LOGZILLA DOCUMENTATION

PCI Compliance

Enable PCI DSS Requirement 10.3 raw message preservation in LogZilla, storing unaltered logs in a segregated directory for audit trail integrity

Administration · Generated May 3, 2026 · logzilla.ai/docs/administration/pci-compliance

LogZilla provides PCI DSS compliance features to meet audit trail requirements for secure log management. The system can maintain unaltered copies of raw log messages in a segregated directory structure, supporting compliance with PCI DSS Requirement 10.3 for audit trail integrity.

Critical Warning: PCI compliance logging will more than **double the log storage requirements** as it creates complete copies of all incoming log messages without deduplication. Monitor disk usage closely and implement log rotation and retention policies immediately after enabling this feature to prevent disk space exhaustion.

PCI Compliance Overview

LogZilla's PCI compliance feature provides:

- **Raw Message Preservation**: Stores original, unmodified log messages
- **Segregated Storage**: Maintains compliance logs separate from operational logs
- **Structured Organization**: Organizes logs by date for easy management
- **Audit Trail Integrity**: Preserves complete message content for compliance audits

Enabling PCI Compliance Logging

Web Interface Configuration

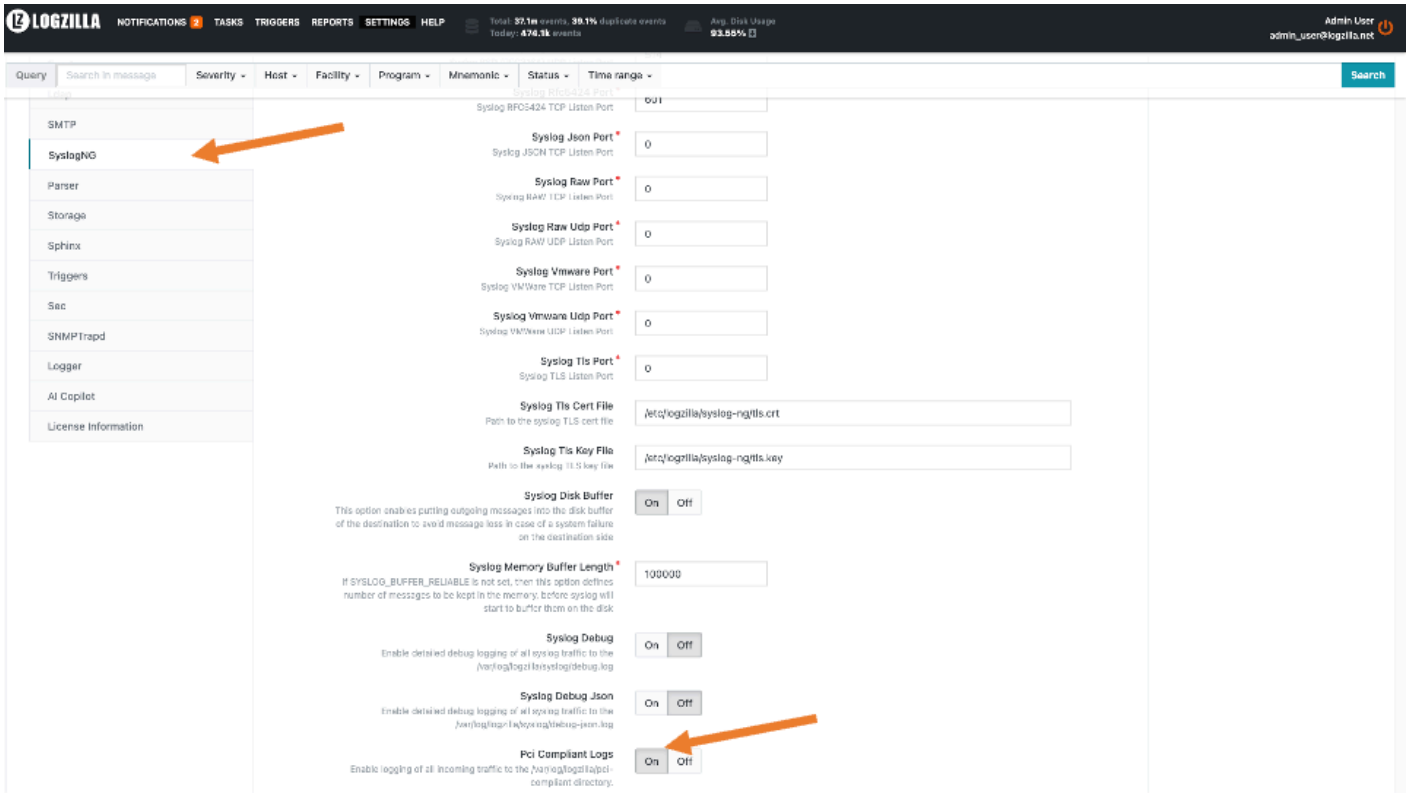
The **recommended approach** for enabling PCI compliance logging is through the LogZilla web interface, which provides a user-friendly experience with immediate validation and status feedback:

Access PCI Settings:

- Log into the LogZilla web interface as an administrator
- Navigate to **Settings** → **Syslog Daemon** → **PCI Compliance**

Enable PCI Logging:

- Locate the **"PCI Compliant Logs"** setting
- Toggle the setting to **"On"**
- Enabling takes effect immediately



Verify Log Generation:

- Check that PCI log files begin appearing in the designated directory under `/var/log/logzilla/pci-compliant/`

Command Line Configuration

Note: Command-line configuration is provided for advanced users who specifically require shell access. Most users should use the web interface above, which provides the same functionality with better user experience.

For users who prefer command-line administration:

```
# Enable PCI compliance logging
logzilla settings update PCI_COMPLIANT_LOGS=true

# Verify the setting
logzilla settings list syslogng | grep PCI
```

PCI Logging Deactivation

Note: Disabling PCI logging stops new log collection but does not automatically remove existing PCI compliance log files. Administrators must manually manage existing files according to their retention policies.

For UI users:

Access PCI Settings:

- Log into the LogZilla web interface as an administrator
- Navigate to **Settings** → **Syslog Daemon** → **PCI Compliance**

Disable PCI Logging:

- Locate the **"PCI Compliant Logs"** setting
- Toggle the setting to **"Off"**
- Disabling takes effect immediately

For command-line users:

```
# Disable PCI compliance logging
logzilla settings update PCI_COMPLIANT_LOGS=false

# Verify the setting is disabled
logzilla settings list syslogng | grep PCI
```

Log Storage and Organization

Directory Structure

PCI compliance logs are stored in a structured format:

```
/var/log/logzilla/pci-compliant/
├── 2025-09/
│   ├── 2025-09-25.log
│   ├── 2025-09-26.log
│   └── ...
├── 2025-08/
│   ├── 2025-08-25.log
│   ├── 2025-08-26.log
│   └── ...
└── checksums (if using external integrity monitoring)
```

Log Content

PCI compliance logs contain:

- **Raw Messages**: Complete, unaltered syslog messages (`$RAWMSG`)
- **Original Format**: Messages exactly as received by LogZilla (standard BSD syslog format, RFC 3164)

- **No Processing**: No parsing, normalization, or modification applied
- **Complete Audit Trail**: All incoming log data for compliance verification

File Integrity and Checksums

Built-in Security

LogZilla provides basic file security:

- **File Permissions**: Logs written with `0644` permissions
- **Directory Permissions**: Directories created with `0755` permissions
- **Segregated Storage**: Compliance logs isolated from operational data

External Integrity Monitoring

Important: The following scripts are **administrator-created examples** and are not built-in LogZilla utilities. These scripts must be created, tested, and maintained by system administrators according to their specific compliance requirements.

For enhanced compliance, implement external file integrity monitoring:

```
# Create daily checksum and compression script
cat << 'EOF' > /usr/local/bin/logzilla-pci-checksum.sh
#!/bin/bash
# LogZilla PCI Compliance Checksum Script
# This is an EXAMPLE script - customize for your environment

PCI_DIR="/var/log/logzilla/pci-compliant"
CHECKSUM_FILE="$PCI_DIR/checksums"
YESTERDAY=$(date -d "yesterday" +%Y-%m-%d)
YESTERDAY_MONTH=$(date -d "yesterday" +%Y-%m)
LOG_FILE="$PCI_DIR/$YESTERDAY_MONTH/$YESTERDAY.log"

# Process yesterday's log file if it exists
if [ -f "$LOG_FILE" ]; then
    echo "$(date): Processing $YESTERDAY.log" >> "$CHECKSUM_FILE"

    # Create SHA256 checksum of original file (before compression)
    sha256sum "$LOG_FILE" >> "$CHECKSUM_FILE"

    # Compress the log file
    gzip "$LOG_FILE"

    # Create SHA256 checksum of compressed file
    sha256sum "$LOG_FILE.gz" >> "$CHECKSUM_FILE"

    echo "$(date): Completed processing $YESTERDAY.log.gz" >> "$CHECKSUM_FILE"
```

```
    # Optional: Set immutable attribute to prevent tampering (requires ext2/3/4)
    # chattr +i "$LOG_FILE.gz"
else
    echo "$(date): Warning - Log file $LOG_FILE not found" >> "$CHECKSUM_FILE"
fi
EOF

# Make script executable
chmod +x /usr/local/bin/logzilla-pci-checksum.sh
```

Automated Daily Processing

Set up automated daily processing:

```
# Create cron job for daily checksum processing
cat << 'EOF' > /etc/cron.d/logzilla-pci
# LogZilla PCI Compliance - Daily log processing
# Runs at 12:01 AM daily to process previous day's logs
# IMPORTANT: This is an example - test thoroughly before production use
1 0 * * * root /usr/local/bin/logzilla-pci-checksum.sh >> /var/log/logzilla/pci-processing.log 2>&1

# Optional: Weekly verification of checksums
# 0 2 * * 0 root /usr/local/bin/logzilla-pci-verify.sh >> /var/log/logzilla/pci-verification.log 2>&1
EOF
```

Compliance Management

Log Retention

Note: The following retention script is an **example template** that administrators should customize for their specific environment and compliance requirements.

Implement appropriate retention policies:

```
# Example: Retain PCI logs for 1 year, then archive
cat << 'EOF' > /usr/local/bin/logzilla-pci-retention.sh
#!/bin/bash
# LogZilla PCI Log Retention Script

PCI_DIR="/var/log/logzilla/pci-compliant"
ARCHIVE_DIR="/backup/logzilla-pci-archive"
RETENTION_DAYS=365

# Create archive directory if it doesn't exist
```

```
mkdir -p "$ARCHIVE_DIR"

# Find and archive logs older than retention period
find "$PCI_DIR" -name "*.log.gz" -mtime +$RETENTION_DAYS -type f \
    -exec mv {} "$ARCHIVE_DIR/" \;

# Log retention activity
echo "$(date): PCI log retention completed" >> "$PCI_DIR/retention.log"
EOF

chmod +x /usr/local/bin/logzilla-pci-retention.sh
```

Backup Procedures

Note: The following backup script is an **example template** that administrators should adapt for their specific backup infrastructure and security requirements.

Implement secure backup procedures:

```
# Example backup script
cat << 'EOF' > /usr/local/bin/logzilla-pci-backup.sh
#!/bin/bash
# LogZilla PCI Compliance Backup Script

PCI_DIR="/var/log/logzilla/pci-compliant"
BACKUP_DEST="/secure/backup/location"
DATE=$(date +%Y%m%d)

# Create compressed backup
tar -czf "$BACKUP_DEST/logzilla-pci-$DATE.tar.gz" -C "$PCI_DIR" .

# Verify backup integrity
if [ $? -eq 0 ]; then
    echo "$(date): PCI backup completed successfully" >> "$PCI_DIR/backup.log"
else
    echo "$(date): PCI backup FAILED" >> "$PCI_DIR/backup.log"
fi
EOF

chmod +x /usr/local/bin/logzilla-pci-backup.sh
```

Monitoring and Verification

Check PCI Logging Status

```
# Verify PCI logging is enabled
logzilla settings list syslogng | grep PCI_COMPLIANT_LOGS

# Check recent PCI log files
ls -la /var/log/logzilla/pci-compliant/$(date +%Y-%m)/

# Monitor log file growth
watch "ls -lh /var/log/logzilla/pci-compliant/$(date +%Y-%m)/$(date +%Y-%m-%d).log"
```

Verify Log Content

```
# Sample recent PCI log entries
tail -n 10 /var/log/logzilla/pci-compliant/$(date +%Y-%m)/$(date +%Y-%m-%d).log

# Check for specific time period
grep "$(date +%H:%M)" /var/log/logzilla/pci-compliant/$(date +%Y-%m)/$(date +%Y-%m-%d).log
```

Integrity Verification

```
# Verify checksums
md5sum -c /var/log/logzilla/pci-compliant/checksums

# Check for file modifications
find /var/log/logzilla/pci-compliant -name "*.log.gz" -newer /var/log/logzilla/pci-
compliant/checksums
```

Best Practices

Security

- **Access Control**: Restrict access to PCI log directories
- **File Permissions**: Maintain appropriate file and directory permissions
- **Secure Backup**: Store backups in secure, encrypted locations
- **Regular Audits**: Periodically verify log integrity and completeness

Performance

- **Disk Space**: Monitor disk usage in PCI log directories
- **I/O Impact**: Consider I/O impact of dual logging on system performance
- **Log Rotation**: Implement timely compression and archival
- **Network Storage**: Consider dedicated storage for compliance logs

Compliance

- **Retention Policies**: Implement policies meeting regulatory requirements
- **Documentation**: Maintain documentation of PCI logging procedures
- **Regular Testing**: Test backup and recovery procedures
- **Audit Preparation**: Ensure logs are readily available for compliance audits

Troubleshooting

Common Issues

PCI Logging Not Working:

```
# Check if setting is enabled
logzilla settings list syslogng | grep PCI_COMPLIANT_LOGS
```

Missing Log Files:

```
# Check directory permissions
ls -ld /var/log/logzilla/pci-compliant/
```

Disk Space Issues:

```
# Check available space
df -h /var/log/logzilla/pci-compliant/

# Find largest log files
find /var/log/logzilla/pci-compliant -type f -exec ls -lh {} \; | sort -k5 -hr | head -10
```