Nyílt forráskódú behatolásjelző rendszerek vizsgálata ipari *vezérlőrendszerekben*

Szabados József Óbudai Egyetem · ipari rendszerek hálózati detekciója

Tegye fel a kezét, aki...

01 **...dolgozott már szerverrel, tűzfallal, SIEM-mel.**

02 **...látott már élőben PLC-t vagy vezérlőszekrényt.**

Ha elszáll, mi történik?

IT

Újraindítod.

A TÉT
adat és információ

PRIORITÁS
C → I → A

OT

Nyitva marad egy szelep.

A TÉT
fizikai következmény

PRIORITÁS
A → I → C (+ safety)

CYBERSEC

Ki kicsoda egy OT-rendszerben?

Három réteg — és egy közös vakfolt.

HMI

AZ ARC / A KÉPERNYŐ

amit a kezelő néz

SCADA

A DISZPÉCSERKÖZPONT

felügyel és összefog

PLC

AZ IZOM

ténylegesen kapcsol

Alattuk: érzékelők és beavatkozók — itt, a fizikai világban ér földet a támadás.

Mi az a PLC?

Egy strapabíró, egyfeladatos ipari számítógép.

01

Érzékel

hőmérséklet, nyomás, szint

→

02

Dönt

egyszerű logika, ciklusban

→

03

Beavatkozik

szelep, motor, szivattyú

↻ **néhány ms-onként** · **20–30 évig** · **vírusirtó nélkül**

„Vak” protokollok

Modbus

1979 óta — idősebb, mint a legtöbbünk

S7comm

Siemens PLC-k · 102/TCP

- ✕ Hitelesítés
- ✕ Titkosítás
- ✕ „Ki vagy?”

Hány sor kód kell egy PLC leállításához?

≈ 3

sor Python.

Nincs exploit. Nincs zero-day. Ez a protokoll rendeltetésszerű működése.

Jelszóvédelem nélkül konfigurált, régebbi S7 PLC esetén, nyílt forráskódú könyvtárral.

Három sor. Nézzük meg.

attacker.py — AI-generált, „monitoring toolnak” álcázva

```
import snap7

plc = snap7.client.Client()

plc.connect("192.168.10.20", 0, 1)  # nincs jelszó

plc.db_write(1, 0, bytearray([0x00]))  # egy írás. Kész.
```

- ✓ **snap7**
  nyílt könyvtár, mérnökök ezrei használják legitim célra
- ✓ **Nincs jelszó**
  a PLC nem kérdez — engedelmeskedik
- ✓ **Nincs exploit**
  a protokoll pontosan így működik

Ugyanezt a snap7-mintát nevezi meg a **CISA AA26-231A** riasztás.

2010-ben egy állam kellett hozzá. *Ma egy laptop.*

2010
Stuxnet
Irán · Siemens S7 PLC-k

2015
Ukrajna
áramszünet több százezernél

2017
Triton
a biztonsági vezérlőt (SIS)

2023
Unitronics
kitett PLC-k vízművekben

2024
FrostyGoop
Lviv: fűtés le, Modbus TCP

2025
Norvég gát
távolról nyitott szelep

2026
CISA AA26-231A
tömeges S7comm-szkennelés

Nem elmélet. Aktív fenyegetés.

AA26-231A · 2026.08.19

> *„This is not a theoretical risk – it is an active threat."*
>
> **— NSA · CISA · FBI · DOE · EPA**

Az AI írja a kódot

lezuhant a belépési küszöb — nem kell hozzá évek tudása

T1588.007

snap7 · S7comm · :102

olvasás és írás közvetlenül a PLC memóriájába

Pre-positioning

az írás most még előkészítés a későbbi fizikai hatáshoz

A CISA ICS-tudatos IDS-t ír elő — a példái kereskedelmiek (Claroty, Dragos, Nozomi). **Az én keretrendszerem: ugyanez, nyílt forrásból.**

Forrás: NSA/CISA/FBI/DOE/EPA — Joint Cybersecurity Advisory AA26-231A, 2026-08-19 · TLP:CLEAR

Miért nem védjük meg őket egyszerűen?

01

30 év életciklus

Egy PLC túlél hat laptopgenerációt. A 2000-es évek eszközei ma is termelnek.

02

Nem patchelhető

Frissítés = leállás = pénz és kockázat. A karbantartási ablak évente egyszer jön.

03

Az IT-eszköz vak

A tűzfal látja: port 502. Azt nem látja: „írj a szivattyú alapjelébe”.

...és az „air gap” ma már többnyire mítosz.

Meg lehet ezt csinálni 0 Ft licenccel?

1. **Csak nyílt forráskód**
2. **Protokollt ért, nem csak portot**
3. **Minden riasztás → MITRE ATT&CK for ICS**

Felépíthető-e kizárólag nyílt forráskódú komponensekből olyan ICS/OT detekciós lánc, amely érti a protokollszintű szemantikát, és amelynek riasztásai visszavezethetők verifikált MITRE ATT&CK for ICS technikákra?

A labor: öt nyílt forráskódú építőkocka

01
Zeek + Suricata
szenzor, DPI

→

02
Filebeat
továbbítás

→

03
Wazuh
elemzés

→

04
Elastic-search
tárolás, EQL

→

05
Kibana
SOC-nézet

4 VM · Ubuntu 22.04

OpenPLC célpont

Passzív SPAN-port

Licenc: 0 Ft

CYBERSEC

A támadás három lépésben

1 · Felderítés

`READ_HOLDING_REGISTERS`

kiolvassa a regisztereket — még ártatlan lehet

→

2 · Manipuláció

`WRITE_SINGLE_REGISTER`

ugyanonnan, ≤ 10 s — átírja a szivattyú alapjelét

→

3 · Riasztás

`T0836 · Modify Parameter`

a korreláció EQL-ben — együtt már támadási minta

Tipp: mennyi idő alatt jelenik meg a riasztás a SOC-ban?

Eredmények

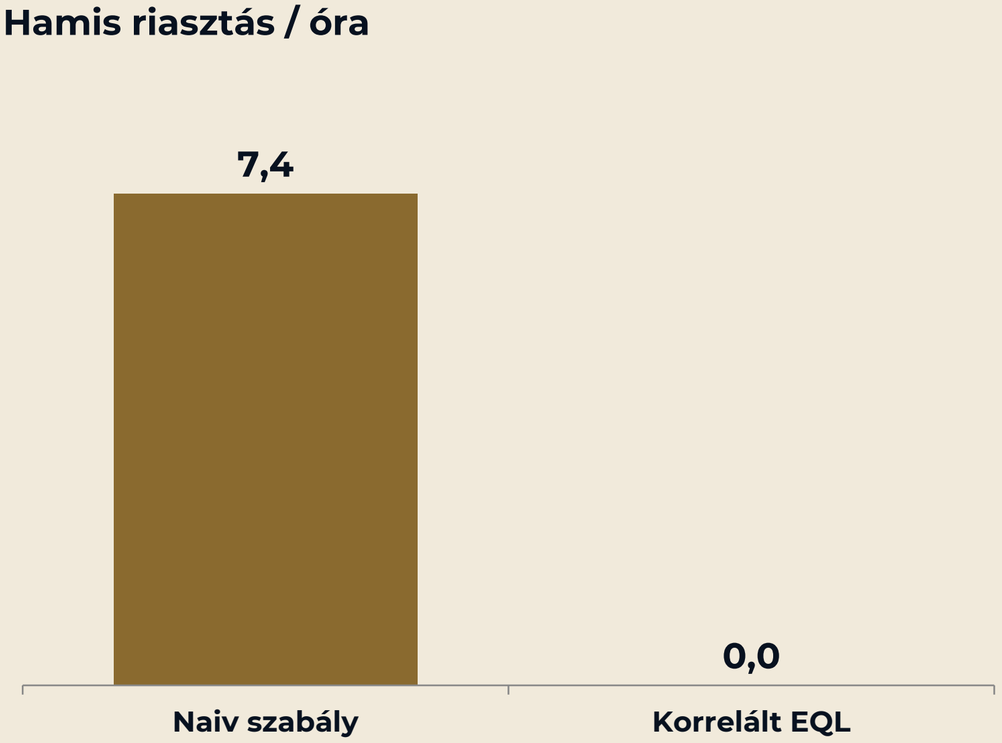
79 ms

medián riasztási idő · T0836 · N = 17 · p95: 107 ms

0,965
macro-F1 a lefedett technikákon

0
hamis pozitív korrelációval

2 ms
PLC STOP detekció (N = 20)



Aha! Az *„olcsó”* nem azt jelenti, hogy *„könnyű”*

Egy nulla, ami mindent eltört

```
0xe  ≠  0x0e
```

Osztályszintű hiba a COTP parserben — a szabályok csendben nem illeszkedtek.

A kész modul vak

```
Filebeat Zeek-modul
```

Nem ismeri az S7comm / COTP forgalmat. Saját pipeline kellett.

Elavult ATT&CK ID-k

```
T0855 → T1692.001
```

v19-ben visszavont technikák — a régi mappingek rossz technikára mutatnak.

Az „Aha!”: a nyílt forráskód működik — de csak ha érted, mit csinál.

Mennyire nehéz bevezetni? Három lépés.

1. **Tükrözött port**
   a cellazóna switchén (SPAN)

2. **Egy szenzor-gép**
   Zeek + Suricata, ICS-szabályokkal

3. **Bekötés a meglévő SIEM-be**
   riasztások ATT&CK for ICS technikákra

A PLC-hez nem nyúlsz.
A termelés nem áll meg.

NIS2 alatt ez már elvárás, **nem luxus.**

CYBERSEC

Három dolog, amit hazavihettek

1

OT-ban a hiba fizikai.

A védelemnek láthatatlannak kell lennie: passzív, nem beavatkozó.

2

A támadás ma egy laptop és pár sor kód.

A védelem nem maradhat a nagy költségvetések luxusa.

3

Nyílt forráskóddal is megy.

Ha érted a protokollt — nem csak a portot.

Köszönöm a figyelmet.

Szabados József · Óbudai Egyetem – Neumann János Informatikai Kar

LinkedIn: linkedin.com/in/szabados-joseph-1998a07a10