.conf2014

YOUR DATA ADVENTURE

# In Depth with Deployment Server

Sanford Owings

Principal Consultant,

Splunk Professional Services

David Shpritz

Security Consultant, Aplura, LLC

splunk>

# Disclaimer

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 the 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.


# What is Splunk Deployment Server?

splunk>

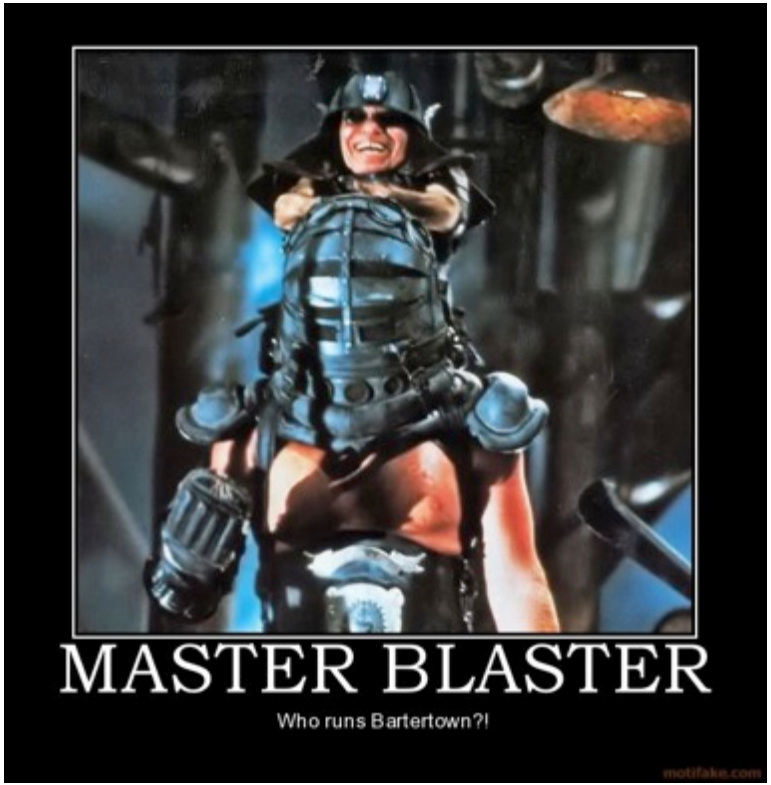
# What is Splunk Deployment Server?

- Doesn't actually deploy Splunk (common misperception)
- Acts as a configuration server
- Configurations are held in "apps" or "configuration bundles"
- Listens on the Splunk management port (8089 by default)
- Serves up lists of apps for clients to download and install
- The server configuration (serverclass.conf) describes what systems should download what apps

# Why use Deployment Server?

- No touching endpoints!
- Distribute add-ons to search heads to give users consistent field extractions
- Make sure you are getting a common set of inputs (satisfy auditors), that is, consistent configs
- Deployment server clients can be any part of Splunk infrastructure (search heads, indexers, forwarders of all types)

  (well, not clustered indexers)



# How can I select what systems are in what class?

- Serverclass.conf!
- Allows you to whitelist/blacklist/filter on different aspects of what is reported to the deployment derver
  - IP address
  - Host
  - ClientName (configured in deploymentclient.conf on the client)
  - machineTypesFilter (OS and architecture)

```
[serverClass:IntermediateHFs]
restartSplunkd = true
whitelist.0 = splk-heavyforwarder*
[serverClass:IntermediateHFs:app:DS-all_departments-IHF-base]
[serverClass:IntermediateHFs:app:DS-all_departments-Input-splunk_tcp_9997]
[serverClass:IntermediateHFs:app:DS-all_departments-Splunk-no_web]
```

# machineTypesFilter

- Acts as just what it says, filters systems based on OS and Arch
- Happens after the whitelist/blacklist
- This means that machineTypesFilter by itself won't match anything
- If you want all windows machines, you would need something like:

```
[serverClass:All-Windows]
restartSplunkd = true
whitelist.0 = *
machineTypesFilter = windows-intel,windows-x64
[serverClass:All-Windows:app:DS-all_departments-Input-windows_logs]
```

# How does this work?

- Serverclass.conf contains stanzas that define classes of systems (servers)
- Clients check in and subscribe to the classes they are included in
- Deployment server (DS) tarballs the deployment app, and hashes it
- The client keeps track of the hash of the app it has installed
- When it checks in, if the hash on the DS differs from what it has, the client downloads the new version
- After downloading, the client deletes the version it has and extracts the new version
- Restarting is optional (configured per serverclass)

# How does this work?

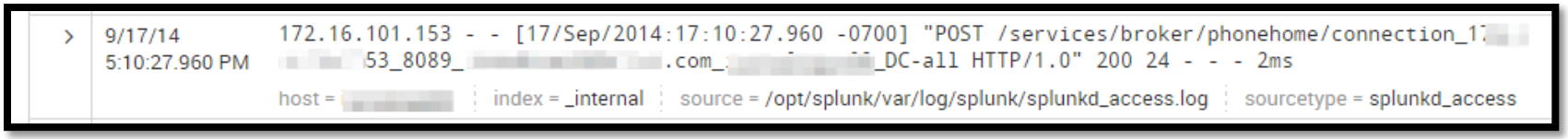
"Hmm, I haven't checked in in a while, better do that."

| | | |
|---|---|---|
| > | 9/9/14 10:05:31.104 AM | 09-09-2014 10:05:31.104 -0700 INFO DC:UpdateServerclassHandler - Changed state from=HandlingPhonehome to=Phonehome<br>host = [redacted] source = E:\Splunk\var\log\splunk\splunkd.log sourcetype = splunkd |
| > | 9/9/14 10:05:31.089 AM | 09-09-2014 10:05:31.089 -0700 INFO DC:UpdateServerclassHandler - Changed state from=Phonehome to=HandlingPhonehome<br>host = [redacted] source = E:\Splunk\var\log\splunk\splunkd.log sourcetype = splunkd |

# How does this work?

TCP 8089 (HTTPS)

“Hi, my name is forwarder1.
My IP is 192.168.1.2.
I have a ClientName of ForwarderSys.
I am running Windows on a 64-bit architecture.
I’m a Sagittarius (okay, not really).”



# How does this work?

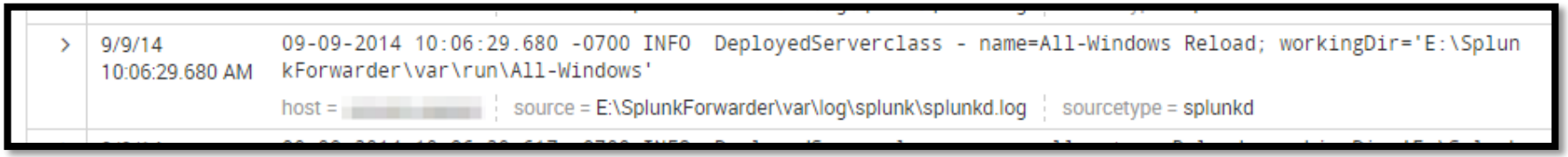
"Hmm. I haven't heard from this client since my last reload. I'll add it to the list of clients.

> 9/8/14 6:30:06.244 PM
> 09-08-2014 18:30:06.244 -0700 INFO ClientSessionsManager - Adding client: ip= uts=linux-x86_64 id=589025c9d87908d473e1b8af83bad99e name=DC-all
> host = source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd

# How does this work?

Response (same TCP connection)

"Hi, forwarder1. You belong in these classes:
- WindowsForwarder
- LocalForwarder"



# How does this work?


"I need a list of the apps in these classes:

- WindowsForwarder
- LocalForwarder"

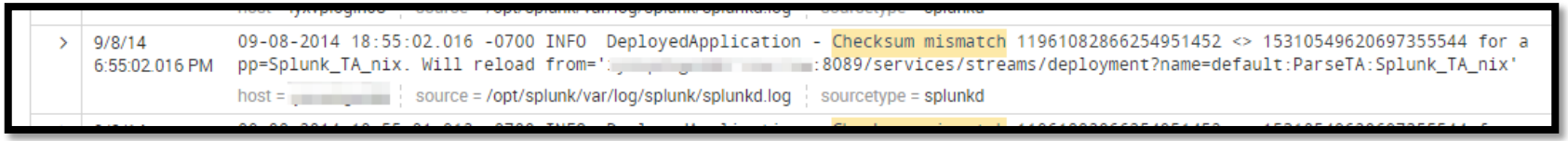
# How does this work?


"Sure, here are the apps and their hashes:

- Splunk_TA_windows  (hash: 9361937492720659309 8)
- Outputs_To_Indexers (hash: 11961082866254951452)"

# How does this work?

"Hmm, I have the right hash for Splunk_TA_windows, but for Outputs_To_Indexers I have a hash of 0. Better download it."



# How does this work?


"I need the latest version of Outputs_To_Indexers."

| | | |
|---|---|---|
| > | 9/9/14 10:06:29.680 AM | 09-09-2014 10:06:29.680 -0700 INFO DeployedServerclass - name=All-Windows Reload; workingDir='E:\SplunkForwarder\var\run\All-Windows' |
| | | host = [redacted] source = E:\SplunkForwarder\var\log\splunk\splunkd.log sourcetype = splunkd |

# How does this work?

Response (same TCP connection)

“Sure, here you go.”

| | | |
|---|---|---|
| > | 9/8/14 6:55:02.053 PM | 09-08-2014 18:55:02.053 -0700 INFO DeployedApplication - Downloaded url=[redacted]:8089/services/streams/deployment?name=default:ParseTA:Splunk_TA_nix to file='/opt/splunk/var/run/ParseTA/Splunk_TA_nix-1410218690.bundle' sizeKB=1120<br>host = [redacted] source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd |

# How does this work?

"Okay, now that I have the new version, I'll delete the existing one, and replace it with the new and shiny version. After that, I'll restart the Splunk services, because I was told to. Then I'll tell the DS the good news."

| | | |
|---|---|---|
| > | 9/8/14 6:55:01.972 PM | 09-08-2014 18:55:01.972 -0700 INFO DeployedApplication - Installing app=Splunk_TA_nix to='/opt/splunk/etc/apps/Splunk_TA_nix'<br>host = [redacted] source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd |

| | | |
|---|---|---|
| > | 9/10/14 12:13:38.805 PM | 09-10-2014 12:13:38.805 -0700 INFO ClientSessionsManager - ip=[redacted] name=DC-all Updating record for sc=SyslogFileInputs app=DS-all_departments-Input-syslog_files: action=Install result=Ok<br>action = Install eventtype = splunkd-log host = iyxvplogld01 index = _internal<br>source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd splunk_server = i[redacted] |

# Things to remember

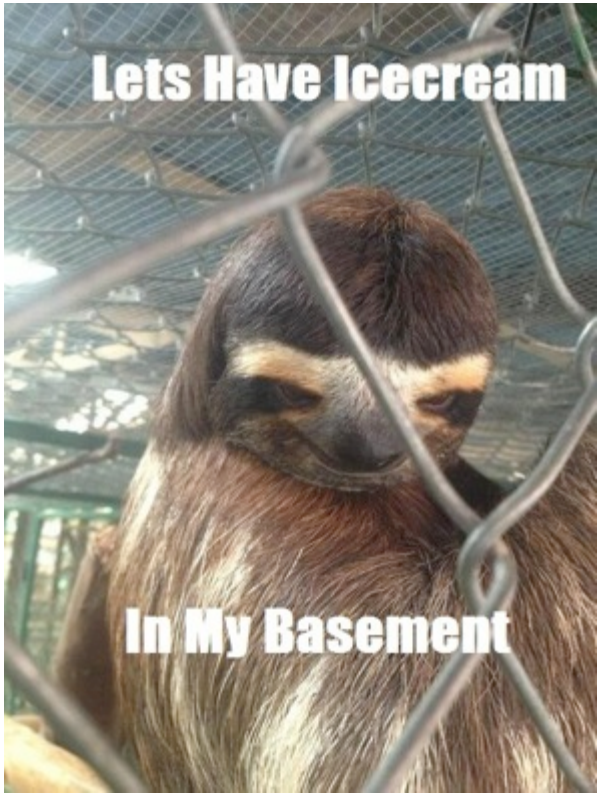
- Think of this as configuration enforcement (DS version wins!)
- Remember that delete portion. It will save you some headache.


Problems can happen…

splunk>

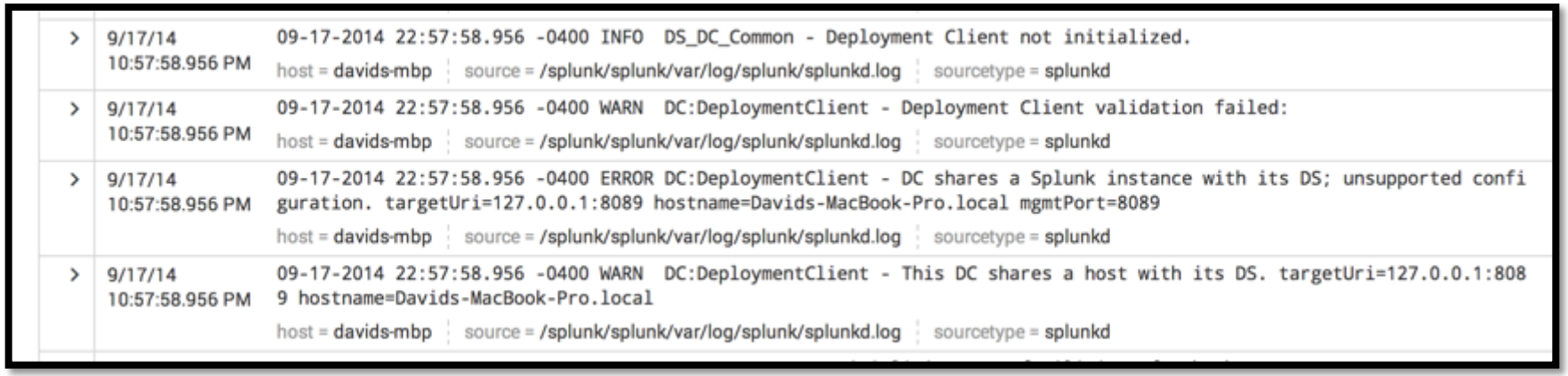
# Gotchas!

- Careful with lookups
- Splunk 6.2 resolves this problem, but older versions will overwrite the local lookup
- Careful with apps that have clickable content (setup GUIs, for example)
- General rule: Don't distribute apps with a UI where users can click to change configs
- Remember that delete thing? Yeah, the saved content would get nuked too
- Careful with what you restart
- Indexers and cluster masters can be touchy, restarting search heads means users may be unhappy



# Gotchas! (continued)

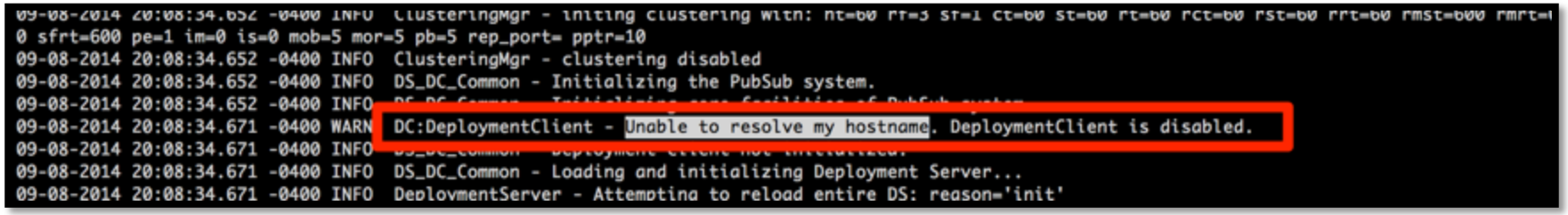
- A Deployment Server cannot deploy to itself
- 6.x will tell you about that, then may kill both.



| | | |
|---|---|---|
| > | 9/17/14 10:57:58.956 PM | 09-17-2014 22:57:58.956 -0400 INFO DS_DC_Common - Deployment Client not initialized.<br>host = davids-mbp source = /splunk/splunk/var/log/splunk/splunkd.log sourcetype = splunkd |
| > | 9/17/14 10:57:58.956 PM | 09-17-2014 22:57:58.956 -0400 WARN DC:DeploymentClient - Deployment Client validation failed:<br>host = davids-mbp source = /splunk/splunk/var/log/splunk/splunkd.log sourcetype = splunkd |
| > | 9/17/14 10:57:58.956 PM | 09-17-2014 22:57:58.956 -0400 ERROR DC:DeploymentClient - DC shares a Splunk instance with its DS; unsupported configuration. targetUri=127.0.0.1:8089 hostname=Davids-MacBook-Pro.local mgmtPort=8089<br>host = davids-mbp source = /splunk/splunk/var/log/splunk/splunkd.log sourcetype = splunkd |
| > | 9/17/14 10:57:58.956 PM | 09-17-2014 22:57:58.956 -0400 WARN DC:DeploymentClient - This DC shares a host with its DS. targetUri=127.0.0.1:8089 hostname=Davids-MacBook-Pro.local<br>host = davids-mbp source = /splunk/splunk/var/log/splunk/splunkd.log sourcetype = splunkd |

# Gotchas! (continued)

- The client hostname is important



# Gotchas! (continued)

- Careful with the numbering of your whitelists/blacklists in serverclass.conf

```
143 [serverClass:Level2Forwarders]
144 restartSplunkd = true
145 whitelist.1 = dns.company.com
146 whitelist.2 = loslobos.company.com
147 whitelist.3 = splunk.dept.*
148 whitelist.4 = 10.10.123.4
149 blacklist.0 = splunk01.dept.company.com
150 blacklist.1 = splunk02.dept.company.com
151 [serverClass:Level2Forwarders:app:DS-all_departments-Input-syslog_generic ]
152
```

# Gotchas! (continued)

- Splunk precedence still applies!
- $SPLUNK_HOME/etc/system/local/*.conf still wins
- The names of your apps still matter
- Splunk configuration layering is king!
- A fun Splunk tongue-twister:

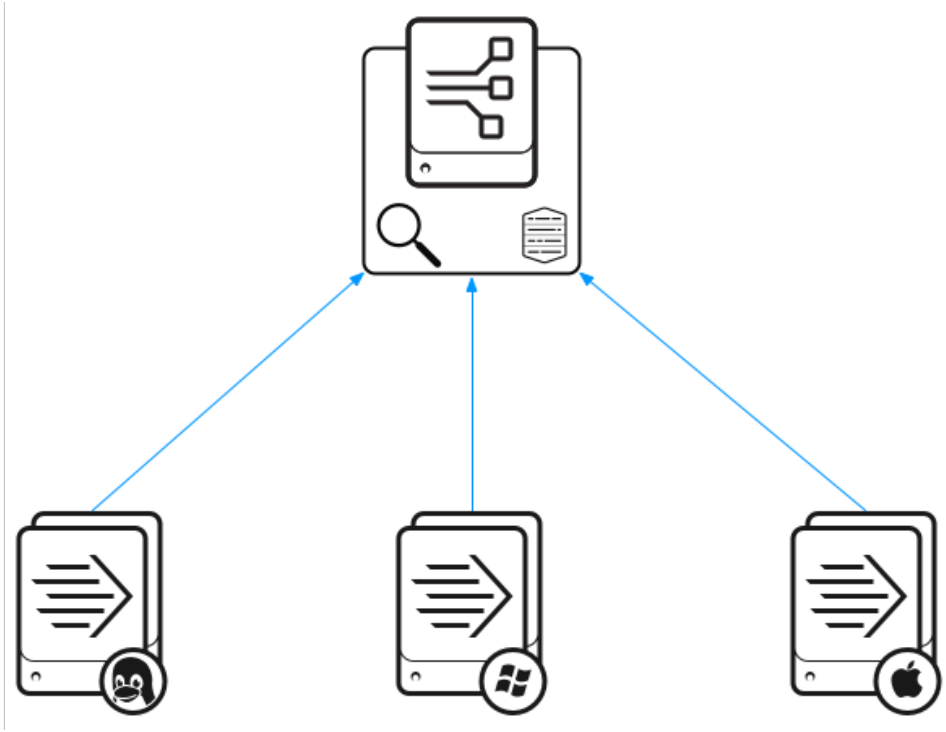
```
grep conf conf.conf | grep -v confdb
```

(run in $SPLUNK_HOME/etc/system/default)

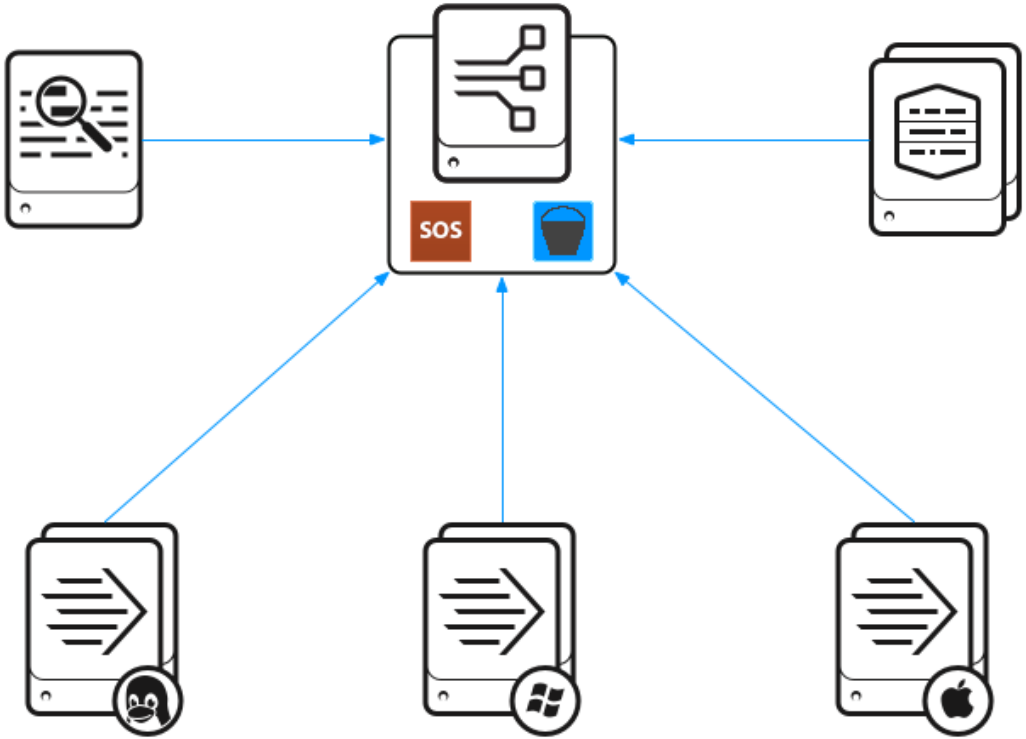

# Typical Deployment Patterns

splunk>

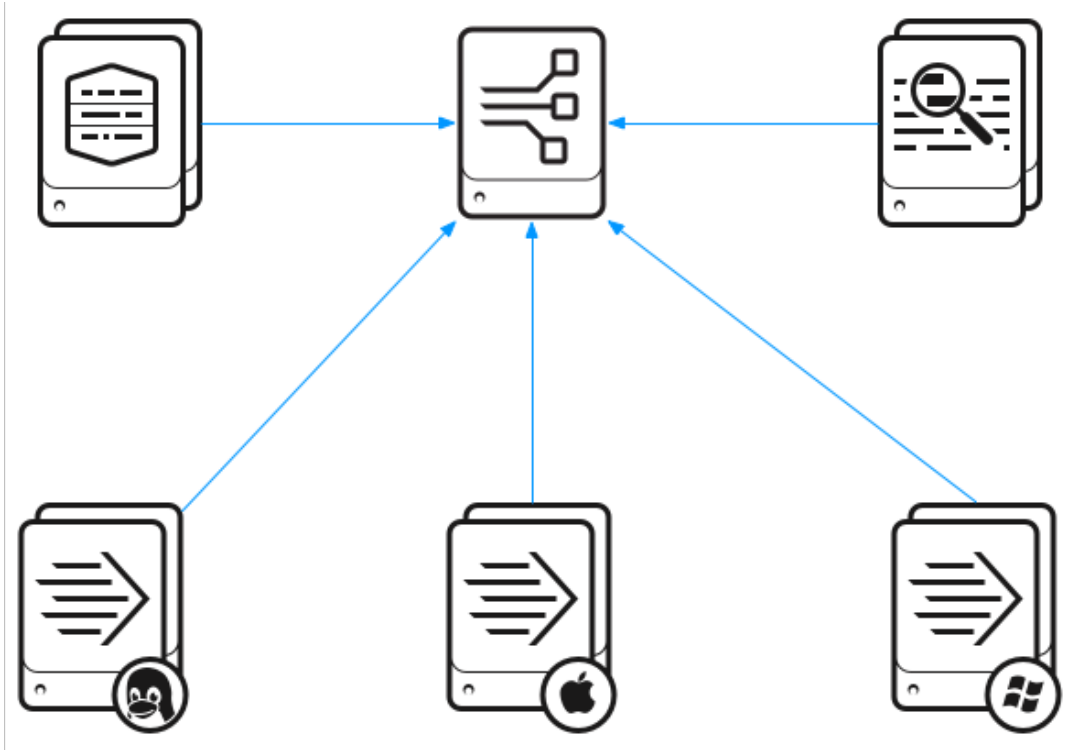
# The all-in-one



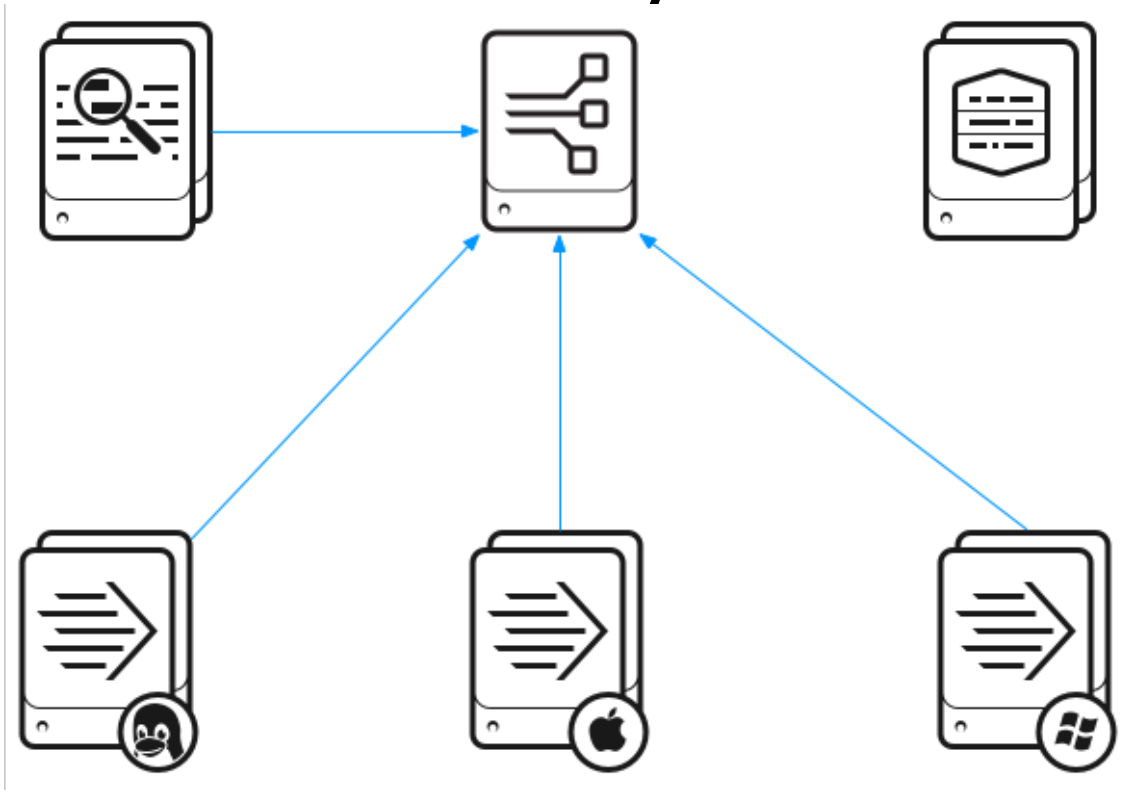
# The Admin Server



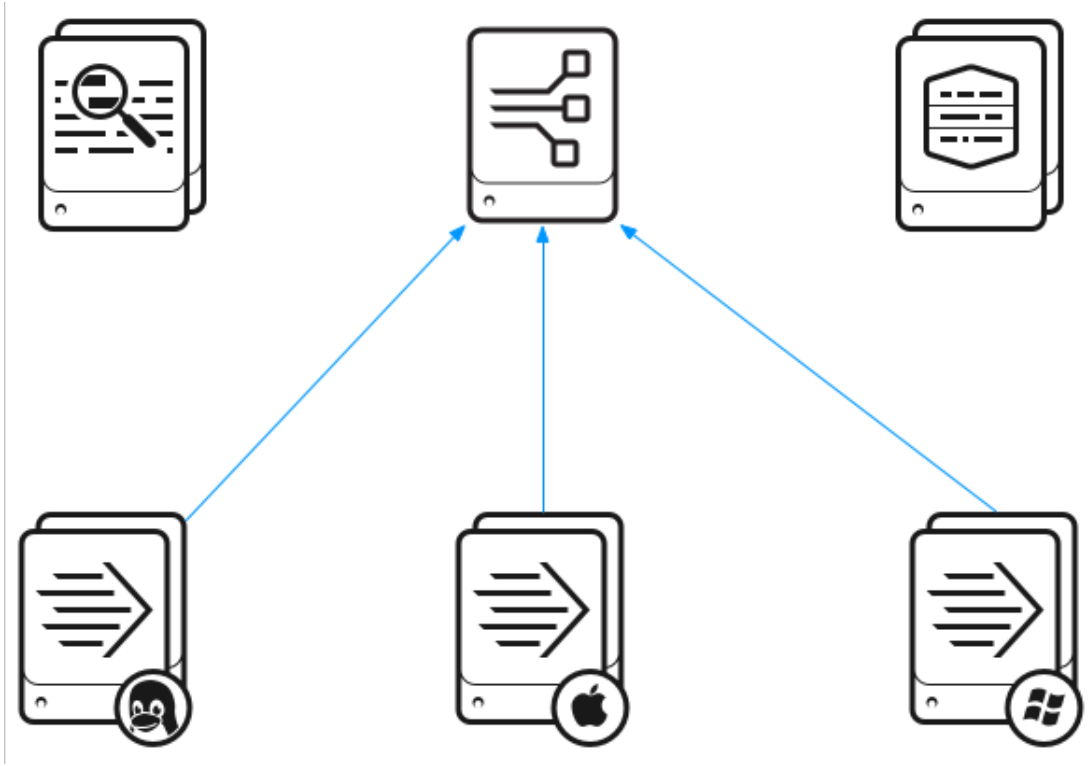
# Dedicated Deployment Server



# Don’t touch my indexers!



# Forwarders Only




# Advanced tips and tricks

splunk>

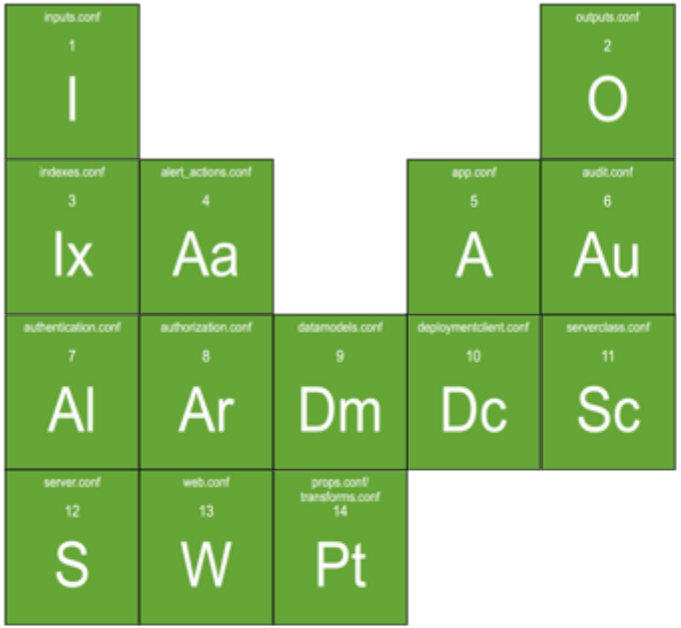
# Don’t chain yourself to a host/IP

- Hosts change (age out, break, need to be upgraded, it’s the circle of life)
- If you are using a host name or an IP in your deploymentclient.conf, and that IP or host name changes, that config file will need to be changed EVERYWHERE
- Instead, use a separate DNS record (A or CNAME) to enter into your client configs (“splunk-ds.mycompany.com”)

# Create smaller, more discrete apps

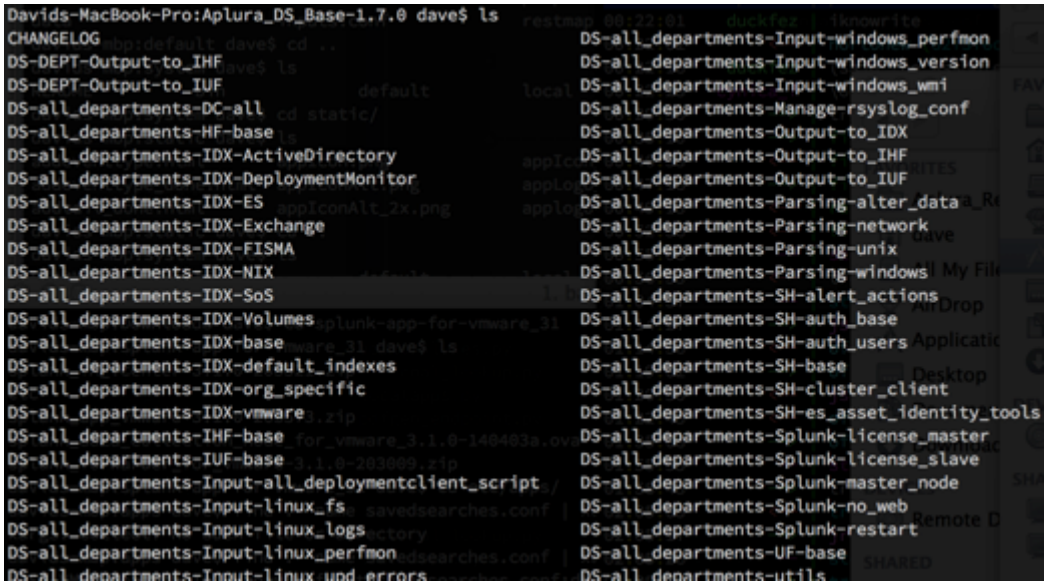
- Keep the number of config files per app low
- This creates smaller, reusable modules
- Lets you take advantage of Splunk's configuration layering
- Turns out, this is easier to debug
- Use a naming convention for the apps
- Example: DS-<org group>-<class>-<description>
  DS-dmz-Output-To_Forwarder
- Create classes of apps
- Input apps
- Index apps
- Web control apps (turn off Splunkweb)

# Atomic apps combine to make larger config molecules





# Yes, you may end up with a lot of apps…




- Naming convention + tab auto-completion FTW!
- On Linux? The “find” command is awesome!

```
# find /opt/splunk/etc/deployment-apps props.conf | xargs grep mysourcetype
```

# Why not larger apps?

- Very hard to reuse
- Configurations quickly become clumsy
- Makes debugging problems more difficult
- Not as flexible

# Remember that whole etc/system/local thing?

- Configuration layering always applies!
- Changing your deployment server? Migrating? Rename?
- $SPLUNK_HOME/etc/system/local/deploymentclient.conf WINS!
- Prepare to touch all your endpoints
- Puppet? Chef? SCOM? Pick your poison
- What about…

# Scripted inputs to the rescue

- Can run a script on a regular basis
- Can run on all of the deployment clients
- .sh, .bat
- Rename or remove the $SPLUNK_HOME/etc/system/local/deploymentclient.conf!
- The “splunk” user should already own the file
- Distribute the app to all systems, or create a server class that only applies to a section of clients

# Breaking up serverclass.conf…

- The configs can get long
- Serverclass.conf is like other Splunk config files, stanzas get added to each other

# … maybe not

- If you use the Forwarder Management GUI, it may fragment the configs in unexpected ways.
- May be the **one** file we actually only want in system/local

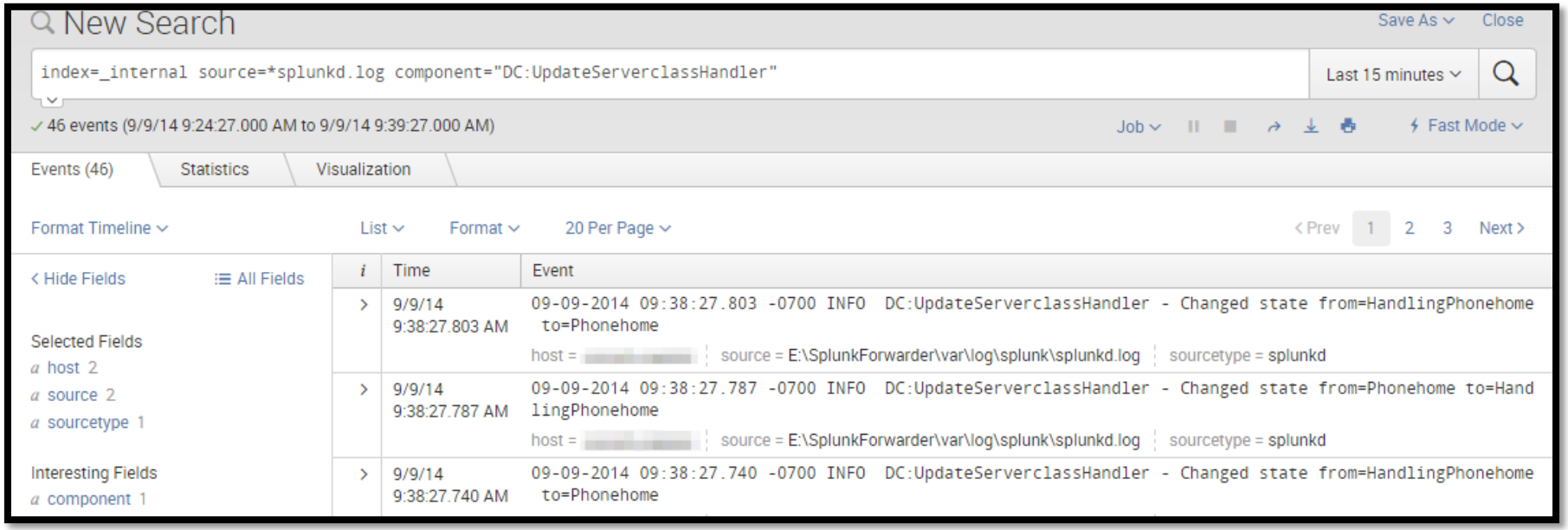

# Troubleshooting Splunk Deployment Server

splunk>

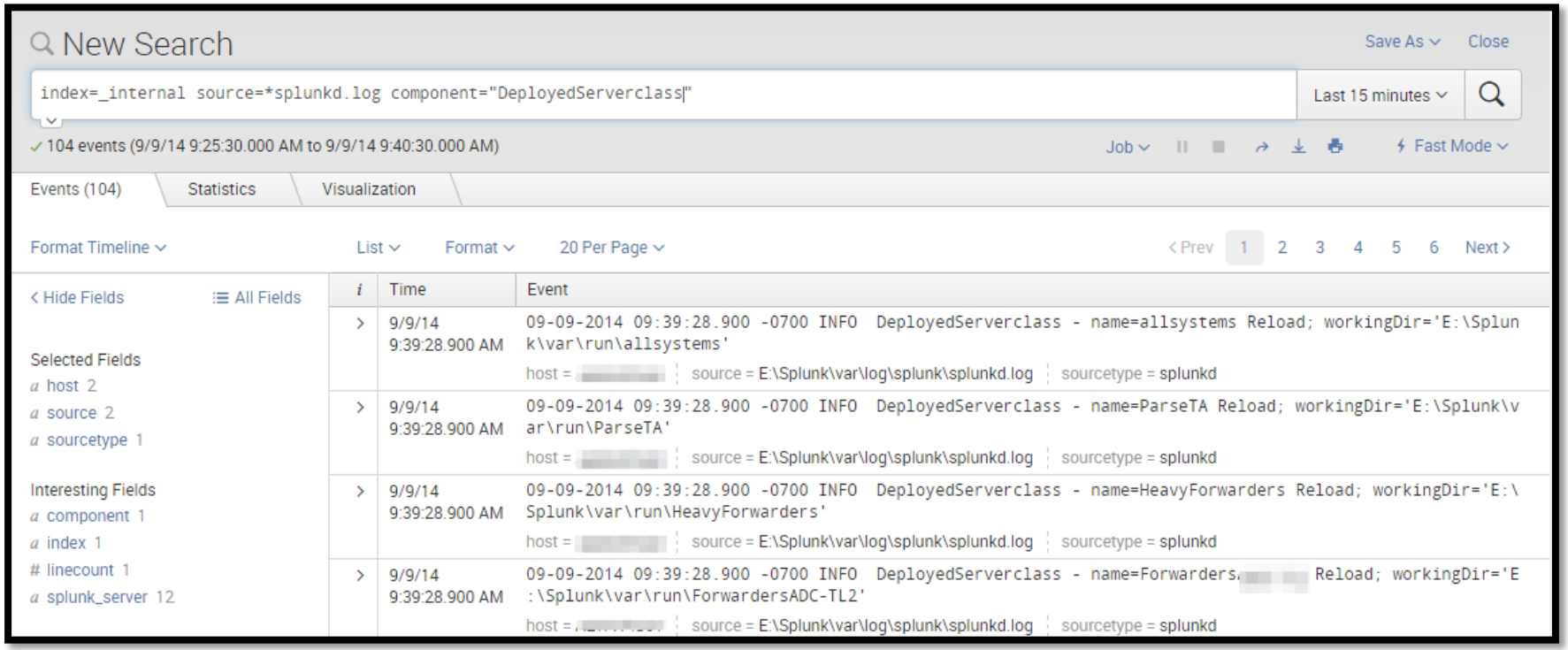
# Troubleshooting Deployment Server

- Host != FQDN
- Can the client resolve the name of the deployment server?
- Can the client communicate?
- Use the GUI to check for check-ins from the client
- Settings > Forwarder Management
- Search: index=_internal source=*splunkd.log ClientSessionManager
- Check that the correct apps are on the client
- Search from DS:
  index=_internal source=*splunkd.log ClientSessionsManager action=*

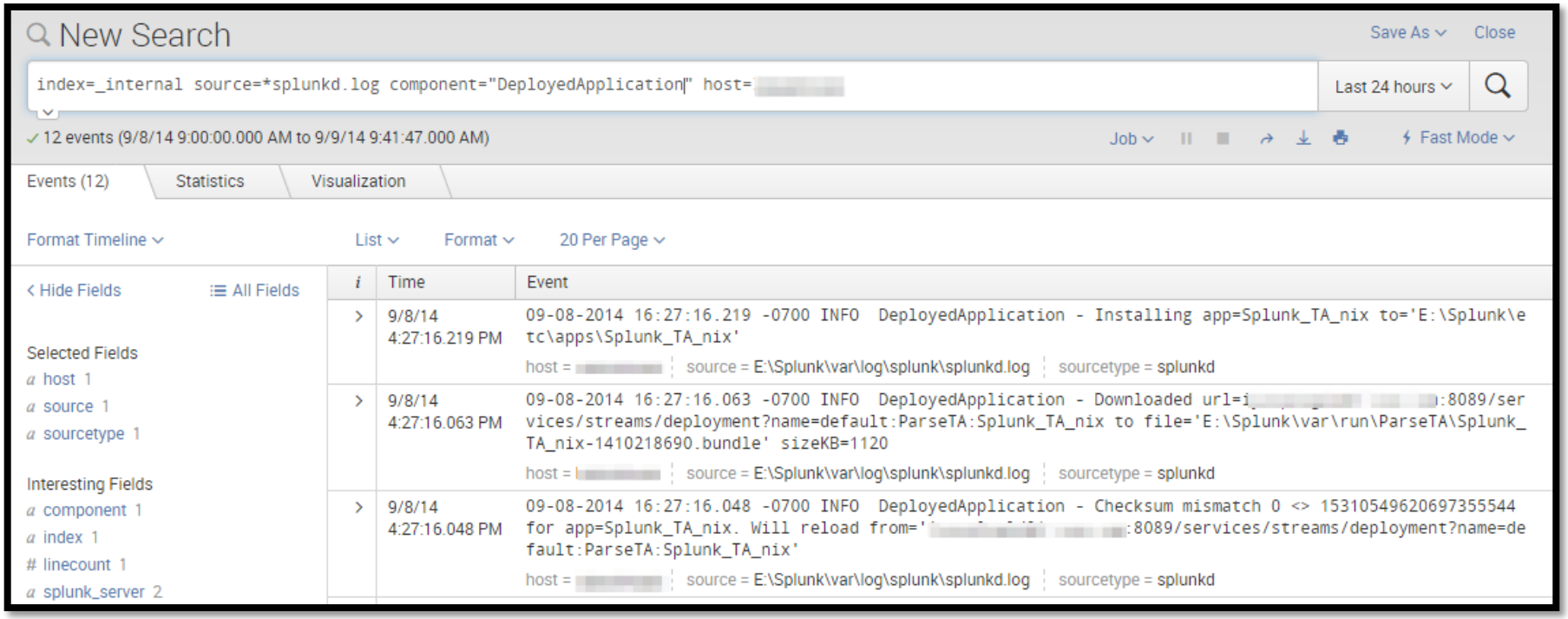
# Client: PhoneHome state



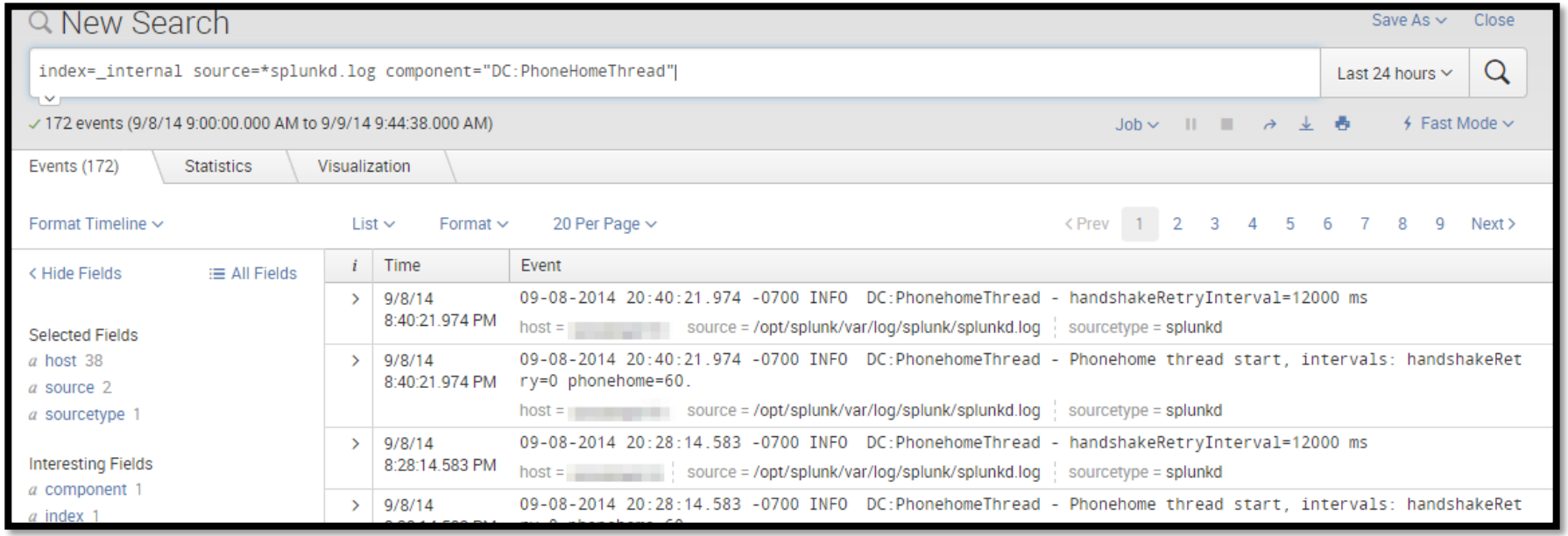
# Client: Refreshing a serverclass



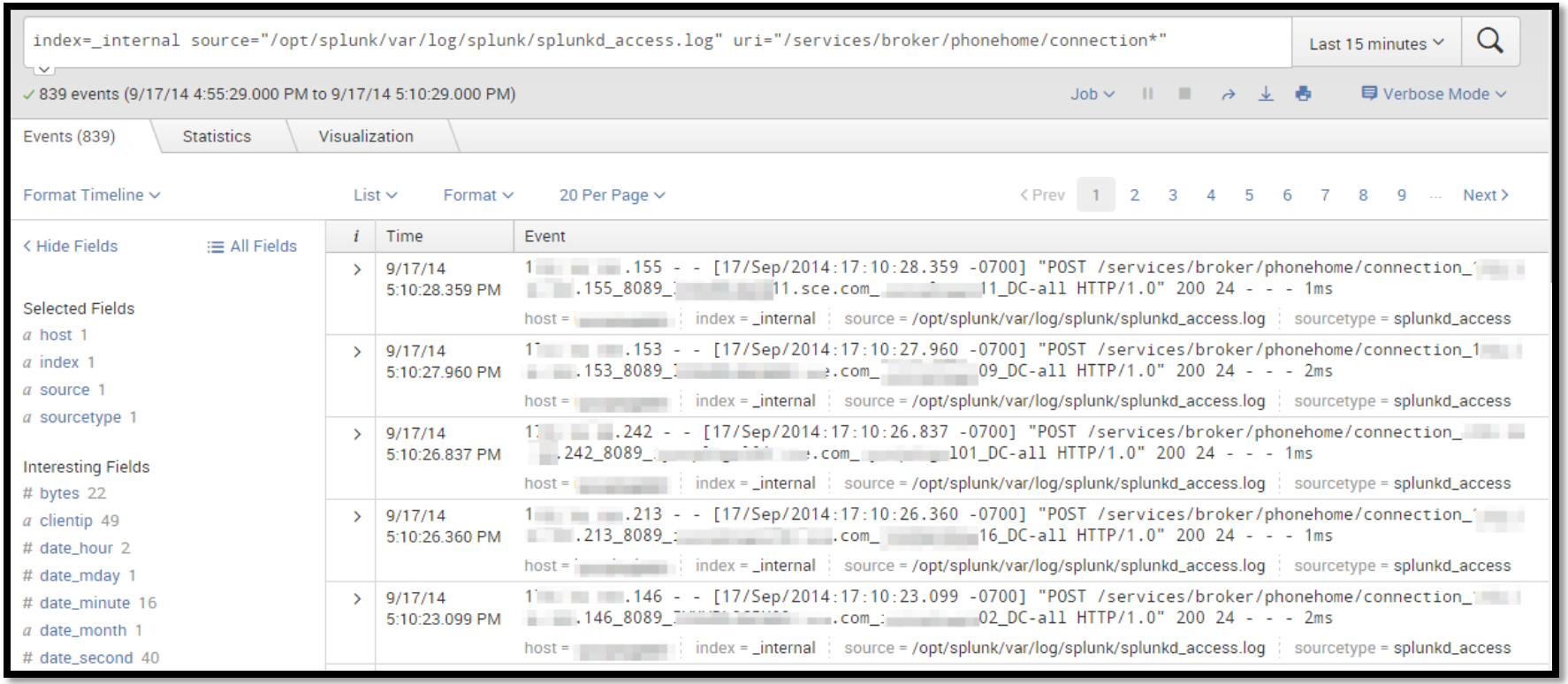
# Client: Downloading and installing apps



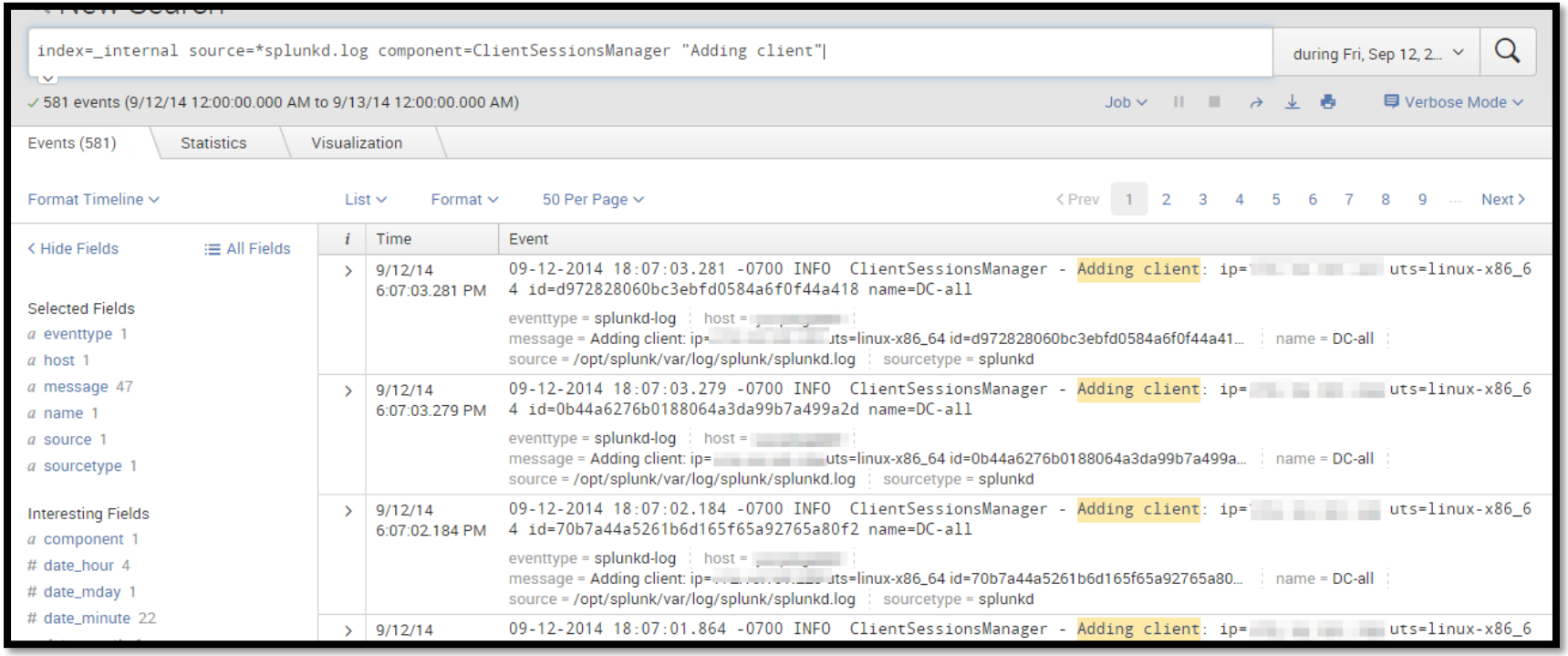
# Client: PhoneHome



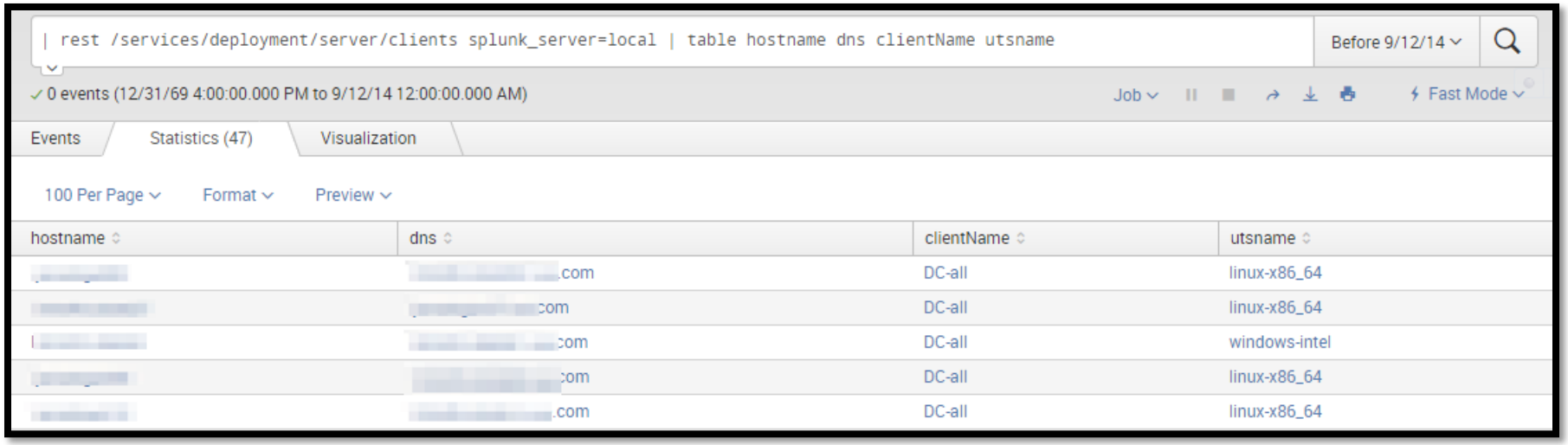
# Server: PhoneHome



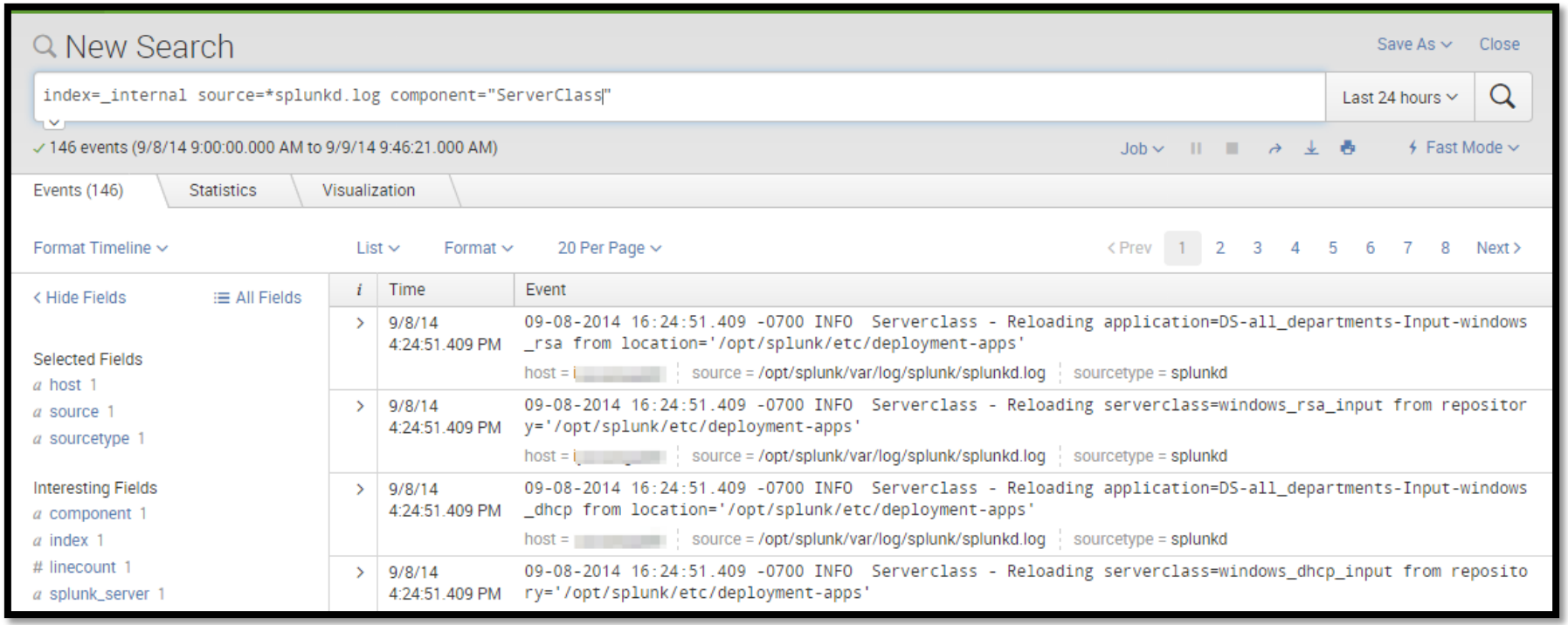
# Server: Recording client check-ins



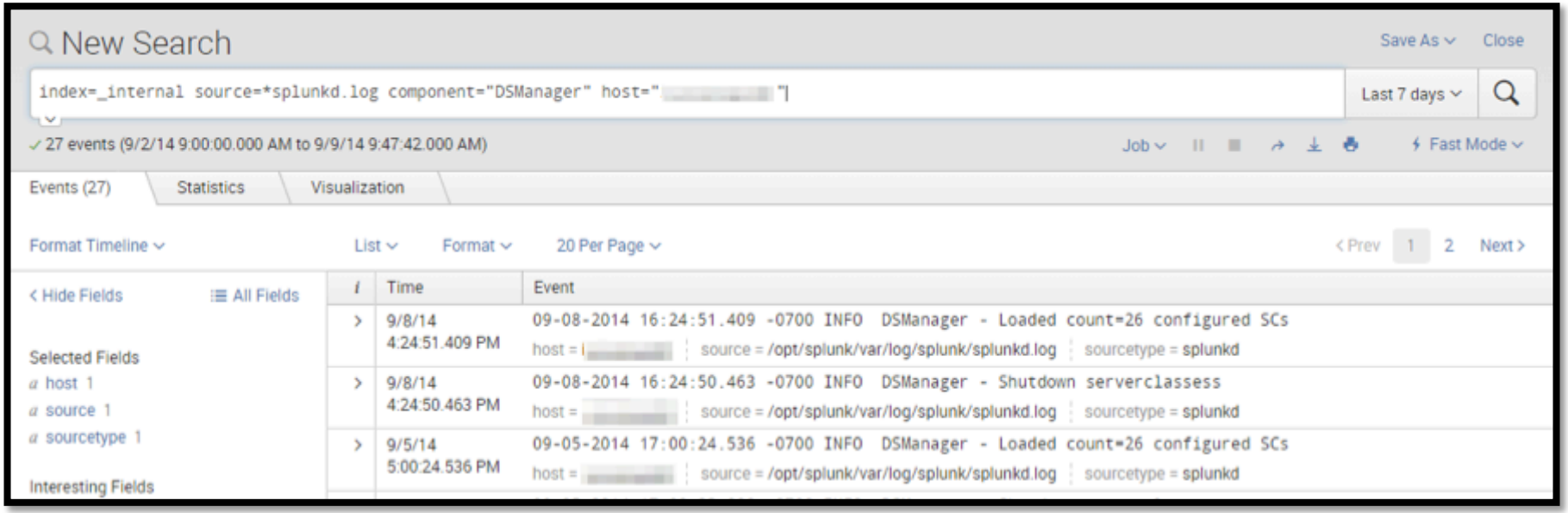
# Server: List the deployment clients



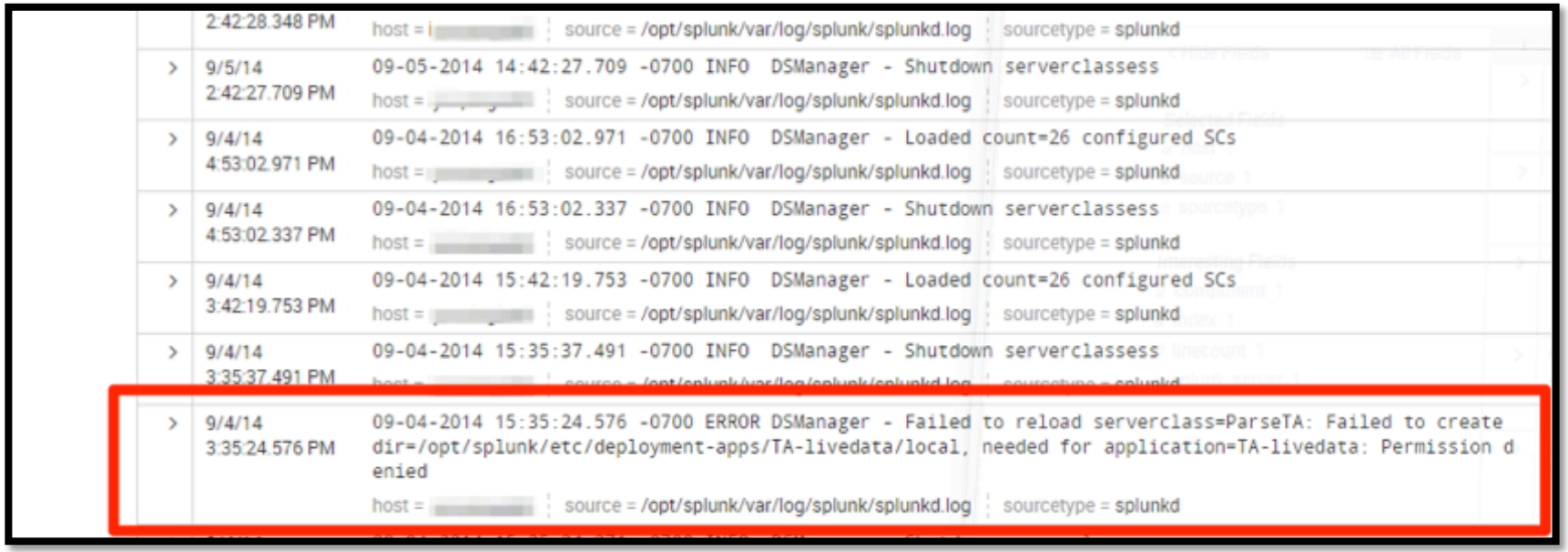
# Server: Loading classes and apps



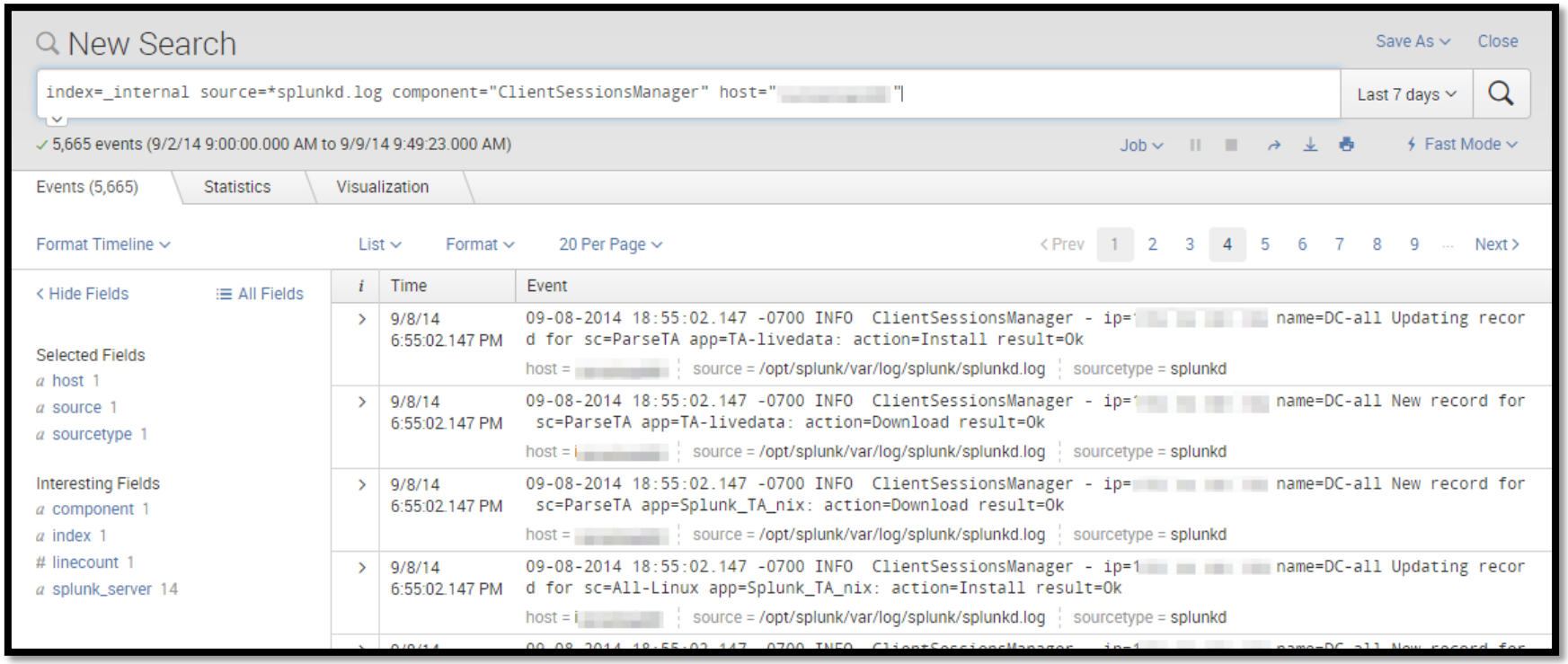
# Server: Reload



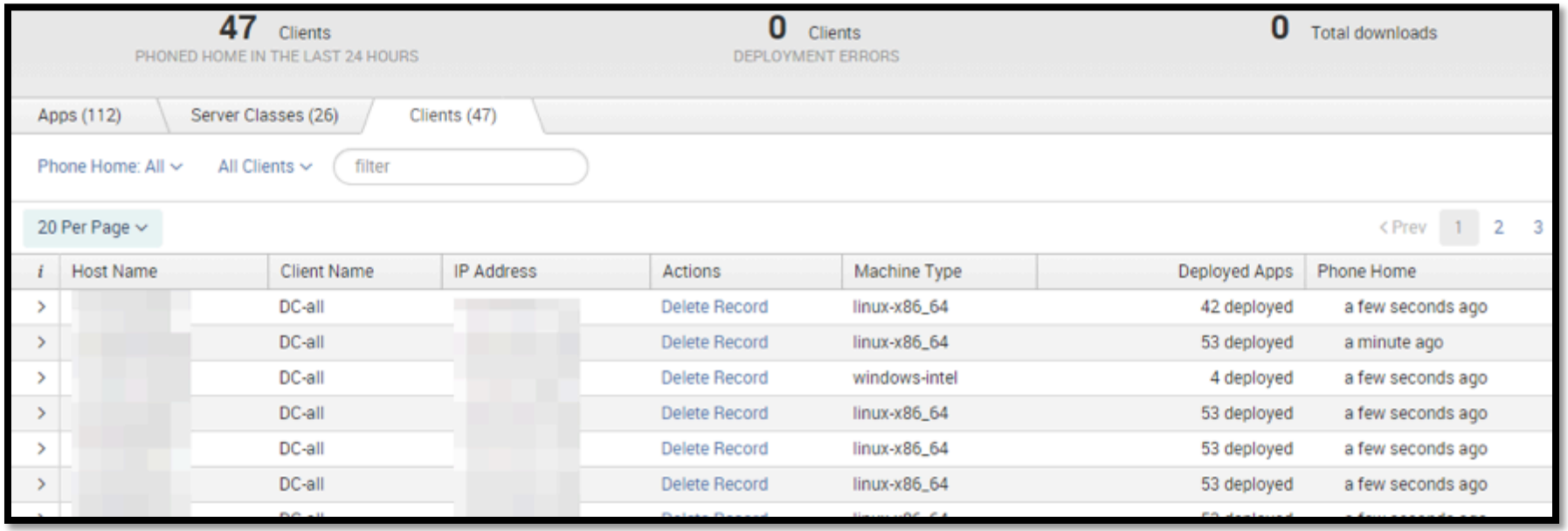
# Server: Reload (oops!)



# Server: Updating app installs

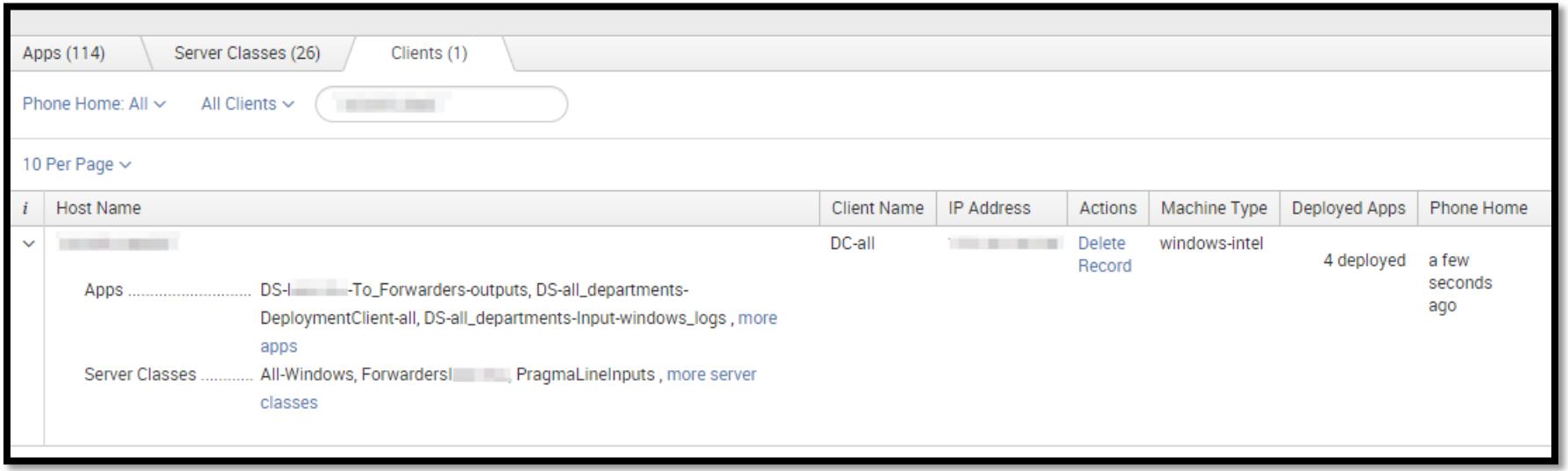


# Oh yeah, Forwarder Management




| i | Host Name | Client Name | IP Address | Actions | Machine Type | Deployed Apps | Phone Home |
|---|---|---|---|---|---|---|---|
| > | | DC-all | | Delete Record | linux-x86_64 | 42 deployed | a few seconds ago |
| > | | DC-all | | Delete Record | linux-x86_64 | 53 deployed | a minute ago |
| > | | DC-all | | Delete Record | windows-intel | 4 deployed | a few seconds ago |
| > | | DC-all | | Delete Record | linux-x86_64 | 53 deployed | a few seconds ago |
| > | | DC-all | | Delete Record | linux-x86_64 | 53 deployed | a few seconds ago |
| > | | DC-all | | Delete Record | linux-x86_64 | 53 deployed | a few seconds ago |

# Oh yeah, Forwarder Management



# Serverclass.xml

- Present on the clients
- Is a copy of the response from the deployment server to the deployment client
- Tells you which server classes the client thinks it belongs to, and which apps it thinks it should have
- But, it's all the way out on the endpoint
- If only we had a way to capture this data, and bring it to a central repository, perhaps index it so that we might be able to search it later ;-)


# Scaling Splunk Deployment Server

splunk>

# How much Deployment Server do I need?

- Not a lot of clients? Maybe a small VM
- Moar clients? MOAR SERVER!

# What if I have a lot of clients?

- Lots of clients = lots of check-ins
- Current maximum number of clients per Deployment server is:
- Windows: 500 – 2,000 (closer to the bottom one)
- Linux: 5,000 – 10,000
- (note that this is using reference hardware)
- By default, these clients check in every minute

# Change the default phoneHomeIntervalInSecs

- Found in deploymentclient.conf
- Defaults to 60 seconds
- How often are you changing those configs?
- Five minutes? Thirty minutes?
- Play the numbers game

# No, really, I mean ALOT of clients

- Currently no built-in solution
- May mean having multiple deployment servers

# Dedicated or Collocated?

- Keep in mind, there will be a lot of connections
- You don't want to run out of sockets
- What if you need to restart the deployment server?
- Remember, a deployment server can't be a client of itself
- Deployment server + license master works well
- If the server that the deployment server is on isn't a client of itself, you have to manage its configuration another way
- Can lead to configuration mismatches and inconsistency

# Load balancing?

- Does not work as expected
- Remember that hash? Yeah, that's the reason
- Not just the files and contents
- Includes modified time and other info
- If the hash doesn't match what the client currently has, it will grab a "new" version. This could mean a loop of restarts (fun!)

# FIN

- Some other talks to check out:
  - Avoid the SSLippery Slope of Default SSL - Duane Waddle and George Starcher
  - Using Lesser Known Commands in Splunk Search Processing Language (SPL) - Kyle Smith
  - Masters of IRC Community Panel
  - Building a Common Information Model (CIM) Compliant Technical Add-on (TA) – Brian Wooden and Jack Coates
  - Curating User Experience: Dashboarding Tips and Tricks – Sanford Owings
  - Getting The Most Out of Your Splunk License: Keeping the Junk Out of Splunk – David Paper
  - How Splunkd Works – Amrit Bath and Jag Karai


# THANK YOU

splunk>