8.4.4 Engineering Workstation to Historian

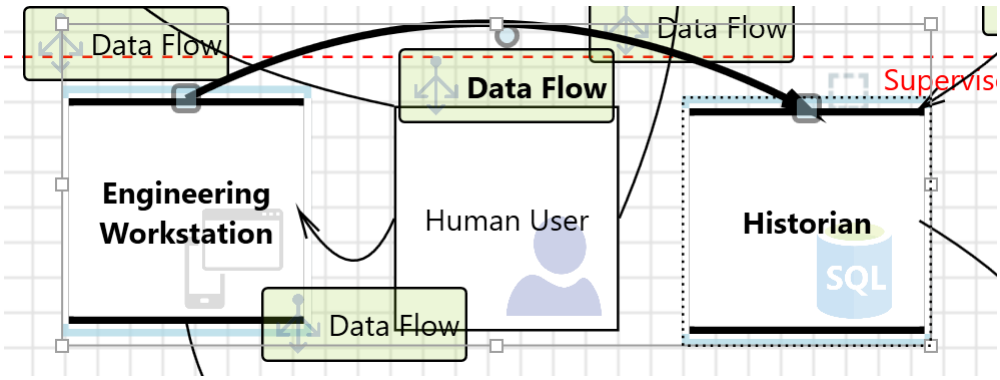

Figure 18: Engineering Workstation to Historian

Figure 18 depicts a data flow between an Engineering Workstation and Historian.

An adversary may leverage the lack of monitoring systems and trigger anomalous traffic to database

Category:	Tampering
Description:	An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to the database
Mitigation:	Threat Detection detects anomalous database activities indicating potential security threats to the database. It provides a new layer of security, enabling customers to detect and respond to potential threats by providing security alerts on anomalous activities. Users can explore the suspicious events using SQL Database Auditing to determine if they result from an attempt to access, breach, or exploit data in the database. Threat Detection makes it simple to address potential threats to the database without the need to be a security expert or manage advanced security monitoring systems

An adversary can gain access to sensitive PII data in Historian.

Category:	Information Disclosure
Description:	Additional controls like Transparent Data Encryption, Column Level Encryption, EKM etc. provide additional protection mechanisms to high value PII or HBI data.
Mitigation:	Use strong encryption algorithms to encrypt data in the database. Encryption algorithms define data transformations that unauthorised users cannot easily reverse. SQL Server allows administrators and developers to choose from among several algorithms, including DES, Triple DES, TRIPLE_DES_3KEY, RC2, RC4, 128-bit RC4, DESX, 128-bit AES, 192-bit AES, and 256-bit AES

An adversary can gain access to sensitive data by performing SQL injection

Category:	Information Disclosure
Description:	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. The malicious code is executed when the stored strings are subsequently concatenated into a dynamic SQL command.
Mitigation:	Ensure that login auditing is enabled on SQL Server. A SQL injection attack exploits vulnerabilities in input validation to run arbitrary commands in the database. It can occur when your application uses input to construct dynamic SQL statements to access the database. It can also occur if your code uses stored procedures that are passed strings that contain raw user input. The attacker can execute arbitrary commands in the database using the SQL injection attack. All SQL statements (including the SQL statements in stored procedures) must be parameterised. Parameterised SQL statements will accept characters with special meaning to SQL (like single quotes) without problems because they are strongly typed.

An adversary can gain unauthorised access to the database due to a lack of network access protection

Category:	Elevation Of Privilege
Description:	If there is no restriction at the network or host firewall level to access the database, then anyone can attempt to connect to the database from an unauthorised location
Mitigation:	Configure a Windows Firewall for Database Engine Access. Firewall systems help prevent unauthorised access to computer resources.

An adversary can gain unauthorised access to SQL database due to weak account policy

Category:	Elevation Of Privilege
Description:	Due to poorly configured account policies, adversaries can launch brute force attacks on Historian
Mitigation:	When possible, use Active Directory Authentication for Connecting to SQL Database. Windows Authentication uses Kerberos security protocol, provides password policy enforcement concerning complex validation for strong passwords, supports account lockout, and supports password expiration.

An adversary can gain unauthorised access to Historian due to weak account policy

Category:	Elevation Of Privilege
Description:	Due to poorly configured account policies, adversary can launch brute force attacks on Historian
Mitigation:	When possible, use Active Directory Authentication for Connecting to SQL Database. Windows Authentication uses Kerberos security protocol, provides password policy enforcement concerning complex validation for strong passwords, supports account lockout, and supports password expiration.

An adversary can gain unauthorised access to Historian due to loose authorisation rules

Category:	Elevation Of Privilege
Description:	Database access should be configured with roles and privileges based on the least privilege and need-to-know principle.
Mitigation:	Ensure that least-privileged accounts are used to connect to the Database server. Least-privileged accounts should be used to connect to the database. Application login should be restricted to the database and only execute selected stored procedures. The application’s login should have no direct table access.