

Stream + NetFlow for Security & Insider Threat Detection

Kelly Feagans | Sales Engineering



Objective

OBJECTIVE

Learn how to use NetFlow with the Splunk Stream Forwarder for outlier detection

USE CASES

- NetFlow: Endpoint outlier detection (overall) and insider threat
- NetFlow: Anomalous connections by host
- NetFlow: Possible data exfiltration
- NetFlow: Extra-Credit Adding proxy data in the mix!

BENEFITS

 Identify anomalies and outliers respective to network usage by time of day, port, bytes sent/received, geographic location, or by functional group.



NetFlow for Outlier (Insider Threat) Detection

NetFlow, when used in conjunction with Splunk Stream, is an effective and rather simple solution to capture wire data moving across an environment.

SPLUNK STREAM USED AS A FLOW COLLECTOR

- Use one or many Stream Forwarder(s) as needed

FAST AND SIMPLE SETUP

- Configure flow data ingestion (streamfwd.conf)
- Configure forwarding of flows to Stream Forwarder
- Supported Flow Protocols: NetFlow v5, v9, IPFIX; sFlow v5, jFlow

FIND OUTLIERS, ANOMALOUS CONNECTIONS, DATA EXFIL, ETC

product.screen?product_1d=FL-DSH-01&JSESSIONID=SD

Start searching!







Use Case 1 : Find the Outlier (by bytes & port)

Kelly Feagans | Sales Engineering



Use Case 1: Find the Outlier by Port

OBJECTIVE

• Use NetFlow data to find the outlier by port

USE CASE

- User activities tracking by port (80)
- Use an extreme number (standard deviation) above the mean (like 10x)

BENEFITS

- Quickly visualize the outlier(s) (different than all the rest) that are communicating on port 80
- View how tightly the outlier(s) are grouped by time

Junk>	мрр. сопт2017	vernow natios					Adminisi	irator • Wes	sages v .	Settings + Activ	ny ne	Fillu	
arch U	Jse Cases 🗸	Reports I	Dashboards								con	f2017 NetF	low Hands-
se Ca	se 1: Fin	d the Ou	utlier by F	Port (80))							Edit	Export 🗸 🗌
This sear	ch evaluates	NetFlow da	ata for the fo	ollowing ma	tches:		Search S	String - Find	the Outlie	r			
1. Destinat 2. Source I 3. Data Flo 4. Destinat 5. Standar Benefits: 1. Easily vi 2. View nu 3. (Second	tion IP is NOT int IP is internal w direction is "e tion Port is 80 d Devation is cal iew outliers and imber of outliers d Panel) View uni	ernal gress" culated at TEN now far away fr and how tightly que outliers by	times above the rom the average y grouped (by tin SRC_IP	average they land re)			index=ne 5/2017:1 search sort 0 events eval A eval 1 eval i table	ttflow source 2:00:00 (dest_ip!=10)_time ttats first(b) VG=round(AVG owerBound=(A sOutlier=if(_time bytes_	type="strea .* src_ip=1 ytes_in) as ,2), STDEV= VG-STDEV), bytes_in < in lowerBou	m:netflow" earli 0.* flow_dir=egr AVG stdev(bytes round(STDEV,2) upperBound=(AVG+ lowerBound OR by nd upperBound	est=09/05 ess dest_ in) as S (10*STDEV tes_in >)	/2017:08:00:00 port=80 TDEV)) upperBound, 1,	latest=09/0
Port 80 O	utliers Foun	d											
	6M												
	5M				•								
	4M												
	3М												
Series 2	2M												12 outlie
	IM												
	0	L		ledwood	- I - I - I - I - I	Landed there	umpha.	madadad	da na 🕠	المراجع المرجع	March	M	<u>مىلە</u>
	-1M 08:00	08:15	08:30	08:45	09:00	09:15	09:30	09:45	10:00	10:15	10:30	10:45	
Search St	tring - Uniqu	e Outliers by	y SRC_IP				Unique (Outliers by S	SRC_IP				
index=net	flow sourcety	pe="stream:ne	etflow" earli	est=09/05/201	7:08:00:00	latest=09/0	_time ≎			src_ip ≎		bytes_in 0	upperBound
5/2017:12	:00:00 dest_in!=10.*	src ip=10.*	flow dir=egr	ess dest nort	=80		2017-09-05	5 08:54:23		10.242.4.117		5062056	34302
sort 0	_time		on_or -egre	in action point			2017-09-05	511:13:50		10.247.24.171		2695880	343028
eventst eval AV	ats first(byt G=round(AVG,2	<pre>>s_in) as AVG), STDEV=rour</pre>	<pre>stdev(bytes_ nd(STDEV,2)</pre>	_in) as STDEV			2017-09-05	511:31:58		10.235.22.22		847107	343028
eval lo	werBound=0, u	perBound=(A	VG+(10*STDEV)) tes in > unno	rBound 1		2017-09-05	5 09:13:29		10.226.200.12		814936	343028
	outries - ri(by	103_11 × 10M6	an adding on Dyl	cos_m > uppe	a sound, 1,	· /	2017-00-06	: 00.22.27		10.226.7.137		755612	
where i	sOutlier=1						2017-09-00	00.23.31				100012	343028
where i table _	sOutlier=1 time src_ip b rc_ip	/tes_in upper	rBound				2017-09-05	5 08:42:55		10.244.3.7		595352	343028 343028





Use Case 1: Outlier - SPL

	App: conf201	7 NetFlow Har	ids 🗸				Administra	ator 🗸 Mess	sages 🗸 🛛 S	Settings 🗸 🛛 Act	ivity∨ He	elp 🗸 🛛 Find	
earch U	Use Cases 🗸	Reports	Dashboards									nf2017 NetF	low Hands-Or
Jse Ca	se 1: Fir	nd the C	outlier by l	Port (80))							Edit	Export 🗸
This sear	rch evaluate	es NetFlow	data for the f	ollowing ma	tches:		Search St	ring - Find t	the Outlie	r			
1. Destinat 2. Source I 3. Data Flo 4. Destinat 5. Standard Benefits: 1. Easily vi 2. View nu 3. (Second	tion IP is NOT i IP is internal ow direction is ' tion Port is 80 rd Devation is c iew outliers anu umber of outlier d Panel) View u	nternal "egress" alculated at TI d how far awa; 's and how tig! nique outliers	IN times above the from the average tily grouped (by tir by SRC_IP	e average they land ne)			index=net 5/2017:12 search e svert or eventst eval AV eval low eval is table _	flow sourcety :00:00 dest_ip!=10. _time ats first(by G=round(AVG, werBound=(AVV Dutlier=if(b) time bytes_in	ype="stream * src_ip=1(tes_in) as 2), STDEV= G-STDEV), t ytes_in < 1 n lowerBour	a:netflow" ear.).* flow_dir=e; AVG stdev(byt round(STDEV,2) upperBound=(AV) lowerBound OR d upperBound	liest=09/05 gress dest_ es_in) as 5 G+(10*STDE\ bytes_in >	5/2017:08:00:00 _port=80 STDEV /)) upperBound, 1,	latest=09/0
Port 80 O Series 2	Dutliers Fou 6M 5M 5M 4M 3M 2M 1M 0	nd	Ĵ				the abolist	n, blowed all	de ut			Mar and a standard	12 outliers
	-1M 08:00	08:15	08:30	08:45	09:00	09:15	09:30	09:45	10:00	10:15	10:30	10:45	
Search St	-1M 08:00	08:15 ue Outliers	08:30	08:45	09:00	09:15	09:30 Unique O	09:45 utliers by Sl	10:00	10:15	10:30	10:45	
Search St index=net	-1M 08:00 tring - Uniqu	08:15 ue Outliers :ype="stream	08:30 by SRC_IP	08:45 est=09/05/201	09:00 7:08:00:00 la	09:15	09:30 Unique Or	09:45 utliers by Sl	10:00 RC_IP	10:15 src_jp 0	10:30	10:45 bytes_in ≎	upperBound 0
Search St index=net: 5/2017:12 L search	-1M 08:00 tring - Uniquest 2:00:00 dest ip!=10	08:15 ue Outliers sype="stream * src ip=10	08:30 by SRC_IP :netflow" earlie * flow dir=egr	08:45 est=09/05/201 ess dest port	09:00 7:08:00:00 la =80	09:15 test=09/0	09:30 Unique O time 0 2017-09-05 0	09:45 utliers by SI 18:54:23	10:00	10:15 src_ip 0 10.242.4.117	10:30	10:45 bytes_in ≎ 5062056	upperBound 0 343028.5
Search St index=net: 5/2017:12 search 0 sort 0	-1M 08:00 tring - Unique tflow sourcet 2:00:00 dest_ip!=10. _time	08:15 ue Outliers ype="stream * src_ip=10	08:30 by SRC_IP netflow" earlin * flow_dir=egro	08:45 est=09/05/201 ess dest_port	09:00 7:08:00:00 la =80	09:15	09:30 Unique Ou _time 0 2017-09-05 0 2017-09-05 1	09:45 utliers by Sl 18:54:23 1:13:50	10:00	10:15 src_ip ≎ 10:242.4.117 10:247.24.171	10:30	10:45 bytes_in © 5062056 2695880	upperBound ≎ 343028.5 343028.5
Search St index=net 5/2017:12 search (sort 0 eventst eventst	-1M 08:00 tring - Uniquest 2:00:00 dest_ip!=10. _time tats first(by Geround(AVG.	08:15 ue Outliers :ype="stream * src_ip=10 rtes_in) as , 2), STDEV=r	08:30 by SRC_IP :netflow" earlin * flow_dir=egro VG stdev(bytes, und(STDEV,2)	08:45 est=09/05/201 ess dest_port _in) as STDEV	09:00 7:08:00:00 la =80	09:15	09:30 Unique Ou _time © 2017-09-05 0 2017-09-05 1 2017-09-05 1	09:45 utliers by Sl 1:13:50 1:31:58	10:00	10:15 src_jp 0 10.242.4.117 10.247.24.171 10.235.22.22	10:30	10:45	upperBound 343028.5 343028.5 343028.5
Search SI index=net 5/2017:12 search (svort 0 eventsXi eval AV(eval 100	-1M 08:00 tring - Uniq cflow sourcet 2:00:00 dest_ip!=10. _time tats first(by %ereBund0,	08:15 ue Outliers ype="stream * src_ip=10 rtes_in) as 2), STDEV=r upperBound=	08:30 by SRC_IP :netflow" earli * flow_dir=egr. WG stdev(bytes, und(STBEV,2) (AVG+(10*STDEV))	08:45 est=09/05/201 ess dest_port _in) as STDEV)	09:00 7:08:00:00 la =80	09:15	09:30 Unique OU _time © 2017-09-05 0 2017-09-05 1 2017-09-05 1 2017-09-05 0	09:45 utliers by Sl 8:54:23 1:13:50 1:31:58 19:13:29	10:00	10:15 src_jp 0 10.242.4.117 10.247.24.171 10.235.22.22 10.226.200.12	10:30	10:45 bytes_in 0 5062056 2695880 847107 814936	upperBound © 343028.5 343028.5 343028.5 343028.5
Search SI index=net 5/2017:12 search 0 sventst: eventst: eval AV eval loi eval isi where i:	-1M 08:00 tring - Unique tflow sourcet 2:00:00 dest_ip!=10. time tats first(by WerBound=0, SOutlier=if(b SOutlier=1	08:15 ue Outliers type="stream * src_ip=10 rtes_in) as 2), STDEV=r upperBound= bytes_in < 10	08:30 by SRC_IP :netflow" earlii * flow_dir=egr. WG stdev(bytes, und(STDEV,2) (WG+C10*TDEV,2) werBound OR by	08:45 est=09/05/201 ess dest_port _in) as STDEV) tes_in > uppe	09:00 7:08:00:00 la =80 , rBound, 1, 0)	09:15	09:30 Unique O _time © 2017-09-05 0 2017-09-05 0 2017-09-05 0 2017-09-05 0	09:45 utliers by Sl 8:54:23 1:13:50 1:31:58 19:13:29 18:23:37	10:00 RC_IP	10:15 src_jp 0 10.242.4.117 10.247.24.171 10.235.22.22 10.226.200.12 10.226.7.137	10:30	10:45 bytes_in © 5062056 2695880 847107 814936 755612	upperBound 0 343028.5 343028.5 343028.5 343028.5 343028.5
Search Si index=net 5/2017:12 i search sort 0 . eventst: eval Vi eval loi eval isi where i: table _ dedue a:	-1M 08:00 tring - Unique tflow sourcet 2:00:00 dest_ip!=10. time tats first(by WerBound=0, SOutlier=if(b) tioutlier=1 time src_ip	08:15 ue Outliers sype="stream * src_ip=10 rtes_in) as i 2), STDEV=r upperBound= sytes_in < 1: bytes_in up;	08:30 by SRC_IP :netflow" earli * flow_dir=egr. VKG stdev(bytes, sund(STDEV,2) (XVG-(10*STDEV,2) wwerBound OR by >erBound	08:45 est=09/05/201 ess dest_port _in) as STDEV) tes_in > uppe	09:00 7:08:00:00 la =80 , rBound, 1, 0)	09:15	09:30 Unique O _time © 2017-09-05 0 2017-09-05 0 2017-09-05 0 2017-09-05 0 2017-09-05 0	09:45 utliers by Sl 8:54:23 1:13:50 1:31:58 19:13:29 18:23:37 18:42:55	10:00 RC_IP	10:15 src_jp 0 10.242.4.117 10.235.22.22 10.226.200.12 10.226.7.137 10.244.3.7	10:30	10:45 bytes_in © 5062056 2695880 847107 843436 755612 595352	upperBound © 343028 5 343028 5 343028 5 343028 5 343028 5 343028 5

```
index=netflow sourcetype="stream:netflow"
earliest=09/05/2017:08:00:00
latest=09/05/2017:17:00:00 src ip=10.232.117.*
 bucket time span=1h@h
 stats dc(dest ip) as count by src ip, time
 eventstats max( time) as maxtime
 stats count as num data samples max(eval(if( time
>= relative time(maxtime, "-1h@h"), 'count',null)))
as "count" avg(eval(if( time<="" or="" 'count'="">
upperBound) AND num data samples >=7, "YES", "NO")
 eval vs="current vs last hour"
 eval howFarAway=count/avg
 table src ip, vs, isOutlier, count, avg,
howFarAway, lowerBound, upperBound
 where isOutlier="YES"
 sort - howFarAway
```



STEP BY STEP GUIDE – Use Case 1

Start with a look at the first search - Find the Outlier:

- Hover over the panel, and click on the magnifying glass icon to launch search
- How many Stats rows do you see? (Statistics Tab)
- Change flow_dir from "egress" to "ingress"
 - How many outliers do you see?

For the second search – Unique Outliers:

- Hover over the panel, and click on the magnifying glass icon to launch search
- Change flow_dir from "egress" to "ingress"
- What SRC_IP do you see as "the" outlier?





Use Case 2 : Anomalous Connections by Host

Kelly Feagans | Sales Engineering



Use Case 2: Anomalous Connections by Host

OBJECTIVE

• Learn how to use NetFlow data to evaluate "anomalous connections by host"

USE CASE

• Find hosts that have deviated in the count of connections, as compared to last hour

BENEFITS

 Identify hosts that could be abnormally affected by Malware, etc.

plunk > A	pp: conf2017 NetFlow H ~ se Cases ~ Reports	Dashboards	Search	Adı	ministrator Ƴ	Messages 🗸 Settings 🗸	Activity Help Conf2017	Find NetFlow Hands-O		
Jse Case	e 2: Anomalou	s Connec	tions by Host	t			Ed	it Export ~		
 Source IP is in the 10.232.117.* range Calculates last hour of connections (dest_jp) by src_jp Calculates standard deviation for last hour (connections) Calculates "lower bounds" and "upper bounds" Calculates "isOutlier" if upper or lower bounds are exceeded Benefits: Find hosts that are deviating by numbers of connections when compared to last timeframe View the "count" as how far "above or below" the bounds the host has been 					<pre>Search String - Anomalous Connections by Host index=netflow sourcetype="stream:netflow" earliest=09/05/2017:08:00:00 latest= 09/05/2017:17:00:00 src_ip=10.232.117.* bucket _time span=1heh stats dc(dest_ip) as count by src_ip, _time eventstats max(_time) as maxtime stats count as num_data_samples max(eval(if(_time >= relative_time(maxtime, "-1heh"), 'count',null))) as "count" avg(eval(if(_time<relative_time(maxtime, "-1heh"), 'count',null))) as avg stdev(eval(if(_time<relative_time(maxtime,"- 1heh"), 'count',null))) as stdev by "src_ip" eval avg=round(avg,2), lowerBound=round((avg-stdev*2),2), upperBound=round ((avg=stdev*2),2) eval isOutlier=if(('count' < lowerBound OR 'count' > upperBound) AND num_dat a_samples >=7, "YES", "NO") eval nowFarAway=count/avg table src_ip, vs, isOutlier, count, avg, howFarAway, lowerBound, upperBound where isOutlier="YES" sort - howFarAway</relative_time(maxtime,"- </relative_time(maxtime, </pre>					
src_ip ≎	vs ≎		isOutlier 0	count ≎	avg ≎	howFarAway ≎	lowerBound 🗘	upperBound ©		
10.232.117.90	current vs la	st hour	YES	91	59.29	1.535	31.54	87.04		
10.232.117.98	current vs la	st hour	YES	84	59.14	1.420	41.96	76.33		
10.232.117.91	current vs la	st hour	YES	76	54.86	1.385	35.11	74.6		
						1 1005				
10.232.117.142	2 current vs la	st hour	YES	125	110.57	1.1305	100.38	120.7		

Note: "Anomalous Connections by Host" is borrowed from a search in the Splunk Security Essentials App. Thank you David Veuve!!!



Use Case 2: Anomalous Connections - SPL

	<pre> stats count a e, "-1h@h"), 'c m_data_samples eval vs="curr eval howFarAw table src_ip, d where isOutli sort - howFar</pre>	<pre>:ount',null))) as "count >=7, "YES", "NO") ent vs last hour" /ay=count/avg vs, isOutlier, count, .er="YES" Away</pre>	avg, howFarAway, lowerf	Bound, upperBoun
0	avg ≎	howFarAway ≎	lowerBound 🗘	upperBound 0
0	avg ≎ 59.29	howFarAway ≎ 1.535	lowerBound ≎ 31.54	upperBound 87.04
¢ 1 4	avg ≎ 59.29 59.14	howFarAway ≎ 1.535 1.420	lowerBound ≎ 31.54 41.96	upperBound ≎ 87.04 76.32
¢ 1 4	avg ≎ 59.29 59.14	howFarAway ≎ 1.535 1.420	lowerBound ≎ 31.54 41.96 35.11	upperBound ≎ 87.04 76.32 74.61

1.1193

96.04

121.96

109.00

```
index=netflow sourcetype="stream:netflow"
earliest=09/05/2017:08:00:00 latest=09/05/2017:17:00:00
src ip=10.232.117.*
 bucket time span=1h@h
  stats dc(dest ip) as count by src ip, time
  eventstats max( time) as maxtime
 stats count as num data samples max(eval(if( time >=
relative time(maxtime, "-1h@h"), 'count', null))) as "count"
avg(eval(if( time<relative time(maxtime,"-</pre>
1h@h"), 'count', null))) as avg
stdev(eval(if( time<relative time(maxtime,"-</pre>
1h@h"), 'count', null))) as stdev by "src ip"
 eval avg=round(avg,2), lowerBound=round((avg-stdev*2),2),
upperBound=round((avg+stdev*2),2)
 eval isOutlier=if(('count' < lowerBound OR 'count' >
upperBound) AND num data samples >=7, "YES", "NO")
  eval vs="current vs last hour"
  eval howFarAway=count/avg
  table src ip, vs, isOutlier, count, avg, howFarAway,
lowerBound, upperBound
 where isOutlier="YES"
  sort - howFarAway
```



Start with a look at the search "Anomalous Connections by Host":

- Hover over the panel, and click on the magnifying glass icon to launch search
- How many Stats rows do you see out of how many events returned?
- Try changing the bounds (avg+/stdev*2) to a 4
 - How many outliers do you see now?
- What do you think this could tell you about hosts in your environment?





Use Case 3 : Possible Data Exfiltration

Kelly Feagans | Sales Engineering

Use Case 3: Possible Data Exfiltration

OBJECTIVE

 Gain an understanding of which hosts internal to an environment are sending a large amount of data outbound in "single" flows.

USE CASE

• Find internal hosts that are not permitted to send data outside their network, and/or to prohibited sites.

BENEFITS

• List sources that are communicating (large byte counts, single flow) with non-U.S. endpoints.

splunk> App: conf2017 NetFlo ~	Administrator V Messages V Settings V Activity V Help V Find
Search Use Cases ~ Reports Dashboards	conf2017 NetFlow Hands-On
Use Case 3: Possible Data Exfiltration	Edit Export ~
 This search evaluates NetFlow data for the following matches: 1. Source IP is in the internal range (10.* network) 2. Destination IP is NOT in the internal range (10.* network) 3. Flow Direction is set to "egress" 4. Country is derived from src.jp 5. This search is looking for "individual flows" that are associated with 2MB or greater 6. The bottom "Missile Map" chart is a visual of the stats table above it Benefits: 1. List sources that are communicating with foreign (non U.S.) countries, that are sending large amounts of data in a single flow 2. This statistics table could be set as a scheduled search/alert for notification 	<pre>Search String - Possible Data Exfiltration to Non-U.S. Locations index="netflow" sourcetype="stream:netflow" flow_dir=egress dest_ip!=10.* by tes_in>=2000000 earliest=09/04/2017:16:00:00 latest=09/04/2017:16:30:00 eval src_ip = if(cidrmatch("10.0.0.0/8",src_ip),"71.56.239.115",src_ip) iplocation dest_ip prefix=end_ iplocation src_ip prefix=start_ eval color="#FF0000" where end_Country!="United States" where mb > 2 table _time,src_ip,dest_ip,mb,dest_port,end_Country</pre>
Secure hosts sending too much data outside - Possible Data Exfiltrati	ion - Single Flow > 2MB (egress)

_time \$	src_ip ≎	dest_ip ≎	mb ≎	dest_port 0	end_Country 0
2017-09-04 16:19:26	71.56.239.115	87.121.161.3	2.29	4000	Bulgaria
2017-09-04 16:19:26	71.56.239.115	162.105.131.198	187.86	80	China





splunk> .conf2017

Use Case 3: Possible Data Exfiltration - SPL

```
index="netflow" sourcetype="stream:netflow" flow dir=eqress dest ip!=10.*
bytes in>=2000000 earliest=09/04/2017:16:00:00 latest=09/04/2017:16:30:00
  eval mb=round(bytes in/1024/1024,2)
  eval src_ip = if(cidrmatch("10.0.0/8", src_ip), "71.56.239.115", src ip)
  iplocation dest ip prefix=end
  iplocation src ip prefix=start
                                                                                                                        Find
                                                                               Administrator 🗸
                                                                                          Messages v
                                                                                                  Settinas ~
                                                                                                          Activity 🗸
                                                                                                                 Help ~
  eval color="#FF0000"
                                                                                                                conf2017 NetFlow Hands-On
  where end Country!="United States"
  where mb > 2
                                                                                                                           Export ~
                                                                                                                      Edit
  table time, src ip, dest ip, mb, dest port, end Country
                                                                                 Search String - Possible Data Exfiltration to Non-U.S. Locations
                                                                                 index="netflow" sourcetype="stream:netflow" flow_dir=egress dest_ip!=10.* byte
                                                                                  s_in>=2000000 earliest=09/04/2017:16:00:00 latest=09/04/2017:16:30:00
                                                                                   eval mb=round(bytes_in/1024/1024,2)
```



eval src_ip = if(cidrmatch("10.0.0.0/8",src_ip),"71.56.239.115",src_ip)

iplocation dest_ip prefix=end_
iplocation src_ip prefix=start_

where end_Country!="United States"

table _time,src_ip,dest_ip,mb,dest_port,end_Country

eval color="#FF0000"

where mb > 2

STEP BY STEP GUIDE – Use Case 3

Start with a look at the search "Possible Data Exfiltration":

- Hover over the panel, and click on the magnifying glass icon to launch search
- How many Stats rows do you see returned?
- Try remove the 'where' command "end_Country!=United States"
 - What do you see now?
- Take a look at the second panel ... "Missile Map":
 - Hover over the panel, and click on the magnifying glass icon to launch search
 - Try remove the 'where' command "end_Country!=United States"
 - What do you see now when clicking on the "Visualization Tab"?





Extra Credit: NetFlow + Proxy

Kelly Feagans | Sales Engineering

Extra Credit: Proxy + NetFlow

OBJECTIVE

 Gain an understanding of hosts that are seen in both NetFlow and Proxy data (cross-reference)

USE CASE

• Find internal hosts that are outside of security policy (communicating with forbidden hosts or sites)

BENEFITS

 Search through hundreds to millions of records to find hosts that fall out of policy

splunk>	App: conf2017	NetF 🗸			Administrator 🗸	Messages 🗸	Settings 🗸	Activity ~	Help 🗸	Find	
Overview	Use Cases 🗸	Reports	Dashboards	Search					conf201	7 NetFlow H	lands-Or
Extra (Credit: Pro	xy + Ne	etFlow						E	dit Export	•
This se	arch evaluates	IP addres	ses that are f	ound in both NetFlo	w and Proxy inc	lexes:					
1. Destir 2. Sourc 3. Data I 4. Destir 5. Stand	nation IP is NOT inte re IP is internal (10.* Flow diretion is "egr nation Port is 80 lard Devation is calo	ernal (10.* net * network) ess" culated at TEN	twork) N times above the	average	,						
Benefits	5:			-							
1. View 2. Use th	how many IP addre hat intersection of d	sses intersec lata to feed th	t two distinct data e following searcl	isets h for destination sources, a	and whehter or not th	ose destinations	are prohibited, C	DK, etc.			
Search	String - Hosts	in both Pro	oxy & Netflow	Data	IP Add	Iresses that	appear in Pro	oxy & Ne	tflow Data	Sources	
(index=	(index="netflow" src_ip=*) OR (index="proxy" src=*)			ipAddr 🤇				netflow	0	proxy 0	
eval fiel	<pre> eval ipAddr=if(isnull(src),src_ip,src) fields index ipAddr</pre>		10.232.4	.55			20	05	1127		
char sear wher	chart c(ipAddr) AS count over ipAddr by index search netflow=1 AND proxy=1 where netflow >= proxy OR netflow <= proxy				10.247.3	10.120			1.	16	4
Search	String - Hosts	Visiting Pr	ohibited Sites	5	Proxy	Info for IPs f	ound				
index="	proxy" sourcetyp	e="bluecoat	:proxysg:acces	s:syslog" src=10.232.	4.55 src 0	dest	0		bytes_in ≎	bytes_out 0	range ¢
OR src= stats	10.247.30.120 sum(bytes_in) a	as bytes_in	sum(bytes_out)	as bytes_out by src,	dest 10.232.4	.55 profile	e.ak.fbcdn.net		105277932	14510095	Ø
eval 00")	ok=case(dest="ww	w.facebook	.com","100",des	t="googletb.skype.com	","1 10.232.4	.55 platfo	rm.ak.facebook.	com	19341272	3142993	Ø
sort	<pre>where netflow >= proxy OR netflow <= proxy earch String - Hosts Visiting Prohibited Sites ndex="proxy" sourcetype="bluecoat:proxysg:access:syslog" src=10.232.4.5 R src=10.247.30.120 stats sum(bytes_in) as bytes_in sum(bytes_out) as bytes_out by src.des eval ok=case(dest="www.facebook.com","100",dest="googletb.skype.com"," or) sort - bytes_out rangemap field=ok severe=100-1000 default=low fields - ok</pre>		10.232.4	.55 www.	facebook.com		45722802	1966758	0		
field	s - ok	<pre>> is calculated at TEN times above the average > addresses intersect two distinct datasets ion of data to feed the following search for destination sources, ar losts in both Proxy & Netflow Data src_ip=*) OR (index="proxy" src=*) (isnull(src),src_ip,src) pAddr) AS count over ipAddr by index >1 AND proxy>1 >> proxy OR netflow <= proxy losts Visiting Prohibited Sites rrcetype="bluecoat:proxysg:access:syslog" src=10.232.4 120 in) as bytes_in sum(bytes_out) as bytes_out by src,d sts="www.facebook.com","100", dest="googletb.skype.com" tt tok severe=100-1000 default=low</pre>	10.232.4	.55 ad.do	ubleclick.net		4916678	1310207	Ø		
					10.232.4	.55 photo	s-a.ak.fbcdn.net		13806220	1047114	I
					10.232.4	.55 photo	s-f.ak.fbcdn.net		14651846	1047024	0
					10.232.4	.55 photo	is-g.ak.fbcdn.net		10479682	814795	0
					10.232.4	.55 photo	s-e.ak.fbcdn.net		10475464	814272	Ø
					10.232.4	.55 photo	s-b.ak.fbcdn.net		10178784	806256	
					10.232.4	.55 photo	s-h.ak.fbcdn.net		8327204	787914	Ø
					10.232.4	.55 creati	ve.ak.fbcdn.net		20172188	756650	
					10.232.4	.55 photo	s-c.ak.fbcdn.net		7651281	582120	S
					10.232.4	.55 photo	s-d.ak.fbcdn.net		5666225	453339	
					10.232.4	l.55 m1.2r	mdn.net		14406599	381548	
											· ·

tegory_id=GIFTS&JSESSIONID=SDISLAFF10ADFF10 HTTP 1.1" 404 720 "http://buttercup-shopping.com/cart.do?action=vriew&itemId=EST-a@@product_id=F1_SU=30 (AST) (A

Extra Credit: Proxy + NetFlow

```
(index="netflow" src_ip=*) OR (index="proxy" src=*)
| eval ipAddr=if(isnull(src),src_ip,src)
| fields index ipAddr
| chart c(ipAddr) AS count over ipAddr by index
| search netflow>1 AND proxy>1
| where netflow >= proxy OR netflow <= proxy</pre>
```

pAddr 🗘	netflow 🗘	proxy 🗘
10.232.4.55	205	1127
10.247.30.120	116	4

```
index="proxy"
sourcetype="bluecoat:proxysg:access:syslog"
src=10.232.4.55 OR src=10.247.30.120
| stats sum(bytes_in) as bytes_in sum(bytes_out) as
bytes_out by src,dest
| eval
ok=case(dest="www.facebook.com","100",dest="googletb
.skype.com","100")
| sort - bytes_out
| rangemap field=ok severe=100-1000 default=low
| fields - ok
```

Proxy Info for	r IPs found			
src ≎	dest 0	bytes_in 🗘	bytes_out 0	range 0
10.232.4.55	profile.ak.fbcdn.net	105277932	14510095	Ø
10.232.4.55	platform.ak.facebook.com	19341272	3142993	
10.232.4.55	www.facebook.com	45722802	1966758	0
10.232.4.55	ad.doubleclick.net	4916678	1310207	Ø
10.232.4.55	photos-a.ak.fbcdn.net	13806220	1047114	Ø
10.232.4.55	photos-f.ak.fbcdn.net	14651846	1047024	Ø
10.232.4.55	photos-g.ak.fbcdn.net	10479682	814795	Ø



STEP BY STEP GUIDE – Extra Credit

Start with a look at the search "Possible Data Exfiltration":

- Hover over the panel, and click on the magnifying glass icon to launch search
- How many Stats rows do you see returned out of how many events?
- Which is the most interesting row?
 - Click on "10.232.4.55", then "View Events"
 - What does that data tell you?
- Take a look at the second panel ... "Proxy Info for IPs Found":
 - Hover over the panel, and click on the magnifying glass icon to launch search
 - What are the apps most in use by these hosts?





NK INC.

Thank you!

Kelly Feagans | Sales Engineering