Server Hardening Guide

Description

This document is designed to provide guidance for design decisions in the Privileged Identity host server configurations. The statements made in this document should be reviewed for accuracy and applicability to each customer's deployment.

Configurations

Program Data Protection

- Encryption of passwords in the database - Use a Hardware Security Module (HSM) such as those offered by Safenet or Entrust nCipher. Although the software encryption is the Federal Information Processing Standards (FIPS) certified algorithm, it is encryption done in software which means the clear text information and encryption keys are in main system memory and CPU. Using an HSM will remove the key management from main system memory and provide further encryption protection as addressed by FIPS 140-2 level 2 and level 3 mechanisms. Whether this is provided as a network device or a local PCI device is up to you; local PCI is obviously faster but can be much harder to manage when dealing with virtual machines or distributed components.
- Use Secure Sockets Layer (SSL) protection for the SQL database connections. SQL Server supports SSL connections for its network traffic and we support those methods as well. This has the benefit of ensuring absolutely no data is transmitted in clear text over the wire. In the same breath, whenever we transmit passwords to or from the database, those values are already encrypted by our crypto provider or the HSM if implemented. Thus we are never sending clear text passwords, though there is other auditing and system information which would be passed in the clear and the SSL connection would help further protect that.
- Enable Transparent Data Encryption (TDE) in the program data store to further protect data at rest.
- Use SSL protection from the web server to the client browser. The use of SSL certificates to implement an HTTPS connection is a function of IIS. For better protection consider disabling SSLv3 at the server level and force the use of the latest TLS scheme.
- Export the encryption key post installation and store in a secured location, preferably as an encrypted file.
- Change the program encryption key periodically. This is a manual process but helps to ensure no data required to access the stored passwords remains static indefinitely.
- Change the default password on the password store access from within the console.

HSM Distribution

The encryption process takes place before information is ever written to the data store (MS SQL/Oracle). This means if the intention is to use a local PCI device, an HSM would be installed wherever there is a management console/deferred processor/zone processor or a web site. Following a fully redundant deployment scenario where there are 6 machines (2 x web, 2 x console, 2 x DB) there would be 4 HSM devices. In this scenario, it could be more cost efficient to implement a network capable HSM, but speed and availability become dependent on network infrastructure for the HSM.

Network Traffic Protection

Implement Internet Protocol Security (IPSec) with Encapsulating Security Payload (ESP) and Authentication Header (AH). Preferred authentication methods would be Public Key Infrastructure (PKI) or Kerberos. IPSec can be selectively implemented to be required whenever traffic is sent from the Privileged Identity console and deferred processor and/or zone processor systems. This is most easily implemented as a group policy and will help in a few ways:

- ESP will protect the entire packet payload
- AH will verify the source and destination systems
- Traffic will no longer appear as the Server Message Block (SMB) protocol, but rather Internet Security Association and Key Management Protocol (ISAKMP), with no relevant information being gleaned from the data

Password Management and Hashing

- For Microsoft Windows systems, either set passwords that are 15 characters or longer or use group policy to disable LAN Manager hashes. If the group policy to disable LAN Manager hashes is not enabled, setting passwords that are 15 characters in length will have the same effect. This helps prevent against rainbow table attacks.
- Never configure a static value password when changing passwords. These passwords are known by the Privileged Identity administrator, and anyone else that password is given to which limits the company's ability to assign accountability should something bad happen involving the account. Further, these passwords will never change unless a new password change job is run against them. Rather, use the random password option and ensure the option in the program is set to use a unique password for each account. This ensures that no two accounts will ever have the same password and that the password will likely be randomized following checkout, which further helps against pass-the-hash attacks.

Web Site Configuration

Do not allow default authenticated user access to the web site. This allows any user not explicitly granted a permission with login access to the web site just as the *Authenticated Users* group does in Windows.

Enable and configure the **Hide Recovered Password** option to automatically hide the displayed password after a certain amount of time.

Force inactive web session timeouts to a short value that is relevant to how your users work. A shorter value will ensure users who recover passwords and then continue to leave the web site open will not be able to leave themselves logged in indefinitely.

Configure the web site options to require secure sessions. If a session is established and SSL is not enabled, the web site will not authenticate the user and will not pass the credentials over the wire.

Use integrated authentication and do not use browsers other than Internet Explorer 9 or later. This helps to ensure passwords are not sent in clear text and helps to guarantee your patch management cycle.

If you disable the copy button, this will stop the copy button feature from working in our product, thus making it harder for the user to place password data onto their clipboard.

Enable the option to **Disable concurrent logins from a single user**.

Enable the option to **Embed a unique identifier with each page** and with each request. This purpose of these options is to help prevent replay attacks and cache poisoning attacks. The downside is that users will only be able to click one link per page before having to re-navigate to the same page to perform a second action. This will also effectively disable the back button just as an online banking site would.

Enable the **Store only Authentication Token in the Cookie** option. This may slow down some page operations for low powered users, but ensures that no information about the user login can be gleaned by examining the cookie.

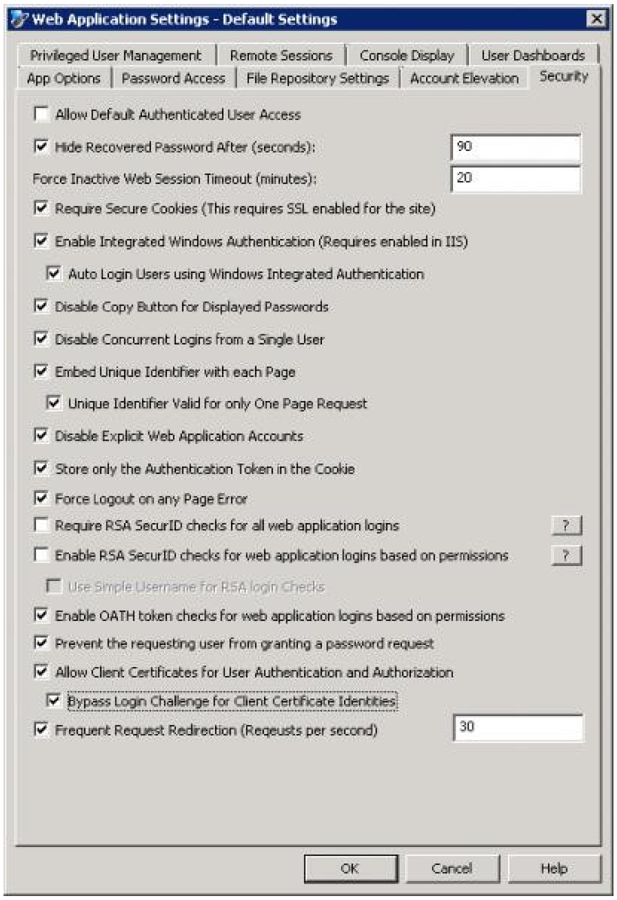
Configure the web site options to **Force Logout on any Page Errors** to help protect against brute force or injection or other types of attacks.

Configure two-factor authentication using either RSA or using OATH (enabled in the screenshot below). Specifically, configure the users with 8 digit HOTP type tokens. This requires the users to have either a soft token on the computer, smartphone, or physical device. TOTP tokens are available but provide weaker security as they will either email or SMS the pass code to the user, which means other users with access to the user's email account can still potentially leverage the user's two-factor token.

Enable **Prevent the requesting user from granting a password request** option to prevent users engaging in a workflow (password request) from being able to grant their own request when permissions would otherwise allow them to grant their own requests.

Use certificate-based authentication rather than delegating to users and groups or explicit accounts where possible. This implies there is some form of certificate management but is again designed to prevent the passing of user credentials over the wire. This requires users have certificates and IIS is configured to at least accept user certificates and also requires the web site to require SSL.

Configure **Frequent Request Redirection** to a value of *10* (this value may need to be adjusted based on actual usage) to prevent DDOS style attacks against the web site.



☐ Allow Default Authenticated User Access

☑ Hide Recovered Password After (seconds): 90

Force Inactive Web Session Timeout (minutes): 20

☑ Require Secure Cookies (This requires SSL enabled for the site)

☑ Enable Integrated Windows Authentication (Requires enabled in IIS)

☑ Auto Login Users using Windows Integrated Authentication

☑ Disable Copy Button for Displayed Passwords

☑ Disable Concurrent Logins from a Single User

☑ Embed Unique Identifier with each Page

☑ Unique Identifier Valid for only One Page Request

☑ Disable Explicit Web Application Accounts

☑ Store only the Authentication Token in the Cookie

☑ Force Logout on any Page Error

☐ Require RSA SecurID checks for all web application logins

☐ Enable RSA SecurID checks for web application logins based on permissions

☐ Use Simple Username for RSA login Checks

☑ Enable OATH token checks for web application logins based on permissions

☑ Prevent the requesting user from granting a password request

☑ Allow Client Certificates for User Authentication and Authorization

☑ Bypass Login Challenge for Client Certificate Identities

☑ Frequent Request Redirection (Reqeusts per second) 30

Win32 Management Console Access

- Ensure the host system does not permit interactive logon except by admins of the product.
- Ensure the host system does not permit network access except by admins of the product and the service account(s).
- Configure the Management console's console delegation to ensure only those specific users can launch the console. These users will still need to be administrators of the host system and be granted specific rights as defined in the **Database Access by Service Accounts** section for the **interactive user** in this document.

Database Access by Service Accounts

- The COM object accesses the database to read and write data and will require the ability to run stored procedures and query views. However, it has no hand in creating tables or stored procedures or views. As such it should not be the same user account as is used to run any of the scheduling services, as it does not require any target system access, nor does it require the same database privilege. It will only need:
  - Server connect/logon
  - Db_datareader
  - DB_datawriter
  - Execute on the database
- When the console is launched, it is responsible for ensuring the integrity of the database and its tables, views, stored procedures, etc. As such, the interactive login account here should have the following permissions:
  - Server connect/logon
  - Db_datareader
  - DB_datawriter
  - Execute on the database
  - Db_ddladmin
  - Optional - View server state server level permission - used for generation of indexes and index defrag and stats full scan operations
- The service accounts which access the program data store have limited rights compared to the interactive users but have more overall rights than the COM object identity, as it is used to manage target systems. As such, it should be a different account than that used for the COM identity. However, the database rights are consistent with those of the COM identity:
  - Server connect/logon
  - Db_datareader
  - DB_datawriter
  - Execute on the database

For Oracle, the rights are simplified to:

- CONNECT
- CREATE TRIGGER
- CREATE SEQUENCE
- CREATE TABLE
- CREATE VIEW

Service Accounts

- The COM object is not typically used to connect to or manage target systems. Though there are two functions which may be performed by the web site which do connect to systems, these functions are not typically allowed or performed. Accordingly, it should not be the same user account as is used to run any of the scheduling services.
- The service accounts which run the deferred processor or zone processor services perform actual system management such as password changes, propagation, and account elevations. As such, it should be a *different* account than that used for the COM identity.

Service Account Rights to Active Directory

On the topic of least privilege, what is required in AD depends on the accounts being managed and the password propagation being included.

Administrator rights of any member server or workstation will be required in order to reset the password of the accounts such as Administrator or anyone in the administrators group. Whether that happens based on local group membership or via AD group membership does not matter.

For Active Directory, domain admin or an administrator membership in the domain is not necessarily a requirement. These requirements depend on what type of account is being managed and if it will also be propagated to a domain controller service/task/etc.. Specifically…

- If managing the password of a regular user, the Privileged Identity service account needs only to be delegated “reset password." Regular user is defined as user who is not in a protected group, such as Account Operators, Server Operators, Administrators, Domain Admins, Enterprise Admins, etc.. This is a Microsoft restriction which can be changed by modifying specific security policies on the enterprise to permit low powered users to reset administrative accounts.
- If managing the password of a protected user (see above), then you must be at least in the administrators (domain local) group of the domain.
- For either of the above cases, if the service account to be managed will also have its password propagated to a domain controller, then the service account must be in at least the administrators (domain local) group or higher in the domain.

In another scenario, regarding service accounts, the deferred processor/zone processor may be configured to run as LocalSystem rather than a user account. This would then require the computer account be granted appropriate permissions in lieu of the service account credentials. The COM identity must, however, run as a real account.

Database Services

- Database should be hosted on its own servers and database files should be on separate spindles than the logs or the OS.
- Database server should be hosted in its own unique instance that is not shared with other applications.
- Database host should not host other database for other programs at all.
- Database instance should not be on a default port (1433 for MS SQL or 1521 for Oracle).
- Database should not host other unnecessary services.
- Use integrated authentication where possible.
- For MS SQL, configure Privileged Identity to use a non-default schema that is set to DBO (DB basic configuration). This ensures a consistent schema usage across all operations.
- Services should run as an AD managed service account if possible.
- Disable unnecessary database services such as SQL Agent and SQL browsing services.

IIS

- Configure web site to run in an application pool that is separate from any other web sites hosted on the web server host.
- Configure the web site to run under non-standard ports (other than 80 or 443).
- Web server can be a core installation of 2008 R2 server or later. BeyondTrust recommends using the most current version of Windows Server.
- Enforce the use of the latest TLS scheme rather than simple SSL (see Microsoft Security Response Center), or require IPSec.

Supplemental

Overall, apply high-security policies but be aware that certain provisions will have to be made to those policies to allow our service accounts to run and connect (see installation guide for more information). There are various other recommendations for High Availability and Disaster Recovery. The basic premise is put each role (web, app, DB) on its own server and add redundancy as is appropriate to each tier.

DoD Security Technical Implementation Guides (STIGs) also provide much guidance for hardening a Windows Server; it will be worthwhile to review these items as well:

- 2008 R2: https://nvd.nist.gov/ncp/checklist/377
- 2012 R2: https://nvd.nist.gov/ncp/checklist/560
- 2016: https://nvd.nist.gov/ncp/checklist/753