Financované Európskou úniou
NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI
ŽILINSKEJ UNIVERZITY V ŽILINE

Bezpečnosť v IoT

Moderné technológie, bezpečnosť cloudu a IoT (Blok VII)

Doc. Ing. Jozef Papán, PhD.

Jozef.papan@uniza.sk

Úvod

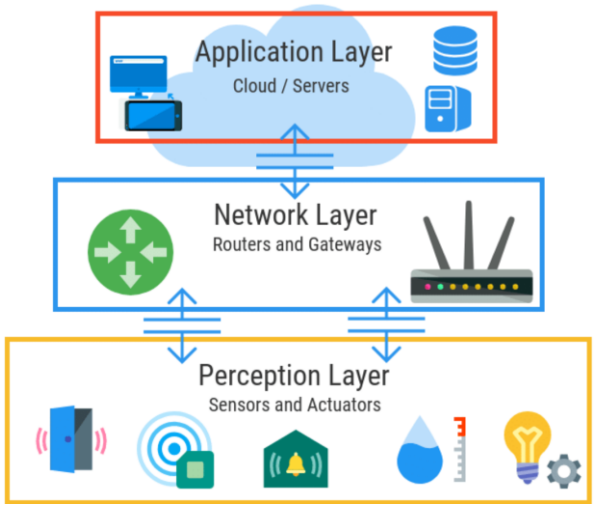
- Okrem masívneho rozmachu IoT treba zdôrazniť aj to, že IoT zariadenia často fungujú v absolútne kritických prostrediach, kde zlyhanie môže viesť k:
  - ohrozeniu zdravia (napr. infúzne pumpy s IoT modulom)
  - narušeniu bezpečnosti budovy (napr. manipulácia so smart zámkom)
  - sabotážam v priemysle (manipulácia s PLC)
  - environmentálnym škodám (chybný senzor tlaku na čerpadle)
- Ďalším faktorom je, že IoT komunikuje naprieč *rôznymi prostrediami naraz* — cloud, edge, mobilné siete, lokálna Wi-Fi, mesh siete. Tým vzniká **multivektorová zraniteľnosť**.
- Dôležitým faktorom sú aj **supply chain útoky** – ohrozenie môže vzniknúť už pri výrobe, montáži alebo aktualizácii zariadenia.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

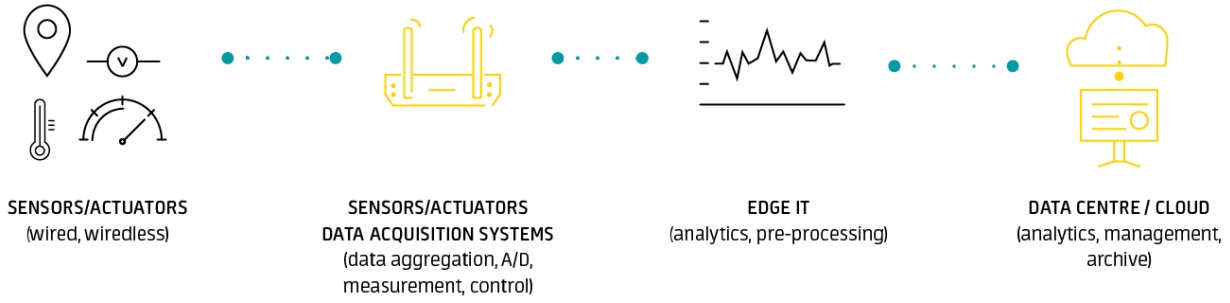
1. Architektúra IoT

Architektúra IoT

1. **Perception / Device vrstva** – fyzické senzory a aktuátory zbierajú údaje z prostredia a základne ich spracúvajú.
2. **Network vrstva** – zabezpečuje prenos dát medzi zariadeniami, bránami (gatewaymi) a cloudom (adresácia, smerovanie, šifrovanie).
3. **Middleware / Service vrstva** – prijíma, ukladá a spracúva dáta, spravuje zariadenia a integruje IoT s ostatnými systémami.
4. **Aplikačná vrstva** – poskytuje používateľské aplikácie, dashboardy a API, kde sa s dátami reálne pracuje.
5. **Business / Management vrstva** – vytvára analýzy, využíva AI/ML nad dátami a určuje pravidlá, procesy a biznis logiku.



Rôzne modely s podobnou štruktúrou



Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

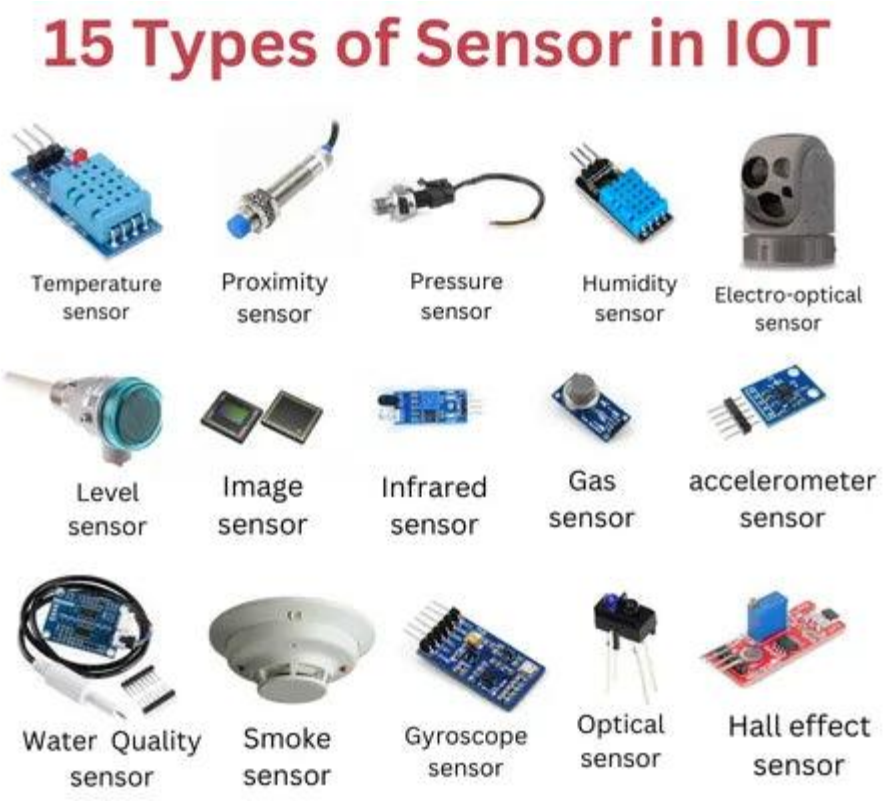
1.1 Percepčná vrstva

- Senzory v priemyselných systémoch často poskytujú *analógové dáta*, ktoré sa digitalizujú až v edge kontroléri. To otvára nové možnosti útokov:
  - fyzické rušenie senzora
  - manipulácia magnetickými poľami
  - vloženie útočníckeho signálu do senzora (napr. akustický útok na mikrofony alebo ultrazvukový útok na MEMS senzory)

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

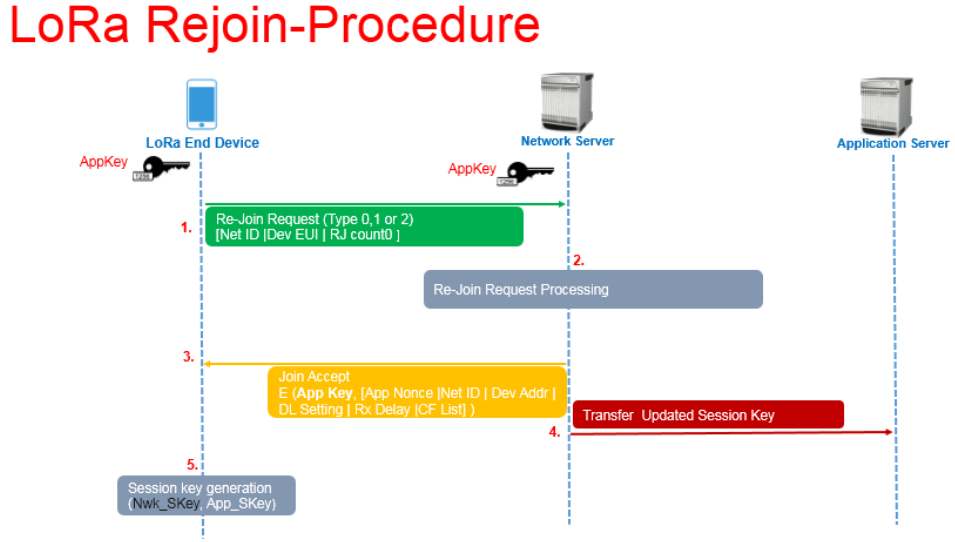
1.1 Percepčná vrstva

- Špeciálna kategória sú **kritické senzory**:
  - tlakové senzory v chemických závodoch
  - senzory vibrácií v turbínach
  - senzory prietoku v ekologických zariadeniach
- Ich zlyhanie môže mať obrovské následky → preto ich bezpečnostná integrita má inú úroveň ako bežné smarthome zariadenia.



1.2 Sieťová vrstva

- Útoky v bezdrôtových IoT sieťach majú jednu špecifickú vlastnosť:
  Útočník **nemusí byť v sieti**, stačí byť v jej dosahu.
  - To otvára dvere pasívnym a aktívnym útokom:
  - **pasívne** odpočúvanie signálu
  - **aktívne** vytvorenie falšovanej Zigbee siete
  - manipulácia mesh routingu (low-power mesh siete sú veľmi zraniteľné)
- Pri LoRaWAN treba zdôrazniť aj riziko:
  - slabého AppKey (ak je generovaný nesprávne)
  - neopravenej verzie LoRaWAN 1.0, kde niektoré kľúče boli statické

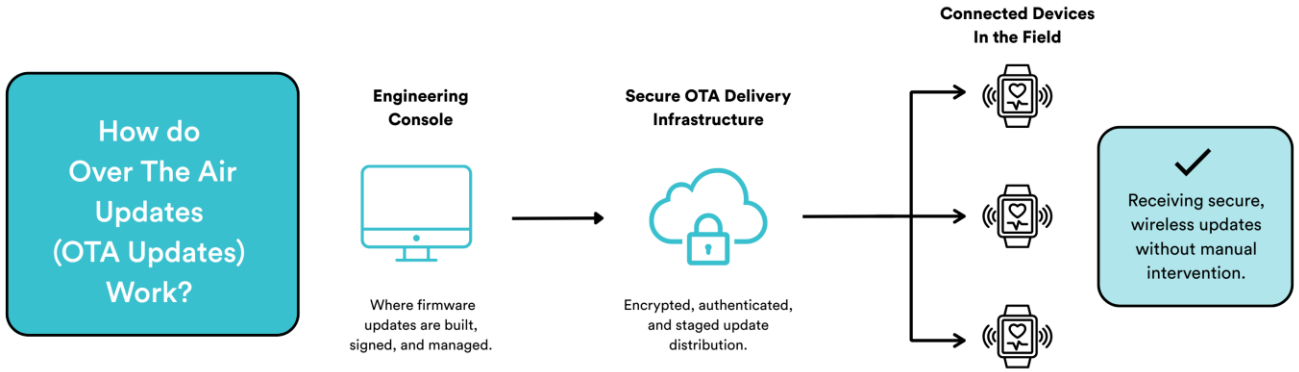


Čo je AppKey v LoRaWAN?

- **AppKey** je 128-bitový tajný kľúč, ktorý má poznať len:
  - zariadenie (node)
  - a sieť/aplikačný server
- Ak útočník pozná AppKey, vie si odvodiť session kľúče a:
  - čítať všetky dáta (dešifrovať)
  - vkladať falošné dáta (podstrčiť pakety)

1.3 Aplikačná vrstva

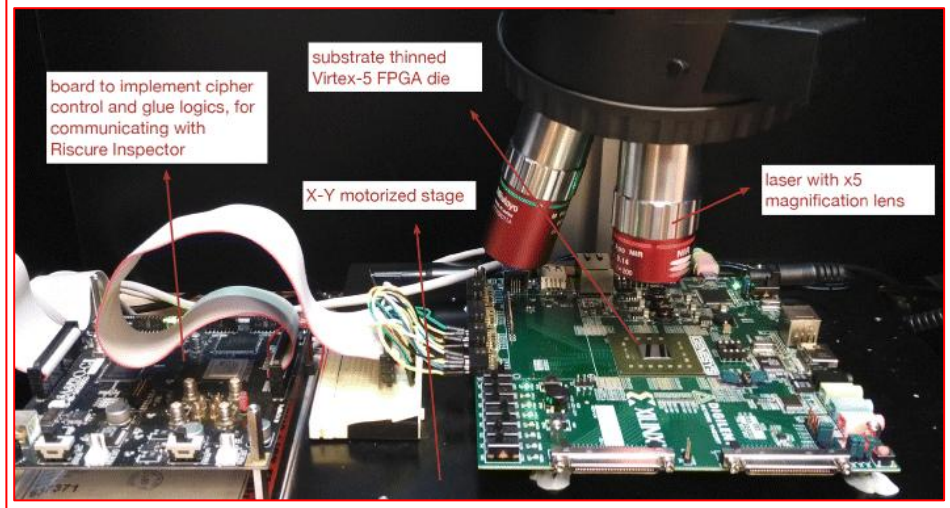
- API v IoT cloude často pracuje s:
  - telemetriou (údaje zo senzora)
  - príkazmi pre zariadenie
  - konfiguráciou (súradnice, prahové hodnoty)
  - OTA systémom (aktualizácie firmware)
- Každý z týchto vektorov predstavuje samostatnú bezpečnostnú hrozbu. Najcitlivejší je OTA – zneužitie môže viesť k *totálnej kompromitácii celej skupiny zariadení*.



2. Hrozby v IoT

2.1 Fyzické útoky

- Typické techniky útočníkov:
  - glitching (časovanie napájania → obídenie ochrany MCU)
  - laser fault injection
  - bus sniffing – čítanie komunikácie cez SPI alebo I²C linku
  - side-channel útoky – analýza spotreby energie zariadenia
- Tieto typy útokov sa bežne vyskytujú v bezpečnostných testoch automobiliek a smartmeter zariadení.

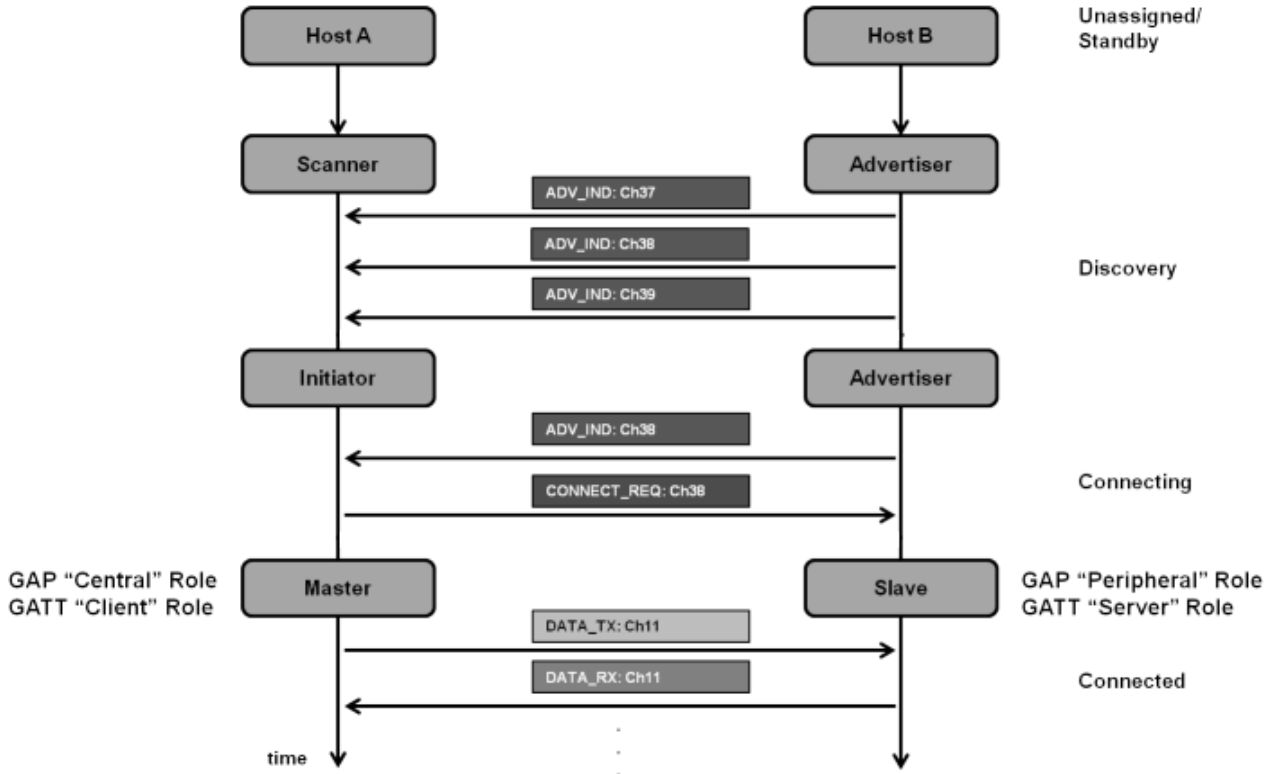


2.1 Fyzické útoky

- **Glitching (časovanie napájania)**
  - Útočník **úmyselne „kopne“ do napájania alebo hodinového signálu** (krátky výpadok, špička, skok).
  - Cieľ: spôsobiť, aby MCU **urobilo chybu v inštrukcii** – napr. preskočilo kontrolu PINu, hesla alebo ochrannú podmienku v boote.
  - Typicky sa to robí opakovane a presne časovane, kým sa nepodarí obísť bezpečnostný check.
- **Laser fault injection**
  - Útočník **zasiahne čip laserom** (cez odkrytý kryt/decapping), čím lokálne naruší jeho činnosť.
  - Laser môže spôsobiť, že sa **bit v registri/inštrukcii zmení** alebo sa vykoná nesprávne, podobne ako pri glitchingu.
  - Dá sa to použiť na **obídenie ochranných mechanizmov** alebo na extrakciu tajných kľúčov.
- **Bus sniffing (SPI / I²C)**
  - Útočník sa **pripojí sondami na dátové linky** (napr. SPI, I²C) medzi MCU a pamäťou alebo perifériami.
  - Pasívne **odpočúva komunikáciu** – čo si MCU číta/zapisuje (firmvér z flash, konfig, kľúče, heslá).
  - Ak nie je komunikácia šifrovaná, vie z nej **priamo vyčítať citlivé údaje**.
- **Side-channel útoky (spotreba energie)**
  - Zariadenie počas práce **mení odber prúdu** v závislosti od toho, čo práve počíta (inštrukcie, bity, kľúče).
  - Útočník meria **spotrebu energie v čase** (napr. cez rezistor v napájacej vetve) a robí z toho analýzu.
    - Jednoduchá analýza – pozriem sa „od oka“ (SPA).
    - Diferenciálna – štatisticky porovnávam veľa meraní (DPA).
  - Z týchto vzorov dokáže **odvodiť kryptografické kľúče** alebo iné tajné dáta – aj keď je protokol kryptograficky správny.

2.2 Sieťové útoky

- Low-power siete ako Zigbee alebo Z-Wave používajú:
  - multicast
  - broadcast
  - open JOIN fázu
- Tieto procesy, ak nie sú správne zabezpečené, môžu byť veľmi ľahko zneužiteľné.
- Pri Bluetooth LE (Low Energy) sú rizikové najmä:
  - pairing bez zabezpečenia - nevie overiť identitu druhej strany → **náchylné na MITM (Man-in-the-middle) útok**
  - sniffovanie reklamného (advertising) kanála
  - útoky na GATT služby



2.2 Sieťové útoky

- **Open JOIN fáza**
  V určitom čase je sieť (koordinátor / controller) v **„otvorenom“ režime prijímania nových zariadení**; ak nie je správne chránená (časovo obmedzená, autentizácia, whitelisting), útočník môže:
  - **pripojiť vlastné zariadenie** (malicious node),
  - alebo **odchytávať JOIN traffic** a získať parametre siete (network key, PAN ID, atď. pri slabšej konfigurácii).
- GATT (Generic Attribute Profile) je kľúčovou súčasťou Bluetooth Low Energy (BLE) pre IoT, ktorá definuje, ako si zariadenia vymieňajú dáta štruktúrovaným spôsobom, pričom organizujú informácie do služieb a charakteristík. Po nadviazaní pripojenia GATT definuje hierarchiu a postupy pre prenos dát, čo umožňuje zariadeniam, ako sú fitness trackery, odosielať dáta, ako je srdcová frekvencia, do smartfónu, ktorý funguje ako klient.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

2.3 Aplikačné útoky

- Cloud platformy pre IoT často používajú:
  - MQTT broker
  - REST API
  - WebSocket kanály
  - tokeny s časovým obmedzením
- Ak je token:
  - príliš dlhý
  - uložený v nezabezpečenej flash pamäti
  - alebo nikdy neexpiruje
- → zariadenie možno dlhodobo zneužiť.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

2.3 Aplikačné útoky

Prečo je token kritický a čo sa stane, keď je zlý

Ak je token:

1. **príliš dlhý (dlhá platnosť / TTL)**
   → útočník má veľké okno na zneužitie po jeho úniku.
   Namiesto „krátkeho incidentu“ z toho vie byť mesiace trvajúci backdoor.
2. **uložený v nezabezpečenej flash pamäti**
   → pri fyzickom prístupe (alebo pri FW dumpnutí cez debug porty) sa dá vytiahnuť.
   Najmä lacné MCU často nemajú secure storage → token je „len string v pamäti“.
3. **nikdy neexpiruje**
   → to je najhoršie: jeden únik = permanentná kontrola zariadenia alebo jeho identity v cloude.
   Útočník vie:

- publikovať falošné dáta do MQTT,
- odoberať telemetriu,
- posielať príkazy (ak má write scope),
- používať zariadenie ako vstupnú bránu do siete (pivoting).

Dlhodobé zneužitie je možné práve preto, že cloud verí tokenu, nie zariadeniu.
Keď má útočník token, je pre cloud „legitímny klient“.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

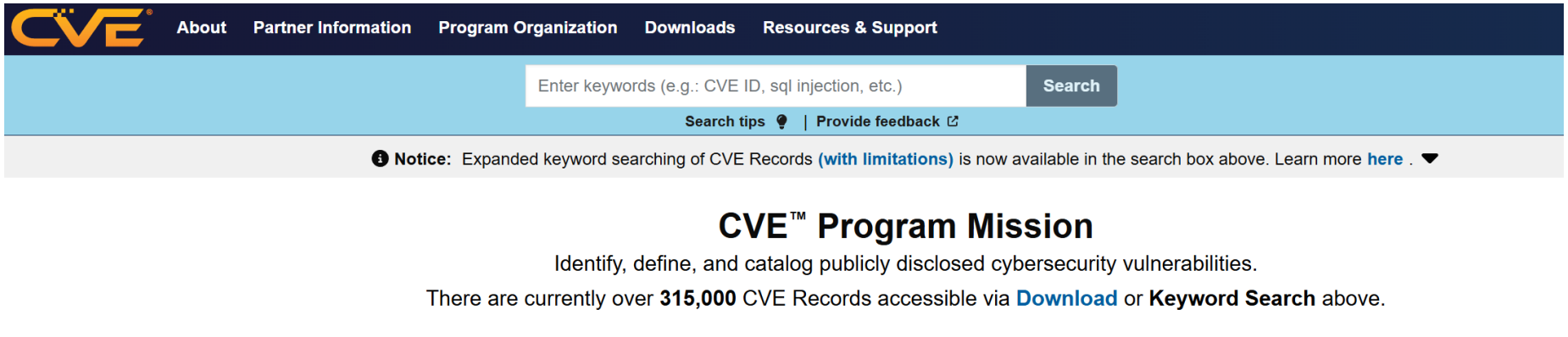
2.3 Aplikačné útoky

- MQTT broker – je centrálny server v protokole MQTT, ktorý robí „poštára“ medzi zariadeniami. **Zariadenia (klienti)** sa pripoja k brokeru cez TCP (často cez TLS).
  - Klient môže:
    - **publikovať (publish)** správy do nejakej *témy* (topic), napr. senzor/teplota
    - **odoberať (subscribe)** správy z tém, ktoré ho zaujímajú.
  - **Broker prijme publish** od jedného klienta a **rozošle ho všetkým**, ktorí majú subscribe na ten topic.
  - Čiže klienti **nekomunikujú priamo medzi sebou**, ale vždy cez brokera.

- REST API – klasické HTTP rozhranie (GET/POST/PUT/DELETE) na integráciu s inými systémami, konfiguráciu zariadení, čítanie historických dát a správu účtov/projektov.
- WebSocket kanály – trvalé obojsmerné spojenie cez web, vhodné pre live dashboardy, okamžité notifikácie a interaktívne ovládanie zariadení bez potreby neustáleho pollingu.
- Tokeny s časovým obmedzením – krátko platné prístupové tokeny (napr. JWT, OAuth2), ktoré sa po expirácii stávajú neplatnými, čím znižujú riziko zneužitia ukradnutých prihlasovacích údajov.

2.4 Malware

- Nové generácie IoT botnetov využívajú:
  - červí mechanizmus (self-propagation)
  - útoky na defaultné heslá
  - exploit známych zraniteľností (napr. CVE – Common Vulnerabilities and Exposures v kamerach Dahua, https://www.cve.org/)



- Niektoré vznikajú aj ako:
  - supply chain implantáty
  - škodlivé aktualizácie
- najčastejšie „lacné IoT riešenia z Číny“

3. Autentifikácia a autorizácia

3. Autentifikácia a autorizácia

- Najdôležitejšie je pochopiť rozdiel:
  - **Autentifikácia** → *Kto si?*
  - **Autorizácia** → *Čo môžeš robiť?*
- V IoT treba oba procesy riešiť na 3 úrovniach:
  1. zariadenie → cloud
  2. cloud → zariadenie
  3. zariadenie → zariadenie (najzložitejšie)

3. Autentifikácia a autorizácia

Problém v IoT je, že autentifikácia aj autorizácia sa kazia na viacerých miestach naraz:

1. **Zariadenie → cloud**

- slabá identita (statické tokeny/heslá),
- kľúče v nechránenej flash,
- chýba rotácia/expirácia → po úniku je zneužitie dlhodobé.

2. **Cloud → zariadenie**

- zariadenie nevie overiť „či príkaz naozaj môže prijať“,
- slabé/žiadne podpisovanie príkazov a OTA updatov.

3. **Zariadenie → zariadenie (najhoršie)**

- bez centrálnej autority musia veriť sami sebe,
- rôzne protokoly + slabý HW,
- často zdieľané kľúče alebo žiadne roly → každý môže všetko.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

3. Autentifikácia a autorizácia

- Moderné riešenia používajú:
  - mTLS (obojstranný TLS)
  - certifikáty X.509
  - krátkodobé tokeny (JWT, JWS, JWE)
  - OAuth2 Device Flow
- Najväčší problém: **bezpečné uloženie privátneho kľúča v zariadení**.
- Ak sa dá vytiahnuť z flashky/debug portu → útočník sa tvári ako legitímne zariadenie.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

3. Autentifikácia a autorizácia

- mTLS (mutual TLS): počas TLS sa overí server aj zariadenie certifikátmi. Zariadenie sa tak preukáže už pri vytváraní spojenia → silná autentifikácia na transportnej vrstve.
- X.509 certifikáty: štandardné TLS certifikáty s verejným kľúčom a identitou, podpísané CA. Sú „občiansky preukaz“ pre mTLS.
- Krátkodobé tokeny (JWT / JWS / JWE):
  - JWT = token s „claims“ (ID, roly, expirácia).
  - JWS = JWT podpísaný (integrita, autenticita).
  - JWE = JWT šifrovaný (utajenie obsahu).Používajú sa krátko, aby ukradnutý token rýchlo prestal platiť → hlavne na autorizáciu API.
- OAuth2 Device Flow: prihlásenie pre zariadenia bez UI. Zariadenie ukáže kód, používateľ ho potvrdí na mobile/PC, potom zariadenie dostane tokeny.

- mTLS/X.509 zabezpečí „kto si“, JWT zabezpečí „čo môžeš“, Device Flow zabezpečí „ako ťa pohodlne prihlásiť keď nemáš UI“

4. Šifrovanie a bezpečnosť komunikácie

3. Autentifikácia a autorizácia

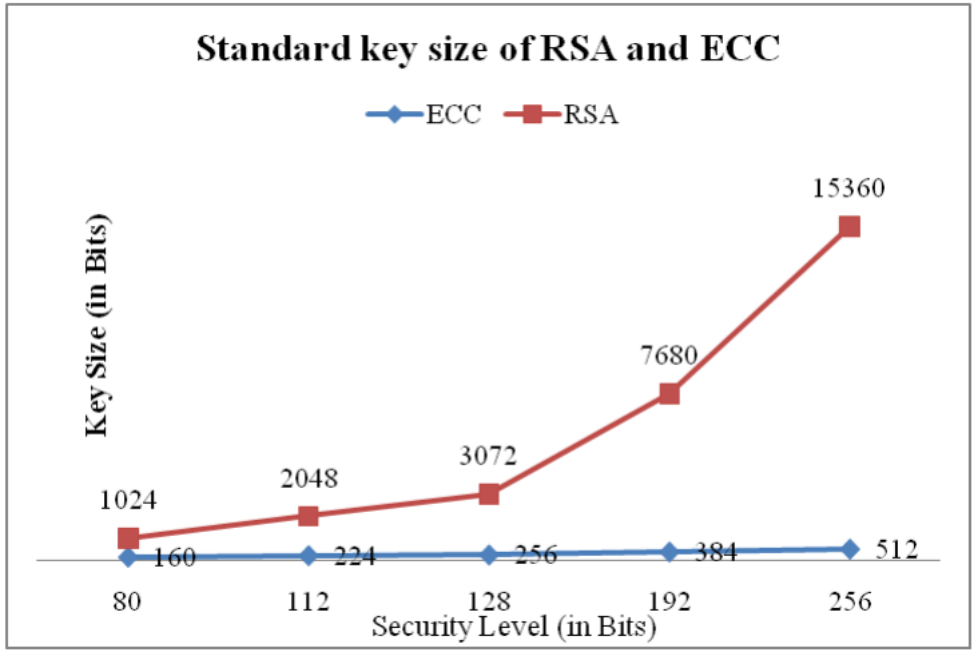
- IoT používa zvyčajne symetrické algoritmy (AES), pretože sú rýchle. Asymetrické (RSA, ECC) sú používané pri:
  - výmene kľúčov
  - autentifikácii
  - podpisovaní firmware
- **Prečo nie RSA?**
  - príliš veľké kľúče
  - príliš pomalé na MCU
- ECC (Elliptic Curve Cryptography) rieši tento problém.
- Treba zdôrazniť aj hrozbu:
  - downgrade útokov
  - nevalidácie certifikátu
  - nesprávne implementovaných TLS knižníc

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

RSA vs ECC

- ECC kľúče sú typicky ~6× až ~30× kratšie (v bitoch) pri rovnakej úrovni bezpečnosti.

Security Bits level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512



3. Autentifikácia a autorizácia

- IoT používa hlavne **symetrickú kryptografiu (AES)**, lebo je rýchla, úsporná na energiu a má malý overhead pre MCU.
- **Asymetria (RSA/ECC)** sa používa len na „štart dôvery“: **výmenu kľúčov, autentifikáciu a podpisovanie firmware**.
  - ECC znamená **Elliptic Curve Cryptography** – kryptografia na eliptických krivkách. Nie je to jedna konkrétna „šifra“, ale **rodina algoritmov** pre:
    - **výmenu kľúčov** (ECDH),
    - **digitálne podpisy** (ECDSA/EdDSA),
    - niekedy aj **šifrovanie dát** (ECIES).
- **RSA sa v IoT nehodí**, lebo potrebuje veľké kľúče (2048+ bit) a je príliš pomalá/výpočtovo náročné pre malé MCU.
- **ECC** dáva rovnakú bezpečnosť s oveľa menšími kľúčmi (napr. 256 bit), takže je rýchlejšie a vhodné pre IoT/TLS.
- Treba rátať aj s hrozbami: **downgrade útoky** (vynútenie slabších šifier/verzií), **nevalidácia certifikátu** (MITM), a **bugy v TLS knižniciach** (zlá implementácia zruší bezpečnosť).

5. Sieťová bezpečnosť

5. Sieťová bezpečnosť

- IoT segmentácia by mala obsahovať:
  - dedikovanú VLAN
  - firewall pravidlá len pre požadované porty
  - oddelený DNS pre IoT (zabraňuje exfiltrácii)
  - monitorovanie anomálií
- Zero Trust architektúra v IoT vyžaduje, aby každé zariadenie bolo:
  - identifikované
  - overené
  - autorizované
  - monitorované
- Aby nevznikli „slepé miesta“ v komunikácii.

ZTNA (Zero Trust Network Access)

- ZTNA (Zero Trust Network Access) je spôsob, ako sprístupniť interné aplikácie tak, aby sa **nikomu a ničomu automaticky neverilo** – ani keď je „v sieti“. Namiesto toho sa **každý prístup posudzuje zvlášť** podľa identity, zariadenia, kontextu a politiky.
- **Nie „pripoj sa do VPN a máš sieť“**, ale **„dostaneš prístup len k tejto konkrétnej aplikácii“**.
- Prístup je **per-session** (a často aj priebežne prehodnocovaný), nie „raz sa prihlásiš a potom už všetko“.

- **Používateľ otvorí aplikáciu** (napr. interný web, Git, CRM).

1. ZTNA riešenie urobí **silnú autentifikáciu**:
   - SSO (SAML/OIDC), MFA, prípadne certifikáty.
2. Zistí **stav zariadenia (device posture)**:
   - je spravované (MDM/EDR)? je šifrované? má aktuálne patchy? beží EDR agent?
3. Vyhodnotí **kontext**:
   - odkiaľ sa pripája (geo/IP), čas, rizikové skóre, typ siete (home vs public Wi-Fi), správanie.
4. **Politika rozhodne** (policy engine):
   - „Tento user + toto zariadenie + tento kontext → môže do aplikácie X, ale nie do Y.“
5. Ak je to OK, ZTNA vytvorí **krátkodobé, úzko obmedzené spojenie**:
   - často cez **broker/gateway** alebo **cloudový edge**.
   - aplikácia sa „neotvorí do internetu“; skôr sa vytvorí kontrolovaný tunel k nej.

- **Priebežná kontrola**:
  - keď sa zmení riziko (napr. vypne sa EDR, zmení sa IP, anomália), prístup sa môže **znížiť** (step-up MFA) alebo **odstrihnúť**.

6. Firmware & Hardware bezpečnosť

6. Firmware & Hardware bezpečnosť

- Útočníci často získajú firmware zo zariadenia:
  - cez UART
  - cez SPI flash
  - cez bootloader
  - cez JTAG
  - cez OTA (ak nie je šifrované)
- Po získaní firmware môžu:
  - extrahovať kľúče
  - analyzovať slabiny
  - vytvoriť vlastný maliciózny firmware
- Preto je dôležité používať:
  - secure boot
  - flash encryption
  - anti-rollback ochranu
  - code signing

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

6. Firmware & Hardware bezpečnosť

Keď útočník získa firmware zo zariadenia (napr. z flash pamäte alebo z OTA balíka), dostane „vnútro“ systému, a tým vie urobiť tieto veci:

- **Extrahovať kľúče** – vo firmvéri bývajú uložené hard-coded kľúče, certifikáty, tokeny alebo ich derivácie; ak ich nájde, môže dešifrovať komunikáciu alebo sa vydávať za zariadenie.
- **Analyzovať slabiny** – vie reverzne prečítať logiku, nájsť chyby (buffer overflow, debug backdoor, slabé RNG, zlé TLS nastavenie) a pripraviť presný exploit.
- **Vytvoriť vlastný maliciózny firmware** – môže ho upraviť (napr. pridať zadné vrátka, vypnúť bezpečnostné kontroly) a pokúsiť sa ho nahrať späť do zariadenia.

Preto sa používajú tieto ochrany:

- **Secure boot** – zariadenie pri štarte spustí iba firmware, ktorého podpis/verziu vie dôveryhodne overiť; bráni nahratiu cudzieho kódu.
- **Flash encryption** – obsah flash pamäte je šifrovaný, takže aj keď ju útočník prečíta „z dosky“, uvidí len nezmysly a nevytiahne kľúče ani logiku.
- **Anti-rollback ochrana** – nedovolí nahrať staršiu (zraniteľnú) verziu firmware; blokuje downgrade útoky.
- **Code signing** – update/firmware je digitálne podpísaný výrobcom a zariadenie podpis kontroluje; bez správneho podpisu sa firmware nenainštaluje.

Spolu to rieši dve veci: **aby útočník nevedel firmware ľahko prečítať** (flash encryption) a **aby nevedel do zariadenia dostať svoj upravený kód** (secure boot + code signing + anti-rollback).

7. Machine Learning a AI v IoT

7. Machine Learning a AI v IoT

- Dnes sa používajú dva veľké prístupy:
- **1. Edge ML**
- Časť výpočtov prebieha na zariadení:
  - menej dát v cloude
  - nižšia latencia
  - lepšia detekcia v reálnom čase
- **2. Cloud ML**
- Analýza veľkého množstva dát:
  - deep learning
  - klasifikácia sieťovej prevádzky
  - predikcia degradácie hardvéru

- Detekčné modely často sledujú:
  - periodicitu prenosov
  - odchýlky v energii
  - neštandardné požiadavky
  - spike-y v dátovej aktivite

7. Machine Learning a AI v IoT

- V IoT sa ML rieši najčastejšie dvoma smermi:
- **1. Edge ML**
  Model (alebo jeho časť) beží priamo na zariadení či gatewayi, takže sa do cloudu posiela menej dát, reakcia je rýchlejšia (nižšia latencia) a anomálie sa vedia zachytiť hneď v reálnom čase, aj keď je pripojenie slabé alebo prerušované.
- **2. Cloud ML**
  Dáta zo zariadení sa zbierajú centrálne a v cloude sa nad veľkými datasetmi robí ťažšia analýza – napr. deep learning, klasifikácia sieťovej prevádzky alebo predikcia degradácie hardvéru – čo je výpočtovo náročné, ale presnejšie pri komplexných úlohách.
- **Čo detekčné modely v IoT typicky sledujú:**
- **Periodicitu prenosov** – veľa IoT zariadení komunikuje pravidelne; zmena rytmu môže znamenať kompromitáciu.
- **Odchýlky v energii/spotrebe** – nečakané skoky v odbere často signalizujú neštandardnú aktivitu alebo malvér.
- **Neštandardné požiadavky** – zariadenie začne volať netypické endpointy, porty alebo posielať iné typy správ.
- **Spike-y v dátovej aktivite** – náhle objemy dát či bursty môžu byť znak DoS, exfiltrácie alebo chyby v zariadení.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

8. Best practices

8. Best practices

- Okrem noriem treba uplatňovať aj:
  - Secure SDLC (Secure Software Development Lifecycle)
  - Penetračné testovanie IoT
  - pravidelné ROTÁCIE kľúčov
  - logovanie a audit trail
  - generovanie entropy pri výrobe
- Veľmi dôležitý bod:
  - **IoT zariadenie musí fungovať bezpečne aj po 10 rokoch.**
- To znamená:
  - používateľ musí vedieť bezpečne aktualizovať
  - zariadenie nesmie používať zastarané protokoly
  - výrobca musí garantovať support

8. Best practices

- Secure SDLC (Secure Software Development Lifecycle) – bezpečnosť sa rieši už počas návrhu, vývoja a testovania firmvéru/aplikácií (threat modeling, secure coding, code review), nie až keď je produkt hotový.
- Penetračné testovanie IoT – systematické „hackovanie“ vlastného zariadenia, firmvéru, mobilnej appky aj cloudu, aby sa našli reálne zraniteľnosti skôr než útočník.
- Pravidelné rotácie kľúčov – kryptografické kľúče (napr. pre komunikáciu či OTA) sa periodicky menia, takže aj keď sa niektorý kompromituje, útočník má len krátke okno na zneužitie.
- Logovanie a audit trail – zariadenia aj cloud musia zapisovať bezpečnostne dôležité udalosti (join, zmeny configu, update, chyby), aby sa dali incidenty spätne analyzovať a odhaliť anomálie.
- Generovanie entropy pri výrobe – každé zariadenie musí dostať unikátne a silne náhodné kľúče/seed už vo výrobe (nie defaultné alebo odvodené), inak sa dá kompromitovať celé produktové portfólio naraz.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

9. Záver

9. Záver

- IoT bude v nasledujúcich rokoch základom:
  - energetiky
  - výroby
  - zdravotníctva
  - smart miest
  - logistiky
  - domácich technológií
- A práve preto musí byť aj bezpečný.
  Dôsledky hackerských útokov na IoT už dávno presahujú virtuálny svet — ide doslova o **bezpečnosť ľudí, podnikov aj národných infraštruktúr**.

Financované Európskou úniou
NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Moderné technológie, bezpečnosť cloudu a IoT (Blok VII)

Doc. Ing. Jozef Papán, PhD.

Jozef.papan@uniza.sk