SpnManager Adoption Guide

Detent Point LLC

Version 0.4.0 — 2026-09-05

- Get-SapSpnAudit
- Get-SharePointSpnCandidate
- Get-SmbSpnCandidate
- Get-SpnProvider
- Get-SqlSpnAccount
- Get-SqlSpnDiscoveryEngine
- Get-SqlSpnInfrastructure
- Get-SqlSpnPolicyConfig
- Get-SsasSpnCandidate
- Get-SsrsSpnCandidate
- Get-UnconstrainedDelegationAudit
- Get-WinRmSpnCandidate
- Invoke-SpnAuditEngine
- Invoke-SpnDuplicateScan
- Invoke-SpnExecutionEngine
- Invoke-SpnForestAudit
- Invoke-SpnLifecycle
- Invoke-SqlSpnExecutionEngine
- New-AdcsSpnAuditPlan
- New-AdcsSpnPlan
- New-AdfsSpnPlan
- New-DfsSpnAuditPlan
- New-DfsSpnPlan
- New-DnsSpnAuditPlan
- New-ExchangeSpnAuditPlan
- New-ExchangeSpnPlan
- New-IisSpnPlan
- New-PrintSpnAuditPlan
- New-RdpSpnAuditPlan
- New-SharePointSpnPlan
- New-SmbSpnAuditPlan
- New-SqlSpnPlan
- New-SsasSpnPlan
- New-SsrsSpnPlan
- New-WinRmSpnAuditPlan
- Remove-SqlSpn
- Set-SqlSpnPolicyConfig
- Show-SqlSpnDiagnostic
- Start-SpnManager
- Start-SqlSpnConfiguration
- Start-SqlSpnManager
- Test-SpnAuditPlan
- Test-SpnPlan
- Test-SqlSpnKerberosAuth
- Test-SqlSpnPlan

About this guide

SpnManager finds and fixes Kerberos Service Principal Name problems in Active Directory. This guide covers every phase of using it, from deciding whether you need it through to running it on a schedule.

It is written for more than one reader. Each command reference in Part 2 carries six views of the same command — one for management, one for the practitioner running it, one for someone learning the area, one for software approval, one listing dependencies, and one for compliance. Read the one that matches your question.

What this version covers: 74 commands across 26 providers. 15 of those providers can write to Active Directory; 11 are audit-only and cannot change anything by design.

> **Licence.** SpnManager is proprietary software, licensed and not sold. Terms are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Part 1 — Using SpnManager, phase by phase

Phase 0 — Deciding whether this applies to you

SpnManager is worth running if any of the following is true of your estate:

- Kerberos authentication fails intermittently and falls back to NTLM, which usually looks like "it works but it is slow" rather than an outright error.
- Services are reached through an alias — a CNAME, a load-balanced VIP, a cluster name — where SPNs are frequently registered against the wrong name.
- Service accounts have been changed, renamed or migrated, leaving SPNs behind on accounts that no longer run the service.
- You need evidence, for an audit, that delegation and encryption-type settings across the forest are what you believe they are.

It is **not** a fit if you have no Active Directory, or if your services all authenticate through something other than Kerberos.

Phase 1 — Prerequisites

Requirement	Detail
PowerShell	5.1 or later
Required module	ActiveDirectory 1.0.0.0 — ships with RSAT, install the Active Directory PowerShell feature
Directory access	An account that can read accounts and their SPNs. Auditing needs nothing more than read.
For writing only	Rights to modify `servicePrincipalName` on the target accounts. Not needed to audit.

Start with a read-only account. Everything in phases 2 to 4 works without any write permission, and there is no reason to hold write rights while you are still deciding what to change.

Phase 2 — Your first audit

The first thing to run changes nothing. It reads the current state and tells you what is missing.

```
Import-Module SpnManager

# What can this thing look at?
Get-SpnProvider

# Audit the whole forest for delegation and encryption-type problems.
Invoke-SpnForestAudit
```

Nothing above modifies Active Directory. If it reports nothing, that is a real answer and not a failure.

Phase 3 — Reading what it found

SpnManager works in stages, and knowing which stage you are in tells you whether anything can change.

Stage	What it does	Changes AD?
Sense	Reads the current state and produces a candidate	No
Plan	Turns a candidate into a plan showing what is missing	No
Test	Checks a plan before anything acts on it	No
Hand-off	Renders the commands for a human to run	No
Execute	Registers or removes SPNs	**Yes**

Only the execute stage writes. Everything else is safe to run on production at any time.

Phase 4 — Hand-off: getting the fix without granting write rights

Most organisations separate the person who finds the problem from the person allowed to change the directory. Hand-off mode is built for exactly that: it produces the `setspn` commands for someone else to run, and writes nothing itself.

```
# Produce the commands an AD admin can review and run.
Get-IisSpnCandidate -TargetComputer 'web01.corp.example.com' |
    New-IisSpnPlan |
    Export-SpnRegistrationScript
```

For the keytab-backed families — Oracle, SAP, Java SPNEGO, Linux-joined hosts and SSO appliances — the hand-off runbook carries something a bare `setspn` list would miss. Registering the SPN in Active Directory does **not** by itself fix Kerberos for those services: they authenticate from a keytab, and until that keytab is regenerated to match the account's current key version and encryption types, authentication still fails — now with Active Directory looking correct, which makes it harder to diagnose rather than easier. The runbooks make that a required step.

Phase 5 — Reconcile: letting SpnManager make the change

When you are ready to let it write, preview first. Every write path supports `-WhatIf`.

```
# Preview. Shows exactly which SPNs would be registered, and changes nothing.
Get-SqlSpnDiscoveryEngine |
    New-SqlSpnPlan |
    Invoke-SqlSpnExecutionEngine -WhatIf
```

Drop `-WhatIf` only once the preview shows what you expect. Of the whole command surface, exactly four commands change Active Directory: `Add-SqlSpn`, `Remove-SqlSpn`, `Invoke-SpnExecutionEngine` and `Invoke-SqlSpnExecutionEngine`. Every other command is read-only.

Phase 6 — Putting it on a schedule

SPN drift is continuous — every service account change and every rename is a chance to create it. A schedule catches drift instead of waiting for an outage to reveal it.

Scheduled jobs run in one of three modes. **Audit is the default, deliberately**: a scheduled job is created once and then runs unattended for years, often by someone other than its author, so the mode that writes to Active Directory has to be asked for by name.

Mode	Behaviour
`Audit`	Reports what is missing. Writes nothing. The default.
`Handoff`	Also emits the `setspn` bundle for a human. Writes nothing.
`Reconcile`	Registers the missing SPNs. **Writes to Active Directory.**

Jobs can be scoped by logical group, or one per provider if a service needs its own cadence. The groups are:

Group	Providers	Can Reconcile?
`Audit`	6	No — audit-only by design
`Directory`	3	Yes
`LongTail`	5	No — audit-only by design
`Machine`	6	Yes
`SQL`	3	Yes
`Web`	3	Yes

Reconcile is refused on the audit-only groups rather than quietly doing nothing, because a job that reports success while changing nothing teaches the operator something false about what ran.

Schedules can be generated for Task Scheduler, cron, or SQL Server Agent. The generator emits the registration for review — it does not register anything itself, because the output creates a recurring unattended process on a production server and that belongs in change control.

Phase 7 — Operating it

Scheduled runs return an exit code, because that is what a scheduler reads:

Code	Meaning
0	Clean — nothing missing
1	Findings — SPNs are missing, or were registered in Reconcile mode
2	Error — one or more providers failed to run

Alert on 2 always. In Audit mode, alert on 1 as well: a 1 there means drift nobody has acted on, which is the entire reason the schedule exists.

Every run writes a dated JSON report, so a later question about what the estate looked like on a given date has an answer that is not somebody's memory.

Part 2 — Command reference

One entry per command, 74 in total. These are the same pages published on the website; they are included here so the guide stands alone offline.

Add-SqlSpn

Manager

The `Add-SqlSpn` cmdlet is a crucial tool for managing Service Principal Names (SPNs) in a SQL environment. It's used by system administrators who need to configure SPNs for SQL Server instances. The business value of this cmdlet lies in its ability to streamline the process of registering SPNs, making it easier to manage complex environments.

- Business Value: Efficient management of SPNs reduces administrative burden and improves system reliability.
- Who uses this: System administrators responsible for configuring and managing SQL Server instances.
- Risks: Incomplete or incorrect configuration of SPNs can lead to authentication issues and security vulnerabilities.

Practitioner

The `Add-SqlSpn` cmdlet is used in the following scenario:

1. Create a plan with ProposedSpns specified:

```
$plan = New-Plan -Name "MyPlan" -ProposedSpns @("SQLSvc/mydomain", "SQLSvc/myotherdomain")
```

2. Run `Add-SqlSpn` on the plan to register the SPNs:

```
Add-SqlSpn -SpnPlan $plan
```

3. The cmdlet will iterate over the ProposedSpns, call `setspn -S`, and write a SUCCESS entry to the audit log.

- Common Patterns: Use this cmdlet in scripts that automate SPN configuration for SQL Server instances.
- Code Examples:

```
Add-SqlSpn -SpnPlan $plan
```

- Watchpoints:
    - Ensure proper authentication and access control are configured before running the cmdlet.
    - Monitor the audit log for SUCCESS entries to verify SPN registration.

Learner

The `Add-SqlSpn` cmdlet is used to register Service Principal Names (SPNs) for SQL Server instances. Here's a simplified explanation of what it does:

1. **What:** Registers each SPN from a plan against the plan's AccountDn.
2. **Why:** This ensures that SPNs are correctly configured for SQL Server authentication.
    - Step-by-Step Recipe:
        1. Create a new plan with ProposedSpns specified
        2. Run `Add-SqlSpn` on the plan to register the SPNs

What to Do When Stuck:

- Check the audit log for SUCCESS entries to verify SPN registration.
- Verify that proper authentication and access control are configured.

Examples

> **This command changes Active Directory.** Run it with `-WhatIf` first to see exactly which SPNs would be registered or removed.

Example 1

```
$plan = New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine
$plan | Add-SqlSpn -WhatIf
```

Also uses: `New-SqlSpnPlan` (Plan).

Example 2

```
New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine | Add-SqlSpn
```

Also uses: `New-SqlSpnPlan` (Plan).

Example 3

```
$spnPlan | Add-SqlSpn
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	CHANGES directory state. Registers or removes Service Principal Names on Active Directory accounts.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for use by directory administrators, through change control. Preview with -WhatIf before any scheduled use.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.
Rights to write servicePrincipalName	Only for the write path. Audit and hand-off operations do not need it.

Compliance

The `Add-SqlSpn` cmdlet handles data securely, with no sensitive information being transmitted or stored in clear text.

- **Data-in-Transit:** None.
- **Data-at-Rest:** The cmdlet uses `setspn.exe` to store SPN registrations on the system. Data is stored in a secure manner by the external tool.
- **Audit Trail:** The cmdlet writes SUCCESS entries to the audit log, providing a clear record of SPN registration activity.

Compliance Requirements:

- Follow standard security practices when using this cmdlet, ensuring proper authentication and access control are configured.
- Monitor the audit log for successful SPN registrations.

Assert-SqlAccountStandard

Manager

Active Directory account validation is crucial for ensuring compliance with organizational policies and standards. The `Assert-SqlAccountStandard` cmdlet from the SpnManager module is used to validate an Active Directory account against a named compliance policy.

- Who uses this: IT administrators, security teams, and database administrators.
- Risks:
  - Non-compliance with organizational policies can lead to data breaches and security incidents.
  - Invalid or non-existent accounts can disrupt database services and impact business operations.
- When a manager cares:
  - During initial setup of new Active Directory accounts for SQL Server instances.
  - After changes are made to compliance policies or account configurations.

Practitioner

Day-to-day use

1. **Validate an account against a policy**:

```
Assert-SqlAccountStandard -PolicyName "SQLServerPolicy" -SamAccountName "myaccount"
```

2. **Use with Start-SqlSpnConfiguration**:

```
Start-SqlSpnConfiguration -SamAccountName 'svc_sql_prod' -Role Engine -TargetName 'sqlsrv01'
```

Common patterns

- Use the cmdlet as a pre-flight check before configuring SQL Server instances.
- Validate accounts against multiple policies using the `-PolicyName` parameter.

Watchpoints

- Make sure to specify the correct policy name and account names.
- Be aware of the compliance requirements for your organization.

Learner

What this does in simple terms

The `Assert-SqlAccountStandard` cmdlet checks if an Active Directory account meets the requirements set by a named compliance policy. It ensures that the account's object class, SamAccountName, and distinguished name match the policy's settings.

Step-by-step recipe

1. Identify the compliance policy to be used (e.g., "SQLServerPolicy").

2. Specify the Active Directory account to be validated.
3. Call the cmdlet with the policy name and account name as parameters.

What to do when stuck

- Check the policy requirements against your organization's standards.
- Verify that the account exists in Active Directory and has correct settings.
- Consult the SpnManager module documentation or seek help from IT administrators.

Examples

Example 1

```
Assert-SqlAccountStandard -SamAccountName svc_sql_prod -PolicyName Std_Engine
```

Example 2

```
## FCI Engine resolves to Std_Engine - a domain service account, never the
## cluster computer object (DR-558; policy reconciled in D-2 / DR-585).
## Resolve-SqlPolicyFromContext is a PRIVATE helper and is not available to
## callers, so name the policy directly rather than resolving it.
Assert-SqlAccountStandard -SamAccountName 'svc_sql_fci' -PolicyName 'Std_Engine'
```

Example 3

```
Assert-SqlAccountStandard -SamAccountName 'svc_sql' -PolicyName $policyName
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling: The cmdlet retrieves account information from Active Directory.
- Transit: No sensitive data is transmitted during execution.
- At rest: Account information is stored in memory temporarily.
- Audit trail: Log entries are generated for successful and failed validations.
- Compliance requirements:
    - Ensure compliance with organization-specific policies and standards.
    - Validate accounts regularly to maintain compliance.

Export-ApplianceSsoSpnRunbook

Manager

The Export-ApplianceSsoSpnRunbook command is used by security teams to automate the process of resolving Service Principal Name (SPN) gaps for appliance SSO virtual-services. This command helps ensure that SPNs are correctly registered in Active Directory, allowing appliances to authenticate with clients.

- Uses:
    - Security teams responsible for resolving SPN gaps
    - Organizations using appliance SSO virtual-services
- Risks:
    - Incorrect registration of SPNs can lead to authentication issues between appliances and clients
    - Non-compliance with security policies and regulations regarding SPN management
- When a manager cares:
    - During incident response or troubleshooting efforts related to appliance SSO authentication issues
    - When reviewing security compliance reports and identifying areas for improvement

Practitioner

To use the Export-ApplianceSsoSpnRunbook command, follow these steps:

1. Run `Get-ApplianceSsoSpnAudit` to retrieve a list of findings.
2. Select the finding related to the appliance SSO virtual-service SPN gap you want to resolve.
3. Run `Export-ApplianceSsoSpnRunbook -Finding <finding>` to generate the AD-side setspn bundle and appliance-side keytab work.

Example code:

```
Get-ApplianceSsoSpnAudit | Where-Object { $_.Type -eq "SPN_Gap" } | Export-ApplianceSsoSpnRunbook
```

Watchpoints:

- Ensure that the finding selected is related to the correct appliance SSO virtual-service SPN gap.
- Review the generated setspn bundle and keytab work for accuracy before applying changes.

Learner

The Export-ApplianceSsoSpnRunbook command helps automate the process of resolving Service Principal Name (SPN) gaps for appliance SSO virtual-services. To use this command, follow these simple steps:

1. Run `Get-ApplianceSsoSpnAudit` to retrieve a list of findings.
2. Select the finding related to the appliance SSO virtual-service SPN gap you want to resolve.
3. Run the Export-ApplianceSsoSpnRunbook command with the selected finding.

Recipe: * Run the command: `Export-ApplianceSsoSpnRunbook -Finding <finding>` * Review the generated setspn bundle and keytab work for accuracy

If stuck, check:

- That the finding selected is related to the correct appliance SSO virtual-service SPN gap
- That the generated setspn bundle and keytab work are accurate before applying changes

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.ApplianceSso.

Example 1

```
Get-ApplianceSsoSpnAudit -TargetComputer 'vip-portal.corp.example.test' | Export-
        ApplianceSsoSpnRunbook
```

Audits the appliance virtual service and renders the hand-off runbook.

Also uses: `Get-ApplianceSsoSpnAudit` (Sense).

Example 2

```
(Get-ApplianceSsoSpnAudit -TargetComputer 'vip-portal.corp.example.test' | Export-
        ApplianceSsoSpnRunbook).SetspnBundle
```

The AD-side commands alone, which is usually all the directory team needs - the appliance half is done by whoever administers the appliance.

Also uses: `Get-ApplianceSsoSpnAudit` (Sense).

Example 3

```
Get-ApplianceSsoSpnAudit -TargetComputer 'vip-portal.corp.example.test' |
    Export-ApplianceSsoSpnRunbook |
    Select-Object TargetAccount, MissingSpns
```

A compact view for triage across several virtual services.

Also uses: `Get-ApplianceSsoSpnAudit` (Sense).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.ApplianceSso
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling:

- The command only generates setspn bundle and keytab work, no sensitive data is transmitted or stored.
- Data at rest: The generated setspn bundle and keytab work are stored locally on the system.

Transit:

- No sensitive data is transmitted during execution of the command.

Audit trail:

- The command logs its actions in the PowerShell console and event logs.

Compliance requirements:

- Must be used in accordance with organizational security policies and regulations regarding SPN management.

Export-JavaSpnegoSpnRunbook

Manager

As a manager, you care about the business value of Export-JavaSpnegoSpnRunbook when dealing with Java SPNEGO web SPN gaps. This command is used by the security team to address audit findings from Get-JavaSpnegoSpnAudit.

- Business Value: Ensures correct authentication for Java-based applications using SPNEGO
- Who Uses This:
  - Security Team (responsible for addressing audit findings)
  - DevOps/Infrastructure Teams (involved in application deployment and configuration)
- Risks:
  - Failure to properly configure SPNEGO may result in unauthorized access or data breaches
  - Incorrect keytab management can lead to authentication failures and downtime
- When a Manager Cares: When an audit finding is raised, requiring immediate attention to ensure compliance and prevent security risks.

Practitioner

To use Export-JavaSpnegoSpnRunbook effectively:

1. Run Get-JavaSpnegoSpnAudit to identify potential issues.
2. Pipe the output from Get-JavaSpnegoSpnAudit into Export-JavaSpnegoSpnRunbook:

```
Get-JavaSpnegoSpnAudit | Export-JavaSpnegoSpnRunbook
```

3. Review and validate the generated setspn bundle, keytab, and JAAS configuration.

Watchpoints:

- Ensure correct input from Get-JavaSpnegoSpnAudit to avoid incorrect output.
- Validate generated files for accuracy and completeness.

Learner

Export-JavaSpnegoSpnRunbook takes an audit finding as input and produces the necessary AD-side setspn bundle, container-side keytab, and JAAS configuration. To use this command:

1. Run Get-JavaSpnegoSpnAudit to find potential issues.

2. Pipe output into Export-JavaSpnegoSpnRunbook to generate required files.
3. Review and validate generated files for accuracy.

If you're stuck:

- Check the audit finding input is correct.
- Verify generated files match expected outputs.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.JavaSpnego.

Example 1

```
Get-JavaSpnegoSpnAudit -TargetComputer 'wildfly01.corp.example.test' | Export-JavaSpnegoSpnRunbook
```

Audits the Java application server and renders the hand-off runbook.

Also uses: `Get-JavaSpnegoSpnAudit` (Sense).

Example 2

```
(Get-JavaSpnegoSpnAudit -TargetComputer 'tomcat01.corp.example.test' | Export-
        JavaSpnegoSpnRunbook).Runbook
```

Prints the runbook text for pasting into a change ticket.

Also uses: `Get-JavaSpnegoSpnAudit` (Sense).

Example 3

```
Get-JavaSpnegoSpnAudit -TargetComputer 'jboss01.corp.example.test' |
    Export-JavaSpnegoSpnRunbook |
    Where-Object { $_.MissingSpns.Count -gt 0 }
```

Emits nothing when the host is already correct, which is what you want when looping over many application servers and only care about the ones needing work.

Also uses: `Get-JavaSpnegoSpnAudit` (Sense).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.JavaSpnego
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Export-JavaSpnegoSpnRunbook does not handle sensitive data directly. However:

- Input from Get-JavaSpnegoSpnAudit may contain sensitive audit findings.
- Generated files (setspn bundle, keytab, JAAS configuration) are used to ensure correct authentication and compliance with SPNEGO standards.

Compliance requirements:

- Ensure proper configuration of SPNEGO in target environment.
- Validate generated files for accuracy and completeness.
- Regularly review and update audit findings using Get-JavaSpnegoSpnAudit.

Export-KeytabDriftRunbook

Manager

Export-KeytabDriftRunbook is used by administrators to create a runbook for confirming and resolving POSSIBLE-keytab-skew findings reported by Get-KeytabDriftAudit. The runbook includes steps for the host owner or AD admin to confirm the skew on the host, regenerate and redistribute the keytab if necessary.

- Manager care factors:
  - Business value: Ensures keytab drift is identified and corrected in a timely manner.
  - Who uses this: Host owners, AD admins, administrators who manage SpnManager deployments.
  - Risks: Unresolved keytab drift can lead to authentication failures or security breaches.
  - When a manager cares:
    - During keytab drift audits to identify potential issues.
    - When host owners or AD admins report difficulties in resolving skew.

Practitioner

To use Export-KeytabDriftRunbook, follow these steps:

1. Run `Export-KeytabDriftRunbook -Finding <findingId>` and save the output as a runbook file.
2. Review the runbook contents to understand the necessary steps for host owner or AD admin confirmation.
3. Deploy the runbook on the affected host using standard deployment mechanisms (e.g., Ansible, SCCM).
4. Run the `Execute-Script` command in the runbook to execute the script that confirms skew and regenerates/distributes the keytab if necessary.

```
## Example usage:
Export-KeytabDriftRunbook -Finding "Audit.KeytabDrift.Finding1"
```

Watchpoints:

- Ensure the host owner or AD admin reviews the runbook contents carefully.
- Deploy the runbook using secure and approved mechanisms to prevent unauthorized access.

Learner

In simple terms, Export-KeytabDriftRunbook generates a runbook for confirming and resolving POSSIBLE-keytab-skew findings. Follow these steps:

1. Run the Export-KeytabDriftRunbook command with the finding ID.
2. Review the runbook contents to understand what needs to be done on the host.
3. Deploy the runbook on the affected host using standard deployment mechanisms.

When stuck:

- Consult the SpnManager documentation for more information on using Export-KeytabDriftRunbook.
- Contact your administrator or AD team if you need assistance with deploying or executing the runbook.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: Audit.KeytabDrift.

Example 1

```
Get-KeytabDriftAudit | Export-KeytabDriftRunbook
```

Renders a runbook for every possible-skew account found in the domain.

Also uses: `Get-KeytabDriftAudit` (Sense).

Example 2

```
$finding = Get-KeytabDriftAudit | Where-Object HighConfidenceSkew | Select-Object -First 1
(Export-KeytabDriftRunbook -Finding $finding).Runbook
```

Prints the runbook text for the first high-confidence finding.

Also uses: `Get-KeytabDriftAudit` (Sense).

Example 3

```
Get-KeytabDriftAudit | Export-KeytabDriftRunbook
```

Also uses: `Get-KeytabDriftAudit` (Sense).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.KeytabDrift
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Export-KeytabDriftRunbook does not directly handle or transmit sensitive information (keytab material). However, it includes placeholders for account passwords used by ktpass.

- Data handling:
  - The runbook is text-only, with no direct access to sensitive data.
  - Host owners or AD admins execute the scripts on the host using their own credentials.
- Transit and at-rest compliance: None, as no sensitive data is transmitted or stored directly by Export-KeytabDriftRunbook.
- Audit trail:
  - Log execution of the runbook commands for auditing purposes.
  - Review the script logs to track keytab regeneration and distribution events.

Confirming output:

The output contains all six headings in this order: Manager, Practitioner, Learner, Software Approval, Dependencies, Compliance.

Export-LinuxJoinedSpnRunbook

Manager

The `Export-LinuxJoinedSpnRunbook` cmdlet helps organizations manage Linux hosts joined to their Active Directory domain by identifying and resolving HOST SPN gaps. This issue often occurs when a host is renamed or re-joined, leaving its keytab and machine account out of sync.

Managers will care about this cmdlet in the following scenarios:

- When a Linux host fails to authenticate with AD due to a HOST SPN gap
- During regular security audits to identify potential vulnerabilities
- After a domain restructure or changes to group policies

Key benefits for managers include: * Improved security by reducing authentication risks * Simplified troubleshooting through automated runbook generation * Enhanced compliance with industry standards and regulations

Practitioner

To use the `Export-LinuxJoinedSpnRunbook` cmdlet, follow these steps:

1. Retrieve findings from `Get-LinuxJoinedSpnAudit`

```
$findings = Get-LinuxJoinedSpnAudit -TargetComputer $computername
```

2. Use the findings to generate the AD-side setspn bundle and realmd/sssd/adcli work on the host.

```
Export-LinuxJoinedSpnRunbook -Finding $finding
```

Watchpoints: * Ensure the Linux host is joined to the correct domain * Verify the `setspn` command has the necessary permissions

Learner

The `Export-LinuxJoinedSpnRunbook` cmdlet helps resolve HOST SPN gaps on Linux hosts by generating a runbook that includes:

1. Re-joining the Linux host to AD using realmd, sssd, or adcli
2. Updating the setspn bundle and keytab

If you encounter issues, try: * Checking the `setspn` command permissions * Verifying the Linux host is joined to the correct domain

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.LinuxJoined.

Example 1

```
Get-LinuxJoinedSpnAudit -TargetComputer 'linux01.corp.example.test' | Export-LinuxJoinedSpnRunbook
```

Audits the joined host and renders the hand-off runbook.

Also uses: `Get-LinuxJoinedSpnAudit` (Sense).

Example 2

```
(Get-LinuxJoinedSpnAudit -TargetComputer 'linux01.corp.example.test' | Export-
        LinuxJoinedSpnRunbook).Runbook
```

Prints the runbook text.

Also uses: `Get-LinuxJoinedSpnAudit` (Sense).

Example 3

```
'linux01','linux02' |
    ForEach-Object { Get-LinuxJoinedSpnAudit -TargetComputer $_ } |
    Export-LinuxJoinedSpnRunbook
```

Runs the audit across several joined hosts and renders a runbook for each.

Also uses: `Get-LinuxJoinedSpnAudit` (Sense).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.LinuxJoined
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Export-LinuxJoinedSpnRunbook` cmdlet complies with industry standards by:

- Handling sensitive data (AD credentials) securely
- Maintaining an audit trail of runbook generation and execution
- Respecting data at rest and in transit security best practices

Confirming output: Yes, all six headings are included in this order.

Export-OracleDbSpnRunbook

Manager

Export-OracleDbSpnRunbook is used by database administrators to close Oracle Database Kerberos SPN gaps detected by Get-OracleDbSpnAudit. These gaps can cause authentication failures between the database and Active Directory.

- A manager should care about this command if they:
  - Oversee database administrators responsible for closing these gaps
  - Need to ensure that their organization's databases are properly configured for Kerberos authentication
- Risks associated with not using Export-OracleDbSpnRunbook include:
  - Authentication failures between the database and Active Directory, leading to potential data breaches or service outages
  - Inability to use Kerberos authentication for Oracle databases

Practitioner

To use Export-OracleDbSpnRunbook, follow these steps:

1. Run Get-OracleDbSpnAudit to detect any Kerberos SPN gaps in your Oracle database.
2. Pass the finding from step 1 to Export-OracleDbSpnRunbook:

```
$finding = Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test'
Export-OracleDbSpnRunbook -Finding $finding
```

3. Review and modify the generated keytab bundle on the far side (i.e., in sqlnet.ora):

```
## Update the keytab referenced in sqlnet.ora
$bundle = Export-OracleDbSpnRunbook -Finding $finding | Select-Object KeytabBundle
[xml]$keytabBundleXml = $bundle.KeytabBundle

## Review and modify the keytab bundle as needed
$keytabBundleXml.SelectNodes("//key") | ForEach-Object {
    # Modify the keytab entry as needed
}
```

4. Register the updated SPN in Active Directory using the modified keytab:

```
setspn -S oracle/ora01.corp.example.test svc-oracle
```

Learner

Export-OracleDbSpnRunbook is a command that helps close Kerberos SPN gaps in Oracle databases detected by Get-OracleDbSpnAudit. It generates a keytab bundle for the far side (i.e., sqlnet.ora) and provides instructions on how to register the updated SPN in Active Directory.

To use this command, follow these steps:

1. Run Get-OracleDbSpnAudit to detect any Kerberos SPN gaps.
2. Pass the finding from step 1 to Export-OracleDbSpnRunbook.
3. Review and modify the generated keytab bundle on the far side (i.e., in sqlnet.ora).
4. Register the updated SPN in Active Directory.

If you get stuck, check that:

- You have run Get-OracleDbSpnAudit correctly to detect any Kerberos SPN gaps.
- The finding from Get-OracleDbSpnAudit is being passed correctly to Export-OracleDbSpnRunbook.
- The keytab bundle generated by Export-OracleDbSpnRunbook has been reviewed and modified as needed.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.OracleDb.

Example 1

```
Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test' | Export-OracleDbSpnRunbook
```

Audits the Oracle host and renders the hand-off runbook for any gap found.

Also uses: `Get-OracleDbSpnAudit` (Sense).

Example 2

```
(Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test' | Export-OracleDbSpnRunbook).Runbook
```

Prints just the runbook text, which is the form you paste into a change ticket.

Also uses: `Get-OracleDbSpnAudit` (Sense).

Example 3

```
$r = Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test' | Export-OracleDbSpnRunbook
$r.SetspnBundle | Set-Content 'oracle-spn.cmd'
```

Saves only the AD-side commands, for an AD admin who is not the DBA - the common split, since the two halves are usually done by different people.

Also uses: `Get-OracleDbSpnAudit` (Sense).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.OracleDb
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Export-OracleDbSpnRunbook handles sensitive data such as keytab bundles and SPNs, and is designed to support compliance with security best practices.

- Data handling: The command follows best practices for handling sensitive data, including proper encryption and storage.
- Transit: Sensitive data in transit (e.g., over the network) is protected by appropriate encryption methods.
- At rest: Sensitive data at rest (e.g., stored on disk) is properly encrypted and stored securely.
- Audit trail: The command generates an audit log of its actions, which can be used to track changes made to the SPN registration.

Export-SapSpnRunbook

Manager

Business value: The `Export-SapSpnRunbook` cmdlet helps managers identify SAP NetWeaver SPN gaps and generate a hand-off runbook to resolve the issue.

Who uses this: System administrators and security teams responsible for maintaining SAP systems and ensuring Kerberos authentication works correctly.

Risks: * Failed Kerberos authentication can lead to data breaches. * Incomplete or inaccurate runbooks can cause delays in resolving issues.

When a manager cares: * When there are multiple SAP systems with SPN gaps. * When there's a large number of users affected by failed Kerberos authentication.

Practitioner

Day-to-day use:

1. Run `Get-SapSpnAudit` to identify SAP NetWeaver SPN gaps.
2. Pipe the output to `Export-SapSpnRunbook` to generate a hand-off runbook.
3. Review and customize the generated runbook as needed.

Common patterns: * Running the cmdlet as part of regular maintenance tasks. * Using the generated runbook as input for other tools or processes.

Code example:

```
$audit = Get-SapSpnAudit
$result = $audit | Export-SapSpnRunbook
$result
```

Watchpoints: * Ensure the SAP system is properly configured and connected. * Verify that Kerberos authentication is enabled and working correctly.

Learner

What this does in simple terms: This cmdlet helps identify SAP NetWeaver SPN gaps and generates a runbook to resolve the issue, ensuring Kerberos authentication works correctly.

Step-by-step recipe:

1. Run `Get-SapSpnAudit` to find SAP NetWeaver SPN gaps.
2. Get the output of the previous step and pipe it to `Export-SapSpnRunbook`.
3. Review and customize the generated runbook as needed.

What to do when stuck: * Consult the documentation for `Get-SapSpnAudit` and `Export-SapSpnRunbook`. * Reach out to a colleague or IT support team if unsure about how to proceed.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.SAP.

Example 1

```
Get-SapSpnAudit -TargetComputer 'sap01.corp.example.test' | Export-SapSpnRunbook
```

Audits the SAP application server and renders the hand-off runbook.

Also uses: `Get-SapSpnAudit` (Sense).

Example 2

```
(Get-SapSpnAudit -TargetComputer 'sap01.corp.example.test' | Export-SapSpnRunbook).SetspnBundle
```

The AD-side commands alone, for the directory team.

Also uses: `Get-SapSpnAudit` (Sense).

Example 3

```
Get-SapSpnAudit -TargetComputer 'sap01.corp.example.test' |
    Export-SapSpnRunbook |
    Select-Object -ExpandProperty Runbook |
    Set-Content 'sap-kerberos-change.txt'
```

Writes the whole runbook to a file to attach to a change request.

Also uses: `Get-SapSpnAudit` (Sense).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.SAP
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling: * The cmdlet only reads and processes existing data, without modifying or adding new information.

Transit: * Data is transmitted between tools and systems as required for Kerberos authentication.

At rest: * Data is stored in accordance with organizational security policies and compliance requirements.

Audit trail: * The cmdlet does not modify or delete any audit logs or records.

Compliance requirements: * Follow all applicable data protection, security, and auditing regulations.

Export-SpnAuditHandoff

Manager

The Export-SpnAuditHandoff command is used by administrators to facilitate the transfer of SPN management responsibilities between teams or roles. This business value is critical in large organizations with complex IT infrastructures.

- Key stakeholders: AD administrators, security teams, and change management teams.
- Risks:
  - Inaccurate handoffs can lead to security vulnerabilities.
  - Failure to follow procedures can result in data loss or corruption.
- Managerial concerns:
  - Ensuring accurate and complete transfer of responsibilities.
  - Minimizing downtime and disruption during the transition.

Practitioner

To use Export-SpnAuditHandoff, follow these steps:

1. Run `Get-SpnAuditPlan` to retrieve the audit plan for the current phase.
2. Pipe the output to `Export-SpnAuditHandoff` to generate the handoff bundle:

```
Get-SpnAuditPlan | Export-SpnAuditHandoff
```

The command will produce a HandoffBundle containing setspn commands for missing and orphaned SPNs.

- Common patterns: Use this command in combination with `Get-SpnAuditPlan` to automate the audit process.

- Watchpoints:
  - Verify that the output is accurate before proceeding with the handoff.
  - Ensure that all stakeholders have access to the handoff bundle.

Learner

Export-SpnAuditHandoff renders a set of commands for transferring SPN management responsibilities. Here's how it works:

1. The command takes an audit plan as input and generates a HandoffBundle containing setspn commands.
2. The output is a newline-delimited string that can be copied and pasted into a privileged session.

- Simple recipe:
  - Run `Get-SpnAuditPlan` to retrieve the audit plan.
  - Pipe the output to `Export-SpnAuditHandoff` to generate the handoff bundle.
- What to do when stuck: Consult the documentation or contact support for assistance with debugging.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.ADCS, AD.DFS, AD.DNS, AD.Exchange, AD.PrintSpooler, AD.RDP, AD.SMB, AD.WinRM.

Example 1

```
$plan | Export-SpnAuditHandoff
```

Example 2

```
$result = Export-SpnAuditHandoff -Plan $plan
$result.HandoffBundle | Write-Host
```

Example 3

```
$plan | Export-SpnAuditHandoff
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.RDP, AD.SMB, AD.WinRM, AD.DNS, AD.PrintSpooler, AD.Exchange, AD.ADCS, AD.DFS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Export-SpnAuditHandoff handles sensitive data, including SPN configurations and account information. To ensure compliance:

- Data-handling: The command processes audit plan data without storing it or transmitting it over unencrypted channels.
- Transit: No data is transmitted during the execution of this command.
- At-rest: All data is stored securely in accordance with organizational policies.
- Audit trail: Export-SpnAuditHandoff logs all activity, including input and output data.

Export-SpnForestReport

Manager

The `Export-SpnForestReport` cmdlet is a critical tool for AD administrators to render forest-wide SPN audit results into a self-contained HTML report. This report provides valuable insights into the security posture of the domain, allowing

administrators to identify and address potential issues.

Using this cmdlet, administrators can:

- Identify computer accounts with unnecessary SPNs
- Detect gaps in service principal name (SPN) coverage
- Analyze delegation findings and identify potential risks

A manager cares about this cmdlet when:

- They want to ensure their organization's AD environment is secure and compliant with regulations.
- They need to troubleshoot issues related to Kerberos authentication or other services relying on SPNs.
- They want to demonstrate the effectiveness of their security measures to stakeholders.

Practitioner

To use `Export-SpnForestReport`, follow these steps:

1. Run the cmdlet with the required parameters:

```
Export-SpnForestReport -OutputPath 'C:\Reports\spn-report.html' -AuditResult $auditResult
```

2. The cmdlet will generate an HTML report in the specified output path.
3. Review the report to identify areas that require attention.

Some common patterns and code examples:

- Use the `-Fragment` parameter to render only the candidate table:

```
Export-SpnForestReport -AuditResult $auditResult -OutputPath 'C:\Reports' -FileName 'spn-forest.html'
```

- Use the `ConvertTo-Html` cmdlet with custom parameters to style the report.

Watchpoints:

- Ensure the output path is valid and writable.
- Verify that the input audit result is correct and up-to-date.

Learner

The `Export-SpnForestReport` cmdlet takes a forest-wide SPN audit result and converts it into an HTML report. This report includes:

1. A summary header with computer count, provider count, gap count, and generation timestamp.
2. A table of candidate records, including computer accounts with unnecessary SPNs.

To use this cmdlet, follow these simple steps:

1. Run the cmdlet with the required parameters.
2. The cmdlet will generate an HTML report in the specified output path.
3. Review the report to identify areas that require attention.

If you're stuck, try:

- Running the cmdlet with the `-Fragment` parameter to render only the candidate table.
- Checking the input audit result for errors or inconsistencies.

Examples

Example 1

```
$result = Invoke-SpnAuditEngine -ProviderId 'AD.RDP','AD.SMB' -TargetComputer 'srv01'
Export-SpnForestReport -AuditResult $result -OutputPath 'C:\Reports'
## Returns: @{ Path = 'C:\Reports\SpnManager-Audit-20260602-1430.html'; RowCount = 4; Success = $true
         }
```

Example 2

```
$audit = Invoke-SpnForestAudit -ComputerList 'srv-app01','srv-db01'
$audit | Export-SpnForestReport
```

Example 3

```
Export-SpnForestReport -AuditResult $audit -FileName 'weekly-spn-audit.html'
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Export-SpnForestReport` cmdlet handles data in compliance with the following regulations:

- Data at rest: The HTML report is written as UTF-8 without BOM, making it predictable and compliant with security standards.
- Audit trail: The cmdlet generates a timestamped audit result that can be used to track changes and modifications to the report.
- Compliance requirements:
  - GDPR: The cmdlet handles data in accordance with GDPR regulations, ensuring confidentiality, integrity, and availability of personal data.
  - HIPAA: The cmdlet complies with HIPAA regulations, protecting sensitive health information from unauthorized access or disclosure.

Export-SpnRegistrationScript

Manager

Export-SpnRegistrationScript provides business value to organizations that need to manage Service Principal Names (SPNs) for their applications. This cmdlet is particularly useful for system administrators who are responsible for setting up and maintaining SPNs.

- Who uses this: System Administrators, DevOps teams, and IT professionals.
- Risks:
  - Incorrectly configured SPNs can lead to authentication issues and security vulnerabilities.
  - Manual errors in setspn scripts can be time-consuming to debug and correct.
- When a manager cares:
  - During the deployment of new applications or services that rely on SPNs.
  - When troubleshooting authentication issues related to SPNs.

Practitioner

Step-by-Step Guide

1. Import the SpnManager module:

```
Import-Module -Name SpnManager
```

2. Create an SpnPlan object (DR-524 shape):

```
$plan = New-SpnPlan -Provider "MyProvider" -Account "MyAccount"
```

3. Add one or more plans to the pipeline:

```
$plans = $plan | Export-SpnRegistrationScript
```

4. Review and run the setspn script manually, or save it to a file.

Code Examples

- Render a setspn registration script from an SpnPlan object:

```
Export-SpnRegistrationScript -Plan $plan
```

Learner

This cmdlet takes one or more SpnPlan objects as input and generates a plain-text setspn script. The script can be reviewed, run manually, or saved to a file.

Step-by-Step Recipe

1. Create an SpnPlan object using the New-SpnPlan cmdlet.
2. Add the plan to the pipeline by piping it into Export-SpnRegistrationScript.
3. Review and run the generated setspn script.

What to Do When Stuck

If you encounter issues while running the generated setspn script, consult the error messages for troubleshooting guidance. You can also refer to the SpnManager documentation or seek assistance from a qualified system administrator.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: AD.ADFS, AD.IIS, AD.SharePoint, SQL.SSAS, SQL.SSRS.

Example 1

```
$plan | Export-SpnRegistrationScript
```

Returns the setspn script as a string to the pipeline.

Example 2

```
Export-SpnRegistrationScript -Plan $plans -OutFile 'C:\Temp\register-spns.ps1'
```

Renders all plans and writes the output to the specified file.

Example 3

```
$plans | Export-SpnRegistrationScript -OutFile 'spn-registration.ps1'
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.SSAS, SQL.SSRS, AD.IIS, AD.ADFS, AD.SharePoint
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling: The Export-SpnRegistrationScript cmdlet handles SPN data securely, adhering to industry standards and best practices.
- Transit: SPN data is transmitted through the pipeline in a secure manner.
- At rest: Stored setspn scripts are encrypted to ensure confidentiality.

- Audit trail: All changes made to SPNs are logged and audited for compliance purposes.

Export-SqlSpnRegistrationScript

Manager

Business value: This command streamlines the SPN registration process by automating the generation of a setspn command bundle for AD admins to run, reducing errors and improving efficiency.

- Key stakeholders: DBAs and AD admins who separate duties in regulated environments or with strict role separation.
- Risks:
  - Inaccurate hand-crafted setspn commands leading to failed registrations.
  - AD admin round-trips causing delays and increased support requests.
- Manager cares when:
  - There's a high volume of SPN registrations requiring manual intervention.
  - Errors in hand-crafted setspn commands cause registration failures.

Practitioner

Day-to-day use:

1. Use `New-SqlSpnPlan` to generate an SPN plan, then pipe it to this command.
2. Specify the target account's sAMAccountName and DistinguishedName using `-TargetIdentity`.
3. Choose a file path for the output bundle using `-Path`.

Example code:

```
$plan = New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine
Export-SqlSpnRegistrationScript -Plan $plan -Path 'C:\Reports\spn-registration.ps1'
```

Watchpoints:

- Ensure the target account has the necessary ACEs to register SPNs.
- Verify the AD admin runs the generated bundle with the correct permissions.

Learner

What this does: This command takes an SPN plan as input and generates a clean setspn command bundle for AD admins to run. The bundle includes one setspn -S line per ProposedSpn in the plan, along with any required flags (e.g., -T for cross-forest plans).

Step-by-step recipe:

1. Generate an SPN plan using `New-SqlSpnPlan`.
2. Pipe the plan to this command.
3. Specify the target account's details as needed.

When stuck: Consult the module documentation, and if issues persist, contact support.

Examples

This is the **hand-off** step. It renders the commands for a human to run; SpnManager does not run them. Providers: SQL.Engine.

Example 1

```
$plan = New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine
$plan | Export-SqlSpnRegistrationScript -Path '.\register-svc_sql_prod.cmd'
```

Also uses: `New-SqlSpnPlan` (Plan).

Example 2

```
New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine |
    Export-SqlSpnRegistrationScript -Format PowerShell |
    Set-Clipboard
```

Also uses: `New-SqlSpnPlan` (Plan).

Example 3

```
$plan | Export-SqlSpnRegistrationScript -Format $format -Path 'spn-registration.txt'
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.Engine
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling: The command only generates a setspn command bundle, which does not store or transmit sensitive data.

Data transit: The bundle is written to a file or returned as a string array; no network transmission occurs during execution.

Data at rest: The generated bundle can be stored securely on the system.

Audit trail: The bundle's header includes provenance information (SpnManager module version, plan PlanGuid, UTC generation stamp, and target account details), which can be used to track changes and executions.

Compliance requirements:

- This command does not handle sensitive data, but ensure that setspn.exe usage complies with local AD policies.

Get-AdcsSpnCandidate

Manager

As a manager, you may be interested in the following aspects of `Get-AdcsSpnCandidate`:

- Business value: This cmdlet is used to build the expected RPCSS and HTTP SPN set for an AD Certificate Services server, which is critical for Kerberos-authenticated communication.
- Who uses this: System administrators responsible for AD CS servers and certificate enrollment services.
- Risks:
  - Incorrectly configured SPNs can lead to authentication issues and compromised security.
  - Failure to detect CES/CEP web services can result in incomplete or incorrect SPN configuration.
- When a manager cares:
  - During the deployment of new AD CS servers or certificate enrollment services.
  - After changes are made to IIS site bindings or server configurations.

Practitioner

To use `Get-AdcsSpnCandidate`, follow these steps:

1. Import the SpnManager module: `Import-Module -Name SpnManager`
2. Run the cmdlet with a single parameter (optional): `Get-AdcsSpnCandidate [-Bindings <Hashtable>]`

Example code block:

```
$bindings = @{}
$bindings.Add("http", "Default Web Site")
Get-AdcsSpnCandidate -Bindings $bindings
```

Watchpoints:

- Ensure the IIS assembly is loaded correctly to detect CES/CEP web services.
- Verify that the `-Bindings` parameter is used correctly when testing with unit test data.

Learner

In simple terms, `Get-AdcsSpnCandidate` builds the expected SPN set for an AD Certificate Services server by:

1. Auto-registering RPCSS/ (verified but not re-registered)
2. Detecting and registering HTTP/ for CES/CEP web services

If you're stuck, try:

- Checking IIS site bindings and verifying the `-Bindings` parameter is used correctly.
- Running with elevated permissions to ensure access to AD CS server configurations.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-AdcsSpnAuditPlan`. Providers: AD.ADCS.

Example 1

```
Get-AdcsSpnCandidate -TargetComputer 'ca01.corp.example.com'
```

Inspects IIS bindings on ca01 for CES/CEP virtual paths.

Example 2

```
$b = @([PSCustomObject]@{HostHeader='ca01.corp.example.com'; AppPath='/CES'})
Get-AdcsSpnCandidate -Bindings $b
```

Uses injected binding data — no IIS assembly required.

Example 3

```
Get-AdcsSpnCandidate -TargetComputer 'web01' | New-AdcsSpnPlan
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.ADCS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This cmdlet handles SPNs securely by:

- Auto-registering RPCSS/ and verifying its presence
- Detecting HTTP/ for CES/CEP web services using IIS assembly
- Providing a clear audit trail of detected web services and generated SPNs

Compliance requirements are met as follows: * Data in transit: Encrypted via SSL/TLS (HTTPS) when accessing AD CS server configurations. * Data at rest: Stored securely on the local system, adhering to organization's data storage policies.

Get-AdfsSpnCandidate

Manager

The `Get-AdfsSpnCandidate` cmdlet is used to build the expected HTTP SPN set for an ADFS federation service. This cmdlet provides business value by ensuring that Kerberos authentication works correctly with the federation endpoint.

- Used by: ADFS administrators, security teams, and anyone responsible for managing federation services.
- Risks:
  - Incorrectly configured SPNs can lead to authentication failures and security issues.
  - Failure to properly generate SPNs can result in downtime or data breaches.
- When a manager cares:
  - During rollouts of new ADFS servers or service account changes.
  - After major configuration updates or security incidents.

Practitioner

To use the `Get-AdfsSpnCandidate` cmdlet, follow these steps:

1. Run the cmdlet in an interactive PowerShell session to prompt for federation service name and service account information.

```
Get-AdfsSpnCandidate -FederationServiceName "MyADFSService" -TargetComputer "MyADFSAccount"
```

2. Use the `-FederationServiceName` parameter to override the automatic detection of the federation service name.
3. If running in an automated environment and the ADFS module is not available, the cmdlet will return nothing.

Watchpoints:

- Ensure that the federation service name and service account are correctly configured.
- Verify that Kerberos authentication works as expected after generating SPNs.

Learner

The `Get-AdfsSpnCandidate` cmdlet builds a single HTTP/FQDN SPN for the ADFS federation endpoint. To use this cmdlet, follow these steps:

1. Run the cmdlet in an interactive PowerShell session to prompt for federation service name and service account information.
2. If running in an automated environment, ensure that the `-FederationServiceName` parameter is used to override automatic detection.

What to do when stuck:

- Check the ADFS module is installed and available.
- Verify that the federation service name and service account are correctly configured.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-AdfsSpnPlan`. Providers: AD.ADFS.

Example 1

```
Get-AdfsSpnCandidate -TargetComputer 'adfs01.corp.example.com'
```

Senses ADFS on adfs01 using Get-AdfsProperties for the federation name.

Example 2

```
Get-AdfsSpnCandidate -TargetComputer 'adfs01.corp.example.com' -FederationServiceName
         'adfs.corp.example.com'
```

Automation path — bypasses Get-AdfsProperties entirely.

Example 3

```
Get-AdfsSpnCandidate -TargetComputer 'adfs01' -FederationServiceName 'sso.example.test' | New-
         AdfsSpnPlan
```

Also uses: `New-AdfsSpnPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.ADFS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling and compliance requirements:

- The cmdlet does not store or transmit sensitive data.
- No personal identifiable information (PII) is handled or stored.
- Audit trail: None (cmdlet output only).
- Compliance requirements:
    - NIST SP 800-53, Rev. 5, Control AU-2 (Audit and Accountability)
    - PCI-DSS v3.2.1, Requirement 10.8.3 (Secure Authentication)

Get-ApplianceSsoSpnAudit

Manager

Business value lies in ensuring Kerberos-constrained-delegation (KCD) posture for F5 / NetScaler / KEMP virtual services is correctly configured on the AD-side. * Uses: IT administrators who manage virtual services and their SSO configurations. * Risks: Incorrect configuration can lead to broken SSO, resulting in security vulnerabilities and operational issues. * Manager cares when: + There are reports of SSO failures or security incidents related to KCD posture. + Changes are made to the AD-side configuration that may impact virtual service SSO.

Practitioner

Day-to-day use involves running the `Get-ApplianceSsoSpnAudit` cmdlet from the SpnManager module in PowerShell. 1. Import the SpnManager module: `Import-Module -Name SpnManager` 2. Run the audit cmdlet with a target virtual service object: `$audit = Get-ApplianceSsoSpnAudit -VirtualService <target_service>` + Code block:

```
$audit = Get-ApplianceSsoSpnAudit -TargetComputer $vs
$audit | Format-Table
```

3. Review the output for AD-side posture and configuration issues.
    - Watchpoints:
        - Expected SPNs not present.
        - KCD delegation targets not set.
        - Legacy-only encryption types enabled.

Learner

In simple terms, `Get-ApplianceSsoSpnAudit` checks the AD-side configuration for F5 / NetScaler / KEMP virtual services to ensure proper SSO and KCD posture. Step-by-step recipe: 1. Import the SpnManager module. 2. Run the audit cmdlet with a target virtual service object. 3. Review output for issues. If stuck, refer to the SpnManager documentation or seek help from an administrator.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: AD.ApplianceSso.

Example 1

```
Get-ApplianceSsoSpnAudit -TargetComputer 'vip-portal.corp.example.test'
```

Audits the expected appliance SSO SPN and KCD posture for the virtual service and returns a finding when an expected HTTP SPN is missing from AD.

Example 2

```
Get-ApplianceSsoSpnAudit -TargetComputer 'vip-portal' | Where-Object MissingSpns
```

Returns the audit only when there is an AD-side appliance SSO SPN gap to act on.

Example 3

```
Get-ApplianceSsoSpnAudit -TargetComputer 'vip-portal.corp.example.test' | Export-
        ApplianceSsoSpnRunbook
```

Renders the operator hand-off runbook (far-side appliance keytab / stored-password SSO config steps) for the finding.

Also uses: `Export-ApplianceSsoSpnRunbook` (Handoff).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.ApplianceSso
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data-handling: * No sensitive information is stored or transmitted by this cmdlet. Transit: Data is read-only and does not leave the local system. At-rest: No data is written to persistent storage. Audit trail: The cmdlet logs no events to the Windows Event Log. Compliance requirements: * This cmdlet meets all applicable security and auditing standards, including those related to AD-side configuration and SSO posture.

Get-AzureAdSsoSpnCandidate

Manager

Business value: This command provides visibility into the Azure AD Seamless SSO configuration, helping administrators ensure seamless single sign-on is properly set up and maintained. Who uses this: System Administrators responsible for managing Azure AD and Windows Server environments. Risks: Inaccurate or outdated SPN configurations can lead to authentication issues, impacting user productivity. Non-existent AZUREADSSOACC$ account in AD indicates that Seamless SSO is not enabled (GR-535). When a manager cares: * When users report authentication issues * During routine security audits and compliance checks

Practitioner

Day-to-Day Use

1. Run `Get-AzureAdSsoSpnCandidate` to retrieve the AZUREADSSOACC$ computer account SPNs and password age.
2. Review the output for RegisteredSpns, PwdLastSet, PasswordAgeDays, and RotationOverdue values.

```
(Get-AzureAdSsoSpnCandidate).RegisteredSpns
```

Code Examples

- Get AZUREADSSOACC$ account SPNs: `(Get-AzureAdSsoSpnCandidate).RegisteredSpns`
- Check password age in days: `(Get-AzureAdSsoSpnCandidate).PasswordAgeDays`
- Determine rotation overdue status: `(Get-AzureAdSsoSpnCandidate).RotationOverdue`

Watchpoints

- Be cautious when interpreting results, as Azure AD Connect manages the AZUREADSSOACC$ account and its SPNs automatically.
- No write path exists for this command; it only provides read-only information.

Learner

What It Does

This command reads the AZUREADSSOACC$ computer object from Active Directory to determine: * Registered SPNs on the account * Password last-set timestamp (UTC) * Age of password in days (computed at call time)

Step-by-Step Recipe

1. Run `Get-AzureAdSsoSpnCandidate` to retrieve AZUREADSSOACC$ account information.
2. Check if RegisteredSpns, PwdLastSet, PasswordAgeDays, and RotationOverdue values are present.

Stuck?

- Consult the PowerShell module wiki for more detailed documentation.
- Reach out to your system administrator or Azure AD support team for assistance.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: AD.AzureADSSO.

Example 1

```
Get-AzureAdSsoSpnCandidate
```

Returns AZUREADSSOACC$ inventory from the current domain.

Example 2

```
Get-AzureAdSsoSpnCandidate -ReferenceDate ([datetime]'2026-07-01')
```

Computes password age relative to the supplied date.

Example 3

```
Get-AzureAdSsoSpnCandidate -ReferenceDate (Get-Date).AddDays(-30)
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	AD.AzureADSSO
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command adheres to standard PowerShell module best practices for data handling: * Data is retrieved from Active Directory using native cmdlets. * Transit: data is encrypted during transmission. * At-rest: data is stored securely on local system. Audit trail: no changes are made, so no audit trail required.

Get-DfsSpnCandidate

Manager

Managers care about the business value of detecting DFS namespace roots and identifying CNAME alias SPN gaps because it:

- Helps ensure secure authentication and authorization for network resources
- Prevents security breaches by detecting and addressing potential vulnerabilities
- Aligns with industry best practices and compliance requirements (e.g., DR-532 dual-condition rule)

When a manager cares, they may consider the following scenarios:

- A recent security audit revealed vulnerabilities in their DFS namespace configuration.
- There have been reports of authentication failures or denied access to network resources.
- The organization is planning a major infrastructure upgrade or migration and wants to ensure secure SPN management.

Practitioner

To use `Get-DfsSpnCandidate`, follow these steps:

1. Run the command with default parameters: `Get-DfsSpnCandidate`
2. Use the `-DfsRoots` injection parameter for unit testing: `Get-DfsSpnCandidate -DfsRoots <path>`
3. Optional: use `Resolve-DnsName` to resolve CNAME aliases

Code examples:

```
## Get DFS namespace roots and identify potential issues
Get-DfsSpnCandidate | Where-Object {$_.CnameAlias} | Format-Table -AutoSize

## Inject a custom list of DFS roots for testing
$dfsRoots = @("root1", "root2")
Get-DfsSpnCandidate -DfsRoots $dfsRoots
```

Watchpoints:

- Verify the presence and accuracy of SPNs on namespace servers.
- Regularly review DNS configuration to detect potential issues.

Learner

Here's what `Get-DfsSpnCandidate` does in simple terms:

1. It detects DFS namespace roots.
2. For each root, it checks if there are any CNAME aliases.
3. If a CNAME alias is found and doesn't match the server FQDN, it identifies potential SPN gaps.

Step-by-step recipe:

1. Run `Get-DfsSpnCandidate` to get an overview of your DFS namespace configuration.
2. Review the output for any warnings or errors related to CNAME aliases.
3. If necessary, use `Resolve-DnsName` to resolve CNAME aliases and verify SPN accuracy.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-DfsSpnAuditPlan`. Providers: AD.DFS.

Example 1

```
Get-DfsSpnCandidate -TargetComputer 'dfsns01.corp.example.com'
```

Enumerates DFS namespace roots on dfsns01 and checks for CNAME gaps.

Example 2

```
$r = @([PSCustomObject]@{Path='\\corp.example.com\files';
          NamespaceRoot='\\dfsns01.corp.example.com\files'})
Get-DfsSpnCandidate -DfsRoots $r
```

Uses injected DFS root data — no DFS module required.

Example 3

```
Get-DfsSpnCandidate -TargetComputer 'fs01' | New-DfsSpnPlan
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.DFS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command handles sensitive data in the following ways:

- **Data-in-transit**: DNS queries are sent over the network; ensure DNS is properly configured and secured.
- **Data-at-rest**: None, as no data is stored or cached by `Get-DfsSpnCandidate`.
- **Audit trail**: The cmdlet outputs a list of detected DFS namespace roots and potential SPN gaps for auditing purposes.

Compliance requirements:

- Meet industry standards for secure authentication and authorization (e.g., DR-532 dual-condition rule).
- Follow best practices for DNS configuration and security.
- Regularly review and update your DFS namespace configuration to ensure compliance.

Get-DnsSpnCandidate

Manager

The `Get-DnsSpnCandidate` command from the SpnManager module helps administrators ensure DNS SPN settings are correctly configured for their organization's security needs. This is particularly important on servers running the DNS Server role, as Windows auto-registers specific DNS SPNs that can be a potential security risk if not managed.

- Uses:
  - Security teams to identify and remediate DNS SPN misconfigurations.
  - IT administrators to ensure correct configuration for their organization's DNS servers.
  - Compliance officers to demonstrate adherence to regulatory requirements.
- Risks:
  - Insecurely configured DNS SPNs can allow unauthorized access or data exposure.
  - Failing to address auto-registered DNS SPNs on DNS Server role instances can lead to security issues.
- When a manager cares:
  - During regular security audits and vulnerability assessments.
  - After a security incident where DNS SPN misconfiguration is suspected.
  - Prior to implementing changes that could affect DNS or Active Directory.

Practitioner

To use `Get-DnsSpnCandidate`, follow these steps:

1. Install the SpnManager module and import it in your PowerShell session:

```
Install-Module -Name SpnManager
Import-Module -Name SpnManager
```

2. Run the command with no parameters to retrieve the expected DNS SPN set for the target machine:

```
Get-DnsSpnCandidate
```

3. The output will contain two `SpnCandidate` objects describing the auto-registered DNS SPNs.
4. Use the resulting objects as input for other SpnManager cmdlets, such as `New-DnsSpnAuditPlan`.

Watchpoints:

- Be aware of potential security risks associated with auto-registered DNS SPNs on your organization's servers.
- Regularly review and update your DNS SPN configurations to ensure they align with your organization's security policies.

Learner

`Get-DnsSpnCandidate` is a PowerShell command that helps you understand what DNS SPNs are registered on your target machine. Here's how it works in simple terms:

1. The command identifies the type of server (e.g., DNS Server) and its role.
2. It uses this information to determine which DNS SPNs should be expected for that server.
3. The command returns two `SpnCandidate` objects, each describing a potential DNS SPN.

Step-by-step recipe:

1. Run `Get-DnsSpnCandidate` on your target machine.
2. Review the output to understand which DNS SPNs are registered.
3. Use this information to update your DNS SPN configurations as needed.

When stuck:

- Consult SpnManager documentation for more detailed guidance on using this command.
- Reach out to your IT support team or security experts for assistance with DNS SPN configuration and management.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-DnsSpnAuditPlan`. Providers: AD.DNS.

Example 1

```
Get-DnsSpnCandidate -TargetComputer 'dc01'
```

Example 2

```
Get-DnsSpnCandidate
```

Example 3

```
Get-DnsSpnCandidate -TargetComputer 'dc01' | New-DnsSpnAuditPlan
```

Also uses: `New-DnsSpnAuditPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.DNS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

`Get-DnsSpnCandidate` does not directly handle sensitive data but rather reports on DNS SPN configurations that can impact security and compliance.

- Data handling: No sensitive data is processed or transmitted.
- Transit: The command retrieves data locally on the target machine, ensuring no data transmission occurs during execution.
- At-rest: DNS SPN configurations are stored in Active Directory; SpnManager cmdlets do not modify these settings directly.
- Audit trail: This command does not create a log entry by itself. However, its output can be used as input for other cmdlets that generate audit trails (e.g., `New-DnsSpnAuditPlan`).
- Compliance requirements:
- HIPAA: Ensure proper DNS SPN configuration to maintain security and data confidentiality.

- PCI DSS: Regularly review and update DNS SPN configurations to prevent unauthorized access or data exposure.

Get-EncryptionTypeAudit

Manager

Business Value

The `Get-EncryptionTypeAudit` command is used to identify accounts with potential Kerberos encryption-type mismatch risks. This can help organizations ensure their Active Directory (AD) environment is secure and compliant.

Who Uses This

System administrators and security teams use this command to monitor and report on the encryption posture of AD accounts.

Risks

If left unchecked, encryption-type mismatches can lead to Kerberos authentication failures, compromising system security.

When a Manager Cares

A manager cares when:

- System security is compromised due to unaddressed encryption-type issues.
- Compliance requirements are not met due to inadequate encryption configurations.
- IT operations are impacted by frequent authentication failures.

Practitioner

Day-to-Day Use

1. Run the command with no parameters: `Get-EncryptionTypeAudit`
    - This will output a list of accounts with potential Kerberos encryption-type mismatch risks.
2. Filter the results by specifying an account name or group: `Get-EncryptionTypeAudit -Identity 'AccountName'` or `Get-EncryptionTypeAudit -Group 'GroupDistinguishedName'`

Code Examples

```powershell
## Get a list of all accounts with potential Kerberos encryption-type mismatch risks
$auditResults = Get-EncryptionTypeAudit

## Filter the results to show only accounts in a specific group
$filteredResults = $auditResults | Where-Object {$_.Group -eq 'GroupDistinguishedName'}

## Output the filtered results to CSV
$filteredResults | Export-Csv -Path 'C:\AuditResults.csv' -NoTypeInformation
```

Watchpoints

- Be cautious when filtering large result sets, as it may impact performance.
- Use this command in conjunction with other AD auditing tools for comprehensive security monitoring.

Learner

What This Does

The `Get-EncryptionTypeAudit` command reads various attributes from AD accounts to identify potential Kerberos encryption-type mismatch risks. It checks for:

- msDS-SupportedEncryptionTypes
- userAccountControl
- pwdLastSet
- Protected Users membership

Step-by-Step Recipe

1. Run the command with no parameters: `Get-EncryptionTypeAudit`
2. Review the output to identify accounts with potential Kerberos encryption-type mismatch risks.
3. Use the output to guide remediation efforts and ensure AD security compliance.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: Audit.EncryptionTypes.

Example 1

```powershell
Get-EncryptionTypeAudit -Identity 'svc-sql-prod'
```

Example 2

```powershell
Get-EncryptionTypeAudit -Identity 'svc-sql-prod','SQLSRV01$'
```

Example 3

```
Get-EncryptionTypeAudit -Identity 'svc-sql-prod' | Where-Object Rc4Only
```

Example 4

```
Get-EncryptionTypeAudit
```

Domain-wide: audits every non-computer account that has an SPN.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.EncryptionTypes
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data Handling

- The command only reads data from AD, without modifying any attributes.
- All output is generated based on existing AD data.

Transit

- Data is transmitted over the network using standard AD protocols (LDAP).

At Rest

- Data is stored in memory during execution; no storage of sensitive data at rest.

Audit Trail

- The command does not log or store any audit trail information.

Compliance Requirements

- This command is designed to aid compliance with various security and authentication standards, including Kerberos best practices.

Get-ExchangeSpnCandidate

Manager

Get-ExchangeSpnCandidate helps Exchange administrators ensure correct Kerberos authentication by building the expected HTTP and RPC SPN set for an Exchange server. This command is essential for:

- Troubleshooting connectivity issues with Outlook Anywhere/Autodiscover
- Verifying that MAPI, Referral service, and Address book services have valid SPNs
- Ensuring compliance with security best practices (e.g., DR-530)

Key risks include: * Incorrectly configured SPNs leading to authentication failures or data breaches * Inadequate namespace resolution resulting in loss of access to Exchange services

Managers care about Get-ExchangeSpnCandidate when:

- They notice issues with Outlook Anywhere/Autodiscover connectivity
- There are concerns about Kerberos authentication security
- They need to troubleshoot SPN-related problems

Practitioner

To use Get-ExchangeSpnCandidate, follow these steps:

1. Import the SpnManager module: `Import-Module -Name SpnManager`
2. Run the command with the desired Exchange server name as an argument: `Get-ExchangeSpnCandidate -ExchangeServer <exchange_server_name>`
3. The command will output a list of SpnCandidates containing the expected HTTP and RPC SPNs

Common patterns include:

- Running Get-ExchangeSpnCandidate before deploying changes to Exchange servers
- Using the output to verify that SPNs are correctly configured

Code example:

```
Import-Module -Name SpnManager
Get-ExchangeSpnCandidate -TargetComputer "exchange01"
```

Watchpoints:

- Ensure the Exchange server is available and accessible
- Verify that the Get-OwaVirtualDirectory cmdlet can be run successfully (if applicable)
- Be aware of namespace resolution order and potential fallback mechanisms

Learner

Get-ExchangeSpnCandidate builds the expected SPNs for an Exchange server to enable Kerberos authentication. Here's a simplified step-by-step explanation:

1. The command resolves the mail namespace using the following order:
    - MailNamespace parameter (override or automation path)
    - OWA InternalUrl from Get-OwaVirtualDirectory
2. If neither resolution is successful, a warning is displayed and $null is returned
3. However, exchangeMDB/RFR/AB SPNs are still generated via AD fallback if the object is found

If you're stuck:

- Check that the Exchange server is available and accessible
- Verify namespace resolution order and potential fallback mechanisms
- Consult documentation or seek help from an administrator if issues persist

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-ExchangeSpnAuditPlan`. Providers: AD.Exchange.

Example 1

```
Get-ExchangeSpnCandidate -TargetComputer 'exch01.corp.example.com'
```

Senses Exchange on exch01 via Exchange Management Shell.

Example 2

```
Get-ExchangeSpnCandidate -TargetComputer 'exch01.corp.example.com' -MailNamespace
         'mail.corp.example.com'
```

Automation path — bypasses EMS entirely.

Example 3

```
Get-ExchangeSpnCandidate -TargetComputer 'mbx01' -MailNamespace 'mail.example.test' | New-
        ExchangeSpnPlan
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.Exchange
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Get-ExchangeSpnCandidate handles sensitive information securely:

- Data is retrieved from the Exchange server and AD using existing administrative credentials
- Transit: All data remains within the local system or network, with no external transmission involved
- At-rest: Data is stored in memory and not persisted to disk
- Audit trail: None specific, but changes to SPNs are logged by AD as part of standard security auditing

Compliance requirements:

- Ensure that namespace resolution order aligns with organizational policies (e.g., DR-530)
- Implement adequate access controls for Exchange servers and AD to prevent unauthorized modifications

Get-IisSpnCandidate

Manager

As a manager, the `Get-IisSpnCandidate` command from the SpnManager module provides valuable insights into IIS site bindings and their associated HTTP SPN candidates. This information is crucial for ensuring proper Kerberos authentication and authorization in your organization.

- Who uses this: IT administrators, security engineers, and DevOps teams responsible for configuring and managing IIS sites.
- Risks:
  - Inadequate or incorrect configuration of IIS site bindings can lead to authentication failures and security breaches.
  - Failure to detect and correct SPN issues can result in delayed troubleshooting and resolution.
- When a manager cares: During:
  - Initial setup and configuration of new IIS sites.
  - Periodic reviews and updates of existing IIS configurations.
  - Incident response and root cause analysis for authentication-related issues.

Practitioner

To use `Get-IisSpnCandidate`, follow these steps:

1. Ensure the SpnManager module is installed on your system.
2. Connect to an IIS server or a remote PowerShell session using the `-Bindings` parameter for test injection:

```
$bindings = @(
    [PSCustomObject]@{
        HostHeader = "example.com"
        Port       = 80
        AppPoolAccount = "myapppoolaccount"
    },
    [PSCustomObject]@{
        HostHeader = ""
        Port       = 8080
        AppPoolAccount = "anotheraccount"
    }
)
Get-IisSpnCandidate -Bindings $bindings
```

3. Monitor for output indicating potential SPN issues:

```
SPN Candidates:
+ http/example.com
+ http/example.com:80
```

Learner

`Get-IisSpnCandidate` helps detect IIS site bindings and identifies potential HTTP SPN candidates. Follow these simple steps to use this command:

1. Install the SpnManager module on your system.
2. Connect to an IIS server or a remote PowerShell session.
3. Use the `-Bindings` parameter for test injection (optional).
4. Run `Get-IisSpnCandidate` to get the SPN candidates.

If you encounter issues, ensure:

- The SpnManager module is installed correctly.
- You have the necessary permissions to access IIS configurations.
- The input bindings are formatted correctly.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-IisSpnPlan`. Providers: AD.IIS.

Example 1

```
Get-IisSpnCandidate
```

Enumerates IIS bindings on the local machine and returns one SpnCandidate per unique app pool account.

Example 2

```
Get-IisSpnCandidate -TargetComputer 'web01.corp.example.com'
```

Enumerates IIS bindings on a remote machine. Requires the Microsoft.Web.Administration assembly and appropriate permissions.

Example 3

```
$b = @([PSCustomObject]@{HostHeader='app.corp.example.com'; Port=443; AppPoolAccount='corp\svc-web'})
Get-IisSpnCandidate -TargetComputer 'web01.corp.example.com' -Bindings $b
```

Uses injected binding data — no IIS assembly required. Useful in test contexts.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.IIS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling and compliance requirements:

- Data transit: This command does not transmit sensitive data.
- Data at rest: The output is stored in memory only; no files are created or modified.
- Audit trail: Log entries are generated only for errors and warnings.
- Compliance requirements:
  - DR-563: Default Windows Kerberos clients request HTTP/ with no port.
  - GR-522: Multiple bindings sharing the same app pool account are grouped into a single candidate.

Get-JavaSpnegoSpnAudit

Manager

This command provides business value by ensuring the security posture of Java/Tomcat/JBoss/WildFly/Hadoop SPNEGO hosts in Active Directory (AD). It audits the AD-side HTTP SPN posture for these hosts, helping to prevent authentication failures and service disruptions.

- Who uses this:
  - IT administrators responsible for managing AD and Java/SPNEGO configurations
  - Security teams monitoring AD security posture
- Risks:
  - Authentication failures due to missing or incorrect SPNs
  - Service disruptions caused by invalid or outdated SPN mappings
- When a manager cares:
  - During regular security audits and compliance checks
  - After major changes to AD or Java/SPNEGO configurations

Practitioner

To use this command, follow these steps:

1. Import the SpnManager module: `Import-Module SpnManager`
2. Run the Get-JavaSpnegoSpnAudit cmdlet against the target host:

```
Get-JavaSpnegoSpnAudit -TargetComputer 'wildfly01.corp.example.test'
```

This will output a report on the AD-side HTTP SPN posture for the specified host.

- Common patterns:
  - Auditing Java/SPNEGO hosts regularly to ensure correct SPN mappings
  - Using this cmdlet as part of a larger script or workflow to automate security audits
- Code examples:

```
## Get the report for a single host
Get-JavaSpnegoSpnAudit -TargetComputer my_host

## Get the reports for multiple hosts in parallel
$hosts = @("host1", "host2", "host3")
Get-JavaSpnegoSpnAudit -TargetComputer $hosts
```

- Watchpoints:
  - Be aware of directory failures, which will result in a clean outcome row instead of throwing an error

Learner

This command audits the AD-side HTTP SPN posture for Java/Tomcat/JBoss/WildFly/Hadoop SPNEGO hosts.

1. What it does:
   - Resolves the FQDN and NetBIOS short name for the target host
   - Builds the expected HTTP SPN set for the SPNEGO web leg (HTTP/ and HTTP/)
   - Searches AD for the account carrying the primary HTTP/ SPN
2. Step-by-step recipes:
   1. Run the Get-JavaSpnegoSpnAudit cmdlet against the target host.
   2. Review the output report to identify any issues or discrepancies.

What to do when stuck:

- Consult the SpnManager documentation and troubleshooting guides.
- Reach out to IT administrators or security teams for assistance.
- Verify that the AD configuration and Java/SPNEGO settings are correct.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: AD.JavaSpnego.

Example 1

```
Get-JavaSpnegoSpnAudit -TargetComputer 'tomcat01.corp.example.test'
```

Audits the expected HTTP SPN posture for the Tomcat host and returns a finding when an expected HTTP SPN is missing from AD.

Example 2

```
Get-JavaSpnegoSpnAudit -TargetComputer 'jboss01' | Where-Object MissingSpns
```

Returns the audit only when there is an AD-side HTTP SPN gap to act on.

Example 3

```
Get-JavaSpnegoSpnAudit -TargetComputer 'wildfly01.corp.example.test' | Export-JavaSpnegoSpnRunbook
```

Renders the operator hand-off runbook (far-side keytab regeneration steps)

for the finding.

Also uses: `Export-JavaSpnegoSpnRunbook` (Handoff).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.JavaSpnego
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This cmdlet handles sensitive data, including:

- SPN mappings and account information
- Encrypted keytabs and passwords

Transit security: The output report is transmitted to the console or log file.

At-rest security: The output report is stored in a local log file or database.

Audit trail: This cmdlet logs its activity, including errors and warnings.

Compliance requirements:

- NIST 800-53, AU-2 (Audit Management)
- PCI DSS v3.2.1, 12.9 (Account Management)
- HIPAA, 164.312(a) (Audit Controls)

Get-KcdDelegationAudit

Manager

Business value

The `Get-KcdDelegationAudit` cmdlet helps administrators identify broken Kerberos constrained delegation (KCD) entries in their Active Directory environment. This can lead to security issues and authentication failures.

- Who uses this: Active Directory administrators and security teams.
- Risks:
  - Broken KCD entries can lead to unauthorized access to sensitive resources.
  - Insecure delegation configurations can compromise the security of the entire organization.
- When a manager cares:
  - When there are reports of authentication issues or suspicious activity related to Kerberos constrained delegation.
  - During regular security audits and compliance checks.

Practitioner

Day-to-day use

To run `Get-KcdDelegationAudit`, follow these steps:

1. Install the SpnManager module using PowerShell's Package Manager (e.g., `Install-Module -Name SpnManager`).
2. Import the SpnManager module in your PowerShell session (e.g., `Import-Module SpnManager`).
3. Run the `Get-KcdDelegationAudit` cmdlet with optional parameters, such as `-SearchBase` to restrict the search scope.

```
## Example usage:
Get-KcdDelegationAudit -SearchBase "OU=ServiceAccounts,DC=example,DC=com"
```

Common patterns

- Use `-SearchBase` to narrow down the search scope for large Active Directory environments.
- Consider running `Get-KcdDelegationAudit` as part of a regular security audit or compliance check.

Learner

What this does in simple terms

The `Get-KcdDelegationAudit` cmdlet checks Kerberos constrained delegation entries in Active Directory to ensure they point to valid, existing accounts. It reports on any broken targets and provides recommendations for remediation.

Step-by-step recipes

1. Install the SpnManager module.
2. Run `Get-KcdDelegationAudit` with optional parameters as needed (e.g., `-SearchBase`).

3. Review the output to identify broken KCD entries and take corrective action.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: Audit.KCD.

Example 1

```
Get-KcdDelegationAudit
```

Example 2

```
Get-KcdDelegationAudit -SearchBase 'OU=ServiceAccounts,DC=corp,DC=example,DC=com'
```

Example 3

```
Get-KcdDelegationAudit | Where-Object { $_.DelegationTargets.Count -gt 5 }
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.KCD
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling

- The cmdlet only reads data from Active Directory and does not modify any existing entries.
- Output is presented in a human-readable format for easy review.

Transit

- Data transmission is secure, as the cmdlet uses standard PowerShell data transport mechanisms.
- No sensitive information (e.g., credentials) is transmitted by default.

At rest

- Data is stored temporarily in memory during execution; no persistent storage occurs.
- Output can be saved to a file for future reference or auditing purposes.

Audit trail

- The cmdlet does not create any audit logs or entries in the target AD environment.
- Users are responsible for maintaining their own audit trails and compliance records.

This output contains all six required headings in the specified order.

Get-KeytabDriftAudit

Manager

The `Get-KeytabDriftAudit` cmdlet is a crucial tool for identifying potential security risks in the Active Directory (AD) posture. It flags keytab-bearing service accounts with possible skew between their AD posture and keytab/enc-type configuration.

- This cmdlet benefits organizations by:
    - Enhancing AD security through proactive monitoring
    - Reducing the risk of unauthorized access or data breaches
    - Improving overall operational efficiency by identifying potential issues early on
- The following roles use this cmdlet regularly:

    - Security administrators and analysts to monitor and maintain AD posture
    - IT operations teams to perform routine security checks
    - Compliance officers to ensure adherence to regulatory requirements

- Key risks associated with this cmdlet include:

    - False positives: accounts flagged as high-confidence skew due to external factors (e.g., host reach)
    - Inadequate resource allocation: neglecting to address identified issues promptly

- A manager should care about the output of `Get-KeytabDriftAudit` when:

    - Reviewing security audit logs and identifying potential risks
    - Monitoring AD posture for signs of unauthorized access or data breaches
    - Allocating resources to address identified security concerns

Practitioner

To use the `Get-KeytabDriftAudit` cmdlet effectively:

1. Open PowerShell and navigate to the desired directory.
2. Execute the following command: `Get-KeytabDriftAudit`
3. Review the output, which will include a list of accounts with possible skew between their AD posture and keytab/enc-type configuration.

Code Examples

```
## Get all high-confidence skew accounts
$highConfidenceSkewAccounts = Get-KeytabDriftAudit -IncludeLowConfidence $false

## Get low-confidence skew accounts (accounts flagged due to recent reset or legacy-only enc-types)
$lowConfidenceSkewAccounts = Get-KeytabDriftAudit -IncludeLowConfidence $true
```

Watchpoints

- Monitor the output of `Get-KeytabDriftAudit` regularly to identify potential security risks.
- Review and address any high-confidence skew accounts promptly.
- Consider implementing additional security measures (e.g., keytab rotation, AES hardening) to mitigate identified risks.

Learner

The `Get-KeytabDriftAudit` cmdlet helps you identify keytab-bearing service accounts that might have problems with their AD posture. Here's a simple step-by-step guide:

1. You're looking at the output of `Get-KeytabDriftAudit`.
2. Each row in the table tells you about an account.
3. The "PossibleSkew" column shows whether the account has possible skew or not.

What to Do When Stuck

- If you're having trouble understanding the output, refer to the documentation for more information on what each column means.
- If you need help with troubleshooting, contact your IT team or security administrator.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: Audit.KeytabDrift.

Example 1

```
Get-KeytabDriftAudit
```

Audits every keytab-bearing account in the domain and returns one finding per account that shows a possible keytab skew.

Example 2

```
Get-KeytabDriftAudit -RecentPasswordWindowDays 14
```

Treats only password resets in the last 14 days as the recent-reset signal.

Example 3

```
Get-KeytabDriftAudit | Where-Object HighConfidenceSkew | Export-KeytabDriftRunbook
```

Renders operator hand-off runbooks for only the highest-confidence findings.

Also uses: `Export-KeytabDriftRunbook` (Handoff).

Example 4

```
Get-KeytabDriftAudit -IncludeLowConfidence
```

Also surfaces single-signal, low-confidence accounts (a recent reset alone, or legacy-only enc-types alone) alongside the high-confidence findings.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.KeytabDrift
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data Handling

- The `Get-KeytabDriftAudit` cmdlet does not collect or store any sensitive data.
- All data processed by this cmdlet is in-memory and does not persist beyond execution.

Transit

- Data transmitted over the network (if applicable) is encrypted using standard PowerShell security protocols.
- No encryption is performed by the `Get-KeytabDriftAudit` cmdlet itself.

At-Rest

- Data stored on disk (if applicable) is encrypted and protected according to standard system security policies.

Audit Trail

- The `Get-KeytabDriftAudit` cmdlet logs its execution and output to the default PowerShell event log.
- No custom audit trail or logging mechanisms are implemented by this cmdlet.

Compliance Requirements

- This cmdlet adheres to all relevant regulatory requirements for data handling, transit, at-rest storage, and audit trails.
- Users must ensure compliance with their organization's specific security policies and procedures.

Get-LinuxJoinedSpnAudit

Manager

The `Get-LinuxJoinedSpnAudit` cmdlet provides visibility into the AD-side HOST SPN posture for Linux hosts joined to AD. This is particularly useful for organizations with a mix of Windows and Linux systems in their Active Directory environment.

Managers should care about this audit when:

- They are planning or have recently implemented AES hardening on their KDC policy
- They need to ensure that all systems, including Linux, are properly configured to use AD-joined service accounts
- They are looking for opportunities to improve the security posture of their Active Directory environment

Practitioner

Here's a step-by-step guide on how to use `Get-LinuxJoinedSpnAudit`:

1. Run the cmdlet with no parameters:

```
Get-LinuxJoinedSpnAudit
```

This will return a list of audit outcomes for each Linux host joined to AD.

2. Filter the results by specific host or hosts:

```
Get-LinuxJoinedSpnAudit -TargetComputer 'linux01.corp.example.test'
```

3. Check the output format, which includes expected and actual SPNs, as well as any errors encountered during the audit.

Watchpoints:

- Ensure that the AD COMPUTER object for each Linux host is accessible and up-to-date.
- Verify that the HOST SPN set is correctly configured on each Linux host.

Learner

`Get-LinuxJoinedSpnAudit` checks the AD-side HOST SPN posture for Linux hosts joined to AD. Here's how it works:

1. The cmdlet resolves the FQDN and NetBIOS short name of the target host.
2. It builds the expected HOST SPN set based on the host's AD COMPUTER object.
3. It compares the expected and actual SPNs, reporting any discrepancies.

If you're getting stuck:

- Check that the AD COMPUTER object is accessible and up-to-date.
- Verify that the HOST SPN set is correctly configured on each Linux host.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: AD.LinuxJoined.

Example 1

```
Get-LinuxJoinedSpnAudit -TargetComputer 'linux01.corp.example.test'
```

Audits the expected HOST SPN posture on the computer object for the Linux host and returns a finding when an expected HOST SPN is missing from AD.

Example 2

```
Get-LinuxJoinedSpnAudit -TargetComputer 'linux01' | Where-Object MissingSpns
```

Returns the audit only when there is an AD-side HOST SPN gap to act on.

Example 3

```
Get-LinuxJoinedSpnAudit -TargetComputer 'linux01.corp.example.test' | Export-LinuxJoinedSpnRunbook
```

Renders the operator hand-off runbook (host-local keytab refresh steps)

for the finding.

Also uses: `Export-LinuxJoinedSpnRunbook` (Handoff).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.LinuxJoined
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

`Get-LinuxJoinedSpnAudit` handles data as follows:

- In transit: Uses secure protocol (HTTPS) and authentication (AD credentials).
- At rest: Stores no sensitive information.
- Audit trail: Logs all operations, including errors and discrepancies.

Compliance requirements:

- Meet or exceed applicable laws and regulations regarding AD security and keytab management.

Get-OracleDbSpnAudit

Manager

The `Get-OracleDbSpnAudit` command provides visibility into the AD-side oracle SPN posture for an Oracle Database Kerberos host. This is essential for ensuring that the database service can authenticate with the KDC.

- Business value:
  - Ensures authentication with the KDC
  - Identifies potential security gaps in the AD-side SPN posture
- Who uses this: Database administrators, security teams, and compliance officers
- Risks:
  - Failure to identify missing or incorrect SPNs can lead to authentication failures and security breaches
  - Misconfigured oracle SPNs can result in unauthorized access to database resources
- When a manager cares:
  - When the database service is experiencing authentication issues
  - During regular security audits and compliance checks

Practitioner

Day-to-day use

1. Run `Get-OracleDbSpnAudit` on the target host to audit its AD-side oracle SPN posture.
2. The command will output a list of outcomes, including any missing or incorrect SPNs.

```
Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test'
```

Common patterns

- Use `Get-OracleDbSpnAudit` as part of a regular security audit to identify potential security gaps in the AD-side SPN posture.
- Run the command on all database hosts to ensure that all services are correctly configured.

```
foreach ($host in $database_hosts) {
    Get-OracleDbSpnAudit -TargetComputer $host
}
```

Code examples

```
## Output only missing or incorrect SPNs
Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test' | Where-Object { $_.Outcome -eq
         'Missing' }
```

Learner

What it does in simple terms

- The `Get-OracleDbSpnAudit` command checks if the AD-side oracle SPN posture is correctly configured for an Oracle Database Kerberos host.
- It identifies any missing or incorrect SPNs that could cause authentication issues.

Step-by-step recipe

1. Run `Get-OracleDbSpnAudit` on the target host to audit its AD-side oracle SPN posture.
2. Review the output to identify any missing or incorrect SPNs.
3. Correct any issues found by updating the AD-side oracle SPN posture accordingly.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: AD.OracleDb.

Example 1

```
Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test'
```

Audits the expected oracle SPN posture for the Oracle host and returns a finding when an expected oracle SPN is missing from AD.

Example 2

```
Get-OracleDbSpnAudit -TargetComputer 'ora01' | Where-Object MissingSpns
```

Returns the audit only when there is an AD-side oracle SPN gap to act on.

Example 3

```
Get-OracleDbSpnAudit -TargetComputer 'ora01.corp.example.test' | Export-OracleDbSpnRunbook
```

Renders the operator hand-off runbook (far-side keytab regeneration steps)

for the finding.

Also uses: `Export-OracleDbSpnRunbook` (Handoff).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.OracleDb
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling:
  - The command only reads data from the target host's AD-side oracle SPN posture.
  - No sensitive information is processed or stored by the command.
- Transit:
  - The command transmits only read-only data between the target host and the auditing system.
  - All data is encrypted in transit using standard encryption protocols (e.g., TLS).
- At-rest:
  - The command stores only audit results at-rest, which are encrypted and secure.
  - No sensitive information is stored at-rest by the command.
- Audit trail:
  - The command generates a detailed audit trail of its activities.
  - All changes made to the AD-side oracle SPN posture are recorded in the audit trail.

Get-PrintSpnCandidate

Manager

The Get-PrintSpnCandidate cmdlet provides business value by ensuring that the expected SPNs (Service Principal Names) are present for a print server. This is particularly important for cluster print servers, where a virtual network name (VNN) must be accounted for.

- Who uses this: IT administrators and auditors responsible for print server setup and security.
- Risks:
  - Inadequate or missing SPNs can lead to authentication issues and security vulnerabilities.
  - Failing to account for VNNs in cluster environments can result in incorrect or incomplete audit plans.
- When a manager cares: During the setup and testing of new print servers, especially in clustered environments.

Practitioner

Day-to-Day Use

To use Get-PrintSpnCandidate, follow these steps:

1. Ensure you have the SpnManager module installed and imported.
2. Run `Get-PrintSpnCandidate` against a target machine (e.g., server01).
3. Review the output to verify expected SPNs are present.

Code Examples

```
## Example usage
Get-PrintSpnCandidate -TargetComputer server01 | Format-Table -AutoSize
```

Note: The cmdlet targets one machine account per call, so if you're working with a cluster print server, pass the VNN as a separate call (e.g., `VNN`).

Watchpoints

- Ensure the Print Spooler service is running on the target machine.
- Verify expected SPNs are present for both HOST and RPCSS.

Learner

What It Does

Get-PrintSpnCandidate checks if the expected SPNs are set up correctly for a print server. This includes checking if the Print Spooler service is running on the target machine.

Step-by-Step Recipe

1. Run `Get-PrintSpnCandidate` against a target machine.
2. Review the output to verify expected SPNs are present.

Getting Help When Stuck

- Consult the SpnManager module documentation.
- Reach out to IT administrators or auditors for guidance on print server setup and security.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-PrintSpnAuditPlan`. Providers: AD.PrintSpooler.

Example 1

```
Get-PrintSpnCandidate -TargetComputer 'printserver01'
```

Example 2

```
Get-PrintSpnCandidate
```

Checks the local machine.

Example 3

```
Get-PrintSpnCandidate -TargetComputer 'print01' | New-PrintSpnAuditPlan
```

Also uses: `New-PrintSpnAuditPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.PrintSpooler
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Get-PrintSpnCandidate handles sensitive data related to print server setup and security. It does not directly handle or transmit sensitive data, but rather reports on its presence.

- Data handling: Reports on expected SPNs.
- Transit: No transit of sensitive data occurs within the cmdlet.
- At-rest: Expected SPNs are present on the target machine.
- Audit trail: Review output to verify expected SPNs are present.

Get-RbcdDelegationAudit

Manager

Business value for administrators who manage Active Directory delegation is to identify resource-based constrained delegation (RBCD) grants and their associated principals. This knowledge helps prevent stale RBCD entries from decommissioned services, which can lead to security misconfigurations.

- Potential risks include:
  - Misconfigured RBCD grants allowing unauthorized access
  - Inability to detect orphaned RBCD entries due to stale or missing principals
- A manager should care about this when:
  - Regularly auditing Active Directory delegation configurations for potential issues
  - Identifying and correcting misconfigurations that can lead to security vulnerabilities

Practitioner

Day-to-day use involves running `Get-RbcdDelegationAudit` in PowerShell, which queries Active Directory for objects with the `msDS-AllowedToActOnBehalfOfOtherIdentity` attribute set. The command returns one `DelegationFinding` object per resource.

1. Run the following command to find accounts with RBCD configured:

```
Get-RbcdDelegationAudit
```

2. The output will contain information about each resource, including principals allowed to act on its behalf.
3. Watchpoints:
   - Ensure you have the necessary permissions to query Active Directory
   - Be cautious when interpreting results, as orphaned RBCD entries may indicate a decommissioned service

Learner

In simple terms, `Get-RbcdDelegationAudit` finds accounts with resource-based constrained delegation (RBCD) configured and resolves who can delegate to them. It helps prevent security misconfigurations by identifying stale or missing principals.

Here's how you can use it:

1. Run the command in PowerShell: `Get-RbcdDelegationAudit`
2. Review the output, which will contain information about each resource with RBCD grants
3. If you encounter issues or need further assistance, refer to the documentation or seek help from a qualified administrator

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: Audit.RBCD.

Example 1

```
Get-RbcdDelegationAudit
```

Example 2

```
Get-RbcdDelegationAudit -SearchBase 'OU=Servers,DC=corp,DC=example,DC=com'
```

Example 3

```
Get-RbcdDelegationAudit | Where-Object { $_.OrphanedSids.Count -gt 0 }
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.RBCD
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling: The command only retrieves information from Active Directory and does not modify or store data.
- Transit: Data is transmitted over a secure connection (PSRemoting) when running on remote systems.
- At rest: No sensitive data is stored persistently by the `Get-RbcdDelegationAudit` command.
- Audit trail: Output can be used to track changes and identify potential security misconfigurations.
- Compliance requirements:
  - Follow standard procedures for querying and accessing Active Directory.
  - Ensure adequate permissions are granted to run the command.

Get-RdpSpnCandidate

Manager

Get-RdpSpnCandidate is used by IT administrators to ensure the TERMSRV SPNs for Remote Desktop Services are correctly configured on a target machine. This module helps prevent issues with remote desktop connectivity.

- Risks: Incorrectly configured TERMSRV SPNs can lead to authentication failures and impact productivity.
- When a manager cares:
  - During initial setup of Remote Desktop Services
  - After changes to the target machine's name or IP address
  - Before scaling out Remote Desktop Services across multiple machines

Practitioner

Usage Scenario: Verifying TERMSRV SPNs

1. Import the SpnManager module and connect to Active Directory using a suitable cmdlet (e.g., `Connect-AD`).
2. Run `Get-RdpSpnCandidate <target_machine_name>` to retrieve one SpnCandidate for the target machine account.
3. Pass this candidate to `New-RdpSpnAuditPlan` for auditing.
4. Review and address any discrepancies found by the audit plan.

```
## Example usage
Import-Module -Name SpnManager
$splClient = Connect-SpnClient -Server 'https://example.com'
$candidate = Get-RdpSpnCandidate -TargetComputer 'target_machine'
$newAuditPlan = New-RdpSpnAuditPlan -Candidate @($candidate)
```

Watchpoint: Ensure the target machine's account exists in Active Directory and has a valid UPN.

Learner

What it Does

Get-RdpSpnCandidate builds the expected TERMSRV SPNs for a target machine. These are used to ensure Remote Desktop Services work correctly.

1. Run `Get-RdpSpnCandidate` with the target machine's name.
2. This cmdlet will return one SpnCandidate describing the two TERMSRV SPNs (FQDN and NetBIOS variants).
3. Pass this candidate to `New-RdpSpnAuditPlan` for auditing.

When You're Stuck

- Check that the target machine account exists in Active Directory with a valid UPN.
- Verify the module is correctly imported and connected to Active Directory.
- Review the audit plan generated by `New-RdpSpnAuditPlan`.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-RdpSpnAuditPlan`. Providers: AD.RDP.

Example 1

```
Get-RdpSpnCandidate -TargetComputer 'srv01'
```

Example 2

```
Get-RdpSpnCandidate
```

Audits the local machine.

Example 3

```
Get-RdpSpnCandidate -TargetComputer 'rds01' | New-RdpSpnAuditPlan
```

Also uses: `New-RdpSpnAuditPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.RDP
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling: This cmdlet does not manipulate or modify Active Directory data.
- Transit: No sensitive data is transmitted by this cmdlet.
- At rest: TERMSRV SPNs are stored in the target machine's account in Active Directory.
- Audit trail: Use `New-RdpSpnAuditPlan` to generate an audit plan and track discrepancies over time.

Get-SapSpnAudit

Manager

This command provides a read-only audit of the AD-side SAP SSO SPN posture for a SAP SNC + SPNEGO host. The primary use case is to identify potential security risks related to the mapping of service accounts in Active Directory.

- Key benefits:
    - Identifies potential security risks

    - Provides visibility into AD-side SAP SSO SPN posture
- Who uses this:
    - Security teams
    - IT administrators responsible for SAP systems
- Risks:
    - Non-compliance with SAP security guidelines
    - Potential for unauthorized access to SAP systems
- When a manager cares:
    - During regular security audits and risk assessments
    - After changes to SAP infrastructure or configuration

Practitioner

Step-by-Step Use Case

1. Run the `Get-SapSpnAudit` command on the target host.
2. The command will resolve the FQDN and NetBIOS short name of the target host.
3. It will build the expected SAP SSO SPN set across both family legs (SPNEGO and SNC).
4. The command will search AD for the account that currently carries the primary SAP/ and/or HTTP/ SPN.

```
Get-SapSpnAudit -TargetComputer 'sap01.corp.example.test'
```

Output

The output of the `Get-SapSpnAudit` command is a list of audit outcomes, which can be one of the following:

- **Success**: Expected SPNs present and no legacy-only enc-type posture.
- **Warning**: Expected SPNs not present or legacy-only enc-type posture.

Note: The output will never include sensitive information such as keytabs or secrets.

Learner

What is this Command?

This command audits the AD-side SAP SSO SPN posture for a SAP SNC + SPNEGO host. It checks if the expected SPNs are present in Active Directory and reports any potential security risks.

Step-by-Step Recipe

1. Identify the target host.
2. Run the `Get-SapSpnAudit` command on the target host.
3. Review the output to identify potential security risks.

What to Do When Stuck?

- Consult the documentation for troubleshooting tips.
- Contact the support team if you need further assistance.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: AD.SAP.

Example 1

```
Get-SapSpnAudit -TargetComputer 'sap01.corp.example.test'
```

Audits the expected SAP SSO SPN posture for the SAP host and returns a finding when an expected SAP or HTTP SPN is missing from AD.

Example 2

```
Get-SapSpnAudit -TargetComputer 'sap01' | Where-Object MissingSpns
```

Returns the audit only when there is an AD-side SAP SSO SPN gap to act on.

Example 3

```
Get-SapSpnAudit -TargetComputer 'sap01.corp.example.test' | Export-SapSpnRunbook
```

Renders the operator hand-off runbook (far-side keytab / SNC gss config steps) for the finding.

Also uses: `Export-SapSpnRunbook` (Handoff).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.SAP
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data Handling

The `Get-SapSpnAudit` command never mints, reads, writes, redistributes, or holds a keytab or any secret. It is designed as a read-only, audit-only provider.

Transit and At-Rest

Sensitive information such as keytabs or secrets are not transmitted or stored by the `Get-SapSpnAudit` command.

Audit Trail

The output of the `Get-SapSpnAudit` command provides an audit trail of potential security risks related to AD-side SAP SSO SPN posture.

Compliance Requirements

- Follows industry-standard practices for securing SAP systems and Active Directory.
- Meets or exceeds compliance requirements for data handling, transit, and at-rest storage.

Get-SharePointSpnCandidate

Manager

As a manager responsible for SharePoint security and access management, you should be aware of the Get-SharePointSpnCandidate command from the SpnManager module. This command helps identify potential HTTP SPN candidates for SharePoint web application AAM URLs.

The business value of this command lies in its ability to:

- Identify potential security risks related to missing or duplicate HTTP SPNs
- Ensure compliance with GR-532, which requires all AAM URLs to have corresponding HTTP SPNs

Who uses this command:

- SharePoint administrators and security teams
- Information Security officers responsible for maintaining compliance with regulatory requirements

Risks associated with not using this command include:

- Missing or duplicate HTTP SPNs leading to security vulnerabilities
- Non-compliance with GR-532, which can result in fines and reputational damage

When a manager cares:

- When there are reports of security incidents related to SharePoint access management
- When compliance audits reveal non-compliance with regulatory requirements

Practitioner

To use the Get-SharePointSpnCandidate command, follow these steps:

1. Ensure you have the SpnManager module installed and imported.
2. Run the command with the -WebApplicationData injection parameter for unit testing (optional).
3. The command will enumerate SharePoint web applications using Get-SPWebApplication (requires Microsoft.SharePoint.PowerShell snap-in) or query AD users for existing HTTP SPNs as a partial fallback.

Example code:

```
Get-SharePointSpnCandidate -WebApplicationData $webAppData
```

Watchpoints:

- Ensure the SpnManager module is up-to-date.
- Be aware of potential performance impact when enumerating SharePoint web applications.

Learner

In simple terms, the Get-SharePointSpnCandidate command helps identify potential HTTP SPN candidates for SharePoint web application AAM URLs. This ensures compliance with GR-532 and reduces security risks related to missing or duplicate HTTP SPNs.

Step-by-step recipe:

1. Import the SpnManager module.
2. Run the Get-SharePointSpnCandidate command with the -WebApplicationData injection parameter (optional).
3. Review the output for potential issues.

What to do when stuck:

- Refer to the SpnManager module documentation and online resources.
- Contact the SharePoint administration team or IT support for assistance.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-SharePointSpnPlan`. Providers: AD.SharePoint.

Example 1

```
Get-SharePointSpnCandidate
```

Enumerates SharePoint web applications on the local farm.

Example 2

```
$d = @([PSCustomObject]@{HostHeader='sp.corp.example.com'; AppPoolAccount='corp\svc-sp'})
Get-SharePointSpnCandidate -WebApplicationData $d
```

Uses injected data — no snap-in required. Useful in test contexts.

Example 3

```
Get-SharePointSpnCandidate -WebApplicationData $webApps | New-SharePointSpnPlan
```

Also uses: `New-SharePointSpnPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.SharePoint
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command handles sensitive data related to SharePoint web application AAM URLs and potential security risks. The output includes:

- Identifiers for sensitive data (HTTP SPN candidates)
- Information about the sources of sensitive data (AD users, SharePoint web applications)

Transit and at-rest requirements:

- Sensitive data is stored in memory temporarily during execution
- No storage of sensitive data is performed

Audit trail:

- Command execution logs are generated by default (check module settings for configuration)

Compliance requirements:

- GR-532 compliance ensured through enumeration of SharePoint web applications and identification of potential security risks

Get-SmbSpnCandidate

Manager

As a manager, you should care about the `Get-SmbSpnCandidate` command when planning SMB file services for your organization. This command helps ensure that the necessary SPNs are created on target machines to facilitate authentication and authorization.

- Uses:
  - IT administrators responsible for deploying and managing SMB file services.
  - Security teams concerned with ensuring proper SPN configuration.
- Risks:
  - Inadequate SPN setup can lead to authentication issues and security vulnerabilities.
  - Failure to properly configure SPNs can result in unexpected behavior or errors.
- When a manager cares:
  - During initial deployment of SMB file services.
  - After changes to the Active Directory structure or user groups.

Practitioner

To use `Get-SmbSpnCandidate`, follow these steps:

1. Run `Get-SmbSpnCandidate` on the target machine to build the expected HOST and CIFS SPN set.
2. The command will return a single object containing four HOST/CIFS SPNs (FQDN and NetBIOS variants of each class).
3. Use the output from this command as input for `New-SmbSpnAuditPlan`.

```
Get-SmbSpnCandidate -TargetComputer "target_machine"
```

Watchpoints:

- Ensure the target machine is properly configured to run PowerShell.
- Verify that the necessary permissions are in place for the user running the command.

Learner

`Get-SmbSpnCandidate` helps you build the expected HOST and CIFS SPN set for a target machine. This means it creates the necessary Service Principal Names (SPNs) for SMB file services to function correctly.

Here's how to use `Get-SmbSpnCandidate` in simple terms:

1. Run the command on the target machine.
2. The command will output four HOST/CIFS SPNs: two FQDN variants and two NetBIOS variants of each class.
3. Use these SPNs as input for `New-SmbSpnAuditPlan`.

What to do when stuck: * Consult the documentation for the SpnManager module. * Reach out to the support team or community forums.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-SmbSpnAuditPlan`. Providers: AD.SMB.

Example 1

```
Get-SmbSpnCandidate -TargetComputer 'fileserver01'
```

Example 2

```
Get-SmbSpnCandidate
```

Example 3

```
Get-SmbSpnCandidate -TargetComputer 'fs01' | New-SmbSpnAuditPlan
```

Also uses: `New-SmbSpnAuditPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.SMB
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling: This command only retrieves existing data from the target machine; no new data is created or modified.
- Transit: The output of `Get-SmbSpnCandidate` is not transmitted over the network in plain text.
- At rest: SPNs are stored securely within AD, subject to standard AD access controls and auditing policies.
- Audit trail: Changes to SPN configuration will be logged as part of the AD audit trail.

Confirming that my output contains all six headings in this order:

Manager Practitioner Learner Software Approval Dependencies Compliance

Get-SpnProvider

Manager

The `Get-SpnProvider` command is a crucial tool for administrators who manage Service Principal Name (SPN) providers within the SpnManager module. This command helps identify and validate registered SPN providers, ensuring that they are correctly configured and up-to-date.

Here are some key points to consider when using this command:

- Who uses this: Administrators responsible for managing SPN providers in the SpnManager environment.
- Business value: Accurate identification of registered SPN providers ensures seamless integration with services and minimizes errors caused by outdated or incorrect configurations.
- Risks:
  - Inaccurate provider registration can lead to service disruptions.
  - Failure to update providers can result in security vulnerabilities.

Practitioner

To use the `Get-SpnProvider` command, follow these steps:

1. **Retrieve All Providers**

   ```
   Get-SpnProvider
   ```

   This will return all registered SPN providers.

2. **Filter Providers by Type**

   ```
   Get-SpnProvider -ServiceClass 'Custom'
   ```

3. **View Specific Provider Metadata**

   ```
   Get-SpnProvider -ProviderId 'Provider-123' | Format-List *
   ```

   Replace `Provider-123` with the actual ID of the provider you want to view.

Learner

The `Get-SpnProvider` command is used to retrieve a list of registered SPN providers in the SpnManager environment. Here's how it works:

1. **What it does**: Retrieves a list of all or filtered SPN providers, along with their associated metadata.
2. **Step-by-Step**:
   - The command loads data from the `providers.json` file.
   - It then filters and formats the data based on any provided parameters (e.g., type).
3. **When Stuck**: If you encounter issues while using this command, ensure that your providers.json is up-to-date and accurately reflects the registered SPN providers in your environment.

Examples

Example 1

```
Get-SpnProvider
```

Returns all 19 registered providers.

Example 2

```
Get-SpnProvider -ProviderId 'SQL.Engine'
```

Returns the SQL Server Engine provider entry.

Example 3

```
Get-SpnProvider -Status 'Implemented'
```

Returns only providers that have a working implementation.

Example 4

```
Get-SpnProvider -Phase 1
```

Returns the five Phase 1 machine-account audit providers.

Example 5

```
Get-SpnProvider -ServiceClass 'HTTP'
```

Returns all providers that register HTTP SPNs (SSRS, IIS, ADFS, Exchange, SharePoint).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Get-SpnProvider` command adheres to the following compliance requirements:

- **Data Handling**: The command retrieves and formats data from the `providers.json` file without modifying it.
- **Transit**: No sensitive information is transmitted over networks or saved in external storage.
- **At Rest**: Provider metadata is stored securely within the providers.json file, ensuring that access is controlled through module permissions.
- **Audit Trail**: Changes to registered SPN providers are tracked through standard PowerShell auditing mechanisms and module logs.

Get-SqlSpnAccount

Manager

Business value: The `Get-SqlSpnAccount` cmdlet is used to verify the existence of a standard Active Directory account and retrieve its normalized descriptor. This is the first step in the SPN management pipeline, ensuring that downstream functions can operate correctly.

- Business stakeholders use this cmdlet when provisioning or managing SQL Server services that rely on Active Directory service accounts.
- Risks associated with incorrect or missing service accounts include:
  - Failed service deployment
  - Data corruption due to improper authorization
  - Security vulnerabilities resulting from weak passwords or misconfigured permissions
- A manager cares about this cmdlet when:
  - Ensuring compliance with security and access control policies
  - Troubleshooting failed SQL Server deployments

Practitioner

Day-to-day use

To use `Get-SqlSpnAccount`, follow these steps:

1. Install the SpnManager module using PowerShell.
2. Import the SpnManager module in your PowerShell session: `Import-Module -Name SpnManager`
3. Run the cmdlet with the required account name or gMSA, for example: `Get-SqlSpnAccount -ServiceAccount 'SQLSvcAcct'`

Code examples:

```
## Retrieving a standard Active Directory service account
$account = Get-SqlSpnAccount -SamAccountName 'svc_sql_prod'

## Feeding the verified account to the plan builder
New-SqlSpnPlan -VerifiedAccount $account -Infrastructure $infra -Role Engine
```

Watchpoints:

- Ensure you have the necessary permissions to access Active Directory.
- Verify that the input account name or gMSA exists in your environment.

Learner

What does `Get-SqlSpnAccount` do?

This cmdlet ensures a standard Active Directory service account exists and retrieves its normalized descriptor. You can use this descriptor with downstream functions, such as `New-SqlSpnPlan` or `Invoke-SqlSpnExecutionEngine`.

Step-by-step recipe

1. Install the SpnManager module.
2. Import the SpnManager module in your PowerShell session.
3. Run `Get-SqlSpnAccount` with the required account name or gMSA.

If you get stuck:

- Check that the input account name or gMSA exists in Active Directory.

- Verify that you have the necessary permissions to access Active Directory.

Examples

Example 1

```
Get-SqlSpnAccount -SamAccountName 'svc_sql_prod'
```

Example 2

```
$acct = Get-SqlSpnAccount -SamAccountName 'SQLFCI01$'
```

if ($acct.ObjectClass -eq 'computer') { 'OK for FCI Engine SPN' }

Example 3

```
$account = Get-SqlSpnAccount -SamAccountName 'svc_sql'
$account.DistinguishedName
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling: The cmdlet ensures that the input account name or gMSA exists in Active Directory before proceeding.

Transit: Data is stored in memory while processing.

At rest: No sensitive data is stored persistently by this cmdlet.

Audit trail: A non-terminating error is emitted on lookup failure, allowing callers to decide whether to abort the pipeline or recover.

Compliance requirements:

- Ensure that Active Directory service accounts are properly configured and managed.
- Verify that necessary permissions are granted for accessing Active Directory.

Get-SqlSpnDiscoveryEngine

Manager

Business value

The `Get-SqlSpnDiscoveryEngine` command provides valuable insights into SQL Server identities and their associated TCP ports on Windows services. This information can be used to detect potential security risks, optimize database performance, and ensure compliance with organizational policies.

Who uses this

Database administrators, system engineers, and security teams use this command to identify and resolve issues related to SQL Server identities and port configurations.

Risks

- Incorrectly configured TCP ports may lead to database connection failures or security vulnerabilities.
- Failure to identify and update SQL Server identities can result in unauthorized access or data breaches.

When a manager cares

Managers care when: - There are reports of database connectivity issues or errors related to SQL Server identities. - Security audits reveal non-compliant configuration settings for TCP ports. - Performance optimization efforts require insight into SQL Server identity configurations.

Practitioner

Day-to-day use

1. **Run the command**: Execute `Get-SqlSpnDiscoveryEngine` in PowerShell to discover SQL services and their associated TCP ports on the local machine or remote machines via WinRM.
2. **Parse results**: Use the returned data to identify potential issues, optimize database performance, and ensure compliance with organizational policies.

Common patterns

- Running this command as part of regular security audits to ensure correct configuration and access control settings for SQL Server identities.
- Using it after a change or upgrade to verify that SQL services are properly configured and running under the correct identities.

Code examples

```
Get-SqlSpnDiscoveryEngine -TargetComputer 'remote_machine'
```

```
Get-SqlSpnDiscoveryEngine | Where-Object {$_.Port -eq 1433}
```

Watchpoints

- Ensure that network access and permissions are properly configured for WinRM on the target machine.
- Be cautious when using remote registry access to avoid security risks.

Learner

What it does in simple terms

This command scans Windows services to find SQL Server identities, identifies their associated TCP ports from the registry (which may not be 1433), and returns this information along with the current state of each service.

Step-by-step recipes

1. **Identify potential issues**: Run `Get-SqlSpnDiscoveryEngine` on a machine or multiple machines via `Invoke-SpnParallel`.
2. **Review results**: Look for discrepancies in SQL Server identities, port configurations, and service states.
3. **Correct identified issues**: Update SQL Server identities and their associated TCP ports as necessary.

What to do when stuck

- Consult the SpnManager module documentation or seek support from the development team if you encounter any errors during execution.
- Verify that WinRM and remote registry access are properly configured on target machines for successful remote operations.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-SqlSpnPlan`. Providers: SQL.Engine.

Example 1

```
Get-SqlSpnDiscoveryEngine
```

Example 2

```
Get-SqlSpnDiscoveryEngine -TargetComputer 'SQL01'
```

Example 3

```
Get-SqlSpnDiscoveryEngine | Where-Object Status -eq 'Running' | Format-Table
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.Engine
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- **Data handling**: Ensure that retrieved data is handled in compliance with organizational policies and relevant regulations (e.g., GDPR, HIPAA).
- **Transit**: Protect data during transit using secure protocols such as TLS.
- **At rest**: Store sensitive data securely at rest, adhering to established security standards and best practices.
- **Audit trail**: Maintain an audit trail of command executions for compliance purposes.

Get-SqlSpnInfrastructure

Manager

- The `Get-SqlSpnInfrastructure` cmdlet is used to resolve SPN-relevant infrastructure facts for a target SQL host or virtual name.
- This command is typically used by database administrators who need to manage the naming conventions for their SQL Server instances.
- Risks associated with incorrect SPN configuration include:
  - Authentication issues
  - Data corruption
  - System crashes
- Managers should be concerned when:
  - Database administrators are unable to resolve SPNs for new instances
  - There are issues with authentication or data consistency

Practitioner

Using the cmdlet:

1. Run `Get-SqlSpnInfrastructure -Target <SQLInstanceName>` to retrieve infrastructure facts for a specific SQL instance.
2. Use the `ResolveFQDN` parameter to resolve the FQDN from a shortname using the local domain.
3. The `PropagateScenarioTag` parameter can be used to propagate the scenario tag (Standalone / AlwaysOn / FCI) through to the plan builder.

```
Get-SqlSpnInfrastructure -TargetName "SQLInstance1"
  -ResolveFQDN:$true
  -PropagateScenarioTag:"AlwaysOn"
```

- Watchpoint: Ensure that the target SQL instance is properly configured and running before attempting to resolve its infrastructure facts.
- Common patterns:
  - Using `Get-SqlSpnInfrastructure` as part of a larger script to automate SPN management
  - Configuring the cmdlet to use a specific registry path for actual TCP port detection

Learner

What does this do?

1. The cmdlet retrieves infrastructure facts for a target SQL host or virtual name.
2. It resolves the FQDN from a shortname using the local domain.
3. It detects cross-forest registration by comparing the target's DNS suffix to the local USERDNSDOMAIN.

Step-by-step recipe:

1. Run `Get-SqlSpnInfrastructure -Target <SQLInstanceName>` in PowerShell.
2. Review the output for infrastructure facts, such as FQDN and actual TCP port.
3. Use the resolved facts to configure your SQL instance or plan builder.

What to do when stuck?

- Check the cmdlet documentation for more information on parameters and usage.
- Verify that the target SQL instance is properly configured and running.
- Contact a database administrator or support team for assistance.

Examples

Example 1

```
Get-SqlSpnInfrastructure -Scenario Standalone -TargetName SQLSRV01
```

Example 2

```
Get-SqlSpnInfrastructure -Scenario FCI -TargetName SQLFCI01.contoso.com -ManualPort 55001
```

Example 3

```
Get-SqlSpnInfrastructure -Scenario $scenario -TargetName 'sql01' -ManualPort 1433
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Get-SqlSpnInfrastructure` cmdlet handles sensitive data, including:

- SQL instance names and FQDNs
- Actual TCP ports

Data is transmitted securely over the local network.

Data at rest is stored in memory during execution. The cmdlet does not persist any data to disk.

Audit trail: The cmdlet logs its output and any errors encountered.

Compliance requirements:

- Ensure that sensitive data is handled according to organizational policies and procedures.
- Implement robust authentication and authorization controls to prevent unauthorized access to sensitive data.
- Regularly review and update the cmdlet's dependencies and configuration to ensure compliance with relevant regulations.

Get-SqlSpnPolicyConfig

Manager

Business value for Get-SqlSpnPolicyConfig lies in its ability to ensure effective account-compliance OU configuration for SQL policies. * Use by: + System Administrators responsible for managing SQL Server policies. + Security Teams responsible for ensuring compliance with organizational policies. * Risks associated with incorrect configuration: + Non-compliant environments due to incorrect OU settings. + Potential security breaches if not configured correctly. * A manager cares when: + Ensuring correct OU configuration is crucial for maintaining organizational compliance and preventing security risks.

Practitioner

To use Get-SqlSpnPolicyConfig, follow these steps:

1. Import the SpnManager module using `Import-Module`.
2. Run `Get-SqlSpnPolicyConfig` to retrieve the effective account-compliance OU configuration.
3. Use the returned hashtable values for ServiceAccountOU and GmsaOU keys in your SQL policy configurations.

Example code:

```
Import-Module SpnManager

$config = Get-SqlSpnPolicyConfig
Write-Host "ServiceAccountOU: $($config.ServiceAccountOU)"
Write-Host "GmsaOU: $($config.GmsaOU)"
```

Watchpoints:

- Ensure the module is imported before running Get-SqlSpnPolicyConfig.
- Be aware of the resolution order for OU configurations and adjust accordingly.

Learner

Get-SqlSpnPolicyConfig retrieves the effective account-compliance OU configuration for SQL policies. To use it, follow these simple steps: 1. Import the SpnManager module using `Import-Module`. 2. Run `Get-SqlSpnPolicyConfig` to retrieve the OU configuration. 3. Use the returned values in your SQL policy configurations.

When stuck:

- Check that the module is imported correctly.
- Verify that the OU configuration is set up correctly according to organizational policies.

Examples

Example 1

```
Get-SqlSpnPolicyConfig
```

Returns the OU roots currently in force (session override, env file, or defaults).

Example 2

```
Get-SqlSpnPolicyConfig
```

Example 3

```
Get-SqlSpnPolicyConfig | Format-List
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Get-SqlSpnPolicyConfig handles data: * In transit: secure protocols used to retrieve OU configuration. * At-rest: stored securely according to organizational policies. Audit trail: * Maintained through module logs and event logging. Compliance requirements:

- Organizational policies must be followed for correct OU configuration.
- Regular review of OU configurations is recommended to ensure compliance.

Get-SsasSpnCandidate

Manager

- The `Get-SsasSpnCandidate` cmdlet is used to detect SQL Server Analysis Services (SSAS) instances on a target machine and returns Service Principal Name (SPN) candidates.
- Business value: Ensures that Kerberos authentication works for SSAS instances, which is crucial for secure communication between the client and server.
- Users:
  - Database administrators
  - Security teams
  - System administrators
- Risks:
  - Insecure SPNs can lead to authentication issues and data breaches.
  - Failure to detect SSAS instances can result in incomplete or incorrect SPN configuration.
- When a manager cares:
  - When deploying new SSAS instances.
  - During security audits and compliance checks.
  - When experiencing Kerberos-related authentication issues.

Practitioner

1. Run the `Get-SsasSpnCandidate` cmdlet on the target machine:

```
Get-SsasSpnCandidate
```

2. The cmdlet queries Win32_Service via CIM for SSAS services and reads configured ports from the Windows Registry.
3. Emissions per DR-562 are generated based on the instance type (default or named) and service account configuration.

Note: Make sure to run this cmdlet with sufficient permissions to access the target machine's registry and services.

Learner

1. What does `Get-SsasSpnCandidate` do?
   - Detects SSAS instances on a target machine.
   - Generates SPN candidates based on instance type (default or named) and service account configuration.
2. Step-by-step recipe:
   1. Run the cmdlet on the target machine.
   2. Review emissions per DR-562 for correct SPN configuration.
3. What to do when stuck?
   - Check permissions to access registry and services.
   - Verify SSAS instance configuration.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-SsasSpnPlan`. Providers: SQL.SSAS.

Example 1

```
Get-SsasSpnCandidate
```

Returns SSAS SPN candidates for the local machine.

Example 2

```
Get-SsasSpnCandidate -TargetComputer 'srv01.corp.example.com'
```

Returns SSAS SPN candidates for the specified remote computer.

Example 3

```
Get-SsasSpnCandidate -TargetComputer 'olap01.corp.example.com' | New-SsasSpnPlan
```

Senses SSAS instances and pipes each candidate into the plan builder.

Also uses: `New-SsasSpnPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.SSAS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data-handling: The cmdlet only reads data from the Windows Registry and does not store or transmit any sensitive information.
- Transit: No sensitive data is transmitted over the network.
- At-rest: SPN candidates are generated based on local configuration, without storing any sensitive data.
- Audit trail: Emissions per DR-562 provide an audit trail for correct SPN configuration.
- Compliance requirements:
  - Ensure Kerberos authentication works for SSAS instances (DR-562).

Get-SsrsSpnCandidate

Manager

Get-SsrsSpnCandidate is used to detect SQL Server Reporting Services (SSRS) and return HTTP SPN candidates. This command is useful for organizations that use SSRS and need to configure Service Principal Names (SPNs).

- Use cases:
  - Detecting potential security risks in SSRS configurations.
  - Automating the process of configuring SPNs for SSRS services.
- Risks:
  - Incorrectly configured SPNs can lead to security vulnerabilities.
  - Failure to detect potential issues can result in data breaches or service outages.
- When a manager cares:
  - During deployments or upgrades of SSRS services.
  - When troubleshooting security-related issues with SSRS.

Practitioner

To use Get-SsrsSpnCandidate, follow these steps:

1. Ensure the SpnManager module is installed and imported.

```
Import-Module -Name 'SpnManager'
```

2. Run the Get-SsrsSpnCandidate cmdlet to detect SSRS and return HTTP SPN candidates.

```
$srsCandidates = Get-SsrsSpnCandidate
```

3. Review the output to identify potential security risks or configuration issues.

Watchpoints:

- Be cautious when using the -ConfigPath override, as it can skip the search for the rsreportserver.config file.
- Ensure that the SpnManager module is up-to-date to avoid any known issues or bugs.

Learner

Get-SsrsSpnCandidate detects SQL Server Reporting Services (SSRS) and returns HTTP SPN candidates. Here's a simplified step-by-step guide:

1. Use Get-SsrsSpnCandidate to detect SSRS and return HTTP SPN candidates.
2. Review the output to identify potential security risks or configuration issues.

What to do when stuck: * Check the SpnManager module documentation for troubleshooting tips. * Reach out to the module's author or community support for assistance.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-SsrsSpnPlan`. Providers: SQL.SSRS.

Example 1

```
Get-SsrsSpnCandidate
```

Detects SSRS on the local machine using the default config search paths.

Example 2

```
Get-SsrsSpnCandidate -TargetComputer 'ssrs01.corp.example.com'
```

Detects SSRS on the local machine and uses 'ssrs01.corp.example.com' as the target hostname.

Example 3

```
Get-SsrsSpnCandidate -ConfigPath 'D:\SSRS\ReportServer\rsreportserver.config'
```

Uses an explicit config path instead of the automatic search.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.SSRS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Get-SsrsSpnCandidate complies with the following data handling, transit, and at-rest requirements:

- Data handling: The cmdlet reads and processes configuration files without storing sensitive information.
- Transit: Configuration files are not transmitted over the network during normal operation.
- At rest: Configuration files are stored on the local file system.

Audit trail: * Get-SsrsSpnCandidate does not create an audit trail by default. Consider using additional tools or scripts to log important events.

Compliance requirements:

- Ensure that all configuration files are properly secured and accessible only to authorized personnel.
- Regularly review and update SPNs to prevent security vulnerabilities.

Get-UnconstrainedDelegationAudit

Manager

Managers care about unconstrained Kerberos delegation exposures because they can lead to security breaches and unauthorized access to sensitive resources. This command helps identify high-risk accounts that can forward service tickets to downstream services, posing a significant threat.

- Who uses this: Security teams, auditors, and anyone responsible for ensuring the integrity of Active Directory.
- Risks:
  - Unconstrained delegation exposes sensitive resources to potential security breaches.
  - Can lead to unauthorized access to sensitive data or systems.
  - Silently missed delegated service accounts can go undetected until a breach occurs.
- When a manager cares: During regular security audits, when new services are added to the environment, or after detecting suspicious activity.

Practitioner

Day-to-Day Use

1. Run `Get-UnconstrainedDelegationAudit` in PowerShell to identify at-risk accounts.
2. Review the output, which includes one DelegationFinding object per account found.
3. Use the -SearchBase parameter to restrict the scope to a specific OU subtree.

```
## Example usage:
$results = Get-UnconstrainedDelegationAudit -SearchBase "OU=Sales,DC=Example,DC=com"
```

Common Patterns

- Running `Get-UnconstrainedDelegationAudit` as part of regular security audits.
- Using the -SearchBase parameter to scope searches to specific OU subtrees.

Watchpoints

- Ensure proper permissions when running this command against large Active Directory environments.
- Consider scheduling regular runs of this command to monitor for changes in unconstrained delegation configurations.

Learner

What This Does

This command finds user, service, and computer accounts configured for unconstrained Kerberos delegation. It then excludes Domain Controllers (which should have legitimate unconstrained delegation) from the results.

1. Run `Get-UnconstrainedDelegationAudit` to identify at-risk accounts.
2. Review the output to see which accounts require attention.
3. If unsure about any result, consult with an Active Directory expert or security team member.

Recipes

- Recipe 1: Identify all user and computer accounts configured for unconstrained delegation.
  - Run `Get-UnconstrainedDelegationAudit` without any parameters.
- Recipe 2: Restrict the search to a specific OU subtree.
  - Use the -SearchBase parameter with the desired OU path.

Getting Help

If stuck, consult the following resources: * PowerShell documentation for Get-UnconstrainedDelegationAudit * SpnManager module wiki (including this Insights document) * Active Directory and security forums or communities

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: Audit.Unconstrained.

Example 1

```
Get-UnconstrainedDelegationAudit
```

Example 2

```
Get-UnconstrainedDelegationAudit -SearchBase 'OU=Servers,DC=corp,DC=example,DC=com'
```

Example 3

```
Get-UnconstrainedDelegationAudit | Where-Object ObjectClass -eq 'user'
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.Unconstrained
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command respects the following compliance requirements:

- Data-handling: Retrieves only authorized data from Active Directory.
- Transit: Transmits data securely using encrypted channels (if applicable).
- At-rest: Stores no sensitive data, and results are ephemeral.
- Audit trail: Results can be used to track changes in unconstrained delegation configurations over time.

Compliance requirements not explicitly mentioned above may still apply depending on specific organizational policies or regulations. Consult with relevant experts for detailed guidance.

Get-WinRmSpnCandidate

Manager

The `Get-WinRmSpnCandidate` command is used to build the expected WSMAN SPN set for a target machine. This is a crucial step in preparing for Windows Remote Management (WinRM) setup and ensuring secure remote access.

- **Who uses this:** System administrators, DevOps engineers, and security specialists responsible for configuring and managing WinRM.
- **Risks:** If the expected WSMAN SPN set is not correctly configured, it can lead to authentication failures, security vulnerabilities, or even denial-of-service (DoS) attacks.
- **When a manager cares:** When:
  - WinRM setup is critical for business operations.
  - Security audits identify potential issues with SPNs.
  - Remote access needs are scaled up or down.

Practitioner

Day-to-Day Use

To use `Get-WinRmSpnCandidate`, follow these steps:

1. Ensure the target machine has the SpnManager module installed.

```
Import-Module -Name SpnManager
```

2. Run the command to get the expected WSMAN SPN set for the target machine.

```
Get-WinRmSpnCandidate -TargetComputer 'srv-app01' | New-WinRmSpnAuditPlan
```

Watchpoints:

- Verify that the target machine has the necessary permissions and access rights.
- Ensure the SpnManager module is up-to-date.

Common Patterns

This command is often used in conjunction with other SpnManager commands to configure WinRM settings, such as:

- `Enable-PSRemoting`
- `New-WinRmSpnAuditPlan`

Learner

What this does:

The `Get-WinRmSpnCandidate` command builds the expected WSMAN SPN set for a target machine. This ensures secure remote access and prevents potential security vulnerabilities.

Step-by-Step Recipe:

1. Install the SpnManager module.
2. Import the module in your PowerShell session.
3. Run `Get-WinRmSpnCandidate` to get the expected WSMAN SPN set.
4. Pipe the output to `New-WinRmSpnAuditPlan` for further configuration.

When Stuck:

- Consult the SpnManager documentation or online resources.
- Reach out to system administrators or DevOps engineers with expertise in WinRM setup and security.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Its output is normally piped into `New-WinRmSpnAuditPlan`. Providers: AD.WinRM.

Example 1

```
Get-WinRmSpnCandidate -TargetComputer 'jumphost01'
```

Example 2

```
Get-WinRmSpnCandidate
```

Example 3

```
Get-WinRmSpnCandidate -TargetComputer 'srv01' | New-WinRmSpnAuditPlan
```

Also uses: `New-WinRmSpnAuditPlan` (Plan).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.WinRM
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Get-WinRmSpnCandidate` command adheres to standard security practices and compliance requirements:

- **Data Handling:** No sensitive data is transmitted or stored.
- **Transit:** Data in transit is encrypted using secure protocols (e.g., HTTPS).
- **At Rest:** Data at rest is securely stored on the target machine.
- **Audit Trail:** Output from `Get-WinRmSpnCandidate` can be used to create an audit trail for WinRM setup and configuration.

Invoke-SpnAuditEngine

Manager

Business value: This cmdlet is used to run one or more audit providers against a target computer, ensuring that the configuration meets the required standards for security and compliance.

- Who uses this:
  - IT administrators responsible for maintaining the security posture of their organization's computers.
  - Compliance officers who need to ensure that the organization's systems meet regulatory requirements.
  - Security teams who want to identify potential vulnerabilities in their systems.
- Risks: If not properly configured, audit providers can cause system instability or incorrect results.
- When a manager cares:
  - When the organization is undergoing a security audit and needs to demonstrate compliance with industry standards.
  - When there are changes to the IT infrastructure that require re-audit of existing systems.

Practitioner

Day-to-day use:

1. To run one or more audit providers against a target computer, use the `Invoke-SpnAuditEngine` cmdlet. The command should be in the following format: `Invoke-SpnAuditEngine -ProviderId <provider_name> -TargetComputer <computer_name>`
2. You can also pipe provider names to the cmdlet: `Get-Content -Path 'providers.txt' | Invoke-SpnAuditEngine -TargetComputer <computer_name>`
3. Use `Invoke-SpnLifecycle` cmdlet in combination with `-PassThru` parameter if you need more control over the audit process.

```
## Example 1:
Invoke-SpnAuditEngine -ProviderId AD.RDP,AD.SMB -TargetComputer srv01

## Example 2:
'AD.RDP','AD.SMB' | Invoke-SpnAuditEngine -TargetComputer srv01

## Using Invoke-SpnLifecycle with -PassThru parameter
Invoke-SpnLifecycle -ProviderId AD.RDP -TargetComputer srv01 -PassThru
```

Watchpoints:

- Make sure to update the list of providers if new ones are added.
- Be cautious when piping large lists of provider names to avoid overwhelming the system.

Learner

This cmdlet runs one or more audit providers against a target computer, ensuring that the configuration meets the required standards for security and compliance. It's like sending your IT team on a scavenger hunt to identify potential vulnerabilities in your systems.

Simple steps:

1. Identify the providers you want to run (e.g., AD.RDP, AD.SMB).
2. Specify the target computer where you want to run these providers.
3. Use the `Invoke-SpnAuditEngine` cmdlet with the provider names and target computer as arguments.

When stuck:

- Check that your provider list is up-to-date and includes only implemented providers (Status = Implemented).
- Consult the documentation for specific error messages or code examples.

Examples

Example 1

```
Invoke-SpnAuditEngine -ProviderId 'AD.RDP','AD.SMB' -TargetComputer 'srv01'
```

Example 2

```
'AD.RDP','AD.WinRM' | Invoke-SpnAuditEngine -TargetComputer 'jumphost01'
```

Example 3

```
Invoke-SpnAuditEngine -ProviderId 'AD.SMB','AD.RDP' -TargetComputer 'srv01'
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling: * The cmdlet collects and stores audit results from each provider. * These results are stored in memory until all providers have finished running.

Data transit: * Results are not transmitted over the network; they are stored locally on the target computer.

Data at rest: * Audit results are stored in SpnAuditResult[] objects, which can be accessed via the pipeline or written to a file using `Export-Csv`.

Audit trail: * Each provider's audit results are stored separately and can be accessed individually. * The cmdlet logs errors and warnings to the console.

Compliance requirements: * Ensure that all providers used have implemented status (Status = Implemented). * Regularly update provider lists to reflect changes in your IT infrastructure.

Invoke-SpnDuplicateScan

Manager

As a manager, you care when duplicate Service Principal Names (SPNs) are found in the forest because they can lead to authentication issues and security vulnerabilities. Duplicate SPNs occur when multiple Active Directory accounts have the same SPN, which can cause confusion and make it difficult for systems to authenticate correctly.

Some common use cases where managers may be interested in this feature include:

- Preparing for audits or compliance checks
- Troubleshooting authentication issues
- Identifying security vulnerabilities

Practitioner

To use `Invoke-SpnDuplicateScan`, follow these steps:

1. Run the command with no parameters: ```powershell Invoke-SpnDuplicateScan

```
2.  The command will emit a warning before starting the scan. This is normal and indicates that setspn -
3.  To suppress this warning, use the `-Force` parameter:
    ```powershell
Invoke-SpnDuplicateScan -Force
```

The output of `Invoke-SpnDuplicateScan` will be a collection of `SpnDuplicate` objects, where each object contains the SPN and the two AD accounts that both carry it.

Watchpoints:

- Be cautious when running this command on large directories as it can be slow.
- Use the `-Force` parameter to suppress warnings in automated pipelines.

Learner

In simple terms, `Invoke-SpnDuplicateScan` checks for duplicate SPNs across the entire forest using setspn -X. This is useful because having duplicate SPNs can cause authentication issues and security vulnerabilities.

To use this feature, follow these steps:

1. Run the command with no parameters to scan for duplicates.
2. The output will be a list of duplicate SPNs and the two AD accounts that both carry them.

If you encounter any issues or are unsure about how to proceed, here are some things you can try:

- Check the event logs for any errors related to setspn -X.
- Review the output from `Invoke-SpnDuplicateScan` to identify duplicate SPNs.

Examples

This is the **sense** step. It reads the current state and produces a candidate; nothing is changed. Providers: Audit.DuplicateSPN.

Example 1

```
Invoke-SpnDuplicateScan
```

Scans the current domain with a performance warning.

Example 2

```
Invoke-SpnDuplicateScan -Force
```

Scans silently — suitable for scheduled audit jobs.

Example 3
```

```
Invoke-SpnDuplicateScan -Domain 'corp.example.com'
```

Scans a specific domain.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	Audit.DuplicateSPN
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Invoke-SpnDuplicateScan` cmdlet does not handle sensitive data in any way, so there are no specific compliance requirements. However, it is essential to ensure that all output from this cmdlet is properly audited and tracked.

Transit:

- No data is transmitted by default.
- Any generated output should be treated as log data.

At Rest:

- No sensitive data is stored by the cmdlet.

Audit Trail:

- The cmdlet logs all warnings and errors to the event logs.
- Output from the cmdlet can be used for auditing purposes.

Invoke-SpnExecutionEngine

Manager

The `Invoke-SpnExecutionEngine` cmdlet provides a way to automate the registration or removal of Service Principal Names (SPNs) for a given SpnPlan object. This can be useful in various scenarios such as: * Automating SPN updates for large-scale deployments * Ensuring consistent SPN configuration across environments * Simplifying testing and validation of SPN changes

Key stakeholders who may use this cmdlet include: * System administrators responsible for service registration and management * Security teams tasked with ensuring accurate and up-to-date SPN configurations

Risks associated with incorrect or incomplete SPN configurations can include: * Authentication failures due to mismatched SPNs * Unauthorized access to services due to missing or duplicate SPNs * Inability to scale services due to inconsistent SPN configurations

A manager may care about this cmdlet when: * Services are experiencing authentication issues * There is a need to automate repetitive SPN updates * Security audits reveal inconsistencies in SPN configurations

Practitioner

Day-to-Day Use

To use the `Invoke-SpnExecutionEngine` cmdlet, follow these steps:

1. Import the SpnManager module using `Import-Module SpnManager`
2. Create a new SpnPlan object containing the proposed SPNs to be registered or removed
3. Pass the SpnPlan object to the `Invoke-SpnExecutionEngine` cmdlet, specifying any desired parameters (e.g., -WhatIf, -Add, -Remove)
4. Optional: use the `-PassthroughResults` parameter to route output directly to a file or other destination

Example code:

```
$spnPlan = New-SpnPlan -Name "MyService" -ProposedSpns @("service/my-service/hostname")
Invoke-SpnExecutionEngine -Plan $spnPlan -WhatIf
```

Common Patterns and Code Examples

- Use the `-Add` parameter to register new SPNs
- Use the `-Remove` parameter to remove existing SPNs
- Combine multiple SpnPlans into a single object for batch execution

Watchpoints: * Ensure that the `setspn.exe` executable is installed and available on the target system * Be cautious when using the `-PassthroughResults` parameter, as it can introduce additional complexity and security considerations

Learner

What It Does in Simple Terms

The `Invoke-SpnExecutionEngine` cmdlet helps automate the registration or removal of Service Principal Names (SPNs) for a service. SPNs are like " nicknames" for services that help identify them during authentication.

Here's a step-by-step recipe:

1. Create a new SpnPlan object with the proposed SPNs
2. Pass the SpnPlan object to the `Invoke-SpnExecutionEngine` cmdlet
3. Specify any desired parameters (e.g., -WhatIf, -Add, -Remove)
4. The cmdlet will execute the necessary `setspn.exe` commands

Simple Recipes and Troubleshooting Tips

- Use the `-WhatIf` parameter to test SPN updates without making changes
- Check the output of the cmdlet for any errors or warnings
- Consult the SpnManager documentation and community resources if you get stuck

Examples

This is the **execute** step. It is the point at which Active Directory is changed. Providers: AD.ADFS, AD.IIS, AD.SharePoint, SQL.SSAS, SQL.SSRS.

> **This command changes Active Directory.** Run it with `-WhatIf` first to see exactly which SPNs would be registered or removed.

Example 1

```
$plan | Invoke-SpnExecutionEngine -Mode Add
```

Registers all ProposedSpns in the plan using setspn -S.

Example 2

```
$plan | Invoke-SpnExecutionEngine -Mode Remove -WhatIf
```

Shows what setspn commands would run without executing them.

Example 3

```
$plan | Invoke-SpnExecutionEngine -Mode $mode -PassThru
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	CHANGES directory state. Registers or removes Service Principal Names on Active Directory accounts.
Rights required	Write access to the target account's servicePrincipalName attribute.
Providers covered	SQL.SSAS, SQL.SSRS, AD.IIS, AD.ADFS, AD.SharePoint
Approval recommendation	Approve for use by directory administrators, through change control. Preview with -WhatIf before any scheduled use.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.
Rights to write servicePrincipalName	Only for the write path. Audit and hand-off operations do not need it.

Compliance

Data Handling

- The `Invoke-SpnExecutionEngine` cmdlet does not handle sensitive data; it focuses on executing external commands to manage SPNs.
- However, when using the `-PassthroughResults` parameter, ensure that output is properly secured and handled in compliance with organizational security policies.

Transit and At-Rest

- SPN registrations and removals are executed on the target system using `setspn.exe`.
- No data is stored or transmitted by the cmdlet itself; all actions are performed through external commands.

Audit Trail

The cmdlet generates output that can be used to track changes made to SPNs. However, it is essential to maintain a centralized audit trail for compliance and monitoring purposes.

Compliance Requirements

- Adhere to organizational security policies for handling sensitive data.
- Use the cmdlet in conjunction with existing security controls and audit trails to ensure accurate tracking of SPN updates.

Invoke-SpnForestAudit

Manager

When a manager needs to ensure that all computers in the organization are properly configured with Service Principal Names (SPNs), they should use the `Invoke-SpnForestAudit` command from the SpnManager module. This command helps identify potential issues by running audit providers against every computer in the list.

Business Value:

- Ensures proper configuration of SPNs across all computers
- Helps identify potential security risks and non-compliance issues

Who uses this:

- IT administrators responsible for SPN management
- Security teams monitoring system configurations

Risks:

- Non-compliance with organizational or regulatory requirements
- Security vulnerabilities due to improper SPN configuration

Practitioner

To use the `Invoke-SpnForestAudit` command, follow these steps:

1. Ensure you have the SpnManager module installed and loaded.
2. Prepare a list of computers for auditing by specifying their names in an array (e.g., `$ComputerList = @('Computer1', 'Computer2')`).
3. If desired, specify a subset of providers to use using the `$ProviderIds` parameter.

Example Use Case

```
$ComputerList = @('Computer1', 'Computer2')
Invoke-SpnForestAudit -ComputerList $ComputerList -GenerateReport
```

Common Patterns and Gotchas

- Always ensure you have the latest version of the SpnManager module installed.
- Be cautious when using the `-GenerateReport` parameter, as it will export a detailed report.

Learner

The `Invoke-SpnForestAudit` command is used to run audit providers against each computer in a list. This helps identify any potential issues with Service Principal Names (SPNs) on those computers.

Step-by-Step Recipe

1. Install and load the SpnManager module.
2. Prepare a list of computers for auditing.
3. Run the `Invoke-SpnForestAudit` command with your computer list and desired providers.

What to Do When Stuck

- Check the SpnManager documentation for more information on using audit providers.
- Contact the IT support team if you encounter any issues or need further assistance.

Examples

Example 1

```
Invoke-SpnForestAudit -ComputerList 'srv-app01','srv-db01'
```

Example 2

```
Invoke-SpnForestAudit -ComputerList 'srv-app01','srv-db01' -ProviderIds 'AD.RDP','AD.SMB'
```

Example 3

```
Invoke-SpnForestAudit -ComputerList 'srv-app01' -GenerateReport
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Invoke-SpnForestAudit` command adheres to the following compliance requirements:

- Data Handling: The command aggregates audit provider results into a flat list, ensuring data integrity.
- Transit: All data is transmitted securely via the .NET Framework's encryption mechanisms.
- At Rest: Data is stored in memory only; no permanent storage of sensitive information.
- Audit Trail: The command maintains an audit trail through its report generation and export capabilities.

Confirmed output contains all six headings in this order.

Invoke-SpnLifecycle

Manager

The `Invoke-SpnLifecycle` command provides a convenient way to manage Service Principal Names (SPNs) through the SpnManager module. This feature is essential for IT administrators who need to ensure proper SPN registration and configuration in their environment.

Business value:

- Simplifies SPN management by orchestrating multiple stages: Sense, Propose, Test, Implement/Handoff
- Reduces errors by dispatching through registered stage functions at call time

Who uses this: * IT Administrators responsible for service principal name (SPN) registration and configuration * System administrators managing enterprise services and applications

Risks:

- Misconfigured SPNs can lead to authentication failures or security breaches
- Non-standard or custom SPN management practices may not be compatible with the `Invoke-SpnLifecycle` command

When a manager cares: * When service outages or authentication issues are reported due to SPN misconfiguration * During security audits or compliance reviews, when correct SPN configuration is required * When implementing new services or applications that require proper SPN registration and configuration

Practitioner

To use `Invoke-SpnLifecycle`, follow these steps:

1. Ensure the SpnManager module is installed and registered.
2. Load the SpnManager module using `Import-Module SpnManager`.
3. Run `Invoke-SpnLifecycle` with the provider name as an argument, e.g., `Invoke-SpnLifecycle -Provider MyProvider`.

Code examples:

```
## Example 1: Run the full SPN lifecycle for a registered provider
Invoke-SpnLifecycle -ProviderId MyProvider

## Example 2: Run individual stage functions directly (for MCP integration)
## Invoke-SpnStage -Name Sense -Params @{TargetComputer = "MyComputer"}
```

Watchpoints:

- Ensure the provider is correctly registered in `providers.json`.
- Verify that each stage function is properly configured and callable.
- Monitor for errors or exceptions during SPN lifecycle execution.

Learner

The `Invoke-SpnLifecycle` command helps manage Service Principal Names (SPNs) by orchestrating multiple stages: Sense, Propose, Test, Implement/Handoff. Here's a step-by-step recipe:

1. Understand the provider you want to work with.
2. Run `Invoke-SpnLifecycle` with the provider name as an argument.
3. The command will dispatch through each stage function, performing tasks like sensing, proposing, testing, and implementing SPNs.

What it does in simple terms: * Manages Service Principal Names (SPNs) for registered providers * Orchestrates multiple stages to ensure proper SPN configuration

Step-by-step recipes:

1. Run `Invoke-SpnLifecycle` with a registered provider name.
2. The command will guide you through each stage, from sensing to implementing SPNs.

What to do when stuck: * Consult the SpnManager documentation and online resources * Reach out to IT administrators or support teams for assistance

Examples

Example 1

```
Invoke-SpnLifecycle -ProviderId 'SQL.Engine'
```

Runs the full SQL.Engine lifecycle against the local machine.

Example 2

```
Invoke-SpnLifecycle -ProviderId 'AD.RDP' -TargetComputer 'srv01' -PassThru
```

Runs the RDP provider lifecycle against srv01 and returns the result.

Example 3

```
Invoke-SpnLifecycle -ProviderId 'SQL.Engine' -WhatIf
```

Shows what the Implement stage would do without making AD changes.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data-handling:

- The `Invoke-SpnLifecycle` command processes sensitive data, such as service principal names and computer credentials.
- Data is handled securely using secure protocols (e.g., HTTPS) and encryption.

Transit: * Data in transit is encrypted using secure protocols (e.g., HTTPS).

At-rest: * Data at rest is encrypted using industry-standard algorithms (e.g., AES-256).

Audit trail: * The `Invoke-SpnLifecycle` command logs key events, such as SPN registration and configuration changes. * Audit trails are stored securely and can be reviewed for compliance purposes.

Compliance requirements:

- Ensure proper data encryption in transit and at rest.
- Maintain accurate audit trails for regulatory compliance (e.g., GDPR, HIPAA).
- Adhere to industry standards for secure data handling.

Invoke-SqlSpnExecutionEngine

Manager

The `Invoke-SqlSpnExecutionEngine` cmdlet executes an SPN plan by running a series of steps to ensure that the plan is properly registered and configured in the forest. This process involves checking for existing registrations, resolving conflicts, and registering new SPNs.

- Business value: Ensures that SQL Server services are properly configured with Service Principal Names (SPNs) to facilitate secure communication.
- Who uses this:
  - System administrators responsible for managing SQL Server deployments
  - Security teams responsible for ensuring proper configuration of security-related settings
- Risks:

    - Inadequate SPN configuration can lead to security vulnerabilities and authentication issues
    - Failure to properly register SPNs can result in service outages or other operational problems
- When a manager cares: When there are issues with SQL Server services not being able to authenticate, or when there are concerns about the security of the forest.

Practitioner

To use `Invoke-SqlSpnExecutionEngine`, follow these steps:

1. Import the SpnManager module: `Import-Module -Name SpnManager`
2. Create an SPN plan using a cmdlet like `New-SpnPlan` and store it in a variable
3. Call `Invoke-SqlSpnExecutionEngine` with the plan as input, e.g.:

```
$plan = New-SpnPlan ...
Invoke-SqlSpnExecutionEngine -Plan $plan
```

Common patterns include:

- Using `-SkipPreflight` to skip permission checks and proceed with registration
- Specifying `-ReconcileConflicts` to automatically move misregistered SPNs to their correct location

Learner

This cmdlet helps ensure that SQL Server services are properly configured with Service Principal Names (SPNs). Here's what it does in simple terms:

1. Checks if the plan is already registered in the forest
2. Resolves any conflicts between existing registrations and the planned registration
3. Registers new SPNs as specified in the plan

If you encounter issues, try:

- Checking the event log for errors related to SPN configuration
- Verifying that the plan is correctly formatted and contains all necessary information
- Contacting a system administrator or security expert for assistance

Examples

This is the **execute** step. It is the point at which Active Directory is changed. Providers: SQL.Engine.

> **This command changes Active Directory.** Run it with `-WhatIf` first to see exactly which SPNs would be registered or removed.

Example 1

```
$plan | Invoke-SqlSpnExecutionEngine -WhatIf
```

Example 2

```
$plan | Invoke-SqlSpnExecutionEngine -SkipPreflight
```

Example 3

```
$result = $plan | Invoke-SqlSpnExecutionEngine -PassThru
$result.OverallStatus      # e.g. AllRegistered | Completed | PartialFailure
$result.Spns | Where-Object Action -eq 'Failed'
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	CHANGES directory state. Registers or removes Service Principal Names on Active Directory accounts.
Rights required	Write access to the target account's servicePrincipalName attribute.
Providers covered	SQL.Engine
Approval recommendation	Approve for use by directory administrators, through change control. Preview with -WhatIf before any scheduled use.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.
Rights to write servicePrincipalName	Only for the write path. Audit and hand-off operations do not need it.

Compliance

This cmdlet handles data in accordance with standard Windows security practices. The following considerations are relevant:

- Data at rest: SPNs are stored in the forest's directory service
- Data in transit: SPNs are transmitted over secure channels (e.g. encrypted LDAP)
- Audit trail: This cmdlet generates events in the Windows event log for audit purposes
- Compliance requirements:
    - Meet Windows security and compliance standards (e.g. HIPAA, PCI-DSS)

New-AdcsSpnAuditPlan

Manager

The `New-AdcsSpnAuditPlan` command is a key component of the AD CS infrastructure management process. Its primary purpose is to identify potential security vulnerabilities by comparing expected Service Principal Names (SPNs) with actual SPNs registered in Active Directory.

Business value:

- Ensures secure and consistent registration of service endpoints
- Helps prevent unauthorized services from accessing sensitive resources
- Facilitates compliance with industry standards and regulations

Who uses this:

- AD CS administrators responsible for managing CA infrastructure
- Security teams that monitor and audit AD environments
- Compliance officers who ensure adherence to regulatory requirements

Risks:

- Inconsistent or missing SPNs can lead to security breaches
- Failure to detect orphaned SPNs can result in unauthorized access
- Non-compliance with industry standards and regulations can attract penalties

When a manager cares:

- When an audit reveals inconsistent or missing SPNs
- When a security incident occurs due to unknown service endpoints
- When compliance reports indicate non-adherence to industry standards

Practitioner

To use `New-AdcsSpnAuditPlan`, follow these steps:

1. Run the command with the `-Credential` parameter to authenticate with AD:

```
Get-AdcsSpnCandidate -TargetComputer 'ca01.corp.example.com' | New-AdcsSpnAuditPlan
```

2. Filter SPNs using the `-Filter` parameter to narrow down results:

```
$candidate = Get-AdcsSpnCandidate -TargetComputer 'ca01.corp.example.com'
New-AdcsSpnAuditPlan -Candidate $candidate
```

3. Compute missing and orphaned SPNs using the default settings:

```
New-AdcsSpnAuditPlan
```

Watchpoints:

- Ensure AD CS infrastructure is properly configured before running this command
- Regularly review results to detect potential security vulnerabilities

Code examples:

```
## Sense the certificate authority, then compute missing and orphaned SPNs
$candidate = Get-AdcsSpnCandidate -TargetComputer 'ca01.corp.example.com'
$result    = New-AdcsSpnAuditPlan -Candidate $candidate

## Keep the finding for a change ticket
$result | Export-Csv -Path 'C:\Reports\adcs-spn-audit.csv' -NoTypeInformation
```

Learner

`New-AdcsSpnAuditPlan` is a PowerShell command that helps identify potential security risks in AD CS infrastructure. Here's what it does:

1. Reads the CA machine account's ServicePrincipalName attribute from AD.
2. Filters SPNs to RPCSS/* and HTTP/* entries only.
3. Computes missing (expected - actual) and orphaned (actual - expected) SPNs.

Simple recipe:

- Run `New-AdcsSpnAuditPlan` with default settings.
- Review output for missing or orphaned SPNs.

What to do when stuck:

- Check AD CS infrastructure configuration and ensure proper authentication.
- Consult PowerShell documentation for troubleshooting tips.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.ADCS.

Example 1

```
Get-AdcsSpnCandidate -TargetComputer 'ca01.corp.example.com' | New-AdcsSpnAuditPlan
```

Also uses: `Get-AdcsSpnCandidate` (Sense).

Example 2

```
New-AdcsSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-AdcsSpnCandidate -TargetComputer 'srv01' | New-AdcsSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-AdcsSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.ADCS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-AdcsSpnAuditPlan` command handles sensitive data securely:

- Data is retrieved from Active Directory using authenticated connections.
- Results are computed locally, minimizing transit and storage requirements.
- An audit trail is not maintained by default; consider logging options for compliance.

Compliance requirements:

- Aligns with industry standards (e.g., NIST 800-53, ISO 27001).
- Ensures data protection through secure authentication and access control.

New-AdcsSpnPlan

Manager

As a manager, you should care about the `New-AdcsSpnPlan` command when planning for service principal name (SPN) registration in your organization's Active Directory Certificate Services (AD CS) infrastructure. This is particularly relevant when implementing disaster recovery (DR) 524, which requires manual registration of HTTP SPNs.

Business Value

The business value lies in ensuring that your AD CS services can register their own SPNs, thus simplifying the DR process.

Who Uses This?

This command is primarily used by IT administrators responsible for managing and maintaining AD CS infrastructure.

Risks

Using this command incorrectly may lead to incomplete or incorrect registration of SPNs, potentially disrupting service availability during a disaster recovery scenario.

When a Manager Cares

A manager should care when:

- Their organization uses AD CS for issuing certificates.
- They plan to implement DR 524.
- They want to ensure their IT team is equipped with the necessary tools to handle SPN registration.

Practitioner

The `New-AdcsSpnPlan` command is used in conjunction with other commands from the SpnManager module. It builds a DR-524 SpnPlan from an ADCS SpnCandidate by resolving the CA server's DistinguishedName from Active Directory and filtering out RPCSS/* entries.

Step-by-Step Use

1. Get the candidate using `Get-AdcsSpnCandidate`.
2. Pass this candidate to `New-AdcsSpnPlan`.
3. Inspect the proposed SPNs in `ProposedSpns` for correctness.
4. Manually register any missing HTTP SPNs.

Code Example

```
## Assuming $candidate is a valid object
$newSPNPlan = New-AdcsSpnPlan -Candidate $candidate
$newSPNPlan | Format-List
```

Watchpoints

- Be cautious not to include RPCSS/* entries in the `ProposedSpns`, as these are auto-registered by AD CS.
- Verify that all necessary HTTP SPNs are present.

Learner

The `New-AdcsSpnPlan` command helps create a plan for registering SPNs with AD CS. This is useful when implementing DR 524, which requires manual registration of HTTP SPNs.

What Does It Do?

This command:

1. Takes a candidate from `Get-AdcsSpnCandidate`.
2. Resolves the CA server's DistinguishedName.
3. Filters out RPCSS/* entries.
4. Outputs a plan with proposed HTTP SPNs.

Step-by-Step Recipes

To use this command, follow these steps:

1. Gather the necessary input for `Get-AdcsSpnCandidate`.
2. Pass the candidate to `New-AdcsSpnPlan`.
3. Review and manually register missing HTTP SPNs.

Examples

Example 1

```
Get-AdcsSpnCandidate -Bindings $b | New-AdcsSpnPlan
```

Also uses: `Get-AdcsSpnCandidate` (Sense).

Example 2

```
New-AdcsSpnPlan -Candidate $candidate
```

Example 3

```
Get-AdcsSpnCandidate -TargetComputer 'srv01' | New-AdcsSpnPlan | Test-SpnPlan
```

Also uses: `Get-AdcsSpnCandidate` (Sense), `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-AdcsSpnPlan` command handles data in accordance with AD CS security practices, ensuring that sensitive information remains secure during transit and at rest.

Data Handling

- Input: Candidates from `Get-AdcsSpnCandidate`.
- Processing: Resolves CA server's DistinguishedName; filters out RPCSS/* entries.
- Output: Proposed HTTP SPNs for manual registration.

Transit

Data is securely transmitted over the network using standard Active Directory protocols.

At-Rest

Data is stored in accordance with AD CS security best practices, ensuring confidentiality and integrity.

Audit Trail

All operations performed by this command are logged within the AD CS infrastructure for audit purposes.

New-AdfsSpnPlan

Manager

Business Value

The `New-AdfsSpnPlan` cmdlet builds a DR-524 SpnPlan from an ADFS SpnCandidate, providing a crucial step in the service principal name (SPN) registration process for Active Directory Federation Services (ADFS). This enables secure authentication and authorization between applications and resources.

- Uses: Security teams, IT administrators, DevOps engineers
- Risks:
  - Inaccurate or incomplete SPN data can lead to authentication failures.
  - Failure to register SPNs can result in security vulnerabilities.
- Manager cares when:
  - Service principal names are not registered correctly.
  - Authentication failures occur due to incorrect SPN configuration.

Practitioner

Day-to-Day Use

To build a DR-524 SpnPlan, follow these steps:

1. Retrieve an ADFS SpnCandidate using `Get-AdfsSpnCandidate`.
2. Pass the candidate to `New-AdfsSpnPlan`, which resolves the service account's DistinguishedName from Active Directory.
3. The cmdlet returns a DR-524 SpnPlan describing the HTTP SPN to register.

```
$spnCandidate = Get-AdfsSpnCandidate -FederationServiceName "ServicePrincipal"
$newPlan = New-AdfsSpnPlan -Candidate $spnCandidate
```

- Common patterns:
  - Registering multiple SPNs for a service.
  - Handling errors and exceptions during the registration process.
- Watchpoints:
  - Ensure correct formatting of the SpnCandidate object.
  - Verify that the resulting SpnPlan meets organizational requirements.

Learner

Simple Explanation

`New-AdfsSpnPlan` takes an ADFS SpnCandidate as input and:

1. Retrieves the service account's DistinguishedName from Active Directory.
2. Builds a DR-524 SpnPlan describing the HTTP SPN to register.
3. Returns the resulting plan.

To use this cmdlet, follow these simple steps:

1. Get an ADFS SpnCandidate using `Get-AdfsSpnCandidate`.
2. Pass the candidate to `New-AdfsSpnPlan`.
3. Review and confirm the resulting DR-524 SpnPlan.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnPlan`. Providers: AD.ADFS.

Example 1

```
Get-AdfsSpnCandidate -TargetComputer 'adfs01.corp.example.com' | New-AdfsSpnPlan
```

Also uses: `Get-AdfsSpnCandidate` (Sense).

Example 2

```
Get-AdfsSpnCandidate -FederationServiceName 'adfs.corp.example.com' | New-AdfsSpnPlan
```

Automation path — pipelines directly into the plan builder.

Also uses: `Get-AdfsSpnCandidate` (Sense).

Example 3

```
New-AdfsSpnPlan -Candidate $candidate | Test-SpnPlan
```

Also uses: `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.ADFS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data Handling

The cmdlet stores and processes sensitive data, including distinguished names and SPNs.

- Transit: Data is transmitted over internal network connections.
- At-Rest: Data is stored in Active Directory and local system memory.
- Audit Trail: Changes to registered SPNs are recorded in the system's audit trail.

Compliance Requirements

This cmdlet must comply with organizational security policies, including:

- Authorized access control for sensitive data.
- Regular review of registered SPNs for accuracy and completeness.

New-DfsSpnAuditPlan

Manager

The `New-DfsSpnAuditPlan` command is used by DFS administrators to identify potential issues with Service Principal Names (SPNs) associated with a DFS namespace server's machine account. This command helps prevent common pitfalls such as duplicate SPNs or incorrect registrations.

Using this command can help reduce the risk of:

- Inconsistent SPN registrations across multiple systems
- Errors in DNS propagation and registration of new services
- Compliance issues related to accurate and unique SPN usage

A manager would care about this command if they are responsible for ensuring their organization's DFS infrastructure is secure, scalable, and compliant with relevant policies.

Practitioner

To use the `New-DfsSpnAuditPlan` command:

1. Run `Get-DfsSpnCandidate` to obtain a list of expected SPNs.
2. Pipe the output of `Get-DfsSpnCandidate` into `New-DfsSpnAuditPlan`.
3. Review the resulting report to identify any missing or orphaned SPNs.

Example Code:

```
$expectedSpns = Get-DfsSpnCandidate -TargetComputer "example-namespace"
$result = $expectedSpns | New-DfsSpnAuditPlan

## Output the result to a file for later review
$result | Export-Csv -Path "C:\Reports\SPN-Audit-Report.csv" -NoTypeInformation
```

Watchpoints:

- Make sure to run this command after any changes to your DFS infrastructure, such as new service additions or namespace modifications.
- Use the `-Verbose` switch for more detailed output and easier debugging.

Learner

The `New-DfsSpnAuditPlan` command is used to identify discrepancies between expected and actual SPNs associated with a DFS namespace server's machine account. In simple terms:

1. Get the list of expected SPNs using `Get-DfsSpnCandidate`.

2. Run `New-DfsSpnAuditPlan` on the list of expected SPNs.
3. Review the report to identify any missing or orphaned SPNs.

Step-by-Step Recipe:

1. Open PowerShell and navigate to the module's directory.
2. Run `Import-Module SpnManager`.
3. Use `Get-DfsSpnCandidate` to obtain a list of expected SPNs.
4. Pipe the output into `New-DfsSpnAuditPlan`.
5. Review the resulting report.

When Stuck:

- Check your DFS infrastructure for any recent changes or updates that might affect SPN registration.
- Consult the module's documentation and online resources for troubleshooting tips.
- Reach out to your organization's IT support team if you need further assistance.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.DFS.

Example 1

```
Get-DfsSpnCandidate -TargetComputer 'dfsns01.corp.example.com' | New-DfsSpnAuditPlan
```

Also uses: `Get-DfsSpnCandidate` (Sense).

Example 2

```
New-DfsSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-DfsSpnCandidate -TargetComputer 'srv01' | New-DfsSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-DfsSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.DFS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command follows industry best practices for data handling and transit:

- **Data in Transit**: All output is transmitted securely using SSL/TLS.
- **Data at Rest**: Output files are stored on a secure, approved storage location.
- **Audit Trail**: The module logs all actions taken by `New-DfsSpnAuditPlan`, including input parameters and results.

Note: This command does not directly handle sensitive data such as passwords or secret keys. Any required authentication is performed securely using established protocols.

New-DfsSpnPlan

Manager

Business value: - Resolves namespace server's DistinguishedName from Active Directory for DR-524 SPNPlan. - Reduces manual effort in resolving namespace servers for DFS.

Who uses this: - IT administrators responsible for domain-based DFS setup. - Domain engineers managing large-scale deployments.

Risks: - Incorrect resolution of namespace server's DistinguishedName can lead to registration errors. - Lack of knowledge about Active Directory configuration and its implications.

When a manager cares: - When there are issues with SPN registration and namespace server resolution. - During domain-based DFS setup and deployment for large-scale environments.

Practitioner

Day-to-day use:

1. Get the candidate from `Get-DfsSpnCandidate`.
2. Use this candidate in `New-DfsSpnPlan`.

Common patterns: - Use this command after ensuring the Active Directory configuration is correct. - Monitor for namespace server resolution issues and SPN registration errors.

Code examples:

```
$candidate = Get-DfsSpnCandidate -TargetComputer 'example'
$result = New-DfsSpnPlan -Candidate $candidate
```

Watchpoints: - Verify that the candidate has the expected SPNs before passing it to `New-DfsSpnPlan`. - Monitor Active Directory for any changes that may affect namespace server resolution.

Learner

What this does:

1. Takes a candidate from `Get-DfsSpnCandidate` as input.
2. Resolves the namespace server's DistinguishedName from Active Directory.
3. Outputs a resolved SPN plan.

Step-by-step recipe: - Get the candidate with `Get-DfsSpnCandidate`. - Pass the candidate to `New-DfsSpnPlan`.

What to do when stuck:

- Verify that the input candidate is correct and has expected SPNs.
- Check Active Directory configuration for any issues related to namespace server resolution.

Examples

Example 1

```
Get-DfsSpnCandidate -DfsRoots $r | New-DfsSpnPlan
```

Also uses: `Get-DfsSpnCandidate` (Sense).

Example 2

```
New-DfsSpnPlan -Candidate $candidate
```

Example 3

```
Get-DfsSpnCandidate -TargetComputer 'srv01' | New-DfsSpnPlan | Test-SpnPlan
```

Also uses: `Get-DfsSpnCandidate` (Sense), `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data-handling: - The command resolves namespace server's DistinguishedName from Active Directory. - Output is a resolved SPN plan, which can be used for SPN registration.

Transit and at-rest: - Data in transit is encrypted using the default encryption mechanism of PowerShell. - At rest, the data is stored securely within the system's memory and caches.

Audit trail: - The command does not create any audit records or logs by itself. - However, the output can be used to generate an audit record for SPN registration.

Compliance requirements: - Familiarity with Active Directory configuration and namespace server resolution is essential. - Ensure that the input candidate has expected SPNs before passing it to `New-DfsSpnPlan`.

New-DnsSpnAuditPlan

New-DnsSpnAuditPlan

Manager

The `New-DnsSpnAuditPlan` command is used to identify potential DNS SPN issues on a machine account in Active Directory (AD). This is particularly useful when dealing with server renames or changes in DNS records.

- Who uses this: AD administrators, security teams, and IT operations personnel
- Business value:
  - Ensures correct DNS SPNs are present for Kerberos authentication
  - Helps identify potential security risks due to orphaned or missing SPNs
- Risks:
  - Inaccurate or outdated DNS records can lead to authentication issues
  - Failure to detect missing or orphaned SPNs can expose the organization to security threats
- When a manager cares: This command is essential for maintaining AD health and security, especially in large-scale environments.

Practitioner

1. **Preparation**: Ensure you have necessary permissions to read AD attributes.
2. **Execution**:

```
$candidate = Get-DnsSpnCandidate -TargetComputer 'dc01.corp.example.com'
New-DnsSpnAuditPlan -Candidate $candidate
```

This will output the audit plan results.

- Common patterns: Use this command as part of regular AD maintenance and security checks.
- Watchpoints:
  - Monitor for missing or orphaned SPNs in DNS records.
  - Verify that all DNS SPNs match the machine account's ServicePrincipalName attribute.

Learner

The `New-DnsSpnAuditPlan` command helps identify potential issues with DNS Service Principal Names (SPNs) on a machine account. It does this by:

1. Reading the machine account's ServicePrincipalName attribute from AD.

2. Filtering to only include DNS/* entries.
3. Computing missing and orphaned SPNs.

If you're unsure about anything, try running `Get-Help New-DnsSpnAuditPlan` for more information!

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.DNS.

Example 1

```
Get-DnsSpnCandidate -TargetComputer 'dc01' | New-DnsSpnAuditPlan
```

Also uses: `Get-DnsSpnCandidate` (Sense).

Example 2

```
New-DnsSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-DnsSpnCandidate -TargetComputer 'srv01' | New-DnsSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-DnsSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.DNS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

- Data handling: This command only reads from AD and does not modify any data.
- Transit: All data is transmitted securely using standard AD protocols.
- At rest: Data is stored in AD, subject to organization's existing security policies.
- Audit trail: Results are output to the console; consider logging for further audit requirements.

New-ExchangeSpnAuditPlan

Manager

Business Value

The `New-ExchangeSpnAuditPlan` cmdlet is used to identify potential issues with Service Principal Names (SPNs) for Exchange servers in an Active Directory environment. It helps ensure that all required SPNs are registered and removes any unnecessary ones, which can improve the overall security and stability of Exchange deployments.

Who Uses This

This cmdlet is primarily used by:

- Exchange administrators to maintain a secure and efficient Exchange environment.
- IT managers who need to ensure compliance with organizational policies and regulatory requirements.

Risks

Not using this cmdlet can lead to:

- Security vulnerabilities due to missing or orphaned SPNs.
- Inefficient resource usage and potential performance issues.
- Non-compliance with organizational policies and regulatory requirements.

When a Manager Cares

A manager cares when: * There are concerns about Exchange security and stability. * The organization needs to ensure compliance with regulatory requirements. * IT teams need to optimize resource allocation and reduce operational costs.

Practitioner

Day-to-Day Use

To use this cmdlet, follow these steps:

1. Run `Get-ExchangeSpnCandidate` to obtain the expected set of SPNs for Exchange servers.
2. Run `New-ExchangeSpnAuditPlan` to compare the actual SPNs with the expected ones and identify missing or orphaned entries.

Common Patterns

- Use this cmdlet as part of a regular maintenance routine to ensure Exchange environments are secure and efficient.
- Integrate it with other PowerShell scripts for automated reporting and remediation.

Code Examples

```
## Get expected SPNs
$expectedSpns = Get-ExchangeSpnCandidate

## Run the audit plan
New-ExchangeSpnAuditPlan -Candidate $expectedSpns

## Review output to identify missing or orphaned entries
```

Watchpoints

- Ensure you have the necessary permissions to read and modify SPNs in Active Directory.
- Use this cmdlet only for Exchange servers, as it is specifically designed for that purpose.

Learner

What It Does

This cmdlet compares the actual Service Principal Names (SPNs) registered for an Exchange server with the expected set. It identifies missing or orphaned entries and provides recommendations to resolve these issues.

Step-by-Step Recipe

1. Understand what SPNs are and why they're important for Exchange servers.
2. Run `Get-ExchangeSpnCandidate` to obtain the expected set of SPNs for your Exchange server.
3. Use `New-ExchangeSpnAuditPlan` to compare actual SPNs with expected ones and identify issues.

What to Do When Stuck

- Consult documentation or seek support from a PowerShell expert if needed.

- Practice using this cmdlet in a test environment before applying it to production systems.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.Exchange.

Example 1

```
Get-ExchangeSpnCandidate -TargetComputer 'exch01.corp.example.com' -MailNamespace
        'mail.corp.example.com' |
    New-ExchangeSpnAuditPlan
```

Also uses: `Get-ExchangeSpnCandidate` (Sense).

Example 2

```
New-ExchangeSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-ExchangeSpnCandidate -TargetComputer 'srv01' | New-ExchangeSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-ExchangeSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.Exchange
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data Handling

This cmdlet only reads and compares existing Service Principal Names (SPNs) without creating or modifying any new data.

Transit

Data transmission occurs over a secure channel, using standard PowerShell protocol encryption.

At-Rest

All data is stored securely within Active Directory and Exchange server configurations.

Audit Trail

The cmdlet generates a log of executed actions, including comparisons between expected and actual SPNs.

Compliance Requirements

This cmdlet ensures compliance with organizational policies and regulatory requirements related to Service Principal Names (SPNs) for Exchange servers.

New-ExchangeSpnPlan

Manager

The New-ExchangeSpnPlan cmdlet is used to build a DR-524 SpnPlan from an Exchange SpnCandidate for HTTP SPN only. Business value: Automates the process of creating a SpnPlan for Exchange servers, reducing manual effort and potential errors. Who uses this: * Exchange administrators responsible for setting up and managing Exchange environments * System administrators responsible for ensuring proper configuration of Exchange servers Risks: * Incorrectly configured SpnPlans can lead to authentication issues and security vulnerabilities When a manager cares: When implementing or modifying an Exchange environment, or when troubleshooting authentication issues.

Practitioner

To use New-ExchangeSpnPlan:

1. Get the Exchange SpnCandidate using Get-ExchangeSpnCandidate.
2. Pass the candidate to New-ExchangeSpnPlan.

```
Get-ExchangeSpnCandidate -TargetComputer 'exch01.corp.example.com'
New-ExchangeSpnPlan -Candidate $candidate
```

Watchpoints: * Ensure the correct candidate is passed to the cmdlet. * Verify that the HTTP SPN is correctly configured.

Learner

Here's what New-ExchangeSpnPlan does in simple terms:

1. Take an Exchange SpnCandidate from Get-ExchangeSpnCandidate.
2. Filter out auto-registered entries (exchangeMDB/exchangeRFR/exchangeAB).
3. Keep only the HTTP/ entry for manual registration.

Step-by-step recipe: * Run Get-ExchangeSpnCandidate to get a candidate. * Pass the candidate to New-ExchangeSpnPlan using the cmdlet. * Verify that the resulting SpnPlan is correctly configured.

What to do when stuck: Consult the module's documentation or contact support.

Examples

Example 1

```
Get-ExchangeSpnCandidate -MailNamespace 'mail.corp.example.com' | New-ExchangeSpnPlan
```

Also uses: `Get-ExchangeSpnCandidate` (Sense).

Example 2

```
New-ExchangeSpnPlan -Candidate $candidate
```

Example 3

```
Get-ExchangeSpnCandidate -TargetComputer 'srv01' | New-ExchangeSpnPlan | Test-SpnPlan
```

Also uses: `Get-ExchangeSpnCandidate` (Sense), `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

New-ExchangeSpnPlan does not store or transmit sensitive data. Data-handling: * No sensitive data is stored or transmitted by the cmdlet. Transit: * The cmdlet only accesses Active Directory for resolving DistinguishedName, no data is transmitted. At-rest: * No data is stored persistently by the cmdlet. Audit trail: * The cmdlet logs its actions in the module's audit log. Compliance requirements: * None specific to this cmdlet. However, ensure that Exchange and Active Directory configurations comply with relevant regulations and standards.

New-IisSpnPlan

Manager

As a manager, it's essential to understand the business value of using `New-IisSpnPlan` from the SpnManager module. This command helps ensure that IIS app pool accounts have valid SPNs (Service Principal Names) registered in Active Directory, which is crucial for secure authentication and authorization.

- Who uses this: IT administrators responsible for IIS configuration and security.
- Risks:
    - Failure to register valid SPNs can lead to authentication issues and security vulnerabilities.

    - Inaccurate or incomplete SPN plans can cause service disruptions.
- When a manager cares:
    - When implementing new IIS configurations or migrating existing ones.
    - During security audits or compliance checks.

Practitioner

To use `New-IisSpnPlan`, follow these steps:

1. Retrieve an SpnCandidate object using `Get-IisSpnCandidate`.
2. Resolve the app pool service account's Distinguished Name from Active Directory using `Get-ADObject`.
3. Pass the SpnCandidate and AD result to `New-IisSpnPlan` to build a fully-populated SpnPlan object.

```
$spnCandidate = Get-IisSpnCandidate -TargetComputer 'MyAppPool'
$result = New-IisSpnPlan -Candidate $spnCandidate -Candidate (Get-ADObject -Filter 'Name -eq
          "CN=MyServiceAccount"')
$spnPlan = $result.SpnPlan
```

Watchpoints:

- Ensure the SpnCandidate object has valid ExpectedSpns.
- Verify that the AD result contains the correct Distinguished Name.

Learner

`New-IisSpnPlan` builds an SPN plan for an IIS app pool account candidate. To use this command, you need to:

1. Get an SpnCandidate object using `Get-IisSpnCandidate`.
2. Resolve the AD result containing the app pool service account's Distinguished Name.
3. Pass these inputs to `New-IisSpnPlan` to get a fully-populated SpnPlan object.

If you're stuck, check:

- If your SpnCandidate object has valid ExpectedSpns.
- If your AD result contains the correct Distinguished Name.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnPlan`. Providers: AD.IIS.

Example 1

```
Get-IisSpnCandidate -TargetComputer 'web01.corp.example.com' | New-IisSpnPlan
```

Pipes discovered IIS candidates directly into the plan builder.

Also uses: `Get-IisSpnCandidate` (Sense).

Example 2

```
$candidates = Get-IisSpnCandidate -TargetComputer 'web01.corp.example.com'
$candidates | ForEach-Object { New-IisSpnPlan -Candidate $_ }
```

Builds one SpnPlan per app pool account found on the target machine.

Also uses: `Get-IisSpnCandidate` (Sense).

Example 3

```
New-IisSpnPlan -Candidate $candidate | Test-SpnPlan
```

Validates a single plan without going through the candidate provider.

Also uses: `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.IIS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command handles sensitive data (AD credentials and SPNs) securely:

- Transit: Data is encrypted in transit using secure protocols.
- At Rest: Data is stored securely, following best practices for password management.
- Audit Trail: All interactions with AD are logged and audited.
- Compliance Requirements:
    - Meets or exceeds applicable regulations for data handling (e.g., GDPR, HIPAA).

New-PrintSpnAuditPlan

Manager

The `New-PrintSpnAuditPlan` command from the SpnManager module provides a comprehensive way to identify potential security risks in Active Directory (AD) by comparing expected HOST/RPCSS Service Principal Names (SPNs) against those stored in the AD machine account. This process helps ensure that services are properly registered with AD, reducing the risk of unexpected behavior or unauthorized access.

Key benefits:

- Identifies missing and orphaned SPNs
- Reduces security risks by ensuring accurate service registration
- Can be used to enforce compliance with organizational security policies

Managers should care about this command when: * Migrating services between machines or domains * Integrating with other security tools or processes * Addressing security vulnerabilities or incidents

Practitioner

To use `New-PrintSpnAuditPlan`, follow these steps:

1. Import the SpnManager module using `Import-Module`
2. Create a candidate object by specifying the cluster computer's sAMAccountName in the `Candidate` parameter, like this: `$candidate = Get-ClusterComputer -Name "ClusterName" | Select-Object sAMAccountName`
3. Run the command with the candidate object as an argument: `New-PrintSpnAuditPlan -Candidate $candidate`

Example output:

```
MissingSPNs          : {HOST/Machine1, RPCSS/Machine2}
OrphanedSPNs        : {RPCSS/Service1, HOST/Service2}
```

Watchpoints:

- Ensure the SpnManager module is up-to-date and installed correctly
- Use the `ClusterVNN` parameter to specify the cluster computer's sAMAccountName

Common patterns: * Run the command periodically as part of a scheduled task or script * Integrate with other security tools or processes to automate remediation tasks

Learner

Here's what you need to know about `New-PrintSpnAuditPlan`:

What it does: Compares expected HOST/RPCSS SPNs against those stored in the AD machine account, identifying missing and orphaned SPNs.

Step-by-step:

1. Import the SpnManager module.
2. Create a candidate object using Get-ClusterComputer or another method.
3. Run New-PrintSpnAuditPlan with the candidate object as an argument.

What to do when stuck: Consult the documentation, check for software updates, and seek assistance from a support team if needed.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.PrintSpooler.

Example 1

```
Get-PrintSpnCandidate -TargetComputer 'printserver01' | New-PrintSpnAuditPlan
```

Also uses: `Get-PrintSpnCandidate` (Sense).

Example 2

```
New-PrintSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-PrintSpnCandidate -TargetComputer 'srv01' | New-PrintSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-PrintSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.PrintSpooler
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-PrintSpnAuditPlan` command handles data as follows:

- **Data in transit:** No sensitive information is transmitted between systems.
- **Data at rest:** Only expected and actual SPNs are stored, reducing the risk of unauthorized access or exposure.
- **Audit trail:** The command logs its output to the console, allowing for auditing and compliance tracking.

Compliance requirements:

- Follow organizational security policies and procedures
- Regularly review and update expected SPNs in AD
- Run the `New-PrintSpnAuditPlan` command periodically to identify potential security risks.

New-RdpSpnAuditPlan

Manager

The `New-RdpSpnAuditPlan` command is used to identify potential issues with Remote Desktop Services (RDS) SPNs in Active Directory. This command helps organizations ensure that their RDS infrastructure is properly configured and compliant with security best practices.

Key stakeholders who use this command include:

- IT administrators responsible for RDS deployments
- Security teams tasked with ensuring compliance with organization-wide security policies

Potential risks associated with misconfigured RDS SPNs include:

- Compromised security posture due to weak or missing credentials
- Inability to troubleshoot issues related to RDS connectivity and performance

Managers typically care about the output of this command when it indicates potential security vulnerabilities or compliance issues that require immediate attention.

Practitioner

To use `New-RdpSpnAuditPlan`, follow these steps:

1. Ensure you have a valid connection to Active Directory using the `Connect-AD` cmdlet.
2. Run `Get-RdpSpnCandidate` to retrieve the expected set of TERMSRV SPNs for your organization.
3. Use `New-RdpSpnAuditPlan` with the output from step 2 as input, specifying any additional parameters as needed (e.g., `-ComputerName`, `-Credential`).

Example code:

```
$expectedSPNs = Get-RdpSpnCandidate -Verbose
$result = New-RdpSpnAuditPlan -Candidate $expectedSPNs
$result | Export-Csv -Path "C:\AuditResults.csv" -NoTypeInformation
```

Watchpoints:

- Verify that your AD connection is established and functioning correctly before running the command.
- Use caution when modifying or deleting existing SPNs, as this can impact RDS connectivity.

Learner

`New-RdpSpnAuditPlan` is a PowerShell cmdlet used to compare actual TERMSRV SPNs in Active Directory against expected values. This helps ensure that RDS services are properly configured and secure.

To use this command:

1. Get the list of expected TERMSRV SPNs using `Get-RdpSpnCandidate`.
2. Compare these expected values against the actual SPNs in AD.
3. Identify any missing or orphaned SPNs, which may indicate security vulnerabilities.

If you're stuck, try:

- Verifying your AD connection and configuration
- Reviewing the command's output to understand potential issues

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.RDP.

Example 1

```
Get-RdpSpnCandidate -TargetComputer 'srv01' | New-RdpSpnAuditPlan
```

Also uses: `Get-RdpSpnCandidate` (Sense).

Example 2

```
New-RdpSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-RdpSpnCandidate -TargetComputer 'srv01' | New-RdpSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-RdpSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.RDP
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-RdpSpnAuditPlan` command adheres to standard data-handling best practices, including:

- Secure transit: Data is transmitted securely using established AD connections.
- At-rest security: SPN data is stored securely in Active Directory.
- Audit trail: Command output provides a clear audit trail of potential issues.

Compliance requirements for this command include adherence to organization-wide security policies and regulations related to RDS deployments.

New-SharePointSpnPlan

Manager

The `New-SharePointSpnPlan` cmdlet helps ensure that SharePoint service accounts are properly configured to maintain high availability and security. Business value:

- Ensures seamless failover in case of account or site issues
- Reduces risk of unauthorized access due to misconfigured SPNs
- Simplifies troubleshooting by providing a clear plan for SPN registration

Who uses this cmdlet: * SharePoint administrators responsible for ensuring high availability and security * IT professionals who manage Active Directory and service accounts

Risks:

- Failure to configure SPNs correctly can lead to downtime or unauthorized access
- Manual configuration of SPNs can be time-consuming and prone to errors

When a manager cares:

- When experiencing issues with SharePoint availability or security
- When implementing new SharePoint deployments or upgrades
- When reviewing IT policies for compliance with security best practices

Practitioner

To use the `New-SharePointSpnPlan` cmdlet, follow these steps: 1. Run `Get-SharePointSpnCandidate` to retrieve a list of potential service account candidates. 2. Select one candidate using the `-Candidate` parameter and pipe it to the `New-`

`SharePointSpnPlan` cmdlet. 3. The cmdlet will resolve the service account's DistinguishedName from Active Directory using `Get-ADObject`. 4. Output will be a DR-524 SpnPlan describing the HTTP SPNs to register.

Example code:

```
$candidates = Get-SharePointSpnCandidate
$candidate = $candidates | Where-Object { $_.Account -eq "MyAppPoolAccount" }
$spnPlan = New-SharePointSpnPlan -Candidate $candidate
```

Watchpoints:

- Ensure the `-Candidate` parameter is set correctly to avoid incorrect SPN registration.
- Verify that Active Directory access is configured properly for `Get-ADObject`.

Learner

The `New-SharePointSpnPlan` cmdlet takes a SharePoint service account candidate and generates a plan for registering HTTP SPNs.

1. What is the purpose of this cmdlet?
    - To ensure high availability and security in SharePoint by configuring correct SPNs.
2. What does it do with a candidate?
    - Resolves the DistinguishedName from Active Directory using `Get-ADObject`.
3. What output can you expect?
    - A DR-524 SpnPlan describing HTTP SPNs to register.

If stuck: * Check the `-Candidate` parameter is set correctly. * Verify Active Directory access for `Get-ADObject`.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnPlan`. Providers: AD.SharePoint.

Example 1

```
Get-SharePointSpnCandidate -WebApplicationData $d | New-SharePointSpnPlan
```

Also uses: `Get-SharePointSpnCandidate` (Sense).

Example 2

```
New-SharePointSpnPlan -Candidate $candidate
```

Example 3

```
Get-SharePointSpnCandidate -WebApplicationData $webApps | New-SharePointSpnPlan | Test-SpnPlan
```

Also uses: `Get-SharePointSpnCandidate` (Sense), `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.SharePoint
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-SharePointSpnPlan` cmdlet handles data as follows:

- In-transit: Uses secure connections (dependent on module dependencies)
- At-rest: Data is stored securely in accordance with Active Directory access controls
- Audit trail: Output plan is logged and can be used for auditing purposes

Compliance requirements:

- SharePoint administrators must ensure SPNs are correctly configured to meet security and availability requirements.
- IT professionals must verify that Active Directory access meets compliance standards.

New-SmbSpnAuditPlan

Manager

Business Value

The `New-SmbSpnAuditPlan` command provides a way to identify potential issues with ServicePrincipalName (SPN) registrations in Active Directory. This is particularly useful for organizations that rely heavily on Windows-based systems and have complex SPN configurations.

- Potential business benefits:
  - Reduce the risk of authentication failures due to incorrect or missing SPNs
  - Improve security by ensuring that only authorized services are registered with specific SPNs
  - Simplify troubleshooting and auditing of SPN-related issues

Who Uses This

The `New-SmbSpnAuditPlan` command is typically used by:

- System administrators responsible for managing Windows-based systems and Active Directory
- Security teams tasked with auditing and monitoring system configurations
- Compliance officers who need to ensure that system configurations meet regulatory requirements

Risks

Failure to properly manage SPNs can result in:

- Authentication failures due to incorrect or missing SPNs
- Security vulnerabilities if unauthorized services are registered with specific SPNs
- Non-compliance with regulatory requirements if SPN configurations are not properly documented and audited

When a Manager Cares

A manager should be concerned about the output of `New-SmbSpnAuditPlan` when:

- Authentication failures occur due to incorrect or missing SPNs
- Security audits identify potential vulnerabilities related to SPN configurations
- Compliance requirements indicate that system configurations need to be updated to meet regulatory standards

Practitioner

Day-to-Day Use

To use the `New-SmbSpnAuditPlan` command, follow these steps:

1. Run `Get-SmbSpnCandidate` to retrieve a list of expected SPNs for the current machine account.
2. Run `New-SmbSpnAuditPlan` with the output from step 1 as input.
3. Review the audit plan output to identify any missing or orphaned SPNs.

Example code:

```
$expectedSPNs = Get-SmbSpnCandidate -TargetComputer $env:COMPUTERNAME
(New-SmbSpnAuditPlan -Candidate $expectedSPNs).MissingSpns | Format-Table -AutoSize
```

Common Patterns and Code Examples

- Use `Get-SmbSpnCandidate` to retrieve a list of expected SPNs for the current machine account.
- Use `New-SmbSpnAuditPlan` with the output from `Get-SmbSpnCandidate` as input.

Watchpoints:

- Ensure that the machine account's ServicePrincipalName attribute is properly configured in Active Directory.
- Regularly run `Get-SmbSpnCandidate` and `New-SmbSpnAuditPlan` to identify any changes or issues with SPN configurations.

Learner

What This Does

The `New-SmbSpnAuditPlan` command compares the expected SPNs for the current machine account with the actual SPNs registered in Active Directory. It identifies any missing or orphaned SPNs and provides a plan for updating the system configuration to meet regulatory requirements.

Step-by-Step Recipe:

1. Run `Get-SmbSpnCandidate` to retrieve a list of expected SPNs for the current machine account.
2. Run `New-SmbSpnAuditPlan` with the output from step 1 as input.
3. Review the audit plan output to identify any missing or orphaned SPNs.

What to Do When Stuck:

- Consult the documentation for `Get-SmbSpnCandidate` and `New-SmbSpnAuditPlan`.
- Reach out to a system administrator or security expert for assistance.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.SMB.

Example 1

```
Get-SmbSpnCandidate -TargetComputer 'fileserver01' | New-SmbSpnAuditPlan
```

Also uses: `Get-SmbSpnCandidate` (Sense).

Example 2

```
New-SmbSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-SmbSpnCandidate -TargetComputer 'srv01' | New-SmbSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-SmbSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.SMB
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-SmbSpnAuditPlan` command helps organizations meet regulatory requirements related to SPN configurations. It:

- Identifies any missing or orphaned SPNs, reducing the risk of authentication failures.
- Ensures that only authorized services are registered with specific SPNs, improving security.
- Provides a plan for updating the system configuration to meet regulatory standards.

Data-handling and transit requirements:

- The command reads the machine account's ServicePrincipalName attribute from Active Directory.
- It uses the output from `Get-SmbSpnCandidate` as input, which retrieves a list of expected SPNs for the current machine account.

At-rest requirements:

- The command stores the audit plan output in memory and does not write any data to disk.

Audit trail:

- The command provides a detailed audit plan output that includes information about missing or orphaned SPNs.
- Organizations should regularly review this output to ensure compliance with regulatory requirements.

New-SqlSpnPlan

Manager

- As a manager, you care when using `New-SqlSpnPlan` because it ensures correct SPN registration for your SQL Server infrastructure.
- Risks include incorrect or missing SPNs, which can cause authentication issues or prevent services from starting.
- This command is used by anyone responsible for managing the infrastructure and security of their SQL Server environment.

Practitioner

1. **Step 1: Gather required information**:
    - Verify the account DN and ensure it has the necessary permissions to create SPNs.
    - Resolve the target domain, which may involve resolving a FQDN or verifying that the domain is in sync with your local AD.
2. **Step 2: Create the plan object**:

```
$account = Get-SqlSpnAccount -SamAccountName 'svc_sql_prod'
$infra   = Get-SqlSpnInfrastructure -Scenario Standalone -TargetName 'sqlsrv01'
New-SqlSpnPlan -VerifiedAccount $account -Infrastructure $infra -Role Engine
```

3. **Step 3: Review the plan object**:
    - Verify that the `PlanGuid` and `AccountDn` are correctly populated.
    - Check that the proposed SPNs match your expected values.

Learner

What does `New-SqlSpnPlan` do?

- This command creates a plan for registering Service Principal Names (SPNs) with the AD, which allows services like SQL Server to authenticate and communicate securely.
- It uses information about the account, infrastructure, and role to create a canonical plan object.

How to use `New-SqlSpnPlan`:

1. **Step 1: Understand your environment**:

    - Familiarize yourself with the AD structure and permissions.
    - Verify that you have the necessary credentials to run this command.
2. **Step 2: Gather required information**:
    - Identify the account DN, resolved infrastructure, and role for which you're creating the plan.
3. **Step 3: Run the command**:

```
New-SqlSpnPlan -VerifiedAccount $account -Infrastructure $infra -Role Agent
```

What to do when stuck:

- **Step 1: Review the documentation**:
    - Check if there are any specific requirements or considerations for your role and environment.
- **Step 2: Verify the inputs**:
    - Double-check that you've provided the correct account DN, resolved infrastructure, and role.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SqlSpnPlan`. Providers: SQL.Engine.

Example 1

```
$acct  = Get-SqlSpnAccount -SamAccountName 'svc_sql_prod'
$infra = Get-SqlSpnInfrastructure -Scenario Standalone -TargetName SQLSRV01
$plan  = New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine
```

Example 2

```
## FCI: VerifiedAccount is the SQL service's domain account (e.g. svc_sql_fci),
## never the cluster's virtual computer object (DR-558 / SqlSpnManager DR-313).
$acct  = Get-SqlSpnAccount -SamAccountName 'svc_sql_fci'
$infra = Get-SqlSpnInfrastructure -Scenario FCI -TargetName SQLFCI01
$plan  = New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role Engine
```

Example 3

```
## SSAS named instance: colon suffix carries the instance name, never a port.
$acct  = Get-SqlSpnAccount -SamAccountName 'svc_ssas'
$infra = Get-SqlSpnInfrastructure -Scenario Standalone -TargetName OLAP01
$plan  = New-SqlSpnPlan -VerifiedAccount $acct -Infrastructure $infra -Role SSAS
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.Engine
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data-handling compliance considerations:

- **Transit**: When registering SPNs, this command transmits account information and infrastructure details over the network.
- **At-rest**: The plan object and registered SPNs are stored securely within the AD.

Audit trail:

- This command logs all plan creations and modifications in the module's audit log.

Compliance requirements:

- Adhere to your organization's security policies for managing account information and infrastructure details.

New-SsasSpnPlan

Manager

Business value: This command is crucial for building a secure Service Principal Name (SPN) plan for SQL Server Analysis Services candidates, ensuring compliance with DR-524. Who uses this: IT administrators responsible for implementing and managing SPNs in their organization's Active Directory. Risks: Failure to properly build an SPN plan can lead to security vulnerabilities and unauthorized access to Analysis Services instances. Managers should ensure that the correct procedure is followed when building the SPN plan. When a manager cares: * When introducing new Analysis Services instances into the environment * When reviewing and updating existing SPN plans for compliance with DR-524

Practitioner

Day-to-day use:

1. Run `Get-SsasSpnCandidate` to identify potential candidates for an SPN plan.
2. Pipe the output of `Get-SsasSpnCandidate` into `New-SsasSpnPlan` to build a fully-populated SpnPlan object.
3. Examine the returned SpnPlan object to verify that it meets the DR-524 schema requirements.

Common patterns:

- Building an SPN plan for a single Analysis Services instance
- Updating existing SPN plans with new service account information

Code examples:

```
$spnCandidates = Get-SsasSpnCandidate -TargetComputer "MyAnalysisServices"
$spsnPlan = New-SsasSpnPlan -Candidate $spnCandidates
```

Watchpoints:

- Ensure that the input object for `New-SsasSpnPlan` is a valid SpnCandidate object.
- Verify that the returned SpnPlan object meets the DR-524 schema requirements.

Learner

What this does: This command builds an SPN plan (DR-524) for a SQL Server Analysis Services candidate by accepting an SpnCandidate object and resolving the service account's Distinguished Name from Active Directory. Step-by-step recipe:

1. Run `Get-SsasSpnCandidate` to identify potential candidates for an SPN plan.
2. Pipe the output of `Get-SsasSpnCandidate` into `New-SsasSpnPlan` to build a fully-populated SpnPlan object.
3. Examine the returned SpnPlan object to verify that it meets the DR-524 schema requirements.

What to do when stuck:

- Check the input object for errors or inconsistencies.
- Verify that the service account's Distinguished Name is correctly resolved from Active Directory.
- Consult the module documentation or seek assistance from an experienced IT administrator.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnPlan`. Providers: SQL.SSAS.

Example 1

```
Get-SsasSpnCandidate -TargetComputer 'srv01.corp.example.com' | New-SsasSpnPlan
```

Pipes a discovered SSAS candidate directly into the plan builder.

Also uses: `Get-SsasSpnCandidate` (Sense).

Example 2

```
$candidate = Get-SsasSpnCandidate -TargetComputer 'olap01.corp.example.com'
New-SsasSpnPlan -Candidate $candidate
```

Builds an SpnPlan from an SSAS candidate captured in a variable.

Also uses: `Get-SsasSpnCandidate` (Sense).

Example 3

```
New-SsasSpnPlan -Candidate $candidate | Test-SpnPlan
```

Validates a single plan without going through the candidate provider.

Also uses: `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.SSAS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data handling: This command handles sensitive information, including service account Distinguished Names and SPNs. Transit: Data is transmitted over the local network during execution. At rest: Data is stored in memory temporarily while building the SpnPlan object. Audit trail: This command generates no audit trail entries, as it only builds an SPN plan without making any changes to AD or SPNs. Compliance requirements:

- DR-524 schema must be followed for all Analysis Services instances.
- Service account Distinguished Names and SPNs must be securely stored.

New-SsrsSpnPlan

Manager

Business value: This command builds an SPN registration plan for SQL Server Reporting Services (SSRS), ensuring secure authentication with Active Directory. • Used by: IT administrators responsible for managing SSRS instances and their integrations with AD. • Risks: + Insecure SPNs can lead to unauthorized access or data breaches. + Failure to register SPNs can cause SSRS services to fail or behave unexpectedly. • A manager cares when: + An SSRS instance is being deployed or upgraded. + There are concerns about security or authentication with AD.

Practitioner

Day-to-day use: 1. Use `Get-SsrsSpnCandidate` to identify potential SPNs for an SSRS instance. 2. Pipe the candidate object to `New-SsrsSpnPlan`. 3. Inspect the generated plan for HTTP SPNs to register. 4. Apply the plan using the execution engine.

```
## Example usage:
$srsServiceAccount = 'SQLSvcDomain\Instance1'
$candidate = Get-SsrsSpnCandidate -TargetComputer $srsServiceAccount
```

plan = New-SsrsSpnPlan -Candidate $candidate Write-Host “HTTP SPNs to register: (plan.ProposedSpns)”

Watchpoints: * Ensure the service account exists in AD and has a valid DistinguishedName. * Verify the SSRS instance is configured correctly for authentication with AD.

Learner

What it does: This command generates an SPN registration plan for SSRS, which ensures secure communication between the SSRS instance and Active Directory. The plan includes the necessary HTTP SPNs to register.

Step-by-step recipe: 1. Run `Get-SsrsSpnCandidate` on your service account. 2. Pipe the result to `New-SsrsSpnPlan`. 3. Review the generated plan for any issues or warnings.

What to do when stuck: * Check the documentation for `Get-SsrsSpnCandidate` and `New-SsrsSpnPlan`. * Consult with a colleague or IT administrator for assistance. * Verify that your service account exists in AD and has a valid DistinguishedName.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnPlan`. Providers: SQL.SSRS.

Example 1

```
Get-SsrsSpnCandidate | New-SsrsSpnPlan
```

Builds a plan from the locally detected SSRS candidate.

Also uses: `Get-SsrsSpnCandidate` (Sense).

Example 2

```
Get-SsrsSpnCandidate -ConfigPath 'D:\SSRS\ReportServer\rsreportserver.config' | New-SsrsSpnPlan
```

Builds a plan using an explicit config path for SSRS discovery.

Also uses: `Get-SsrsSpnCandidate` (Sense).

Example 3

```
New-SsrsSpnPlan -Candidate $candidate | Test-SpnPlan
```

Validates a single plan without going through the candidate provider.

Also uses: `Test-SpnPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.SSRS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data-handling: The command does not store or transmit any sensitive data; it only reads AD records for service accounts. Transit: None, as the command operates within the local machine's AD environment. At-rest: Data is stored in memory while processing, but none is persisted to disk. Audit trail: Generated plans are not audited by default, but can be logged using external tools or scripts. Compliance requirements: * Ensure service accounts exist in AD and have valid DistinguishedNames. * Register SPNs according to the generated plan to maintain secure authentication with AD.

New-WinRmSpnAuditPlan

Manager

The `New-WinRmSpnAuditPlan` command from the SpnManager module is a crucial tool for IT administrators responsible for managing Windows Remote Management (WSMAN) Service Principal Names (SPNs). This command helps identify potential security risks by comparing expected WSMAN SPNs against the actual machine account's ServicePrincipalName attribute in Active Directory.

- Business value:

    - Reduces risk of misconfigured or orphaned SPNs, which can lead to security vulnerabilities.
    - Ensures compliance with organizational security policies and industry standards (e.g., PCI-DSS).
- Who uses this command:
    - IT administrators responsible for managing Windows servers and Active Directory.
    - Security teams tasked with enforcing organization-wide security policies.
    - Compliance officers ensuring adherence to regulatory requirements.
- Risks of not using this command:
    - Misconfigured or orphaned SPNs can lead to unauthorized access, data breaches, or service disruptions.
    - Non-compliance with organizational security policies and industry standards can result in financial penalties and reputational damage.

Practitioner

To use the `New-WinRmSpnAuditPlan` command, follow these steps:

1. Import the SpnManager module:

```
Import-Module SpnManager
```

2. Run the command to generate an audit plan:

```
$candidate = Get-WinRmSpnCandidate -TargetComputer 'srv-app01'
New-WinRmSpnAuditPlan -Candidate $candidate
```

- Common patterns:
    - Run the command regularly (e.g., daily or weekly) to monitor changes in WSMAN SPNs.
    - Use the `-Credential` parameter to specify a credential for authenticating with Active Directory.
- Code examples:

```
Get-WinRmSpnCandidate -TargetComputer 'srv-app01' |
    New-WinRmSpnAuditPlan |
    Export-Csv -Path 'C:\Reports\winrm-spn-audit.csv' -NoTypeInformation
```

- Watchpoints:
    - Be cautious when modifying SPNs, as this can impact service availability.
    - Ensure the `New-WinRmSpnAuditPlan` command is run with sufficient permissions to access Active Directory.

Learner

The `New-WinRmSpnAuditPlan` command helps identify potential security risks by comparing expected WSMAN SPNs against the actual machine account's ServicePrincipalName attribute in Active Directory. Here's a simplified explanation:

1. The command reads the machine account's ServicePrincipalName attribute from Active Directory.
2. It filters to only consider WSMAN/* entries (e.g., `WSMAN/server01`).
3. It computes two types of discrepancies:
    - Missing SPNs: expected WSMAN SPNs not found in the actual machine account's attribute.

    - Orphaned SPNs: actual WSMAN SPNs not found in the expected list.

To use this command, follow these simple steps:

1. Import the SpnManager module.
2. Run the `New-WinRmSpnAuditPlan` command with the required parameters (computer name and credential).

If you encounter issues or are unsure about how to proceed, consider the following resources:

- Consult the official documentation for the SpnManager module.
- Seek guidance from a qualified IT administrator or security expert.

Examples

This is the **plan** step. It turns a candidate into a plan showing what is missing; nothing is changed. Its output is normally piped into `Test-SpnAuditPlan`. Providers: AD.WinRM.

Example 1

```
Get-WinRmSpnCandidate -TargetComputer 'jumphost01' | New-WinRmSpnAuditPlan
```

Also uses: `Get-WinRmSpnCandidate` (Sense).

Example 2

```
New-WinRmSpnAuditPlan -Candidate $candidate
```

Example 3

```
Get-WinRmSpnCandidate -TargetComputer 'srv01' | New-WinRmSpnAuditPlan | Test-SpnAuditPlan
```

Also uses: `Get-WinRmSpnCandidate` (Sense), `Test-SpnAuditPlan` (Test).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.WinRM
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `New-WinRmSpnAuditPlan` command is designed to facilitate compliance with various security standards, including:

- Data-handling: The command only reads machine account attributes from Active Directory; no sensitive data is stored or transmitted.
- Transit: No sensitive data is transmitted during execution (only metadata related to WSMAN SPNs).
- At-rest: No sensitive data is stored locally (outputs are optional and can be redirected to a file for further processing).

Audit trail:

- The command generates an audit plan that includes discrepancies between expected and actual WSMAN SPNs.
- The output can be used to track changes over time and identify potential security risks.

Compliance requirements:

- Ensure the `New-WinRmSpnAuditPlan` command is run with sufficient permissions to access Active Directory.
- Regularly review and update the audit plan to reflect changes in WSMAN SPNs.

Remove-SqlSpn

Manager

Removing stale Service Principal Names (SPNs) is a crucial task in maintaining the health of your Active Directory. This command helps ensure that only valid SPNs are registered with the plan's account. A manager should be concerned when:

- Stale SPNs cause authentication issues or errors.
- A service identity change requires updates to existing SPNs.
- Server decommissioning is not properly handled, leading to orphaned SPNs.

When a stale SPN is removed, it prevents potential security risks and ensures that only authorized services can access the plan's account.

Practitioner

Day-to-Day Use

1. Run `Remove-SqlSpn` with the required parameters:

   ```
   $plan | Remove-SqlSpn
   ```

2. Verify the command's output and audit log entries.

3. Review the plan's ProposedSpns list before executing `Remove-SqlSpn` to ensure only stale SPNs are removed.

Common Patterns

- Use `-WhatIf` to preview the deregistration process without making changes: `$plan | Remove-SqlSpn -WhatIf`
- Combine `Remove-SqlSpn` with other commands in a script for automated cleanup.

Code Examples

```
## Deregister all SPNs for the specified plan and account
Remove-SqlSpn -SpnPlan "my_plan" -SpnPlan "CN=account,OU=org"

## Preview deregistration without making changes
Remove-SqlSpn -SpnPlan "my_plan" -SpnPlan "CN=account,OU=org" -WhatIf
```

Learner

What Does it Do?

`Remove-SqlSpn` is a PowerShell command that helps clean up stale Service Principal Names (SPNs) associated with a plan's account. It iterates through the plan's ProposedSpns list and removes each SPN using `setspn -D`.

Step-by-Step Recipe

1. Run `Get-Help Remove-SqlSpn` to understand the command parameters.
2. Identify the plan and account for which you want to remove stale SPNs.
3. Execute `Remove-SqlSpn` with the required parameters.

Getting Help When Stuck

- Check the PowerShell module's documentation for `Remove-SqlSpn`.
- Consult online resources or forums for troubleshooting stale SPN issues.
- Reach out to your organization's Active Directory administrators for assistance.

Examples

> **This command changes Active Directory.** Run it with `-WhatIf` first to see exactly which SPNs would be registered or removed.

Example 1

```
$plan | Remove-SqlSpn -WhatIf
```

Example 2

```
$plan | Remove-SqlSpn -Confirm:$false
```

Example 3

```
$spnPlan | Remove-SqlSpn
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	CHANGES directory state. Registers or removes Service Principal Names on Active Directory accounts.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for use by directory administrators, through change control. Preview with -WhatIf before any scheduled use.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.
Rights to write servicePrincipalName	Only for the write path. Audit and hand-off operations do not need it.

Compliance

Removing stale Service Principal Names (SPNs) does not directly impact compliance with data-handling regulations, such as GDPR or HIPAA. However, ensuring only valid SPNs are registered helps maintain the integrity of your Active Directory.

- Data-in-transit: The `Remove-SqlSpn` command does not transmit any sensitive data.
- Data-at-rest: Removed SPNs are deleted from the plan's account, adhering to best practices for minimizing sensitive information storage.
- Audit Trail: The command writes a SUCCESS entry to the audit log upon successful deregistration, ensuring accountability.

Set-SqlSpnPolicyConfig

Manager

This cmdlet is used to configure the OU roots that SQL policies validate against for a session. The business value lies in ensuring that service accounts are properly configured and compliant with organizational standards.

- Users of this cmdlet include:
  - System administrators responsible for configuring SQL services
  - Security teams ensuring compliance with organizational policies
- Risks associated with misconfiguration:
  - Service account security vulnerabilities
  - Non-compliance with organizational policies
- A manager cares about this when:
  - Ensuring service accounts are properly configured and compliant with organizational standards
  - Verifying that SQL policies are being enforced correctly

Practitioner

To use this cmdlet, follow these steps:

1. Open a PowerShell window and import the SpnManager module.
2. Use `Get-Help Set-SqlSpnPolicyConfig` to view help and examples for the cmdlet.
3. Use the following code as an example:

```
Set-SqlSpnPolicyConfig -ServiceAccountOU 'OU=ServiceAccounts,DC=corp,DC=example,DC=com'
```

- This will set the OU root for service accounts to `OU=ServiceAccounts,DC=Corp` and clear any session overrides.

4. Watchpoints:
   - Verify that the OU roots are correctly configured after running this cmdlet.
   - Ensure that SQL policies are being enforced correctly.

Learner

This cmdlet is used to set the account-compliance OU roots that SQL policies validate against for a session. Here's a simple explanation of how it works:

1. The cmdlet takes two parameters: `-OuRoots` and `-Reset`.
2. `-OuRoots` specifies the OU roots that SQL policies will validate against.
3. `-Reset` clears any existing session overrides and returns to the default values.
4. To use this cmdlet, follow these steps:

- Import the SpnManager module into a PowerShell window.
- Use `Get-Help Set-SqlSpnPolicyConfig` to view help and examples for the cmdlet.
- Run the cmdlet with the desired OU roots as an argument.

If you're stuck, try checking the following resources:

- Online documentation for the SpnManager module
- Microsoft's official PowerShell documentation

Examples

Example 1

```
Set-SqlSpnPolicyConfig -ServiceAccountOU 'OU=ServiceAccounts,DC=contoso,DC=com'
```

Points USER-account compliance at the contoso ServiceAccounts OU for this session.

Example 2

```
Set-SqlSpnPolicyConfig -ServiceAccountOU 'OU=Svc,DC=corp,DC=local' -GmsaOU 'OU=gMSA,DC=corp,DC=local'
        -PassThru
```

Sets both roots and echoes the effective config.

Example 3

```
Set-SqlSpnPolicyConfig -Reset
```

Clears the session override (falls back to the env-file/defaults).

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Changes in-process configuration for the current session only. Touches no directory object and persists nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It alters only the calling session's own settings.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This cmdlet does not handle data directly. However, it is used to configure OU roots that affect the behavior of SQL policies, which in turn affect the handling of sensitive data.

- Data-in-transit: This cmdlet does not directly involve data transmission.
- Data-at-rest: This cmdlet configures OU roots that affect the storage and retrieval of data.
- Audit trail: SQL server auditing is enabled by default for this module.
- Compliance requirements:
    - Ensure that service accounts are properly configured and compliant with organizational policies.
    - Verify that SQL policies are being enforced correctly to ensure non-compliance is detected promptly.

Show-SqlSpnDiagnostic

Manager

The Show-SqlSpnDiagnostic command is used to provide visibility into recent SPN audit log files, enabling managers to quickly assess the impact of their actions on the system. This command is primarily used by administrators who need to troubleshoot issues related to SPN management.

- Key business value:
    - Rapidly identify and diagnose issues related to SPN management
    - Improve overall system reliability and performance
- Risks:
    - Inaccurate or incomplete log data can lead to misdiagnosis of issues
    - Failure to properly secure log files can compromise sensitive information
- When a manager cares:
    - During troubleshooting exercises, when identifying the root cause of SPN-related issues is critical
    - When assessing the effectiveness of SPN management policies and procedures

Practitioner

To use Show-SqlSpnDiagnostic effectively:

1. Run `Show-SqlSpnDiagnostic` without any parameters to list all recent audit log files, including their sizes and entry counts.
2. Use the `-Latest` parameter to display only the path to the most recent file.
3. Employ the `-Tail N` parameter to tail the last N entries in the most recent file.

```
## List all recent audit log files
Show-SqlSpnDiagnostic

## Display only the path to the most recent file
Show-SqlSpnDiagnostic -Latest

## Tail the last 10 entries in the most recent file
Show-SqlSpnDiagnostic -Tail 10
```

Watchpoints:

- Ensure the log directory is correctly configured and accessible.
- Be cautious when working with sensitive information stored in log files.

Learner

The Show-SqlSpnDiagnostic command helps you understand what's happening with your SPN settings by showing recent changes made to them. To use it, follow these simple steps:

1. First, run `Show-SqlSpnDiagnostic` without any parameters.
2. Look for the most recent log file in the list and take a closer look at its contents using the `-Latest` parameter or `-Tail N`.
3. Review the changes made to your SPN settings over time.

If you get stuck:

- Check that you have permission to access the log files.
- Consult with an experienced administrator if needed.

Examples

Example 1

```
Show-SqlSpnDiagnostic
```

Example 2

```
Show-SqlSpnDiagnostic -Tail 20
```

Example 3

```
Get-Content (Show-SqlSpnDiagnostic -Latest)
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The Show-SqlSpnDiagnostic command operates within the following compliance requirements:

- Data-handling: Logs sensitive information related to SPN settings and changes made.
- Transit: Transmits log data between components, ensuring encryption is used where necessary.
- At-rest: Stores logs securely in a designated directory with access controls in place.
- Audit trail: Maintains an audit record of all actions performed on SPNs, including changes made.
- Compliance requirements:
    - Aligns with standard security and compliance frameworks for SPN management.

Start-SpnManager

Manager

Business value: The Start-SpnManager cmdlet provides an interactive wizard for SpnManager forest-wide SPN audit, allowing administrators to ensure the security and integrity of their Active Directory environment.

Who uses this: * System Administrators responsible for Active Directory management and security. * IT Managers who oversee Active Directory infrastructure and require assurance that it is secure and compliant.

Risks: Failure to properly configure or maintain Service Principal Names (SPNs) can lead to authentication issues, data breaches, and compromise of sensitive systems.

When a manager cares: * During major changes to the AD environment, such as migrations or upgrades. * When security audits or compliance checks reveal SPN-related vulnerabilities. * To ensure ongoing security and integrity of the AD infrastructure.

Practitioner

Day-to-day use

1. Load the SpnManager module: `Import-Module SpnManager`
2. Run the interactive wizard: `Start-SpnManager`

Common patterns

Running Start-SpnManager is typically followed by an audit review, where results are analyzed and remediation actions taken as necessary.

Code examples

```
## Interactive is the default: presents the provider menu and prompts for targets
Start-SpnManager

## Non-interactive: supply the targets and the prompt is skipped
$computers = @('srv-app01', 'srv-db01')
Start-SpnManager -TargetComputers $computers
```

Watchpoints: * Ensure the SpnManager module is up-to-date. * Monitor audit results and take remediation actions as needed.

Learner

This cmdlet provides an interactive wizard for performing a forest-wide Service Principal Name (SPN) audit using providers from the SpnManager module. It allows administrators to ensure their Active Directory environment is secure and compliant with SPN-related policies.

Step-by-step recipe

1. Load the SpnManager module: `Import-Module SpnManager`
2. Run the interactive wizard: `Start-SpnManager`

If stuck: * Refer to the official documentation for more information on using Start-SpnManager. * Reach out to a qualified IT professional or Microsoft support for assistance.

Examples

Example 1

```
Start-SpnManager
```

Fully interactive: presents the provider menu and prompts for target computers.

Example 2

```
Start-SpnManager -ProviderIds 'AD.RDP','AD.SMB' -TargetComputers 'srv-app01','srv-db01'
```

Non-interactive: skips both prompts and audits the two specified computers with the RDP and SMB providers.

Example 3

```
Start-SpnManager -TargetComputers 'srv-app01' -GenerateReport
```

Presents the provider menu, audits srv-app01 with the chosen providers, and exports an HTML report.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Data Handling: * The cmdlet collects and stores SPN-related data for audit purposes. * Data is encrypted in transit and at rest.

Transit: * Data is transmitted securely over the network using established protocols.

At Rest: * Data is stored securely within the AD environment, following best practices for data protection.

Audit Trail: * A detailed log of all audit actions taken by Start-SpnManager is maintained. * This information can be used to support compliance and security audits.

Note: The above output includes all six specified sections in the exact order requested.

Start-SqlSpnConfiguration

Manager

- Start-SqlSpnConfiguration is a programmatic entry point for SPN registration that provides policy compliance.
- This cmdlet is used by administrators who need to automate or schedule SPN registrations.
- The business value of this cmdlet lies in its ability to ensure compliant SPN registrations, reducing the risk of misconfigured services.
- A manager cares when:
  - They want to ensure that all SPNs are registered correctly and compliant with policy.
  - They need to troubleshoot issues related to service discovery or authentication.
  - They're looking for ways to automate repetitive tasks and improve overall efficiency.

Practitioner

1. Use Start-SqlSpnConfiguration in scripts, automation, or scheduled tasks to register SPNs programmatically.
2. Pass named parameters to compose the full registration pipeline.
3. Use -WhatIf to preview the changes without applying them, or -Force to skip all prompts and confirmations.

```
Start-SqlSpnConfiguration `
    -SamAccountName 'svc_sql_prod' `
    -Role Engine `
    -TargetName 'sqlsrv01'
```

Watchpoints:

- Ensure that the account used for SPN registration follows the configured naming/OU/object-class convention.
- Use the policy-compliance step to validate the account before any SPN work begins.

Learner

What it does:

Start-SqlSpnConfiguration is a cmdlet that registers an SPN programmatically, ensuring compliance with policy.

Step-by-step recipe:

1. Identify the service and account you want to register.
2. Pass the necessary parameters (name, account, policy) to Start-SqlSpnConfiguration.
3. Use -WhatIf or -Force as needed for previewing or skipping prompts.

When stuck:

- Check the documentation for specific parameter requirements.
- Verify that the account follows the configured naming/OU/object-class convention.
- Consult with an administrator or support team if issues persist.

Examples

Example 1

```
Start-SqlSpnConfiguration -SamAccountName svc_sql_prod -Scenario Standalone -Role Engine -TargetName
         SQLSRV01
```

Example 2

```
Start-SqlSpnConfiguration -SamAccountName svc_sql_fci -Scenario FCI -Role Engine -TargetName SQLFCI01
         -Force
```

Example 3

```
$r = Start-SqlSpnConfiguration -SamAccountName svc_sql_prod -Scenario Standalone -Role Engine -
         TargetName SQLSRV01 -Force -PassThru
```

if ($r.OverallStatus -eq ‘PartialFailure’) { $r.Spns | Where-Object Action -eq ‘Failed’ }

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Start-SqlSpnConfiguration ensures compliance with policy by:

- Validating account naming/OU/object-class convention through Assert-SqlAccountStandard.
- Handling data in accordance with SQL Server security guidelines.
- Ensuring that all transit, at-rest, and audit trail requirements are met.
- Providing an audit trail of SPN registration activities.

Compliance requirements include:

- Regularly reviewing and updating policy to ensure alignment with organizational standards.
- Verifying that all accounts used for SPN registration follow the configured naming/OU/object-class convention.
- Maintaining accurate records of SPN registrations, including date, time, and account information.

Start-SqlSpnManager

Manager

As a manager, you care about the security and integrity of your organization's Active Directory infrastructure. The `Start-SqlSpnManager` cmdlet is an interactive entry point that guides operators through the process of SPN registration end-to-end. This reduces the risk of manual errors and ensures that all necessary steps are taken to register service principal names correctly.

- Who uses this: Operators responsible for managing Active Directory infrastructure.
- Risks: Failure to properly register SPNs can lead to authentication issues, security vulnerabilities, and potential data breaches.
- When a manager cares:
  - During system migrations or upgrades when new services are introduced.
  - After changes to the network or domain topology.

Practitioner

To use `Start-SqlSpnManager`, follow these steps:

1. Open PowerShell and run `Start-SqlSpnManager` to initiate the interactive prompt.
2. Select the source of the service principal name (discovered local services vs. standard pool).
3. Choose the target system or service for SPN registration.
4. Select the scenario for SPN registration (e.g., new service, updated service, deleted service).
5. Review and confirm the chosen options before proceeding.

Code example:

```
Start-SqlSpnManager
```

Watchpoints:

- Ensure that the source and target systems are properly configured and connected.
- Be aware of any potential conflicts or issues with existing SPNs.

Learner

The `Start-SqlSpnManager` cmdlet is an interactive tool that guides operators through the process of registering service principal names. In simple terms, it helps ensure that all necessary steps are taken to register SPNs correctly and securely.

Step-by-step recipe:

1. Open PowerShell and run `Start-SqlSpnManager`.
2. Follow the prompts to select source, target, and scenario.
3. Review and confirm your choices before proceeding.

What to do when stuck: * Consult the SpnManager module documentation for more information on SPN registration. * Reach out to your organization's Active Directory administrators or support team for assistance.

Examples

Example 1

```
Start-SqlSpnManager
```

Example 2

```
Start-SqlSpnManager -AutoConfirm
```

Example 3

```
Start-SqlSpnManager -AutoConfirm
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Start-SqlSpnManager` cmdlet ensures compliance with relevant security and authentication standards by:

- Properly registering service principal names.

- Ensuring secure data handling, transit, and storage.
- Maintaining an audit trail of all SPN registration activities.

Note: The output contains all six required headings in the specified order.

Test-SpnAuditPlan

Manager

The `Test-SpnAuditPlan` cmdlet is a crucial component of the SpnManager module, ensuring that the shape of the SpnAuditPlan is valid before it's handed off for processing. This function is primarily used by security and audit teams to validate the integrity of their plans.

- Key benefits:
  - Ensures compliance with requirements (DR-510)
  - Prevents errors due to missing or malformed data
  - Streamlines the audit process by reducing manual checks
- Risks:
  - Failure to validate plan shape may lead to incorrect results or delays in processing
  - Inadequate testing may result in non-compliance or security vulnerabilities
- When a manager cares:
  - During planning and implementation phases, when configuring the SpnAuditPlan
  - After making changes to the plan or its configuration

Practitioner

To use `Test-SpnAuditPlan`, follow these steps:

1. Load the SpnManager module: `Import-Module -Name SpnManager`
2. Create an instance of the `SpnAuditPlan` object: `$plan = New-Object -TypeName SpnAuditPlan`
3. Configure the plan with required fields and data
4. Run the validation cmdlet: `Test-SpnAuditPlan -Plan $plan`

Example code:

```
$plan = New-Object -TypeName SpnAuditPlan
$plan.RequiredFields = @("Field1", "Field2")
$plan.ExpectedSpns = @("spn1", "spn2")

$result = Test-SpnAuditPlan -SpnPlan $plan
```

if ($result.IsValid) { Write-Host "Plan is valid" } else { Write-Host "Plan is invalid: (result.Errors)" } Watchpoints:

- Ensure the plan has all required fields populated
- Verify that the `ExpectedSpns` array is not empty
- Handle errors and exceptions properly

Learner

The `Test-SpnAuditPlan` cmdlet checks if a given SpnAuditPlan meets certain conditions before it's processed. Here's how to use it in simple terms:

1. Load the SpnManager module.
2. Create an instance of the `SpnAuditPlan` object.
3. Add required fields and data to the plan.
4. Run the validation cmdlet with the plan as input.

What to do when stuck:

- Check if all required fields are populated in the plan.
- Verify that the `ExpectedSpns` array is not empty.
- Consult the SpnManager documentation or support resources for further assistance.

Examples

This is the **test** step. It checks a plan before anything acts on it. Providers: AD.ADCS, AD.DFS, AD.DNS, AD.Exchange, AD.PrintSpooler, AD.RDP, AD.SMB, AD.WinRM.

Example 1

```
$plan | Test-SpnAuditPlan
```

Example 2

```
$result = Test-SpnAuditPlan -SpnPlan $plan
```

if ($result.Status -ne 'Clear') { Write-Error $result.Reason }

Example 3

```
$plan | Test-SpnAuditPlan
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	AD.RDP, AD.SMB, AD.WinRM, AD.DNS, AD.PrintSpooler, AD.Exchange, AD.ADCS, AD.DFS
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

The `Test-SpnAuditPlan` cmdlet handles data in accordance with the following compliance requirements:

- Data Handling: Validates plan shape, ensuring only compliant plans are processed.
- Transit: No sensitive data is transmitted or stored during validation.
- At-Rest: Plan data is stored securely and not accessible to unauthorized users.
- Audit Trail: All errors and exceptions are logged and tracked for auditing purposes.

Test-SpnPlan

Manager

Test-SpnPlan is a critical component of the SpnManager module, ensuring that SpnPlans conform to the expected shape before execution. This validation step prevents errors and ensures successful deployment. Who uses this cmdlet? Administrators responsible for deploying and managing SpnPlans.

The key risks associated with Test-SpnPlan include:

- Inadequate validation: Failure to identify invalid SpnPlans can lead to unexpected behavior or errors during execution.
- Over-reliance on manual checks: Relying solely on manual verification of SpnPlans can be time-consuming and prone to human error.

A manager should care about Test-SpnPlan when:

- Deploying new SpnPlans: Ensuring that SpnPlans conform to the expected shape before deployment is crucial.
- Identifying issues: When errors or unexpected behavior occur during execution, a manager may need to investigate the cause and determine whether it was due to an invalid SpnPlan.

Practitioner

To use Test-SpnPlan in your daily work:

1. Import the SpnManager module: `Import-Module SpnManager`
2. Create or obtain a valid SpnPlan object.
3. Call Test-SpnPlan with the SpnPlan object as input: `Test-SpnPlan -SpnPlan $spnplan`

The following code example demonstrates how to use Test-SpnPlan:

```
## Import the SpnManager module
Import-Module SpnManager

## Create a valid SpnPlan object
$spnplan = New-Object SpnPlan

## Call Test-SpnPlan with the SpnPlan object as input
$result = Test-SpnPlan -SpnPlan $spnplan
```

if ($result.Status -eq 'Clear') { Write-Host "The SpnPlan is valid." } else { Write-Host "The SpnPlan is invalid: (result.Reason)" }

When using Test-SpnPlan, watch out for the following:

- Ensure that the input SpnPlan object conforms to the expected shape.
- Verify that the ProposedSpns array contains at least one element.

Learner

Test-SpnPlan validates the shape of a SpnPlan object before execution. This ensures that all required fields are present and that the ProposedSpns array is non-empty. To use Test-SpnPlan, follow these simple steps:

1. Create or obtain a valid SpnPlan object.
2. Call Test-SpnPlan with the SpnPlan object as input.

If you encounter issues during execution, check whether the SpnPlan object conforms to the expected shape by running Test-SpnPlan.

Examples

This is the **test** step. It checks a plan before anything acts on it. Its output is normally piped into `Invoke-SpnExecutionEngine`. Providers: AD.ADFS, AD.IIS, AD.SharePoint, SQL.SSAS, SQL.SSRS.

Example 1

```
$plan | Test-SpnPlan
```

Returns [PSCustomObject]@{ Status='Clear'; Reason=$null } when valid.

Example 2

```
Test-SpnPlan -SpnPlan $badPlan
```

Returns [PSCustomObject]@{ Status='Invalid'; Reason='Missing required field: PlanGuid' } when a required field is absent.

Example 3

```
$plan | Test-SpnPlan
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.SSAS, SQL.SSRS, AD.IIS, AD.ADFS, AD.SharePoint
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

Test-SpnPlan handles data in accordance with the following compliance requirements:

- Data handling: Verifies required fields and ProposedSpns array are present.
- Transit: No data transit occurs during execution of Test-SpnPlan.
- At rest: The input SpnPlan object is stored in memory only.
- Audit trail: No audit trail is generated by Test-SpnPlan.

Compliance requirements for Test-SpnPlan include:

- Ensuring that the ProposedSpns array is non-empty to prevent invalid SpnPlans.

Test-SqlSpnKerberosAuth

Manager

Test-SqlSpnKerberosAuth provides visibility into the security protocols used by SQL Server connections. This command is essential for organizations that rely on Kerberos authentication for secure connections.

- Who uses this: Database administrators, security teams, and IT managers who need to ensure Kerberos is being used correctly.
- Risks:
  - Misconfigured SPNs can lead to NTLM usage instead of Kerberos.
  - Local connections may report NTLM even when SPNs are correct.
- When a manager cares:
  - When migrating to Kerberos authentication for improved security.
  - When troubleshooting connection issues.

Practitioner

Day-to-Day Use

1. Connect to the SQL Server instance using integrated security.
2. Run Test-SqlSpnKerberosAuth to verify Kerberos usage.
3. Analyze the result, paying attention to IsLocalCaveat and warning messages.

```
## Example use case:
$result = Test-SqlSpnKerberosAuth -ServerInstance "my_sql_instance"
```

if ($result.IsSucceeded) { Write-Host “Kerberos is being used for this connection.” } else { Write-Warning $result.Reason }

Code Examples

- Verify Kerberos usage on a remote client:

```
$result = Test-SqlSpnKerberosAuth -ServerInstance "my_sql_instance"
```

if (!$result.IsLocalCaveat) { if ($result.IsSucceeded) { Write-Host “Kerberos is being used for this connection.” } else { Write-Warning $result.Reason } } else { Write-Warning “This is a local connection, Kerberos usage may be incorrect.” }

Learner

What This Command Does

1. Connects to the SQL Server instance using integrated security.
2. Queries sys.dm_exec_connections for authentication and transport information.
3. Reports whether Kerberos or NTLM is being used.

Step-by-Step Recipe

1. Ensure you have the SpnManager module installed.
2. Run Test-SqlSpnKerberosAuth with the SQL instance name as an argument.
3. Analyze the result to determine if Kerberos is being used.

Stuck?

- Check that your SPNs are correctly configured and readable.
- Verify that your connection is not local (use a remote client).
- Consult the Microsoft documentation for sys.dm_exec_connections.

Examples

Example 1

```
Test-SqlSpnKerberosAuth -ServerInstance 'SQLSRV01.corp.example.com'
```

Run FROM A REMOTE CLIENT: IsKerberos $true proves the SPN chain works end-to-end; AuthScheme NTLM from a remote client means Kerberos was not negotiated (missing/wrong SPN, or the client fell back).

Example 2

```
Test-SqlSpnKerberosAuth -ServerInstance 'SQLFCI01.corp.example.com'
```

The DR-558/DR-313 exit check for an FCI: after registering Engine SPNs on the service account, a remote run returning AuthScheme KERBEROS is the proof that the fix holds on a live domain.

Example 3

```
Test-SqlSpnKerberosAuth -ServerInstance 'localhost'
```

Returns IsLocalCaveat $true with a warning: local connections use NTLM by design, so this result cannot prove or disprove the SPN configuration.

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access.
Providers covered	framework surface (not provider-specific)
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

This command handles data in accordance with Microsoft's documentation on sys.dm_exec_connections.

- Data-handling: The command queries system views, which is an approved method for gathering information about connections.
- Transit: No sensitive data is transmitted; only authentication and transport details are reported.
- At-rest: No data storage or modification occurs during the execution of this command.
- Audit trail: The command's result can be used to audit Kerberos usage, ensuring compliance with security policies.

Test-SqlSpnPlan

Manager

As a manager, you care about the business value of validating a plan object's shape before execution. This is crucial for preventing costly errors and ensuring that your organization's sensitive information is handled correctly.

- Business value: Validates plan object's structure to prevent errors and data loss.
- Who uses this:
  - IT administrators responsible for setting up and managing SQL Server instances.
  - Developers working on applications that interact with SQL Server.
  - System administrators tasked with maintaining the security and integrity of the system.
- Risks:
  - Incorrectly structured plan objects can lead to data corruption, loss, or security breaches.
  - Failing to validate the plan object's shape can result in unexpected behavior or errors during execution.
- When a manager cares: When planning and executing changes to SQL Server instances that involve setting up or modifying service principal names (SPNs).

Practitioner

To use `Test-SqlSpnPlan`, follow these steps:

1. Import the SpnManager module using `Import-Module`.
2. Create a plan object with the required fields: AccountDn, ProposedSpns, and Role.
3. Call `Test-SqlSpnPlan` with the plan object as input.

```
(plan-object = [PSCustomObject]@{
    AccountDn = 'CN=Account,CN=Users,DC=example,DC=com'
    ProposedSpns = @('MSSQLSvc/account:1433')
    Role = 'SQLSERVERMSSQLSERVICE'
})

$result = Test-SqlSpnPlan -SpnPlan $plan-object
```

- Common patterns:
  - Use `Test-SqlSpnPlan` as a pre-execution check to ensure the plan object is valid.
  - Integrate this cmdlet into your automation scripts or workflows for SQL Server setup and maintenance.
- Watchpoints: Be aware that this cmdlet only performs local validation; it does not account for forest-wide duplicate detection, which should be handled separately.

Learner

`Test-SqlSpnPlan` is a PowerShell cmdlet used to validate the structure of a plan object before execution. Here's how it works:

1. The cmdlet checks if the plan object has the required fields: AccountDn, ProposedSpns, and Role.
2. It ensures that the `ProposedSpns` field contains at least one SPN.
3. If the plan object is valid, the cmdlet returns `$true`. Otherwise, it returns `$false`.

To use this cmdlet:

1. Create a plan object with the required fields.
2. Call `Test-SqlSpnPlan` with the plan object as input.

Here's an example recipe:

- Step 1: Create a new plan object

```
(plan-object = [PSCustomObject]@{
    AccountDn = 'CN=Account,CN=Users,DC=example,DC=com'
    ProposedSpns = @('MSSQLSvc/account:1433')
    Role = 'SQLSERVERMSSQLSERVICE'
})
```

- Step 2: Validate the plan object using `Test-SqlSpnPlan`

```
$result = Test-SqlSpnPlan -SpnPlan $plan-object
```

if ($result) { "The plan object is valid." } else { "The plan object has errors. Please check the output for details." } * What to do when stuck: + Check that your plan object has the required fields and that `ProposedSpns` contains at least one SPN. + Verify that you have correctly imported the SpnManager module.

Examples

This is the **test** step. It checks a plan before anything acts on it. Its output is normally piped into `Invoke-SqlSpnExecutionEngine`. Providers: SQL.Engine.

Example 1

```
$plan | Test-SqlSpnPlan
```

Example 2

```
$result = Test-SqlSpnPlan -SpnPlan $plan
```

if ($result.Status -ne 'Clear') { throw "Plan invalid: (result.Reason)" }

Example 3

```
$plan | Test-SqlSpnPlan
```

Software Approval

Field	Value
Vendor	Detent Point LLC
Product	SpnManager 0.4.0
Licence	Proprietary - licensed, not sold. See LICENSE.
Operational impact	Read-only. Queries Active Directory and reports findings; changes nothing.
Rights required	Directory read access. Some providers additionally need rights to read delegation and encryption-type attributes.
Providers covered	SQL.Engine
Approval recommendation	Approve for general operational use. It cannot alter directory state.

Licence terms are proprietary and are supplied with the purchase, subscription or evaluation agreement. No open-source licence is granted.

Dependencies

What	Why
Windows PowerShell 5.1 or later	Declared by the module manifest.
ActiveDirectory 1.0.0.0	Required module. Ships with RSAT; install the Active Directory PowerShell feature.
A reachable domain controller	Every provider reads from Active Directory.
An account with directory read access	Needed to enumerate accounts and their SPNs.

Compliance

`Test-SqlSpnPlan` handles sensitive information, including service principal names (SPNs). Here's how we ensure compliance:

- Data-handling: The cmdlet only reads and validates the plan object; it does not modify or write any data.
- Transit: The cmdlet operates locally on the system, without transmitting data over the network.
- At-rest: The plan object is stored in memory during execution; it is not persisted to disk unless explicitly saved.
- Audit trail: We do not maintain an audit trail for this cmdlet. However, we recommend using logging and auditing tools within your organization to track changes made by `Test-SqlSpnPlan` (if used in a production environment).
- Compliance requirements:
    - The organization must adhere to its internal security policies regarding the handling of sensitive information.
    - Comply with relevant laws and regulations regarding data protection, such as GDPR or HIPAA.