Guardian One Portable Demo Site Deployment Guide

Guardian One documentation | Version 1.1 | Reviewed 30 July 2026

Purpose and Audience

Current version	1.1
Last reviewed	30 July 2026
Document owner	Product owner: Adam Timothy
Change summary	Manager-readability and controlled-publication review

Version history

Version	Date	Change	Status
1.0	Initial issue	Initial controlled document	Superseded
1.1	30 July 2026	Manager-readability and publication review	Current

This document gives a site administrator a complete, controlled procedure for deploying Guardian One Portable Assessment Mode and producing genuine reports from up to five Microsoft SQL Server instances without installing a separate Guardian One SQL Server repository. It is intended for customer DBAs, Windows administrators, security reviewers, deployment engineers and the Guardian One licence administrator.

Portable Demo Site means a restricted product mode, not synthetic assessment data. Guardian One connects to each registered target, collects live point-in-time evidence, evaluates the normal entitled rules and creates the normal individual HTML report. Only the persistence layer changes: authoritative results are stored in a protected local SQLite database.

The five-target limit (by distinct SQL Server name, not by number of assessment reports run - reassessing a registered instance is unlimited) also applies to an unlicensed Full SQL Server repository installation. Full repository mode supports a larger estate only after a signed production licence is installed: the target capacity follows the licence's highest enabled service level - Foundation 100 targets, Advisor 250, Compliance 500, Quantum 1,000.

Version History

Version	Date	Change
1.1	30 July 2026	Aligned packaged commands, protected paths and licence-request instructions with the tested r4 installer.
1.0	30 July 2026	Initial controlled Portable Demo deployment guide.

Deployment Outcome

At completion the site has:

- One customer-controlled Windows machine running Guardian One;
- Microsoft ODBC Driver 18 for SQL Server and the packaged Guardian One runtime;
- A signed licence bound to that Windows installation;
- Windows-authenticated, least-privilege connectivity to each approved target;
- A local GuardianOne_Portable.db containing assessment results and integrity digests;
- Optional standalone HTML reports for controlled sharing; and
- No dependency on GuardianOne_Toolkit or another SQL Server repository database.

Portable Demo Restrictions

The following restrictions are part of the supported mode and must be understood before the first target is assessed:

Restriction	Behaviour
SQL Server capacity	Five distinct SQL Server instance identities over the lifetime of the portable database.
Repeat assessments	No count limit for an already registered identity while the evaluation or signed subscription remains active. This is not unlimited product access.
Access and duration	Still restricted by the fourteen-day evaluation period or signed subscription, enabled service levels, installation binding, Windows/SQL permissions, and subscription lifecycle state.
Sixth instance	Rejected before an assessment record is created.
Delete or free a slot	No supported function exists. Deleting assessment history does not release a slot.
Replace or import an instance	No supported import, replacement, reassignment or slot-reset function exists.
Identity	The case-insensitive canonical value returned by SERVERPROPERTY('ServerName'). Aliases do not create a supported way to change identity.
Repository	Local SQLite only; the enterprise SQL repository is not required.
Assessment	Real live assessment, subject to target permissions and evidence availability.
HTML	Generated as an optional file; HTML is not stored in SQLite.
Encryption	SQLite is not encrypted by Guardian One. Windows ACLs and customer-approved BitLocker or equivalent protection are required where appropriate.
Local administrator	The five-slot control is an application boundary, not tamper-proof DRM against a filesystem administrator. Editing or replacing the database is unsupported.

Choose the five targets carefully. Use the canonical server and instance names, record the approved list in the deployment record, and test network/authentication with the intended Guardian One identity before allowing the first assessment to register a slot.

Platform Prerequisites

Guardian One Windows Host

Provide one customer-controlled, supported 64-bit Windows Server or Windows client installation. Apply the customer's current operating-system security baseline and patches before installing Guardian One. Join it to the appropriate domain where Windows integrated authentication requires domain trust. Set an accurate time source because licence validity, assessment timestamps and evidence windows depend on the system clock.

Use a dedicated Guardian One service identity rather than an administrator's interactive account. Permit that identity to read the application and licence files and write only to the approved repository, log, temporary and optional report locations. Do not make it a local administrator unless a separately approved Windows-collection design requires that access.

Capacity and Storage

Provide sufficient space for the application, virtual environment or packaged runtime, SQLite evidence, logs, temporary report construction and backups. For a first portable demonstration, reserve at least 2 GB of free working space in addition to operating-system headroom, then measure actual growth from representative assessments. Evidence size varies with database count, enabled collectors and assessment history; 2 GB is a starting allowance, not a retention guarantee.

Keep GuardianOne_Portable.db on a protected local NTFS volume. Do not place it in a web root, general file share, OneDrive-synchronised directory or removable media. Apply Windows ACLs to the Guardian One identity and approved administrators, and use customer-approved BitLocker or equivalent disk protection where required.

Choose the Runtime

For a customer demonstration, prefer the signed packaged Guardian One build supplied for that engagement. A source deployment additionally requires the specifically supported 64-bit Python release and the dependencies pinned by requirements.txt. Do not assume that any Python version installed on the machine is compatible. Record the package version, Python version where applicable, installation digest and signature-verification result in the deployment record.

Verify a source runtime from PowerShell:

python --version python -c "import platform; print(platform.architecture()[0])"

The output must match the release notes and show a 64-bit runtime.

Install Microsoft ODBC Driver 18 for SQL Server

- From the Guardian One Windows host, open Microsoft's current ODBC Driver for SQL Server download page.
- Download the current supported x64 Microsoft ODBC Driver 18 installer. Do not download it from an unofficial mirror.
- If the installer reports a prerequisite failure, install the current supported Microsoft Visual C++ Redistributable identified by Microsoft, then rerun the ODBC installer.
- Accept Microsoft's licence terms through the customer's normal software-approval process and complete the client-component installation. Guardian One does not require the ODBC SDK headers.
- Close and reopen PowerShell so driver discovery uses the updated installation state.

Verify that the exact driver name expected by Guardian One is installed:

Get-OdbcDriver -Name "ODBC Driver 18 for SQL Server" -Platform "64-bit"

The command must return a 64-bit driver named ODBC Driver 18 for SQL Server. If it returns nothing, do not continue. Confirm the Visual C++ prerequisite, architecture and installer outcome. Microsoft recommends keeping the driver on its latest supported minor release for fixes and security updates; upgrades must still follow customer change control.

If Get-OdbcDriver returns Driver 17 or another SQL Server driver but not Driver 18, Guardian One Setup stops with an explicit prerequisite message. Existing applications may continue using their configured driver; Guardian One does not silently select it because driver defaults, TLS behaviour and the release validation baseline may differ. Obtain the approved x64 Driver 18 installer on an internet-connected transfer workstation, verify it using Microsoft's published controls, transfer it through the customer's offline software route, and install it side by side before rerunning Setup.

Prepare the SQL Server Targets

The target may be SQL Server 2012 or later where supported by the supplied Guardian One release. Record the exact product version, edition, instance name and support status before assessment. Older releases may provide less evidence because newer dynamic management views or permissions do not exist; the report must show those limitations rather than inventing values.

A disposable SQL Server virtual machine is suitable for a demonstration. Use a current, customer-approved SQL Server build in an isolated non-production network. Select compute and storage appropriate to the demonstration workload, prevent unnecessary public access, configure automatic shutdown where appropriate, and verify Microsoft licensing, support status and customer security policy independently.

On each proposed target, confirm the canonical identity before it can consume one of the five permanent slots:

SELECT CONVERT(nvarchar(256), SERVERPROPERTY('ServerName')) AS CanonicalServerName, CONVERT(nvarchar(128), SERVERPROPERTY('InstanceName')) AS InstanceName, CONVERT(nvarchar(50), SERVERPROPERTY('ProductVersion')) AS ProductVersion, CONVERT(nvarchar(256), SERVERPROPERTY('Edition')) AS Edition;

Network and Security Readiness

Allow only the required outbound connection from the Guardian One host to each approved SQL Server endpoint. For a named instance, ensure the customer has either a fixed approved TCP port or an approved SQL Browser design. Validate DNS, routing, firewall, SPNs and domain trust under the actual Guardian One identity.

Do not expose SQL Server or RDP directly to the public internet merely to simplify a demonstration. Use the customer’s approved private network, VPN, firewall, bastion and TLS design. Internet or intranet publication of the Guardian One website is customer-managed and requires separate authentication, TLS and availability controls.

Prerequisite Acceptance Checklist

- Windows host is supported, patched, time-synchronised and customer-controlled.
- Guardian One identity and filesystem permissions are approved.
- Runtime architecture and version match the supplied release.
- The 64-bit ODBC Driver 18 verification command returns the expected driver.
- Each target’s canonical identity and product build are recorded.
- Encrypted ODBC connectivity succeeds under the Guardian One identity.
- SQLite storage is local, protected, excluded from the web root and covered by backup policy.
- Optional WinRM collection is either approved and tested or explicitly disabled.
- The proposed five-instance list has been approved before the first assessment.

Service Identity and SQL Permissions

Run Guardian One under a dedicated customer-controlled Windows identity. Assessments may be started manually whenever required. Windows Task Scheduler or another customer-approved automation method is optional and is not required by Portable Demo Mode. Grant the identity network logon and SQL Server access only to the five approved targets. Guardian One uses Windows integrated authentication; do not place passwords in config/.env.

For an interactive manual demonstration, sign in using an approved operator account that has the required target permissions. For unattended execution, use the dedicated Guardian One identity and grant only the Windows right required by the chosen host mechanism: Log on as a batch job for Windows Task Scheduler, or Log on as a service if Guardian One is installed as a Windows service. Do not grant both automatically.

The exact least-privilege grant set varies by SQL Server version and entitled collectors. Establish a login for the service identity, grant CONNECT SQL and VIEW SERVER STATE or the supported version-specific equivalent, and grant VIEW ANY DEFINITION where required for metadata. Database inventory and Query Store review require suitable CONNECT and metadata/state visibility in assessed databases. SQL Server and SQL Agent log assessment requires the supported least-privilege log-reading permission or a customer-approved certificate-signed wrapper. Do not grant sysadmin solely for Guardian One.

Some evidence will be unavailable when permissions are intentionally withheld. Guardian One must record that status as not assessed rather than treating it as a pass. Approve grants through customer change control and validate them against every SQL Server version in scope.

Optional Windows Evidence

The Guardian One machine can collect Windows metrics from remote SQL Server hosts through PowerShell remoting over WinRM. This is optional. No remote Windows password is stored by Guardian One, and Guardian One does not enable WinRM, modify firewalls or add group membership automatically.

Permissions Required by the Current Collector

Create a domain security group such as GG_GuardianOne_Windows_Readers and add the Guardian One runtime identity. On each approved target host, grant that group:

Collector operation	Required access
Open the standard PowerShell remoting endpoint	Membership of the target's local Remote Management Users group, or an equivalent explicitly secured endpoint permission.
Read CPU and disk performance counters through Get-Counter	Membership of the target's local Performance Monitor Users group.
Read operating system, computer, processor, page-file and volume information	Read and Remote Enable access to WMI/CIM namespace root\cimv2 where the host's policy does not already grant it through the remoting endpoint.

The current collector does not read Windows Event Logs, so Event Log Readers is not required for this release. It also does not require local Administrators, Domain Admins or Backup Operators. Do not grant those memberships merely to make collection work.

Group Policy is preferred for multiple targets. Use Computer Configuration → Preferences → Control Panel Settings → Local Users and Groups to add the domain reader group to the two local groups without replacing their existing membership. For a controlled single-host demonstration, an administrator may run these commands on the target:

Add-LocalGroupMember -Group "Remote Management Users" -Member "CONTOSO\GG_GuardianOne_Windows_Readers" Add-LocalGroupMember -Group "Performance Monitor Users" -Member "CONTOSO\GG_GuardianOne_Windows_Readers"

Replace CONTOSO with the customer domain. Log off/on or restart the relevant service or scheduled task after changing group membership so the runtime identity receives a new access token.

If CIM queries return Access Denied after those memberships are in place, an authorised Windows administrator should open WMI Control (wmimgmt.msc), select Properties → Security → Root → CIMV2 → Security, add the domain reader group, and grant only Enable Account, Remote Enable and Execute Methods where the customer's supported Windows version requires it. Apply the permission to the namespace and permitted descendants. Do not grant Full Write, Provider Write or namespace-editing rights.

Enable and Restrict WinRM

Prefer Kerberos on domain-joined hosts in a trusted domain. Under Kerberos, WinRM HTTP traffic is message-encrypted; restrict TCP 5985 to the Guardian One machine or approved management subnet. Use WinRM HTTPS on TCP 5986 when the approved trust design requires it, with a certificate whose subject matches the target name and whose chain is trusted by the Guardian One host.

An authorised Windows administrator can enable the standard remoting endpoint on an opted-in target:

Enable-PSRemoting -Force Get-PSSessionConfiguration -Name Microsoft.PowerShell Get-NetFirewallRule -DisplayGroup "Windows Remote Management"

Then restrict the applicable inbound firewall rule to the Guardian One collector address. The exact rule name varies by Windows version and policy; identify it first rather than copying an assumed name:

Get-NetFirewallRule -DisplayGroup "Windows Remote Management" | Get-NetFirewallAddressFilter

Apply the remote-address restriction through centrally managed firewall policy where available. Do not configure TrustedHosts=*, enable Basic authentication, allow unencrypted transport or open WinRM to the internet.

Validate Under the Actual Identity

From the Guardian One host, while running as the Guardian One identity, test endpoint negotiation and every operation used by the collector:

Test-WSMan SQLHOST01 -Authentication Kerberos Invoke-Command -ComputerName SQLHOST01 -Authentication Kerberos -ScriptBlock { Get-CimInstance Win32_OperatingSystem | Select-Object Caption, LastBootUpTime Get-CimInstance Win32_ComputerSystem | Select-Object Domain, TotalPhysicalMemory Get-CimInstance Win32_PageFileUsage Get-CimInstance Win32_Volume -Filter 'DriveType=3' | Select-Object DriveLetter, Name, Capacity, FreeSpace Get-Counter '\Processor(_Total)\% Processor Time' -MaxSamples 1 Get-Counter '\PhysicalDisk(_Total)\Avg. Disk sec/Read' -MaxSamples 1 }

All commands should return read-only evidence without an elevation prompt. Test a drive-letter volume and a mount point where both exist. If one command is denied, grant only the permission associated with that evidence group or accept that group as unavailable; do not escalate the identity to local Administrator as a blanket remedy.

Guardian One attempts this evidence automatically. The following protected runtime settings select the approved transport and bound the attempt; a failure is recorded as unavailable evidence and does not interrupt SQL assessment:

GUARDIAN_WINDOWS_COLLECTION_ENABLED=true GUARDIAN_WINDOWS_AUTHENTICATION=Kerberos GUARDIAN_WINDOWS_USE_SSL=false GUARDIAN_WINDOWS_TIMEOUT_SECONDS=45

If Windows collection is disabled or blocked, SQL-derived evidence continues. The report states that Windows evidence was not configured or was unavailable. Missing Windows metrics do not indicate either a healthy or unhealthy machine.

Install the Application

The Guardian One installation directory is the folder containing the packaged executables and their immutable supporting files. It does not mean the current user's home directory. Use one of these approved locations:

- Packaged deployment: C:\Program Files\Guardian One, created and secured by the signed installer.
- Authorised source demonstration: C:\GuardianOne, created specifically for the demonstration.

Do not run Guardian One from Downloads, Desktop, a user profile, a shared drive, a web root or a source-control working directory at a customer site.

Filesystem Permissions

Installation requires a local administrator or approved software-deployment system to create the application directory, install the ODBC prerequisite, place the protected licence/configuration files and create writable data folders. The identity running an assessment does not require local Administrator.

During packaged installation, retain the default Restricted evidence-access option and enter the customer-approved Windows identity or group. This value varies by customer and may be a domain group, service account or group managed service account. Setup validates the name by applying the ACL and stores only the principal name, not a password. The Compatibility - all local users option broadens access to every local user and requires an explicit customer risk decision.

For an approved silent deployment, include /EvidenceAccess=restricted /RuntimeIdentity="DOMAIN\Guardian One Operators". Use /EvidenceAccess=compatibility only when that exception has been approved and recorded.

Use this access model:

Location	Runtime identity	Administrators/deployment system	Ordinary users
Application code and .venv	Read and execute	Full control	No access unless approved
C:\ProgramData\Guardian One\Config configuration, public verification key and licence.key	Read	Modify/full control	No access
C:\ProgramData\Guardian One\Repository	Modify	Full control	No access
Application reports and logs folders	Modify	Full control	Read only when explicitly required, otherwise no access

Modify on the repository is required because SQLite creates and updates the database and may create temporary journal files in the same directory. Do not grant the runtime identity Full Control over the application code, licence or configuration files.

For a source demonstration, an administrator creates and populates the folder first. Replace the example source path with the approved package staging location:

New-Item -ItemType Directory -Path 'C:\GuardianOne' Copy-Item -Path 'D:\ApprovedPackage\GuardianOne*' -Destination 'C:\GuardianOne' -Recurse Set-Location 'C:\GuardianOne'

The administrator should apply ACLs through the customer’s software-deployment policy or icacls, using the access model above and the actual Guardian One identity. Review the effective permissions in Properties → Security → Advanced before proceeding. Avoid a generic command copied from documentation when domain names and inherited permissions have not been reviewed.

Create the Source Runtime

After Set-Location, this command confirms that PowerShell is in the correct directory:

Get-ChildItem .\src\main.py, .\requirements.txt, .\config

All three paths must exist. The approved installer/administrator can then create the virtual environment and install pinned dependencies:

python -m venv .venv .\.venv\Scripts\Activate.ps1 python -m pip install -r requirements-lock.txt

After installation, remove Modify permission from the runtime identity on application code and .venv; retain Modify only on the repository, reports and logs locations. A normal assessment should be run from a non-elevated PowerShell session under the approved runtime identity.

For a packaged customer build, follow the signed installer instructions instead. Confirm the executable or package signature and supplied digest before use. Do not deploy development signing keys, issuer scripts, tests, sample credentials or a populated developer .env file.

Configure Portable Mode

What Is Being Configured

Guardian One reads local runtime settings from a plain-text file named .env inside the installation's config folder. For the source demonstration path used in this guide, the complete filename is:

C:\GuardianOne\config\.env

For the packaged installation, the configuration file is C:\ProgramData\Guardian One\Config\.env. This is a configuration file, not a PowerShell script and not a command to paste into a console. It contains no section headings: each non-comment line is one NAME=value setting.

The deployment administrator creates and changes the file. The service owner approves the target and operating choices. The security owner approves encryption, certificate and optional WinRM choices. The Guardian One runtime identity receives Read permission only. Do not include passwords, licence contents or private keys in this file.

Create the File

From the installed Guardian One directory, an administrator copies the supplied template:

Set-Location 'C:\GuardianOne' Copy-Item '.\config\.env.example' '.\config\.env' notepad.exe '.\config\.env'

If .env already exists, stop and review it rather than overwriting it. Enter one setting per line. Spaces after the equals sign become part of some values, so use the form shown. A backslash in a SQL Server instance or Windows path is literal and is not doubled.

The following is an example for a fictional named instance. UKSQLDEMO01\DEMO must be replaced by the approved customer target:

SQL_SERVER=UKSQLDEMO01\DEMO SQL_TARGET_DATABASE=tempdb GUARDIAN_PORTAL_PORT=8080 SQL_ODBC_DRIVER=ODBC Driver 18 for SQL Server SQL_ENCRYPT=true SQL_TRUST_CERTIFICATE=false SQL_LOGIN_TIMEOUT=15 SQL_QUERY_TIMEOUT=30

GUARDIAN_REPOSITORY_MODE=portable GUARDIAN_PORTABLE_DATABASE=C:\ProgramData\Guardian One\Repository\GuardianOne_Portable.db

GUARDIAN_WINDOWS_COLLECTION_ENABLED=true SQL_OPTIONAL_FIRST_RESPONDER_EXECUTION_ENABLED=false

Meaning and Approval of Each Setting

Setting	Meaning	Required decision
SQL_SERVER	Real SQL Server network and instance name to assess. It is not the Guardian One repository.	DBA confirms it matches one approved portable target before the first run.
SQL_TARGET_DATABASE	Initial database used to establish target discovery.	Keep the supported tempdb default. Repository creation uses master separately and deliberately.
GUARDIAN_PORTAL_PORT	Local portal TCP port.	Defaults to 8080; choose another free port from 1 to 65535 through Setup or Guardian One Configuration.
SQL_ODBC_DRIVER	Exact installed Windows ODBC driver name.	Must match the 64-bit driver returned by Get-OdbcDriver.
SQL_ENCRYPT	Requests encrypted SQL connectivity.	Must remain true for a customer deployment.
SQL_TRUST_CERTIFICATE	When false, the target certificate name and trust chain must validate. When true, encryption remains enabled but certificate identity is not validated.	Keep false. A temporary true requires explicit security approval and a remediation date; it is not the preferred fix for certificate errors.
SQL_LOGIN_TIMEOUT	Seconds allowed to establish the target connection.	15 is the default starting value; change only for a documented network reason.
SQL_QUERY_TIMEOUT	Default maximum seconds for an individual ordinary assessment query.	30 is the default. Specialist bounded collectors may have their own documented timeout.
GUARDIAN_REPOSITORY_MODE	Selects local portable persistence instead of the enterprise SQL repository.	Must be exactly portable for this guide.
GUARDIAN_PORTABLE_DATABASE	Absolute path to the authoritative SQLite evidence file and five-slot registry.	Service owner approves the local protected path and backup scope. Do not point it at a web or shared folder.
GUARDIAN_WINDOWS_COLLECTION_ENABLED	Compatibility setting retained for existing deployments; the current Configuration workflow writes and runs with automatic Windows evidence enabled.	Keep true. If customer policy blocks WinRM, Guardian One records the bounded attempt as unavailable and continues with SQL evidence.
SQL_OPTIONAL_FIRST_RESPONDER_EXECUTION_ENABLED	Controls additional optional Brent Ozar procedures beyond the established sp_Blitz assessment.	Remains false for the current demonstration release.

SQL_DATABASE and SQL_REPOSITORY_SERVER are not required in Portable Demo Mode. Remove inherited development values for them from the customer .env so they cannot be mistaken for portable dependencies.

Create C:\ProgramData\Guardian One\Repository and restrict it to the Guardian One runtime identity, designated administrators and the approved backup operator. Prevent ordinary portal or report users from reading or modifying the SQLite file. Do not place it in a web root or shared report folder.

Manager Acceptance Evidence

Before approving the first run, the manager should receive a redacted copy of the setting names and non-sensitive operating choices, the approved canonical target list, the responsible service identity, the repository path, evidence of its ACL review, the licence identifier and validity—not the licence key itself—and confirmation of whether remote Windows access is approved and expected to succeed. The .env file itself should remain protected operational configuration rather than being emailed as an approval document.

Obtain and Install the Licence Key

Guardian One verifies a signed, installation-bound licence before any collector runs. For a packaged installation, open Guardian One from the Start menu and select Create licence request. Alternatively, run:

& 'C:\Program Files\Guardian One\GuardianOne.exe' --registration-request --name "Authorised contact" --company "Customer organisation" --contact "Telephone or agreed secure contact route"

Send the generated request through the agreed fulfilment route. It contains a one-way installation identifier and the supplied contact fields; the command does not transmit it automatically. The Guardian One licence administrator returns a signed licence.key and the approved public verification key where these are not embedded in the package. Install them at the paths supplied with the customer build, normally:

C:\ProgramData\Guardian One\Config\licence.key C:\Program Files\Guardian One\config\licence-public-key.pem

Alternatively set GUARDIAN_LICENSE_KEY_FILE and GUARDIAN_LICENSE_PUBLIC_KEY_FILE to protected absolute paths. Only the public verification key belongs at the customer site. Never deploy the private licence-signing key or development issuer material.

The licence must enable baseline_assessment through the appropriate Foundation or higher entitlement, match the installed product version, be within its validity period and match the Windows installation identity. A missing, malformed, altered, expired, wrong-machine or insufficient licence stops execution before target collection. Copying the code and licence to another Windows machine does not transfer the entitlement; create a new request and follow the agreed transfer process.

Connectivity and Evidence Validation

The DBA validates the target under the actual runtime identity; the Windows administrator checks host access and ACLs; the service owner approves the evidence record. Guardian One Configuration does not expose preflight or Windows-evidence choices. During assessment, Windows evidence is attempted automatically and each failure becomes a sanitised collection status and diagnostic code in the operator log.

For every remote Windows target where customer policy permits WinRM:

- Use the exact identity entered during Setup (for example CONTOSO\\GuardianOne-Readers or CONTOSO\\svc-guardianone). Add it to the target's Remote Management Users and Performance Monitor Users local groups.
- Enable WinRM on the target (winrm quickconfig) and allow inbound TCP 5985 for Kerberos or 5986 for approved HTTPS from the Guardian One host only. Do not open these ports to the whole network.
- In WMI Control (wmimgmt.msc), grant that identity Enable Account, Remote Enable, and Read Security on root\\cimv2. The identity does not need local or domain Administrators membership.
- Confirm forward/reverse DNS, domain trust and time synchronisation. Kerberos will fail when the target name, SPN or clock is wrong.
- From the Guardian One host, run Test-WSMan TARGETNAME as the runtime identity. A successful result proves WinRM transport; Guardian One then performs the read-only WMI evidence query.

If customer policy does not permit remote Windows access, do not broaden permissions merely to make the attempt pass. Guardian One records the Windows evidence as unavailable, continues SQL collection, and does not interpret the gap as a pass.

1. Confirm the Canonical SQL Server Identity

Using SSMS or the customer’s approved SQL query tool while authenticated as the Guardian One identity, connect to the proposed target and run:

SELECT CONVERT(nvarchar(256), SERVERPROPERTY('ServerName')) AS CanonicalServerName, CONVERT(nvarchar(128), SERVERPROPERTY('InstanceName')) AS InstanceName;

The DBA records the returned CanonicalServerName against one of the five approved slots. It must identify the intended instance. Stop if it is blank, unexpected, a different cluster identity or not on the approved list.

2. Prove the Target Connection Is Encrypted

In the same session run:

SELECT encrypt_option, auth_scheme, net_transport FROM sys.dm_exec_connections WHERE session_id = @@SPID;

Accept only encrypt_option = TRUE. Record the authentication scheme and transport. If the connection fails with certificate validation enabled, correct target certificate name/trust; do not silently change SQL_TRUST_CERTIFICATE to true.

3. Approve the Five-Instance Register

The deployment record must contain five numbered rows with the canonical identity, business owner, environment, approval state and date. Fewer than five may be populated initially. The service owner and DBA sign or approve the record through the customer’s normal change system. Compare SQL_SERVER and the query result above with that record before every first assessment of a new identity. Do not use aliases as substitutes for canonical names.

4. Verify SQLite Location and Access

The Windows administrator runs:

$repository = 'C:\ProgramData\Guardian One\Repository' Test-Path $repository Get-Item $repository | Select-Object FullName, Attributes Get-Acl $repository | Format-List Owner, AccessToString

Test-Path must return True; FullName must be the approved local path; and the ACL must show only the runtime identity, approved administrators and backup identity with the agreed access. The security reviewer separately confirms that the directory is not beneath the website document root and that the backup job includes the SQLite database. Do not create a dummy database or copy customer evidence merely to test backup.

5. Verify Licence File Access

Run these checks once under the Guardian One identity and once under an ordinary test user:

Test-Path 'C:\ProgramData\Guardian One\Config\licence.key' Test-Path 'C:\Program Files\Guardian One\config\licence-public-key.pem' Get-Acl 'C:\ProgramData\Guardian One\Config\licence.key' | Format-List Owner, AccessToString

The runtime identity must receive True for both Test-Path calls and have Read access. The ordinary user must not have Modify, Write, Delete or permission-changing rights. The deployment administrator retains controlled modification rights. Do not print or paste licence contents into the acceptance record.

6. Confirm Automatic Windows Evidence Handling

Verify that .env contains exactly:

GUARDIAN_WINDOWS_COLLECTION_ENABLED=true

Where remote access is approved, complete the WinRM validation commands in the Optional Windows Evidence section under the actual runtime identity and attach the sanitised successful output. Where access is not approved or unavailable, record the expected Windows evidence unavailable result; Guardian One will continue with SQL evidence. A ping response alone is not sufficient proof.

7. Confirm Clean Output Locations

Before introducing customer data, the deployment administrator inventories the writable locations:

Get-ChildItem 'C:\ProgramData\Guardian One\Reports' -Force -ErrorAction SilentlyContinue Get-ChildItem 'C:\ProgramData\Guardian One\Logs' -Force -ErrorAction SilentlyContinue Get-ChildItem 'C:\ProgramData\Guardian One\Repository' -Force

Only supplied non-customer assets or explicitly documented empty folders may be present. Remove development reports, test logs and validation databases through the customer’s approved secure-deletion process before the first run. Do not delete an existing portable database merely to clear registered slots; if one exists, stop and establish its provenance.

Preflight Approval

The service owner authorises the first assessment only when all seven checks have an identified performer, date, result and evidence reference. Any failed or ambiguous check is a stop condition, not a warning to bypass.

Illustrative installation, configuration and preflight control record

Run One Real Assessment

Open a non-elevated PowerShell session under the approved runtime identity, change to the installed directory, and run:

Set-Location 'C:\Program Files\Guardian One' .\GuardianOne.exe --portable --open-report

No Python installation or source tree is required for the packaged build. The source-layout command belongs only in the separately controlled source deployment procedure.

Guardian One prints a timestamp and stage indicator as it proceeds. It verifies the licence, connects to the real target, registers or recognises its canonical identity, creates a portable assessment record, collects available evidence, applies entitled rules, saves the complete JSON result and digest, and generates the HTML report. --open-report is optional.

To assess another approved instance, change only SQL_SERVER, validate the identity and run the same command. Do not automate a target catalogue until all proposed names and the irreversible slot effect have been approved.

Output and Evidence Locations

The authoritative portable evidence is GuardianOne_Portable.db. Each completed assessment includes the full result JSON, status, health result and SHA-256 digest. The generated HTML report is a presentation artifact and may be placed in the configured optional report folder for controlled sharing. Losing the HTML does not delete the SQLite evidence; losing the SQLite database loses the authoritative portable history and slot registry.

Back up the SQLite database only with an application-consistent method while no assessment write is active. Protect backup media to the same standard as the live evidence. Test restore into an isolated authorised recovery environment. Restoring an earlier file to evade or reassign registered slots is unsupported.

Verification After Each Run

Confirm that:

- All numbered stages completed or clearly recorded a bounded collection limitation;
- The report identifies the intended canonical SQL Server instance;
- The assessment completion time and health summary are present;
- Unavailable Windows or third-party evidence is not displayed as a pass;
- Risks include evidence, impact, recommendation and confidence where applicable;
- The SQLite file modification time changed and remains inaccessible to unauthorised users; and
- The HTML report contains no unexpected customer-sensitive data before sharing.

Website Publication

Portable mode always produces the individual standalone report. If a web server is used, publish only approved static HTML and assets or an explicitly supported portable portal build—never the SQLite database, licence file, .env, logs or source tree. The customer owns DNS, TLS certificates, authentication, reverse proxy, firewall exposure, availability, backup and vulnerability management.

Choose a Publication Method

The customer has three supported operating choices:

- No website: Open the generated HTML report locally in the approved browser or transfer it through the customer's controlled document-sharing process. This is the simplest choice for a short demonstration.
- Static IIS website: Use Microsoft IIS to publish only approved Guardian One website assets, documentation PDFs and selected standalone HTML reports. This is suitable when no database-backed portal function is required.
- Authenticated portal behind a reverse proxy: For a supported database-backed portal deployment, run the Guardian One portal service on a loopback or restricted internal address and place the customer's approved IIS, application gateway or other reverse proxy in front of it. The proxy terminates TLS and enforces the customer's authentication, request filtering and network policy. This requires a separate deployment design and is not created merely by sharing the application folder.

Static IIS Example

An authorised IIS administrator should:

- Install the IIS Web Server and Static Content features through the customer's server-management process.
- Create a dedicated physical directory such as C:\inetpub\GuardianOnePublished.
- Copy only the approved files from the Guardian One site folder and explicitly approved standalone reports into that directory. Do not point IIS at C:\GuardianOne.
- Grant the IIS application-pool identity Read access to the published directory only. It does not need access to config, data, logs, .venv, source code or the SQLite repository.
- Create a dedicated IIS site or application with the customer-approved host name and HTTPS binding. Use a trusted certificate and redirect or disable HTTP according to policy.
- Enable the customer's approved authentication method and remove anonymous access when the documents or reports are not public information.
- Disable directory browsing, restrict MIME types to those actually published, apply request filtering and test that direct requests for .env, .db, .key, .pem, source and log extensions return Not Found or Forbidden.
- Establish an approval process for publishing each new report; Guardian One does not copy every generated report into IIS automatically.

The security reviewer should verify from a separate client that the intended home page and approved report open over HTTPS, authentication is enforced, directory listings are unavailable, and repository/configuration paths cannot be retrieved.

Python's http.server or another directory-serving development command may be used only for a short local development check on a non-public loopback address. It is not the alternative for a customer website because it lacks the required authentication, TLS, request filtering, service management and hardened publication boundary.

Common Customer Questions

Will Guardian One Work with a Web Front End?

Yes. Enterprise deployments can use the database-backed Guardian One web portal for estate-wide reporting, compliance comparison, retained history, machine templates and governed exceptions. Customers can also publish approved standalone reports and documentation through IIS. The customer controls the web address, TLS certificate, authentication, reverse proxy, firewall exposure, availability and backup. The SQLite database, SQL repository credentials, licence, .env, logs and application source must never be placed in the web root.

Does Guardian One Have to Have a Web Front End?

No. Assessment collection, rule evaluation, findings, recommendations and repository storage do not depend on a website. An administrator can run Guardian One manually or through optional customer-approved automation and open the generated standalone HTML report locally. If the web front end is unavailable, assessments and storage can continue. Only portal access is affected. Portable Demo Mode also provides an optional local estate portal backed by integrity-checked SQLite evidence.

What Happens if SQL Server Authentication Is Required?

The current release supports Windows integrated authentication. SQL Server usernames and passwords must not be placed in config\.env, target catalogues, SQLite, reports, logs or the Guardian One SQL repository. Where available, use a dedicated domain identity or group managed service account so Windows manages authentication and password rotation.

Vault-backed SQL Server authentication is a future controlled capability, not a current workaround. Before it can be supported, Guardian One must retrieve a password at runtime from Windows Credential Manager with DPAPI protection or a customer-approved enterprise vault, store only a secret reference in configuration, restrict retrieval to the runtime identity, keep the value in memory only, sanitize failures, and support rotation without rewriting target files. Until that implementation is completed and tested, a customer requiring SQL Server authentication has a deployment blocker that must be resolved rather than bypassed.

Does Portable Demo Mode Perform a Real Assessment?

Yes. Guardian One connects to the approved real SQL Server target and runs the enabled collectors and rules. Portable describes local SQLite persistence; it does not mean generated or fictional assessment evidence. Results still depend on the permissions and evidence available at collection time.

Where Is Portable Assessment Data Stored?

The authoritative result is stored in GuardianOne_Portable.db, normally under C:\ProgramData\Guardian One\Repository. It contains the complete assessment JSON, health result and integrity digest. HTML is a generated presentation artifact and is not stored in SQLite.

Can One of the Five SQL Server Slots Be Deleted or Replaced?

No. Portable Demo Mode permits five distinct canonical SQL Server identities over the lifetime of the portable database. Those five instances can be reassessed without limit, but there is no supported delete, import, replacement, reassignment or slot-reset function. A sixth identity is rejected.

What Happens When Windows Metrics Are Blocked?

SQL-derived evidence continues. Guardian One records Windows collection as not configured, unreachable, authentication failed, permission denied or partial as appropriate. Missing Windows evidence is not presented as either a healthy or unhealthy machine and is never converted into a pass.

Does Guardian One Depend on Third-Party SQL Toolkits?

No. Native Guardian One discovery, findings and recommendations retain value when optional third-party procedures are absent. Approved Brent Ozar, Erik Darling or Ola Hallengren capabilities may enrich evidence where installed, permitted and enabled, but missing toolkit evidence is identified rather than treated as a pass.

Can the Application and Licence Be Copied to Another Machine?

The application files can technically be copied, but the signed licence is bound to the authorised Windows installation and will not validate on a different machine. Generate a new registration request and follow the agreed transfer or replacement process. Copying the portable database is also subject to customer evidence-handling controls and does not create a supported way to reset the five-slot registry.

Where Are Contractual Terms Obtained?

Customer-specific licensing, support and contractual documents are supplied separately through the agreed commercial process. They are not published in the public documentation centre.

Troubleshooting

Licence Missing or Invalid

Check: Confirm the configured licence and public-key paths, file Read permission, licence validity dates, enabled service, installed product version and Windows installation binding.

Action: Generate a registration request on this machine and use the agreed fulfilment route. Do not copy a key from another installation or edit the signed licence.

Login Failed or Cannot Open the Target Database

Check: Confirm the assessment is running under the intended Windows identity. Review SQL_SERVER, SQL_TARGET_DATABASE=tempdb, DNS, SPN/domain trust and the target SQL login grants.

Action: Have the DBA test an encrypted connection to master under the same identity. Correct the specific network, identity or SQL permission failure; do not substitute sysadmin as a general fix.

Certificate-Chain or Name Validation Failure

Check: Confirm the SQL Server certificate is current, has Server Authentication usage, contains the target DNS name and chains to a certificate authority trusted by the Guardian One host.

Action: Install or correct the approved server certificate and trust chain. A temporary trust-certificate exception requires written security approval and a remediation date.

Sixth SQL Server Target Rejected

Cause: The portable repository has already registered its lifetime maximum of five distinct canonical SQL Server identities.

Action: Continue assessing one of those five instances or obtain the appropriate non-demo deployment. Deleting reports, editing SQLite or replacing the database to free a slot is unsupported.

Windows Evidence Unavailable

Check: Use the WinRM validation sequence in this guide to distinguish DNS, TCP/firewall, authentication, remoting-endpoint, CIM and performance-counter permission failures.

Action: Correct only the failed layer or leave Windows collection explicitly disabled. SQL-derived assessment should continue, and unavailable Windows evidence must not appear as a pass.

SQL Server or SQL Agent Logs Not Assessed

Check: Review the runtime identity’s supported log-reading permission, SQL Agent availability, archive retention and the configured seven-day evidence scope.

Action: Grant only the approved version-specific permission or signed-wrapper access, then reassess. Do not grant sysadmin solely for log collection.

SQLite Database Is Locked

Check: Confirm that another assessment, backup, antivirus product or file-synchronisation tool is not holding GuardianOne_Portable.db or its journal files.

Action: Allow the active writer to finish and reschedule the conflicting process. Do not copy a live database manually or terminate a writer without establishing recovery consequences.

HTML Opens but Expected Data Is Absent

Check: Review the timestamped stage output, collection status and limitations for the missing section. Confirm the HTML belongs to the intended assessment run.

Action: Correct the relevant target permission or configuration and run a new assessment. The HTML is derived presentation; the SQLite result is the authoritative portable evidence.

Decommissioning and Handover

Decommissioning is a controlled customer decision, not simply deletion of the application directory. The service owner initiates it; the Guardian One administrator inventories the deployment; the DBA removes target access; the Windows/IIS administrator removes runtime and publication access; and the information owner approves evidence retention or destruction.

1. Freeze New Assessment Activity

Record the agreed final-assessment time. Complete or formally cancel any assessment already running. Do not terminate Python, the packaged executable or SQLite file activity while a write is active.

If optional Task Scheduler automation was used, identify it without deleting anything:

Get-ScheduledTask | Where-Object TaskName -Like '*Guardian One*' | Select-Object TaskPath, TaskName, State

The Windows administrator disables the confirmed Guardian One tasks through the customer's change process and records their names and final states. If assessments were manual, record No assessment automation configured. Do not imply that a scheduled task must exist.

2. Withdraw Website Publication

Remove the Guardian One IIS binding, application or reverse-proxy route from service according to the approved web change. Preserve the configuration record long enough to support rollback. Confirm from a separate client that the former URL no longer serves reports or documentation and that no directory listing appears.

Stopping website publication does not remove files copied to other document systems, email, tickets or backups. The information owner must include those copies in the evidence-disposition decision.

3. Create the Final Inventory

While signed in as an authorised administrator, inventory but do not expose file contents:

Get-Item 'C:\ProgramData\Guardian One\Repository\GuardianOne_Portable.db' | Select-Object FullName, Length, CreationTimeUtc, LastWriteTimeUtc Get-FileHash 'C:\ProgramData\Guardian One\Repository\GuardianOne_Portable.db' -Algorithm SHA256 Get-ChildItem 'C:\GuardianOne\reports' -File -ErrorAction SilentlyContinue | Select-Object Name, Length, LastWriteTimeUtc

Record the SQLite filename, size, timestamps and SHA-256 digest; the five registered canonical instance identities; the final assessment identifier and time for each instance; the list of published/exported reports; and the backup locations known to the customer. Do not copy raw licence or .env contents into the handover record.

4. Decide Evidence Disposition

The information owner chooses one of these outcomes and records the retention authority, owner, location and review/destruction date:

- Retain operationally: Keep the protected SQLite database and approved reports accessible only to the nominated custodian.
- Archive: Create one application-consistent final backup while no assessment is writing, verify its digest after transfer, restrict the archive and remove working copies only through the customer's approved process.
- Destroy: Use the customer's approved secure-deletion and backup-expiry process for SQLite, reports, logs, temporary files and published copies. Obtain completion evidence from the responsible system owners.

Guardian One must not offer an application delete/reset function to recycle portable instance slots. Decommissioning destruction is a customer information-governance process, not a way to reuse the five-slot licence restriction.

5. Remove Technical Access

After evidence handling is complete and rollback is no longer required:

- The DBA removes or disables the Guardian One SQL login and database permissions on each target through an approved SQL change.
- The Windows administrator removes the runtime identity from target Remote Management Users, Performance Monitor Users and any explicitly granted root\cimv2 permissions.
- The firewall administrator removes Guardian One-specific WinRM rules or source-address allowances that are no longer used.
- The Windows administrator removes Log on as a batch job or Log on as a service if it was granted solely for Guardian One.
- The identity owner disables or removes the dedicated account only after confirming it has no other approved use.
- The application owner removes licence/configuration files and application binaries through the approved software-removal process.

Each owner verifies the effective state after the change. Do not assume that deleting C:\GuardianOne removes SQL logins, group memberships, firewall rules, IIS publication, scheduled tasks or backups.

6. Licence Closure

Record the licence ID, installation identifier, validity period and enabled services from the approved licence record without reproducing the signed key. Removing licence.key prevents further authorised starts on that installation but does not revoke copies or notify the licence administrator automatically. Follow the separately agreed process for closure, transfer or replacement; a licence copied to another Windows installation remains invalid because of machine binding.

7. Verify Closure

The service owner obtains evidence that:

- No Guardian One assessment process, optional task or service remains active;
- The website/reverse-proxy URL no longer publishes Guardian One content;
- Target SQL and optional Windows access has been removed where no longer required;
- The evidence disposition matches the information owner’s instruction;
- Remaining archives have a named custodian, access restriction and retention date;
- Application, configuration and licence working files have been removed or retained exactly as approved; and
- The final handover record is stored in the customer’s controlled change or service-management system.

Handover Record

The completed record should identify the customer/site, Guardian One host, licence ID, portable database digest, five-slot registry, last assessment per instance, reports and publication locations, evidence decision, retained backups, access-removal changes, exceptions, responsible owners, completion date and service-owner approval. An unresolved item prevents closure and must have an owner and due date.

The closure panel in the earlier deployment-control figure shows the minimum visible completion states. It is an illustrative Guardian One record, not a substitute for the customer’s change evidence.

Deployment Acceptance Record

The site owner should retain approval for the host, service identity, five intended canonical SQL Server names, SQL permissions, Windows collection choice, repository path, backup policy, report recipients, website exposure, licence ID, validity period, enabled services, test results and acceptance date. Any deviation should be risk-assessed before collection begins.