Professional Services Access Audit - Quick Start Guide

Overview

This PowerShell script performs automated access audits across **Clio**, **NetSuite**, and **QuickBooks Online** to help professional services firms maintain compliance and identify segregation of duties violations.

What the Script Does

🔍 **Audits User Access** across all three platforms 📊 **Generates CSV Reports** with normalized permission data
⚠️ **Identifies Risk Patterns** like terminated employees with active access 🛡️ **Flags SoD Violations** such as users with both admin and financial access 📈 **Creates Summary Statistics** for executive reporting

Prerequisites

- **PowerShell 7+** (cross-platform)
- **API Access** to your Clio, NetSuite, and/or QuickBooks accounts
- **Administrative permissions** to create API credentials

Quick Setup (5 Minutes)

Step 1: Download the Script

```
# Save the main script as AccessAudit.ps1
# Save the config template as audit_config.json
```

Step 2: Configure API Credentials

Edit `audit_config.json` with your credentials:

Clio Setup:

```
"Clio": {
  "BaseURL": "https://app.clio.com/api/v4",
  "AccessToken": "YOUR_ACTUAL_TOKEN_HERE"
}
```

NetSuite Setup:

```
"NetSuite": {
  "AccountId": "123456_SB1",
  "ConsumerKey": "your_consumer_key",
  "ConsumerSecret": "your_consumer_secret",
  "TokenId": "your_token_id",
  "TokenSecret": "your_token_secret"
}
```

QuickBooks Setup:

```
"QuickBooks": {
  "BaseURL": "https://sandbox-quickbooks.api.intuit.com",
  "AccessToken": "your_access_token",
  "CompanyId": "123456789"
}
```

Step 3: Run the Audit

```
# Run full audit (all platforms)
.\AccessAudit.ps1

# Audit specific platforms only
.\AccessAudit.ps1 -Platforms "Clio","NetSuite"

# Custom output location
.\AccessAudit.ps1 -OutputPath "C:\Audit_Reports"
```

Generated Reports

The script creates several CSV files:

📋 **Combined_Access_Audit_[DATE].csv** - All users across platforms 🔍 **[Platform]*Access_Audit*[DATE].csv** - Platform-specific reports ⚠️ **Risk_Analysis_[DATE].csv** - High-risk access patterns 📊 **Audit_Summary_[DATE].csv** - Executive summary statistics

Key Data Fields

Each audit record includes:

Field	Description
Platform	Clio, NetSuite, or QuickBooks
UserName	Full name of user
Email	User email address
Role	System role/title
Status	Active/Inactive
AdminRights	Yes/No/Unknown
FinancialAccess	Access to billing/accounting
ClientDataAccess	Access to client information
EmployeeStatus	Active/Terminated
Permissions	Detailed permission list

Common API Setup Issues

Clio Authentication

❌ **Problem:** 401 Unauthorized

✅ **Solution:** Ensure you're using the correct region URL (US vs EU)

NetSuite Connection

❌ **Problem:** OAuth signature errors

✅ **Solution:** Verify all TBA credentials and account ID format

QuickBooks Rate Limits

❌ **Problem:** 429 Too Many Requests

✅ **Solution:** Script includes automatic retry logic

Getting API Credentials

Clio Developer Account

1. Visit Clio Developer Portal
2. Create developer account and new application
3. Follow OAuth 2.0 flow to get access token
4. Required scopes: `users`, `contacts`, `matters`

NetSuite Token-Based Authentication

1. **Setup > Integration > Manage Integrations > New**
2. Create integration with Token-Based Authentication
3. **Setup > Users/Roles > Manage Roles** - Create role with Web Services (Full)
4. **Setup > Users/Roles > Access Tokens > New**
5. Generate token for your integration and role

QuickBooks Developer Portal

1. Visit Intuit Developer Portal
2. Create app with QuickBooks Online Accounting scope
3. Use OAuth 2.0 Playground to get tokens
4. Find Company ID with Ctrl+Alt+? in QuickBooks

Security Best Practices

🔐 **Never commit credentials** to version control

🔄 **Rotate tokens regularly** (every 90 days)

📝 **Log all access** and monitor usage

🔒 **Use service accounts** with minimal permissions

🛡️ **Encrypt config files** in production environments

Troubleshooting

Script won't run:

```powershell
# Check PowerShell version
$PSVersionTable.PSVersion

# Set execution policy if needed
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

No data returned:

- Verify API credentials are correct
- Check network connectivity to API endpoints
- Ensure service account has proper permissions
- Review error messages in PowerShell output

Rate limiting issues:

- Script includes exponential backoff
- Spread audits across different times
- Contact API providers for higher limits

Sample Output

```
🚀 Professional Services Access Audit Tool
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
Platforms: Clio, NetSuite, QuickBooks
Output: .\audit_reports

🔍 Fetching Clio users from: https://app.clio.com/api/v4/users
✅ Retrieved 25 Clio users
🔍 Fetching NetSuite employees...
✅ Retrieved 18 NetSuite employees
🔍 Fetching QuickBooks employees...
✅ Retrieved 22 QuickBooks employees

📊 AUDIT SUMMARY
Total Users Audited: 65
  • Clio: 25
  • NetSuite: 18
  • QuickBooks: 22

Access Status:
  • Active: 58
  • Inactive: 7
  • Admin Rights: 8
  • Financial Access: 12
  • Terminated w/ Access: 2
```

Next Steps

1. **Review Risk Analysis** for immediate action items
2. **Validate SoD violations** with business stakeholders
3. **Remove access** for terminated employees

4. **Schedule regular audits** (quarterly recommended)
5. **Document remediation** for compliance evidence

Support

For technical issues:

- Check PowerShell error messages
- Verify API credential setup
- Test individual platform connections
- Review audit logs for patterns

This audit tool helps professional services firms maintain security compliance while providing actionable insights for access governance.