...
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	...
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	...

So what can we capture?

- Well, we aren't really capturing and indexing packets
- Forwarders capture packets, analyze the protocols
- What protocols (a lot):
 - TCP/UDP
 - Application protocols (HTTP, databases, email, file sharing, chat)
 - About 30 different protocols currently
 - <http://docs.splunk.com/Documentation/StreamApp/latest/DeployStreamApp/Whattypeofdatadoesthisappcollect>



```
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 other

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  INCR  OOM  MEM
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  1000  1000  1000  1000  1000
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  1  72  1000  1000  1000  1000  1000
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  1000  1000  1000  1000  1000
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  1  25  1000  1000  1000  1000  1000
```

Why to use Splunk Stream

```
0 used, 620984 buff/cache, 615976 avail Mem
```

PPID	PID	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	GROUP
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	splunk
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72 tcpdump
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18072	5584	S	1	named
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	vim
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	python

Cloud

- Many cloud services don't offer logs on things
- No chokepoints

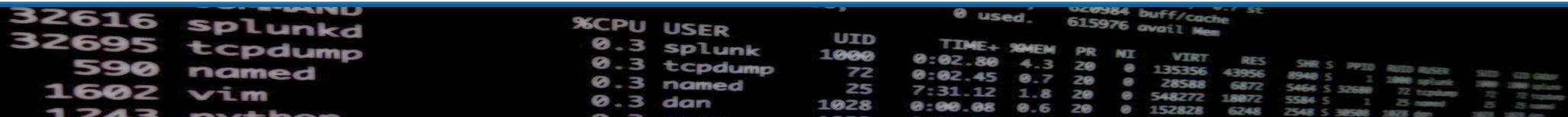


```
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSZ  DSIZ  CSIZ  DMAP
0 used, 615976 avail Mem
0.3 splunk 1000 0:02.80 4.3 20 0 135356 43956 8940 S 1 2880 splunk 2880 2880 splunk
0.3 tcpdump 72 0:02.45 0.7 20 0 28588 6872 5464 S 32680 72 tcpdump 72 72 tcpdump
0.3 named 25 7:31.12 1.8 20 0 548272 18872 5584 S 1 25 named 25 25 named
0.3 dan 1028 0:00.08 0.6 20 0 152828 6248 2548 S 38584 1828 python
```


Other features

- Filtering
- Aggregation
- Ephemeral Streams (short term)
- SSL decrypt
- Centralized management
- Integration with ES
 - Start a stream after Notable event
 - Protocol analysis dashboards

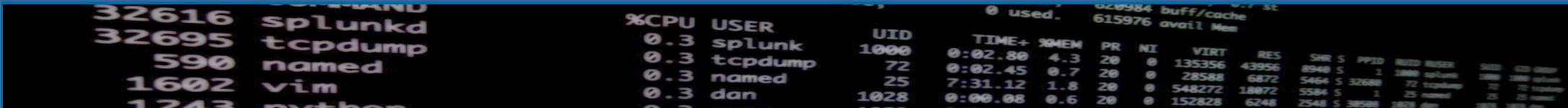
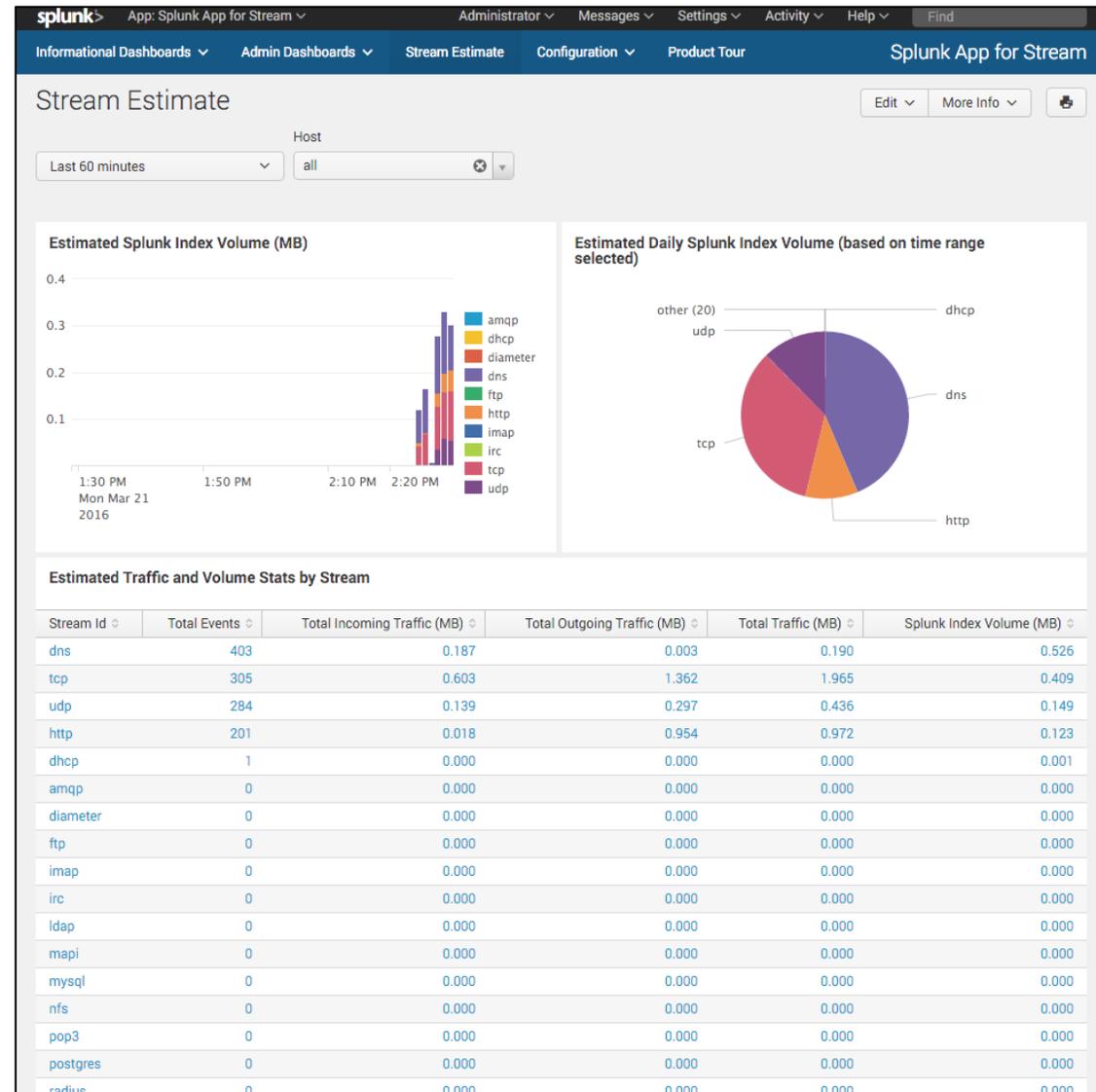


A terminal window showing system metrics and a process list. The top part shows memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below that is a table with columns: %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, RES, SHR, S, PPID, RUID, RUSER, SUID, GID, GROUP. The process list includes:

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUID	RUSER	SUID	GID	GROUP
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2888	splunk	2888	2888	splunk
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72	tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1028	dan	1028	1028	dan

Data Estimation

- “What if I turn this on?”
- Tells you how much data you would be indexing



Granular control of the data

- Not just which systems, but also what data, which fields

Configure Stream - http
HTTP Protocol Events

< Back to streams

Mode: Enabled Estimate Disabled

Splunk Index: default

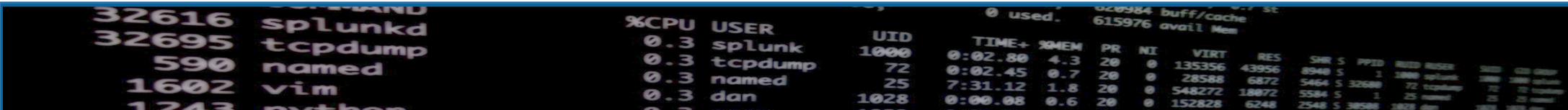
Protocol: HTTP

Aggregation: No Yes, every seconds

Filters: 0 filters configured [View Filters](#)

Fields:

Enable	Name	Description	Type	Term	Action
<input checked="" type="checkbox"/>	bytes	The total number of bytes transferred	Original	flow.bytes	▼
<input checked="" type="checkbox"/>	bytes_in	The number of bytes sent from client to server	Original	flow.cs-bytes	▼
<input checked="" type="checkbox"/>	bytes_out	The number of bytes sent from server to client	Original	flow.sc-bytes	▼
<input checked="" type="checkbox"/>	cookie	The Cookie HTTP request header	Original	http.cookie	▼
<input checked="" type="checkbox"/>	dest_ip	IP address of the server in dot-quad notation	Original	flow.s-ip	▼



Global Filters

- Filter out noise from the enterprise
- Things like vulnerability scanners

IP Address Filters

Use whitelist/blacklist filter rules to capture/ignore network data based on IP address. ?

Whitelist IP Addresses

Define a whitelist to capture data from IP addresses on that list only.

Blacklist IP Addresses

Define a blacklist to ignore those IP addresses, and allow data capture from all other IP addresses.

PID	USER	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	USER	MEM
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	3880	splunk	3880
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38580	1602	vim	1602

Distributed Forwarder Management

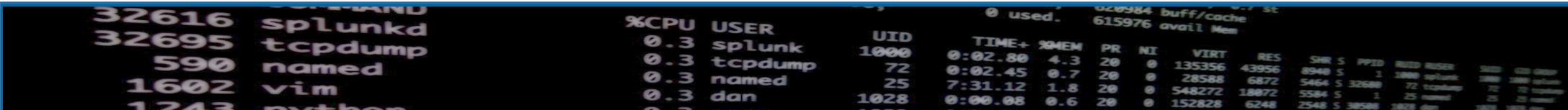
- Set up groups for capture
- Uses regex for groups on the “Forwarder ID”
- Forwarder ID is configurable via XML config file
- Yes, it’s another Splunk deployment/control mechanism

Distributed Forwarder Management

Create Stream Forwarder groups using pattern match.

2 groups

i	Name	Description	Rule	Include Ephemeral Streams?	Contains Streams	Actions
>	AppServers	These are my app servers that get fun stuff	^.*Pro.*\$	Yes	3	▼
>	defaultgroup	Used when there is no matching group found for a given stream forwarder ID		Yes	41	▼





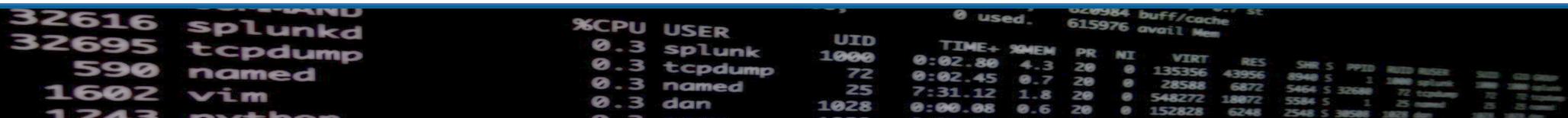
Where to use Splunk Stream

```
0 used, 628984 buff/cache, 615976 avail Mem
```

PPID	UID	%CPU	USER	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	INSTR	INSTR	INSTR	INSTR
32616	0	0.3	splunkd	0:02.80	4.3	20	0	135356	43956	8940	S	1	2888	tcpdump	1888	1888
32695	72	0.3	splunk	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72
590	25	0.3	named	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25
1602	1028	0.3	vim	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	1602	1602
1243	0	0.3	python													

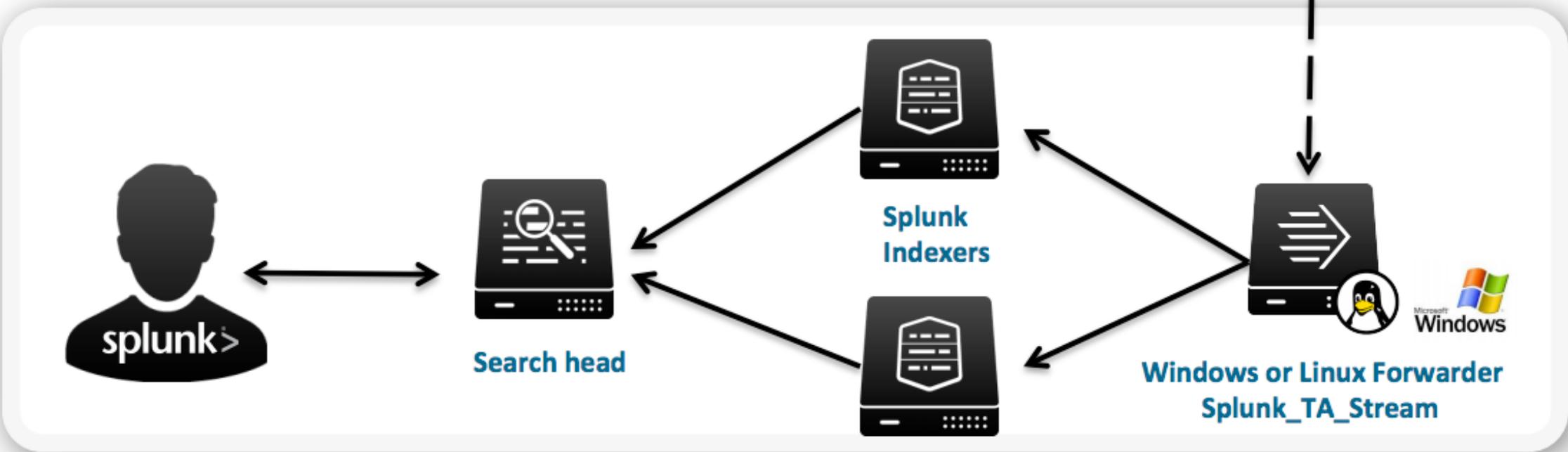
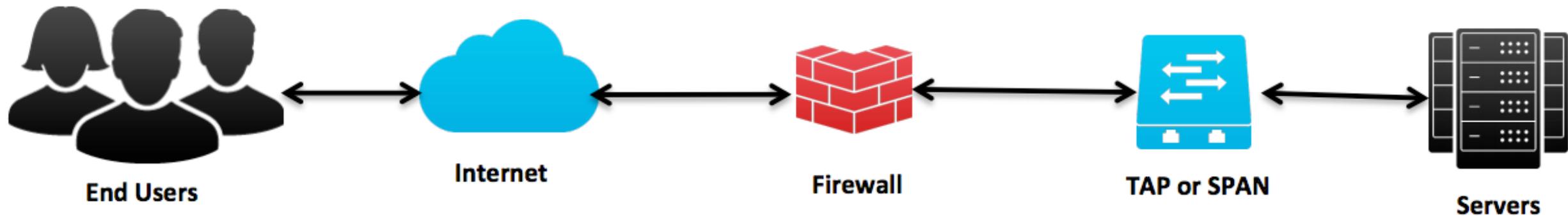
Dedicated Stream Forwarders

- Send data off of a switch Span or Tap
- Tools like Gigamon, Ixia, Etc.
 - You need these for really big pipes to spread the love
- Purpose built
 - Higher CPU and RAM
 - Better network cards
- Also a good option is you want to perform SSL decrypt
- Note that if you do this you will want to change some of your kernel settings (buffer sizes)
- Make sure to monitor your forwarders for thruput warnings!



A terminal window showing system statistics and a list of processes. The top part shows memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below that is a table of processes with columns for PID, USER, %CPU, and UID. The processes listed are splunkd, tcpdump, named, vim, and other.

PID	USER	%CPU	UID
32616	splunkd	0.3	1000
32695	tcpdump	0.3	72
590	named	0.3	25
1602	vim	0.3	1028
1243	other	0.3	0



```

32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

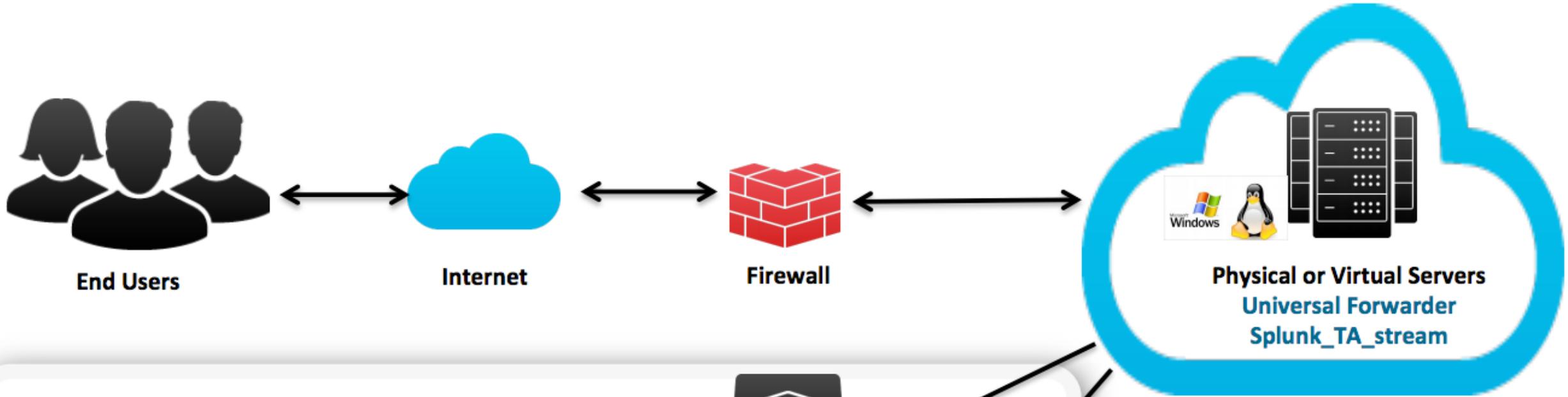
%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSSD  STATE
0.3  splunk  1000    0:02.80  4.3  20  0  135356  43956  8940  S  1  2880  splunk  2880  2880  splunk
0.3  tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  32689  72  tcpdump  72  72  tcpdump
0.3  named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  named    25  25  named
0.3  dan     1028    0:00.08  0.6  20  0  152828  6248  2548  S  38588  1828  dan     25  25  dan
  
```

Deploy to the Endpoints

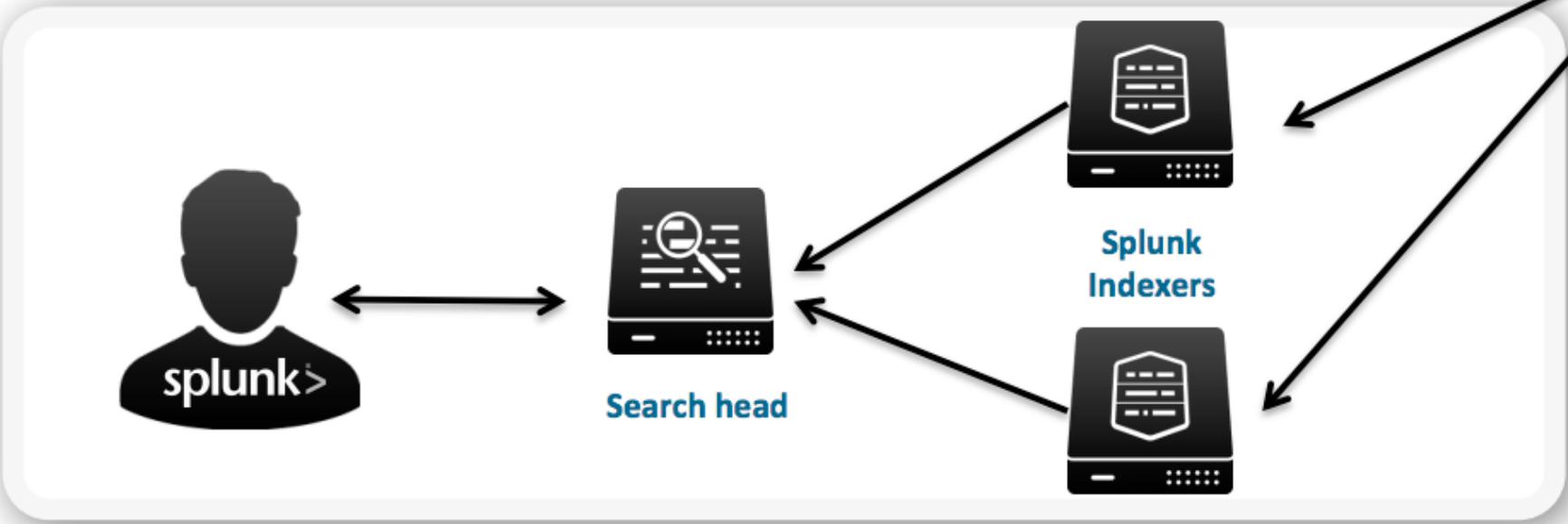
- Deploy directly to the systems you want to monitor
- Good for application debugging
- Nice option for Splunk ES
- Can be done from Deployment Server
- Granular control over groups
- Could mean a lot of “hand on”

```
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  CHILD  GROUP
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8940  S  1  2880  1000  1000  splunk
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  1  72  1000  1000  tcpdump
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  1000  1000  named
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  1  25  1000  1000  dan
```



- On premises
- Cloud



```

32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSZ  RSSM  RSSV  RSSP  RSSC  RSSD
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8940  S  1  2880  splunk  2880  2880  splunk
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  32680  72  tcpdump  72  72  tcpdump
0.3 named    25     7:31.12  1.8  20  0  548272  18072  5584  S  1  25  named  25  25  named
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  38580  1828  dan  25  25  dan
  
```

Deploying Splunk Stream

Schwinn

```
020984 buff/cache
615976 avail Mem
```

PPID	PID	UID	%CPU	USER	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	INSTR	INSTR	INSTR	INSTR	INSTR
32616	32616	1000	0.3	splunkd	0:02.80	4.3	20	0	135356	43956	8940	S	1	2880	aplunk	2880	2880	aplunk
32695	32695	72	0.3	tcpdump	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72	tcpdump
590	590	25	0.3	named	7:31.12	1.8	20	0	548272	18072	5584	S	1	25	named	25	25	named
1602	1602	1028	0.3	vim	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602	vim	1602	1602	vim
1243	1243		0.3	python														

Install the Splunk App for Stream

- Can co-locate with ES
- Can co-locate with DMC
- In smaller (less than 100 forwarders) don't use with the DS
 - Possible exhausted connections (DS and Stream poll separately)
- Installs just like any other Splunk app

```
020984 buff/cache
615976 avail Mem

COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  RMEM  TIO  TIO  TIO  TIO
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  1000  1000  1000  1000  1000  1000
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  1  72  1000  1000  1000  1000  1000  1000
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  1000  1000  1000  1000  1000  1000
0.3 dan      1028   0:00.08  0.6  20  0  152828  6248  2548  S  1  3858  1000  1000  1000  1000  1000  1000
```

Harvest the Add On

- Installs to a few places
- `$SPLUNK_HOME/etc/apps/Splunk_TA_stream`
- `$SPLUNK_HOME/etc/apps/splunk_app_stream/install/Splunk_TA_stream`
- `$SPLUNK_HOME/etc/deployment-apps/Splunk_TA_stream`
 - Will create the local `inputs.conf` with the app server location

```
Dauids-MacBook-Pro-2:local root# cat inputs.conf
[streamfwd://streamfwd]
splunk_stream_app_location = https://Dauids-MacBook-Pro-2.local:8000/en-us/custom/splunk_app_stream/
stream_forwarder_id =
disabled = 0
```

* Skip this is your SH is your DS

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	3888	splunk	3888	3888	3888	3888	
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	72	72	72	72	
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	25	25	
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	1602	1602	1602	1602	
1243	python	0.3	python	1000	0:00.00	0.0	20	0	152828	6248	2548	S	38588	1243	python	1243	1243	1243	1243	

Make sure your forwarders can talk back

- Your forwarders will need to be able to talk to the SH with splunk_app_stream installed
- The port is the same as the GUI for your SH

```
0 used, 615976 avail Mem
%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  CHILD  GROUP
32616 splunkd   1000     0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  2888  splunk  2888  2888  splunk
32695 tcpdump    72      0:02.45  0.7  20  0  28588  6872  5464  S  1  72  12288  72  72  tapdev
590  named     25      7:31.12  1.8  20  0  548272  18872  5584  S  1  25  25  named  25  25  named
1602 vim       1028    0:00.08  0.6  20  0  152828  6248  2548  S  1  25  25  vim    25  25  vim
1243 python
```

Configure your forwarders

- Don't have to be root on Linux
 - Use the included setuid.sh script
- Must be local admin or local system on Windows
- On UFs you should monitor your thruput limits

```
0 used, 615976 avail Mem
```

PPID	PID	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	NAME	MEM	MEM	MEM
32616	32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	1000	splunkd	1000	1000	splunkd
32695	32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	1	72	tcpdump	72	72	tcpdump
590	590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named
1602	1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	1	1028	vim	1028	1028	vim
1243	1243	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	1	1028	python	1028	1028	python

Inputs.conf

- Remember that the inputs.conf is layerable
- Just like other Splunk configs
- Doesn't have to be in the Splunk_TA_stream
- On the DS you can deploy two apps, one with the input to point back to the splunk_app_stream
- Then also deploy the Splunk_TA_stream

```
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSZ  RSSM  RSSV  RSSP  RSSC  RSSD  RSSZ  RSSM  RSSV  RSSP  RSSC
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  splunk  1000  1000  splunk
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  32688  72  tcpdump  72  72  tcpdump
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  named  25  25  named
0.3 dan     1028    0:00.08  0.6  20  0  152828  6248  2548  S  38588  1828  dan  38588  1828  dan
```

Configure your streams

- The defaults may send more fields than you need
- Can tell forwarders which parts of the data you want
- You can have different configs for different groups!

Configure Stream - http

HTTP Protocol Events

[Clone](#) [Delete](#) [Cancel](#) [Save](#)

[< Back to streams](#)

Mode: Enabled Estimate Disabled

Splunk Index:

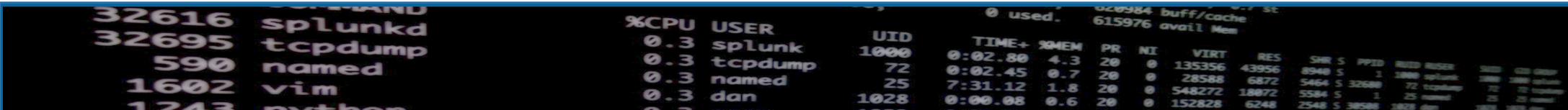
Protocol: HTTP

Aggregation: No Yes, every seconds

Filters: 0 filters configured [View Filters](#)

Fields

Enable	Name	Description	Type	Term	Action
<input checked="" type="checkbox"/>	bytes	The total number of bytes transferred	Original	flow.bytes	▼
<input checked="" type="checkbox"/>	bytes_in	The number of bytes sent from client to server	Original	flow.cs-bytes	▼
<input checked="" type="checkbox"/>	bytes_out	The number of bytes sent from server to client	Original	flow.sc-bytes	▼
<input checked="" type="checkbox"/>	cookie	The Cookie HTTP request header	Original	http.cookie	▼



Configure your forwarder groups

- Uses good ol' regex
- Lets you say ahead of time if Ephemeral Streams should be allowed

Create New Forwarder Group

Name:

Description:

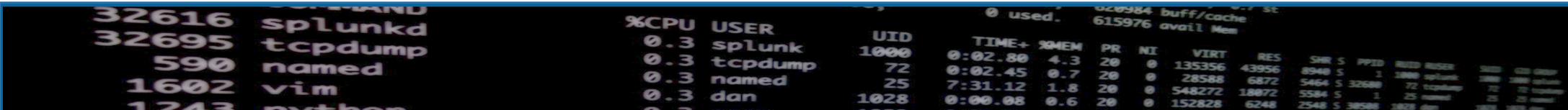
Include Ephemeral Streams?

Matched Forwarders (New Rule)

Regex Rule:

Preview of Matched Forwarders: No matches found.

Forwarder ID	Last Known Event
--------------	------------------



Gotcha with Groups

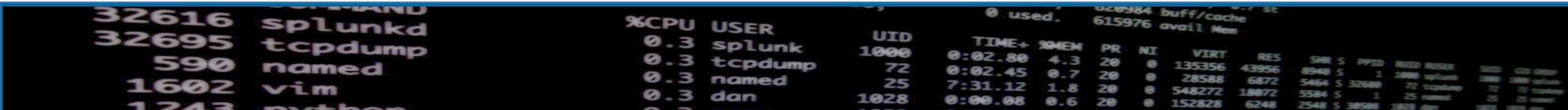
- Just regex on the Stream forwarder ID (not IP, hostname)
- This is configured in an XML file
- Messy
- The “defaultgroup” forwarder group for all unmatched hosts will gather ALL THE THINGS

Distributed Forwarder Management Create New Group

Create Stream Forwarder groups using pattern match.

3 groups

i	Name	Description	Rule	Include Ephemeral Streams?	Contains Streams	Actions
>	AppServers	These are my app servers that get fun stuff	^.*Pro.*\$	Yes	3	⌵
>	DBServer	Database servers in the App group	appdbhost[\d]+	No	0	⌵
>	defaultgroup	Used when there is no matching group found for a given stream forwarder ID		Yes	41	⌵



Wait for data to flow in

- That's pretty much it!
- Docs make it look a lot harder

```
020984 buff/cache 0 used, 615976 avail Mem
```

COMMAND	%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSER	RUID	GROUP
32616 splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	splunk	splunk	splunk
32695 tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	tcpdump
590 named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	named	named	named
1602 vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30588	1028	dan	dan
1243 python	0.3														

Questions?

```
0 used, size 984 buff/cache 615976 avail Mem
```

PPID	PID	UID	%CPU	USER	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	NAME
32616	32616	1000	0.3	splunkd	0:02.80	4.3	20	0	135356	43956	8940	S	1	splunkd
32695	32695	72	0.3	tcpdump	0:02.45	0.7	20	0	28588	6872	5464	S	32680	tcpdump
590	590	25	0.3	named	7:31.12	1.8	20	0	548272	18072	5584	S	1	named
1602	1602	1028	0.3	vim	0:00.08	0.6	20	0	152828	6248	2548	S	30580	vim
1243	1243	0	0.3	python								S		python

Credits

- Thanks to the Baltimore Area Splunk User Group
- Cover Slide: Upper Swallow Falls in Oakland, MD, Chris Flees, <http://fineartamerica.com/profiles/chris-flees.html?tab=artwork&page=7>
- Slide 3: Potomac River in Maryland, Terry J. Adams, <http://www.fhwa.dot.gov/byways/byways/60807/photos>
- Slide 7: Timanus Mill on the Jones Falls in Baltimore, “Monument City”, <http://www.panoramio.com/photo/57148558>
- Slide 8: “Missing Homework Log” by “Red Beetle RB”. <https://www.teacherspayteachers.com/Product/Missing-Homework-Log-4112>
- Slide 9: Rotton log, National Wildlife Foundation, <https://www.nwf.org/kids/family-fun/outdoor-activities/investigate-a-rotten-log.aspx>
- Slide 10: The Simpsons, <http://i.imgur.com/91sn32Q.jpg?fb>
- Slide 11: Bro Network Security Monitor, <https://www.bro.org/>
- Slide 17: Ian Adams Photography, <http://ianadamsphotography.com/news/galleries/bridges/>
- Slides 19 and 21: Splunk Conf 2015, “Splunk App for Stream Deployments in the Real World: Enhance Operational Intelligence Across Application Delivery, IT Ops, Security and More”, http://conf.splunk.com/session/2015/conf2015_SUdovicic_CChing_MDickey_Splunk_SplunkEntWhatsNew_StreamDeploymentsInTheReal.pdf
- Slide 22: Gunpowder Falls in Baltimore County, MD, <http://hdrcreme.com/photos/1818-gunpowder-falls>
- Slide 23: Splunk Docs, <http://docs.splunk.com/Documentation/StreamApp/latest/DeployStreamApp/DeploymentArchitecture>
- Slide 34: Youghiogheny River at Friendsville, MD by Joe Dawson, <https://www.flickr.com/photos/jmd41280/5066756138>

