Building a Splunk-based Lumber Mill

Turning a bunch of logs into useful products
About us

- Bob Bregant
  - Senior IT Security Engineer
  - Takes top billing when he can

- Joe Barnes
  - Interim CISO
  - Puts up with me
Prerequisites

• Central logging
  – Can’t do anything interesting with the data that you don’t have

• Or at least somewhat central
  – Splunk can help us out here
Central Logging

• Mostly a couple of big sources
  – Total of 430 GB/day
  – Bro (275 GB), Firewalls (90 GB), DNS (30 GB), Exchange (10 GB), AD (10 GB)
  – Everything else (Shibboleth, VPN, networking devices, servers): 15 GB

• That’s a thousand devices that take less than 150GB of space (over 90 days, 10:1 zip ratio)
Going Splunking

• Alternatives
  – Open Source
  – Commercial

• Why we went this route
  – Access control
  – Flexible fields
  – Nice API
Splunk Architecture

- **Indexers**
  - One per 150-250 GB

- **Search Heads**
  - Can pool, split for performance

- **Cluster?**

- **Other Stuff**
  - DB, Syslog (Heavy) Forwarder
Syslog Integration

• Syslog forwarder is important
  – Splunk can grab data from DB’s, flat files
  – Syslog provides vendor-neutral backup
  – Can always import data from archive

• Heavy forwarder required for syslog
  – Not perfect, but it gets the job done
  – Configs in Appendix B
Extracting Data

• Search-time extraction
  – Can change fields at any time, retroactive
  – Aliases for democratic field naming

• Get good at PCRE
  – [Link to Splunk regular expressions]
  – (?!^S+\sS+\sRT_FLOW:\s(\w*)\s(\d+)\s(\d+)\s(\d+))
  – Don’t let that scare you!
Extracting Data

- Computed fields
  - Great for normalization

- Example - MAC normalization:
  ```python
  lower(replace(replace(src_mac,"([-\.:])",""),"(\w\w)(\w\w)(\w\w)(\w\w)(\w\w)(\w\w)","1:2:3:4:5:6"))
  ```
Extracting Data

splunk>

New Search

src_mac=de:ad:be:ef:12:34 | sort _time | table _raw

25 events (10/13/14 5:20:00.000 PM to 10/14/14 12:00:00.000 PM)

Events

20 Per Page Format Preview

_raw

2014-10-13T12:20:13-05:00 dhcpserver dhcpd[21427]: DHCP on 172.0.0.10 to dead:be:ef:12:34 (bobscphone) via bond0 relay bond0 lease-duration 3600 (RENEW)

2014-10-13T12:20:13-05:00 dhcpserver dhcpd[21427]: DHCPREQUEST for 172.0.0.10 from dead:be:ef:12:34 (bobscphone) via bond0 (RENEW)

2014-10-13T12:40:36.899201-05:00 radiusserver radius.auth1[14548]: [wireless-wifi-local] Auth OK for bregant2@illinois.edu (Outer EAP Identity bregant2@illinois.edu) on DE-AD-BE-EF-12-34 via 172.0.0.12:12-34-56-78-90-AB:total

2014-10-13T12:40:41-05:00 dhcpserver dhcpd[21427]: DHCP on 172.0.0.10 to dead:be:ef:12:34 (bobscphone) via bond0 relay 172.0.0.1 lease-duration 3600 (RENEW)

2014-10-13T12:40:41-05:00 dhcpserver dhcpd[21427]: DHCPREQUEST for 172.0.0.10 from dead:be:ef:12:34 (bobscphone) via 172.0.0.1 (RENEW)

2014-10-13T12:40:41-05:00 dhcpserver dhcpd[26719]: DHCP on 172.0.0.10 to dead:be:ef:12:34 (bobscphone) via bond0 relay 172.0.0.1 lease-duration 3600 (RENEW)

2014-10-13T12:40:41-05:00 dhcpserver dhcpd[26719]: DHCPREQUEST for 172.0.0.10 from dead:be:ef:12:34 (bobscphone) via 172.0.0.1 (RENEW)

2014-10-13T12:41:28.231521-05:00 radiusserver radius.auth1[14548]: [wireless-wifi-local] Auth OK for bregant2@illinois.edu (Outer EAP Identity bregant2@illinois.edu) on DE-AD-BE-EF-12-34 via 172.0.0.12:12-34-56-78-90-AB:total
Searching Data

• Map looks like what you want
  – It’s not. (It works, but not well)

• Subsearch
  – Correlation
    – index=wireless [some search that returns a list of macs] | other stuff that you want to do, maybe a lookup table of AP locations
Some actual searches, part 1

• Basic search to show how we chain commands and manipulate data: What percentage of our users login to the VPN with two-factor authentication?

index=vpn login_result=”Success”

• Only care about successful logins to the VPN
Some actual searches, part 1

- What percentage of our users login to the VPN with two-factor authentication?

```bash
index=vpn login_result=“Success” | stats count(_raw) as total, count(eval(auth_ip=“1.2.3.4”)) as duo
```

- We need both the total and the number who logged in with Duo to get the percent (subbing in the real Duo IP for “1.2.3.4”).
Some actual searches, part 1

- What percentage of our users login to the VPN with two-factor authentication?

```
index=vpn login_result=“Success” | stats count(_raw) as total, count(eval(auth_ip="1.2.3.4")) as duo | eval duo_pct=(duo/(total-duo))*100
```

- Calculating the actual percentage here
Some actual searches, part 1

• What percentage of our users login to the VPN with two-factor authentication?

```bash
index=vpn login_result="Success" | stats count(_raw) as total, count(eval(auth_ip="1.2.3.4")) as duo | eval duo_pct=(duo/(total-duo))*100 | table duo_pct
```

• And now a quick table to show off the result
Some actual searches, part 1

VPN

VPN 2FA Usage
Some actual searches, part 2

• Actual Correlation: Simplified Copyright Attribution

index=firewall nat_src_ip=$SRC_IP$ AND nat_src_port=$SRC_PORT$ | head 1 | fields src_ip

• Quick bit of flow verification and NAT translation
Some actual searches, part 2

• Actual Correlation: Simplified Copyright Attribution

```
index=dhcp action=DHCPACK [search index=firewall
nat_src_ip=$SRC_IP$ AND nat_src_port=$SRC_PORT
$ | head 1 | fields src_ip] | head 1 | fields src_mac
```

• DHCP lookup, since we need a MAC to search some authentication logs
Some actual searches, part 2

• Actual Correlation: Simplified Copyright Attribution

(index=radius AND login_result="Auth OK") OR (index=quickconnect AND action="login") [search index= dhcp action=DHCPACK [search index= firewall nat_src_ip=$SRC_IP$ AND nat_src_port=$SRC_PORT$ | head 1 | fields src_ip] | head 1 | fields src_mac] | dedup user | fields user

• Now we use that MAC to find the user
Some actual searches, part 2

Yes, the IP is anonymized, but there were real logs behind that search.
Users and Permissions

• AD authentication and **authorization**
  – Help Desk, for example
  – `srchFilter = event_id=644 OR event_id=4740`

• **Summary indices**
  – For secure C-level view

• Apps also provide some separation
Bringing in new data

- Central Logging
  - Inefficient, but reliable
- Splunk Agents/Splunk DB
  - Can be dangerous
  - Use Splunk to syslog
Lessons Learned

• Have a plan for fields (Splunk Common Information Model, Apps)
• Private IP space isn’t your friend
• Deployment server is your friend
• Users will want apps
• Be prepared to be popular
Selling the Dream

• Disclaimer
• Know Your Audience
• Address Needs
• Success Stories
• Tell Everyone
Disclaimer

The following statements apply to Splunk as our logging frontend. After exploring and building a central logging environment in a heterogeneous, decentralized environment like the University of Illinois at Urbana-Champaign, Splunk was selected because it met our needs with respect to creating a scalable, manageable, and easily distributable platform that could be used to combine, correlate, and share data with functional groups so they can more effectively make decisions to their appropriate area. Regardless of the tool or solution you are currently utilizing, trying to expand, or just considering piloting, the remaining advice should be used as a guide for gaining support across your organization.
Know Your Audience

• Just because you know a tool can provide IT security intelligence to those who need it and that excites you, that is mostly meaningless to others.

• Just because you know a tool can provide IT operational intelligence to those who need it and that excites some, that is still mostly meaningless to others.

• Just because you know a tool can provide business intelligence to those who need it and that excites some, that is still mostly meaningless to others.
Know Your Audience

• Just because you know a tool can enhance research efforts to those who need it and that excites some, that is still mostly meaningless to others.
• Just because you know a tool can help fight crime and ensure public safety efforts and that excites some, that is still surprising meaningless to some.
The Point

• All valid uses of Splunk
• What’s in it for them?
  – Put yourself in their shoes
  – Everyone cares about something different
• There will never be a single selling point
  – Lead with what they care about
  – Follow up with additional bonuses
Address Needs

• Know your Audience ➔ Determine their Needs

• How can Splunk make their life’s easier?
  – Missing some metrics?
  – Reduce efforts around repetitive or manual tasks?
  – Clarity to questions that involve complex actions?
  – Justify current or projected funding needs?
Success Stories

• You can never have enough
  – Have multiple for each audience type
• Always be on the lookout for more
• Know them by heart
• Keep it simple and meaningful
• Borrow from others (as needed)
Tell Everyone

- Tell anyone and everyone who will listen
- If you aren’t sure if you’ve told someone about it, tell them again
- If they don’t listen or don’t seem to care keep trying
- You know you’re being successful when others start selling the dream for you
- At times you may feel like you oversold, don’t worry about that, just make sure you manage expectations.
Questions
Appendix A: Log Architecture
Appendix B: Splunk Configs

- Splunk2syslog config
- General input configs
Appendix B: Splunk Configs

Splunk to Syslog outputs.conf (on heavy forwarder):

[syslog]
defaultGroup = syslog-forwarders

[syslog:syslog-forwarders]
server = your.central.log.server:514
type = tcp
timestampformat = %b %e %H:%M:%S
Appendix B: Splunk Configs

Splunk to Syslog props.conf (on heavy forwarder):

[default]
TRANSFORMS-routing=syslogRouting
SEDCMD= s/[
\r]/ /g
Appendix B: Splunk Configs

Splunk to Syslog outputs.conf (on agents):

[tcpout]
defaultGroup = syslog-relay,indexers
maxQueueSize = 128MB
maxConnectionsPerIndexer = 10

[tcpout:syslog-relay]
server = your.splunk.to.syslog.heavy.forwarder:8514
useACK = true

[tcpout:indexers]
server = cname.for.all.of.your.indexers:8514
useACK = true
autoLB = true
Appendix B: Splunk Configs

General Input props.conf (on forwarders/indexers):

[your_sourcetype]
TIME_PREFIX = \stext_preceding_timestamp:\s
TIME_FORMAT = %s.%6N
MAX_TIMESTAMP_LOOKAHEAD = 125
SHOULD_LINEMERGE = False
LINE_BREAKER = ([\n\r]+)
TRUNCATE = 10240
Appendix B: Splunk Configs

General Input inputs.conf (on agent):
[monitor:///var/central_logs/bro.local/.../bro_http*]
disabled = false
sourcetype=bro_http
followTail = 0
host_segment = 5
index = bro
ignoreOlderThan = 5d
followSymlink = false
blacklist = \.(gz|bz2|z|zip|tar\..gz|tgz|tbz|tbz2)
Appendix C: Syslog Configs

• Bro Manager Configuration (an example endpoint)
• Forwarder Configuration
• Central Log Archive Configuration
• Template to allow resolved names in forwarded messages
Appendix C: Syslog Configs

- **Example Rsyslog Endpoint:**

  ```
  ### MODULES ###
  module(load="imuxsock") # provides support for local system logging (e.g. via logger command)
  module(load="imklog")  # provides kernel logging support (previously done by rklogd)
  #module(load="immark")  # provides --MARK-- message capability
  #module(load="imfile")
  module(load="omrelp")

  ### GLOBAL DIRECTIVES ###
  $ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat # Use default timestamp format
  $RepeatedMsgReduction off   # "the previous message was repeated n times" is expensive
  $PreserveFQDN on            # Always use full names
  $WorkDirectory /var/rsyslog # default location for work (spool) files
  $MainMsgQueueDequeueBatchSize 1024
  ```
Appendix C: Syslog Configs

• Example Rsylog Endpoint (continued):

```
####  Inputs ####
input(type="imfile" \
File="/var/log/application/somelog.log" \
Tag="some_log:" \
StateFile="state-some_log" \
Severity="info" \
Facility="local1")

# Repeat as many times as needed
```
Appendix C: Syslog Configs

- Example Rsystlog Endpoint (continued):

  #### Security Syslog Forwarding ####

  $\text{ActionQueueType LinkedList} \quad \# \text{ use asynchronous processing}$
  $\text{ActionQueueFileName this_servers_queue} \quad \# \text{ set file name, also enables disk mode}$
  $\text{ActionQueueDequeueBatchSize 1024}$

  $\ast.\ast :\text{omrelp:forwarder1:514;RSYSLOG_ForwardFormat}$
  $\text{ActionExecOnlyWhenPreviousIsSuspended on}$
  & $\ast.\ast :\text{omrelp:forwarder2:514;RSYSLOG_ForwardFormat}$
  $\text{ActionExecOnlyWhenPreviousIsSuspended off}$

  local1.\ast stop
Appendix C: Syslog Configs

- **Example Rsyslog Forwarder:**

  ```
  ## General Options ##
  $AbortOnUncleanConfig on
  $RepeatedMsgReduction off
  $PreserveFQDN on
  $EscapeControlCharacterTab off
  $WorkDirectory /var

  ## Load Modules ##
  module(load="/var/lib/rsyslog/impstats" interval="60" severity="7")
  module(load="/var/lib/rsyslog/imrelp")
  module(load="/var/lib/rsyslog/imudp")
  module(load="/var/lib/rsyslog/imtcp")
  module(load="/var/lib/rsyslog/omrelp")
  ```
Appendix C: Syslog Configs

- Example Rsyslog Forwarder (continued):

```plaintext
##Stats Output##
if $syslogfacility-text == 'syslog' and $syslogseverity-text == 'debug' then {
    action(type="omfile" dynafile="StatsFile") #DynaFile Templates must be defined elsewhere
    $ActionQueueType LinkedList
    $ActionQueueFileName stats_forward_queue
    $ActionQueueDequeueBatchSize 1024
    .*:* :omrelp:archive.server.address:514;RSYSLOG_ForwardFormat
    stop
}
##Log the rsyslog messages somewhere useful##
if $syslogfacility-text == 'syslog' then {
    action(type="omfile" file="/var/log/rsyslog.log")
    stop
}
```
Appendix C: Syslog Configs

- Example Rsyslog Forwarder (continued):

```plaintext
##Forward all the logs##
ruleset(name="standard_forward" queue.type="linkedlist" queue.dequeuebatchsize="1024") {
  if $fromhost startswith 'mail' then {
    $ActionQueueType LinkedList
    $ActionQueueFileName mail_forward_queue
    $ActionQueueDequeueBatchSize 1024
    *.* :omrelp:archive.server.address:514;RSYSLOG_ForwardFormat
    stop
  }
  else {
    $ActionQueueType LinkedList
    $ActionQueueFileName standard_forward_queue
    $ActionQueueDequeueBatchSize 1024
    *.* :omrelp:archive.server.address:514;ResolvedFWDFformat
    stop
  }
stop}
```
Appendix C: Syslog Configs

• Example Rs syslog Forwarder (continued):

  ##RELP Input on port 1514##
  input(type="imrelp" port="1514" ruleset="networking_forward")

  ##UDP Input on port 1514##
  input(type="imudp" port="1514" ruleset="networking_forward")

  ##RELP Input on port 1515##
  input(type="imrelp" port="1515" ruleset="standard_forward")

  ##UDP Input on port 1515##
  input(type="imudp" port="1515" ruleset="standard_forward")
Appendix C: Syslog Configs

- Example Central Archive:
- Very much the same as a forwarder (rulesets, if/thens, and inputs), except instead of forwarding we write out the logs:

```plaintext
action(type="omfile" name="ShibLogs" dynafile="HourlyFile" fileowner="syslog" filegroup="syslog" dirowner="syslog" dirgroup="syslog" dircreatemode="0755" filecreatemode="0644" queue.dequeuebatchsize="1024")
```
Appendix C: Syslog Configs

• Template for resolved names in forwarded logs (is used instead of RSYSLOG_ForwardFormat on Forwarder):

```plaintext
# Format to use resolved hostname
template(name="ResolvedFWDFormat" type="list") {
    constant(value="<")
    property(name="pri")
    constant(value=">")
    property(name="timestamp" dateFormat="rfc3339")
    constant(value=" ")
    property(name="fromhost")
    constant(value=" ")
    property(name="syslogtag" position.from="1" position.to="32")
    constant(value=" ")
    property(name="msg" spifno1stsp="on")
    property(name="msg" droplastlf="on")
    constant(value="\n")
}
```