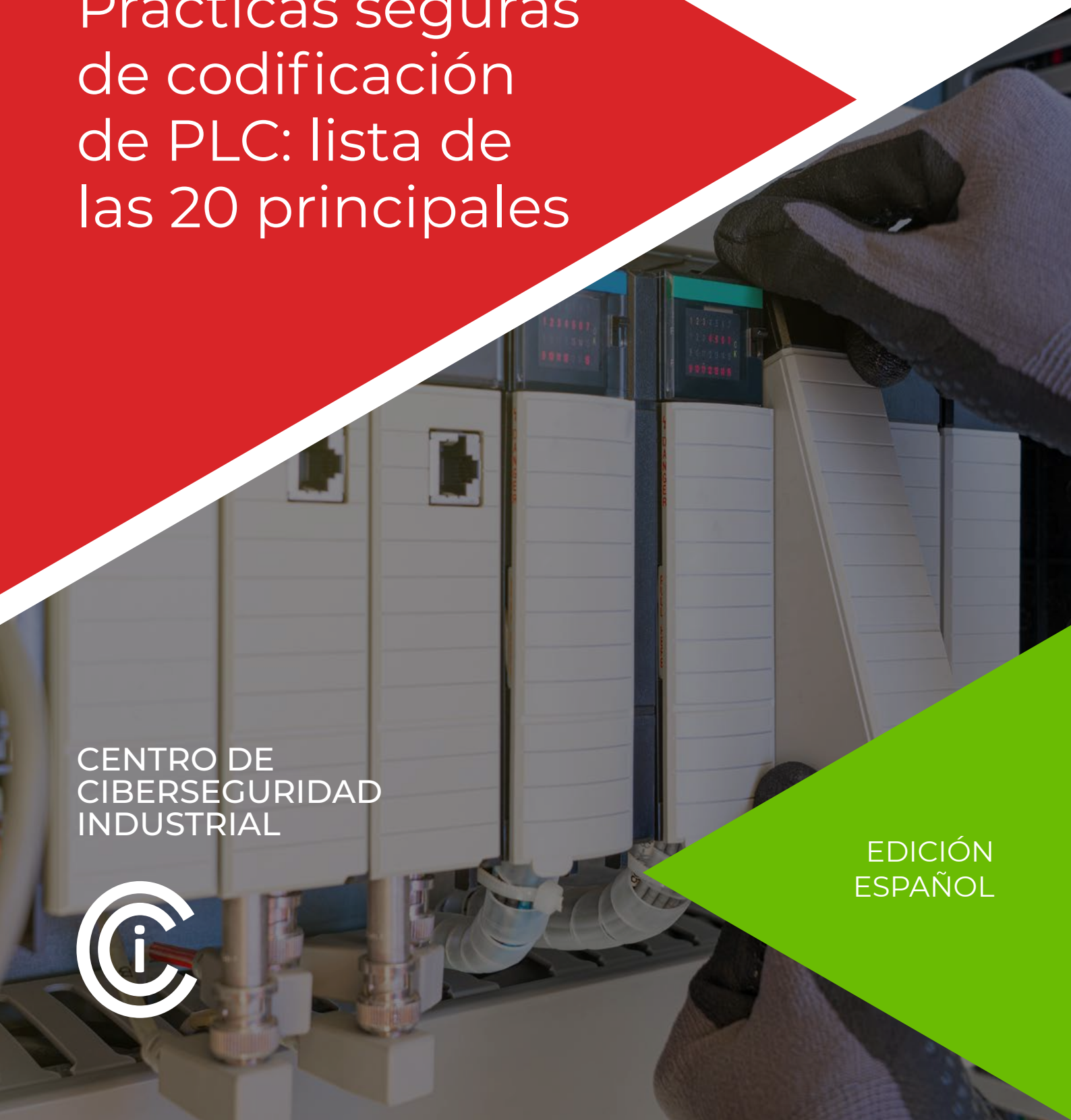
Top 20 Secure PLC Coding Practices

PLC Security
TOP 20 LIST

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



CENTRO DE CIBERSEGURIDAD INDUSTRIAL

EDICIÓN ESPAÑOL

CENTRO DE CIBERSEGURIDAD INDUSTRIAL

El **Centro de Ciberseguridad Industrial** (CCI) es una organización independiente, sin ánimo de lucro, cuya misión es impulsar y contribuir a la mejora de la Ciberseguridad Industrial, en un contexto en el que las organizaciones de sectores como el de fabricación o el energético juegan un papel crítico en la construcción de la sociedad actual, como puntales del estado del bienestar.

El CCI afronta ese reto mediante el desarrollo de actividades de investigación y análisis, generación de opinión, elaboración y publicación de estudios y herramientas, e intercambio de información y conocimiento, sobre la influencia, tanto de las tecnologías, incluidos sus procesos y prácticas, como de los individuos, en lo relativo a los riesgos -y su gestión- derivados de la integración de los procesos e infraestructuras industriales en el Ciberespacio.

CCI es, hoy, el ecosistema y el punto de encuentro de las entidades -privadas y públicas- y de los profesionales afectados, preocupados u ocupados de la Ciberseguridad Industrial; y es, asimismo, la referencia hispanohablante para el intercambio de experiencias y la dinamización de los sectores involucrados en este ámbito.

TOP 20 SECURE PLC CODING PRACTICES

PLC Security
TOP 20 LIST

Escrito para ingenieros por 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.

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.

Por todo ello, este proyecto ve necesario la colaboración con otras entidades y profesionales que puedan hacer llegar estas prácticas a todos ellos en diferentes idiomas y en este caso, junto con el **Centro de Ciberseguridad Industrial**, quienes se han sumado para aportar este documento en español con el fin de llegar a toda la comunidad hispanohablante.

índice

CONSEJOS:
Al hacer clic sobre el pie o número de página regresas al índice
Alt+flecha izquierda para volver a la vista anterior después de ir a un hipervínculo.

1.

Modularizar el código del PLC

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

Objetivo de seguridad	Grupo objetivo
Integridad de la lógica del PLC	Proveedor del código

ORIENTACIÓN

No programe toda la lógica del PLC en un solo sitio, 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, la gestión del control de acceso, la 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 ha sido meticulosamente probado, cualquier modificación de estos módulos puede ser verificada con el hash del código original, por ejemplo, guardando un hash de cada uno de estos módulos (cuando es una opción en el PLC). This way, modules can be validated during the FAT/SAT or if the integrity of the code is in question after an incident.

EJEMPLO

La lógica de la turbina de gas está separada en “puesta en marcha”, “control de los álabes de entrada”, “control de la válvula de purga”, etc., para que pueda aplicar la lógica estándar de forma sistemática. Esto también ayuda a solucionar rápidamente los problemas en caso de que se produzca un incidente de seguridad.

Los bloques de funciones personalizados que se prueban rigurosamente pueden reutilizarse sin alteraciones (y recibir una alerta si se intentan hacer cambios) y bloquearse contra el abuso/mal uso con una contraseña/firma digital.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	Facilita la detección de porciones de código recién añadidas que podrían ser maliciosas. Ayuda a la estandarización de la lógica, la coherencia y el bloqueo contra las modificaciones no autorizadas.
Fiabilidad	Ayuda a controlar la secuencia de flujo del programa y a evitar los bucles, que podrían hacer que la lógica no reaccionara correctamente o se bloqueara.
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 pueden utilizarse para otros PLC, lo que permite utilizar un código común e identificarlo en distintos PLC. Esto puede ayudar al personal de mantenimiento a reconocer rápidamente los módulos comunes durante la resolución de problemas.

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA002 -Técnica de ejecución **Técnica:** T0844 - Unidades de organización del programa
ISA 62443-3-3	**SR 3.4:** Integridad del software y de la información
ISA 62443-4-2	**CR 3.4:** Integridad del software y de la información
ISA 62443-4-1	**SI-2:** Normas de codificación segura
MITRE CWE	**CWE-1120:** Complejidad excesiva del código **CWE-653:** Compartimentación insuficiente

2.

Seguir los modos operativos

Mantener 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 de activos

ORIENTACIÓN

Si los PLC no están en modo RUN (por ejemplo, en modo PROGRAM), su código podría cambiarse para seguir el modo RUN. Algunos PLC tienen una suma de comprobación para alertar de los cambios de código, pero si no la tienen, hay al menos un indicador indirecto de un posible problema mientras se rastrean los modos operativos:

- Si los PLC no están en modo RUN, debe haber una alarma para los operadores. Si saben que alguien debe estar trabajando en ese sistema de control, pueden reaccionar ante la alarma y seguir adelante.
- La HMI debe estar configurada 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 todo el personal o los contratistas de la planta que realicen trabajos que puedan afectar al proceso.

Caso de excepción: Si la central está en fase de pruebas o desarrollo, considere la posibilidad de desactivar esta alarma, pero la central debe estar aislada de los niveles superiores de la red.

EJEMPLO

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

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	El modo operativo (ejecutar / editar / escribir; para los PLC de Allen Bradley: RUN / PROGram / REMote) 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

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA009 - Inhibir la función de respuesta **Técnica:** T0858 - Utilizar/cambiar el modo operativo
ISA/IEC 62443-4-1	**SI-1:** Revisión de la aplicación de la seguridad

3.

Dejar la lógica operativa en el PLC siempre que sea posible

Dejar la mayor parte de la lógica operativa, por ejemplo, la totalización o la integración, directamente en el PLC. La HMI no recibe suficientes actualizaciones para hacerlo bien.

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

ORIENTACIÓN

Las HMI ofrecen 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 más bien en el PLC para seguir siendo 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 hacer la totalización / integración bien. Además, siempre hay latencia entre la HMI y el PLC. Además, cuando el código está en el PLC, y una HMI se reinicia, siempre puede recibir los totalizadores/cuentas de un PLC.

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

Para analizar los valores de los datos del proceso a lo largo del tiempo, un historiador de datos del proceso es la mejor opción que la HMI. Utilizar consultas en una base de datos del historiador de procesos para comparar los valores totalizados (durante un período, durante un lote, durante un ciclo de proceso) con los totales agregados localmente en la lógica del PLC. La alerta sobre una varianza mayor puede explicarse por las diferencias en la granularidad de los datos.

EJEMPLO

- Código para establecer las condiciones de habilitación/deshabilitación de los controles: las acciones de habilitación/deshabilitación deben ser controladas en la capa del PLC, de lo contrario, se pueden realizar acciones en la HMI (o a través de la red) en el PLC, aunque no se cumplan las condiciones (previstas).

- Los temporizadores para permitir acciones al operador (temporizador de retardo para inicios consecutivos del motor, temporizador para considerar las válvulas cerradas/abiertas o el motor parado) no deben ponerse en la capa de la HMI sino en el PLC que gestiona dicho motor/válvula.

- Los umbrales de las alarmas tienen que formar parte de los códigos del PLC aunque se muestren en las HMI.

- Depósito de agua con volumen cambiante: El PLC que controla el flujo de entrada y salida del tanque puede totalizar fácilmente el volumen (y validar de forma cruzada los totales). La HMI también podría hacerlo, pero tendría que obtener primero los valores del PLC. Estos valores necesitarían marcas de tiempo precisas para obtener totales correctos en caso de latencia o y podrían perder valores si la HMI se reinicia.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	1. Permite la coherencia en la verificación de los cambios de código. La codificación de la 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), no permitiendo a los propietarios del sistema tener una visión completa e incluso perdiendo consideraciones importantes. Las HMI no incluyen “señales forzadas” o listas de valores cambiados como los PLC o SCADA, por lo que los cambios a nivel de la HMI son más difíciles de detectar, siendo prácticamente imposible que formen parte de un plan de gestión de cambios de autorización. H 2. Para un atacante, es más difícil manipular los totales distribuidos en muchos PLC que manipular los totales calculados en la HMI. 3. Si una parte de las funciones de habilitación/deshabilitación no están en el PLC, los atacantes podrían ser capaces de manipular el PLC y las E/S sin tener que trabajar la parte de la HMI, ya que la información adecuada ya está ofuscada en la pantalla del operador.

¿Beneficioso para...?	¿Por qué?
Fiabilidad	1. Los cálculos son más eficientes y precisos si están más cerca del campo. Además, los totales y los recuentos seguirán estando disponibles si la HMI se reinicia (los PLC no se reinician tan a menudo y suelen almacenar estos valores en la memoria no volátil). 2. Las diferentes fuentes de entradas y enclavamientos pueden significar fallos no esperados. En una planta puede haber diferentes tecnologías para las HMI (capa SCADA, pero también paneles de control de campo) y los cambios en una de ellas no se difundirán por el resto de capas, lo que provocará incoherencias en la visualización y posibles fallos en la operación.
Mantenimiento	La codificación es fácil de entender y transferir de PLC a PLC, no tanto de HMI a HMI.

REFERENCIAS

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

4.

Utilizar indicadores de PLC como comprobaciones de integridad

Poner contadores en los indicadores de error del PLC para capturar cualquier problema matemático.

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

ORIENTACIÓN

Si el código del PLC funcionaba bien pero de repente hace una división por cero, hay que investigarlo. Si algo se está comunicando peer to peer desde otro PLC y la función/lógica hace una división por cero cuando no se esperaba, hay que investigarlo.

La mayoría de los programadores ignorarán el problema como si fuera un error matemático o, peor aún, podrían suponer que su código es perfecto y dejar que el PLC entre en un estado de fallo duro. Durante el desarrollo del código, los ingenieros necesitan probar y validar sus módulos de código (fragmentos o rutinas) introduciendo datos fuera de los límites previstos. Esto puede denominarse pruebas unitarias.

Asigne segmentos de memoria diferentes y bloqueados para el firmware, la lógica y la pila de protocolos. Probar la pila de protocolos para los casos de abuso. Los casos de abuso podrían ser condiciones peculiares de indicadores en la cabecera de un paquete.

EJEMPLO

Los fallos del PLC causados por datos fuera de los límites son muy comunes. Esto ocurre, por ejemplo, cuando un valor de entrada hace que los índices de la matriz se salgan de los límites, o los temporizadores con preajustes negativos, o las excepciones de división por cero.

Los indicadores típicos de interés son:

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

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	Los ataques a los PLC pueden incluir la modificación de su lógica, la activación de un nuevo programa, la prueba de un nuevo código, la carga de una nueva fórmula de proceso, la inserción de una lógica auxiliar para enviar mensajes o la activación de alguna función. Dado que la mayoría de los PLC no proporcionan comprobaciones de integridad criptográfica, los indicadores pueden ser una buena señal si se produce uno de los cambios lógicos anteriores.
Fiabilidad	Los indicadores tomados en serio pueden evitar que el PLC funcione con errores de programación o de E/S. Además, si se produce un error, la fuente de la falla es más obvia.
Mantenimiento	/

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control del proceso **Técnica:** T0836 - Modificar parámetro
ISA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
ISA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
ISA 62443-4-1	**SI-2:** Normas de codificación segura **SVV-1:** Prueba de requisitos de seguridad
MITRE CWE	**CWE-128:** Envolvente **CWE-190:** Desbordamiento de enteros **CWE-369:** Dividir por cero **CWE-754:** Verificación incorrecta de condiciones inusuales o excepcionales

5.

Realizar comprobaciones de integridad criptográficas y/o de suma de comprobación para el código PLC

Utilizar hashes criptográficos, o sumas de comprobación si los hashes criptográficos no están disponibles, para comprobar la integridad del código del PLC y emitir una alarma cuando cambien.

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

ORIENTACIÓN

A) Sumas de comprobación

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

Si la función de suma de comprobación no está disponible de forma nativa en el controlador, ésta también puede generarse en el EWS/HMI y comprobarse, por ejemplo, una vez al día para compararla con el hash del código original en el PLC para verificar que coinciden. Aunque esto no proporcionará alertas en tiempo real, es lo suficientemente bueno para rastrear si alguien está tratando de hacer cambios en el código del PLC.

El valor de la suma de comprobación también puede trasladarse a un registro del PLC y configurarse para una alarma cuando cambie, el valor puede enviarse a los historiadores, etc.

B) Hashes

Las CPUs del PLC no suelen tener capacidad de procesamiento para generar o comprobar los hashes en funcionamiento. De hecho, intentar un hash puede hacer que el PLC se bloquee. Pero el software de ingeniería del PLC podría ser capaz de calcular los hashes a partir del código del PLC y guardarlos en el PLC o en algún otro lugar del sistema de control.

EJEMPLO

Proveedores de PLC que se sabe que tienen funciones de suma de comprobación:

- Siemens (ver ejemplo)
- Rockwell

Además, se puede utilizar software externo para generar sumas de comprobación:

- Version dog
- Asset Guardian
- PAS

Ejemplo de implementación de Siemens

Ejemplo de creación de sumas de comprobación en un PLC Siemens S7-1500:

El bloque GetChecksum-Function lee la suma de comprobación real y con un script ligero se puede almacenar la "SAT-Checksum" como referencia. Una desviación de la suma de control de referencia puede almacenarse con la función de registro de datos.

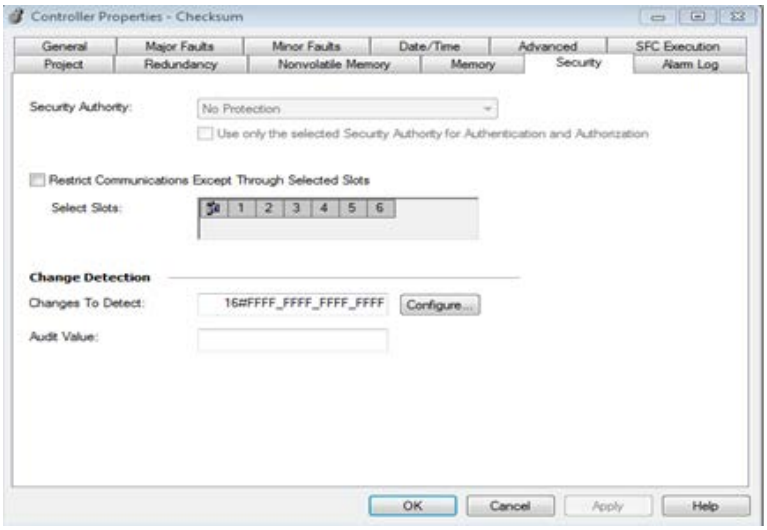
	Date	UTC Time	Referenz	Aktuell
1	11/21/2019	9:55:11	84 2A 76 DF 5B 31 F4 16	FF 2C EA 71 44 D7 81 04
2	11/21/2019	9:57:33	FF 2C EA 71 44 D7 81 04	FF 2C EA 71 44 D7 81 04
3	11/21/2019	9:58:17	FF 2C EA 71 44 D7 81 04	5B 7C 57 7E E2 3E EF C3
4	11/21/2019	9:58:36	FF 2C EA 71 44 D7 81 04	5B 7C 57 7E E2 3E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

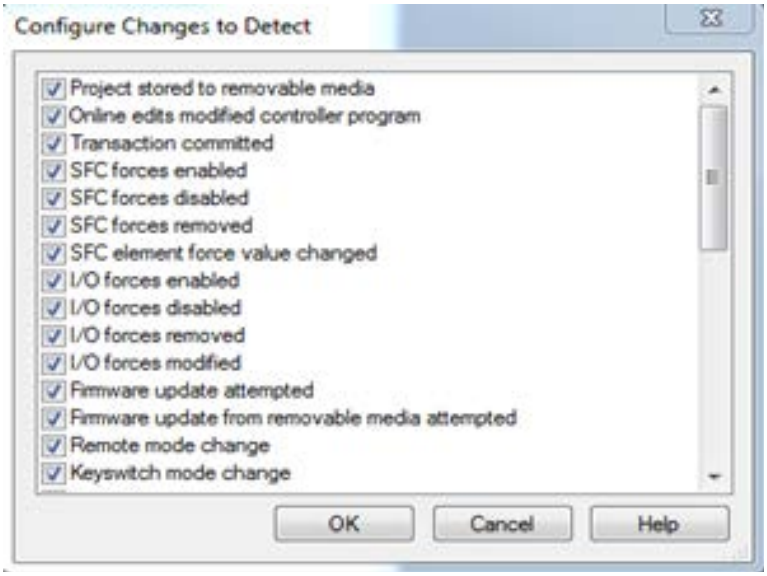
Este es un ejemplo parcial de cómo una organización puede desarrollar un nivel de capacidad de detección de cambios en el programa PLC dentro de su entorno ICS. Este ejemplo es específicamente para un PLC ControlLogix de Rockwell Automation y no está completo; sin embargo, ilustra cómo recuperar el estado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

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

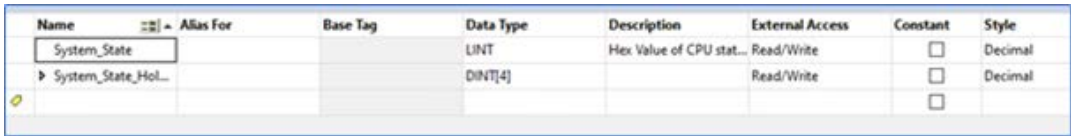
1. En el cuadro de diálogo Propiedades del controlador, seleccione el botón de configuración en “Cambiar para detectar”



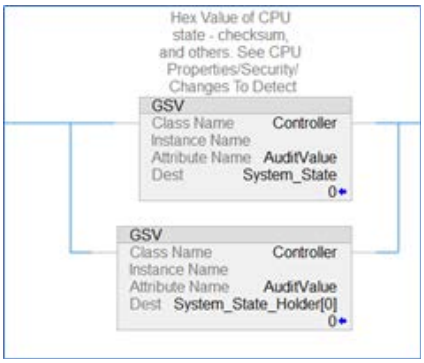
2. Dentro de la ventana de selección, elija todos los elementos a controlar



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”



4. Utilice la instrucción Get System Values (GSV) para obtener la información del estado del procesador de la memoria y trasladarla a una etiqueta que pueda utilizarse en la lógica o leerse en la HMI



¿POR QUÉ?

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

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA002 - Ejecución, TA010 - Deterioro del Control del Proceso **Técnica:** T0873 - Infección de archivos de proyecto, T0833 - Modificar la lógica de control
ISA 62443-3-3	**SR 3.4 :** Integridad del software y de la información
ISA 62443-4-2	**CR 3.4 :** Integridad del software y de la información **EDR 3.12 :** Aprovisionamiento de las raíces de confianza de los proveedores de productos
ISA 62443-4-1	**SI-1 :** Revisión de la aplicación de la seguridad **SVV-1** Prueba de requisitos de seguridad
MITRE CWE	**CWE-345:** verificación insuficiente de la autenticidad de los datos • **(hijo) CWE-353:** Falta el soporte para la verificación de integridad • **(hijo) CWE-354:** Validación incorrecta del valor de verificación de integridad

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 recuentos hacia atrás por debajo de cero.

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

ORIENTACIÓN

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

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

- no dejar que la HMI escriba en el temporizador o en el contador directamente, sino pasar por una lógica de validación
- validar las preselecciones y los valores de tiempo de espera en el PLC

La validación de las entradas del temporizador y del contador es fácil de hacer directamente en el PLC (sin necesidad de ningún dispositivo de red capaz de realizar una inspección profunda de paquetes), ya que el PLC “sabe” cuál es el estado o contexto del proceso. Puede validar “qué” recibe y “cuándo” recibe las órdenes o consignas.

EJEMPLO

Durante la puesta en marcha del PLC, los temporizadores y contadores suelen estar preestablecidos a determinados valores.

Si hay un temporizador que activa las alarmas a los 1,3 segundos, pero ese temporizador está preconfigurado maliciosamente a 5 minutos, podría no activar la alarma.

Si hay un contador que hace que un proceso se detenga cuando llega a 10.000, pero que está puesto a 11.000 desde el principio, el proceso podría no detenerse.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	Si las E/S, los temporizadores o las preconfiguraciones se escriben directamente en las E/S, sin ser validadas 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 preselecciona accidentalmente valores erróneos del temporizador o del contador.
Mantenimiento	Tener rangos válidos para los temporizadores y contadores documentados y validados automáticamente puede ayudar al actualizar la lógica.

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control del proceso **Técnica:** T0836 - Modificar parámetro
ISA 62443-3-3	**SR 3.5:** Validación de entrada
ISA 62443-4-2	**CR 3.5:** Validación de entrada
ISA 62443-4-1	**SI-2:** Normas de codificación segura **SVV-1:** prueba de requisitos de seguridad

7.

Validar y alertar sobre entradas / salidas emparejadas

Si tiene señales emparejadas, asegúrese de que ambas señales no se afirmen juntas. Alarma al operador cuando ocurren estados de entrada / salida que no son físicamente factibles. Considere la posibilidad de independizar las señales emparejadas o de añadir temporizadores de retardo cuando la conmutación de las salidas pueda ser perjudicial para los actuadores.

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

ORIENTACIÓN

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

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 conmutan rápidamente de manera que se dañan los actuadores.

EJEMPLO

Ejemplos de señales emparejadas:

- INICIO y PARADA
  - Inicio y parada independientes: Configure el inicio y la parada como salidas discretas en lugar de tener una única salida que pueda activarse y desactivarse. Por diseño, esto no permite disparos simultáneos. Para un atacante, es mucho más complicado activar/desactivar rápidamente si hay que configurar dos salidas diferentes.
  - Temporizador para el reinicio: Considera también la posibilidad de añadir un temporizador para el reinicio después de una parada, para evitar la desconexión rápida de las señales de inicio/parada.
- AVANCE y RETROCESO
- ABRIR y CERRAR

Ejemplos para alternar señales emparejadas que podrían ser perjudiciales:

Si el PLC / MCC acepta una entrada discreta, esto proporciona una opción fácil para un atacante para causar daños físicos en los actuadores. El escenario más conocido para alternar las salidas para hacer daño sería un MCC, pero esta práctica se aplica a todos los escenarios donde alternar las salidas podría hacer daño. Una prueba de concepto en la que la alternancia rápida de las salidas podía causar daños reales fue la prueba del generador Aurora, realizada en 2007 por el Laboratorio Nacional de Idaho, en la que alternar las salidas fuera de sincronización causó daños en los disyuntores.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	1. Si los programas del PLC no tienen en cuenta lo que va a suceder si ambas señales de entrada emparejadas se afirman al mismo tiempo, este es un buen vector de ataque. 2. Las dos señales de entrada emparejadas que se afirman son una advertencia de que hay un error de funcionamiento, un error de programación o que está ocurriendo algo malicioso. 3. Esto evita un escenario de ataque en el que se pueden causar daños físicos 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 producirse inadvertidamente.

Mantenimiento	/

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control del proceso **Técnica:** T0836 - Modificar parámetro, T0806 - E/S de fuerza bruta
ISA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
ISA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6 :** Salida determinista
ISA 62443-4-1	**SI-2:** Normas de codificación segura **SVV-1:** Prueba de requisitos de seguridad
MITRE CWE	**CWE-754:** Verificación incorrecta de condiciones inusuales o excepcionales

8.

Validar las variables de entrada de la HMI en el nivel del PLC, no sólo en la 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 deben añadirse otras comprobaciones cruzadas en el PLC para evitar, o alertar sobre, valores fuera de los rangos aceptables que están programados en la HMI.

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

ORIENTACIÓN

La validación de la entrada podría incluir comprobaciones de valores operativos válidos, así como de valores válidos en términos de tipos de datos relativos al proceso.

Si una variable del PLC recibe un valor que está fuera de los límites, proporcionar la lógica del PLC para

- introducir un **valor por defecto** a esa variable que no afecte negativamente al proceso, y que pueda ser utilizado como un indicador de alerta, o
- introducir el **último valor correcto** a ese valor y registrar el evento para su posterior análisis.

EJEMPLO

Ejemplo 1

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

```
IF V1 < 0 OR IF V1 > 100, SET V1 = 0.
```

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

Ejemplo 2

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

1. La entrada de la HMI a la variable V2 tiene un rango restringido de -32768 a 32767 (dec.) programado en la HMI.

1. El PLC dispone de una lógica de comprobación cruzada del tipo de datos que supervisa la variable de desbordamiento (V3), que existe justo después de V2 en la estructura de memoria del PLC:

```
IF V2 = -32768 OR IF V2 = 32767 AND V3 != 0,

SET V2 = 0 AND SET V3 = 0 AND SET DataTypeOverflowAlarm = TRUE.
```

Ejemplo 3

Escalar PV (Valor de Proceso), SP (Punto de Ajuste) y CV (Variable de Control) para PID (Controlador Proporcional, Integral, Derivativo) a unidades consistentes o crudas para eliminar los errores de escala que causan problemas de control. El escalado incorrecto puede dar lugar a casos de abuso involuntarios.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	1. Mientras que las HMI suelen proporcionar algún tipo de validación de entrada, un operador malicioso puede crear o reproducir paquetes modificados para enviar valores arbitrarios a las variables del PLC que están abiertas a la influencia externa (abiertas a los valores pasados desde una HMI, por ejemplo). 2. Los protocolos PLC se suelen comercializar como protocolos "abiertos" y se publican para el público en general, por lo que la creación de malware que utilice información de protocolos "abiertos" puede ser trivial de desarrollar. El mapeo de variables PLC puede ocurrir típicamente a través del análisis de tráfico durante las fases de reconocimiento de un ataque, proporcionando así al intruso la información necesaria para elaborar tráfico malicioso hacia el objetivo y así manipular un proceso con herramientas no autorizadas. La comprobación 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 estableciendo forzosamente rangos seguros cuando se detecta un valor fuera de los límites durante el curso de la exploración del PLC.
Fiabilidad	/
Mantenimiento	/

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control del proceso **Técnica:** T0836 - Modificar parámetro
ISA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
ISA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6 :** Salida determinista
ISA 62443-4-1	**SI-2:** Normas de codificación segura **SVV-1:** Prueba de requisitos de seguridad
MITRE CWE	**CWE-1320:** Protección inadecuada para las alertas de nivel de señal fuera de los límites

9.

Validar indirecciones

Valide las indirecciones envenenando los extremos de la matriz para detectar errores en los postes de la cerca.

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

ORIENTACIÓN

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

Los ejemplos de indirecciones necesarias son:

- Variadores de frecuencia variable (VFD) que activan diferentes acciones para diferentes frecuencias utilizando tablas de búsqueda.
- Para decidir qué bomba se pone en marcha primero en función de sus tiempos de funcionamiento actuales.

Los PLC no suelen tener un indicador de "fin de matriz" por lo que es una buena idea crearla en el software; el objetivo es evitar operaciones inusuales/no planificadas del PLC.

EJEMPLO

PROGRAMACIÓN POR LISTA DE INSTRUCCIONES (IL)

El enfoque puede convertirse en unos pocos bloques de función y posiblemente incluso reutilizarse para otras aplicaciones.

1. Crear una máscara de matriz

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

```
[21  31  41  51  61]
```

definir una matriz de 8:

```
[x  x  21  31  41  51  61  x]
```

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

Advertencia: ¡el índice comienza en 0!

```
[21  31  41  51  61]
___________^
```

Índice: 3

añadir una compensación para compensar el final envenenado. El desplazamiento puede ser 1 o mayor, en este caso es 2:

```
[x  x  21  31  41  51  61  x]
_______________^
```

Índice incluyendo desplazamiento: `3 + 2 = 5`

y luego AND el índice incluyendo 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:

```
6 AND 0x07 devolvería

  6,7 AND 0x07 devolvería 78

  AND 0x07 devolvería 0,9

  AND 0x07 devolvería 1.
```

Esto garantiza que siempre se dirija a un valor de la matriz.

2. Insertar extremos envenenados

El envenenamiento de los extremos es opcional. Podrías detectar indirecciones manipuladas sin el envenenamiento, pero el envenenamiento ayuda a detectar los errores de los postes de la valla porque obtienes de vuelta un valor que no tiene sentido.

La cuestión es que en el índice 0 de la matriz debería haber un valor no válido, como -1 o 65535. Esto 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. Registro de valor de dirección indirecta sin máscara

A continuación, registre el valor de la dirección indirecta sin la máscara AND y el desplazamiento:

en este ejemplo, registraría 51 para el índice 3.

[21 31 41 51 61]

_____^

_____Index 3

4. Ejecutar la máscara AND y comparar los valores (=validación de la indirección)

Compare su valor registrado con el valor después de haber hecho el desplazamiento y la máscara AND.

4a. Caso A: Indirección correcta

En primer lugar, desplazamiento:

Índice + Desplazamiento = 3 + 2 = 5

En segundo lugar, máscara:

5 AND 0x07 = 5

En tercer lugar, verificación de indirección

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

_____^

Índice que incluye el desplazamiento: 5

Valor = 51 es igual al valor registrado, por lo que todo está bien.

4b. Caso B: Indirección manipulada

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

En primer lugar, compensación:

índice + compensación = 7 + 2 = 9

En segundo lugar, máscara:

```
9 AND 0x07 = 1
```

En tercer lugar, verificación de indirección:

```
[ -1 -1 21 31 41 51 61 -1]
____ ^
```

Índice que incluye el desplazamiento: **1**

`Valor = -1` no es igual al valor registrado y también indica su extremo envenenado, por lo que sabría que su indirección está manipulada.

5. Ejecutar fallo/alerta de programador

Si este valor validado es diferente al registrado, entonces sabe que algo está mal. Genere una alarma de calidad del software.

Luego, verifique el valor de indirección. Si se trata de un valor envenenado, debería generar otra alarma de calidad del software. Esto es una indicación de un error en el poste de la valla.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	La mayoría de los PLC no tienen ninguna función para manejar los índices fuera de los límites de las matrices. Hay **dos escenarios potencialmente peligrosos** que pueden derivarse de los errores de indirección: **En primer lugar**, si una indirección conduce a la lectura del registro equivocado, el programa se ejecuta utilizando valores erróneos. **En segundo lugar**, si una indirección errónea lleva a escribir en el registro incorrecto, el programa sobrescribe el código o los valores que desea conservar. En ambos casos, los errores de indirección pueden ser difíciles de detectar y pueden tener graves consecuencias. Pueden ser causados por un error humano, pero también pueden ser insertados maliciosamente.
Fiabilidad	Identifica los errores humanos no maliciosos en la programación.
Mantenimiento	/

REFERENCIAS

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

10.

Asignar bloques de registro designados por función (lectura / escritura / validación)

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

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

ORIENTACIÓN

La memoria temporal, también conocida como memoria “scratch pad”, es un área de memoria fácilmente explotable si no se sigue esta práctica. Por ejemplo, la simple escritura en un registro “Modbus” que está fuera de los límites podría conducir a la sobreescritura de los registros de memoria utilizados para los cálculos temporales.

Generalmente, la memoria de registro puede ser accedida por otros dispositivos a través de la red del PLC para operaciones de lectura y escritura. Algunos registros podrían ser leídos por una HMI, y otros podrían ser escritos por un sistema SCADA, etc. Tener matrices de registro específicas para una determinada aplicación también facilita (en el controlador o se utiliza un firewall externo) configurar el acceso de sólo lectura desde otro dispositivo/otra HMI.

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

- lectura
- escritura (desde HMI / controlador / otro dispositivo externo)
- validando escrituras
- cálculos

Garantizar las escrituras externas en los registros permitidos también ayuda a evitar errores de reinicio de la memoria principal, ya sea debido a una ejecución fuera de los límites o a intentos maliciosos. Estos bloques de registro designados pueden utilizarse como búferes para las escrituras de E/S, temporizadores y contadores, validando que el búfer está completamente escrito (no contiene parte de datos antiguos y parte de datos nuevos) y validando todos los datos en el búfer.

Contexto:

La memoria principal y la memoria de registro se utilizan de forma diferente. La memoria principal se utiliza para almacenar la lógica del programa que se está ejecutando, mientras que la memoria de registro se utiliza como memoria temporal por la lógica que se está ejecutando. Aunque la memoria de registro es temporal, ya que está siendo utilizada por la lógica de ejecución está destinada a contener algunas variables importantes que afectarían a la lógica principal.

EJEMPLO

Ejemplos de lo que podría ocurrir si no se aplica esta práctica:

El enfoque puede convertirse en unos pocos bloques de función y posiblemente incluso reutilizarse para otras aplicaciones.

(Referencia: G. P. H. Sandaruwan, P. S. Ranaweera, Vladimir A. Oleshchuk, PLC Security and Critical Infrastructure Protection):

- Siemens suele utilizar la memoria scratchpad en el área de indicadores desde el indicador 200.0 hasta el indicador 255.7. Si se cambia un bit dentro de esta área, existe la probabilidad de que se produzca un mal funcionamiento grave del PLC en función de la importancia de ese bit o byte.

- Supongamos que un atacante puede acceder a una de las máquinas de la red del PLC e infectar esa máquina con un gusano capaz de escribir valores arbitrarios en la memoria de registro. Como los valores de la memoria del registro cambian arbitrariamente, puede cambiar el valor de la presión.

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

Ejemplos de aplicación de esta práctica:

- En un escenario en el que 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 SOLO LECTURA en la zona de seguridad.

- En otro escenario, podría haber algunos registros con capacidad de escritura, y otros son de sólo 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). Facilita el trabajo de los firewalls sensibles al protocolo: Las reglas se simplifican porque está muy claro a qué bloques de registro puede acceder la HMI. Facilita la gestión de las reglas (más sencillas) del firewall. Realizar cambios no autorizados en la memoria temporal interna es una vulnerabilidad fácilmente explotable (By-pass Logic Attack). Cuando las entradas y salidas de las rutinas del PLC se validan adecuadamente, cualquier cambio (por un actor malicioso o por error) puede ser detectado fácilmente en lugar de permanecer en la secuencia lógica durante mucho tiempo y arrojar errores / causar problemas más adelante en la ejecución.
Fiabilidad	Hace que las lecturas y escrituras sean más rápidas porque se reduce el número de transacciones. 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 de comunicación en los mensajes largos pueden dar lugar a errores involuntarios si no se comprueba la validez de los datos antes de procesarlos.
Mantenimiento	Los errores de programación que provocan la escritura en la memoria temporal pueden dificultar la búsqueda de errores, por lo que el problema puede evitarse asignando registros específicos para las escrituras.

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA009 - Función de respuesta de inhibición, TA010 - Control de proceso de deterioro **Técnica:** T0835 - Manipular imagen de E / S , T0836 - Modificar parámetro
ISA 62443-3-3	**SR 3.4:** Integridad del software y de la información **SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
ISA 62443-4-1	**SD-4:** Mejores prácticas de diseño seguro **SI-1:** Revisión de la implementación de seguridad **SI-2:** Normas de codificación seguras **SVV-1:** prueba de requisitos de seguridad
ISA 62443-4-2	**CR 3.4:** Integridad del software y de la información **CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
MITRE CWE	**CWE-787:** Escritura fuera de límites **CWE-653:** Compartimentación insuficiente

11.

Instrumentar el control de plausibilidad

Instrumentar el proceso de forma que permita comprobar la verosimilitud mediante la comprobación cruzada de diferentes mediciones.

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

ORIENTACIÓN

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

a) Comparar las mediciones integradas e independientes del tiempo

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

b) Comparar diferentes fuentes de medición

Asimismo, medir el mismo fenómeno de diferentes maneras puede ser una buena comprobación de plausibilidad.

Las diferentes fuentes de medición no tienen por qué ser necesariamente sensores físicos diferentes, sino que también pueden significar el uso de canales de comunicación alternativos (ver ejemplos).

EJEMPLO

a) Comparar medidas integradas e independientes del tiempo

- Bomba dosificada y medidor de nivel del depósito: el cambio volumétrico debe ser igual al caudal integrado.
- Quemador en una caldera: el calor calórico añadido debe ser igual al aumento de temperatura.

b) Comparar diferentes fuentes de medición

- Utilizar la velocidad del aire, el horizonte artificial, la velocidad vertical y la altitud en el avión para medir el fenómeno del ascenso/descenso del avión.
- Comparación de los valores de los parámetros del proceso procedentes de registradores de datos independientes (vinculados a bucles de 4-20mA o contactos de relé y transmitidos a través de canales de comunicación independientes) con los datos del sistema SCADA (que llegan de forma “normal” a través del PLC y la HMI) y alerta sobre las desviaciones y los valores significativamente fuera de especificación.

¿POR QUÉ?

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

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Tactic:** TA010 - Deterioro del control del proceso **Técnica:** T0806 - Fuerza bruta E/S
ISA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6:** Salida determinista
ISA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
MITRE CWE	**CWE-754:** Verificación incorrecta de condiciones inusuales o excepcionales

12.

Validar entradas basadas en plausibilidad física

Asegúrese de que los operadores sólo pueden introducir lo que es práctico o físicamente factible en el proceso. Establezca un temporizador para una operación con la duración que debe tener físicamente. Considere alertar cuando haya desviaciones. Avise también cuando hay una inactividad inesperada.

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

ORIENTACIÓN

a) Controlar las duraciones físicas previstas

Si la operación tarda más de lo previsto en ir de un extremo a otro, eso merece una alarma. Por otra parte, si lo hace demasiado rápido, eso también merece una alarma.

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

Por ejemplo, el paso "mover el 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 demasiado pronto o demasiado tarde, el tiempo de espera del paso se activa.

b) Controlar la actividad física repetitiva esperada

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

EJEMPLO

a) Controlar las duraciones físicas esperadas

- Las compuertas de una presa tardan un tiempo determinado en pasar de completamente cerradas a completamente abiertas
- En un servicio de aguas residuales, una fosa húmeda tarda cierto tiempo en llenarse

b) Controlar la actividad de repetición física prevista

- El proceso de fabricación o la dosificación de las tuberías deben alternar regularmente entre los rangos de control o los modos operativos.
- Las plantas municipales de tratamiento de aguas residuales suelen tener un ciclo diurno de actividad / patrón de caudales afluentes.

c) Limitar la entrada del operador para los puntos de ajuste a lo que es práctico/físicamente posible.

- Por ejemplo, el caso de Oldsmar, Florida, permitió la entrada de un operador que es a) miles de veces más de lo que normalmente se necesita b) que es físicamente imposible. Intente configurar los límites operativos en el código del PLC siempre que sea posible en lugar de utilizar scripts de la HMI.

¿POR QUÉ?

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

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA010 - Deterioro del control del proceso **Técnica:** T0806 - Fuerza bruta E/S
ISA 62443-3-3	**SR 3.5:** Validación de entrada **SR 3.6 :** Salida determinista
ISA 62443-4-2	**CR 3.5:** Validación de entrada **CR 3.6:** Salida determinista
MITRE CWE	**CWE-754:** Verificación incorrecta de condiciones inusuales o excepcionales

13.

Desactivar los puertos y protocolos de comunicación innecesarios/no utilizados

Los controladores del PLC y los módulos de interfaz de red soportan generalmente varios protocolos de comunicación que están activados por defecto. Desactive los puertos y protocolos que no sean necesarios para la aplicación.

Objetivo de seguridad	Grupo objetivo
Endurecimiento	Proveedor de servicios de integración / mantenimiento

ORIENTACIÓN

Los protocolos comunes que suelen estar 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 describa las comunicaciones necesarias 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 protocolos de red necesarios y desactivar todos los demás.

EJEMPLO

Por ejemplo, muchos PLC incluyen un servidor web integrado para el mantenimiento y la resolución de problemas. Si no se va a utilizar esta función, si es posible, debería desactivarse, ya que podría ser un vector de ataque.

¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	Cada puerto y protocolo habilitado se suma a la superficie de ataque potencial del PLC. La forma más fácil de asegurarse de que un atacante no pueda utilizarlas para una comunicación no autorizada es desactivarlos 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 / malformados.
Mantenimiento	Desactivar los puertos y protocolos no utilizados también facilita el mantenimiento, ya que reduce la complejidad general del PLC. Lo que no está ahí no necesita ser administrado o actualizado.

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK para ICS	**Táctica:** TA005 - Descubrimiento **Técnica:** T0808 - Identificación del dispositivo de control, **T0841 -** Escaneo de servicios de red, **T0854 -** Enumeración de conexiones en serie
ISA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad **SR 7.7:** Funcionalidad mínima
ISA 62443-4-2	**EDR 2.13:** Uso de interfaces de prueba y diagnóstico físico
ISA 62443-4-1	**SD-4:** Mejores prácticas de diseño seguro **SI-1:** Revisión de la aplicación de seguridad **SVV-1:** Prueba de requisitos de seguridad

14.

Restringir las interfaces de datos de terceros

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

Objetivo de seguridad	Grupo objetivo
Endurecimiento	Proveedor de servicios de integración / mantenimiento

ORIENTACIÓN

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

Los siguientes principios deben tenerse en cuenta y seguirse, siempre que sea posible, a la hora de diseñar e implementar una interfaz de intercambio de datos de terceros:

- Utilizar un módulo de comunicaciones dedicado, bien conectado directamente al PLC de la tercera parte o al equipo de intercambio de datos, o bien utilizar un equipo de red dedicado y físicamente segregado de la red principal de cada parte.
- La dirección MAC de los dispositivos conectados suele estar disponible en las variables del sistema para cualquier dispositivo habilitado para Ethernet de ICS, lo que permite verificar la identidad del dispositivo con un enfoque multifactorial (dirección IP + código de fabricante MAC = dispositivo de confianza). Esta práctica no es infalible, ya que las direcciones MAC e IP pueden ser falsificadas, pero sirve para elevar el nivel de las comunicaciones entre los sistemas y dispositivos ICS de confianza.
- Al seleccionar un protocolo para las interfaces de terceros, elija un protocolo que minimice la capacidad del tercero de escribir datos en el sistema del propietario.
- Elija un método de conexión y un puerto de conexión que impida que el tercero pueda configurar el PLC o el equipo de intercambio de datos del propietario.
- El tercero no debe poder leer o escribir en ningún dato que no haya sido definido explícitamente y puesto a disposición.
- Utilice un temporizador de vigilancia para supervisar la comunicación de modo que no se envíen órdenes a un PLC en modo de fallo.
- Conexión en serie: Utilice un módulo de comunicación dedicado para cada interfaz de terceros con una matriz de datos restringida. Asegúrese de que el lado del propietario de la conexión es el iniciador y que el tercero es el respondedor.

- 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 de los módulos de comunicación para limitar el intercambio de datos a un subconjunto predefinido. Si estas funciones están disponibles y se utiliza un protocolo Ethernet/IP, asegúrese de que las funciones están activadas y configuradas.

- Cuando los requisitos operativos o contractuales impidan al propietario llevar a cabo los puntos anteriores, considere la posibilidad de utilizar un PLC "concentrador de datos" independiente (también conocido como proxy/DMZ) para almacenar los datos y proteger al propietario de escrituras/programaciones no deseadas de terceros. Asegúrese de que el backplane de este PLC no puede ser atravesado desde la red de terceros.

EJEMPLO

- Unidades de Transferencia Automática de Custodia (LACT) que transfieren y miden los hidrocarburos o el agua que se intercambia entre una empresa productora o de oleoductos y gasoductos y una empresa de oleoductos y gasoductos con conexiones en red o en serie que comparten información de medición, estado y permisos entre las empresas.

- 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. Limitar la exposición a redes y equipos de terceros. 2. Autenticar los dispositivos externos para evitar la suplantación de identidad.
Fiabilidad	Limita la posibilidad de realizar modificaciones o accesos, intencionados o no, desde ubicaciones o equipos de terceros.
Mantenimiento	

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica:** TA010 - Deterioro del control del proceso **Técnica:** T0836 - Modificar el parámetro
ISA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad **SR 7.7:** Funcionalidad mínima
ISA 62443-4-2	**SR 7.6:** Ajustes de configuración de red y seguridad **SR 7.7:** Funcionalidad mínima
ISA 62443-4-1	**SD-4:** Mejores prácticas de diseño seguro **SI-1:** Revisión de la aplicación de seguridad **SVV-1:** Prueba de requisitos de seguridad

15.

Definir un estado de proceso seguro en caso de reinicio del PLC

Definir estados seguros para el proceso en caso de reinicio del PLC (por ejemplo, energizar los contactos, desenergizar, mantener el estado anterior).

Objetivo de seguridad	Grupo objetivo
Resiliencia	Proveedor del código Proveedor de servicios de integración / mantenimiento

ORIENTACIÓN

Si algo ordena a un PLC que se reinicie en medio de un proceso de trabajo, debemos esperar que el programa se reanude sin problemas con una interrupción mínima del proceso. Asegúrese de que el proceso que controla es seguro para el reinicio.

Si no resulta práctico configurar el PLC para que se reinicie de forma segura, asegúrese de que le avisa de este hecho y de que no emite ningún comando nuevo. Además, para ese caso, asegúrese de que los Procedimientos Operativos Estándar (POE) tengan instrucciones muy claras para configurar los controles manuales de manera que el PLC ponga en marcha el proceso correctamente.

Además, documente todos los procedimientos de puesta en marcha, 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 reiniciarlo. En el caso de muchos PLC, no es tan difícil de hacer, porque muchos PLC no pueden hacer frente a entradas inesperadas o a demasiado tráfico. Mientras que hay varios diagnósticos para las acciones del controlador mientras se está ejecutando, la forma en que maneja el inicio con un proceso en ejecución no suele ser clara. 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 encender un PLC, la máquina de estado se inicializa a un estado con algunas condiciones que no permiten que el proceso se inicie, y el operador no puede normalizar el sistema, un técnico tendría que entrar en el programa del PLC para forzar las condiciones para ir al estado deseado para poder iniciar la operación. Esto podría provocar retrasos y pérdidas de producción.
Mantenimiento	/

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica:** TA009 - Inhibir la función de respuesta **Técnica:** T0816 - Reinicio/apagado del dispositivo
ISA 62443-3-3	**SR 3.6:** Salida determinista
ISA 62443-4-2	**CR 3.6:** Salida determinista
ISA 62443-4-1	**SVV-1:** Prueba de requisitos de seguridad

16.

Resumir los tiempos de ciclo del PLC y la tendencia en la HMI

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

Objetivo de seguridad	Grupo objetivo
Supervisión	Proveedor de servicios de integración / mantenimiento

ORIENTACIÓN

Los tiempos de ciclo suelen ser variables del sistema en un PLC y pueden utilizarse para resumir en el código del PLC. Se debe hacer un resumen para calcular los tiempos de ciclo medio, máximo y mínimo. La IHM debe establecer una 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 la lógica para el PLC. Las iteraciones son la combinación de diagramas de escalera (LD), diagramas de bloques de función (FBD), lista de instrucciones (IL) y texto estructurado (ST). Estos componentes lógicos pueden unirse con los cuadros de funciones secuenciales (SFC).

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

- el entorno de red
- la lógica del PLC
- el proceso

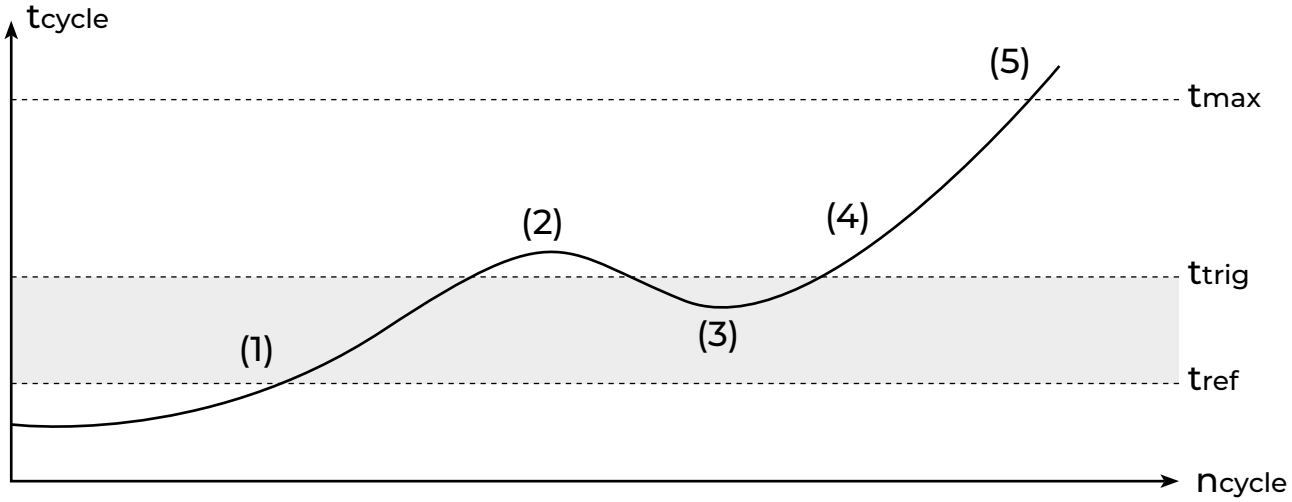
Por lo tanto, los cambios inusuales en el tiempo de ciclo pueden ser un indicador de que la lógica del PLC cambió y, por lo tanto, proporcionar información valiosa para las comprobaciones de integridad.

La visualización de los valores a lo largo del tiempo mediante un gráfico proporciona una forma intuitiva de llamar la atención sobre las anomalías que serían más difíciles de percibir si sólo se dispusiera de valores absolutos.

EJEMPLO

Muchos PLC tienen un control del “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).

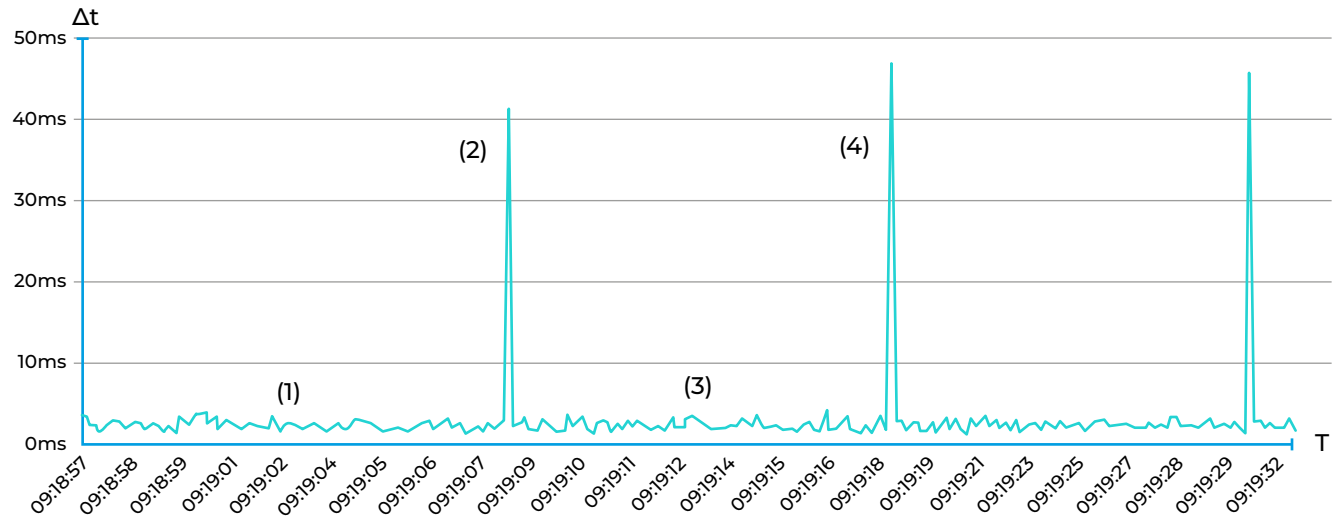
Por supuesto, los atacantes son conscientes de ello y mantendrán un posible código de ataque lo más reducido posible para minimizar el impacto en el tiempo de ciclo general. En un programa adicional de control del tiempo de ciclo, se define un tiempo de ciclo de referencia tref como tiempo de ciclo base. Dado que las pequeñas fluctuaciones son naturales, es necesario definir un umbral aceptable (1,3) La supervisión del ciclo se activa si se supera el umbral (2,4).



Cualquier desviación de la hora de referencia puede almacenarse en un archivo de registro como éste:

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



¿POR QUÉ?

¿Beneficioso para...?	¿Por qué?
Seguridad	Los ataques a los PLC incluyen el cambio de su lógica, la activación de un nuevo programa, la prueba de un nuevo código, la carga de una nueva fórmula de proceso, la inserción de una lógica auxiliar para enviar mensajes o activar alguna función. Para la mayoría de los PLC, las comprobaciones tradicionales de integridad criptográfica 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	Consulte seguridad, pero por causas no malintencionadas.
Mantenimiento	/

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica:** TA002 - Ejecución **Técnica:** T0873 - Infección del archivo del proyecto
ISA 62443-3-3	**SR 3.4:** Integridad del software y de la información
ISA 62443-4-2	**EDR 3.2:** Protección contra códigos maliciosos
MITRE CWE	**CWE-754:** Verificación incorrecta de condiciones inusuales o excepcionales

17.

Registrar el tiempo de actividad del PLC y su tendencia en la HMI

Registre el tiempo de actividad del PLC para saber cuándo se reinició. Tendencia y registro del tiempo de actividad en la HMI para el diagnóstico.

Objetivo de seguridad	Grupo objetivo
Supervisión	Proveedor de servicios de integración / mantenimiento

ORIENTACIÓN

Mantener un registro 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 / alguna implementación SNMP
- de forma externa mediante, por ejemplo, SNMP

Si el PLC tiene SNMP con MIB-2, lo cual es muy común, el OID para el tiempo de funcionamiento “sysUpTimeInstance(0)” es 1.3.6.1.2.1.3. Los reinicios del tiempo de funcionamiento son indicadores importantes para los reinicios del PLC. Asegúrese de que la HMI avisa de cualquier tipo de reinicio del PLC.

El tiempo de actividad correlacionado con los códigos de error son buenos diagnósticos.

EJEMPLO

/

¿POR QUÉ?

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

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica:** TA009 - Inhibir la función de respuesta **Técnica:** T0816 - Reinicio/apagado del dispositivo
ISA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad
ISA 62443-4-2	**SR 7.6:** Ajustes de configuración de red y seguridad
MITRE CWE	**CWE-778:** Registro insuficiente

18.

Registrar las paradas duras del PLC y realice la tendencia de ellas en la HMI

Almacena los eventos de parada dura del PLC por fallos o apagados para que los sistemas de alarma de la HMI los consulten antes de reiniciar el PLC. Sincronización horaria para obtener datos más precisos.

Objetivo de seguridad	Grupo objetivo
Supervisión	Proveedor de servicios de integración / mantenimiento

ORIENTACIÓN

Los eventos de fallo indican la razón por la que un PLC se ha apagado, de modo que el problema puede solucionarse antes de un reinicio.

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 elimínelos. Podría ser una buena idea reportar esos errores a la HMI como datos informativos o tal vez a un servidor syslog, si esas características y esa infraestructura existen.

La mayoría de los PLC también tienen algún tipo de función de primera exploración que genera eventos. Es un comportamiento que casi todos los equipos 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 resolución de problemas en caso de un incidente. Antes de que un PLC entre en funcionamiento, sobre todo después de haber tenido problemas, es importante asegurarse de que es digno de confianza.
Fiabilidad	Los registros también son una buena fuente de depuración si el evento no fue causado de manera maliciosa.
Mantenimiento	/

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica:** TA009 - Inhibir la función de respuesta **Técnica:** T0816 - Reinicio/apagado del dispositivo 1
ISA 62443-3-3	**SR 7.6:** Ajustes de configuración de red y seguridad
ISA 62443-4-2	**SR 7.6:** Ajustes de configuración de red y seguridad
MITRE CWE	**CWE-778:** Registro insuficiente

19.

Supervisar el uso de la memoria del PLC y su tendencia en la HMI

Medir y proporcionar una línea de base para el uso de la memoria para cada controlador desplegado en el entorno de producción y la tendencia en la HMI.

Objetivo de seguridad	Grupo objetivo
Supervisión	Proveedor de servicios de integración / mantenimiento Propietario de los activos

ORIENTACIÓN

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

EJEMPLO

En los PLC de Rockwell Allen Bradley, se puede establecer una línea base en un controlador y se puede hacer un seguimiento del uso de la memoria utilizando la herramienta de monitorización de tareas RS-Logix 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 memoria puede ser un indicador de que el PLC ejecuta código alterado.
Fiabilidad	El seguimiento del uso de la memoria de los programas en ejecución podría ser útil para evitar el consumo total de memoria y el eventual estado de fallo del controlador del PLC.
Mantenimiento	El seguimiento del uso de la memoria podría utilizarse para ajustar y encontrar el mejor tiempo de escaneo para el controlador supervisado, pero también para solucionar problemas y cuestiones relacionadas con estados defectuosos.

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica:** TA002 - Ejecución **Técnica:** T0873 - Infección del archivo del proyecto
ISA 62443-3-3	**SR 3.4:** Integridad del software y de la información
ISA 62443-4-2	**EDR 3.2:** Protección contra códigos maliciosos

20.

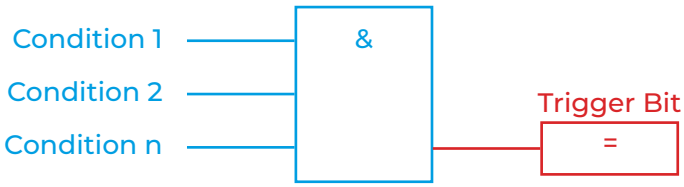
Programar trampa de falsos negativos y falsos positivos para alertas críticas

Identificar las alertas críticas y programar una trampa para esas alertas. Configure la trampa para supervisar las condiciones de activación y el estado de alerta para cualquier desviación.

Objetivo de seguridad	Grupo objetivo
Supervisión	Proveedor de servicios de integración / mantenimiento

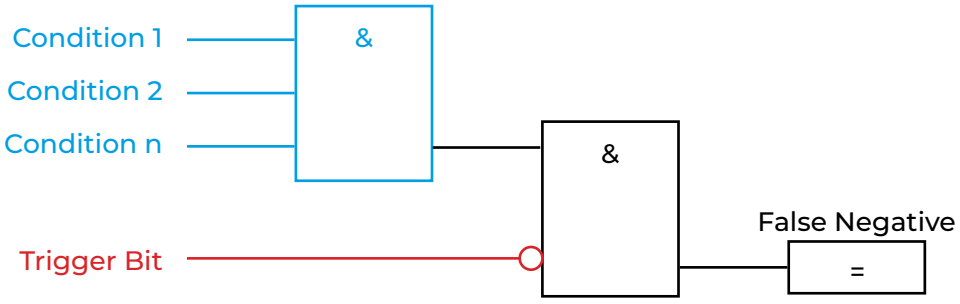
ORIENTACIÓN

En la mayoría de los casos, los estados de alerta son booleanos (Verdadero, Falso) y se activan por determinadas condiciones, como se muestra a continuación. Por ejemplo, el bit de activación de la alerta "sobrepresión" se convierte en TRUE, si la condición 1 "presostato 1", la condición 2 "valor del sensor de presión por encima del umbral crítico", hasta n., son TRUE.



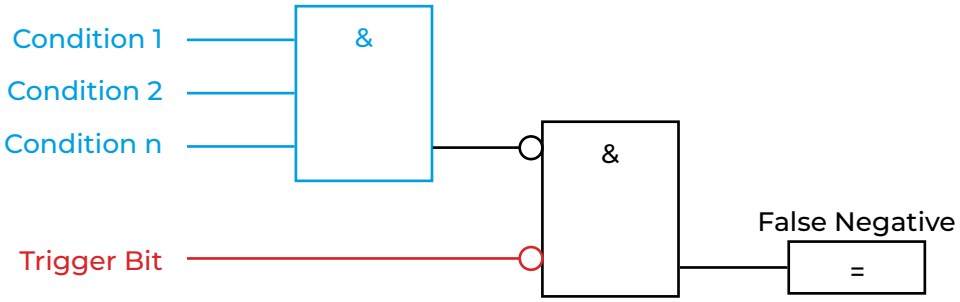
Para enmascarar un ataque, un adversario podría suprimir el bit de activación de alerta y provocar un falso negativo.

Una trampa para falsos negativos controla las condiciones del bit de disparo y el propio bit de disparo negado. Con esta sencilla configuración, se detecta un falso negativo. Véase 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 ser detectados monitoreando el bit de activación de la alerta y si se cumplen las condiciones de activación. Si las condiciones NO se cumplen, pero el bit de disparo está activo, se detecta un falso positivo: Véase la siguiente imagen:



EJEMPLO

Ejemplo 1: Siemens ofrece en sus productos Siemens S7-1200/1500 un servidor web con una amplia gama de funciones, por ejemplo, la visualización del estado del PLC, el tiempo de ciclo o los registros de cobertura. También tiene la opción de ver y modificar las tablas de datos y las variables. Los derechos de acceso al servidor web pueden modificarse en la configuración del hardware del PLC. En caso de que los derechos de acceso estén mal configurados, un adversario podría acceder 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 de Triton/Trisys/HatMan, el código malicioso suprimió los estados de alerta.

Ejemplo 3: Un ataque de inyección de bus podría enviar una alerta falsa positiva a un cliente SCADA de alto nivel.

¿POR QUÉ?

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

REFERENCIAS

Norma / marco	Mapeo
MITRE ATT&CK ICS	**Táctica :** TA009 - Inhibir la función de respuesta **Técnica:** T0878 - Supresión de alarmas
ISA 62443-3-3	**SR 3.5:** Validación de entrada
ISA 62443-4-2	**CR 3.5:** Validación de entrada
ISA 62443-4-1	**SI-1:** Revisión de la aplicación de la seguridad
MITRE CWE	**CWE-754:** Verificación incorrecta de condiciones inusuales o excepcionales

Acerca del proyecto de programación segura de PLC

Durante muchos años, los controladores lógicos programables (PLC) han sido inseguros debido a su diseño. Varios años de personalización y aplicación de las mejores prácticas de TI dieron lugar a protocolos seguros, comunicaciones cifradas, segmentación de la red, etc. Sin embargo, hasta la fecha, no se ha prestado atención a la utilización de las características de los PLC (o SCADA/DCS) para la seguridad, ni a cómo programar los PLC teniendo en cuenta la seguridad. Este proyecto - inspirado en las actuales Prácticas de Codificación Segura para la Informática - llena ese vacío.

¿QUIÉN DEBE LEER Y PONER EN PRÁCTICA LAS PRÁCTICAS SEGURAS DE CODIFICACIÓN DEL PLC?

Estas prácticas han sido diseñadas para los ingenieros. El objetivo de este proyecto es proporcionar directrices a los ingenieros que crean software (lógica de escalera, diagramas 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 necesitan pocas o ninguna herramienta de software o hardware adicional para implementar estas prácticas. Todos ellos pueden encajar en el flujo de trabajo normal de programación y operación de PLC. Se necesita más que experiencia en seguridad, un buen conocimiento de los PLC que se protegerán, su lógica y el proceso subyacente para implementar estas prácticas.

¿CUÁL ES EL ALCANCE DE ESTA LISTA / CÓMO SE DEFINE LA CODIFICACIÓN DE PLC?

Para ajustarse al alcance de la lista de las 20 prácticas principales de codificación segura de PLC, las prácticas deben incluir cambios realizados directamente en un PLC. Lo que ve en este documento es una selección de las 20 principales de un mayor número de prácticas potenciales de codificación segura de PLC. También hay prácticas preliminares adicionales relacionadas con la arquitectura general, las HMI o la documentación. Estos no encajan en el ámbito de la codificación segura del PLC, pero podrían estar en una futura lista sobre el entorno seguro del PLC.

¿CUÁLES SON LAS VENTAJAS DE APLICAR PRÁCTICAS SEGURAS DE CODIFICACIÓN DEL PLC?

El uso de estas prácticas tiene claramente ventajas en materia de seguridad, sobre todo al reducir la superficie de ataque o permitir una resolución más rápida de los problemas en caso de que se produzca 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 fiable, más fácil de depurar y mantener, más fácil de comunicar y, posiblemente, también más sencillo. Además, las prácticas de codificación segura del PLC no sólo ayudan a los usuarios en caso de un atacante malintencionado, sino que también hacen que el código del PLC sea más robusto 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 seguras para los PLC”.

Después de la conferencia, Dale Peterson inició el proyecto Top 20. Jake Brodsky y Sarah Fluchs pasaron varias horas al teléfono para llevar al papel las prácticas de codificación segura de PLC propuestas por Jake. Posteriormente, Dale, Jake y Sarah crearon una plataforma en top20.isa.org, con el apoyo de la GCA de la ISA, para estructurar y recoger aportaciones adicionales de las comunidades de ingenieros y de seguridad de ICS.

Los debates y la consolidación de los textos de las prácticas, así como la confección de una lista de las 20 prácticas más relevantes, llevaron aproximadamente un año; el proceso se aceleró gracias a Vivek Ponnada, que además de contribuir y revisar el contenido, organizó llamadas periódicas hasta que se resolvieron todos los comentarios sobre las prácticas; Mohamed Abdelmoez Sakesli, que añadió todas las referencias de las normas en un gran esfuerzo, el equipo de MITRE CWE, que proporcionó las referencias de CWE en el último momento, Sarah, que recopiló el documento que está leyendo ahora, y Jake, Dale, John Cusimano, Dirk Rotermund, Josh Ruff, Thomas Rabenstein, Gus Serino, Walter Speth, Agustín Valencia Gil-Ortega, Marcel Rick-Cen y Al Ratheesh R, que hicieron aportaciones durante las llamadas regulares.

LISTA DE DONANTES

El Proyecto de Codificación Segura de PLC es, y sigue siendo, un verdadero esfuerzo comunitario, que no habría sido posible sin los innumerables colaboradores que comparten generosamente su tiempo y sus conocimientos sobre el PLC y la seguridad. Un total de 943 usuarios se registraron en la plataforma para debatir y contribuir. A continuación se presenta una lista alfabética de todos los que aceptaron explícitamente ser nombrados. Gracias a todos los que se han tomado la molestia de apoyar este proyecto.

Aagam Shah	Heiko Rudolph	Miguel Angel Frias
Adam Paturej	Isiah Jones	Mohamed Abdelmoez Sakesli
Agustin Valencia Gil-Ortega	Jacob Brodsky	Luna Eluvangal Chandran
Aitor García Almiñana	Javier Perez Quezada	Nahuel Iglesias
Alec Summers	J-D Bamford	Nalini Kanth
Al Ratheesh. R	Joe Weiss	NarasHMIa S. Himakuntala
Andreas Falk	John Cusimano	Omar Morando
Anton Shipulin	John Hoyt	Oscar J. Delgado-Melo
Arkaitz Gamino	John Powell	Päivi Brunou
Carlos Olave	John Kingsley	Peter Donnelly
Chris van den Hooven	Joseph J. Januszewski	Peter Jackson
Chris Sistrunk	Josh Ruff	Ravindra Deshakulakarni
Christos Alexopoulos	Josie Houghton	Rick Booij
Cris DeWitt	Jozef Sulwinski	Robert Albach
Dale Peterson	Juan Pablo Angel Espejo	Rushi Purohit
Dene Yandle	Khalid Ansari	Sarah Fluchs
Dennis Verschoor	Marc Weber	Sergei Biberdorf
Dirk Rotermund	Marcel Rick-Cen	Stephan Beirer
Edorta Echave García	Martin Huddleston	Steve Christey Coley
Gananand Kini	Massimiliano Zonta	Thomas Rabenstein
George Alex Holburn	Matthew Loong	Tim Gale
Gus Serino	Matthias Müller	Vivek Ponnada
Hakija Agic	Michael Thompson	Vytautas Butrimas
Hector Medrano	Michal Stepien	Walter Speth

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

Paseo de las Delicias, 30 - 2º. 28045 Madrid
+34 910 910 751
info@cci-es.org
www.cci-es.org
blog.cci-es.org
@info_cci
www.linkedin.com/in/centrociberseguridadindustrial

PLC Security
TOP 20 LIST

https://plc-security.com/
@securePLC