Smjernice za pisanje sigurnog PLC koda

Verzija 1.0 (15. lipanj 2021.)

PLC Security TOP 20 LIST

1. Modularna izvedba PLC koda

PLC kod treba biti izveden modularno (blokovski), i to tako da se koriste funkcije, funkcijski blokovi ili neku drugi vid programske podrutine. Pojedine programske module treba ispitivati izolirano i nezavisno.

2. Nadzor režima rada PLC uređaja

PLC uređaj treba biti u RUN načinu rada kad god je to moguće. Bilo koji drugi režim rada treba alarmirati operativno osoblje.

3. Izvršavanje programske logike u PLC uređaju

Programska logika čija je namjena nadzor i/ili upravljanje postrojenjem treba se izvršavati izravno u PLC uređaju. HMI uređaji nisu namijenjeni za ovu zadaću.

4. Korištenje statusnih riječi i zastavica za provjeru integriteta

Sve matematičke i slične programske pogreške treba pratiti, pamtiti i brojati pomoću ugrađenih statusnih riječi i pripadajućih zastavica (flagova).

5. Korištenje kriptografije i zaštitnih suma za provjeru integriteta programskog koda

Preporuča se korištenje kriptografskih hash funkcija (ili zaštitne sume ukoliko kriptografske funkcije nisu raspoložive) za provjeru integriteta cjelokupnog PLC programskog koda. Bilo kakva nepredviđena promjena u integritetu programskog koda treba generirati alarm.

6. Provjera kronometara i brojača

Ukoliko se postavne vrijednosti kronometara i/ili brojača pišu eksterno (izvan PLC uređaja), treba provjeravati njihov fizikalni i općeniti smisao.

7. Provjera komplementarnih ulaza/izlaza

Ukoliko postoje, komplementarni signali ne smiju biti aktivni/aktivirani istovremeno. Bilo koji slučaj fizičke ili logičke diskrepancije treba generirati alarm. Komplementarne ulazne signale valja analizirati nezavisno, a aktivacije komplementarnih izlaza treba razdvojiti vremenskom zadrškom kako bi se izbjegle štete na aktuatorima.

8. Provjera HMI unosa na strani PLC uređaja

Manipulacija PLC varijablama od strane HMI uređaja treba biti ograničena na strogo definirani očekivani opseg. Provjeru tog opsega treba vršiti ne samo na HMI uređaju već i sa strane PLC uređaja. Vrijednosti izvan očekivanog opsega treba pokušati spriječiti, a ukoliko se one ipak dogode o tome treba izvijestiti operativno osoblje.

9. Provjera neizravnog adresiranja

Neizravno adresiranje valja provjeravati na način da se rubni dijelovi matrica namjeno kompromitiraju (popune neispravnim vrijednostima), a sve s ciljem detekcije namjernog ili nenamjernog pogrešnog adresiranja (adresiranja izvan granica dimenzija te matrice)

10. Dodjeljivanje memorijskih registara ovisno o funkciji

Memorijske registre treba dodjeljivati ovisno o funkciji za koju su predviđeni (čitanje, pisanje i provjera). Cilj ove preporuke jest sprječavanje prekoračenja kapaciteta međuspremnika te onemogućavanje neautoriziranog pisanja u zaštićena memorijska područja.

11. Provjera plauzibilnosti

Instrumentacija treba biti odabrana tako da omogućuje provjeru plauzibilnosti, i to na način da se unakrsno provjeravaju različita mjerila.

12. Provjera fizikalnog smisla

Valja osigurati da operativno osoblje može unositi samo one procesne vrijednosti koje imaju fizikalni smisao i koje se uklapaju u trenutni kontekst procesa. Isto tako preporuča se mjeriti trajanje važnih procesa i procesnih koraka te alarmirati u slučaju odstupanja od predviđenih vrijednosti. Neobično miran ili potpuno neaktivan proces također je razlog za oprez.

13. Onemogućavanje nekorištenih komunikacijskih portova i protokola

Većina modernih PLC uređaja je opremljena sučeljima za korištenje višestrukih komunikacijskih protokola, od kojih je većina omogućena u zadanim tvorničkim postavkama. Svi komunikacijski portovi i protokoli koji nisu nužni za rad sustava trebaju biti onemogućeni.

14. Ograničavanje komunikacijskih sučelja prema trećim stranama

Način spajanja i podaci koji su raspoloživi sustavima trećih strana trebaju biti ograničeni i pod stalnim nadzorom. Komunikacijska i podatkovna sučelja prema trećim stranama moraju biti jasno definirana i ograničena samo na podatkovni promet koji je potreban za normalno funkcioniranje sustava.

15. Definiranje sigurnog procesnog stanja kod pokretanja PLC uređaja

Jasno definirati sigurno stanje procesa kod pokretanja PLC uređaja (npr. aktivacija aktuatora, deaktivacija aktuatora, zadržavanje zadnjeg zapamćenog stanja…)

16. Nadzor trajanja izvršavanja PLC koda

Preporuča se periodički (svake 2-3 sekunde) izmjeriti vrijeme trajanja izvršavanja programskog koda, a izmjerenu vrijednost poslati na neki od HMI uređaja za potrebe vizualizacije na dijagramu.

17. Nadzor vremena neprekidnog rada

Vrijeme neprekidnog rada valja pratiti kako bi se znala točna vremena i ukupan broj zastoja. Izmjerene vrijednosti treba bilježiti na HMI uređajima.

18. Zapisivanje uzroka zastoja

Događaje koji su prouzrokovali zastoj PLC-a valja zabilježiti i prikazati na HMI sustavu za prikaz poruka, kako bi se mogli analizirati prije nego se pokuša ponovno pokrenuti PLC. Za što kvalitetniju analizu preporuča se vremenski sinkronizirati sve uređaje.

19. Nadzor angažmana memorije

Preporuča se za svaki korišteni kontroler mjeriti angažman raspoloživih memorijskih resursa. Izmjerene vrijednosti valja zabilježiti i prikazati na HMI uređajima.

20. Nadzor lažno pozitivnih i lažno negativnih kritičnih alarma

Nakon identifikacije kritičnih alarma valja implementirati logiku za detekciju lažnih uvjeta koji ih okidaju. Svaki takav slučaj mora generirati poruku.

O projektu smjernica za pisanje sigurnog PLC koda

1. Modularna izvedba PLC koda

PLC kod treba biti izveden modularno (blokovski), i to tako da se koriste funkcije, funkcijski blokovi ili neku drugi vid programske podrutine. Pojedine programske module treba ispitivati izolirano i nezavisno.

Cilj smjernice	Ciljana skupina
Integritet programskog koda	Sistem integratori

Obrazloženje

Centralizacija PLC programskog koda na jednom mjestu (u jednoj programskoj rutini ili u jednom organizacijskom bloku) smatra se lošom praksom. Ispravan pristup jest organizacija programskog koda na način da se on logički grupira odgovarajuće subrutine: funkcije ili funkcijske blokove. Veličinu svake pojedine subrutine i vrijeme njezinog izvršavanja valja redovito provjeravati radi detekcije eventualnih nepravilnosti.

Oni dijelovi programske logike koji funkcioniraju neovisno trebaju biti izvedeni kao odvojene cjeline. Ovakva praksa olakšava provjeru ulaznih varijabli, kontrolu pristupa, verifikaciju integriteta programskog koda i slično.

Programski kod koji je izveden modularno olakšava ispitivanje i kontrolu integriteta pojedinih modula. Naime, kada je programski blok temeljito ispitan, bilo kakva promjena programskog koda unutar samog modula može biti jednostavno detektirana provjerom kriptografskog hash otiska (ukoliko takva opcija postoji na korištenom PLC uređaju). Na ovaj način svaki pojedini modul može pojedinačno biti provjeren za vrijeme FAT/SAT procedure ili ukoliko se sumnja na integritet programskog koda uslijed kibernetičkog incidenta.

Primjer

Program koji upravlja plinskom turbinom može se podijeliti na cjeline: „Inicijalizacija“, „Otvaranje ulaznog ventila“, „Upravljanje ventilom drenaže“ i slično. Osim što omogućuje sistematičnu primjenu standardne programske logike za upravljanje postrojenjem ove vrste, predložena organizacija programskog koda olakšava dijagnostiku i ubrzava pronalazak i otklanjanje pogrešaka uslijed eventualnog sigurnosnog incidenta.

Dodatna prednost korištenja rigorozno testiranih programskih blokova jest mogućnost zaštite od izmjene programskog koda (zaštita lozinkom ili kriptografijom), što smanjuje mogućnost pogrešaka i zlonamjernih manipulacija algoritmom upravljanja.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Mogućnost detekcije novih ili nepoznatih dijelova programskog koda koji bi mogli biti maliciozni. Viši stupanj standardizacije i konzistencije programskog koda. Zaštita od neautoriziranih izmjena programskog koda.
Pouzdanost	Smanjena mogućnost uobičajenih programskih pogrešaka, poput beskonačnih petlji i sličnog, uslijed kojih može doći do ozbiljnih poremećaja u algoritmu upravljanja ili čak do zastoja u radu PLC uređaja.
Održavanje	Logično podijeljen i modularno izveden PLC kod ne samo da je jednostavniji za ispravljanje pogrešaka (eng. debugging), već je i lakši za nadogradnju i održavanje. Osim toga, ovako napisan programski kod je prenosiv i ponovno iskoristiv.

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic:** TA002 - Execution **Technique:** T0844 - Program Organization Units
ISA 62443-3-3	**SR 3.4**: Software and information integrity
ISA 62443-4-2	**CR 3.4**: Software and information integrity
ISA 62443-4-1	**SI-2**: Secure coding standards
MITRE CWE	**CWE-1120:** Excessive Code Complexity **CWE-653:** Insufficient Compartmentalization

2. Nadzor režima rada PLC uređaja

PLC uređaj treba biti u RUN načinu rada kad god je to moguće. Bilo koji drugi režim rada treba alarmirati operativno osoblje.

Cilj smjernice	Ciljana skupina
Integritet programskog koda	Sistem integratori Pružatelji usluge održavanja Vlasnici imovine

Obrazloženje

Programski kod treba biti napisan tako da ima mogućnost detekcije režima rada PLC uređaja (STOP mod, PROGRAM mod i slično, ovisno o proizvođaču PLC-a). Shodno tome, ukoliko PLC uređaj nije u predviđenom režimu rada valja o tome obavijestiti operativno osoblje. Neki PLC-ovi imaju mogućnost provjere zaštitne sume (eng. checksum) kao ugrađeni mehanizam zaštite od neovlaštenih promjena programskog koda. Ukoliko takva mogućnost nije na raspolaganju, treba koristiti neizravne indikatore potencijalnih problema koji mogu proizaći iz rada u nepredviđenom režimu:

- Ukoliko PLC nije u RUN načinu rada treba bez iznimke generirati alarm. Ukoliko se radi o predviđenom zastoju operateri mogu potvrditi taj alarm i nastaviti s radom.
- HMI uređaji trebaju biti podešeni (programirani) tako da se alarm nepredviđenog režima rada ponavlja na kraju svake smjene, s ciljem prijenosa informacije o radovima u tijeku koji mogu utjecati na sigurnost procesa.

Iznimno, ukoliko se radi o fazi ispitivanja i puštanja u pogon ovakvi alarmi mogu biti onemogućeni, ali tada valja voditi računa o tome da se dio postrojenja koji se ispituje izolira od drugih dijelova procesne mreže.

Primjer

Ukoliko PLC uređaj nema hardversku preklopku za odabir režima rada u kojem su programske preinake nemoguće, valja razmotriti korištenje bilo kojeg drugog mehanizma koji može otežati ili potpuno onemogućiti izmjene programskog koda (npr. zaštita inženjerskog softvera lozinkom).

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Režim rada (RUN / EDIT / WRITE; za PLC-ove proizvođača Allen-Bradley RUN / PROG / REM) izravno određuje mogućnosti udaljene manipulacije programskim kodom. Ukoliko je PLC u ranjivom režimu rada (npr. REM za A-B PLC-ove) tehnički je moguće iskoristiti komunikacijska sučelja za neovlaštene preinake programskog koda čak i u RUN režimu rada.
Pouzdanost	/
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA009 - Inhibit Response Function **Technique:** T0858 - Utilize/Change Operating Mode
ISA/IEC 62443-4-1	**SI-1** : Security implementation review

3. Izvršavanje programske logike u PLC uređaju

Programska logika čija je namjena nadzor i/ili upravljanje postrojenjem treba se izvršavati izravno u PLC uređaju. HMI uređaji nisu namijenjeni za ovu zadaću.

Cilj smjernice	Ciljana skupina
Integritet programskog koda	Sistem integratori Pružatelji usluge održavanja Vlasnici imovine

Obrazloženje

Većina modernih HMI uređaja ima mogućnost izvršavanja korisničkog programskog koda. Najčešće se radi o programskim skriptama čija je prvotna namjena proširenje osnovnih mogućnosti vizualizacije, alarmiranja i izvještavanja. Međutim, prisutna je i praksa da se ove mogućnosti koriste za pisanje i izvršavanje programskog koda koji bi zbog integriteta i koherencije trebao biti izvršen u PLC uređaju.

Generalna je preporuka da izračune bilo koje vrste valja izvršavati što je moguće bliže mjestu nastanka signala (u postrojenju). Frekvencija osvježavanja HMI uređaja nije dovoljna za ispravan rad totalizatora i integratora. Osim toga, uvijek je prisutna latencija između PLC i HMI uređaja. Nadalje, u slučaju ponovnog pokretanja (restarta) HMI uređaja vrijednosti totalizatora i integratora ostaju zapisane u PLC uređaju i transferiraju se na HMI čim je to moguće.

Osobito valja izbjegavati situaciju da se u HMI uređaju izvršava bilo kakav programski kod vezan za funkcije procesne sigurnosti, kao što su primjerice dozvole, zaštite i blokade.

Za funkciju kratkoročne i/ili dugoročne analitike procesa također valja izbjegavati HMI uređaje i za tu namjenu koristiti specijalizirane programske pakete – data historian – iz čije se baze podataka upitom može dohvatiti povijesne vrijednosti za usporedbu sa trenutno izračunatim vrijednostima u PLC uređaju. Značajnije razlike kod važnijih mjerenja koje ne mogu biti objašnjene tehnološkim ili tehničkim razlozima trebaju generirati upozorenja ili alarme.

Primjeri

- Programski kod za omogućavanje i/ili onemogućavanje upravljačkih gumba: U programski kod PLC uređaja obavezno uključiti logiku za omogućavanje/onemogućavanje upravljačkih tipki (gumba) na HMI. U suprotnom postoji rizik od zadavanja i izvršavanja komandi za koje ne postoje tehnički ili tehnološki uvjeti.
- Vremenska zadrška između uzastopnih pokušaja pokretanja izvršnih elemenata kao što je motorni pogon ili elektromotorni ventil treba se uvijek izvršavati na PLC uređaju, nikako na HMI.
- Alarmne granice također trebaju biti dio PLC programskog koda.
- Primjer spremnika vode čija se razina mijenja kroz vrijeme: PLC koji bez ograničenja nadzire ulazni i izlazni protok može kroz mehanizam sumiranja nadzirati volumen (razinu) u tanku. HMI uređaj također može vršiti ovu zadaću, ali za to mora dobiti prihvaćene i obrađene procesne podatke iz PLC uređaja sa ispravnim vremenskim žigom, što je značajno otežano uslijed latencije podatkovnog prometa ili procedure ponovnog pokretanja HMI uređaja.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	**1.** Olakšava se konzistencija u praćenju izmjena programskog koda. Naime, HMI uređaji također imaju neke mehanizme za verifikaciju izmjena programskog koda, ali je taj mehanizam u pravilu odvojen i manje rigorozan od verifikacije izmjena PLC programskog koda. Transferom programske logike na HMI uređaje gubi se na jasnoći logike koja nadzire i upravlja postrojenjem, uslijed čega može doći i do gubitka nekih važnih spoznaja. Isto tako, HMI uređaji nemaju mogućnost forsiranja ili pregleda izmijenjenih signala, pa je samim time otežana detekcija onih izmjena koje mogu imati važne implikacije na sigurnost procesa. Sve to zajedno rezultira činjenicom da je izmjene programskog koda u HMI uređajima jako teško ili gotovo nemoguće uključiti u procedure praćenja izmjena. **2.** Iz perspektive napadača, puno je teže manipulirati programskog logikom koja se izvršava na distribuirano na više PLC uređaja nego programskim kodom koji se izvršava centralizirano, na HMI uređaju. **3.** Ukoliko se neki dio logike omogućavanja i/ili onemogućavanja upravljačkih tipki ne izvršava na PLC uređaju, hipotetskom je napadaču olakšana izravna manipulacija ulazno/izlaznim varijablama PLC-a, jer je ispravna signalizacija iz postrojenja ionako već upitna.
Pouzdanost	**1.** Različiti izračuni (totali, integratori i slično) su puno točniji, precizniji i pouzdaniji kada se izvršavaju blizu mjesta nastanka signala i u ciklusu PLC uređaja. Na taj se način izbjegavaju potencijalni problemi poput aproksimacija izračunatih vrijednosti kada HMI uređaj ne radi (primjerice za vrijeme restarta), jer se takve vrijednosti najčešće akumuliraju i pamte u retentitivnoj memoriji PLC uređaja. **2.** Različiti izvori za logiku dozvola, zaštita i blokada mogu biti uzrok neočekivanih problema u radu postrojenja. Primjerice, to može biti uzrokovano različitim tehnologijama koje se koriste na razini HMI uređaja: nekontrolirana promjena u nekoj od tih tehnologija može lako dovesti do nekonzistencije u vizualizaciji postrojenja ili čak do grešaka i pogrešnih manipulacija uslijed kojih može biti ugrožena sigurnost.
Održavanje	Čitljivost, jasnoću i prenosivost programskog koda puno je lakše postići između PLC uređaja nego između HMI uređaja.

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter
ISA 62443-3-3	**SR 3.6** : Deterministic Output
ISA 62443-4-2	**CR 3.6** : Deterministic Output

4. Korištenje statusnih riječi i zastavica za provjeru integriteta

Sve matematičke i slične programske pogreške treba pratiti, pamtiti i brojati pomoću ugrađenih statusnih riječi i pripadajućih zastavica (flagova).

Cilj smjernice	Ciljana skupina
Integritet programskog koda	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja Vlasnici imovine

Obrazloženje

Ukoliko PLC program duže vrijeme radi normalno, a onda odjednom dođe do dijeljenja s nulom, to je indikator mogućih problema i dobar razlog za istragu i analizu. Ukoliko neki PLC uređaj komunicira izravno s drugim PLC uređajem u programu koji izvršava tu programsku rutinu dođe do dijeljenja s nulom, to je također razlog za oprez.

Mnogi PLC programeri zanemaruju obradu grešaka te uslijed vjerovanja u potpunu ispravnost programskog koda često dovode PLC u nepoželjan režim rada (npr. FAULT ili STOP). Programski kod (ili pojedine njegove dijelove – logičke cjeline) valja testirati na način da mu se na ulaz dovedu podaci koji su izvan očekivanog raspona.

Za *firmware*, programsku logiku i protokolarni složaj valja dodijeliti odvojene i zaštićene segmente memorije. Posebno valja obratiti pažnju na testiranje zloupotrebe protokolarnih složaja. Neuobičajena vrijednost zastavice ili nekog drugog statusnog indikatora može biti znak zlouporabe protokolarnog složaja.

Primjer

Stanje greške PLC uređaja koje je prouzrokovano obradom (čitanjem ili pisanjem) podataka koji su izvan predviđenog raspona je vrlo čest slučaj. Indeks koji se nalazi izvan dimenzija matrice, negativne početne vrijednosti mjerača vremena ili dijeljenje s nulom samo su neki od slučajeva. Posebno valja obratiti pažnju na sljedeće zastavice:

- Dijeljenje s nulom
- Prelijevanje brojača
- Negativne zadane vrijednosti brojača ili mjerača vremena
- Čitanje ili pisanje ulazno/izlaznih podataka koji se nalaze izvan raspona procesne slike

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Maliciozna aktivnost na PLC uređaju može uključivati promjene u programskoj logici, potpuno nove dijelove programskog koda, učitavanje novog procesnog recepta ili vanjsku logiku, a sve sa svrhom kako bi se aktivirale funkcije ili generirale poruke koje mogu pomoći u postizanju napadačevih ciljeva. S obzirom da većina PLC uređaja ne podržava kriptografsku provjeru integriteta programskog koda, provjera statusnih riječi, zastavica i sličnih programskih indikatora mogu pomoći u identifikaciji malicioznih aktivnosti.

Što je poboljšano?	Kako je poboljšano?
Pouzdanost	Ukoliko se shvate dovoljno ozbiljno, statusi koji upućuju na malicioznu aktivnosti mogu pomoći da se programski kod izvršava bez grešaka. Osim toga, analiza zastavica pomaže u pronalasku izvora greške.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic** : TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter
ISA 62443-3-3	**SR 3.5**: Input Validation **SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.5**: Input Validation **CR 3.6**: Deterministic Output
ISA 62443-4-1	**SI-2**: Secure coding standards **SVV-1**: Security requirements testing
MITRE CWE	**CWE-128:** Wrap-around **CWE-190:** Integer Overflow **CWE-369:** Divide by Zero **CWE-754:** Improper Check for Unusual or Exceptional Conditions

5. Korištenje kriptografije i zaštitnih suma za provjeru integriteta programskog koda

Preporuča se korištenje kriptografskih hash funkcija (ili zaštitne sume ukoliko kriptografske funkcije nisu raspoložive) za provjeru integriteta cjelokupnog PLC programskog koda. Bilo kakva nepredviđena promjena u integritetu programskog koda treba generirati alarm.

Cilj smjernice	Ciljana skupina
Integritet programskog koda	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja Vlasnici imovine

Obrazloženje

A) Zaštitne sume

Ukoliko kriptografske hash funkcije nisu na raspolaganju, zaštitna suma je prihvatljiva alternativa. Neki PLC uređaji generiraju jedinstvenu zaštitnu sumu u trenutku kada se programski kod spušta na sam PLC uređaj. Takva zaštitna suma treba biti dokumentirana od strane proizvođača ili sistem integratora odmah po završetku pogonskog testa prihvatljivosti (SAT) te treba biti sastavni dio kada se razmatraju uvjeti održavanja.

Ukoliko funkcija generiranja zaštitne sume nije podržana na korištenom kontroleru, treba ju generirati na nekom od HMI uređaja ili na inženjerskoj stanici. Tako generiranu sumu valja periodički (npr. jednom dnevno) uspoređivati sa originalom koji se nalazi na PLC uređaju. Ovakav pristup ne pruža zaštitu u realnom vremenu, ali je dovoljno dobra metoda detekcije neovlaštene modifikacije programskog koda.

Vrijednost zaštitne sume se može obrađivati i u programskog kodu, što olakšava generiranje i propagaciju odgovarajućeg alarma na SCADA računala i/ili poslužitelje za pohranu povijesnih podataka.

B) Hash funkcije

Procesorska snaga PLC uređaja obično nije dovoljna da bi se kriptografske hash funkcije računale u realnom vremenu. Takav pokušaj čak može rezultirati odlaskom PLC-a u STOP režim zbog prekoračenja maksimalno vremena predviđenog za izvršavanje cikličkih funkcija (eng. Watchdog). Međutim, moderna razvojna okruženja za programiranje PLC uređaja na PC računalima imaju dovoljno procesorskih i memorijskih resursa da se hash funkcije mogu bez problema računati i pohranjivati na PLC uređaj ili na neku alternativnu lokaciju.

Primjeri

Proizvođači PLC uređaja koji podržavaju generiranje zaštitnih suma:

- Siemens (vidi primjer)
- Rockwell

Alternativni programski paketi pomoću kojih je moguće generirati zaštitnu sumu:

- Version dog
- Asset Guardian

- PAS

Implementacija za Siemens kontrolere

Primjer implementacije za Siemens S7-1500 PLC:

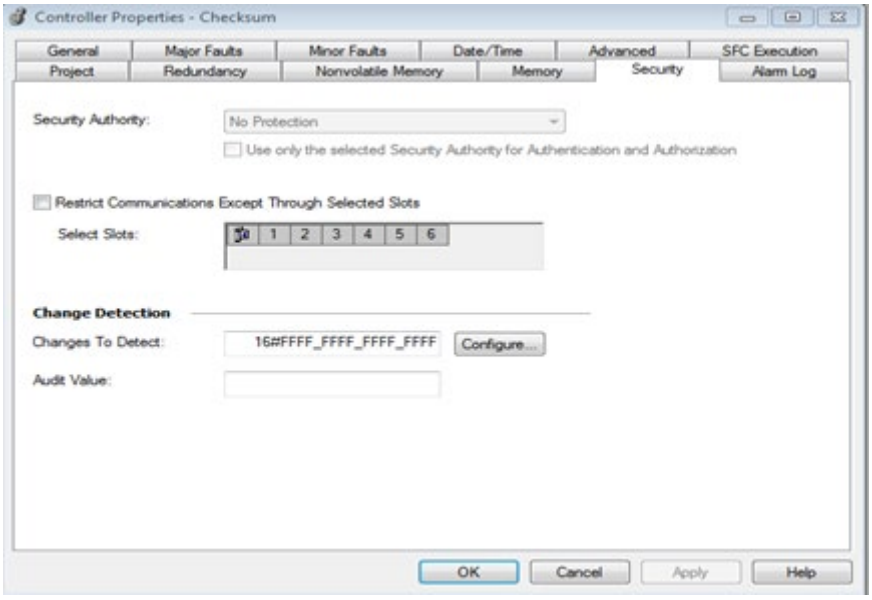
GetChecksum funkcijski blok čita zaštitnu aktualnu zaštitnu sumu, koja se skriptom SAT-Checksum može pohraniti kao referenca. Odstupanje se pohranjuju koristeći Datalog funkciju.

	Datum	UTC vrijeme	Referentna suma	Aktualna suma
1	21.11.2019.	9:57:11	84 2A 76 DF 5B 31 F4 16	FF 2C EA 71 44 D7 81 84
2	21.11.2019.	9:57:33	FF 2C EA 71 44 D7 81 84	FF 2C EA 71 44 D7 81 84
3	21.11.2019.	9:58:17	FF 2C EA 71 44 D7 81 84	5B 7C 57 7E E2 3E EF C3
4	21.11.2019.	9:58:36	FF 2C EA 71 44 D7 81 84	5B 7C 57 7E E2 3E EF C3
5	21.11.2019.	9:58:44	5B 7C 57 7E E2 3E EF C3	5B 7C 57 7E E2 3E EF C3

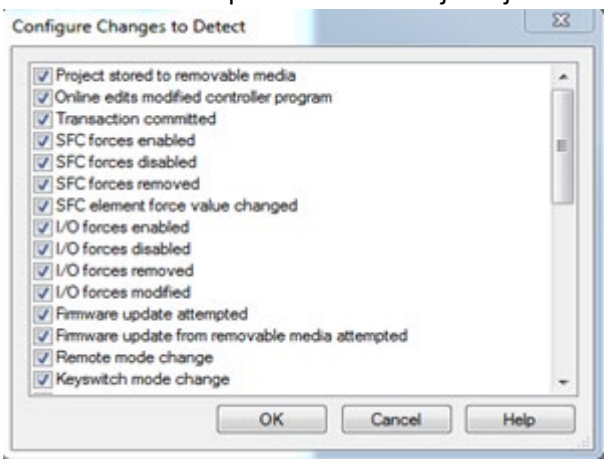
Implementacija za Rockwell kontrolere

U nastavku je opisan (parcijalan) primjer kako neka organizacija može razviti mehanizam detekcije pokušaja neovlaštene modifikacije PLC programskog koda. Valja naglasiti kako je opisani primjer specifičan za Rockwell Automation ControlLogix PLC te da nipošto nije kompletan: valja ga shvatiti samo kao ilustraciju kako je moguće pohraniti trenutno stanje PLC uređaja u PLC memorijske registre. Tako pohranjena informacija se kasnije može koristiti za razne namjene: primjerice kako bi se generirao alarm promjene konfiguracije ili za pohranu u kratkoročnu i/ili dugoročnu arhivu za slučaj naknadne analize ili forenzike incidenta.

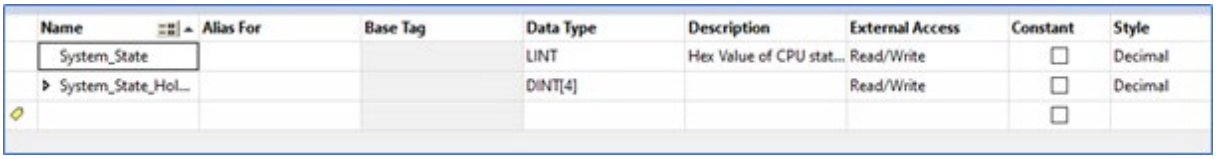
1. 1. U dijaloškom okviru „Controller Properties“ -> „Change detection“ valja pritisnuti gumb „Configure“



2. U novootvorenom prozoru za selekciju valja odabrati sve ponuđene stavke

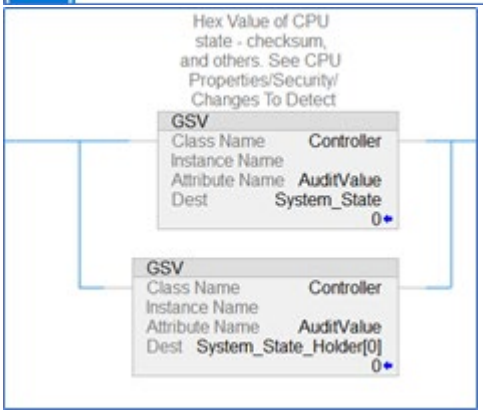


3. Kreirati procesni tag za pohranu informacije o trenutnom stanju PLC-a. Tip podatka može biti LINT ili matrica od 2 DINT-a



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. Pomoću Get System Values (GSV) instrukcije valja pročitati trenutno stanje CPU-a te ga pohraniti u ranije kreirani procesni tag. Taj procesni tag se nadalje može koristiti u PLC logici ili za prikaz na HMI



Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Detekcija neovlaštenih promjena konfiguracije ili programskog koda od ključne je važnosti za siguran i pouzdan rad postrojenja.
Pouzdanost	Provjera zaštitne sume omogućava provjeru da je programski kod koji se izvršava odobren od strane proizvođača ili sistem integratora.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA002 - Execution, TA010 - Impair Process Control **Technique:** T0873 - Project File Infection, T0833 - Modify Control Logic
ISA 62443-3-3	**SR 3.4** : Software and information integrity
ISA 62443-4-2	**CR 3.4** : Software and information integrity **EDR 3.12** : Provisioning product supplier roots of trust
ISA 62443-4-1	**SI-1** : Security implementation review **SVV-1** Security requirements testing
MITRE CWE	**CWE-345:** Insufficient Verification of Data Authenticity • **(child) CWE-353:** Missing Support for Integrity Check • **(child) CWE-354:** Improper Validation of Integrity Check Value

6. Provjera kronometara i brojača

Ukoliko se postavne vrijednosti kronometara i/ili brojača pišu eksterno (izvan PLC uređaja), treba provjeravati njihov fizikalni i općeniti smisao.

Cilj smjernice	Ciljana skupina
Integritet programskog koda	Sistem integratori Pružatelji usluge održavanja Vlasnici imovine

Obrazloženje

S tehničke strane, kronometri i brojači mogu biti pretpodešeni na bilo koju vrijednost. Iz tog je razloga potrebno provjeravati (i ograničavati) navedena predpodešenja kako bi ona ostala u predvidivom i fizikalno (procesno) smislenom opsegu.

Ukoliko se navedene vrijednost zadaju sa HMI uređaja, valja imati na umu sljedeće smjernice:

- Predpodešenja ne treba pisati izravno u kronometre i brojače, već ih proje toga treba provjeriti kontrolnim algoritmom
- Predpodešenja treba provjeravati (i) na samom PLC uređaju

Validacija ovih parametara je zadatak koji se relativno jednostavno može implementirati u PLC uređaju (nije potrebno imati mrežne uređaje s mogućnošću duboke inspekcije paketa). Naime, PLC najbolje poznaje procesno stanje i kontekst u okviru kojeg se može napraviti provjera ispravnosti i fizikalne smislenosti promatranih varijabli.

Primjer

Za vrijeme pokretanja PLC uređaja (eng. PLC *Startup* rutina) kronometri i brojači se obično reinicijaliziraju na početne vrijednosti. Ukoliko PLC logika sadrži alarm koji ima vremensku zadršku od 1.3s, a koja je zlonamjerno promijenjena na 5 minuta, alarm neće biti generiran, ili će biti generiran prekasno.

Ukoliko PLC logika sadrži brojač koji zaustavlja proces kada dosegne vrijednost od 10.000, promjenom te vrijednosti na 11.000 proces neće stati u trenutku kada je to predviđeno.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Ukoliko predpodešenja koja se pišu sa HMI uređaja nisu provjerena od strane PLC logike, cijeli jedan sloj sigurnosti je poništen, a to znači da je mrežnim uređajima trećih strana (od kojih su neki potencijalno maliciozni) dodijeljena neopravdano visoka razina povjerenja.
Pouzdanost	Implementacija ranije opisane provjere podiže sveukupnu razinu pouzdanosti rada, jer osim malicioznih može detektirati pogrešne vrijednosti koje su uzrokovane nepažnjom ili pogreškom operatera.
Održavanje	Uredno dokumentiran prihvatljiv raspon svih parametara postrojenja pomaže kod promjena na sustavu (promjene programske logike).

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic** : TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter
ISA 62443-3-3	**SR 3.5** : Input Validation
ISA 62443-4-2	**CR 3.5** : Input Validation
ISA 62443-4-1	**SI-2** : Secure coding standards **SVV-1** : Security requirements testing

7. Provjera komplementarnih ulaza/izlaza

Ukoliko postoje, komplementarni signali ne smiju biti aktivni/aktivirani istovremeno. Bilo koji slučaj fizičke ili logičke diskrepancije treba generirati alarm. Komplementarne ulazne signale valja analizirati nezavisno, a aktivacije komplementarnih izlaza treba razdvojiti vremenskom zadrškom kako bi se izbjegle štete na aktuatorima.

Cilj smjernice	Ciljana skupina
Integritet PLC varijabli Pouzdanost sustava	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Komplementarnost u kontekstu ove smjernice podrazumijeva nemogućnost istovremene pojave dva signala (ulaza ili izlaza): komplementarni signali su međusobno isključivi. Istovremena pojava takva dva signala ukazuje na kvar ili na eventualnu malicioznu aktivnost. Nažalost, inženjeri često ne vrše provjeru ovakvih signala, niti se osiguravaju dodatnim mehanizmima kako ne bi došlo do aktivacije komplementarnih izlaza.

Navedenu provjeru i zaštitne mehanizme preporučeno je implementirati u PLC uređaju, jer je unutar PLC logike najjednostavnije procijeniti procesno stanje i općenito kontekst u okviru kojeg se validacija provodi. Stanje komplementarne signala lakše je pratiti ukoliko se toj zadaći podredi i konvencija imenovanja (npr. ulaz 1 i ulaz 2).

Scenarij vrijedan razmatranja je i situacija u kojoj se ne provjeravaju varijable koje aktiviraju izlaze, ili ako se ne provjeravaju blagovremeno. U tom slučaju može doći do istovremene aktivacije komplementarnih izlaza, a to može fizički oštetiti aktuatore.

Primjer

Primjeri komplementarnih signala:

- POKRENI I ZAUSTAVI
  - Nezavisni izlazi za pokretanje i zaustavljanje aktuatora: Dobra je inženjerska praksa imati fizički i logički odvojene te međusobno neovisne signale za pokretanje i zaustavljanje aktuatora (nepoželjna alternativa je jedna preklopna varijabla). Ovakva konfiguracija je robusnija jer je potencijalni napad teže izvesti ukoliko se istovremeno moraju aktivirati dva izlaza. Osim toga, odvojeni signali olakšavaju izradu kontrolne logike, jer se istovremena aktivacija može trajno onemogućiti na razini PLC logike.
  - Vremensko zatezanje ponovnog pokretanja: Gdje god je to moguće, valja razmotriti implementaciju vremenskog zatezanja ponovnog pokretanja, kako bi se izbjegla mogućnost rapidnog mijenjanja stanja, što također može oštetiti aktuatore.
- NAPRIJED I NATRAG
- OTVORI I ZATVORI

Primjeri komplementarnih signala koji mogu izazvati štetu:

Ukoliko dizajn PLC-a/MCC-a tolerira diskretne izlaze za aktuatore, radi se o ranjivosti koju potencijalni napadač može iskoristiti kako bi fizički oštetio aktuator. Dobro poznati scenariji korištenja izlaza s ciljem uzrokovanja štete se uglavnom odvijaju na razini MCC-a, iako nadzorne i kontrolne mehanizme

valja primijeniti u svim slučajevima gdje preklop može prouzrokovati štetu. Dokaz koncepta da rapidno mijenjanje stanja izlaza može prouzročiti štetu je proveden u laboratoriju za sigurnost u Idahu (Aurora Generator Test), gdje je nesinkronizirana izmjena stanja prouzročila oštećenje zaštitnog prekidača.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	1. Ukoliko PLC program ne vodi računa o tome što će se desiti u slučaju istovremenih komplementarnih izlaza, radi se o potencijalno iskoristivom vektoru napada. 2. Mogućnost generiranja upozorenja da su u postrojenju dešava nešto neobično (pogreška operatera, kvar opreme ili maliciozna aktivnost). 3. Mogućnost eliminacije scenarija fizičkog oštećenja aktuatora uslijed istovremeno aktiviranih komplementarnih izlaza.
Pouzdanost	1. Mogućnost lociranja mehaničkog kvara koji je prouzrokovao istovremeno aktivne komplementarne ulaze (primjerice mehanički zaglavljen krajnji kontakt). 2. Osim uslijed kvara ili maliciozne aktivnosti, istovremena aktivacija komplementarnih izlaza (ili rapidna aktivacija jednog izlaza) može biti i nenamjerna, uslijed pogreške. Prema tome, provjera i implementacija zaštitnih mehanizama svakako podiže ukupnu pouzdanost sustava.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter, T0806 - Brute Force I/O
ISA 62443-3-3	**SR 3.5**: Input Validation **SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.5**: Input Validation **CR 3.6**: Deterministic Output
ISA 62443-4-1	**SI-2**: Secure coding standards **SVV-1**: Security requirements testing
MITRE CWE	**CWE-754:** Improper Check for Unusual or Exceptional Conditions

8. Provjera HMI unosa na strani PLC uređaja

Manipulacija PLC varijablama od strane HMI uređaja treba biti ograničena na strogo definirani očekivani opseg. Provjeru tog opsega treba vršiti ne samo na HMI uređaju već i sa strane PLC uređaja. Vrijednosti izvan očekivanog opsega treba pokušati spriječiti, a ukoliko se one ipak dogode o tome treba izvijestiti operativno osoblje.

Cilj smjernice	Ciljana skupina
Integritet PLC varijabli	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Validacija unosa u okviru ove smjernice može se primjerice odnositi na provjeru raspona varijabli, ispravnost tipa podatka ili na smislenost samog unosa u odnosu na proces.

Ukoliko PLC primi varijablu koja je izvan granica očekivanog opsega, zadaća je kontrolne PLC logike da:

- neispravnu varijablu zamjeni sa ranije definiranom zadanom vrijednošću koja ne utječe negativno na proces, a koja nadalje može poslužiti za generiranje poruke o primljenoj varijabli koja nije prihvatljiva, ili
- neispravnu varijablu zamjeni sa zadnjom primljenom ispravnom vrijednošću, a događaj poprati odgovarajućom porukom, to jest upozorenjem ili alarmom koji valja arhivirati za potrebe buduće analize.

Primjeri

Primjer 1

Na HMI uređaju je potrebno unijeti postavnu vrijednost tlaka za tlačni ventil. Predviđeni raspon unosa je 0-100, a korisnički unos se prenosi sa HMI uređaja u PLC varijablu V1. U ovom slučaju:

1. Polje za unos varijable na HMI uređaju je ograničeno na vrijednosti od 0 do 100 (decimalno).
2. PLC radi dopunsku provjeru, a primjer kontrolne logike je sljedeći:

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

Na ovaj se način, u slučaju neispravnog unosa postavne vrijednosti, postrojenje dovodi u pretpostavljeno sigurno stanje.

Primjer 2

Potreban je unos pragova za varijablu čiji je pretpostavljeni tip podatka INT2. Korisnički unos sa HMI uređaja se prenosi u PLC varijablu V2, a koja se nalazi u 16-bitnom podatkovnom registru.

1. Polje za unos varijable na HMI uređaju ograničeno je na raspon od -32768 (decimalno) do +32767 (decimalno).
2. PLC izvršava provjeru ispravnosti primljenog podatka na način da prati stanje varijable V3, a koja se u PLC logici nalazi odmah iza varijable V2:

```
IF V2 = -32768 OR IF V2 = 32767 AND V3 != 0,
SET V2 = 0 AND SET V3 = 0 AND SET DataTypeOverflowAlarm = TRUE.
```

Primjer 3

Preporuča se skaliranje procesne vrijednosti (engl. PV = Process Value), postavne vrijednosti (engl. SP = *Setpoint*) i manipulativne varijable (engl. MV = Manipulative Value) na konzistentne i bezdimenzijske brojeve kako bi se izbjegle moguće pogreške u skaliranju, a koje mogu dovesti do nehotičnih, ali ozbiljnih pogrešaka.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	**1.** Iako HMI uređaji u pravilu imaju ugrađene mehanizme provjere i kontrole ispravnosti korisničkog unosa, maliciozni pojedinac ima mogućnost modifikacije paketa i slanja nasumičnih vrijednosti u PLC varijable. **2.** PLC protokoli su u pravilu otvoreni protokoli, čije su specifikacije detaljno dokumentirane i dostupne širokom krugu korisnika, pa i potencijalnim napadačima. Prema tome, razvoj malicioznog programskog koda koji iskorištava sigurnosne propuste u komunikacijskim protokolima može biti trivijalan. Mapiranje ciljanih varijabli najčešće se odvija za vrijeme faze prikupljanja informacija, obično analizom mrežnog prometa. Na taj način potencijalni napadač može prikupiti dovoljno korisnih informacija za kreiranje malicioznog programskog koda a koji može biti iskorišten za neovlašteno manipuliranje PLC varijablama. Unakrsna provjera svih vrijednosti koje na PLC pristižu sa HMI ili drugih mrežnih uređaja prije njihove primjene u programskoj logici smanjuje mogućnost pojave neprihvatljivih vrijednosti (na način da se neispravne vrijednosti zamijene onima koje se smatraju sigurnima za proces).
Pouzdanost	/
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter
ISA 62443-3-3	**SR 3.5**: Input Validation **SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.5**: Input Validation **CR 3.6**: Deterministic Output
ISA 62443-4-1	**SI-2**: Secure coding standards **SVV-1**: Security requirements testing
MITRE CWE	**CWE-1320:** Improper Protection for Out of Bounds Signal Level Alerts

9. Provjera neizravnog adresiranja

Neizravno adresiranje valja provjeravati na način da se rubni dijelovi matrica namjeno kompromitiraju (popune neispravnim vrijednostima), a sve s ciljem detekcije namjernog ili nenamjernog pogrešnog adresiranja (adresiranja izvan granica dimenzija te matrice)

Cilj smjernice	Ciljana skupina
Integritet PLC varijabli	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Neizravno adresiranje je programska tehnika u okviru koje se vrijednost jednog podatkovnog registra nalazi u nekom drugom registru, dok je vrijednost originalnog registra zapravo adresa onog registra gdje se vrijednost nalazi.

Više je razloga zbog kojih se može koristiti neizravno adresiranje, primjerice:

- Motor s izmjenjivom brzinom vrtnje čija referenca se odabire iz tablice pretraživanja ovisno o stanju procesa ili neke druge varijable.
- Odabir pumpe (ili nekog drugog izvršnog elementa) s manjim brojem radnih sati.

Većina danas korištenih PLC uređaja nema ugrađene mehanizme zaštite od adresiranja izvan granica dimenzija ciljanog memorijskog područja (matrice), pa se preporuča takav mehanizam ugraditi na razini korisničkog PLC programskog koda. Cilj primijene takvog mehanizma jest spriječiti nepredviđen (nedeterministički) režim rada PLC uređaja uslijed adresiranja pogrešnih vrijednosti.

Primjeri

Primjer programiranja u programskom jeziku IL (engl. *Instruction List*)

Uz neke prilagodbe dolje navedeni pristup može se realizirati u obliku funkcijskog bloka kao univerzalni mehanizam provjere za sve probleme ovog ili sličnog tipa.

1. **Kreiranje, provjera i prilagodba matrica**

Najprije je potrebno provjeriti dimenzije matrice, koje bi trebale biti takve da omogućavaju binarne manipulacije na razini bajtova (eng. *binary sized*). Ukoliko matrica nije takva, treba ju proširiti (korigirati, prilagoditi). Primjerice, ukoliko matrica ima 5 elemenata (engl. *non-binary sized*), treba ju proširiti sa 3 elementa tako da ima 8 elemenata:

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

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

Nakon toga treba odabrati indeks matrice – za potrebe ovog primjera neka to bude 3 (imati na umu da indeksiranje matrice započinje s nulom)!

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

Index: 3

Odabranom indeksu sada treba dodati pomak kako bi se kompenzirali naknadno dodani krajevi matrice. Pomak može biti bilo koji broj veći od nule, u ovom slučaju to je 2:

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

Index sa pomakom: `3 + 2 = 5`

Sada treba izvršiti logičku operaciju AND između indeksa koji je uvećan za pomak i ukupne dimenzije matrice. U ovom slučaju dimenzija matrice je 8, a indeks uvećan za pomak je 7, pa će binarna maska matrice biti 0x07. Ovako izračunata maska osigurava da najveći mogući indeks odgovara relativnom položaju posljednje ispravne vrijednosti u matrici, što se može vidjeti u sljedećim primjerima:

`6 AND 0x07` za rezultat daje 6.
`7 AND 0x07` za rezultat daje 7
`8 AND 0x07` za rezultat daje 0.
`9 AND 0x07` za rezultat daje 1.

Na ovaj se način adresirana vrijednost uvijek nalazi negdje unutar dimenzija korigirane matrice.

2. Kompromitiranje rubnih dijelova matrice

Kompromitiranje rubnih dijelova matrice je opcionalno. Naime, adresiranje varijabli koje se nalaze izvan dimenzija matrice može se detektirati nadogradnjom programskog koda iz točke 1. Međutim, kompromitiranje krajeva matrice olakšava detekciju jer se za rezultat dobivaju brojevi koji u zadanom tehničkom ili procesnom kontekstu nemaju smisla.

Poanta je u tome da se rubni elementi matrice, a to su oni u kojima nisu sadržane korisne vrijednosti, u startu namjerno popune neispravnim („zatrovanim“) vrijednostima (npr. -1 ili 65535). Za matricu iz razmatranog primjera to izgleda ovako:

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

3. Čitanje vrijednosti iz korigirane matrice i usporedba s originalom (provjera neizravnosti)

U nastavku algoritma valja pročitati vrijednost iz originalne matrice (bez pomaka indeksa i bez maske). U navedenom slučaju, za originalnu matricu i nepromijenjeni indeks čita se vrijednost 51.

```
[21 31 41 51 61]
         ^
_______Index 3
```

4. Čitanje vrijednosti iz korigirane matrice i usporedba s originalom (provjera neizravnosti)

Kada je pročitana vrijednost iz originalne matrice s originalnim indeksom, treba primijeniti algoritam opisan u točki 1: pomak indeksa i maskiranje prilagođene matrice.

4a. Slučaj A: ispravno adresiranje

Najprije se izračuna pomak:
`Index + pomak = 3 + 2 = 5`

Nakon toga se računa maska:

```
5 AND 0x07 = 5
```

Potom se čita vrijednost iz korigirane matrice koristeći pomak indeksa:

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

Index + pomak: 5

Vrijednost koja je pročitana iz korigirane matrice je 51, što odgovara originalno pročitanoj vrijednosti. Indirektno adresiranje je provjereno i sve je u redu.

4b. Slučaj B: neispravno adresiranje (manipulirano neizravno adresiranje)

Za potrebe ovog primjera pretpostavit će se zlonamjerna manipulacija indeksom matrice (neka je ciljani indeks matrice 7, što se nalazi izvan granica dimenzija originalne matrice). Algoritam provjere je identičan kao u slučaju A):

Najprije se izračuna pomak:

```
Index + pomak = 7 + 2 = 9
```

Nakon toga se računa maska:

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

Potom se čita vrijednost iz korigirane matrice koristeći pomak indeksa:

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

Index s pomakom: **1**

Vrijednost koja je pročitana iz korigirane matrice je -1, što ne odgovara originalno pročitanoj vrijednosti. Indirektno adresiranje je neispravno.

5. Generiranje programske greške

U slučaju da je detektirana anomalija u neizravnom adresiranju (originalna i korigirana vrijednost nisu jednake), uzrok može biti (nenamjerna) programska pogreška (eng. Bug). U svakom slučaju radi se o događaju uslijed kojeg valja generirati poruku o grešci kvalitete softvera.

Osobito valja voditi računa o slučaju kada se iz matrice pročita kompromitirana vrijednost. U tom slučaju programski kod ne samo da čita neispravnu vrijednost, već ju pokušava čitati iz pogrešnog raspona memorijskog područja, a to pak može biti indikator da se u pozadini odvija maliciozna aktivnost. Takav događaj treba generirati poruku o pokušaju adresiranja izvan predviđenog memorijskog područja.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Većina PLC uređaja nema ugrađenu mogućnost provjere ispravnosti neispravnog adresiranja. Međutim, neispravno neizravno adresiranje može dovesti do dva potencijalno opasna scenarija: **Prvo**, moguće je da se tražene vrijednosti čitaju iz pogrešnih registara, što može dovesti do ozbiljnih posljedica na proces,

Što je poboljšano?	Kako je poboljšano?
	Drugo, moguće je da se tražene vrijednosti pišu u pogrešno memorijsko područje, što može dovesti do prepisivanja i nepovratnog gubitka podataka koji su važni za proces kojim se upravlja. U oba navedena slučaja može doći do vrlo ozbiljnih posljedica. Otegotna okolnost je to što su greške proizašle iz neizravnog adresiranja u pravilu teške za detekciju. Uzrok može biti programerska pogreška, ali i maliciozna aktivnost.
Pouzdanost	Navedeni mehanizam pomaže u detekciji programskih pogrešaka koje nisu uzrokovane malicioznom aktivnošću
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter
ISA 62443-3-3	**SR 3.5**: Input Validation **SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.5**: Input Validation **CR 3.6**: Deterministic Output
ISA 62443-4-1	**SI-2**: Secure coding standards **SVV-1**: Security requirements testing
MITRE CWE	**CWE-129:** Improper Validation of Array Index

10. Dodjeljivanje memorijskih registara ovisno o funkciji

Memorijske registre treba dodjeljivati ovisno o funkciji za koju su predviđeni (čitanje, pisanje i provjera). Cilj ove preporuke jest sprječavanje prekoračenja kapaciteta međuspremnika te onemogućavanje neautoriziranog pisanja u zaštićena memorijska područja.

Cilj smjernice	Ciljana skupina
Integritet PLC varijabli	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Priručna memorija, u tehničkom žargonu poznata i kao *scratchpad* memorija, jest izrazito volatilno memorijsko područje koje je podložno mnogim vrstama eksploatacije ukoliko se ne slijede pravila i dobre inženjerske prakse za njezino korištenje. Primjerice, pisanje u Modbus registre izvan njegovog raspona može dovesti do prepisivanja privremenog memorijskog područja koje se u programskom kodu koristi za neku vrstu izračuna.

Sasvim općenito govoreći, priručno memorijsko područje PLC-a je dohvatljivo drugim uređajima na mreži za funkcije čitanja i pisanja. Neki registri su predviđeni za funkciju čitanja sa HMI uređaja, dok su drugi predviđeni za zadavanje parametara sa istih tih HMI uređaja ili angažiranih SCADA sustava. Dodjeljivanje strogo određenog memorijskog područja za pojedine funkcije olakšava kontrolu razine prava pristupa, primjerice ako se koriste vanjski vatrozidi.

Primjeri funkcija za koje se mogu dodijeliti strogo definirani i odvojeni memorijski resursi jesu:

- Čitanje PLC varijabli
- Pisanje PLC varijabli (sa HMI uređaja, PLC-ova ili drugih mrežnih uređaja)
- Provjera ispravnosti upisanih vrijednosti
- Izračuni

Praksa da se pisanje PLC varijabli omogući samo u strogo definiranom memorijskom području pomaže u očuvanju integriteta memorije, a koji može biti kompromitiran uslijed prekoračenja kapaciteta međuspremnika ili uslijed maliciozne aktivnosti. Osim toga, strogo određeni memorijski resursi mogu se koristiti kao međuspremnici za I/O podatke te kronometre i brojače, to jest za provjeru kompletnosti i integriteta zapisanih podataka.

Važna napomena:

Glavna memorija (engl. *Main memory*) i priručna (privremena) memorija (engl. *Register memory*) nisu isto, niti se koriste na isti način. Iako nomenklatura ovisi o arhitekturi PLC-a i organizaciji memorijskih resursa koja se razlikuje za svakog pojedinog proizvođača, načelno se može reći da glavna memorija sadrži programski kod koji se trenutno izvršava te s njim povezane podatke. S druge strane, memorijski registri o kojima se u ovoj preporuci govori jesu priručna memorija. Iako se u praksi ova memorija najčešće koristi kao privremena memorija, ona ipak može sadržavati podatke od velike važnosti za izvršavanje programske logike.

Primjer

Primjer negativnog scenarija ukoliko navedena praksa nije implementirana:
(*Reference: G. P. H. Sandaruwan, P. S. Ranaweera, Vladimir A. Oleshchuk, PLC Security and Critical Infrastructure Protection):*

- Siemens obično za interne potrebe koristi memorijsko područje u rasponu od M250.0 do M255.7. Ukoliko se stanje nekog važnog podatka u ovom memorijskom području nekontrolirano promijeni, može doći do ozbiljnih pogrešaka u programskoj logici koja to memorijsko područje koristi za svoje potrebe.
- Za potrebe ovog primjera pretpostavit će se da je zlonamjeran pojedinac zarazio jedno od računala koje su u dohvatu ciljanog PLC uređaja. Računalo je zaraženo malicioznim programom s mogućnošću pisanja proizvoljnih vrijednosti u podatkovne registre PLC-a, u kojima je sadržana postavna vrijednost za regulaciju tlaka.
- Programska logika ne prepoznaje maliciozno podmetnutu postavnu vrijednost, što može dovesti postrojenje u nesigurno stanje, a što pak može dovesti do ozbiljnih posljedica po zdravlje i živote ljudi ili do velike materijalne štete.

Primjer implementacije ove preporuke:

- U nekom zamišljenom scenariju može se pretpostaviti da postoji zona s pojačanim zahtjevima za sigurnost. Instalacijom vatrozida može se spriječiti sve pokušaje pisanja u memorijsko područje koje se smatra važnim za sigurnost postrojenja.
- U nekom drugom scenariju, gdje već postroje registri strogo preddefinirani za čitanje ili za pisanje, oni trebaju biti organizirani u obliku matrice, jer to olakšava sigurnosna podešenja u PLC-u ili u vatrozidu.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Funkcionalno razdvajanje (čitanje/pisanje/provjera) olakšava zaštitu podataka. Također je olakšana konfiguracija i korištenje vatrozida s mogućnošću prepoznavanja protokola: pravila su jasnija i jednostavnija zbog precizno razdvojenih funkcija za svaki pojedini registar ili raspon registara. Olakšano je i održavanje vatrozida. Neautorizirane izmjene priručne memorije su lako iskoristiva ranjivost. Dosljednom provjerom svih varijabli koje ulaze u PLC ili koje iz njega izlaze, bilo kakva slučajna ili maliciozna promjena može biti jednostavno detektirana i spriječena u kompromitiranju programskog koda.
Pouzdanost	Ciklusi čitanja i pisanja postaju brži zbog smanjenog broja transakcija. Čak i autorizirane promjene i/ili programske pogreške mogu dovesti do kvara ukoliko se priručna memorija adekvatno ne zaštiti. Mrežne i komunikacijske greške također mogu dovesti do nenamjernih programskih pogrešaka ukoliko tako pristigli podaci nisu podvrgnuti provjeri.
Održavanje	Programske greške uzrokovane nepredviđenim ponašanjem priručne memorije su teške za analizu – teško je detektirati izvorište pogreške. Problem se može u potpunosti zaobići koristeći ovdje navedene preporuke.

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic** : TA009 - Inhibit Response Function, TA010 - Impair Process Control **Technique:** T0835 - Manipulate I/O image , T0836 - Modify Parameter
ISA 62443-3-3	**SR 3.4** : Software and information integrity **SR 3.5** : Input Validation **SR 3.6** : Deterministic Output
ISA 62443-4-1	**SD-4**: Secure design best practices **SI-1**: Security implementation review **SI-2** : Secure coding standards **SVV-1** : Security requirements testing
ISA 62443-4-2	**CR 3.4** : Software and information integrity **CR 3.5** : Input Validation **CR 3.6** : Deterministic Output
MITRE CWE	**CWE-787:** Out-of-bounds Write **CWE-653:** Insufficient Compartmentalization

11. Provjera plauzibilnosti

Instrumentacija treba biti odabrana tako da omogućuje provjeru plauzibilnosti, i to na način da se unakrsno provjeravaju različita mjerila.

Cilj smjernice	Ciljana skupina
Integritet I/O varijabli	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Postoje razni načini provjere fizičke plauzibilnosti s ciljem verifikacije izmjerenih vrijednosti:

a) Usporedba vremenski ovisnih i vremenski neovisnih varijabli
Provjera plauzibilnosti može se provesti tako da se vremenski neovisne varijable uspoređuju sa varijablama koje su nastale integracijom ili derivacijom vremenski ovisnih varijabli.

b) Usporedba mjerenja iz različitih izvora
Mjerenje iste fizikalne veličine na više različitih načina je vrlo dobra metoda za provjeru plauzibilnosti. Različiti izvori ne moraju uvijek implicirati različite senzore – ponekad je dovoljno da mjerena vrijednost na PLC dolazi alternativnim komunikacijskim kanalom (vidi primjer).

Primjer

a) Usporedba vremenski ovisnih i vremenski neovisnih varijabli

- Mjerilo protoka i mjerilo razine u spremniku: razina u spremniku mora odgovarati ukupno integriranom protoku koju u spremnik ulazi ili iz njega izlazi.
- Gorionik u kotlu: Količina utrošenog goriva (angažirane toplinske energije) mora odgovarati porastu temperature kotla.

b) Usporedba mjerenja iz različitih izvora

- Istovremenim mjerenjem brzine vjetra, nagiba umjetnog horizonta, vertikalne brzine i nadmorske visine može se pouzdano odrediti stupanj uspinjanja ili poniranja zrakoplova.
- Usporedba procesnih mjerenja čije su izvorište međusobno neovisni podatkovni *loggeri* čiji su senzori povezani uobičajenim strujnim mjernim petljama 4-20mA, ali čije se mjerene vrijednosti na PLC ili HMI prenose različitim komunikacijskim kanalima. Vrijednosti se uspoređuju i analiziraju, a bilo kakva diskrepancija treba generirati odgovarajuću poruku.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Olakšava praćenje eventualno manipuliranim mjernim podacima, uz pretpostavku da se ne manipulira svim senzorima istovremeno.
Pouzdanost	Olakšava identifikaciju i sprječava korištenje neispravnih mjernih vrijednosti.
Održavanje	Pomaže u isključenju fizičkog kvara kao uzroka greške mjerenja.

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0806 - Brute Force I/O
ISA 62443-3-3	**SR 3.5**: Input Validation **SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.5**: Input Validation **CR 3.6**: Deterministic Output
MITRE CWE	**CWE-754:** Improper Check for Unusual or Exceptional Conditions

12. Provjera fizikalnog smisla

Valja osigurati da operativno osoblje može unositi samo one procesne vrijednosti koje imaju fizikalni smisao i koje se uklapaju u trenutni kontekst procesa. Isto tako preporuča se mjeriti trajanje važnih procesa i procesnih koraka te alarmirati u slučaju odstupanja od predviđenih vrijednosti. Neobično miran ili potpuno neaktivan proces također je razlog za oprez.

Cilj smjernice	Ciljana skupina
Integritet I/O varijabli	Sistem integratori Pružatelji usluge održavanja

Obrazloženje

a) Mjerenje predviđenog trajanja

Ukoliko neka operacija, proces ili procesni korak traju duže nego je to predviđeno, to je dobar razlog za generirati poruku. Isto vrijedi i obratno – ukoliko je trajanje prekratko.

Eklatantan primjer za implementaciju ove preporuke je alarm predugog vremena trajanja koraka kod slijednog (sekvencijalnog) načina upravljanja. Primjerice, procesni korak čija je zadaća transport materijala s mjesta A na mjesto B u pravilu traje cca 5 sekundi. Ukoliko iz bilo kojeg razloga taj korak završi puno ranije ili puno kasnije od navedenog uobičajenog vremenskog perioda, valja generirati poruku o vremenskom prekoračenju.

b) Nadzor uobičajenih i/ili repetitivnih aktivnosti

Provjera fizičke plauzibilnosti ponekad se može odnositi na analizu redovitih repetitivnih događaja (ponavljanje šarži ili neki drugi predvidivi obrasci ponašanja). Preduga neaktivnost ovakvih događaja može ukazivati na problem ili pozadinsku malicioznu aktivnost.

Primjeri

a) Mjerenje predviđenog trajanja

- Pokretni element (zaklopka ili ventil s elektromotornim prigonom) s konačnim vremenom hoda između krajnjih pozicija.
- U postrojenju za obradu otpadnih voda, sabirno okno treba određeno vrijeme da se napuni.

b) Nadzor uobičajenih i/ili repetitivnih aktivnosti

- Šaržni proizvodni proces u obično ima pravilne cikluse i režime rada.
- Postrojenja za obradu otpadnih voda također imaju predvidive cikluse i obrasce ponašanja (primjerice, protok influenta).

c) Ograničenje unosa postavki postrojenja isključivo na fizikalno smislene vrijednosti.

- U poznatom slučaju Oldsmar Florida operateru je bilo omogućeno da unese postavnu vrijednosti 1000 puta veću od one koja je fizički uopće moguća u danim okolnostima. Granice za unos se mogu konfigurirati i na PLC i na HMI uređaju, a preporuka je da se provjera smislenosti svakako vrši barem na PLC uređaju.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	**1.** Odstupanje od predviđenog stanja može ukazivati na to da aktuator u startu nije bio u početnoj poziciji, ili da netko pokušava lažirati procesne podatke (I/O) koristeći reply tehniku napada. **2.** Neuobičajena neaktivnost određenih dijelova postrojenja ili zamrznute procesne vrijednosti mogu ukazivati na pozadinsku malicioznu aktivnost.
Pouzdanost	**1.** Odstupanja mogu biti indikator postojećeg ili skorog mehaničkog ili električkog kvara uređaja. **2.** Upozorenje o neaktivnosti može pomoći u dijagnosticiranju onih regulacijskih petlji čiji je pogrešan rad uzrokovan fizičkim kvarom, greškom u upravljačkom algoritmu ili neispravnim unosom od strane operatera.
Održavanje	

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0806 - Brute Force I/O
ISA 62443-3-3	**SR 3.5**: Input Validation **SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.5**: Input Validation **CR 3.6**: Deterministic Output
MITRE CWE	**CWE-754:** Improper Check for Unusual or Exceptional Conditions

13. Onemogućavanje nekorištenih komunikacijskih portova i protokola

Većina modernih PLC uređaja je opremljena sučeljima za korištenje višestrukih komunikacijskih protokola, od kojih je većina omogućena u zadanim tvorničkim postavkama. Svi komunikacijski portovi i protokoli koji nisu nužni za rad sustava trebaju biti onemogućeni.

Cilj smjernice	Ciljana skupina
Ojačavanje sustava	Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Komunikacijski protokoli na PLC uređaju obično su omogućeni prema zadanim (eng. *default*) tvorničkim postavkama (HTTP, HTTPS, SNMP, Telnet, FTP, Modbus, PROFIBUS, EtherNet/IP, ICMP i slični).

Najbolja (i ujedno preporučena) praksa jest izraditi detaljnu mrežnu topologiju i dijagram toka podataka između PLC-a i dugih komponenti sustava.

Takav dijagram pomaže da se identificiraju svi korišteni komunikacijski portovi, kao i sve mreže na koje su ti portovi priključeni. Za svaki fizički port valja izraditi listu korištenih mrežnih protokola, dok svi drugi protokoli trebaju biti onemogućeni.

Primjer

Većina PLC-ova ima ugrađen Web server za potrebe održavanja i/ili rješavanja problema. Međutim, ukoliko se ta mogućnost ne koristi, potrebno je onemogućiti Web server, jer se radi o potencijalnom vektoru napada.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Svaki omogućeni komunikacijski port ili protokol povećava napadnu površinu. Najjednostavniji način za rješavanje ovog problema jest da ih se onemogući.
Pouzdanost	Ukoliko je neki komunikacijski protokol onemogućen, to također znači da ne može niti proizvesti niti konzumirati mrežni promet, a to pak znači da je manja mogućnost kvara uslijed nenamjerno ili namjerno kompromitiranih mrežnih paketa.
Održavanje	Onemogućavanje nekorištenih protokola i portova ima pozitivan utjecaj i na održavanje, jer se smanjuje složenost sustava – ono čega nema ne kvari se, niti treba biti održavano (administrirano).

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK for ICS	**Tactic**: TA005 - Discovery **Technique:** T0808 - Control Device Identification , T0841 - Network Service Scanning, T0854 - Serial Connection Enumeration
ISA 62443-3-3	**SR 7.6**: Network and security configuration settings **SR 7.7**: Least functionality

Standard / okvir	Cjelina / poglavlje
ISA 62443-4-2	**EDR 2.13** : Use of physical diagnostic and test interfaces
ISA 62443-4-1	**SD-4**: Secure design best practices **SI-1**: Security implementation review **SVV-1**: Security requirements testing

14. Ograničavanje komunikacijskih sučelja prema trećim stranama

Način spajanja i podaci koji su raspoloživi sustavima trećih strana trebaju biti ograničeni i pod stalnim nadzorom. Komunikacijska i podatkovna sučelja prema trećim stranama moraju biti jasno definirana i ograničena samo na podatkovni promet koji je potreban za normalno funkcioniranje sustava.

Cilj smjernice	Ciljana skupina
Ojačavanje sustava	Sistem integratori Pružatelji usluge održavanja

Obrazloženje

U određenim slučajevima, primjerice kod velikih udaljenosti ili kod velikog volumena izmijenjenih podataka, opravdano je podatke između dva sustava izmjenjivati putem nekog standardnog komunikacijskog protokola, kao alternativa klasičnom (ali i sigurnom) povezivanju žicama.

Kod takve izmjene podataka valja voditi računa o sljedećim pravilima struke i dobrim praksama:

- Koristiti odvojen, samo tome namijenjen komunikacijski modul izravno spojen na PLC uređaj treće strane, ili na neku drugu komunikacijsku komponentu. Alternativno koristiti mrežnu opremu fizički odvojenu od osnovne procesne mreže (s obje strane).
- Koristeći fizičku (MAC) adresu međusobno spojenih uređaja valja realizirati multifaktorsku identifikaciju pouzdanih uređaja (ispravna IP adresa + ispravna MAC adresa = pouzdan uređaj). Iako se ne radi o nepremostivoj prepreci (i IP i MAC adresa mogu se lažirati različitim tehnikama), ali svakako podiže ljestvicu vještina koju potencijalni napadač mora posjedovati kako bi zaobišao ovo ograničenje.
- Kod odabira komunikacijskog protokola prednost treba dati onim protokolima koji minimiziraju mogućnost neovlaštenog (neautoriziranog) pisanja ili čitanja.
- Konekcija i komunikacijski port trebaju biti odabrani (konfigurirani) tako da onemogućavaju uređaje trećih strana da na bilo koji način utječu na konfiguraciju PLC uređaja koji se štiti.
- Uređajima trećih strana treba u potpunosti onemogućiti čitanje i pisanje podataka koji se nalaze izvan raspona dogovorenog za komunikaciju.
- Ugraditi mehanizam za kontrolu komunikacije (eng. *Watchdog)* kako se podaci ne bi izmjenjivali u režimima kada to može biti opasno, ili kada to nije potrebno.
- Serijska komunikacija: koristiti odvojen komunikacijski modul za svako sučelje prema trećim stranama, a podaci neka budu složeni u obliku matrice. Dobra je praksa da komunikaciju uvijek inicira vlasnički (glavni, veliki) PLC, dok je PLC treće strane pasivan.
- Ethernet/IP: U nekim PLC konfiguracijama komunikacijski modul može vršiti funkciju vatrozida i dubinske inspekcije paketa, ili se može podesiti da na neki drugi način ograničava izmjenu podataka. Ukoliko ovakva mogućnost postoji, a Ethernet/IP se koristi za komunikaciju, treba ju omogućiti, konfigurirati i koristiti.
- Kada pogonski ili administrativni (ugovorni) uvjeti onemogućavaju korištenje gore navedenih dobrih praksi, valja razmotriti korištenje koncentratora podataka (posredničkog/DMZ PLC-a) kao barijere koja sprječava neželjeno ili neovlašteno čitanje i/ili pisanje podataka.

Primjeri

- Cjevovod koji vrši transport ugljikovodika ili nekog drugog tekućeg materijala. Dodirne točke između dvije razine transporta ili između dvije različite tvrtke u sustavu su uobičajen slučaj izmjene podataka putem mrežnih ili serijskih komunikacijskih protokola. Obično se izmjenjuju podaci o izmjerenim vrijednostima te važnim pogonskim stanjima (dozvole i zaštite).
- Regionalni uvoznik i distributer vode koji dijeli informacije o trenutnom protoku prema svojim kupcima.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	1. Ograničavanje izloženosti sustavima trećih strana. 2. Autentifikacija vanjskih uređaja s ciljem sprječavanja lažiranja procesnih podataka.
Pouzdanost	Ograničava mogućnost namjerne ili nenamjerne modifikacije procesa uslijed pogreške u komunikaciji prema sustavima trećih strana.
Održavanje	

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic**: TA010 - Impair Process Control **Technique:** T0836 - Modify Parameter
ISA 62443-3-3	**SR 7.6**: Network and security configuration settings **SR 7.7**: Least functionality
ISA 62443-4-2	**CR 7.6**: Network and security configuration settings **CR 7.7**: Least functionality
ISA 62443-4-1	**SD-4**: Secure design best practices **SI-1**: Security implementation review **SVV-1:** Security requirements testing

15. Definiranje sigurnog procesnog stanja kod pokretanja PLC uređaja

Jasno definirati sigurno stanje procesa kod pokretanja PLC uređaja (npr. aktivacija aktuatora, deaktivacija aktuatora, zadržavanje zadnjeg zapamćenog stanja…)

Cilj smjernice	Ciljana skupina
Pouzdanost sustava	Proizvođači opreme Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Ukoliko se iz bilo kojeg razloga dogodi da se PLC ponovno pokrene u sred procesa koji kontrolira, od upravljačkog programa očekujemo da taj postupak protekne što je moguće mirnije i s minimalnim utjecajem na proces. Svakako valja voditi računa o tome da je proces otporan na ovakve događaje.

Ukoliko gore navedenu preporuku nije moguće implementirati u praksi, to valja jasno naglasiti, uz čvrstu preporuku da se za vrijeme procedure ponovnog pokretanja ne izdaju nikakve nove komande. Isto tako u tom slučaju treba postojati SOP (Standardni Operativni Postupak) s jasnim uputstvima kako proces pokrenuti ručno (manipulacijama operatera).

Sve procedure koje su tranzijentne (pokretanje, zaustavljanje, ponovno pokretanje, ponovno pokretanje „u letu“…) moraju biti detaljno dokumentirane.

Primjer

/

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	**Eliminacija mogućnosti nepredviđenog ponašanja:** Najosnovniji zamislivi napad na PLC uređaj započinje forsiranjem STOP režima, nakon čega slijedi procedura ponovnog pokretanja ili RESTART. Valja naglasiti da se radi o relativno jednostavnom napadu, jer većina PLC-ova nema mehanizme da se zaštiti od nepredviđenih podataka ili da se nosi sa prekomjernim mrežnim prometom. I dok je normalan (RUN) režim rada u pravilu poznat i dobro definiran, procedure ponovnog pokretanja obično nisu niti dobro definirane niti su dokumentirane. Ovo jest neobično, ali radi se o najosnovnijem napadnom vektoru iz perspektive zlonamjernog pojedinca**.**
Pouzdanost	**Izbjegavanje neželjenih zastoja:** Ukoliko nakon procedure ponovnog pokretanja operativno osoblje nije u mogućnosti dovesti postrojenje u normalan režim rada, to će raditi tehničar ili netko drugi iz osoblja održavanja, i to na način da analizom PLC programskog kod pokuša dovesti postrojenje u željeno stanje. Ovo može prouzročiti značajna kašnjenja u proizvodnji.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic**: TA009 - Inhibit Response Function **Technique:** T0816 - Device Restart/Shutdown
ISA 62443-3-3	**SR 3.6**: Deterministic Output
ISA 62443-4-2	**CR 3.6**: Deterministic Output
ISA 62443-4-1	**SVV-1**: Security requirements testing

16. Nadzor trajanja izvršavanja PLC koda

Preporuča se periodički (svake 2-3 sekunde) izmjeriti vrijeme trajanja izvršavanja programskog koda, a izmjerenu vrijednost poslati na neki od HMI uređaja za potrebe vizualizacije na dijagramu.

Cilj smjernice	Ciljana skupina
Nadzor sustava	Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Ukupno vrijeme ciklusa izvršavanja programskog koda je lako dohvatljiva sistemska varijabla koja se jednostavno može koristiti u programskoj logici. Programsku logiku treba realizirati tako da se računa prosječna vrijednost te minimalno i maksimalno vrijeme izvršavanja. Za vizualizaciju ovih podataka valja koristiti HMI uređaje, na kojima isto tako treba generirati poruku ukoliko dođe do značajnijeg odstupanja od predviđenih vrijednosti.

Vrijeme ciklusa po definiciji je ukupno vrijeme koje je potrebno da se izvrši jedna iteracija ukupnog korisničkog programskog koda, zajedno sa svim popratnim aktivnostima PLC-a (čitanje i pisanje I/O varijabli, obrada komunikacije i slično). Korisnički programski kod je kombinacija ljestvičastih (engl. *Ladder*) dijagrama, FB (engl. *Function Block*) dijagrama, liste instrukcija (engl. *Instruction List*) i strukturiranih naredbi (engl. Structure Text). Ponekad se nabrojani programski jezici kombiniraju u neki viši oblik, poput SFC (eng. *Sequential Function Chart*) dijagrama.

U normalnim okolnostima vrijeme ciklusa izvršavanja programskog koda je konstantno. Događaji koji mogu utjecati na značajnije promjene u vremenu izvršavanja jesu.

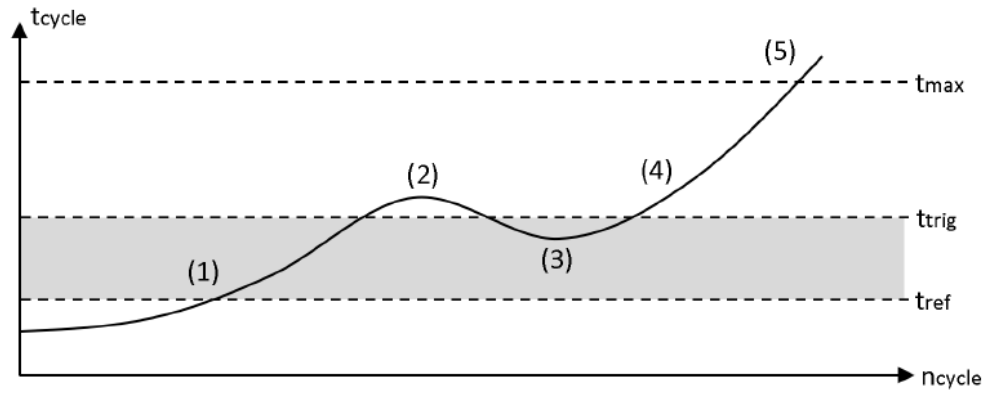
- Promijenjeni uvjeti u mrežnom okruženju
- Značajnije promjene u programskoj logici
- Određeni događaji u procesu kojim se upravlja

Neuobičajeno dugo (ili neuobičajeno kratko) vrijeme ciklusa izvršavanja programskog koda može biti indikator da se programska logika promijenila, što je indikacija za provjeru integriteta.

Vizualizacija ovih vrijednosti na HMI dijagramu pomaže da se na intuitivan način detektiraju anomalije, a koje možda ne bi bile lako vidljive kad bi se analizirale samo sirove, apsolutne brojke.

Primjer

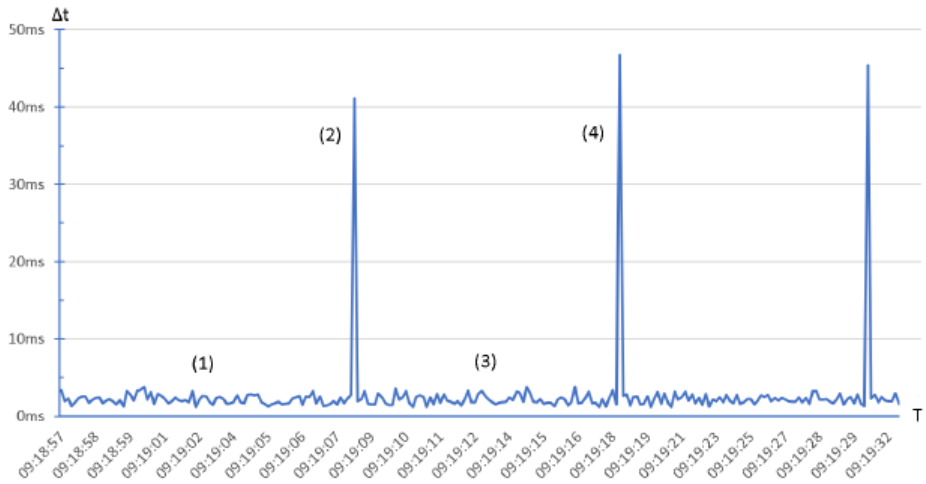
Većina PLC uređaja ima ugrađenu funkciju nadzora maksimalnog vremena izvršavanja programskog koda (engl. *Watchdog*), i to na razini hardvera. Prekoračenje zadanog maksimalnog vremena postavlja PLC u STOP režim (5)
Međutim, potencijalni napadači su ovoga svjesni, pa će napad pokušati izvesti tako da utjecaj na ciklus izvršavanja bude što je moguće manji. U odvojenoj funkciji čija je zadaća nadzor ciklusa izvršavanja definiran je referentni ciklus izvršavanja tref kao osnova za daljnju analizu. S obzirom na to da su određene fluktuacije u vremenu izvršavanja normalne, potrebno je definirati pragove koji omeđuju normalan režim rada (1, 3). Prekoračenje praga t_{trig} je poziv na oprez i jasan indikator da se u pozadini događa nešto neobično (2, 4).



Ukoliko (i kada) se dogode, odstupanja se može bilježiti u tablicu:

	Datum	UTC vrijeme	Vrijeme ciklusa
1	22.11.2019.	09:05:50.021	40.821 ms
2	22.11.2019.	09:06:01.069	40.391 ms
3	22.11.2019.	09:06:10.120	40.994 ms
4	22.11.2019.	09:06:20.166	40.561 ms
5	22.11.2019.	09:06:30.211	40.725 ms

Vizualizacija na HMI uređajima u obliku vremenskog dijagrama omogućuje gotovo trenutačnu detekciju visokog opterećenja CPU-a. Sljedeća slika prikazuje vremenski dijagram ciklusa izvršavanja PLC uređaja koji osmim korisničkog periodički izvršava i maliciozni programski kod. Segmenti dijagrama označeni brojkama 1 i 3 prikazuju normalne fluktuacije u vremenu izvršavanja, dok segmenti 2 i 4 prikazuju trenutke kada se izvršava napadačev kod, koji značajno i vidljivo povećava vrijeme ciklusa.



Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Napad na PLC može biti realiziran tako da se promijeni programska logika, što može značiti izvršavanje potpuno novih dijelova programskog koda, nove procesne recepte ili aktivaciju nekih drugih značajki koji za krajnji cilj imaju realizaciju napadačevih ciljeva. Na većini današnjih PLC-ova integritet programskog koda nije moguće provjeravati koristeći kriptografske funkcije. S obzirom da je vrijeme ciklusa u normalnom okolnostima više-manje konstantno, njegov nadzor valja shvatiti kao posrednu metodu detekcije neželjenih promjena u programskog kodu.
Pouzdanost	Isto kao i za sigurnost, ali za slučajeve kada promjene nisu malicioznog porijekla.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic**: TA002 - Execution **Technique:** T0873 - Project File Infection
ISA 62443-3-3	**SR 3.4**: Software and information integrity
ISA 62443-4-2	**EDR 3.2**: Protection from malicious code
MITRE CWE	**CWE-754:** Improper Check for Unusual or Exceptional Conditions

17. Nadzor vremena neprekidnog rada

Vrijeme neprekidnog rada valja pratiti kako bi se znala točna vremena i ukupan broj zastoja. Izmjerene vrijednosti treba bilježiti na HMI uređajima.

Cilj smjernice	Ciljana skupina
Nadzor sustava	Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Vrijeme neprekidnog rada može se mjeriti na više načina:

- Na samom PLC uređaju (ukoliko je vrijeme neprekidnog rada raspoloživa sistemska varijabla)
- Na samom PLC uređaju (ukoliko ima MIB-2 ili neku drugu vrstu SNPM implementacije)
- Eksterno, pomoću SNMP uređaja/protokola

Ukoliko PLC ima implementiran SNMP sa MIB-2, što je čest slučaj, OID identifikator za „sysUpTimeInstance(0)“ je 1.3.6.1.2.1.1.3. Reset ove varijable je važan indikator da se PLC možda ponovno pokrenuo (RESTART). HMI bi trebao generirati poruku na svaku sumnju da se PLC isključio i/ili ponovno pokrenuo.

Kombinacija vremena neprekidnog rada i eventualnih kodova grešaka su vrlo dobar izvor dijagnostičkih informacija.

Primjer

/

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Najosnovniji zamislivi napad na PLC uređaj započinje forsiranjem STOP režima, nakon čega slijedi procedura ponovnog pokretanja ili RESTART. Valja naglasiti da se radi o relativno jednostavnom napadu, jer većina PLC-ova nema mehanizme da se zaštiti od nepredviđenih podataka ili da se nosi sa prekomjernim mrežnim prometom. Bilo koji izravni ili neizravni indikator da se PLC ponovno pokrenuo jest poziv na oprez i znak da se u pozadini događa nešto neobično.
Pouzdanost	Zabilježeni događaji zastoja i ponovnog pokretanja mogu biti korisni u lociranju i analizi kvarova u postrojenju, ali i za nadzor inženjerskih aktivnosti na PLC-ovima.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic**: TA009 - Inhibit Response Function **Technique:** T0816 - Device Restart/Shutdown
ISA 62443-3-3	**SR 7.6**: Network and security configuration settings
ISA 62443-4-2	**CR 7.6**: Network and security configuration settings
MITRE CWE	**CWE-778:** Insufficient Logging

18. Zapisivanje uzroka zastoja

Događaje koji su prouzrokovali zastoj PLC-a valja zabilježiti i prikazati na HMI sustavu za prikaz poruka, kako bi se mogli analizirati prije nego se pokuša ponovno pokrenuti PLC. Za što kvalitetniju analizu preporuča se vremenski sinkronizirati sve uređaje.

Cilj smjernice	Ciljana skupina
Nadzor sustava	Sistem integratori Pružatelji usluge održavanja

Obrazloženje

Događaji koji su prethodili nekom incidentu (zaustavljanju ili ponovnom pokretanju PLC uređaja) daju važne informacije o tome zašto se taj događaj desio, te ukazuju ono što treba provjeriti/popraviti prije nego se pokuša ponovno pokrenuti PLC.

Neki PLC-ovi imaju mogućnost pamćenja oznake greške koja je prouzrokovala prethodni kvar. Takve unose valja trajno zabilježiti i tek potom obrisati iz memorije PLC-a. Također se preporuča te događaje zabilježiti i prikazati na HMI uređajima ili čak na Syslog poslužitelju, ukoliko takva mogućnost postoji.

Isto tako, većina (ili svi) PLC-ovi na neki način dojavljuju trenutak kada se izvršava prvi ciklus nakon zastoja (engl. *First Scan*). U pravilu se radi o zastavici ili o rutini koja se izvršava samo u prvom ciklusu kada se PLC pokrene. Taj signal također treba zabilježiti i zapisati na HMI uređaj.

Primjer

/

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Povijesni zapis ovakvih događaja olakšava analizu potencijalnog incidenta i otklanjanje kvara koji ga je prouzrokovao. Detaljan zapis slijeda kritičnih događaja omogućuje kvalitetnu procjenu pouzdanosti PLC-a nakon incidenta.
Pouzdanost	Ovakav zapis pomaže i u otklanjanju programskih pogrešaka, ukoliko incident nije prouzrokovan malicioznom aktivnošću.
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic**: TA009 - Inhibit Response Function **Technique:** T0816 - Device Restart/Shutdown 1
ISA 62443-3-3	**SR 7.6**: Network and security configuration settings
ISA 62443-4-2	**CR 7.6**: Network and security configuration settings
MITRE CWE	**CWE-778:** Insufficient Logging

19. Nadzor angažmana memorije

Preporuča se za svaki korišteni kontroler mjeriti angažman raspoloživih memorijskih resursa. Izmjerene vrijednosti valja zabilježiti i prikazati na HMI uređajima.

Cilj smjernice	Ciljana skupina
Nadzor sustava	Sistem integratori Pružatelji usluge održavanja Vlasnici opreme

Obrazloženje

Povećanje broja linija programskog koda u pravilu prati i veći angažman memorijskih resursa, bilo kod programiranja, bilo kod izvršavanja. Dobra je programerska praksa kontinuirano pratiti angažirane memorijske resurse te izvještavati operativno osoblje u slučaju značajnijih odstupanja od predviđenih vrijednosti.

Primjer

Na nekim PLC uređajima tipa Rockwell Automation Allen Bradley može se definirati predviđeno zauzeće memorijskih resursa, te naknadno, pomoću ugrađene funkcije RSLogix 5000 Task Monitoring Tool, pratiti odstupanje od te vrijednosti. Osim glavne memorije, koristeći ovu funkciju moguće je pratiti i neke druge važne pokazatelje.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Značajno povećanje angažmana memorijskih resursa može biti znak da se na PLC-u izvršava programski kod koji je izmijenjen u odnosu na originalni.
Pouzdanost	Kontinuirani nadzor memorijskih resursa pomaže da ne dođe do situacije da su iskorišteni svi raspoloživi memorijski kapaciteti, što može biti uzrok mnogim problemima.
Održavanje	Kontinuirani nadzor memorijskih resursa pomaže kod podešavanja postrojenja, te može pomoći da se ustanove optimalni parametri s obzirom na angažman memorijskih resursa i vrijeme ciklusa. Ove informacije mogu biti od koristi i kod analize i otklanjanja problema.

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic**: TA002 - Execution **Technique:** T0873 - Project File Infection
ISA 62443-3-3	**SR 3.4**: Software and information integrity
ISA 62443-4-2	**EDR 3.2**: Protection from malicious code

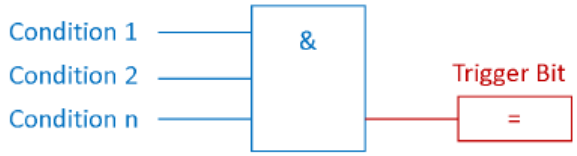
20. Nadzor lažno pozitivnih i lažno negativnih kritičnih alarma

Nakon identifikacije kritičnih alarma valja implementirati logiku za detekciju lažnih uvjeta koji ih okidaju. Svaki takav slučaj mora generirati poruku.

Cilj smjernice	Ciljana skupina
Nadzor sustava	Sistem integratori Pružatelji usluge održavanja

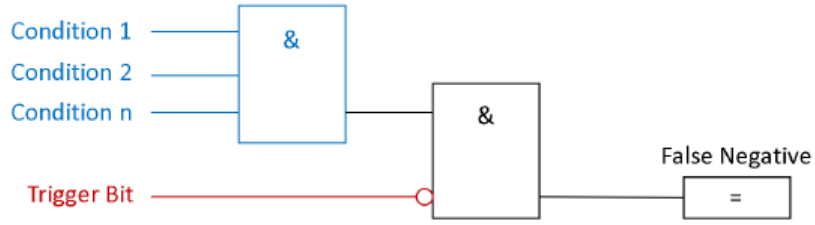
Obrazloženje

U velikoj većini slučajeva alarmni signali su binarni (eng. *True*/*False*), a uvjeti koji ih okidaju slični su onima prikazanima na slici ispod. U navedenom slučaju bit koji okida alarm previsokog tlaka bit će u stanju logičke jedinice samo ako su zadovoljeni svi uvjeti u logici koja mu prethodi.



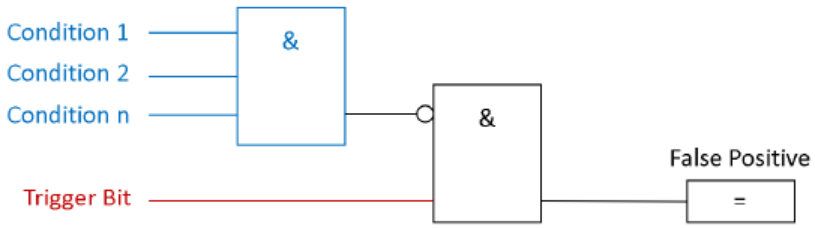
Kako bi prikrio napad, zlonamjeran pojedinac može pokušati potisnuti ovaj alarm, generirajući na taj način lažno negativan okidač.

Logika za detekciju lažno negativnih okidača je prikazana na slici ispod: binarna logika prati sve uvjete koji okidaju alarm, a logički rezultat ove operacije se uspoređuje sa invertiranim okidačem alarm. Na taj se način jednostavno kreira varijabla koja označava lažno negativan okidač:



U nekom drugom slučaju napadač može namjerno generirati lažno pozitivan alarm, s ciljem skretanja pažnje operatera.

Na sličan način kako se može detektirati lažno negativan okidač može se detektirati i lažno pozitivan. Binarna logika ponovno prati uvjete koji okidaju alarm (ovaj put s invertiranim rezultatom) te ih logički uspoređuje sa stanjem okidača alarma. Ukoliko nema uvjeta za generiranje alarma, a okidač alarma je svejedno u stanju logičke jedinice, detektiran je lažno pozitivan okidač:



Primjer

Primjer 1: Kontroleri iz serije Siemens S71200/1500 imaju ugrađen web poslužitelj s vrlo širokom paletom funkcija. Primjerice, pomoću njega moguć je detaljan prikaz stanja PLC-a, monitoring ciklusa izvršavanja i slično. Između ostalog, tu je i funkcija nadzora i modifikacije PLC varijabli. Kontrola pristupa ovim varijablama i razina korisničkih prava podešavaju se u postavkama samog Web poslužitelja. Ukoliko razine prava nisu na odgovarajući način podešene, potencijalnom je napadaču omogućen pristup PLC varijablama, a samim time i generiranje lažno pozitivnih i/ili lažno negativnih alarma.

Primjer 2: U slučaju Triton/Trisys/Hatman napada potiskivanje alarma je sastavni dio malicioznog koda.

Primjer 3: U slučaju napada injekcijom podatkovnog prometa moguće je generirati lažne alarme na HMI i SCADA uređajima.

Zašto?

Što je poboljšano?	Kako je poboljšano?
Sigurnost	Smanjena mogućnost prikrivanja napada u okviru kojeg napadač potiskuje alarme ili generira lažne s ciljem skretanja pozornosti.
Pouzdanost	/
Održavanje	/

Reference

Standard / okvir	Cjelina / poglavlje
MITRE ATT&CK ICS	**Tactic** : TA009 - Inhibit Response Function **Technique:** T0878 - Alarm Suppression
ISA 62443-3-3	**SR 3.5** : Input Validation
ISA 62443-4-2	**CR 3.5** : Input Validation
ISA 62443-4-1	**SI-1** : Security implementation review
MITRE CWE	**CWE-754:** Improper Check for Unusual or Exceptional Conditions

O projektu smjernica za pisanje sigurnog PLC koda

Godinama su programibilni logički kontroleri bili kibernetički nesigurni, a ta nesigurnost je proizlazila iz njihovog dizajna. U posljednjih nekoliko godina u području industrijskih računalnih sustava uhodale su se neke dobre prakse iz svijeta kibernetičke sigurnosti „normalnih“ računalnih sustava, kao što je korištenje sigurnih protokola, enkripcija podataka, segmentacija mreže i slično. Kako god, sve do pojave ovog dokumenta nije bilo sustavnog pokušaja da se fokus sigurnost prebaci na korištenje ugrađenih funkcija PLC-a (ili SCADA-e), ili da se na jasan i jednostavan način opiše kako se može pisati kibernetički siguran (sigurniji) PLC kod. Ovaj projekt – inspiriran sličnim dokumentom koji postoji za tradicionalne IT sustave – pokušava ispuniti tu zadaću.

Kome je ovaj dokument namijenjen?

Ovaj dokument prvenstveno je namijenjen inženjerima za automatizaciju. Cilj projekta je pružiti smjernice za pisanje programskog koda (ljestvičasta logika, funkcijski nacrti i slično) koji može unaprijediti sigurnosne karakteristike samog PLC-a ili sustava u cjelini. Navedene smjernice oslanjaju se prvenstveno na funkcionalnosti koje su nativno raspoložive u većini PLC uređaja, stoga za njihovu implementaciju nije potreban (ili je minimalno potreban) angažman vanjskih hardverskih ili softverskih alata. Sve predložene prakse mogu se relativno jednostavno uklopiti u uobičajen način programiranja/konfiguracije industrijskih računalnih sustava. Isto tako, za njihovu primjenu nije potrebno opširno znanje kibernetičke sigurnosti PLC sustava – važnije je dobro poznavati angažirane PLC i HMI uređaje te proces kojim se upravlja.

Opseg dokumenta

Ovaj dokument sadrži 20 smjernica za kibernetički sigurno programiranje PLC-a, a koje su odabrane iz puno većeg skupa preporuka koje se tiču kibernetičke sigurnosti industrijskih računalnih sustava. U okviru tog skupa postoji još niz preporuka koje nisu odabrane u 20 najvažnijih, ili onih koje su u fazi nacrta, a osim na PLC mogu se odnositi na dokumentaciju ili arhitekturu sustava. Takve smjernice ne spadaju u opseg ovog dokumenta, ali bi se mogle naći na nekoj budućoj listi koja se tiče cjelokupnog ekosustava (okoline) u kojem se predmetni PLC nalazi.

Koji su benefiti primijene ovih smjernica?

Benefiti primijene ovih smjernica su očigledni – bilo u tome da se značajno smanjuje napadna površina, bilo u tome da se olakšava forenzika u slučaju da do incidenta zaista i dođe. Osim na sigurnost, primjena ovih smjernica pozitivno djeluje i na niz drugih aspekata: većina njih čini programski kod pouzdanijim, robusnijim i lakšim za održavanje. Implementacija navedenih smjernica pozitivno djeluje na robusnost sustava čak i kada uzrok problema nije malicioznog karaktera, to jest kada se radi o programskoj pogrešci ili pogrešci u manipulaciji sustavom.

Tko stoji iza ovog projekta?

Ideja je stvorena nakon što je g. Jake Brodsky na S4x20 konferenciji održao poznato predavanje pod naslovom “Secure Coding Practices for PLC’s”.

Projekt je iniciran od strane g. Dale Petersona nakon te konferencije. Jake Brodsky i Sarah Fluchs proveli su nekoliko sati u telefonskom razgovoru kako bi predložene smjernice prenijeli na papir. Nakon toga su Jake, Sarah i Dale uspostavili platformu top20.isa.org (podržanu od strane ISA-e) s ciljem okupljanja svih zainteresiranih pojedinaca koji svojim iskustvom i znanjem mogu pomoći u ostvarenju ovog projekta.

Prikupljanje podataka i ideja, rasprave i kreiranje liste trajali su oko godinu dana. Proces je bio ubrzan kada se projektu pridružio Vivek Ponnada, koji je osim sudjelovanja u realizaciji projekta organizirao redovite konferencijske pozive na kojima su se koordinirale aktivnosti i usuglašavala stajališta. Mohamed Abelomez Sakesli je uložio puno truda u reference na sve primjenjive standarde, a MITRE CWE tim je u zadnji trenutak prije objavljivanja kreirao CWE reference. Sarah je kreirala završnu verziju dokumenta koji upravo čitate, a osim ranije navedenih pojedinaca svojim vrlo vrijednim doprinosima su se istakli John Cusimano, Dirk Rotermund, Josh Ruff, Thomas Rabenstein, Gus Serino, Walter Speth, Agustin Valencia Gil-Ortega, Marcel Rick-Cen i Al Ratheesh R.

Popis osoba koje su podržale projekt

Projekt izrade smjernica za pisanje sigurnog PLC kora je pravi primjer projekta zajednice, koji ne bi bio moguć bez doprinosa pojedinaca koji su nesebično stavili na raspolaganje svoje vrijeme, znanje i iskustvo. Ukupno su registrirana 943 korisnika koji su svojim komentiranjem ili glasovanjem pomogli da se ovaj projekt uspješno okonča. U nastavku se nalazi abecedni popis ljudi koji su eksplicitno pristali da budu navedeni u dokumentu. Još jednom hvala svima koji su pomogli u ostvarenju ovog projekta!

Aagam Shah
Adam Paturej
Agustin Valencia Gil-Ortega
Aitor García Almiñana
Alec Summers
Al Ratheesh. R
Andreas Falk
Anton Shipulin
Arkaitz Gamino
Carlos Olave
Chris van den Hooven
Chris Sistrunk
Christos Alexopoulos
Cris DeWitt
Dale Peterson
Dene Yandle
Dennis Verschoor
Dirk Rotermund
Edorta Echave García
Gananand Kini
George Alex Holburn
Gus Serino
Hakija Agic
Josie Houghton
Jozef Sulwinski
Juan Pablo Angel Espejo
Khalid Ansari
Marc Weber
Marcel Rick-Cen
Martin Huddleston
Massimiliano Zonta
Matthew Loong
Matthias Müller
Michael Thompson
Michal Stepien
Miguel Angel Frias
Mohamed Abdelmoez Sakesli
Moon Eluvangal Chandran
Nahuel Iglesias
Nalini Kanth
Narasimha S. Himakuntala
Omar Morando
Oscar J. Delgado-Melo
Päivi Brunou
Peter Donnelly
Peter Jackson

Hector Medrano
Heiko Rudolph
Isiah Jones
Jacob Brodsky
Javier Perez Quezada
J-D Bamford
Joe Weiss
John Cusimano
John Hoyt
John Powell
John Kingsley
Joseph J. Januszewski
Josh Ruff
Ravindra Deshakulakarni
Rick Booij
Robert Albach
Rushi Purohit
Sarah Fluchs
Sergei Biberdorf
Stephan Beirer
Steve Christey Coley
Thomas Rabenstein
Tim Gale
Vivek Ponnada
Vytautas Butrimas
Walter Speth

Veliko hvala dolje navedenim organizacijama koje u velikodušno osigurale neophodnu infrastrukturu za provedbu i održavanje projekta (domena, web dizajn, hosting, grafički dizajn…):

Prijevod na hrvatski jezik: Kruno Jurlina

ATO Inženjering d.o.o. Osijek

Hrvatski institut za kibernetičku sigurnost - HIKS