SPLUNK TROUBLESHOOTING ESSENTIALS

A Field Guide to Data Onboarding

- Systematic Troubleshooting Methodology
- Database Audit Log Onboarding
- Performance Optimization
- Real-World Case Studies

MARCUS HOUSE

Electrical Engineer | Splunk Enterprise Architect
10+ Years Splunk Experience
Commercial, Federal & DoD

SPLUNK TROUBLESHOOTING ESSENTIALS

A Field Guide to Data Onboarding

Marcus House

Splunk Enterprise Architect

Splunk Troubleshooting Essentials: A Field Guide to Data Onboarding
Copyright 2026 Marcus House. All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form without prior written permission.

Splunk is a registered trademark of Splunk Inc. This book is not affiliated with or endorsed by Splunk Inc.

First Edition: January 2026
Published by GIC Engineering Consultants

Table of Contents

Preface

I wrote the original version of this troubleshooting guide over six years ago, during my early days as a junior Splunk administrator. At that time, I was constantly scrambling to diagnose why data was not flowing, why apps were not deploying, and why hosts seemed to vanish from the deployment server.

What I needed then, and what I hope this book provides you now, is a systematic, step-by-step approach to diagnosing the most common data onboarding problems in Splunk. Not theory. Not architecture diagrams. Just practical, field-tested procedures that work.

This updated edition reflects everything I have learned since then, including changes introduced in Splunk 9.x, modern diagnostic tools like btool, and lessons learned from managing enterprise-scale deployments in production environments.

Who This Book Is For

This book is designed for junior to mid-level Splunk administrators who need structured troubleshooting procedures, system administrators responsible for Splunk forwarder deployments, operations teams who manage data onboarding but are not Splunk specialists, and anyone who has ever stared at a Universal Forwarder wondering why it will not send data.

How to Use This Book

This is not a book meant to be read cover to cover. It is a field guide. Keep it at your desk, and when something breaks, flip to the relevant chapter.

Each problem chapter follows the same structure: problem description, step-by-step diagnostic procedure, commands with expected output, and administrator escalation actions. The chapters are organized by symptom, not by root cause. When you are troubleshooting at 2 AM, you know the symptom (no data is showing up), not the root cause.

Chapter 1: The Data Flow Model

Before diving into troubleshooting, you need a mental model of how data flows through a Splunk deployment. Every troubleshooting procedure in this book traces this path:

1. Source Data: Log files, metrics, or other data sitting on a server

2. Universal Forwarder (UF): Reads and forwards data to the next tier

3. Heavy Forwarder (HF): Optional intermediate tier for parsing, routing, or aggregation

4. Indexer(s): Stores data and makes it searchable

5. Search Head: Where you query and visualize data

When data does not appear in Splunk, the problem exists somewhere along this chain. Your job is to isolate which link is broken.

The Deployment Server Model

Configuration management adds another dimension:

Deployment Server (DS): Centralized configuration management

Deployment Client: Any Splunk instance that receives configurations from the DS

Server Class: A group of clients that receive the same apps

Deployment App: A bundle of configuration files pushed to clients

The deployment client periodically phones home to the deployment server, checks for new configurations, and downloads any updates. If this process breaks, your forwarders will not receive their input configurations and no data will flow.

Chapter 2: Prerequisites Checklist

Before troubleshooting any specific issue, run through this prerequisites checklist. Many mysterious problems turn out to be basic infrastructure issues.

1. Network Connectivity

Firewall rules must be in place for all Splunk instances. Never assume connectivity works. Verify it.

```
telnet <deployment_server> 8089
```

```
telnet <indexer_or_HF> 9997
```

2. Splunk Service Status

Verify Splunk is actually running on all relevant servers:

```
$SPLUNK_HOME/bin/splunk status
```

3. Receiving Port Configuration

If you are sending data to a Heavy Forwarder or Indexer, verify it is listening on the expected port:

```
netstat -an | grep 9997
```

4. File Ownership and Permissions

All files in $SPLUNK_HOME must be owned by the Splunk user:

```
ls -la $SPLUNK_HOME
```

FROM THE FIELD: The 755 Permission Trap

During a STIG compliance review, I discovered multiple Splunk servers had paths set to 755 permissions instead of the required restrictive settings. The fix was straightforward, but finding every non-compliant path across dozens of servers required systematic checking. Always verify permissions match your security baseline after any system changes.

5. Boot-Start Configuration

Splunk should be configured to start automatically at boot:

```
$SPLUNK_HOME/bin/splunk enable boot-start
```

6. Unique Hostnames

Every server sending data to Splunk must have a unique hostname. Duplicate hostnames cause data attribution issues and make troubleshooting nearly impossible.

7. Cloned VM GUID Issue

This is one of the most common and most overlooked problems in enterprise deployments. When you clone a virtual machine that has Splunk installed, the clone inherits the original machine GUID. This causes deployment server showing incorrect forwarder counts, license volume attribution errors, and conflicting configurations.

```
rm $SPLUNK_HOME/etc/instance.cfg
```

```
$SPLUNK_HOME/bin/splunk restart
```

Chapter 3: Essential Diagnostic Tools

Before diving into specific problems, master these diagnostic tools. They will be your primary weapons throughout every troubleshooting session.

btool: The Configuration Validator

The btool command is the single most important troubleshooting tool for Splunk configuration issues. It shows you the merged, final configuration that Splunk will actually use after all the layering and precedence rules are applied.

```
$SPLUNK_HOME/bin/splunk btool inputs list --debug
```

```
$SPLUNK_HOME/bin/splunk btool outputs list --debug
```

```
$SPLUNK_HOME/bin/splunk btool deploymentclient list --debug
```

splunk list Commands

These commands show runtime status:

```
$SPLUNK_HOME/bin/splunk list forward-server
```

```
$SPLUNK_HOME/bin/splunk list monitor
```

```
$SPLUNK_HOME/bin/splunk list deploy-clients
```

Log Analysis with grep

The splunkd.log file contains diagnostic information about everything Splunk does. Key log patterns to search for:

```
grep -i error $SPLUNK_HOME/var/log/splunk/splunkd.log
```

```
grep -i "phonehome" $SPLUNK_HOME/var/log/splunk/splunkd.log
```

```
grep -i "connection" $SPLUNK_HOME/var/log/splunk/splunkd.log
```

SPL Queries for Internal Logs

When command-line tools are not enough, search the _internal index directly:

```
index=_internal sourcetype=splunkd ERROR
```

```
index=_internal sourcetype=splunkd component=TcpOutputProc
```

```
index=_internal sourcetype=splunkd "deployment client"
```

Chapter 4: No Deployment Apps on Forwarder

This is one of the most common issues in new Splunk deployments. The forwarder is installed and pointed at the deployment server, but configurations never arrive.

Step 1: Verify Deployment Client Configuration on Endpoint

```
$SPLUNK_HOME/bin/splunk btool deploymentclient list --debug
```

Expected output should include targetUri pointing to your deployment server.

Step 2: Check for Conflicting Configuration Files

```
find $SPLUNK_HOME -name deploymentclient.conf
```

You should see only ONE file. Multiple files can cause conflicts.

Step 3: Verify Phone Home to Deployment Server

```
grep -i phonehome $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -20
```

Successful communication shows that the client is communicating and there is no firewall blockage.

Step 4: Verify Deployment Apps Exist on Endpoint

```
ls -la $SPLUNK_HOME/etc/apps/
```

Apps pushed from the deployment server should appear here.

Step 5: Administrator Verification

If Steps 1-4 pass but apps are still not appearing, verify in the DS Web GUI that the client appears in Forwarder Management, the expected apps are assigned to the correct server class, and the server class whitelist/blacklist rules include this client.

Chapter 5: All Data Stopped from Heavy Forwarder

When all data stops flowing through a Heavy Forwarder, the problem is typically at the HF itself, not the individual Universal Forwarders sending to it. This chapter covers systematic diagnosis of HF failures.

```
$SPLUNK_HOME/bin/splunk status
```

```
ps aux | grep splunk
```

If Splunk is not running, check /var/log/messages or journalctl for why it stopped.

Step 2: Verify HF is Listening on Receiving Port

```
netstat -an | grep 9997
```

Expected output:

```
tcp  0  0  0.0.0.0:9997  0.0.0.0:*  LISTEN
```

If the port is not listening, check inputs.conf on the HF for the [splunktcp://9997] stanza.

Step 3: Check for Blocked Queues

Heavy Forwarders can experience queue blockage when downstream indexers are unavailable or slow:

```
$SPLUNK_HOME/bin/splunk list inputstatus
```

Look for queues showing high fill percentages. If queues are blocked, data backs up and eventually drops.

> **FROM THE FIELD: The Blocked Queue Mystery**
>
> *I repeatedly encountered an issue where Splunk queues would become blocked, causing data loss. After multiple incidents, I identified that the root cause was downstream indexer capacity. When indexers fell behind, the HF queues filled up. The corrective action was twofold: increase indexer capacity and configure queue limits to prevent cascading failures.*

Step 4: Verify Source Data is Arriving at HF

On a Universal Forwarder sending to this HF, verify data exists and the UF can reach the HF:

```
telnet <HF_hostname> 9997
```

If telnet fails, the issue is network connectivity between the UF and HF.

Step 5: Check TCP Output Status

```
$SPLUNK_HOME/bin/splunk list tcp
```

Look for errors like "Connection refused", "Connection reset", or "Network is unreachable".

Step 6: Check Forward Server Status

```
watch -n 5 '$SPLUNK_HOME/bin/splunk list forward-server'
```

The watch command will refresh every 5 seconds, allowing you to see if the connection state changes.

Step 7: Check HF Internal Logs

```
grep -i error $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -50
```

```
grep -i "queue" $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -20
```

```
$SPLUNK_HOME/var/log/splunk/splunkd.log | tail -20
```

Step 8: Verify Disk Space

Heavy Forwarders need adequate disk space for queue persistence:

```
df -h $SPLUNK_HOME
```

WARNING: A full disk on an HF will cause immediate data loss. Splunk cannot queue data if there is no disk space. Monitor HF disk usage proactively.

Administrator Escalation Actions

If the above steps do not resolve the issue:

1. Check firewall rules between HF and indexers
2. Verify indexer cluster health (if applicable)
3. Review any recent configuration changes to the HF
4. Check for certificate expiration if using SSL

```
index=_internal host=<HF_hostname> sourcetype=splunkd ERROR
```

Chapter 6: Data Never Showed Up from Some UFs

This scenario indicates an issue specific to certain forwarders, not a systemic problem. Focus troubleshooting on the affected UFs, but also look for patterns - are all affected hosts in the same network segment, running the same OS, or sharing some other characteristic?

```
telnet <deployment_server> 8089
```

```
telnet <indexer_or_HF> 9997
```

Both connections must succeed. If telnet fails, the issue is network connectivity or firewall rules.

Step 2: Verify Splunk is Running

```
$SPLUNK_HOME/bin/splunk status
```

If Splunk is not running, start it and check logs for why it stopped.

> **FROM THE FIELD: Running But Not Sending**
>
> *I encountered a puzzling situation where Splunk showed as running on multiple hosts, but no data was being received. The splunk status command showed everything healthy. The root cause turned out to be a configuration issue where inputs.conf was present but malformed - Splunk started successfully but the inputs were not being processed. Always verify with btool, not just service status.*

Step 3: Verify App Deployment

```
ls -la $SPLUNK_HOME/etc/apps/
```

Verify the apps that define inputs and outputs exist. If missing, troubleshoot deployment server communication (see Chapter 4).

Step 4: Verify Data Exists and Splunk Can Read It

```
ls -la /var/log/
```

```
head -20 /var/log/messages
```

Confirm the log files exist and contain current data.

Step 5: Check File Permissions

```
ls -la /var/log/secure
```

```
id splunk
```

The Splunk user must have read access to monitored files. This is a common issue with security-sensitive logs.

> **TIP:** On Linux systems, add the splunk user to the adm group to grant read access to most system logs: usermod -aG adm splunk

Step 6: Check Monitored Inputs

```
$SPLUNK_HOME/bin/splunk list monitor
```

The expected paths should appear in this list. If missing, the inputs.conf is not being applied.

Step 7: Check for Connection Failures

```
grep -i "connection" $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -30
```

Step 8: Check TCP Output and Forward Server

```
$SPLUNK_HOME/bin/splunk list tcp
```

```
$SPLUNK_HOME/bin/splunk list forward-server
```

Step 9: Force a Restart and Watch Logs

```
$SPLUNK_HOME/bin/splunk restart
```

```
tail -f $SPLUNK_HOME/var/log/splunk/splunkd.log
```

Watch for errors during startup that might indicate the problem.

Common Root Causes for Selective UF Failures

1. Network segmentation - hosts in a specific VLAN cannot reach Splunk infrastructure
2. Host-based firewall enabled on affected systems
3. Splunk user account locked or password expired on specific hosts
4. Disk full on affected hosts preventing Splunk from writing state files
5. SELinux or AppArmor blocking Splunk on specific systems

Chapter 7: Data Stopped from UF (Direct to Indexer)

This simpler architecture has fewer potential failure points. The troubleshooting process is streamlined because there is no intermediate Heavy Forwarder. However, this also means there is no buffer - if the UF cannot reach the indexer, data is lost.

> **FROM THE FIELD: The Weekly PowerShell Problem**
>
> *I had a recurring issue where approximately a dozen Windows hosts would stop sending PowerShell logs each week. Every week I would restore functionality, and the following week it would break again. The root cause turned out to be a group policy that was resetting PowerShell logging configuration, which conflicted with the Splunk inputs.conf settings. The permanent fix required coordinating with the Windows team to align group policy with Splunk requirements.*

Diagnostic Steps

Follow these steps in order:

Step 1: Verify Network Connectivity to Indexer

```
telnet <indexer> 9997
```

If this fails, you have found your problem. Check firewalls and network routing.

Step 2: Verify Splunk is Running

```
$SPLUNK_HOME/bin/splunk status
```

Step 3: Verify Network Resolution to Deployment Server

```
telnet <deployment_server> 8089
```

Step 4: Verify App Deployment

```
ls -la $SPLUNK_HOME/etc/apps/
```

Confirm the apps containing inputs.conf and outputs.conf are present.

Step 5: Check Monitored Inputs

```
$SPLUNK_HOME/bin/splunk list monitor
```

```
$SPLUNK_HOME/bin/splunk btool inputs list --debug
```

Step 6: Verify Data Exists and Permissions

Ensure the log files exist and the Splunk user has read permissions.

```
ls -la /path/to/monitored/logs
```

Step 7: Check Forward Server Status

```
$SPLUNK_HOME/bin/splunk list forward-server
```

Expected output:

```
Active forwards:     <indexer>:9997         status=connected
```

Step 8: Check TCP Output Status

```
$SPLUNK_HOME/bin/splunk list tcp
```

Look for connection errors or data transmission failures.

FROM THE FIELD: The xd_winevents Alert

An alert I configured flagged a list of hosts that stopped sending xd_winevents logs. These were Windows security event logs - critical for security monitoring. Investigation showed the Windows Event Log service had been restarted on these hosts, which temporarily broke the WMI connection Splunk uses to read events. A simple Splunk restart on each affected host resolved the issue, but it highlighted the need for monitoring not just Splunk health, but also the health of log sources themselves.

Step 9: Check for Specific Log Type Issues

If only certain log types are missing (like PowerShell or security logs), verify the specific input configuration:

```
$SPLUNK_HOME/bin/splunk btool inputs list --debug | grep -A5 WinEventLog
```

Windows-Specific Checks

For Windows Universal Forwarders, also verify:

1. The Splunk service account has "Log on as a service" rights

2. The service account can read the Security event log (requires specific permissions)

3. WMI service is running and healthy

4. No group policy is overriding local Splunk configurations

Chapter 8: Unable to Achieve Handshaking

The handshake is the initial communication between a deployment client and the deployment server. Without a successful handshake, no apps will be deployed.

Step 1: Verify Network Connectivity

```
telnet <deployment_server> 8089
```

If this fails, the issue is network connectivity or firewall rules blocking port 8089.

Step 2: Verify Deployment Client Configuration

```
$SPLUNK_HOME/bin/splunk btool deploymentclient list --debug
```

Expected output should include targetUri pointing to your deployment server.

Step 3: Check for Conflicting Files

```
find $SPLUNK_HOME -name deploymentclient.conf
```

Multiple deploymentclient.conf files can cause precedence conflicts. Ideally, you should see only one file.

Step 4: Check Phone Home Logs

```
grep -i phonehome $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -20
```

Successful handshake:

```
INFO  DeploymentClient - Handshake done.
```

Failed handshake (example):

```
ERROR DeploymentClient - Handshake failed. Will retry.
```

Step 5: Force a Phone Home

```
$SPLUNK_HOME/bin/splunk reload deploy-server
```

Then immediately check logs for handshake activity.

Step 6: Verify Deployment Server is Running

```
$SPLUNK_HOME/bin/splunk show deploy-server
```

Chapter 9: Some Data Showing Up But Not All

Partial data issues are often the most frustrating to troubleshoot because the infrastructure mostly works. The problem is usually with file tracking, CRC issues, or selective input configuration.

> **FROM THE FIELD: The Incorrect Timestamp Mystery**
>
> *I discovered that CSV logs being ingested had incorrect timestamps - they were hours off from the actual event times. After investigation, I identified that the props.conf TIME_FORMAT setting did not match the actual timestamp format in the data. The fix required analyzing sample data carefully and updating the timestamp extraction configuration.*

Initial Diagnostic Steps

Step 1: Verify the App Managing the Missing Data Exists

```
ls -la $SPLUNK_HOME/etc/apps/
```

Ensure the app that defines inputs for the missing data type is present.

Step 2: Verify Data Exists and Splunk Can Read It

```
ls -la /path/to/data
```

```
head -20 /path/to/data/somefile.log
```

Step 3: Check Monitored Inputs

```
$SPLUNK_HOME/bin/splunk btool inputs list --debug
```

Verify that the paths for the missing data are actually being monitored.

Check for CRC Errors

CRC (Cyclic Redundancy Check) issues are a common cause of partial data ingestion:

```
grep -i crc $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -20
```

Understanding CRC Issues

Splunk uses the first 256 bytes (and last 256 bytes) of a file to compute a CRC for tracking which files have been indexed. This prevents re-indexing the same data after a forwarder restart.

Common CRC problems: files smaller than 256 bytes, duplicate content in file headers (identical first 256 bytes), file system read errors, files with headers that change frequently.

CRC Solutions

Solution 1: Add crcSalt to Differentiate Similar Files

```
[monitor:///var/log/app/*.log]
```

```
crcSalt = <SOURCE>
```

Solution 2: Increase CRC Calculation Length

```
[monitor:///var/log/app/*.log]
```

```
initCrcLength = 1024
```

Solution 3: Check Fishbucket Status

```
$SPLUNK_HOME/bin/splunk cmd btprobe -d
$SPLUNK_HOME/var/lib/splunk/fishbucket/splunk_private_db --file
/path/to/file
```

Chapter 10: Quick Reference

This chapter provides a condensed reference of the most commonly used diagnostic commands and searches.

Diagnostic Commands

```
$SPLUNK_HOME/bin/splunk status
```

```
$SPLUNK_HOME/bin/splunk list forward-server
```

```
$SPLUNK_HOME/bin/splunk list monitor
```

```
$SPLUNK_HOME/bin/splunk list deploy-clients
```

```
$SPLUNK_HOME/bin/splunk btool inputs list --debug
```

```
$SPLUNK_HOME/bin/splunk btool outputs list --debug
```

```
$SPLUNK_HOME/bin/splunk btool deploymentclient list --debug
```

Log Analysis Commands

```
grep -i error $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -50
```

```
grep -i phonehome $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -20
```

```
grep -i connection $SPLUNK_HOME/var/log/splunk/splunkd.log | tail -30
```

SPL Searches

```
index=_internal sourcetype=splunkd ERROR
```

```
index=_internal sourcetype=splunkd component=TcpOutputProc
```

```
index=_internal host=<forwarder> "deployment client"
```

Default Ports

8089 - Management/REST API (deployment server communication)

9997 - Data receiving (forwarders to indexers/HF)

8000 - Splunk Web interface

8191 – KVStore

Troubleshooting Decision Tree

1. Is Splunk running? No: Start Splunk and check for errors. Yes: Continue.

2. Can you telnet to the receiving port? No: Check firewall rules. Yes: Continue.

3. Is the deployment handshake successful? No: See Chapter 8. Yes: Continue.

4. Are the expected apps deployed? No: See Chapter 4. Yes: Continue.

5. Are the expected inputs monitored? No: Check inputs.conf and btool. Yes: Continue.

6. Is the forward-server status active? No: Check outputs.conf. Yes: Check for CRC issues.

About the Author

Marcus House is a Splunk Enterprise Architect with extensive experience in Commercial, Department of Defense, and Federal Government environments. He has spent years managing enterprise-scale Splunk deployments, troubleshooting complex data onboarding issues, and developing operational procedures that help teams maintain reliable observability infrastructure.

This book grew out of a troubleshooting guide Marcus wrote during his early days as a junior administrator. A guide he wishes he had when he first started. It has been updated and expanded to reflect the lessons learned from years of production operations and the changes introduced in modern Splunk releases.

Marcus is the founder of GIC Engineering Consultants, where he provides Splunk consulting services to commercial and government clients.

For consulting inquiries: marcus@gicengineeringconsultants.com
Website: gicengineeringconsultants.com
LinkedIn: linkedin.com/in/mhouse3

- End -