Machine Translated by Google

Prácticas seguras de codificación de PLC: lista de las 20 principales

Versión 1.0 (15 de junio de 2021)

PLC Security TOP 20 LIST

1. Modularizar código PLC

Divida el código del PLC en módulos, utilizando diferentes bloques de funciones (subrutinas). Pruebe los módulos de forma independiente.

2. Seguimiento de los modos de funcionamiento

Mantenga el PLC en modo RUN. Si los PLC no están en modo RUN, debe haber una alarma para los operadores.

3. Deje la lógica operativa en el PLC siempre que sea factible

Deje la mayor cantidad posible de lógica operativa, p. ej., totalización o integración, directamente en el PLC. La HMI no recibe suficientes actualizaciones para hacer esto bien.

4. Use banderas de PLC como controles de integridad

Ponga contadores en las banderas de error del PLC para capturar cualquier problema matemático.

5. Utilice comprobaciones de integridad criptográficas y/o de suma de comprobación para el código de PLC

Use hashes criptográficos, o sumas de verificación si los hashes criptográficos no están disponibles, para verificar la integridad del código del PLC y generar una alarma cuando cambien.

6. Validar temporizadores y contadores

Si los valores de los temporizadores y contadores se escriben en el programa del PLC, el PLC debe validarlos para verificar que sean razonables y verificar los conteos hacia atrás por debajo de cero.

7. Validar y alertar por entradas/salidas emparejadas

Si tiene señales emparejadas, asegúrese de que ambas señales no se afirman juntas. Alarma al operador cuando ocurren estados de entrada/salida que no son físicamente factibles. Considere hacer que las señales emparejadas sean independientes o agregar temporizadores de retardo cuando alternar las salidas podría dañar los actuadores.

8. Valide las variables de entrada de HMI a nivel de PLC, no solo en HMI

El acceso de la HMI a las variables del PLC puede (y debe) restringirse a un rango de valores operativos válidos en la HMI, pero se deben agregar verificaciones cruzadas adicionales en el PLC para prevenir o alertar sobre valores fuera de los rangos aceptables que están programados en el IHM.

9. Validar direcciones indirectas

Valide los direccionamientos mediante el envenenamiento de los extremos de la matriz para detectar errores de vallas.

10. Asigne bloques de registro designados por función (leer/escribir/validar)

Asigne bloques de registro designados para funciones específicas con el fin de validar datos, evitar desbordamientos de búfer y bloquear escrituras externas no autorizadas para proteger los datos del controlador.

11. Instrumento para controles de plausibilidad

Instrumente el proceso de una manera que permita verificaciones de plausibilidad mediante la verificación cruzada de diferentes mediciones.

12. Valide las entradas en función de la plausibilidad física

Asegúrese de que los operadores solo puedan ingresar lo que es práctico o físicamente factible en el proceso. Establezca un temporizador para una operación con la duración que debería tomar físicamente. Considere alertar cuando haya desviaciones.
También alerta cuando hay inactividad inesperada.

Machine Translated by Google

13. Deshabilitar puertos y protocolos de comunicación innecesarios/no utilizados

Los controladores PLC y los módulos de interfaz de red generalmente admiten múltiples protocolos de comunicación que están habilitados de forma predeterminada. Deshabilite los puertos y protocolos que no son necesarios para la aplicación.

14. Restrinja las interfaces de datos de terceros

Restrinja el tipo de conexiones y datos disponibles para interfaces de terceros. Las conexiones y/o interfaces de datos deben estar bien definidas y restringidas para permitir solo capacidades de lectura/escritura para la transferencia de datos requerida.

15. Defina un estado de proceso seguro en caso de un reinicio del PLC

Defina estados seguros para el proceso en caso de que el PLC se reinicie (p. ej., energizar contactos, desenergizar, mantener el estado anterior).

16. Resuma los tiempos de ciclo del PLC y trátelos en la HMI

Resuma el tiempo de ciclo del PLC cada 2-3 segundos e informe a la HMI para su visualización en un gráfico.

17. Registre el tiempo de actividad del PLC y trátelo en la HMI

Registre el tiempo de actividad del PLC para saber cuándo se ha reiniciado. Tendencia y registro del tiempo de actividad en la HMI para diagnósticos.

18. Registre las paradas duras del PLC y trátelas en la HMI

Almacene los eventos de parada dura del PLC de fallas o paradas para que los sistemas de alarma HMI los recuperen para consultarlos antes de que el PLC se reinicie. Sincronización de tiempo para datos más precisos.

19. Supervise el uso de la memoria del PLC y realice una tendencia en la HMI

Mida y proporcione una línea de base para el uso de la memoria para cada controlador implementado en el entorno de producción y trátelo en la HMI.

20. Atrapa falsos negativos y falsos positivos para alertas críticas

Identifique alertas críticas y programe una trampa para esas alertas. Configure la trampa para monitorear las condiciones de activación y el estado de alerta para cualquier desviación.

Sobre el proyecto Programación segura de PLC

Machine Translated by Google

1. Modularizar código PLC

Divida el código del PLC en módulos, utilizando diferentes bloques de funciones (subrutinas). Pruebe los módulos de forma independiente.

Seguridad Objetivo	Grupo objetivo
Integridad de la lógica del PLC	Proveedor de productos

Guia

No programe toda la lógica del PLC en un solo lugar, por ejemplo, en el bloque de organización principal o en la rutina principal. En su lugar, divídalo en diferentes bloques de funciones (subrutinas) y controle su tiempo de ejecución y su tamaño en Kb.

Cree segmentos separados para la lógica que funciona de forma independiente. Esto ayuda en la validación de entrada, gestión de control de acceso, verificación de integridad, etc.

El código modularizado también facilita las pruebas y el seguimiento de la integridad de los módulos de código. Si el código dentro del módulo se ha probado meticulosamente, cualquier modificación a estos módulos se puede verificar con el hash del código original, por ejemplo, guardando un hash de cada uno de estos módulos (cuando esa es una opción en el PLC). De esta manera, los módulos pueden validarse durante el FAT/SAT o si la integridad del código está en duda después de un incidente.

Ejemplo

La lógica de la turbina de gas se divide en "arranque", "control de álabes de guía de entrada", "control de válvula de purga", etc. para que pueda aplicar la lógica estándar de forma sistemática. Esto también ayuda a solucionar problemas rápidamente si se produjera un incidente de seguridad.

Los bloques de funciones personalizados que se prueban rigurosamente se pueden reutilizar sin alteración (y recibir alertas si se realizan intentos de cambio) y se bloquean contra abuso/uso indebido con una contraseña/firma digital.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Facilita la detección de porciones de código recién agregadas que podrían ser maliciosas. Ayuda en la estandarización lógica, la consistencia y el bloqueo contra modificaciones no autorizadas.
Fiabilidad	Ayuda a controlar la secuencia de flujo del programa y evita los bucles, lo que podría hacer que la lógica no reaccione correctamente o se bloquee.
Mantenimiento	El código modular no solo es más fácil de depurar (los módulos se pueden probar de forma independiente), sino que también es más fácil de mantener y actualizar. Además, los módulos se pueden usar para PLC adicionales, lo que permite que se use e identifique un código común en PLC separados. Esto puede ayudar al personal de mantenimiento a reconocer rápidamente los módulos comunes durante la resolución de problemas.

Machine Translated by Google

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA002 - Ejecución **Técnica:** T0844 - Unidades de organización del programa
NIA 62443-3-3	**SR 3.4:** Software e integridad de la información
NIA 62443-4-2	**CR 3.4:** Software e integridad de la información
NIA 62443-4-1	**SI-2:** Estándares de codificación seguros
INGLETE CWE	**CWE-1120:** Complejidad de código excesiva **CWE-653:** Compartimentación insuficiente

Machine Translated by Google

2. Seguimiento de los modos de funcionamiento

Mantenga el PLC en modo RUN. Si los PLC no están en modo RUN, debe haber una alarma para los operadores.

Objetivo de seguridad	Grupo objetivo
Integridad de la lógica del PLC	Proveedor de servicios de integración/mantenimiento Propietario del activo

Guia

Si los PLC no están en modo EJECUTAR (por ejemplo, modo PROGRAMA), su código podría cambiarse para seguir el modo EJECUTAR. Algunos PLC tienen una suma de verificación para alertar sobre cambios en el código, pero si no la tienen, hay al menos un indicador indirecto de un posible problema al rastrear los modos operativos:

- Si los PLC no están en modo RUN, debe haber una alarma para los operadores. Si saben que se supone que alguien debe estar trabajando en ese sistema de control, pueden reconocer la alarma y seguir adelante.

- La HMI debe configurarse para volver a alertar al operador hacia el final del turno sobre la presencia de la alarma. El objetivo debe ser realizar un seguimiento de cualquier personal o contratista en la planta que realice trabajos que puedan afectar el proceso.

Caso de excepción: si la planta se encuentra en una fase de prueba o desarrollo, considere deshabilitar esta alarma, pero la planta debe aislarse de los niveles más altos de la red.

Ejemplo

Si el PLC no tiene un interruptor de hardware para cambiar los modos de operación, se recomienda al menos hacer uso de mecanismos de software que puedan restringir el cambio de código de PLC, por ejemplo, protección con contraseña en software de ingeniería para leer y escribir código de PLC.

¿Por qué?

Beneficioso para...?	¿Por qué?
Seguridad	El modo de funcionamiento (ejecutar/editar/escribir; para los PLC de Allen Bradley: EJECUTAR/PROGRAMAR/REMota) determina si el PLC puede ser manipulado. Si el interruptor de llave está en estado REmote, es técnicamente posible realizar cambios en el programa del PLC a través de las interfaces de comunicación incluso si el PLC está en funcionamiento.
Fiabilidad	/
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA009 - Función de inhibición de respuesta **Técnica:** T0858 - Utilizar/cambiar el modo de funcionamiento
ISA/CEI 62443-4-1	**SI-1 :** Revisión de la implementación de seguridad

Machine Translated by Google

3. Deje la lógica operativa en el PLC siempre que sea factible

Deje la mayor cantidad posible de lógica operativa, p. ej., totalización o integración, directamente en el PLC. La HMI no recibe suficientes actualizaciones para hacer esto bien.

Objetivo de seguridad	Grupo objetivo
Integridad de la lógica del PLC	Proveedor de productos Proveedor de servicios de integración/mantenimiento Propietario del activo

Guia

Las HMI brindan cierto nivel de capacidades de codificación, originalmente destinadas a ayudar a los operadores a mejorar la visualización y las alarmas, que algunos programadores han empleado para crear código que debería permanecer en el PLC para permanecer completo y auditable.

Calcular los valores lo más cerca posible del campo hace que estos cálculos sean más precisos. La HMI no recibe suficientes actualizaciones para realizar bien la totalización/integración. Además, siempre hay latencia entre HMI y PLC. Además, cuando el código está en el PLC y se reinicia una HMI, siempre puede recibir totalizadores/conteos de un PLC.

En particular, el código HMI que debe evitarse es cualquier cosa relacionada con la seguridad o las funciones de protección, como enclavamientos, temporizadores, retenciones o permisos.

Para analizar valores de datos de procesos a lo largo del tiempo, un historial de datos de procesos es la mejor opción que la HMI. Utilice consultas en una base de datos histórica de procesos para comparar valores totalizados (sobre un período, sobre un lote, sobre un ciclo de proceso) con los totales agregados localmente en la lógica del PLC. Alerta sobre una variación mayor que la que puede explicarse por diferencias en la granularidad de los datos.

Ejemplo

- ÿ Código para establecer condiciones para habilitar/deshabilitar botones: las acciones de habilitar/deshabilitar deben controlarse en la capa del PLC; de lo contrario, las acciones se pueden realizar en la HMI (o a través de la red) en el PLC, aunque no cumplan las condiciones (previstas).
- ÿ Los temporizadores para permitir acciones al operador (temporizador de retardo para arranques de motor consecutivos, temporizador para considerar válvulas cerradas/abiertas o motor parado) no deben colocarse en la capa HMI sino en el PLC que gobierna dicho motor/válvula.
- ÿ Los umbrales para las alarmas deben ser parte de los códigos del PLC aunque se muestren en las HMI.
- ÿ Tanque de agua con volumen cambiante: El PLC que controla el flujo dentro y fuera del tanque puede totalice fácilmente el volumen (y realice una validación cruzada de los totales). La HMI también podría hacer esto, pero primero necesitaría obtener los valores del PLC. Estos valores necesitarían marcas de tiempo precisas para obtener los totales correctos en caso de latencia o podrían perder valores si la HMI se reinicia.

¿Por qué?

Beneficioso para...?	¿Por qué?
Seguridad	**1.** Permite la consistencia en la verificación de cambios de código. La codificación HMI tiene su control de cambios aparte del PLC, generalmente no con el mismo rigor (especialmente en las fases de construcción y puesta en marcha), impidiendo que los propietarios del sistema tengan una visión completa e incluso perdiendo consideraciones importantes. Las HMI no incluyen "señales forzadas" o listas de valores modificados como PLC o SCADA, por lo que el nivel de HMI cambia

Machine Translated by Google

Beneficioso para...?	¿Por qué?
	son más difíciles de detectar, prácticamente imposibles de formar parte de un plan de gestión de cambios de autorización. **2.** Para un atacante, es más difícil manipular los totales distribuidos en muchos PLC que manipular todos los totales calculados en la HMI. **3.** Si una parte de las funciones de habilitación/deshabilitación no están en el PLC, los atacantes podrían manipular el PLC y las E/S sin tener que trabajar con la parte de HMI, ya que la información adecuada ya está ofuscada en la pantalla del operador.
Fiabilidad	**1.** Los cálculos son más eficientes y precisos si están más cerca del campo. Además, los totales y los conteos seguirán estando disponibles si se reinicia la HMI (los PLC no se reinician con tanta frecuencia y, por lo general, almacenan estos valores en la memoria no volátil). **2.** Las diferentes fuentes de entradas y enclavamientos pueden significar que no fracasos esperados. Puede haber diferentes tecnologías para HMIs en una planta (capa SCADA, pero también paneles de control de campo) y los cambios en una de ellas no se difundirán al resto de capas, lo que generará inconsistencias en la visualización y posibles fallas en la operación.
Mantenimiento	La codificación es fácil de entender y transferir de PLC a PLC, no tanto de HMI a HMI.

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro
NIA 62443-3-3	**SR 3.6 :** Salida determinista
NIA 62443-4-2	**CR 3.6 :** Salida determinista

Machine Translated by Google

4. Use banderas de PLC como controles de integridad

Ponga contadores en las banderas de error del PLC para capturar cualquier problema matemático.

Objetivo de seguridad	Grupo objetivo
Integridad de la lógica del PLC	Proveedor de productos Proveedor de servicios de integración/mantenimiento

Guia

Si el código del PLC funcionaba bien pero de repente se divide por cero, investigue. Si algo se comunica entre pares desde otro PLC y la función/lógica divide por cero cuando no se esperaba, investigue.

La mayoría de los programadores ignorarán el problema como un error matemático o, peor aún, podrían suponer que su código es perfecto y dejar que el PLC entre en un estado de falla grave. Durante el desarrollo del código, los ingenieros deben probar y validar sus módulos de código (fragmentos o rutinas) ingresando datos fuera de los límites esperados.
Esto puede denominarse Prueba de nivel de unidad.

Asigne diferentes segmentos de memoria bloqueados para el firmware, la lógica y la pila de protocolos. Pruebe la pila de protocolos para casos de abuso. Los casos de abuso pueden ser condiciones de marca peculiares en el encabezado de un paquete.

Ejemplo

Las fallas de PLC causadas por datos fuera de los límites son muy comunes. Esto sucede, por ejemplo, cuando un valor de entrada hace que los índices de la matriz se salgan de los límites, o los temporizadores tengan preajustes negativos, o se dividan por cero excepciones.

Las banderas típicas de interés son

- ÿ dividir por cero
- ÿ desbordamiento del contador
- ÿ contador negativo o temporizador preestablecido
- ÿ Desbordamiento de exploración de E/S

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Los ataques a los PLC pueden incluir cambiar su lógica, activar un nuevo programa, probar un nuevo código, cargar una nueva receta de proceso, insertar lógica auxiliar para enviar mensajes o activar alguna función. Dado que la mayoría de los PLC no brindan verificaciones de integridad criptográfica, las banderas pueden ser un buen indicador si ocurre uno de los cambios lógicos anteriores.
Fiabilidad	Las banderas que se toman en serio pueden evitar que el PLC se ejecute con errores de programación o de E/S. Además, si ocurre un error, la fuente de la falla es más obvia.
Mantenimiento	/

Machine Translated by Google

Referencias

Táctica de mapeo estándar / marco :	
MITRE ATT&CK para ICS	TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro **SR 3.5:**
NIA 62443-3-3	Validación de entrada **SR 3.6:** Salida determinista **CR 3.5:** Validación de entrada **CR 3.6:** Salida
NIA 62443-4-2	determinista **SI-2:** Estándares de codificación segura **SVV-1:** Prueba de requisitos de seguridad
NIA 62443-4-1	**CWE-128:** Wrap-around **CWE-190:** Integer Overflow **CWE-369 :** Dividir por cero **CWE-754:**
INGLETE CWE	Comprobación incorrecta de condiciones inusuales o excepcionales

Machine Translated by Google

5. Utilice comprobaciones de integridad criptográficas y/o de suma de comprobación para PLC código

Use hashes criptográficos, o sumas de verificación si los hashes criptográficos no están disponibles, para verificar la integridad del código del PLC y generar una alarma cuando cambien.

Grupo destinatario del objetivo de seguridad	
Integridad de la lógica del PLC	Proveedor de productos Proveedor de servicios de integración/mantenimiento Propietario del activo

Guia

A) Sumas de verificación

Cuando los hashes (criptográficos) no son factibles, las sumas de verificación pueden ser una opción. Algunos PLC generan una suma de verificación única cuando el código se descarga en el hardware del PLC. El Checksum debe ser documentado por el fabricante/integrador después del SAT y ser parte de las condiciones de garantía/servicio.

Si la función de suma de verificación no está disponible de forma nativa en el controlador, también se puede generar en el EWS/HMI y probar, por ejemplo, una vez al día para comparar con el hash del código original en el PLC para verificar que coincidan. Si bien esto no proporcionará alertas en tiempo real, es lo suficientemente bueno para rastrear si alguien está intentando cambios en el código del PLC.

El valor de la suma de verificación también se puede mover a un registro de PLC y configurar para una alarma cuando cambia, el valor se puede enviar a los historiadores, etc.

B) hashes

Las CPU de PLC generalmente no tienen la capacidad de procesamiento para generar o verificar hashes mientras se ejecutan. Intentar un hash en realidad podría causar que el PLC se bloquee. Pero el software de ingeniería del PLC podría calcular hashes del código del PLC y guardarlos en el PLC o en algún otro lugar en el sistema de control

Ejemplo

Proveedores de PLC que se sabe que tienen características de suma de verificación:

- ÿ Siemens (ver ejemplo)
- ÿ Rockwell

Además, se puede usar software externo para generar sumas de verificación:

- ÿ Versión perro
- ÿ Guardián de Activos
- ÿ PAS

Ejemplo de implementación de Siemens

Ejemplo para crear sumas de verificación en Siemens S7-1500 PLC:

GetChecksum-Function Block lee la suma de verificación real y con un script liviano, la "Suma de verificación SAT" se puede almacenar como referencia. Una desviación de la suma de verificación de referencia se puede almacenar con la función de registro de datos.

Machine Translated by Google

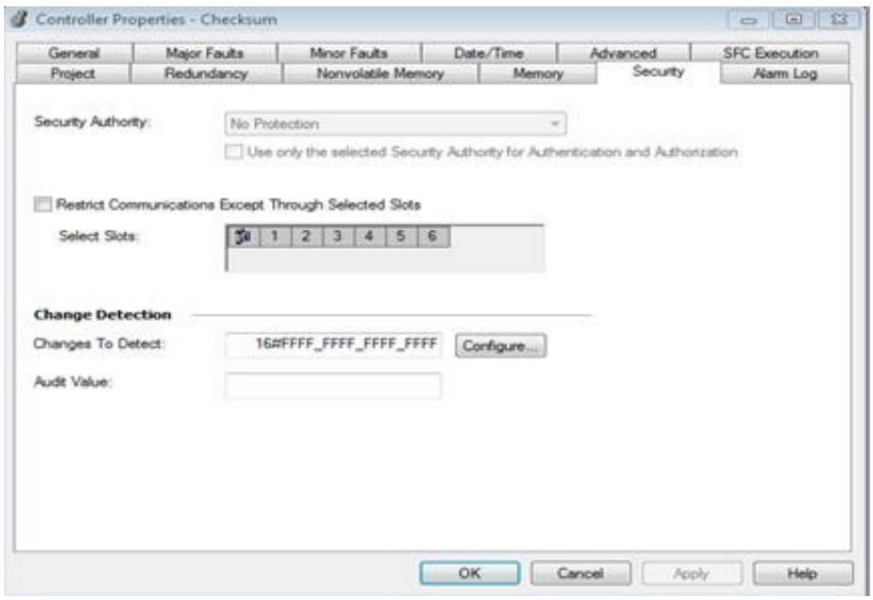
	Date	UTC Time	Referenz	Aktuell
1	11/21/2019	9:55:11	84 2A 76 DF 5B 31 F4 16	FF 2C EA 71 44 D7 81 04
2	11/21/2019	9:57:33	FF 2C EA 71 44 D7 81 04	FF 2C EA 71 44 D7 81 04
3	11/21/2019	9:58:17	FF 2C EA 71 44 D7 81 04	5B 7C 57 7E E2 3E EF C3
4	11/21/2019	9:58:36	FF 2C EA 71 44 D7 81 04	5B 7C 57 7E E2 3E EF C3
5	11/21/2019	9:58:44	5B 7C 57 7E E2 3E EF C3	5B 7C 57 7E E2 3E EF C3

Ejemplo de implementación de Rockwell:

Este es un ejemplo parcial de cómo una organización puede desarrollar un nivel de capacidad de detección de cambio de programa de PLC dentro de su entorno ICS. Este ejemplo es específicamente para un PLC ControlLogix de Rockwell Automation y no está completo; sin embargo, ilustra cómo recuperar el estado del procesador del PLC en un registro dentro del PLC. Una vez en un registro en el PLC, la organización puede usarlo para crear una alarma de cambio de configuración para mostrar en una HMI, transmitir la información de estado sin procesar a una HMI para tendencias y monitoreo, o enviarla a un Historiador para captura a largo plazo.

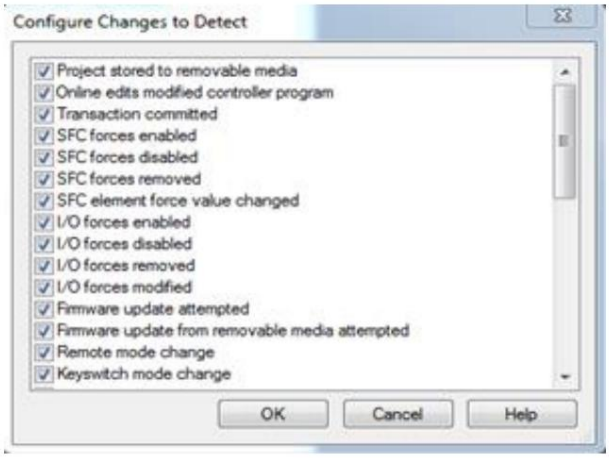
Esta práctica brinda una oportunidad, utilizando las herramientas y capacidades existentes, para obtener una conciencia situacional de cuándo cambian los activos cibernéticos críticos. Depende de la organización completar el uso de este ejemplo en un método que funcione mejor en su entorno.

1. En el cuadro de diálogo Propiedades del controlador, seleccione el botón de configuración en "Cambiar para detectar"

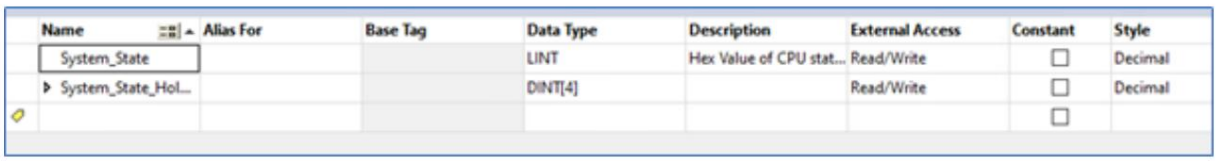


Machine Translated by Google

2. Dentro de la ventana de selección, elija todos los elementos a monitorear

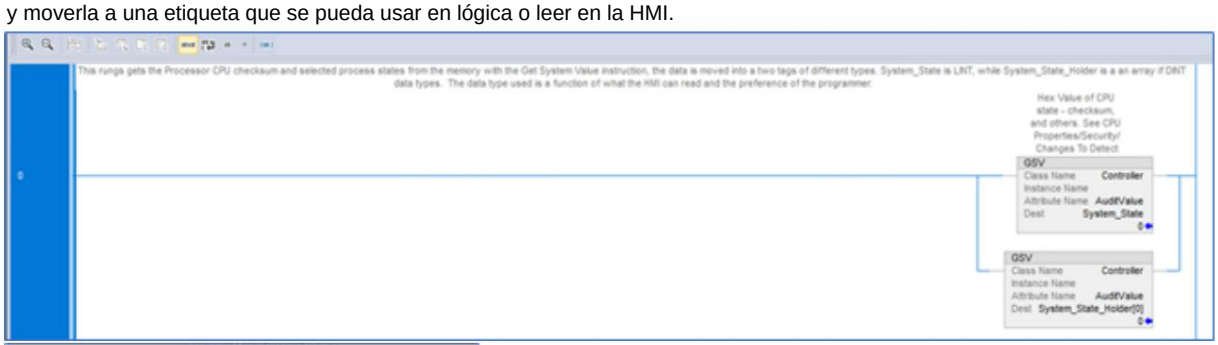


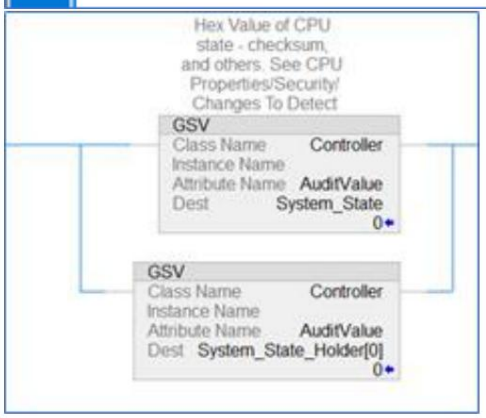
3. Cree una etiqueta para recibir la información del estado del procesador. Esta etiqueta puede ser de tipo "LINT" o una matriz de 2 palabras de tipo "DINT"



Name	Alias For	Base Tag	Data Type	Description	External Access	Constant	Style
System_State			LINT	Hex Value of CPU stat...	Read/Write	☐	Decimal
System_State_Hol...			DINT[4]		Read/Write	☐	Decimal
						☐	

4. Use la instrucción Get System Values (GSV) para obtener la información del estado del procesador de la memoria y moverla a una etiqueta que se pueda usar en lógica o leer en la HMI.





Machine Translated by Google

¿Por qué?

Beneficioso para...?	¿Por qué?
Seguridad	Saber si el código del PLC fue manipulado es esencial tanto para detectar un compromiso como para verificar si un PLC es seguro para operar después de un posible compromiso.
Fiabilidad	Los hashes o las sumas de verificación también pueden ser un medio para verificar si el PLC (todavía) está ejecutando el código aprobado por el integrador/fabricante.
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA002 - Ejecución, TA010 - Deterioro del control de procesos **Técnica:** T0873 - Infección de archivo de proyecto, T0833 - Modificar Lógica de Control
NIA 62443-3-3	**SR 3.4 :** Software e integridad de la información
NIA 62443-4-2	**CR 3.4 :** Software e integridad de la información **EDR 3.12 :** Suministro de raíces de confianza del proveedor del producto
NIA 62443-4-1	**SI-1 :** Revisión de la implementación de seguridad Pruebas de requisitos de seguridad **SVV-1**
INGLETE CWE	**CWE-345:** Verificación insuficiente de la autenticidad de los datos • **(niño) CWE-353:** Falta soporte para verificación de integridad • **(niño) CWE-354:** Validación incorrecta del valor de verificación de integridad

Machine Translated by Google

6. Validar temporizadores y contadores

Si los valores de los temporizadores y contadores se escriben en el programa del PLC, el PLC debe validarlos para verificar que sean razonables y verificar los conteos hacia atrás por debajo de cero.

Objetivo de seguridad	Grupo objetivo
Integridad de las variables del PLC	Proveedor de servicios de integración/mantenimiento Propietario del activo

Guia

Los temporizadores y contadores técnicamente se pueden preestablecer en cualquier valor. Por lo tanto, el rango válido para preestablecer un temporizador o contador debe restringirse para cumplir con los requisitos operativos.

Si los dispositivos remotos, como una HMI, escriben valores de temporizador o contador en un programa:

- ÿ no permita que la HMI escriba en el temporizador o contador directamente, sino que pase por una lógica de validación
- ÿ validar valores predeterminados y de tiempo de espera en el PLC

La validación de entradas de temporizadores y contadores es fácil de hacer directamente en el PLC (sin necesidad de ningún dispositivo de red capaz de Inspección Profunda de Paquetes), ya que el PLC “sabe” cuál es el estado o contexto del proceso. Puede validar "qué" obtiene y "cuándo" obtiene los comandos o puntos de ajuste.

Ejemplo

Durante el inicio del PLC, los temporizadores y contadores suelen estar preestablecidos en ciertos valores.

Si hay un temporizador que activa las alarmas a los 1,3 segundos, pero ese temporizador está preestablecido malintencionadamente en 5 minutos, es posible que no active la alarma.

Si hay un contador que hace que un proceso se detenga cuando llega a 10.000 pero está configurado en 11.000 desde el principio, es posible que el proceso no se detenga.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Si las E/S, los temporizadores o los ajustes preestablecidos se escriben directamente en las E/S, sin ser validados por el PLC, se evade la capa de validación del PLC y se asigna a la HMI (u otros dispositivos de red) un nivel de confianza injustificado.
Fiabilidad	El PLC también puede validar cuando un operador preestablece accidentalmente valores incorrectos de temporizador o contador.
Mantenimiento	Tener rangos válidos para temporizadores y contadores documentados y validados automáticamente puede ayudar al actualizar la lógica.

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica :** TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro
NIA 62443-3-3	**SR 3.5 :** Validación de entrada
NIA 62443-4-2	**CR 3.5 :** Validación de entrada
NIA 62443-4-1	**SI-2 :** Estándares de codificación segura **SVV-1 :** Prueba de requisitos de seguridad

Machine Translated by Google

7. Validar y alertar por entradas/salidas emparejadas

Si tiene señales emparejadas, asegúrese de que ambas señales no se afirman juntas. Alarma al operador cuando ocurren estados de entrada/salida que no son físicamente factibles. Considere hacer que las señales emparejadas sean independientes o agregar temporizadores de retardo cuando alternar las salidas podría dañar los actuadores.

Seguridad Objetivo	Grupo objetivo
Integridad de las variables del PLC Resiliencia	Proveedor de productos Proveedor de servicios de integración/mantenimiento

Guia

Las entradas o salidas emparejadas son aquellas que físicamente no pueden ocurrir al mismo tiempo; son mutuamente excluyentes. Aunque las señales emparejadas no se pueden afirmar al mismo tiempo a menos que haya una falla o una actividad maliciosa, los programadores de PLC a menudo no evitan que ocurra esa afirmación.

La validación es más fácil de hacer directamente en el PLC, porque el PLC es consciente del estado o contexto del proceso. Las señales emparejadas son más fáciles de reconocer y rastrear si tienen direcciones secuenciales (por ejemplo, entrada 1 y entrada 2).

Otro escenario en el que las entradas o salidas emparejadas podrían causar problemas es cuando no se afirman al mismo tiempo, sino que se alternan rápidamente de una manera que daña los actuadores.

Ejemplo

Ejemplos de señales emparejadas:

- ÿ INICIO y PARADA
  - o Inicio y parada independientes: configure el inicio y la parada como salidas discretas en lugar de tener una sola salida que se pueda activar o desactivar. Por diseño, esto no permite disparadores simultáneos. Para un atacante, es mucho más complicado activar/desactivar rápidamente si se deben configurar dos salidas diferentes.
  - o Temporizador para reiniciar: Considere también agregar un temporizador para reiniciar después de que se emita una parada para evite el apagado rápido de las señales de inicio/parada.
- ÿ AVANCE y RETROCESO
- ÿ ABRIR y CERRAR

Ejemplos para alternar señales emparejadas que podrían ser dañinas:

Si el PLC/MCC acepta una entrada discreta, esto proporciona una opción fácil para que un atacante cause daño físico a los actuadores. El escenario bien conocido para alternar salidas para causar daños sería un MCC, pero esta práctica se aplica a todos los escenarios en los que alternar salidas podría causar daños. Una prueba de concepto en la que cambiar rápidamente las salidas podría causar un daño real fue la prueba del generador Aurora en 2007 realizada por el Laboratorio Nacional de Idaho, donde cambiar las salidas fuera de sincronismo provocó daños en el interruptor automático.

Machine Translated by Google

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	1. Si los programas de PLC no tienen en cuenta lo que sucederá si ambas señales de entrada emparejadas se afirman al mismo tiempo, este es un buen vector de ataque. 2. La afirmación de ambas señales de entrada emparejadas es una advertencia de que hay un error operativo, un error de programación o algo malicioso está sucediendo. 3. Esto evita un escenario de ataque donde el daño físico puede ser causado a los actuadores.
Fiabilidad	1. Las señales de entrada emparejadas pueden indicar que un sensor está roto o mal cableado o que hay un problema mecánico como un interruptor atascado. 2. Alternar rápidamente el inicio y la parada también se puede hacer por error, por lo que esto también evita daños que podrían ocurrir sin darse cuenta.
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro, T0806 - E/S de fuerza bruta **SR 3.5:**
NIA 62443-3-3	Validación de entrada **SR 3.6:** Salida determinista
NIA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
NIA 62443-4-1	**SI-2:** Estándares de codificación seguros **SVV-1:** Pruebas de requisitos de seguridad
INGLETE CWE	**CWE-754:** Comprobación incorrecta de condiciones inusuales o excepcionales

Machine Translated by Google

8. Valide las variables de entrada de HMI a nivel de PLC, no solo en HMI

El acceso de la HMI a las variables del PLC puede (y debe) restringirse a un rango de valores operativos válidos en la HMI, pero se deben agregar verificaciones cruzadas adicionales en el PLC para prevenir o alertar sobre valores fuera de los rangos aceptables que están programados en el IHM.

Objetivo de seguridad	Grupo objetivo
Integridad de las variables del PLC	Proveedor de productos Proveedor de servicios de integración/mantenimiento

Guia

La validación de entrada podría incluir verificaciones fuera de los límites para valores operativos válidos, así como valores válidos en términos de tipos de datos que son relativos al proceso.

Si una variable de PLC recibe un valor que está fuera de los límites, proporcione lógica de PLC a

- ingrese un **valor predeterminado** para esa variable que no afecta negativamente el proceso y puede utilizarse como indicador de alertas, o
- ingrese el **último valor correcto** a ese valor y registre el evento para un análisis posterior.

Ejemplo

Ejemplo 1

Una operación requiere que un usuario ingrese un valor en una HMI para la presión de la válvula. Los rangos válidos para esta operación son 0-100, y la entrada del usuario se pasa de la función de entrada del usuario en la HMI a la variable V1 en el PLC. En este caso,

1. La entrada de la HMI a la variable V1 tiene un rango restringido de 0-100 (dec.) programado en la HMI.
2. El PLC tiene una lógica de verificación cruzada que establece:

SI V1 < 0 O SI V1 > 100, AJUSTAR V1 = 0.

Esto proporciona una respuesta positiva de un valor presumiblemente seguro a una entrada no válida para esa variable.

Ejemplo 2

Una operación requiere la entrada del usuario para los umbrales de medición de una variable que siempre debe estar dentro de un rango de datos INT2. La entrada del usuario se pasa de la HMI a la variable V2 en el PLC, que es un registro de datos de 16 bits.

1. La entrada HMI a la variable V2 tiene un rango restringido de -32768 a 32767 (dec.) programado en el IHM.
2. El PLC tiene una lógica de verificación cruzada de tipo de datos que monitorea la variable de desbordamiento (V3), que existe justo después de V2 en la estructura de memoria del PLC:

SI V2 = -32768 O SI V2 = 32767 Y V3 != 0,

CONFIGURAR V2 = 0 Y CONFIGURAR V3 = 0 Y CONFIGURAR DataTypeOverflowAlarm = TRUE.

Ejemplo 3

Escale PV (Valor de proceso), SP (Punto de ajuste) y CV (Variable de control) para PID (Controlador proporcional, integral, derivado) a unidades consistentes o sin procesar para eliminar errores de escala que causan problemas de control.
Un escalado incorrecto puede dar lugar a casos de abuso involuntario.

Machine Translated by Google

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	**1.** Si bien las HMI generalmente brindan algún tipo de validación de entrada, un actor malicioso puede crear o reproducir paquetes modificados para enviar valores arbitrarios a las variables en el PLC que están abiertas a influencias externas (abiertas a valores pasados desde una HMI, por ejemplo). **2.** Los protocolos de PLC generalmente se comercializan como protocolos "abiertos" y publicado al público en general, por lo que la creación de malware que utiliza información de protocolo "abierto" puede ser trivial de desarrollar. El mapeo de variables de PLC generalmente puede ocurrir a través del análisis de tráfico durante las fases de reconocimiento de un ataque, lo que proporciona al intruso la información necesaria para enviar tráfico malicioso al objetivo y, por lo tanto, manipular un proceso con herramientas no autorizadas. La verificación cruzada de los valores pasados al PLC antes de implementar esos datos en el proceso garantiza rangos de datos válidos y mitiga un valor no válido en esas ubicaciones de memoria al establecer rangos seguros a la fuerza cuando se detecta un valor fuera de los límites durante el transcurso del PLC. escanear.
Fiabilidad	/
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro
NIA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
NIA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
NIA 62443-4-1	**SI-2:** Estándares de codificación seguros **SVV-1:** Pruebas de requisitos de seguridad
INGLETE CWE	**CWE-1320:** Protección inadecuada para alertas de nivel de señal fuera de límites

Machine Translated by Google

9. Validar direcciones indirectas

Valide los direccionamientos mediante el envenenamiento de los extremos de la matriz para detectar errores de vallas.

Objetivo de seguridad Grupo objetivo Proveedor de productos Integración /	
Integridad de las variables del PLC	Mantenimiento Proveedor de servicios

Orientación

Una indirección es el uso del valor de un registro en otro registro. Hay muchas razones para usar indirectas.

Ejemplos de indirectas necesarias son:

- ÿ Unidades de frecuencia variable (VFD) que activan diferentes acciones para diferentes frecuencias usando tablas de búsqueda.
- ÿ Para decidir qué bomba empezar a funcionar primero en función de sus tiempos de funcionamiento actuales

Los PLC normalmente no tienen un indicador de "fin de una matriz", por lo que es una buena idea crearlo en el software; el objetivo es evitar operaciones de PLC inusuales/no planificadas.

Ejemplo de programación de lista de instrucciones (IL) El enfoque puede convertirse en unos pocos bloques de funciones y posiblemente incluso reutilizarse para otras aplicaciones.

1. Crear máscara de matriz

Compruebe si la matriz tiene un tamaño binario. Si no tiene un tamaño binario, cree una máscara del siguiente tamaño en una escala binaria. por ejemplo, si necesita 5 registros (no de tamaño binario):

[21 31 41 51 61]

definir una matriz de 8:

[xx 21 31 41 51 61 x]

A continuación, tome el valor del índice para recoger la indirección; en este ejemplo, es 3.

Advertencia: ¡el índice comienza en 0!

[21 31 41 **51** 61] _____^

Índice: 3

agregue un desplazamiento para compensar el extremo envenenado. El desplazamiento puede ser 1 o superior, en este caso es 2:

[xx 21 31 41 **51** 61 x] _____^

Índice incluyendo compensación: 3 + 2 = 5

y luego Y el índice que incluye el desplazamiento con una máscara que es igual al tamaño de la matriz.
En este ejemplo, el tamaño de la matriz es 8, por lo que el índice es 7, por lo que la máscara sería 0x07. La máscara se asegura de que el índice máximo que puede obtener sea 7, por ejemplo:

Machine Translated by Google

6 Y 0x07 devolvería 6.
7 AND 0x07 devolvería 7 8 AND 0x07
devolvería 0.
9 Y 0x07 devolvería 1.

Esto asegura que siempre aborde un valor en la matriz.

2. Insertar extremos envenenados

El envenenamiento de extremos es opcional. Podría detectar indireccionamientos manipulados sin el envenenamiento, pero el envenenamiento ayuda a detectar errores de poste de cerca porque obtiene un valor que no tiene sentido.

El punto es que en el índice 0 de la matriz, debe haber un valor que no sea válido, como -1 o 65535.
Este es “el final envenenado”. Del mismo modo, en los últimos elementos de la matriz haces lo mismo:

Entonces, para la matriz anterior, la versión envenenada podría verse así:

[-1 -1 21 31 41 51 61 -1]

3. Registre el valor de la dirección de direccionamiento indirecto sin máscara

Luego registre el valor de la dirección indirecta sin máscara AND y desplazamiento:
En este ejemplo, registraría 51 para el índice 3.

[21 31 41 **51** 61]
_____^
_____Índice 3

4. Ejecute la máscara AND y compare los valores (= validación indirecta)

Compare su valor registrado con el valor después de haber realizado la compensación y la máscara AND.

4a. Caso A: Indirección correcta

Primero, compensar:
Índice + Desplazamiento = 3 + 2 = 5

En segundo lugar, máscara:
5 Y 0x07 = 5

Tercero, verificación de direccionamiento indirecto: [-1 -1 21 31 41 **51** 61 -1] _____^ Índice que incluye compensación: 5 Valor = 51 es igual al valor registrado, por lo que todo está bien.

4b. Caso B: Desvío manipulado

Si ahora tuviera una indirección manipulada, digamos 7, veamos qué sucede:

Machine Translated by Google

Primero, compensar:

Índice + Desplazamiento = 7 + 2 = 9

En segundo lugar, máscara:

9 Y 0x07 = 1

Tercero, verificación de direccionamiento indirecto:

[-1 **-1** 21 31 41 51 61 -1]

_____^

Índice incluyendo compensación: **1**

Valor = **-1** no es igual al valor registrado y también indica su extremo envenenado, por lo que sabría que su direccionamiento indirecto está manipulado.

5. Ejecutar falla/alerta del programador

Si este valor validado es diferente del registrado, entonces sabrá que algo anda mal. Activar una alarma de calidad del software.

Luego, verifique el valor de direccionamiento indirecto. Si es un valor envenenado, debe generar otra alarma de calidad del software. Esta es una indicación de un error de poste de cerca.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	La mayoría de los PLC no tienen ninguna característica para manejar índices fuera de los límites para arreglos. Hay **dos escenarios potencialmente peligrosos** que pueden derivarse de errores de direccionamiento indirecto: **Primero,** si una indirección conduce a la lectura de un registro incorrecto, el programa se ejecuta con valores incorrectos. **En segundo lugar,** si un desvío incorrecto lleva a escribir en el registro incorrecto, el programa sobrescribe el código o los valores que desea conservar. En ambos casos, los errores de direccionamiento indirecto pueden ser difíciles de detectar y pueden tener serios impactos. Pueden ser causados por un error humano pero también ser insertados maliciosamente.
Fiabilidad	Identifica errores humanos no maliciosos en la programación.
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro
NIA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
NIA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
NIA 62443-4-1	**SI-2:** Estándares de codificación seguros **SVV-1:** Pruebas de requisitos de seguridad
INGLETE CWE	**CWE-129:** Validación incorrecta del índice de matriz

Machine Translated by Google

10. Asigne bloques de registro designados por función (leer/escribir/validar)

Asigne bloques de registro designados para funciones específicas con el fin de validar datos, evitar desbordamientos de búfer y bloquear escrituras externas no autorizadas para proteger los datos del controlador.

Objetivo de seguridad	Grupo objetivo
Integridad de las variables del PLC	Proveedor de productos Proveedor de servicios de integración/mantenimiento

Guia

La memoria temporal, también conocida como memoria de borrador, es un área de la memoria fácilmente explotable si no se sigue esta práctica. por ejemplo, simplemente escribir en un registro "Modbus" que está fuera de los límites podría llevar a sobrescribir los registros de memoria utilizados para cálculos temporales.

En general, otros dispositivos pueden acceder a la memoria de registro a través de la red del PLC para operaciones de lectura y escritura. Algunos registros pueden ser leídos por una HMI, y otros pueden ser escritos por un sistema SCADA, etc. Tener arreglos de registros específicos para una determinada aplicación también facilita (en el controlador o se usa un firewall externo) configurar el acceso de solo lectura desde otro dispositivo/HMI.

Ejemplos de funciones para las que tienen sentido los bloques de registro designados son:

- leyendo
- escritura (desde HMI/controlador/otro dispositivo externo)
- validar escrituras
- cálculos

Asegurar escrituras externas en los registros permitidos también ayuda a evitar errores de reinicio de la memoria principal debido a una ejecución fuera de límite o intentos maliciosos. Estos bloques de registro designados se pueden usar como búferes para escrituras de E/S, temporizadores y contadores al validar que el búfer esté completamente escrito (no contiene parte de datos antiguos y parte de datos nuevos) y validando todos los datos en el búfer.

Fondo:

La memoria principal y la memoria de registro se utilizan de manera diferente. La memoria principal se usa para almacenar la lógica del programa que se está ejecutando actualmente, mientras que la memoria de registro se usa como una memoria temporal por la lógica que se está ejecutando actualmente. Aunque la memoria de registro es temporal, dado que está siendo utilizada por la lógica de ejecución, seguramente contendrá algunas variables importantes que afectarían la lógica principal.

Ejemplo

Ejemplos de lo que podría pasar si no se implementa esta práctica:
(Referencia: GPH Sandaruwan, PS Ranaweera, Vladimir A. Oleshchuk, PLC Security and Critical Infrastructure Protection):

- Siemens normalmente usa la memoria del bloc de notas en el área de banderas desde la bandera 200.0 hasta la bandera 255.7. Si se cambia un bit dentro de esta área, existe la posibilidad de un mal funcionamiento grave del PLC en función de la importancia de ese bit o byte.
- Suponga que un atacante puede obtener acceso a una de las máquinas en la red PLC y infectar esa máquina con un gusano que es capaz de escribir valores arbitrarios en el registro

Machine Translated by Google

memoria. Dado que los valores de la memoria de registro cambiaron arbitrariamente, puede cambiar el valor de la presión.

- La ejecución de la lógica establecerá un nuevo valor basado en el cambio y eso puede causar que el sistema exceda sus márgenes de seguridad y posiblemente conduzca a una falla.

Ejemplos para implementar esta práctica:

- En un escenario donde hay una zona de seguridad (pero el DCS puede leer), el firewall puede registrar cualquier intento de "escritura" con una regla de que estos registros son de SÓLO LECTURA en la zona de seguridad.
- En otro escenario, podría haber algunos registros con capacidad de escritura y otros de solo lectura, pero tener todos los registros de SÓLO LECTURA en una sola matriz hace que sea más fácil configurarlos en el controlador (o en un firewall).

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Facilita la protección de los datos del controlador por función (lectura/escritura/validación). Hace que sea más fácil para los firewalls sensibles al protocolo hacer su trabajo: las reglas se vuelven más simples porque es muy claro a qué bloques de registros puede acceder la HMI. Facilita la gestión de las reglas (más sencillas) en el cortafuegos Hacer cambios no autorizados en la memoria temporal interna es una tarea sencilla. vulnerabilidad explotable (By-pass Logic Attack). Cuando las entradas y salidas a las rutinas del PLC se validan correctamente, cualquier cambio (por un actor malintencionado o por error) puede detectarse fácilmente en lugar de permanecer en la secuencia lógica durante mucho tiempo y generar errores/causar problemas más adelante en la ejecución.
Fiabilidad	Hace que las lecturas y escrituras sean más rápidas porque el número de transacciones es reducido. Incluso los cambios autorizados y los errores de programación pueden causar un mal funcionamiento si la memoria temporal no está protegida. Los errores de red y comunicaciones en mensajes largos pueden generar errores no deseados si no se verifica la validez de los datos antes del procesamiento.
Mantenimiento	Los errores de programación que provocan la escritura en la memoria temporal pueden dificultar la búsqueda de errores, por lo que el problema se puede evitar asignando registros específicos para las escrituras.

Machine Translated by Google

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica :** TA009 - Función de inhibición de respuesta, TA010 - Deterioro del control de procesos **Técnica:** T0835 - Manipular imagen de E/S , T0836 - Modificar parámetro
NIA 62443-3-3	**SR 3.4 :** Software e integridad de la información **SR 3.5 :** Validación de entrada **SR 3.6 :** Salida determinista
NIA 62443-4-1	**SD-4:** mejores prácticas de diseño seguro **SI-1:** Revisión de la implementación de seguridad **SI-2 :** Estándares de codificación segura **SVV-1 :** Prueba de requisitos de seguridad
NIA 62443-4-2	**CR 3.4 :** Software e integridad de la información **CR 3.5 :** Validación de entrada **CR 3.6 :** Salida determinista
INGLETE CWE	**CWE-787:** Escritura fuera de los límites **CWE-653:** Compartimentación insuficiente

Machine Translated by Google

11. Instrumento para controles de plausibilidad

Instrumente el proceso de una manera que permita verificaciones de plausibilidad mediante la verificación cruzada de diferentes mediciones.

Objetivo de seguridad	Grupo objetivo
Integridad de los valores de E/S	Proveedor de productos Proveedor de servicios de integración/mantenimiento

Guia

Hay diferentes formas de utilizar la plausibilidad física para validar las mediciones:

a) Comparar mediciones integradas e independientes del tiempo

Las comprobaciones de plausibilidad se pueden realizar integrando o diferenciando valores dependientes del tiempo durante un período de tiempo y comparándolos con mediciones independientes del tiempo.

b) Comparar diferentes fuentes de medición

Además, medir el mismo fenómeno de diferentes maneras puede ser una buena verificación de plausibilidad.
Las diferentes fuentes de medición no necesariamente tienen que ser diferentes sensores físicos, sino que también pueden significar el uso de canales de comunicación alternativos (ver ejemplos).

Ejemplo

a) Comparar mediciones integradas e independientes del tiempo

- ÿ Bomba dosificadora y medidor de nivel del tanque: el cambio volumétrico debe ser igual al flujo integrado.
- ÿ Quemador en una caldera: el calor calórico añadido debe igualar el aumento de temperatura.

b) Comparar diferentes fuentes de medición

- ÿ Usar la velocidad del aire, el horizonte artificial, la velocidad vertical y la altitud en el avión para medir la fenómeno del avión que sube/desciende.
- ÿ Comparación de valores de parámetros de proceso de registradores de datos independientes (vinculados a lazos de 4-20 mA o contactos de relé y transmitidos a través de canales de comunicación independientes) con datos del sistema SCADA (que llegan de forma "normal" a través de PLC y HMI) y alertas sobre desviaciones y desviaciones significativas. -valores especificados.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Facilita el monitoreo de valores manipulados (suponiendo que no todos los sensores se manipulen a la vez).
Fiabilidad	Evita la aceptación o identifica (para acciones futuras) mediciones corruptas o incorrectas como entradas.
Mantenimiento	Descarta las posibles causas físicas de los fallos con mayor rapidez.

Machine Translated by Google

Referencias

Táctica de **mapeo estándar / marco :**	
MITRE ATT&CK para ICS	TA010 - Deterioro del control de procesos **Técnica:** T0806 - E/S de fuerza bruta **SR 3.5:**
NIA 62443-3-3	Validación de entrada **SR 3.6:** Salida determinista **CR 3.5:** Validación de entrada **CR 3.6:** Salida
NIA 62443-4-2	determinista **CWE-754:** Comprobación incorrecta de condiciones inusuales o excepcionales
INGLETE CWE	

Machine Translated by Google

12. Valide las entradas en función de la plausibilidad física

Asegúrese de que los operadores solo puedan ingresar lo que es práctico o físicamente factible en el proceso.

Establezca un temporizador para una operación con la duración que debería tomar físicamente. Considere alertar cuando haya desviaciones.

También alerta cuando hay inactividad inesperada.

Seguridad Objetivo	Grupo objetivo
Integridad de los valores de E/S	Proveedor de servicios de integración/mantenimiento

Guia

a) Monitorear las duraciones físicas esperadas

Si la operación tarda más de lo previsto en pasar de un extremo al otro, es digno de alarma. Alternativamente, si lo hace demasiado rápido, eso también es digno de una alarma.

Una solución simple podría ser una alerta de tiempo de espera de paso. Esto sería útil para tareas controladas por secuencias/pasos.

Por ejemplo, el paso "mover objeto de A a B" tarda 5 segundos desde el inicio del paso hasta que se cumple la condición de transición (sensor: el objeto llegó a B).
Si la condición se cumple considerablemente demasiado pronto o demasiado tarde, se activa la alerta de tiempo de espera de paso.

b) Monitorear la actividad física repetitiva esperada

La verificación de la plausibilidad física también puede significar una alerta de inactividad físicamente improbable: si se espera un ciclo de eventos regular y repetitivo (por ejemplo, lotes, patrones diurnos), un temporizador de inactividad alertaría si algo que se espera que cambie (discreto o analógico). valor) permanece estático durante demasiado tiempo.

Ejemplo

a) Monitorear las duraciones físicas esperadas

- ÿ Las compuertas de una presa tardan cierto tiempo en pasar de completamente cerradas a completamente abiertas
- ÿ En una empresa de servicios públicos de aguas residuales, un pozo húmedo tarda cierto tiempo en llenarse

b) Monitorear la actividad física repetitiva esperada

- ÿ El procesamiento por lotes del proceso de fabricación o de la tubería debe realizar un ciclo regular entre los rangos de control o modos de funcionamiento.
- ÿ Las plantas de tratamiento de aguas residuales municipales suelen tener un ciclo diurno de actividad/patrón de caudales de entrada.

c) Limite la entrada del operador para puntos de referencia a lo que sea práctico/físicamente posible.

- ÿ por ejemplo, el caso de Oldsmar Florida permitió que el operador ingresara a) miles de veces más de lo que normalmente se necesitaba b) eso no es físicamente posible. Intente configurar los límites operativos en el código del PLC siempre que sea posible en lugar de usar scripts HMI.

Machine Translated by Google

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	**1.** Las desviaciones pueden indicar que un actuador ya estaba en medio de un estado de viaje o que alguien está tratando de falsificar la E/S, por ejemplo, realizando un ataque de repetición. **2.** Las alertas de inactividad facilitan el seguimiento de congelados o forzados valores constantes que podrían ser el resultado de la manipulación del sistema o del dispositivo.
Fiabilidad	**1.** Las desviaciones le dan una alerta temprana de equipos rotos debido fallas eléctricas o mecánicas. **2.** Las alertas de inactividad ayudan a marcar las mediciones o los bucles de control del sistema que pueden estar fallando (por lo tanto, estáticos) debido a una falla del dispositivo físico o un problema con el algoritmo de control lógico o una entrada fallida o incorrecta del operador.
Mantenimiento	

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control de procesos **Técnica:** T0806 - E/S de fuerza bruta **SR 3.5:**
NIA 62443-3-3	Validación de entrada **SR 3.6:** Salida determinista
NIA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
INGLETE CWE	**CWE-754:** Comprobación incorrecta de condiciones inusuales o excepcionales

Machine Translated by Google

13. Deshabilitar puertos y protocolos de comunicación innecesarios/no utilizados

Los controladores PLC y los módulos de interfaz de red generalmente admiten múltiples protocolos de comunicación que están habilitados de forma predeterminada. Deshabilite los puertos y protocolos que no son necesarios para la aplicación.

Endurecimiento de	Grupo objetivo
	Proveedor de servicios de integración/mantenimiento

Guia

Los protocolos comunes generalmente habilitados por defecto son, por ejemplo, HTTP, HTTPS, SNMP, Telnet, FTP, MODBUS, PROFIBUS, EtherNet/IP, ICMP, etc.

La mejor práctica es desarrollar un diagrama de flujo de datos que represente las comunicaciones requeridas entre el PLC y otros componentes del sistema.

El diagrama de flujo de datos debe mostrar tanto los puertos físicos del PLC como las redes lógicas a las que están conectados. Para cada puerto físico, se debe identificar una lista de los protocolos de red requeridos y se deben deshabilitar todos los demás.

Ejemplo

Por ejemplo, muchos PLC incluyen un servidor web incorporado para mantenimiento y resolución de problemas. Si no se utilizará esta función, si es posible, debe desactivarse ya que podría ser un vector de ataque.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Cada puerto y protocolo habilitados se suman a la superficie de ataque potencial del PLC. La forma más fácil de asegurarse de que un atacante no pueda usarlos para comunicaciones no autorizadas es deshabilitarlos por completo.
Fiabilidad	Si un PLC no puede comunicarse a través de un determinado puerto o protocolo, esto también reduce la cantidad potencial de tráfico (malformado), ya sea malicioso o no, lo que disminuye las posibilidades de que el PLC se bloquee debido a paquetes de comunicación no deseados o malformados.
Mantenimiento	La desactivación de puertos y protocolos no utilizados también facilita el mantenimiento, ya que reduce la complejidad general del PLC. Lo que no está no necesita ser administrado o actualizado.

Referencias

Mapeo estándar/marco	
MITRE ATT&CK para ICS	**Táctica:** TA005 - Descubrimiento **Técnica:** T0808 - Escaneo del servicio de identificación de dispositivos de control, T0841 - Red , T0854 - Enumeración de conexión serial
NIA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad **SR 7.7:** funcionalidad mínima
NIA 62443-4-2	**EDR 2.13 :** Uso de interfaces físicas de diagnóstico y prueba
NIA 62443-4-1	**SD-4:** mejores prácticas de diseño seguro **SI-1:** Revisión de la implementación de seguridad **SVV-1:** Pruebas de requisitos de seguridad

Machine Translated by Google

14. Restrinja las interfaces de datos de terceros

Restrinja el tipo de conexiones y datos disponibles para interfaces de terceros. Las conexiones y/o interfaces de datos deben estar bien definidas y restringidas para permitir solo capacidades de lectura/escritura para la transferencia de datos requerida.

Endurecimiento de	Grupo objetivo
	Proveedor de servicios de integración/mantenimiento

Guia

En algunos casos, debido a los largos tramos de cable o un gran intercambio de datos, las conexiones de datos interconectadas presentan un mejor caso comercial que el intercambio de datos por cable entre dos partes separadas.

Se deben considerar y seguir las siguientes pautas cuando sea práctico al diseñar e implementar una interfaz de intercambio de datos de terceros:

- ÿ Use un módulo de comunicaciones dedicado, ya sea conectado directamente al PLC de terceros o al equipo de intercambio de datos, o use un equipo de red dedicado separado físicamente de la red central de cada parte.
- ÿ La dirección MAC de los dispositivos conectados suele estar disponible en las variables del sistema para cualquier dispositivo ICS habilitado para Ethernet, lo que permite verificar la identidad del dispositivo con un enfoque multifactorial (dirección IP + código de fabricante MAC = dispositivo confiable). Esta práctica ciertamente no es infalible, ya que las direcciones MAC e IP pueden falsificarse, pero sirve para elevar el nivel en términos de comunicaciones entre sistemas y dispositivos ICS confiables.
- ÿ Al seleccionar un protocolo para interfaces de terceros, elija un protocolo que minimice la capacidad del tercero para escribir datos en el sistema del propietario.
- ÿ Elija un método de conexión y un puerto de conexión que impida que terceros sean capaz de configurar el PLC del propietario o el equipo de intercambio de datos.
- ÿ El tercero no debe poder leer ni escribir ningún dato que no haya sido explícitamente definido y disponible.
- ÿ Use un temporizador de vigilancia para monitorear la comunicación de modo que los comandos no se envíen a un PLC en modo de falla.
- ÿ Conexión en serie: use un módulo de comunicación dedicado para cada interfaz de terceros con una matriz restringida de datos. Asegúrese de que el lado del propietario de la conexión sea el iniciador y que el tercero sea el respondedor.
- ÿ Ethernet/IP: algunos PLC permiten que los módulos de comunicación funcionen como un firewall y pueden realizar una inspección profunda de paquetes (DPI) o restringir las interfaces del módulo de comunicación para limitar el intercambio de datos a un subconjunto predefinido. Si estas funciones están disponibles y se está utilizando un protocolo Ethernet/IP, asegúrese de que las funciones estén habilitadas y configuradas.
- ÿ Cuando los requisitos operativos o contractuales impidan al propietario cumplir con los elementos anteriores, considere usar un PLC de "concentrador de datos" (también conocido como proxy/DMZ) separado para almacenar los datos y proteger al propietario de escrituras/programaciones no deseadas de terceros. Asegúrese de que la placa posterior de este PLC no se pueda atravesar desde la red de terceros.

Machine Translated by Google

Ejemplo

ÿ Unidades de Transferencia Automática de Custodia (LACT) de Pipeline o Arrendamiento que transfieren y miden hidrocarburos o agua intercambiada entre una empresa productora o de oleoductos upstream y una empresa de oleoductos midstream con conexiones de red o interconectadas en serie que comparten información de medición, estado y permisiva entre empresas.

ÿ Proveedor regional de agua potable (importador) que comparte el caudal de agua de desvío que se entrega a la planta de agua de un municipio local.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	1. Limite la exposición a redes y equipos de terceros. 2. Autentique los dispositivos externos para evitar la suplantación de identidad.
Fiabilidad	Limita la capacidad de modificaciones intencionales o no intencionales o el acceso desde ubicaciones o equipos de terceros.
Mantenimiento	

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica:** TA010 - Deterioro del control de procesos **Técnica:** T0836 - Modificar parámetro
NIA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad **SR 7.7:** Funcionalidad mínima
NIA 62443-4-2	**CR 7.6:** Ajustes de configuración de red y seguridad **CR 7.7:** Funcionalidad mínima
NIA 62443-4-1	**SD-4:** mejores prácticas de diseño seguro **SI-1:** Revisión de la implementación de seguridad **SVV-1:** Pruebas de requisitos de seguridad

Machine Translated by Google

15. Defina un estado de proceso seguro en caso de un reinicio del PLC

Defina estados seguros para el proceso en caso de reinicios del PLC (p. ej., energizar contactos, desenergizar, mantener el estado anterior).

Objetivo de seguridad	Grupo objetivo
Resiliencia	Proveedor de productos Proveedor de servicios de integración/mantenimiento

Guia

Si algo le ordena a un PLC que se reinicie en medio de un proceso de trabajo, debemos esperar que el programa se recupere sin problemas con una interrupción mínima del proceso. Asegúrese de que el proceso que controla sea reiniciable.

Si no resulta práctico configurar el PLC para que se reinicie de forma segura, asegúrese de que le avise de este hecho y de que no emita ningún comando nuevo. Además, para ese caso, asegúrese de que los Procedimientos operativos estándar (SOP) tengan instrucciones muy claras para configurar los controles manuales para que el PLC inicie el proceso correctamente.

Además, documente todos los procedimientos de arranque, apagado, control de estado estable y reinicio del sistema de control de vuelo.

Ejemplo

/

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	**Elimina posibles comportamientos inesperados:** El vector de ataque más básico para un PLC es forzarlo a bloquearse y/o reiniciarse. Para muchos PLC, no es tan difícil de hacer, porque muchos PLC no pueden hacer frente a entradas inesperadas o demasiado tráfico. Si bien hay varios diagnósticos para las acciones del controlador mientras se está ejecutando, generalmente no está claro cómo maneja el inicio con un proceso en ejecución. Esto puede ser poco común, pero es un vector de ataque básico si tenemos en cuenta el comportamiento malicioso de un atacante.
Fiabilidad	**Evite retrasos inesperados:** Si después de un encendido del PLC, la máquina de estado se inicializa a un estado con algunas condiciones que no permiten que se inicie el proceso, y el operador no puede normalizar el sistema, un técnico deberá ingresar al programa del PLC para forzar que las condiciones continúen. al estado deseado para poder iniciar la operación. Esto podría causar retrasos y pérdidas de producción.
Mantenimiento	/

Machine Translated by Google

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica:** TA009 - Función de inhibición de respuesta **Técnica:** T0816 - Reinicio/apagado del dispositivo
NIA 62443-3-3	**SR 3.6:** Salida determinista
NIA 62443-4-2	**CR 3.6:** Salida determinista
NIA 62443-4-1	**SVV-1:** Pruebas de requisitos de seguridad

Machine Translated by Google

16. Resuma los tiempos de ciclo del PLC y trátelos en la HMI

Resuma el tiempo de ciclo del PLC cada 2-3 segundos e informe a la HMI para su visualización en un gráfico.

Monitoreo de objetivos	Grupo objetivo
	Proveedor de servicios de integración/mantenimiento

Guia

Los tiempos de ciclo suelen ser variables del sistema en un PLC y se pueden utilizar para resumir en código de PLC.
Se debe hacer un resumen para calcular los tiempos de ciclo promedio, pico y mínimo. La HMI debe mostrar la tendencia de estos valores y alertar si hay cambios significativos.

El tiempo de ciclo es el tiempo que se tarda en calcular cada iteración de lógica para el PLC. Las iteraciones son la combinación de diagramas de escalera (LD), diagramas de bloques de funciones (FBD), lista de instrucciones (IL) y texto estructurado (ST). Estos componentes lógicos se pueden unir con los gráficos de funciones secuenciales (SFC).

Los tiempos de ciclo deben ser constantes en un PLC a menos que haya cambios en, por ejemplo,

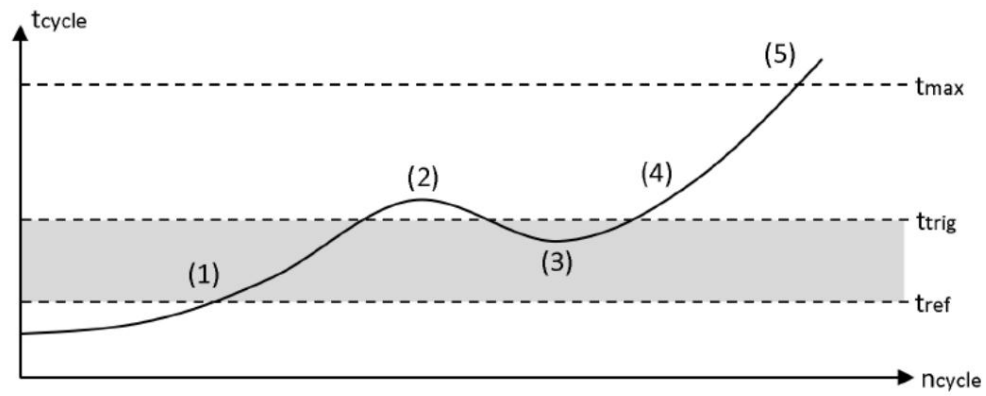
- ÿ entorno de red
- ÿ Lógica del PLC
- ÿ proceso

Por lo tanto, los cambios inusuales en el tiempo de ciclo pueden ser un indicador de que la lógica del PLC cambió y, por lo tanto, brindar información valiosa para las verificaciones de integridad.

La visualización de valores a lo largo del tiempo mediante un gráfico proporciona una forma intuitiva de llamar la atención sobre anomalías que serían más difíciles de notar si solo se tuvieran valores absolutos.

Ejemplo

Muchos PLC tienen un control de "tiempo de ciclo máximo" a nivel de hardware. Si el tiempo de ciclo supera el valor máximo, el hardware pone la CPU en STOP (5).
Por supuesto, los atacantes son conscientes de esto y mantendrán un posible código de ataque lo más reducido posible para minimizar el impacto en el tiempo de ciclo general. En un programa de monitoreo de tiempo de ciclo de software adicional, un tiempo de ciclo de referencia tref se define como tiempo de ciclo base. Como las pequeñas fluctuaciones son naturales, es necesario definir un umbral aceptable (1,3). La monitorización del ciclo se activa si se supera el umbral (2,4).

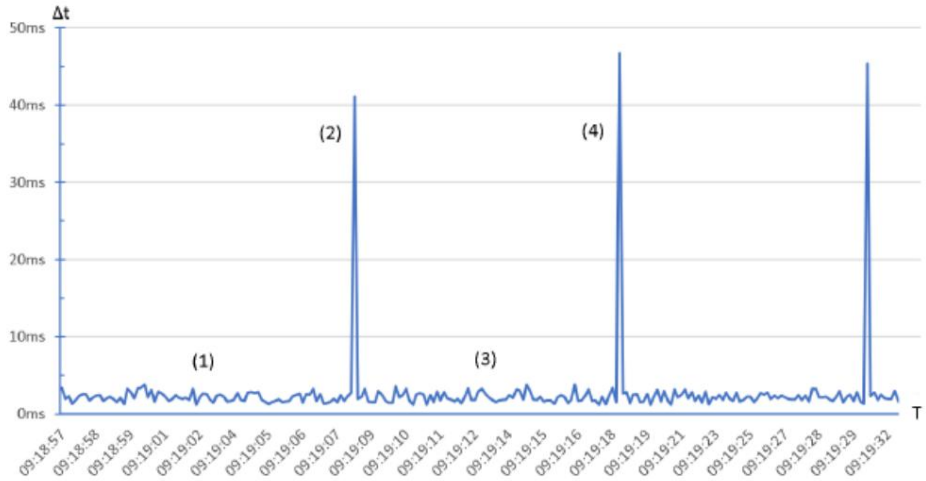


Machine Translated by Google

Cualquier desviación del tiempo de referencia se puede almacenar en un archivo de registro como este:

SeqNo	Date	UTC Time	Abweichung
1	2019-11-22	09:05:50.021	40,821ms
2	2019-11-22	09:06:00.069	44,391ms
3	2019-11-22	09:06:10.120	44,994ms
4	2019-11-22	09:06:20.166	40,561ms
5	2019-11-22	09:06:30.211	40,725ms

Si los tiempos de ciclo se muestran en la HMI, las cargas pesadas de la CPU son visibles de un vistazo. El siguiente diagrama de ejemplo muestra un programa de PLC con código malicioso ejecutado periódicamente. (1,3) muestran fluctuaciones de tiempo de ciclo aceptables ("ruido") durante el funcionamiento normal, el código de ataque se ejecuta en (2,4) que aumenta el tiempo de ciclo.



¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Los ataques a los PLC incluyen cambiar su lógica, activar un nuevo programa, probar un nuevo código, cargar una nueva receta de proceso, insertar lógica auxiliar para enviar mensajes o activar alguna característica. Para la mayoría de los PLC, las verificaciones de integridad criptográfica tradicionales no son factibles. Sin embargo, es bueno alertar si ocurre alguno de los cambios lógicos anteriores. Dado que los tiempos de ciclo son bastante constantes en circunstancias normales, los cambios en los tiempos de ciclo son un buen indicador de que la lógica en uno de los componentes lógicos anteriores ha cambiado.
Fiabilidad	Ver seguridad, pero por causas no maliciosas.
Mantenimiento	/

Machine Translated by Google

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica:** TA002 - Ejecución **Técnica:** T0873 - Infección del archivo del proyecto
NIA 62443-3-3	**SR 3.4:** Software e integridad de la información
NIA 62443-4-2	**EDR 3.2:** Protección contra código malicioso
INGLETE CWE	**CWE-754:** Comprobación incorrecta de condiciones inusuales o excepcionales

Machine Translated by Google

17. Registre el tiempo de actividad del PLC y trátelo en la HMI

Registre el tiempo de actividad del PLC para saber cuándo se ha reiniciado. Tendencia y registro del tiempo de actividad en la HMI para diagnósticos.

Monitoreo de objetivos	Grupo objetivo
	Proveedor de servicios de integración/mantenimiento

Guia

Realice un seguimiento del tiempo de actividad del PLC

- en el propio PLC (si el tiempo de actividad es una variable del sistema en el PLC)
- en el propio PLC si tiene MIB-2/cualquier implementación SNMP
- externamente mediante, por ejemplo, SNMP

Si el PLC tiene SNMP con MIB-2, que es muy común, el OID para uptime “sysUpTimeInstance(0)” es 1.3.6.1.2.1.1.3. Los restablecimientos de tiempo de actividad son indicadores importantes para los reinicios de PLC. Asegúrese de que la HMI avise de cualquier tipo de reinicio del PLC.

El tiempo de actividad correlacionado con los códigos de error es un buen diagnóstico.

Ejemplo

/

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	El vector de ataque más básico para un PLC es forzarlo a bloquearse y/o reiniciarse. Para muchos PLC, no es tan difícil de hacer, porque muchos PLC no pueden hacer frente a entradas inesperadas o demasiado tráfico. Por lo tanto, los reinicios inesperados pueden ser un indicador de que el PLC encuentra acciones inusuales.
Fiabilidad	Los reinicios de PLC también son buenos para el diagnóstico en caso de fallas y para monitorear en qué PLC se está trabajando en qué momento.
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica:** TA009 - Función de inhibición de respuesta **Técnica:** T0816 - Reinicio/apagado del dispositivo
NIA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad
NIA 62443-4-2	**CR 7.6:** Ajustes de configuración de red y seguridad
INGLETE CWE	**CWE-778:** Registro insuficiente

Machine Translated by Google

18. Registre las paradas duras del PLC y trátelas en la HMI

Almacene los eventos de parada dura del PLC de fallas o paradas para que los sistemas de alarma HMI los recuperen para consultarlos antes de que el PLC se reinicie. Sincronización de tiempo para datos más precisos.

Monitoreo de objetivos	Grupo objetivo
	Proveedor de servicios de integración/mantenimiento

Guia

Los eventos de falla indican por qué un PLC se apagó para que el problema se pueda abordar antes de reiniciar.

Algunos PLC pueden tener códigos de error del último caso en el que el PLC falló o se apagó incorrectamente. Registre esos errores y luego bórrelos. Podría ser una buena idea informar esos errores a la HMI como datos informativos o quizás a un servidor syslog, si existen esas características y esa infraestructura.

La mayoría de los PLC también tienen algún tipo de función de primer escaneo que genera eventos. Es un comportamiento que casi todos los equipos de PLC tienen de alguna forma. Es básicamente uno o más indicadores, o una rutina designada que se ejecuta en el primer escaneo de un PLC después de que se "despierta". Este primer escaneo debe registrarse y rastrearse.

Ejemplo

/

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Los registros permiten la solución de problemas en caso de un incidente. Antes de que un PLC entre en funcionamiento, especialmente después de haber experimentado problemas, es importante asegurarse de que sea confiable.
Fiabilidad	Los registros también son buenas fuentes para la depuración si el evento no fue causado de manera malintencionada.
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica:** TA009 - Función de inhibición de respuesta **Técnica:** T0816 - Reinicio/apagado del dispositivo 1
NIA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad
NIA 62443-4-2	**CR 7.6:** Ajustes de configuración de red y seguridad
INGLETE CWE	**CWE-778:** Registro insuficiente

Machine Translated by Google

19. Supervise el uso de la memoria del PLC y realice una tendencia en la HMI

Mida y proporcione una línea de base para el uso de la memoria para cada controlador implementado en el entorno de producción y trátelo en la HMI.

Objetivo de seguridad	Grupo objetivo
Supervisión	Proveedor de servicios de integración/mantenimiento Propietario del activo

Guia

Dado que el aumento de líneas de código en la lógica también puede conducir a un mayor consumo de memoria en tiempo de ejecución, se recomienda que los programadores de PLC realicen un seguimiento de cualquier desviación de la línea de base y dediquen una clase de alarma a este evento.

Ejemplo

En los PLC Rockwell Allen Bradley, se puede establecer una línea de base en un controlador y se puede realizar un seguimiento del uso de la memoria mediante la herramienta de supervisión de tareas RSLogix 5000. No solo la memoria principal, sino también la memoria de E/S y la memoria Ladder/Tag se pueden rastrear usando tendencias.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	El aumento del uso de la memoria puede ser un indicador de que el PLC ejecuta un código alterado.
Fiabilidad	El seguimiento del uso de la memoria para los programas en ejecución podría ser útil para evitar el consumo total de memoria y el eventual estado de falla del controlador PLC.
Mantenimiento	El seguimiento del uso de la memoria podría usarse para ajustar y encontrar el mejor tiempo de escaneo para el controlador monitoreado, pero también para solucionar problemas y problemas relacionados con estados defectuosos.

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica:** TA002 - Ejecución **Técnica:** T0873 - Infección del archivo del proyecto
NIA 62443-3-3	**SR 3.4:** Software e integridad de la información
NIA 62443-4-2	**EDR 3.2:** Protección contra código malicioso

Machine Translated by Google

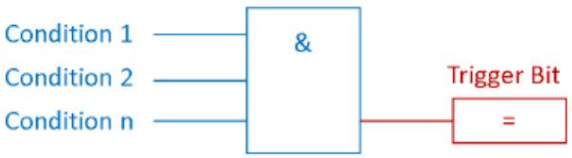
20. Atrapa falsos negativos y falsos positivos para alertas críticas

Identifique alertas críticas y programe una trampa para esas alertas. Configure la trampa para monitorear las condiciones de activación y el estado de alerta para cualquier desviación.

Monitoreo de objetivos	Grupo objetivo
	Proveedor de servicios de integración/mantenimiento

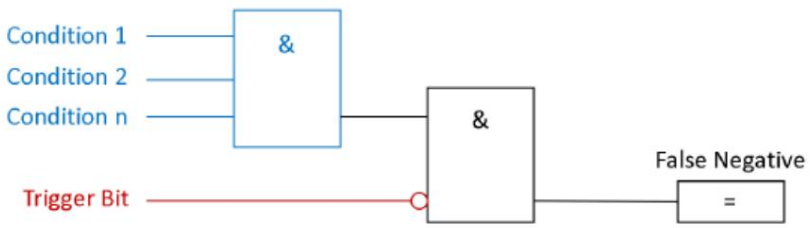
Guia

En la mayoría de los casos, los estados de alerta son booleanos (Verdadero, Falso) y se desencadenan por ciertas condiciones, como se muestra a continuación. Por ejemplo, el bit de activación para la alerta 'sobrepresión' se vuelve VERDADERO, si la Condición 1 'interruptor de presión 1', la Condición 2 'valor del sensor de presión sobre el umbral crítico', hasta n., son VERDADEROS.



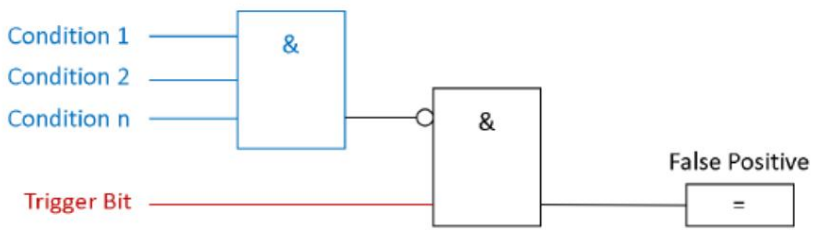
Para enmascarar un ataque, un adversario podría suprimir el bit de activación de la alerta y provocar un falso negativo.

Una trampa para falsos negativos monitorea las condiciones para el bit de activación y el propio bit de activación negado.
Con esta sencilla configuración, se detecta un falso negativo. Vea la siguiente imagen:



En otros casos, un adversario podría causar deliberadamente falsos positivos para desgastar la atención del operador del proceso.

De la misma manera que la trampa de falsos negativos, los falsos positivos también pueden detectarse monitoreando el bit de activación de alerta y si se cumplen las condiciones de activación. Si NO se cumplen las condiciones, pero el bit de disparo está activo, se detecta un falso positivo: Vea la siguiente imagen:



Machine Translated by Google

Ejemplo

Ejemplo 1: Siemens ofrece en sus productos Siemens S7-1200/1500 un servidor web con una amplia gama de funciones, por ejemplo, visualización del estado del PLC, tiempo de ciclo o registros de alcance. También tiene la opción de ver y modificar tablas de datos y variables. Los derechos de acceso al servidor web se pueden modificar en la configuración de hardware del PLC. En caso de derechos de acceso mal configurados, un adversario podría obtener acceso a las variables y bloques de datos del PLC. Para crear un falso positivo, el adversario selecciona un bit de activación de alerta y altera el estado.

Ejemplo 2: en el ataque Triton/Trisys/HatMan, el código no autorizado suprimió los estados de alerta.

Ejemplo 3: un ataque de inyección de bus podría enviar una alerta de falso positivo a un cliente SCADA de alto nivel.

¿Por qué?

Beneficioso para…?	¿Por qué?
Seguridad	Mitiga los falsos negativos o falsos positivos de los mensajes de alerta críticos causados por un adversario que ofusca su ataque (es decir, código malicioso, inyección de bus, manipulación de tablas de estado de PLC accesibles en servidores web no seguros).
Fiabilidad	/
Mantenimiento	/

Referencias

Mapeo estándar/marco	
MITRE ATT&CK ICS	**Táctica :** TA009 - Función de inhibición de respuesta **Técnica:** T0878 - Supresión de alarma **SR 3.5 :**
NIA 62443-3-3	Validación de entrada
NIA 62443-4-2	**CR 3.5 :** Validación de entrada
NIA 62443-4-1	**SI-1 :** Revisión de la implementación de seguridad
INGLETE CWE	**CWE-754:** Comprobación incorrecta de condiciones inusuales o excepcionales

Machine Translated by Google

Sobre el proyecto Programación segura de PLC

Durante muchos años, los controladores lógicos programables (PLC) han sido inseguros por diseño. Varios años de personalización y aplicación de las mejores prácticas de TI dieron lugar a protocolos seguros, comunicaciones cifradas, segmentación de redes, etc. Sin embargo, hasta la fecha, no se ha centrado en el uso de las funciones características de los PLC (o SCADA/DCS) para la seguridad, o cómo programar PLCs con la seguridad en mente. Este proyecto, inspirado en las prácticas de codificación segura existentes para TI, llena ese vacío.

¿Quién debe leer e implementar las Prácticas de codificación seguras de PLC?

Estas prácticas han sido escritas para ingenieros. El objetivo de este proyecto es proporcionar pautas a los ingenieros que están creando software (lógica de escalera, gráficos de funciones, etc.) para ayudar a mejorar la postura de seguridad de los sistemas de control industrial. Estas prácticas aprovechan la funcionalidad disponible de forma nativa en el PLC/DCS. Se necesita poca o ninguna herramienta de software o hardware adicional para implementar estas prácticas.
Todos pueden encajar en el flujo de trabajo operativo y de programación normal de PLC. Más que experiencia en seguridad, se necesita un buen conocimiento de los PLC a proteger, su lógica y el proceso subyacente para implementar estas prácticas.

¿Cuál es el alcance de esta lista? ¿Cómo define la codificación de PLC?

Para ajustarse al alcance de la lista de las 20 mejores prácticas seguras de codificación de PLC, las prácticas deben incluir cambios realizados directamente en un PLC. Lo que ve en este documento es una selección de los 20 principales de un mayor número de posibles prácticas seguras de codificación de PLC. También hay borradores de prácticas adicionales relacionadas con la arquitectura general, las HMI o la documentación. Esos no se ajustan al alcance de la codificación de PLC seguro, pero podrían estar en una lista futura en un entorno de PLC seguro.

¿Cuáles son los beneficios de aplicar prácticas seguras de codificación de PLC?

El uso de estas prácticas claramente tiene beneficios de seguridad, principalmente reduciendo la superficie de ataque o permitiendo una resolución de problemas más rápida si ocurriera un incidente de seguridad. Sin embargo, muchas prácticas tienen beneficios adicionales más allá de la seguridad. Algunos también hacen que el código del PLC sea más confiable, más fácil de depurar y mantener, más fácil de comunicar y posiblemente también más delgado. Además, las prácticas seguras de codificación de PLC no solo ayudan a los usuarios en caso de un atacante malicioso, sino que también hacen que el código de PLC sea más sólido para resistir una mala configuración accidental o un error humano.

¿Quién está detrás de este proyecto?

Todo comenzó con la charla S4x20 de Jake Brodsky "Prácticas de codificación segura para PLC".

Después de la conferencia, Dale Peterson inició el proyecto Top 20. Jake Brodsky y Sarah Fluchs pasaron varias horas al teléfono para plasmar en papel las prácticas de codificación segura de PLC propuestas por Jake.
Posteriormente, Dale, Jake y Sarah establecieron una plataforma en top20.isa.org, respaldada por ISA GCA, para estructurar y recopilar información adicional de las comunidades de seguridad e ingenieros de ICS.

Las discusiones y la consolidación de los textos de práctica, y la elaboración de una lista de las 20 prácticas principales más relevantes, tomó alrededor de un año; el proceso fue acelerado por Vivek Ponnada quien además de contribuir y revisar el contenido, también organizó llamadas periódicas hasta que se resolvieron todos los comentarios sobre las prácticas, Mohamed Abdelmoez Sakesli, quien agregó todas las referencias de los estándares en un gran esfuerzo, el equipo de MITRE CWE, que proporcionó las referencias de CWE en el último minuto, Sarah, que compiló el documento que usted están leyendo ahora, y Jake, Dale, John Cusimano, Dirk Rotermund, Josh Ruff, Thomas Rabenstein, Gus Serino, Walter Speth, Agustin Valencia Gil-Ortega, Marcel Rick-Cen y Al Ratheesh R, quienes brindaron información durante las llamadas regulares. .

Machine Translated by Google

lista de seguidores

El Proyecto de codificación segura de PLC es, y continúa siendo, un verdadero esfuerzo de la comunidad, que no habría sido posible sin innumerables colaboradores que compartieron generosamente su tiempo y conocimientos de seguridad/PLC. Un total de 943 Usuarios registrados en la plataforma para discutir y contribuir. Aquí hay una lista alfabética de todos los que aceptaron explícitamente ser nombrados. ¡Gracias a todos los que se tomaron el tiempo para apoyar este proyecto!

Aagam Shah

Adán Paturej

Agustín Valencia Gil-Ortega

Aitor García Almiñana

alec veranos

Al Ratheesh. R

andreas falk

Antón Shipulín

Arkaitz Gamino

Carlos Olave

Chris van den Hooven

chris sistrunk

Cristos Alexopoulos

Cris DeWitt

dale peterson

den yandle

dennis verschoor

dirk rotermund

Edorta Echave Garcia

Gananand Kini

Jorge Alex Holburn

Gus Serino

Josie Houghton

jozef sulwinski

Juan Pablo Ángel Espejo

Khalid Ansari

marc weber

Marcel Rick-Cen

Martín Huddleston

Massimiliano Zonta

mateo loong

Matthias Muller

miguel thompson

michal stepien

Miguel Ángel Frías

Mohamed Abdelmoez Sakesli

Luna Eluvangal Chandran

Nahuel Iglesias

Nalini Kanth

Narasimha S. Himakuntala

Omar Morando

Oscar J. Delgado-Melo

paivi brunou

Pedro Donnelly

Machine Translated by Google

Hakija Agic

Héctor Medrano

heiko rodolfo

isia jones

jacob brodsky

Javier Pérez Quezada

J. D. Bamford

Joe Weiss

Juan Cusimano

Juan Hoyt

Juan Powell

Juan Kingsley

José J. Januszewski

jose ruff

pedro jackson

Ravindra Deshakulakarni

Rick Booij

Roberto Albach

Rushi Purohit

Sara Fluchs

sergei biberdorf

Stephan Beirer

Steve Christey Coley

Tomas Rabenstein

Tim vendaval

Vivek Ponnada

Vitautas Butrimas

walter speth

Un agradecimiento especial a estas organizaciones que generosamente proporcionaron infraestructura para usar con el equipo del proyecto, como dominios, alojamiento y diseño web y diseño gráfico: