RemoteApp-Nutzung auswerten: welche Anwendungen laufen wirklich?



WISSEN
Rollen, Aufbau und Betrieb der Remotedesktopdienste im Überblick.
› **Remote Desktop Services**

BERATUNG
Anwendungsbestand bewerten, Farm entrümpeln und Migration planen.
› **Beratung zu Remote Desktop Services**

SCHULUNG
PowerShell, Ereignisprotokolle und Leistungsindikatoren für Terminalserver üben.
› **Schulung Remote Desktop Services**

Jede RDS-Farm hat ein Startmenü, das aussieht wie der Keller nach zwölf Jahren im selben Haus. Irgendwo zwischen ERP-Client und Office stehen ein PDF-Werkzeug von 2014, ein „Testapp_alt“ und eine Zeiterfassung, die angeblich keiner mehr benutzt, die aber auch niemand abschalten will, weil „die Buchhaltung die vielleicht noch braucht“. Wenn du die RemoteApp-Nutzung auswerten willst, geht es um genau diese Frage: Welche veröffentlichten Anwendungen laufen auf deinen Session Hosts tatsächlich, wie oft, wie lange, auf welchen Hosts und mit welchem Ressourcenhunger? Und welche warten seit Jahren geduldig auf einen Benutzer, der nie kommt?

Dieser Beitrag zeigt dir, wie du das mit Bordmitteln herausfindest: mit Get-RDRemoteApp für den Soll-Zustand, mit Prozess-Auditing (Ereignis 4688) für die Starts, mit CIM-Momentaufnahmen für die Laufzeit und mit Leistungsindikatoren je Prozess für den Ressourcenverbrauch. Und er zeigt dir, wo die Grenze verläuft: Du wertest je Anwendung und je Host aus, nicht je Person. Die Grundlagen zu Sammlungen, Connection Broker und Session Host (RDSH) findest du in der Übersicht zu Remote Desktop Services. In den Beispielen heißt die Domäne contoso.local, der Broker RDCB01, die Session Hosts RDSH01 bis RDSH06 verteilen sich auf die Sammlungen „Office-Desktop“ und „ERP-RemoteApps“ der Standorte Frankfurt, Köln und Hamburg.

FAKTEN · Das Wichtigste in Kürze

Get-RDRemoteApp aus dem Modul RemoteDesktop zeigt dir, was veröffentlicht ist. Ob es jemand benutzt, zeigt es dir nicht. Dafür gibt es in den Remotedesktopdiensten keine eingebaute Nutzungsstatistik.

Die Starts liefert das Prozess-Auditing: Ist die Überwachung der Prozesserstellung aktiv, schreibt Windows für jeden neuen Prozess das Ereignis 4688 ins Sicherheitsprotokoll. Standardmäßig ist das nicht der Fall.

Die Befehlszeile im Ereignis 4688 ist eine separate, standardmäßig deaktivierte Einstellung. Für die Nutzungsauswertung brauchst du sie nicht, und aus Datenschutzsicht willst du sie auch nicht.

Laufzeit und Ressourcenverbrauch misst du mit CIM (Win32_Process) und den Leistungsindikatoren Process(*) sowie User Input Delay per Process, Letztere ab Windows Server 2019.

Die Rohdaten enthalten Benutzerkonten. Werte je Anwendung und Host aus, nicht je Person, und beteilige Datenschutz und Betriebsrat, bevor du das Auditing einschaltest.

Messfenster: mindestens ein voller Geschäftszyklus, also vier bis sechs Wochen inklusive Monatsabschluss.

Ausgangslage: viele RemoteApps, wenig Gewissheit

Der Anlass ist selten Neugier. Meistens steht eine der folgenden Fragen im Raum, und zwar mit Termin: Der Softwarehersteller kündigt ein Lizenzaudit an und will wissen, auf wie vielen Servern seine Anwendung installiert ist. Die Session Hosts der Sammlung „ERP-RemoteApps“ laufen am Monatsende am Anschlag, und niemand weiß, welche Anwendung den Arbeitsspeicher frisst. Oder die Migration auf Windows Server 2025 steht an, und die Liste der Anwendungen, die mitumziehen müssen, ist 47 Einträge lang, von denen gefühlt zwölf benutzt werden.

Aus Sicht der Anwender ist die Lage übersichtlicher: Sie sehen im Web-Feed oder im Startmenü eine lange Liste von Symbolen, klicken die drei an, die sie kennen, und ignorieren den Rest mit der Gelassenheit langjähriger Übung. Fragst du sie, ob sie Anwendung X noch brauchen, antworten sie zuverlässig mit „Ja, auf keinen Fall abschalten“. Das ist keine Lüge, sondern Risikovermeidung. Deshalb gilt: Messen schlägt Umfrage.



Skizze: Fragestellungen und passende Kennzahlen der Anwendungsnutzung

Anlass	Was du wissen willst	Kennzahl	Bordmittel
Lizenzaudit des Herstellers	Wo ist die Anwendung installiert, wo läuft sie, wie viele gleichzeitig?	Hosts, Spitze gleichzeitiger Instanzen	CIM Win32_Process, Get-RDAvailableApp

Kapazitätsengpass	Welche Anwendung verbraucht CPU und Arbeitsspeicher?	Mittelwert und Spitze je Anwendung	Leistungsindikatoren Process(*)
Aufräumen	Was ist veröffentlicht, aber ungenutzt?	Starts im Messfenster	Get-RDRemoteApp, Ereignis 4688
Migration	Was muss auf die neuen Hosts, in welcher Version?	Anwendung, Version, Hosts, Nutzung	Get-RDRemoteApp, CIM, Ereignis 4688
Beschwerden über Trägheit	Welche Anwendung reagiert verzögert?	Eingabeverzögerung je Prozess	User Input Delay per Process

Beim Lizenzthema solltest du sauber trennen: Die RDS-CALs hängen an Benutzern oder Geräten, nicht an Anwendungen. Für die Lizenzen der Anwendungen selbst gelten die Bedingungen des jeweiligen Herstellers, und die zählen oft je Installation, je Gerät oder je gleichzeitigem Benutzer. Wie das bei Microsoft 365 Apps und Office auf dem Session Host aussieht, beschreibt Office auf dem Terminalserver richtig lizenzieren.

Hintergrund: warum RDS die RemoteApp-Nutzung nicht von selbst auswertet

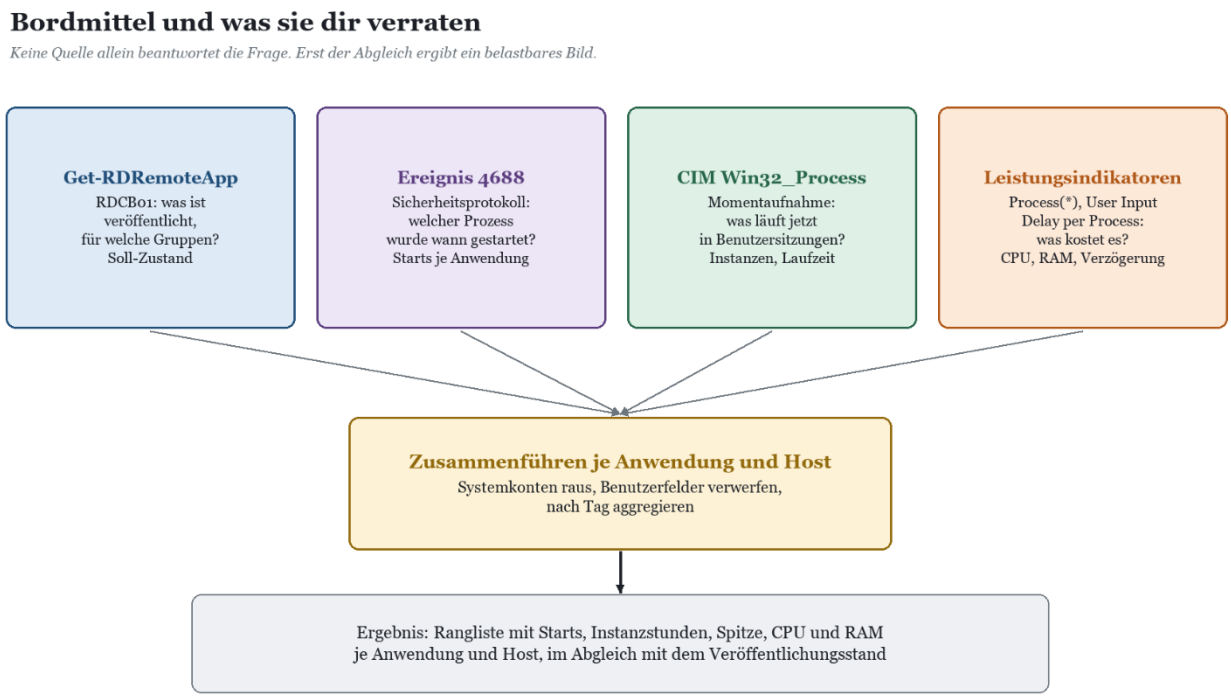
Eine RemoteApp ist aus Sicht der Remotedesktopdienste keine eigene Instanz einer Anwendung, sondern eine Veröffentlichung: ein Eintrag in der Sammlung auf dem Connection Broker mit Pfad, Anzeigename, Symbol, optionalen Befehlszeilenparametern und Benutzergruppen. Startet ein Benutzer die RemoteApp, baut der Client eine normale RDP-Sitzung zum Session Host auf, nur ohne Desktop. Statt des Explorers läuft in der Sitzung die RemoteApp-Shell, und die startet das veröffentlichte Programm als ganz gewöhnlichen Prozess.

Daraus folgen drei Dinge, die dir die Auswertung erschweren. Erstens protokolliert der Connection Broker Verbindungen und Sitzungen, aber keine Programmstarts. Zweitens kann ein Benutzer aus einer RemoteApp heraus weitere Programme starten, etwa einen Dateidialog, einen PDF-Betrachter oder den Browser über einen Link. Die laufen dann in derselben Sitzung, ohne je veröffentlicht worden zu sein. Und drittens öffnen mehrere RemoteApps desselben Benutzers auf demselben Host in der Regel keine zweite Sitzung, sondern teilen sich eine. Sitzungen zählen ist also nicht Anwendungen zählen.

WICHTIG · Veröffentlicht, installiert, gestartet: drei verschiedene Listen

Veröffentlicht ist, was Get-RDRemoteApp auf dem Broker zurückgibt. Installiert ist, was auf den Session Hosts liegt, und das muss nicht auf jedem Host dasselbe sein. Gestartet ist, was im Sicherheitsprotokoll oder in der Prozessliste auftaucht. Die interessanten Befunde liegen in den Differenzen: veröffentlicht, aber nie gestartet; gestartet, aber nicht veröffentlicht; veröffentlicht, aber nur auf einem Teil der Hosts installiert. Letzteres führt zu dem Klassiker, dass eine Anwendung „manchmal“ nicht startet, nämlich immer dann, wenn der Broker den Benutzer auf den falschen Host schickt.

Welche Bordmittel was leisten



Skizze: Datenquellen für die RemoteApp-Auswertung mit Bordmitteln, Contoso-Beispiel

Bordmittel	Liefert	Liefert nicht	Aufwand
Get-RDRemoteApp	Veröffentlichte Programme je Sammlung mit Pfad und Benutzergruppen	Jede Aussage zur Nutzung	gering, nur lesend
Get-RDAvailableApp	Programme aus dem Startmenü eines Hosts, die veröffentlicht werden könnten	Programme ohne Startmenüeintrag, Nutzung	gering
Ereignis 4688 (Prozesserstellung)	Jeden Prozessstart mit Pfad, Zeitpunkt, Konto	Laufzeit, Ressourcen; ohne Audit gar nichts	mittel: GPO, Protokollgröße, Auswertung
Ereignis 4689 (Prozessbeendigung)	Ende eines Prozesses	zuverlässige Laufzeit, wenn Ereignisse überschrieben wurden	hoch: Start und Ende paaren
CIM Win32_Process	Was jetzt läuft, je Sitzung, mit Arbeitsspeicher	Vergangenheit; nur Momentaufnahme	gering, als geplante Aufgabe
Leistungsindikatoren Process(*)	CPU-Zeit und privater Arbeitsspeicher je Prozess	Zuordnung zur Veröffentlichung	mittel: Datensammler
User Input Delay per Process	Eingabeverzögerung je Prozess, ab Windows Server 2019	Prozesse unter SYSTEM	mittel

RemoteApp-Nutzung auswerten: Diagnose Schritt für Schritt

Die Reihenfolge ist wie immer: erst lesen, dann ändern. Die Schritte 1 bis 4 sind rein lesend und laufen von einer Verwaltungsstation mit den RSAT-Modulen RemoteDesktop und GroupPolicy. Für die Session Hosts nutzt du CIM und Get-WinEvent über das Netzwerk, also WinRM bzw. Remote-Eventlog.

Schritt 1: Soll-Zustand vom Connection Broker holen

Im Server-Manager findest du die veröffentlichten Programme unter Remotedesktopdienste › Sammlungen › deine Sammlung › RemoteApp-Programme. Für eine Auswertung brauchst du sie aber als Liste, und zwar mit Dateipfad, denn der Pfad ist das, was du später im Sicherheitsprotokoll wiederfindest. Der Anzeigename „Buchhaltung“ taucht dort nicht auf, der Prozess fibu32.exe schon.

```
Import-Module RemoteDesktop
$cb = 'RDCB01.contoso.local'

# Sammlungen und ihre Session Hosts
Get-RDSessionCollection -ConnectionBroker $cb
Get-RDSessionHost -CollectionName 'ERP-RemoteApps' -ConnectionBroker $cb

# Veröffentlichte RemoteApps mit Pfad und Benutzergruppen
$soll = Get-RDRemoteApp -ConnectionBroker $cb |
  Select-Object CollectionName, Alias, DisplayName, FilePath, UserGroups, ShowInWebAccess
$soll | Sort-Object CollectionName, DisplayName | Format-Table -AutoSize

# Sicherung für den Rückweg: vollständige Konfiguration ablegen
Get-RDRemoteApp -ConnectionBroker $cb | Export-Clixml C:\RDS\remoteapps-$(Get-Date -f yyyyMMdd).xml
```

Gleich dazu der zweite Abgleich: Get-RDAvailableApp zeigt dir, welche Programme auf den Hosts einer Sammlung über das Startmenü verfügbar, aber nicht unbedingt veröffentlicht sind. Das ist die Liste für die Frage „Was ist installiert, ohne dass es jemand bestellt hat?“.

```
Get-RDAvailableApp -CollectionName 'ERP-RemoteApps' -ConnectionBroker $cb |
  Select-Object DisplayName, FilePath | Sort-Object DisplayName
```

Schritt 2: Momentaufnahme, was gerade läuft

Bevor du irgendein Auditing einschaltest, lohnt ein Blick auf den Ist-Zustand. Über CIM liest du die Prozesse aller Hosts einer Sammlung aus, beschränkt auf Benutzersitzungen (Sitzungs-ID größer als 0, die Sitzung 0 gehört den Diensten). Das Ergebnis gruppierst du je Host und Prozessname, die Benutzerkonten fragst du gar nicht erst ab.

```
$hosts = (Get-RDSessionHost -CollectionName 'ERP-RemoteApps' -ConnectionBroker $cb).SessionHost

Get-CimInstance -ClassName Win32_Process -ComputerName $hosts -Filter 'SessionId > 0' |
  Group-Object PSComputerName, Name |
  Select-Object Count, Name,
    @{n='RAM_MB'; e={ [math]::Round(($_.Group | Measure-Object WorkingSetSize -Sum).Sum / 1MB) }} |
  Sort-Object RAM_MB -Descending | Select-Object -First 25
```

Läuft das als geplante Aufgabe alle 15 Minuten und schreibst du die Zählung in eine CSV-Datei, bekommst du ohne jedes Auditing eine brauchbare Näherung für Instanzstunden: Anzahl Instanzen mal Intervall. Kurze Starts, die zwischen zwei Messpunkten beginnen und enden, entgehen dir dabei. Für die Frage „Läuft das überhaupt?“ reicht das oft schon, für „Wie oft wird es gestartet?“ nicht.

Schritt 3: Prüfen, ob Prozess-Auditing schon aktiv ist

Viele Umgebungen haben die Überwachung der Prozesserstellung bereits, etwa aus einer Sicherheits-Baseline. Dann liegen die Daten schon im Sicherheitsprotokoll, und die Frage ist nur, wie weit es zurückreicht. Prüfe zuerst die effektive Einstellung und die Protokollgröße. Die Unterkategorie fragst du über ihre GUID ab, dann funktioniert der Befehl auf deutschen und englischen Systemen gleichermaßen.

```
# Effektive Überwachungsrichtlinie: Prozesserstellung (Detaillierte Nachverfolgung)
Invoke-Command -ComputerName $hosts -ScriptBlock {
  auditpol /get /subcategory:'{0CCE922B-69AE-11D9-BED3-505054503030}'
}

# Größe und Reichweite des Sicherheitsprotokolls
foreach ($h in $hosts) {
  $log = Get-WinEvent -ListLog Security -ComputerName $h
  $old = Get-WinEvent -LogName Security -ComputerName $h -Oldest -MaxEvents 1
  [pscustomobject]@{ Host = $h; MaxMB = [math]::Round($log.MaximumSizeInBytes / 1MB)
                     Anzahl = $log.RecordCount; Ältester = $old.TimeCreated }
}

# Welche GPOs wirken auf die OU der Session Hosts?
Get-GPInheritance -Target 'OU=SessionHosts,OU=RDS,DC=contoso,DC=local' |
  Select-Object -ExpandProperty InheritedGpoLinks | Select-Object DisplayName, Enabled, Order
```

Steht bei der Prozesserstellung „Keine Überwachung“, hast du keine historischen Starts und musst mit Schritt 5 erst messen. Reicht das Sicherheitsprotokoll nur drei Tage zurück, ist das kein Messfenster, sondern ein Schnappschuss. Auf einem gut besuchten Session Host erzeugt jede Anmeldung schon Dutzende Prozesse, die Protokolle laufen entsprechend schnell voll.

Schritt 4: Starts aus dem Sicherheitsprotokoll zählen

Liegen Ereignisse 4688 vor, zählst du sie je Programm. Das folgende Skript liest die letzten sieben Tage, wirft Prozesse der Systemkonten (SYSTEM, LOKALER DIENST, NETZWERKDIENST) heraus und behält vom Ereignis nur den Programmnamen. Benutzername und Konto-SID landen nirgends in der Ausgabe. Das ist kein Zufall, sondern Absicht.

```
$start  = (Get-Date).AddDays(-7)
$system = 'S-1-5-18', 'S-1-5-19', 'S-1-5-20'

$starts = foreach ($h in $hosts) {
  Get-WinEvent -ComputerName $h -FilterHashtable @{ LogName = 'Security'; Id = 4688; StartTime =
$start } |
    ForEach-Object {
      $d = @{}; ([xml]$_.ToXml()).Event.EventData.Data | ForEach-Object { $d[$_.Name] = $_.'#text' }
      if ($d['SubjectUserSid'] -notin $system) {
        [pscustomobject]@{ Host = $h; Tag = $_.TimeCreated.Date
                           Programm = (Split-Path $d['NewProcessName'] -Leaf).ToLower() }
      }
    }
}

$starts | Group-Object Programm | Sort-Object Count -Descending |
  Select-Object Count, Name, @{n='Hosts'; e={ ($_.Group.Host | Sort-Object -Unique) -join ', ' }} |
  Select-Object -First 40
```

Im Ergebnis siehst du neben deinen Fachanwendungen erwartbar viel Grundrauschen: Anmeldeprozesse, conhost.exe, Hilfsprozesse von Office und Browsern. Den Abgleich mit dem Soll machst du über den Dateinamen aus FilePath aus Schritt 1. Was in $soll steht und in $starts gar nicht vorkommt, ist dein erster Kandidat für das Aufräumen. Bei sehr großen Protokollen ist Get-WinEvent mit XML-Auswertung langsam; dann lässt du das Skript besser lokal auf jedem Host per Invoke-Command laufen und holst nur die verdichtete Zählung ab.

TIPP · Dateiname statt Anzeigename abgleichen

Verknüpfe Soll und Ist über den Dateinamen: (Split-Path $_.FilePath -Leaf).ToLower(). So findest du auch Fälle, in denen zwei RemoteApps auf dieselbe EXE zeigen, nur mit unterschiedlichen Befehlszeilenparametern, etwa „ERP Frankfurt“ und „ERP Köln“. Ohne Befehlszeile im Ereignis kannst du die beiden nicht trennen. Das ist der Preis der Datensparsamkeit, und meistens ist er es

wert.

Schritt 5: Leistungsindikatoren je Prozess

Für die Kapazitätsfrage reichen Starts nicht. Hier helfen die Leistungsindikatoren der Klasse Process: Prozessorzeit und privater Arbeitsspeicher je Prozessinstanz. Ab Windows Server 2019 kommt der Indikator User Input Delay per Process hinzu, der die längste Wartezeit zwischen Benutzereingabe und deren Verarbeitung durch die Anwendung im Messintervall angibt. Microsoft beschreibt das in Leistungsindikatoren zur Diagnose von Reaktionsproblemen auf Session Hosts (Microsoft Learn). Prozesse, die unter SYSTEM laufen, erscheinen dort nicht, und das ist hier sogar praktisch.

```
$c = '\Process(*)\% Processor Time', '\Process(*)\Working Set - Private'
$s = Get-Counter -ComputerName RDSH04 -Counter $c -SampleInterval 60 -MaxSamples 30

$s.CounterSamples | ForEach-Object {
  [pscustomobject]@{ Prozess = ($_.InstanceName -replace '#\d+$', ''); Pfad = $_.Path; Wert =
$_.CookedValue }
} | Where-Object Prozess -notin '_total', 'idle' |
  Group-Object Prozess, { ($_.Pfad -split '\\')[-1] } |
  Select-Object Name, @{n='Mittel'; e={ [math]::Round(($_.Group.Wert | Measure-Object -
Average).Average, 1) }} |
  Sort-Object Mittel -Descending | Select-Object -First 30
```

WARNUNG · Zwei Fallen bei den Leistungsindikatoren

Erstens: Get-Counter erwartet auf einem deutschsprachigen Windows unter Umständen die lokalisierten Indikatornamen. Liefert der Befehl „Der angegebene Indikatorpfad wurde nicht gefunden“, nimm die Namen, die dir Get-Counter -ListSet Pro* auf dem Zielsystem anzeigt.

Zweitens: % Processor Time je Prozess bezieht sich auf einen logischen Prozessor. Auf einem Host mit 16 logischen Prozessoren kann ein Prozess also 400 Prozent anzeigen, ohne dass der Server brennt. Teile durch die Anzahl der logischen Prozessoren, bevor du jemanden alarmierst.

Wenn du die Messung grundsätzlich angehen willst, also mit Datensammlersätzen, sinnvollen Intervallen und Schwellenwerten für die ganze Farm, findest du das ausführlich in Terminalserver langsam: Leistung richtig messen statt raten.

Lösung: Nutzung messen, auswerten, aufräumen

Liefert die Diagnose zu wenig, weil das Auditing fehlt oder das Protokoll zu kurz ist, baust du eine saubere Messung auf. Die folgenden Schritte ändern etwas, deshalb steht bei jedem der Rückweg dabei.



Skizze: Zeitlicher Ablauf einer RemoteApp-Auswertung mit Rückweg

Schritt 1: Zweck festlegen und Mitbestimmung klären

Klingt nach Formalie, ist aber der Schritt, der über den Rest entscheidet. Schreib in zwei Sätzen auf, wofür du misst, etwa „Bereinigung der veröffentlichten RemoteApps und Kapazitätsplanung der Sammlung ERP-RemoteApps vor der Migration“, wie lange, welche Daten du erhebst, wie du sie verdichtest und wann du sie löschst. Damit gehst du zum Datenschutz und, wo vorhanden, zum Betriebsrat, bevor die erste GPO verknüpft ist. Die Details stehen weiter unten im Abschnitt zum Datenschutz.

Schritt 2: Prozess-Auditing per Gruppenrichtlinie aktivieren

Lege eine eigene GPO an, statt eine bestehende Sicherheits-GPO zu erweitern. Dann kannst du sie nach dem Messfenster mit einem Handgriff wieder abschalten. In der Gruppenrichtlinienverwaltung setzt du unter Computerkonfiguration › Richtlinien › Windows-Einstellungen › Sicherheitseinstellungen › Erweiterte Überwachungsrichtlinienkonfiguration › Überwachungsrichtlinien › Detaillierte Nachverfolgung die Einstellung „Prozesserstellung überwachen“ auf „Erfolg“. Die Einstellung zur Befehlszeile unter Administrative Vorlagen › System › Überwachung: Prozesserstellung lässt du ausdrücklich deaktiviert. Befehlszeilen enthalten Dateinamen, Pfade in Benutzerprofilen und im schlimmsten Fall Kennwörter, die irgendein Skript als Parameter übergibt.

```
Import-Module GroupPolicy
$gpo = New-GPO -Name 'RDS - Messung Anwendungsnutzung' -Domain contoso.local `
  -Comment 'Prozesserstellung überwachen, ohne Befehlszeile. Messfenster bis 31.01.2027.'

# Erweiterte Überwachung und Protokollgröße im GPO-Editor setzen (siehe Text), dann verknüpfen:
New-GPLink -Guid $gpo.Id -Target 'OU=SessionHosts,OU=RDS,DC=contoso,DC=local' -LinkEnabled Yes

# Kontrolle auf einem Host nach gpupdate
Invoke-Command -ComputerName RDSH01 { auditpol /get /subcategory:'{0CCE922B-69AE-11D9-BED3-
505054503030}' }
```

Im selben GPO erhöhst du unter Computerkonfiguration › Richtlinien › Administrative Vorlagen › Windows-Komponenten › Ereignisprotokolldienst › Sicherheit die maximale Protokollgröße, damit mindestens die Spanne zwischen zwei Auswertungsläufen hineinpasst. Wie groß, hängt von der Zahl der Anmeldungen ab: Miss nach zwei Tagen, wie viele Ereignisse 4688 pro Tag anfallen, und rechne hoch. Rückweg: Set-GPLink -Guid $gpo.Id -Target '<OU>' -LinkEnabled No, danach

gpupdate. Die Überwachung endet mit der nächsten Richtlinienaktualisierung, bereits geschriebene Ereignisse bleiben, bis das Protokoll sie überschreibt.

WARNUNG · Erweiterte Überwachung und klassische Überwachung nicht mischen

Sobald eine GPO Unterkategorien der erweiterten Überwachungsrichtlinie setzt, ignoriert Windows die klassischen Überwachungseinstellungen unter Lokale Richtlinien › Überwachungsrichtlinie, sofern die Sicherheitsoption „Überwachung: Unterkategorieeinstellungen der Überwachungsrichtlinie erzwingen“ aktiv ist; in aktuellen Windows-Versionen ist sie standardmäßig wirksam. Gibt es bei dir noch klassische Einstellungen, prüfe vorher mit auditpol /get /category:*, was danach tatsächlich gilt. Sonst misst du Anwendungen und verlierst nebenbei die Anmeldeüberwachung, die die Revision so gern liest.

Schritt 3: Täglich verdichten, Rohdaten nicht sammeln

Statt Ereignisse wochenlang in einen zentralen Speicher zu kippen, verdichtest du einmal täglich auf jedem Host: Programm, Host, Tag, Anzahl Starts. Dazu die CIM-Momentaufnahmen aus Schritt 2 der Diagnose für die Instanzstunden. Die Ausgabe ist eine kleine CSV-Datei ohne Personenbezug. Läuft das als geplante Aufgabe unter einem Dienstkonto mit Leserechten auf das Sicherheitsprotokoll, etwa über die Gruppe „Ereignisprotokollleser“, brauchst du dafür keine Administratorrechte auf den Hosts.

```
# Auszug: Verdichtung des Vortags auf einem Host, Ausgabe ohne Benutzerbezug
$von = (Get-Date).Date.AddDays(-1); $bis = (Get-Date).Date
Get-WinEvent -FilterHashtable @{ LogName = 'Security'; Id = 4688; StartTime = $von; EndTime = $bis }
|
  ForEach-Object { $d = @{}; ([xml]$_.ToXml()).Event.EventData.Data | ForEach-Object { $d[$_.Name] =
$_.'#text' }
    if ($d['SubjectUserSid'] -notin 'S-1-5-18','S-1-5-19','S-1-5-20') { (Split-Path
$d['NewProcessName'] -Leaf).ToLower() } } |
  Group-Object | Select-Object @{n='Tag';e={$von.ToString('yyyy-MM-dd')}},
@{n='Host';e={$env:COMPUTERNAME}},
    @{n='Programm';e={$_.Name}}, @{n='Starts';e={$_.Count}} |
  Export-Csv \\FS01.contoso.local\RDS-Messung$\starts.csv -Append -NoTypeInformation -Encoding UTF8
```

Schritt 4: Auswerten und mit den Fachbereichen abstimmen

Nach dem Messfenster legst du Soll und Ist nebeneinander. Daraus ergibt sich eine Tabelle mit vier Kategorien, die du mit den verantwortlichen Fachbereichen durchgehst. Wichtig ist der Tonfall: „Die Anwendung wurde in sechs Wochen nicht gestartet“ ist eine Feststellung. „Ihr benutzt das ja eh nicht“ ist der Anfang eines langen Meetings.

Befund	Beispiel Contoso	Empfehlung
Veröffentlicht, nie gestartet	PDF-Tool 2014 in ERP-RemoteApps, 0 Starts in 6 Wochen	ausblenden, nach 4 Wochen entfernen, dann deinstallieren
Veröffentlicht, selten gestartet	CAD-Viewer, 2 Starts in 6 Wochen	Benutzergruppe einschränken, Bedarf mit Fachbereich klären
Veröffentlicht, nur auf einem Teil der Hosts installiert	Zeiterfassung auf RDSH01 bis RDSH03, nicht auf RDSH04 bis RDSH06	auf allen Hosts der Sammlung angleichen oder eigene Sammlung
Gestartet, aber nicht veröffentlicht	Hilfsprogramm, das aus dem ERP-Client heraus startet	dokumentieren, bei Migration mitnehmen
Installiert, weder veröffentlicht noch gestartet	Testapp_alt auf RDSH02	deinstallieren, Image bereinigen

Hoher Ressourcenbedarf	ERP-Client mit 1,2 GB privatem Arbeitsspeicher je Instanz	in Kapazitätsplanung und Lasttest einrechnen

Der Befund „nur auf einem Teil der Hosts installiert“ ist streng genommen keine Nutzungsfrage, sondern Konfigurationsdrift. Wie du solche Abweichungen innerhalb einer Sammlung systematisch findest, beschreibt Session Hosts verhalten sich unterschiedlich: Konfigurationsdrift erkennen. Und die gemessenen Ressourcenwerte gehören in die Kapazitätsplanung, dazu mehr in Wie viele Benutzer pro Terminalserver? Lasttest statt Daumenregel.

Schritt 5: Ausblenden vor Entfernen

Der sanfteste Weg, eine vermutlich tote Anwendung zu beerdigen, ist der Schrei-Test: Du blendest sie im Web-Feed aus oder schränkst die Benutzergruppe auf eine leere Gruppe ein und wartest, ob sich jemand meldet. Meldet sich niemand, entfernst du sie. Meldet sich jemand, hast du die Anwendung in Minuten zurück und einen Ansprechpartner gewonnen, den du vorher nicht kanntest.

```
# Ausblenden (Rückweg: -ShowInWebAccess $true)
Set-RDRemoteApp -CollectionName 'ERP-RemoteApps' -Alias 'pdftool2014' -ShowInWebAccess $false -
ConnectionBroker $cb

# Alternativ: nur noch für eine Gruppe sichtbar (Rückweg: alte UserGroups aus der Sicherung setzen)
Set-RDRemoteApp -CollectionName 'ERP-RemoteApps' -Alias 'cadviewer' -UserGroups 'CONTOSO\APP-CAD-
Viewer' -ConnectionBroker $cb

# Nach der Karenzzeit endgültig entfernen
Remove-RDRemoteApp -CollectionName 'ERP-RemoteApps' -Alias 'pdftool2014' -ConnectionBroker $cb -
Force
```

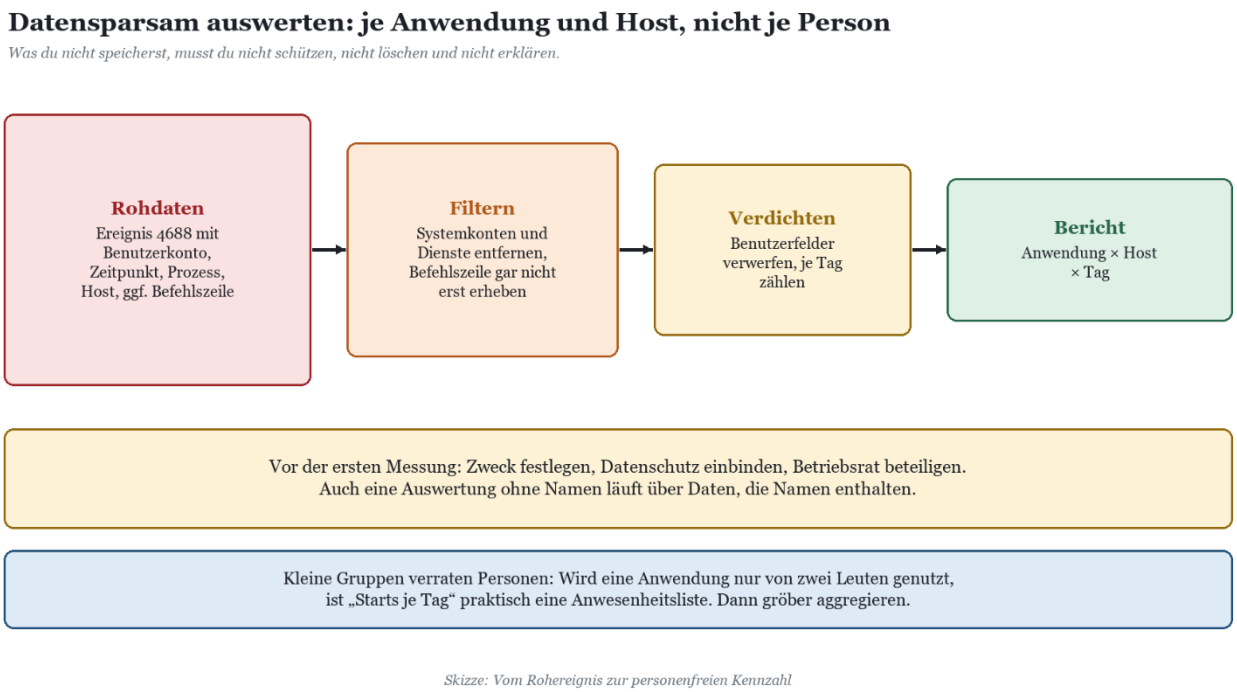
Rückweg nach dem Entfernen: Aus der in Schritt 1 der Diagnose angelegten Clixml-Sicherung liest du die alten Werte und legst die RemoteApp mit New-RDRemoteApp und denselben Werten für CollectionName, Alias, DisplayName, FilePath, CommandLineSetting, RequiredCommandLine, UserGroups und ShowInWebAccess neu an. Deinstallieren auf den Hosts solltest du erst, wenn die Karenzzeit vorbei ist, und dann auf allen Hosts der Sammlung gleichzeitig, sonst erzeugst du genau die Drift, die du gerade beseitigt hast.

AUS DER PRAXIS · Ein Maschinenbauer mit 300 Benutzern und 61 RemoteApps

Vor der Migration auf Windows Server 2025 hatte die Sammlung für die Konstruktion 61 veröffentlichte Programme. Nach sechs Wochen Messung inklusive Quartalsabschluss waren 23 davon nie gestartet worden, weitere neun weniger als fünfmal. Nach dem Schrei-Test meldeten sich genau zwei Personen, beide wegen derselben Anwendung, die dann eine eigene Benutzergruppe bekam. Auf die neuen Session Hosts zogen 31 Programme um. Das Testen der Migration wurde dadurch nicht nur kürzer, sondern überhaupt erst planbar.

Grenzen und Datenschutz: je Anwendung und Host, nicht je Person

Jetzt der Teil, bei dem Admins gern die Augen verdrehen und der trotzdem über das Gelingen entscheidet. Das Ereignis 4688 enthält das Konto, unter dem ein Prozess gestartet wurde. Wer die Ereignisse vollständig auswertet, weiß also, wer wann welche Anwendung geöffnet hat. Das ist eine Verhaltens- und unter Umständen eine Leistungskontrolle, ob du das beabsichtigst oder nicht.



Skizze: Vom Rohereignis zur personenfreien Kennzahl

Was du technisch tun kannst

- **Personenbezug gar nicht erst auswerten.** Die Skripte in diesem Beitrag lesen das Konto nur, um Systemkonten herauszufiltern, und geben es nicht aus. Ausgewertet wird je Anwendung, Host und Tag.
- **Keine Befehlszeile erheben.** Die Einstellung zur Befehlszeile in Prozesserstellungsereignissen bleibt aus. Microsoft weist selbst darauf hin, dass Befehlszeilen sensible Informationen wie Kennwörter enthalten können und jeder mit Leserecht auf das Sicherheitsprotokoll sie sieht.
- **Kleine Gruppen vergröbern.** Nutzen nur zwei Personen eine Anwendung, sind „Starts je Tag“ praktisch eine Anwesenheitsliste. Dann wertest du je Woche oder nur „genutzt ja/nein“ aus.
- **Rohdaten nicht sammeln, Aggregate befristen.** Das Sicherheitsprotokoll auf den Hosts überschreibt sich ohnehin. Die verdichteten CSV-Dateien löschst du nach Abschluss des Projekts, und den Termin dafür schreibst du vorher auf.
- **Zugriff begrenzen.** Wer das Sicherheitsprotokoll lesen darf, sieht trotzdem alle Details. Halte den Kreis klein und dokumentiert.

Was du organisatorisch klären musst

Nach deutschem Betriebsverfassungsrecht hat der Betriebsrat bei der Einführung und Anwendung technischer Einrichtungen mitzubestimmen, die dazu bestimmt sind, Verhalten oder Leistung der Beschäftigten zu überwachen (§ 87 Abs. 1 Nr. 6 BetrVG). In der Rechtsprechung genügt dafür bereits die objektive Eignung zur Überwachung, auf deine gute Absicht kommt es nicht an. Prozess-Auditing auf einem Terminalserver ist objektiv geeignet. Hinzu kommt die DSGVO mit Zweckbindung, Datenminimierung und Speicherbegrenzung. Das hier ist keine Rechtsberatung, aber eine klare Empfehlung: Rede vorher, nicht nachher. Eine kurze Vereinbarung mit Zweck, Datenumfang, Aggregationsstufe, Speicherdauer und Zugriffskreis ist schnell geschrieben und erspart dir die Diskussion, warum das Projekt gestoppt wurde.

Dieselben Fragen stellen sich bei jeder Sitzungsübersicht. Wie du angemeldete Benutzer anzeigst, ohne versehentlich eine Anwesenheitskontrolle zu bauen, beschreibt Angemeldete Benutzer auf dem Terminalserver anzeigen, mit Blick auf den Datenschutz.

WICHTIG · Auditing für die Sicherheit ist ein anderer Zweck

Hast du die Prozesserstellung ohnehin für die Angriffserkennung aktiv, etwa für ein SIEM, ist der Zweck dort die Sicherheit. Die Daten nicht einfach nebenbei für die Nutzungsauswertung mitverwenden: Die Zweckbindung gilt auch für Daten, die schon da sind. Entweder der neue Zweck ist mit abgedeckt, oder du klärst ihn, wie oben beschrieben.

Wo die Bordmittel an Grenzen stoßen

Grenze	Auswirkung	Umgang damit
Sicherheitsprotokoll läuft über	Lücken im Messfenster	Protokollgröße erhöhen, täglich verdichten
Kein Anzeigename im Ereignis	Zwei RemoteApps mit derselben EXE nicht unterscheidbar	über Dateinamen gruppieren, Unterschied fachlich klären
Momentaufnahmen per CIM	Kurze Starts zwischen zwei Messpunkten fehlen	mit Ereignis 4688 ergänzen
Prozesse ohne Veröffentlichung	Hilfsprozesse sehen aus wie eigene Anwendungen	Liste der Hilfsprozesse je Fachanwendung pflegen
Leistungsindikatoren lokalisiert	Skripte brechen auf deutschen Systemen	Namen mit Get-Counter -ListSet auf dem Ziel ermitteln
Viele Hosts, viele Ereignisse	Get-WinEvent über das Netz wird langsam	lokal per Invoke-Command auswerten, nur Aggregate übertragen

Vorbeugung: damit der Keller nicht wieder vollläuft

Aufgeräumt ist schnell, aufgeräumt bleiben ist die eigentliche Kunst. Drei Gewohnheiten helfen.

- **Jede RemoteApp hat einen Verantwortlichen.** Beim Veröffentlichen wird ein Fachbereich und eine Ansprechperson notiert, am besten in der Farmdokumentation. Was in eine gute Dokumentation gehört, steht in RDS-Farm dokumentieren: was in eine gute Farmdokumentation gehört.
- **Benutzergruppen statt „Domänen-Benutzer“.** Wer eine RemoteApp nur für die Gruppe freigibt, die sie braucht, sieht beim nächsten Mal sofort, wen er fragen muss. Und das Startmenü aller anderen bleibt übersichtlich.
- **Regelmäßig wiederholen.** Einmal im Jahr und vor jeder Migration eine Messung mit demselben Verfahren und denselben Datenschutzregeln. Am besten als Teil des regelmäßigen Farm-Checks, wie ihn RDS Health Check: Farm regelmäßig prüfen statt nur reagieren beschreibt.
- **Migration als Anlass nutzen.** Windows Server 2016 erhält nach dem 12.01.2027 keine Sicherheitsupdates mehr. Jede Anwendung, die du vorher entfernst, musst du auf Windows Server 2025 nicht testen. Die Reihenfolge der Migration beschreibt RDS-Migration von Windows Server 2016 auf 2025: die richtige Reihenfolge.

TIPP · Die Messung nach dem Projekt wieder abbauen

Die GPO „RDS - Messung Anwendungsnutzung“ hat ein Ablaufdatum im Kommentar. Halte dich daran: Link deaktivieren, nach einigen Wochen GPO löschen, CSV-Dateien nach der vereinbarten Frist entfernen. Eine Messung, die nach dem Projekt weiterläuft, ist keine Messung mehr, sondern eine Dauerüberwachung, für die du keine Vereinbarung hast.

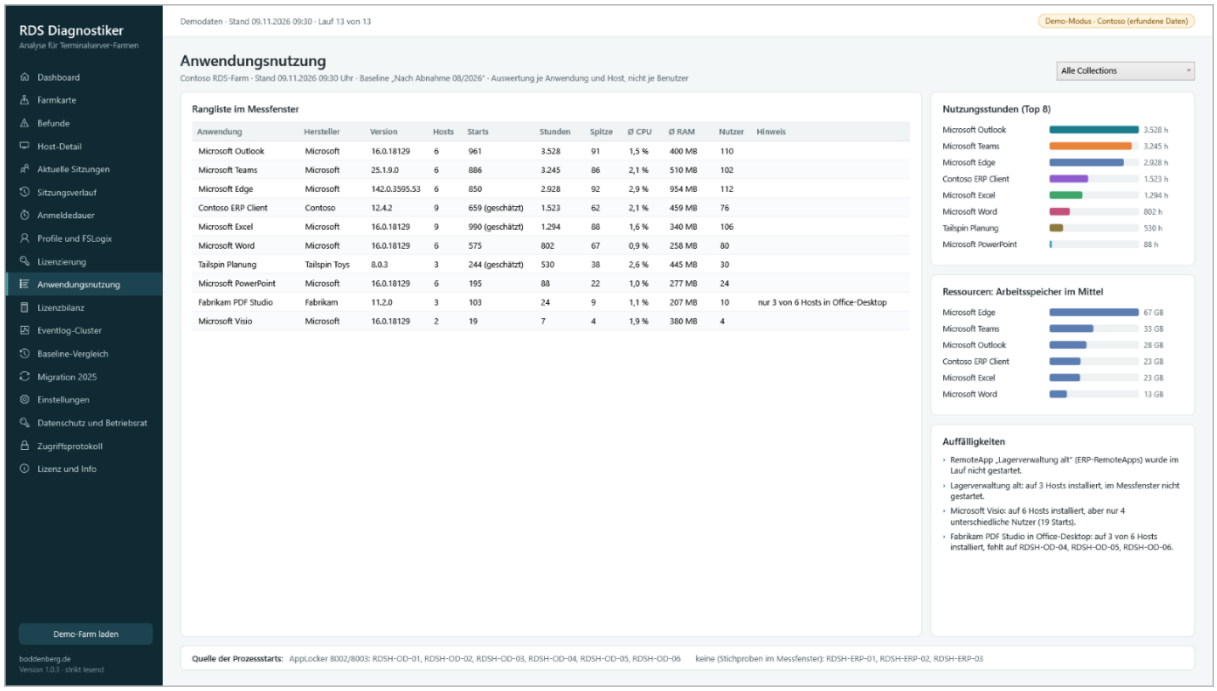
SO FINDET ES DER RDS DIAGNOSTIKER · Anwendungsnutzung je Anwendung und Host

Der RDS Diagnostiker erfasst die Farm agentlos und strikt lesend über WinRM, CIM und Remote-Eventlog. Er installiert nichts auf den Session Hosts und ändert keine Einstellungen.

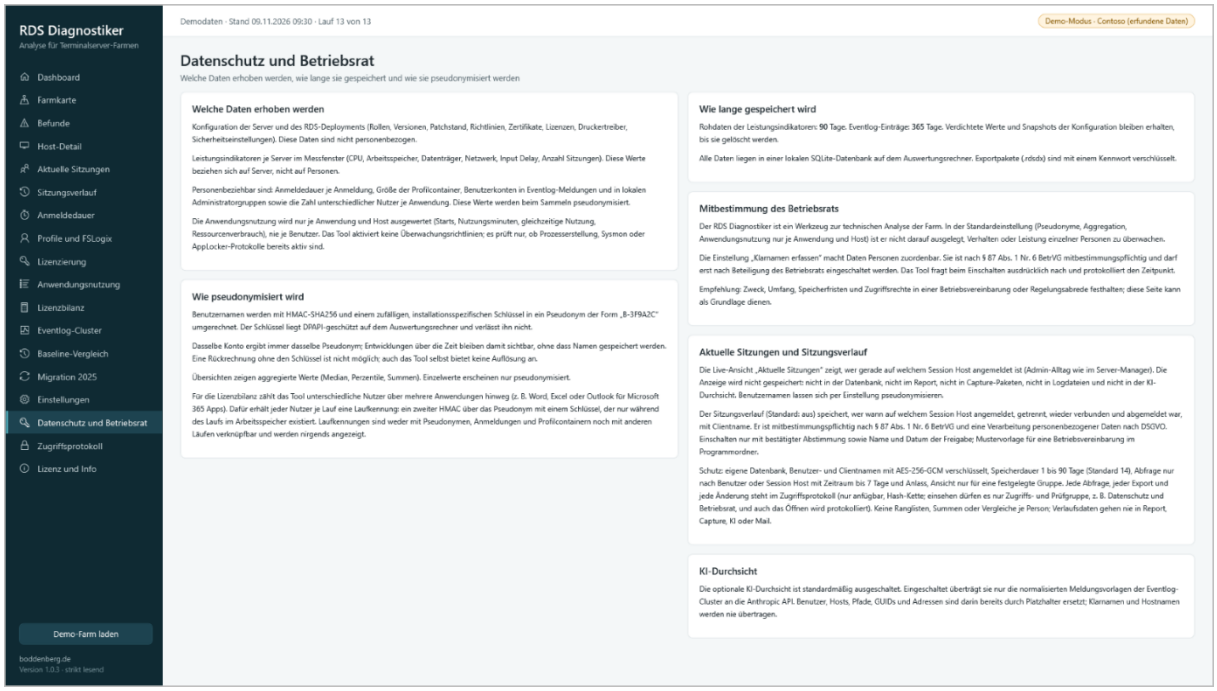
Die Ansicht Anwendungsnutzung zeigt im gewählten Messfenster je Anwendung und Host eine Rangliste mit Starts, Nutzungsstunden, Spitze sowie mittlerem CPU- und Arbeitsspeicherbedarf und weist auf Auffälligkeiten hin, etwa wenn eine Anwendung nur auf einem Teil der Hosts einer Sammlung installiert ist. Die Auswertung bleibt auf Ebene von Anwendung und Host.

Benutzerbezogene Angaben erscheinen standardmäßig pseudonymisiert. Die Datenschutzansicht dokumentiert, welche Daten erhoben werden, und ein Zugriffsprotokoll hält fest, wer Sitzungsdaten eingesehen hat. Regelmäßige Läufe richtest du über die Aufgabenplanung ein, die Ergebnisse gehen als Word-Report mit Aktionsplan an Fachbereich, Datenschutz oder Betriebsrat.

Mehr dazu auf der Produktseite RDS Diagnostiker. Du kannst den RDS Diagnostiker 15 Tage kostenlos testen.



So siehst du im RDS Diagnostiker, welche Anwendungen wie oft und wie lange auf welchen Hosts laufen, hier mit Demo-Daten.



So dokumentiert der RDS Diagnostiker, welche Daten er erhebt, wie lange er sie speichert und wie er Benutzernamen pseudonymisiert, hier mit Demo-Daten.

Willst du nach der Auswertung neue Session Hosts für die verbleibenden Anwendungen aus geprüften PowerShell-Bausteinen mit Trockenlauf und Rückbau-Skript aufbauen, hilft dir dabei der UB Mechaniker für RDS.

Wenn du den Anwendungsbestand vor einer Migration gemeinsam bewerten oder die Abstimmung mit Fachbereichen und Betriebsrat vorbereiten willst, unterstützt dich die Beratung zu Remote Desktop Services. Wer PowerShell, Ereignisprotokolle und Leistungsindikatoren für Terminalserver in Ruhe am Lab üben will, ist in der Schulung Remote Desktop Services richtig.

FAQ: RemoteApp-Nutzung auswerten

Wie finde ich heraus, welche RemoteApps benutzt werden?

Mit Bordmitteln über drei Quellen: Get-RDRemoteApp für die veröffentlichten Programme, das Ereignis 4688 im Sicherheitsprotokoll der Session Hosts für die Starts und CIM-Abfragen auf Win32_Process für das, was gerade läuft. Abgeglichen wird über den Dateinamen der Anwendung. Eine eingebaute Nutzungsstatistik haben die Remotedesktopdienste nicht.

Kann ich auf dem Terminalserver protokollieren, welche Programme gestartet werden?

Ja, mit der Überwachungsrichtlinie „Prozesserstellung überwachen“ unter Erweiterte Überwachungsrichtlinienkonfiguration › Detaillierte Nachverfolgung. Dann schreibt Windows je Prozessstart das Ereignis 4688. Vorher Datenschutz und Betriebsrat einbinden, denn das Ereignis enthält das Benutzerkonto.

Zeigt Get-RDRemoteApp, wie oft eine RemoteApp gestartet wurde?

Nein. Get-RDRemoteApp liefert nur die Konfiguration der Veröffentlichung, also Sammlung, Alias, Anzeigename, Pfad, Befehlszeilenoptionen und Benutzergruppen. Über die Nutzung sagt das Cmdlet nichts.

Welche Anwendung verbraucht auf dem Session Host am meisten Arbeitsspeicher?

Für eine Momentaufnahme gruppierst du Win32_Process per Get-CimInstance nach Prozessname und summierst WorkingSetSize. Über einen längeren Zeitraum misst du mit den Leistungsindikatoren Process(*) Working Set - Private und Prozessorzeit. Beachte, dass die Prozessorzeit je Prozess auf einen logischen Prozessor bezogen ist.

Darf ich die Anwendungsnutzung auf dem RDSH je Benutzer auswerten?

Technisch geht es, rechtlich ist es heikel. Eine Auswertung je Person ist eine Verhaltens- und gegebenenfalls Leistungskontrolle, bei der in Deutschland der Betriebsrat mitbestimmt und die DSGVO Zweckbindung und Datenminimierung verlangt. Für Lizenzen, Kapazität, Aufräumen und Migration reicht eine Auswertung je Anwendung und Host.

Wie lange sollte ich die Nutzung auf den Remotedesktopdiensten messen?

Mindestens einen vollen Geschäftszyklus, in der Regel vier bis sechs Wochen inklusive Monatsabschluss. Anwendungen für Quartals- oder Jahresabschluss erwischst du nur, wenn das Messfenster sie einschließt, oder indem du die Fachbereiche gezielt fragst.

Wie entferne ich eine RemoteApp, ohne dass sich jemand beschwert?

Erst sichern, dann ausblenden, dann entfernen: Konfiguration mit Get-RDRemoteApp | Export-Clixml sichern, mit Set-RDRemoteApp -ShowInWebAccess $false ausblenden, einige Wochen warten und erst dann mit Remove-RDRemoteApp entfernen. Beschwert sich jemand, legst du die RemoteApp mit New-RDRemoteApp und den gesicherten Werten neu an.

Warum startet eine RemoteApp nur manchmal?

Häufig ist die Anwendung nicht auf allen Session Hosts der Sammlung installiert. Der Connection Broker verteilt die Sitzung auf irgendeinen Host der Sammlung, und landet der Benutzer auf einem Host ohne die Anwendung, schlägt der Start fehl. Prüfe mit Get-RDAvailableApp und einem Vergleich der installierten Programme je Host.

Fazit: erst messen, dann ausmisten

Die Remotedesktopdienste wissen genau, was du veröffentlicht hast, und erstaunlich wenig darüber, was davon jemand benutzt. Mit Get-RDRemoteApp, Prozess-Auditing, CIM-Momentaufnahmen und Leistungsindikatoren je Prozess bekommst du diese Antwort trotzdem, und zwar belastbar genug für Lizenzgespräche, Kapazitätsplanung, Aufräumaktionen und die Migration. Entscheidend ist die Reihenfolge: Zweck klären, Mitbestimmung klären, lesend messen, je Anwendung und Host verdichten, ausblenden vor entfernen, Messung wieder abbauen. Dann wird das Startmenü kürzer, die neuen Session Hosts werden schlanker, und das PDF-Tool von 2014 bekommt endlich die Beerdigung, die es seit Jahren verdient. Trauergäste werden erfahrungsgemäß keine erwartet.

WEITERLESEN · Weiter in dieser Serie

› Session Hosts verhalten sich unterschiedlich: Konfigurationsdrift erkennen
› Wie viele Benutzer pro Terminalserver? Lasttest statt Daumenregel
› Terminalserver langsam: Leistung richtig messen statt raten
› Office auf dem Terminalserver richtig lizenzieren
› RDS-Migration von Windows Server 2016 auf 2025: die richtige Reihenfolge
› Angemeldete Benutzer auf dem Terminalserver anzeigen, mit Blick auf den Datenschutz
› Wichtige RDS-Ereignis-IDs und was sie bedeuten