Destroy IOT

Twój Stary Linuksiarz

2 Kwietnia 2005 - 21:37

Spis treści

1 Wykład 1: Wprowadzenie do IoT

1.1 Czym jest IoT?

- **Internet:**
  - Sieć sieci
  - Wykorzystuje sieci przewodowe lub łącza bezprzewodowe
  - Przejście do IoT
- **Definicje IoT:**
  - **RFC 7452** - trend, w którym duża liczba wbudowanych systemów zwanych również inteligentnymi obiektami stosuje usługi komunikacyjne oferowane przez protokoły internetowe.
  - **Inna definicja** - Internet rzeczy to globalna infrastruktura społeczeństwa informacyjnego umożliwiająca zaawansowanie usługi poprzez łączenie fizycznych bądź wirtualnych rzeczy opartych o istniejące lub rozwijające się technologie komunikacyjne.
  - **Słownik Oxford'u** - Internet rzeczy to wzajemne połączenie wbudowanych urządzeń w przedmiotach codziennego użytku, które umożliwiają wysyłanie i odbieranie danych za pośrednictwem Internetu.

1.2 Ewolucja Internetu

Etap	**Nazwa**	**Charakterystyka**
Etap 1	Łączność	Digitalizacja dostępu do informacji • E-mail • Przeglądarka WWW • Wyszukiwanie
Etap 2	Usieciowiona gospodarka	Digitalizacja procesów biznesowych • Handel elektroniczny (E-commerce) • Digitalizacja łańcucha dostaw • Współpraca (Collaboration)
Etap 3	Doświadczenia oparte na współpracy	Digitalizacja interakcji (biznesowych i towarzyskich) • Media społecznościowe • Mobilność • Chmura • Wideo

Etap 4	Internet Wszechrzeczy	Digitalizacja świata i łączenie: • Ludzi • Procesów • Danych • Rzeczy

1.3 Rozwój technologiczny

- **Prawo Moore'a** — Liczba tranzystorów znajdujących się w układach scalonych często zwiększa się dwukrotnie co dwa lata, powodując wzrost ich mocy obliczeniowej.
- **Prawo Metcalfe'a** — Użyteczność sieci wzrasta w sposób proporcjonalny do kwadratu ilości użytkowników podłączonych do tej sieci.
- **Prawo Reeda** — Użyteczność sieci wzrasta w sposób wykładniczy w przypadku, gdy możliwości grup dwuosobowych, trzyosobowych itd. w jakie można połączyć użytkowników, zostaną zsumowane.

1.4 Sieci jako fundament IoT

- Pięćdziesiąt miliardów urządzeń udostępnia biliony gigabajtów danych
- Sieci wciąż się rozwijają np. konwergencja
- Sieci różnych rozmiarów
- Komponenty sieci
- Urządzenia końcowe

1.5 Dynamiczny rozwój IoT - przyczyny

- Wszechobecność łączy
- Powszechne stosowanie protokołu IP
- Inwestowanie w rozwój informatyki
- Miniaturyzacja
- Rozwój i analiza danych
- Rosnące możliwości przetwarzania w chmurze obliczeniowej

1.6 Możliwości jakie daje Internet Rzeczy

Kategoria	**Opis**	**Przykłady**

Człowiek	Urządzenia wszczepione lub noszone przy ciele człowieka	Urządzenia monitorujące pracę serca i inne czynności życiowe, zarządzanie chorobami, wysiłkiem fizycznym, jak i produktywnością
Dom	Gospodarstwa domowe	Urządzenia sterujące światłem, ociepleniem, klimatyzacją, roletami, urządzeniami RTV, AGD, systemy bezpieczeństwa
Sprzedaż detaliczna	Miejsca, w których konsument angażuje się w handel	Sklepy, banki, restauracje, galerie handlowe, kasy samoobsługowe, optymalizacja inwentaryzacji
Biuro	Miejsca pracy umysłowej	Zarządzanie energią, bezpieczeństwem, zwiększenie produktywności oraz możliwości pracy zdalnej
Fabryki	Środowiska produkcyjne	Optymalizacja w miejscach pracy powtarzalnej, lepsze wykorzystanie sprzętu i przestrzeni magazynowej w fabrykach, kopalniach, platformach wiertniczych, poprawienie bezpieczeństwa
Pojazdy	Systemy stosowane w pojazdach	Systemy bezpieczeństwa, wspomagające kierowanie, informowanie o problemach drogowych i wspomagające utrzymanie
Miasta	Środowiska miejskie	Miejsca publiczne, sterowanie ruchem, inteligentne metra, tramwaje, środowiska monitoringu, zarządzanie zasobami

1.7 Internet wszystkiego (IoE)

IoE łączy cztery kluczowe elementy:

- **Ludzie** - Łączenie ludzi za pomocą bardziej odpowiednich i wartościowych sposobów
- **Dane** - Transformacja danych w bardziej przydatne informacje w procesach podejmowania decyzji
- **Procesy** - Dostarczanie odpowiednich informacji do właściwej osoby (lub maszyny) we właściwym czasie
- **Urządzenia** - Fizyczne urządzenia i przedmioty podłączone do Internetu i połączone ze sobą, wykorzystywane do inteligentnego podejmowania dobrych decyzji

1.8 Cechy systemów IoE

Cecha	**Opis**

Hiperświadoma	• Bierze pod uwagę lokalizację, status, stan i otoczenie firmy • Monitoruje zachowania klientów i ich nastroje w czasie rzeczywistym • Określa potrzeby rynku i zmiany konkurencyjne
Przewidująca	• Przewiduje zmiany na rynku • Optymalizuje wydajność aktywów • Przewiduje i proaktywnie odpowiada na pojawiające się zagrożenia bezpieczeństwa
Elastyczna	• Osiąga różnicowanie konkurencyjne odpowiadając na konkurencję • Dba o rozwój innowacyjności poprzez budowanie „platform"umożliwiających utrzymywanie trwałej przewagi konkurencyjnej • Odpowiada na szybko ewoluujące zagrożenia • Przyspiesza innowacje

1.9 Główne priorytety IoE

1. **Doświadczenie klienta** - Poprawianie relacji z klientami w celu zdobycia większego udziału w rynku
2. **Innowacje** - Skrócenie czasu potrzebnego do sprzedaży produktów i poprawa rozwoju produktu, aby dobrze odpowiadać na potrzeby klienta
3. **Wydajność pracowników** - Umożliwienie większej wydajności i skalowalności
4. **Wykorzystanie zasobów** - Osiąganie niższych kosztów
5. **Dostawa** - Identyfikacja obszarów występowania marnotrawstwa i opóźnień przy jednoczesnym zwiększeniu efektywności logistycznej

1.10 IoE a przemysł

- Wielowymiarowa analiza danych w czasie rzeczywistym, zintegrowana współpraca wykorzystująca transmisje wideo oraz zdalne śledzenie zasobów fizycznych
- Integracja danych odbieranych z czujników, możliwości sprawnego kierowania personelem oraz tworzenie analiz predykcyjnych np. zużycia prądu

- Transmisja wideo, analiza zachowań klienta, analiza i wizualizacja danych oraz marketing zależny od lokalizacji i dostępny na wszystkich urządzeniach

1.11 Typy komunikacji w IoT

1. **Urządzenie — Urządzenie**
   - Bezpośrednia komunikacja między urządzeniami (np. czujnik temperatury komunikujący się z włącznikiem światła)
2. **Urządzenie — Chmura**
   - Urządzenia łączą się bezpośrednio z usługą w chmurze
3. **Urządzenie — Brama**
   - Urządzenia łączą się z pośrednim urządzeniem (bramą), która przetwarza dane przed wysłaniem ich dalej
4. **Back-end Data-sharing Model**
   - Wymiana danych między różnymi usługami w chmurze
5. **Fog Computing**
   - Przetwarzanie danych bliżej urządzeń końcowych, zmniejszające opóźnienia w komunikacji

2 Wykład 2: Bezpieczeństwo IoT

2.1 Wstęp do bezpieczeństwa IoT

- Według przewidywań Gartnera do 2020 roku na świecie miało być ponad 20 miliardów urządzeń z kategorii Internetu rzeczy
- Najwięcej urządzeń pojawia się w dziedzinie:
  - Sportu i zdrowego trybu życia
  - Transportu
  - Wyposażenia domu
- W przypadku rozwijania się tzw. inteligentnych miast (smart cities) najistotniejsze sektory to:
  - Zarządzanie ruchem drogowym i komunikacją miejską
  - Oszczędzanie energii (np. automatyzacja oświetlenia)
  - Bezpieczeństwo publiczne (inteligentny monitoring)
- Producenci prześcigają się w implementowaniu modułów Wi-Fi nawet w drobnym AGD, co zwiększa ryzyko uzyskania dostępu do naszych sieci przez osoby niepożądane

2.2 Główny problem bezpieczeństwa IoT

Bezpieczeństwo IoT jest obecnie bagatelizowane, ponieważ:

- Brakuje odpowiednich regulacji, certyfikatów, ograniczeń, ustaw, które wymagałyby od producentów podjęcia środków ostrożności
- Nowe produkty masowo wypuszczane są na rynek — ich tworzenie nie jest drogie i zajmują się tym często młode firmy oraz start'upy

2.3 Rodzaje zagrożeń w IoT

2.3.1 Rzeczy niemożliwe do zabezpieczenia

- Niektóre urządzenia IoT, szczególnie typu "plug-and-play", mogą być trudne lub niemożliwe do konfiguracji
- Inteligentne urządzenia AGD mogą być rekrutowane do botnetów lub wykorzystywane do zbierania informacji o sieci domowej

2.3.2 "Słabe"punkty dostępowe

- Luki w zabezpieczeniach IoT są wykorzystywane do ataków DDoS
- Hasła o niskiej sile (domyślne hasła jak "root", "admin", "123456"lub "test")
- Większość użytkowników nie zmienia haseł domyślnych

2.3.3 Ataki na zakłady produkcyjne

- IIoT (Industrial Internet of Things) wymaga otwartych systemów informatycznych, co zwiększa ryzyko ataków
- Przykład: Atak na irańskie obiekty nuklearne

2.3.4 Przechwytywanie dźwięku i obrazu

- Cyberprzestępcy mogą uzyskać dostęp do mikrofonu lub kamery urządzenia podłączonego do sieci
- Przechwycone urządzenie może służyć jako "brama"do innych urządzeń w sieci

2.3.5 Zagrożenia dla zdrowia i życia ludzi

- Przykład 6: Ataki na pojazdy (włamanie do Jeepa Cherokee)
- Przykład 7: Ataki na infrastrukturę medyczną (rozruszniki serca, urządzenia do monitorowania)

2.3.6 Kradzież tożsamości

- Istniejące systemy zabezpieczeń sieci często nie widzą urządzeń IoT
- Gromadzenie danych o użytkownikach umożliwia wyrafinowane ataki
- Przykłady: wyłudzanie kredytów, wynajmowanie domów na cudze dane, podszywanie się pod menedżerów w celu wyłudzenia pieniędzy

2.3.7 Kradzież technologii w przemyśle

- Tradycyjne metody szpiegostwa przemysłowego: telefony komórkowe, dyktafony, aparaty fotograficzne
- Nowe metody wykorzystujące IoT: smartwatch, smart pen, inteligentne okulary, pamięci USB z modułem Wi-Fi

2.4 Jak zabezpieczać IoT?

2.4.1 W domu:

- Aktualizacja oprogramowania urządzeń IoT (aktualizacje zabezpieczeń)
- Odpowiednia konfiguracja urządzeń (np. zmiana domyślnych haseł dostępowych)
- Izolowanie urządzeń IoT w odpowiednio chronionych sieciach
- Korzystanie z urządzeń dostarczanych przez sprawdzonych dostawców o zaufanej opinii
- Odłączanie od sieci urządzeń, z których aktualnie nie korzystamy
- Przemyślane dołączanie urządzeń IoT do sieci — czy w ogóle jest to dla nas konieczne?

2.4.2 W firmie:

- Aktualizacja oprogramowania urządzeń IoT (aktualizacje zabezpieczeń)
- Odpowiednia konfiguracja urządzeń (np. zmiana domyślnych haseł dostępowych)
- Izolowanie urządzeń IoT w odpowiednio chronionych sieciach (rozdział sieci biurowej od sieci urządzeń przemysłowych/maszyn)
- Korzystanie z urządzeń dostarczanych przez sprawdzonych dostawców o zaufanej opinii
- Przestrzeganie podstawowych zasad bezpieczeństwa przez pracowników (nieotwieranie podejrzanych wiadomości, aktualne oprogramowanie antywirusowe)

2.5 Wyzwania dla twórców IoT

- Wysokiej jakości ochrona informacji zbieranych do chmury przez producentów urządzeń
- Dostarczanie częstych aktualizacji oprogramowania dla urządzeń zawierających aktualne „łatki"zabezpie
- Szybka reakcja na „dziury"wykrywane w oprogramowaniu urządzeń IoT
- Uzyskiwanie odpowiednich certyfikacji zabezpieczeń oferowanych przez siebie urządzeń
- Korzystanie z wysokiej jakości oprogramowania dla urządzeń

2.6 Przykłady ataków na IoT

2.6.1 Mirai Botnet

- Malware atakujący urządzenia IoT z domyślnymi danymi logowania
- Kamery CCTV stanowią większość celów Mirai
- Używa słownikowego ataku brute force
- Październik 2016 - Atak na Dyn (dostawcę DNS), powodując przerwy w dostępie do internetu dla milionów użytkowników w USA i Europie

2.6.2 IT i OT w sektorze produkcyjnym

- Dwie odrębne domeny sieciowe w organizacjach:
  - Technologia informacyjna (IT) - urządzenia w centrum danych, w chmurze, BYOD i tysiące czujników
  - Technologia operacyjna (OT) - systemy sterowania przemysłowego, SCADA i wszystkie podłączone do nich urządzenia
- Świat produkcji się zmienia - IT i OT zaczynają współpracować

2.7 Przypadki użycia IoT

2.7.1 Inteligentny dom

- Przykład transformacji sposobu życia, pracy i rozrywki dzięki IoT
- Urządzenia inteligentnego domu: oświetlenie, termostaty, systemy bezpieczeństwa, wykrywanie dymu i ognia, urządzenia, telewizory, drzwi, okna

2.7.2 Opieka zdrowotna

- Osobiste urządzenia fitness: komunikacja z aplikacją w chmurze, połączenie Bluetooth z telefonem
- Monitorowanie opieki zdrowotnej: zbieranie i ocena danych pacjenta, zdalne monitorowanie pacjenta (RPM)
- IoT w szpitalu: do 20 urządzeń medycznych w jednej sali szpitalnej, monitorowanie i przesyłanie danych pacjenta

2.7.3 Zagrożenia w opiece zdrowotnej

- Atakowanie rozruszników serca - podatności w urządzeniach medycznych
- Podatności sieci opieki zdrowotnej - słaba autentykacja, niezabezpieczone procesy serwerowe
- Długi okres użytkowania urządzeń medycznych - stare i niezaktualizowane systemy operacyjne
- Słabe regulacje dla urządzeń medycznych

2.7.4 Ryzyko w opiece zdrowotnej

- Manipulowanie urządzeniami medycznymi może prowadzić do obrażeń lub śmierci pacjenta
- Kradzież danych przechowywanych na urządzeniach lub w sieci
- Kradzież danych osobowych pacjentów
- Naruszenie przepisów dotyczących ochrony danych osobowych

2.7.5 Środki zaradcze w opiece zdrowotnej

- Producenci urządzeń: projektowanie i budowanie urządzeń z myślą o bezpieczeństwie
- Administratorzy opieki zdrowotnej: zapewnienie bezpieczeństwa zakupionych urządzeń
- Personel IT: zapewnienie niezawodnych środków aktualizacji urządzeń podłączonych do sieci
- Architektura sieci: izolacja sieci danych i kontroli
- Personel opieki zdrowotnej: szkolenia zwiększające świadomość bezpieczeństwa

3 Wprowadzenie do IoT i Niezabezpieczonych Urządzeń

3.1 Obietnica IoT (Promise of IoT)

- Internet Rzeczy (IoT) ma potencjał do transformacji wielu aspektów naszego życia ("Co moglibyśmy zrobić bez ograniczeń?").
- Powszechność urządzeń podłączonych do sieci stale rośnie:
  - Smartfony, laptopy, tablety.
  - Urządzenia do śledzenia aktywności fizycznej (wearables).
  - Konsole do gier.
  - Systemy do zdalnego sterowania termostatem.
  - Podłączone lodówki i inne urządzenia AGD.

3.2 Anatomia Ataku IoT (Anatomy of an IoT Attack)

- IoT **rozszerzył możliwości** działania dla atakujących (threat actors) przeciwko naszym sieciom.
- Urządzenia IoT są **coraz częściej kompromitowane**.
- Urządzenia IoT są wykorzystywane w szerokiej gamie ataków, ponieważ **brakuje im krytycznych zabezpieczeń**, takich jak:
  - Silne hasła (często mają domyślne, łatwe do odgadnięcia).
  - Aktualne systemy operacyjne (często nie są aktualizowane).
  - Segmentacja sieci (często podłączone do tej samej sieci co inne, ważniejsze systemy).

3.3 Przykład: Botnet Mirai (Mirai Botnet)

- Mirai to złośliwe oprogramowanie (malware) celujące w urządzenia IoT skonfigurowane z **domyślnymi danymi logowania**.
- Głównym celem Mirai były kamery **CCTV**.
- Mirai wykorzystuje atak **słownikowy typu brute force**, przeszukując listę domyślnych nazw użytkowników i haseł.
- W październiku 2016 roku zaatakowano usługi dostawcy DNS - firmy **Dyn**, co spowodowało przerwy w dostępie do Internetu dla milionów użytkowników w USA i Europie. Atak został przeprowadzony przez botnet składający się głównie z zainfekowanych urządzeń IoT.

4 Unikalne Ryzyko IoT - Konwergencja Domen

4.1 IT i OT w Sektorze Produkcyjnym

W organizacjach istnieją dwie odrębne domeny sieciowe:

- **Technologia Informacyjna (IT - Information Technology):** Obejmuje urządzenia w centrum danych, chmurze, urządzenia prywatne (BYOD) oraz tysiące czujników i aktuatorów podłączonych w terenie. Zarządza aplikacjami biznesowymi.
- **Technologia Operacyjna (OT - Operational Technology):** Obejmuje przemysłowe systemy sterowania (ICS), systemy nadzoru i akwizycji danych (SCADA) oraz wszystkie urządzenia, które łączą się z tymi systemami. Odpowiada za płynne działanie np. linii produkcyjnej.
- Historycznie OT utrzymywało działanie zakładu, a IT zarządzało aplikacjami biznesowymi z biura.
- Świat produkcji się zmienia - następuje **konwergencja IT i OT**:
  - Menedżerowie operacyjni IT i OT używają narzędzi IT do analizy ogromnych ilości danych operacyjnych i podejmowania decyzji w czasie rzeczywistym.
  - Zespoły IT mogą wykorzystywać te dane do innowacji, np. poprawy łańcucha dostaw i redukcji przestojów.

4.2 Technologia Konsumencka (Consumer Technology - CT)

- CT obejmuje podłączone urządzenia w domu, technologię noszoną (wearables), inteligentne samochody i inne.
- Rośnie liczba urządzeń wykorzystywanych do komunikacji.
- W 2016 roku ruch internetowy z urządzeń CT stanowił **61%** całego ruchu IP. Z tego ruchu CT, aż **81%** stanowił ruch wideo.

4.3 Model Bezpieczeństwa IoT

- Niezależnie od tego, czy urządzenie IoT należy do domeny IT, OT, CT, czy kombinacji tych trzech, **wymagane jest silne bezpieczeństwo**.
- **Dostawcy usług (Service Providers)** to organizacje, które podłączają nasze urządzenia do Internetu.
- Mają oni możliwość oferowania usług odpowiadających na potrzeby bezpieczeństwa IoT swoich klientów.
- Konceptualny model bezpieczeństwa IoT pokazuje te trzy domeny (IT, OT, CT) jako wewnętrzne obszary, otoczone przez dostawców usług, a wszystko to w ramach ogólnego bezpieczeństwa IoT.

5 Przypadki Użycia IoT (Use Cases)

5.1 Inteligentny Dom (Smart Home)

- Inteligentny dom jest przykładem, jak IoT transformuje sposób, w jaki żyjemy, pracujemy i bawimy się.

- Urządzenia inteligentnego domu obejmują: oświetlenie, termostaty, systemy bezpieczeństwa, czujniki dymu i ognia, urządzenia AGD, telewizory, drzwi, okna i wszystko, co może być zdalnie monitorowane i kontrolowane.
- (Wzmianka o laboratorium Packet Tracer do eksploracji inteligentnego domu).
- (Wzmianka o laboratorium do oceny produktów automatyki domowej, uwzględniającym potencjalne wymagania dotyczące usług monitorowania/konserwacji i wpływ lokalizacji geograficznej).

5.2 Opieka Zdrowotna (Healthcare)

- Omówienie przypadków użycia IoT w opiece zdrowotnej, podatności, ryzyka i mitygacji.

5.2.1 Osobiste Urządzenia Fitness (Personal Fitness Devices)

- Są jednymi z najpopularniejszych komercyjnych produktów IoT.
- Komunikacja:
  - Niektóre komunikują się z aplikacją chmurową.
  - Niektóre używają połączenia Bluetooth do telefonu, a następnie połączenia danych komórkowych lub Wi-Fi do internetu i chmury.
- Formy: zegarki na rękę, opaski na głowę, kaski, słuchawki.
- Komponenty: Zazwyczaj składają się z czujnika tętna i akcelerometru (do wykrywania ruchu w formie kroków).
- Funkcjonalność chmury: Umożliwia przechowywanie danych fitness, pulpit analityczny i szeroki zakres ustawień konfiguracyjnych.

5.2.2 Monitorowanie Stanu Zdrowia (Healthcare Monitoring)

- Jedna z funkcji urządzeń IoT w opiece zdrowotnej.
- Polega na zbieraniu i ocenie danych pacjenta przez pewien okres czasu.
- Zdalne monitorowanie pacjenta w czasie rzeczywistym (**RPM - Remote Patient Monitoring**) jest możliwe dzięki IoT.
- Pacjenci mogą być monitorowani w domu.
- Urządzenia monitorujące noszone przez pacjenta są podłączone do Internetu.
- **Brama (Gateway)** łączy sygnały z czujników i bezpiecznie przesyła dane do chmury.
- System RPM: Pacjent nosi czujniki tworzące sieć czujników ciała (**BSN - Body Sensor Network**). Dane (tętno, tlen, ciśnienie, glukoza, EKG, ruch, oddech itp.) są przesyłane do magazynu danych i analizowane.

5.2.3 IoT w Szpitalu (IoT in the Hospital)

- W jednym pokoju szpitalnym może znajdować się nawet **20 urządzeń medycznych**.
- IoT zapewnia różne funkcjonalności dla podłączonych urządzeń medycznych:
  - **Monitorowanie i przesyłanie danych pacjenta.**
  - **Śledzenie lokalizacji** urządzeń IoT i monitorowanie ich działania w celu wykrywania problemów i zapobiegania awariom.
  - **Urządzenia terapeutyczne** używają aktuatorów kontrolowanych przez oprogramowanie do regulacji podawania leków, płynów i tlenu.
- IoT zwiększa efektywność operacji, ale stwarza również **wyzwania** dla działów IT i specjalistów ds. bezpieczeństwa danych.

5.2.4 Przykład Ataku: Hakowanie Rozrusznika Serca (Hacking a Pacemaker)

- Podatności bezpieczeństwa znaleziono w podłączonych urządzeniach medycznych, takich jak pompy infuzyjne i insulinowe, defibrylatory z Bluetooth, lodówki do przechowywania leków i krwi, oraz wiele innych.
- Sierpień 2017: Amerykańska Agencja Żywności i Leków (**FDA**) zatwierdziła aktualizację oprogramowania, która łatała lukę bezpieczeństwa w implantowanych rozrusznikach serca z obsługą częstotliwości radiowej (RF).
- Rozruszniki zawierają wbudowany mikroprocesor i firmware, które są **podatne na zdalne ataki przez RF**.
- Aktualizacja firmware mogła być wykonana przez RF **bez konieczności usuwania lub wymiany urządzenia**.
- Szacuje się, że problem dotyczył **465 000 urządzeń**.

5.2.5 Podatności w Opiece Zdrowotnej (Healthcare Vulnerabilities)

- Podatności sieci opieki zdrowotnej obejmują:
  - Słabe lub nieistniejące uwierzytelnianie.
  - Niezabezpieczone wbudowane procesy serwerowe.
  - Niepotrzebnie podatne aplikacje, które mogą być skompromitowane z powodu błędu użytkownika.
- Personel medyczny może używać klientów webowych działających na urządzeniach i systemach medycznych do przeglądania internetu i czytania e-maili, czyniąc je podatnymi na te same ataki co zwykły komputer.
- Wiele urządzeń medycznych ma **długi okres użytkowania**, a komputery używane do ich obsługi działają na **starych i nieaktualizowanych systemach operacyjnych**.
- Urządzenia medyczne są **słabo regulowane** i często nie są projektowane zgodnie ze standardami bezpieczeństwa sprzętu i oprogramowania.

5.2.6 Ryzyka w Opiece Zdrowotnej (Healthcare Risks)

- Podatności w podłączonych urządzeniach medycznych prowadzą do wielu ryzyk:
  - Mogą być manipulowane, przerywane lub wyłączane, co skutkuje **obrażeniami lub śmiercią pacjenta**.
- Słabe bezpieczeństwo urządzeń może pozwolić atakującemu na dostęp do danych przechowywanych na podłączonym urządzeniu medycznym lub urządzenie może zapewnić dostęp do danych przechowywanych w sieci.
- Dane osobowe umożliwiające identyfikację (**PII - Personally-Identifiable Information**) pacjentów mogą być **skradzione lub zmanipulowane**.
- Regulacje rządowe dotyczące postępowania z PII mogą skutkować **surowymi karami** dla organizacji opieki zdrowotnej.

6 Podsumowanie

- Urządzenia IoT są coraz częściej kompromitowane i wykorzystywane w atakach z powodu braku podstawowych zabezpieczeń (silne hasła, aktualne OS, segmentacja sieci).
- Konwergencja IT i OT umożliwia menedżerom operacyjnym wykorzystanie narzędzi IT do podejmowania decyzji w czasie rzeczywistym na podstawie danych operacyjnych, a zespołom IT do innowacji (np. optymalizacja łańcucha dostaw).
- Inteligentny dom jest przykładem transformacji życia codziennego przez IoT.
- Aplikacja chmurowa (np. dla urządzeń fitness) umożliwia przechowywanie danych, analizę i konfigurację.
- Pacjenci mogą być monitorowani w domu za pomocą urządzeń noszonych (BSN) podłączonych do internetu.
- Wykorzystanie IoT w opiece zdrowotnej zwiększa efektywność, ale stwarza wyzwania dla IT i bezpieczeństwa danych (przykład: podatne rozruszniki serca).

7 Modele Sieciowe

7.1 Modele OSI i TCP/IP

- **Modele warstwowe** są używane do ilustrowania, jak odbywa się komunikacja danych od końca do końca (end-to-end).
- **Korzyści** z używania modelu warstwowego do wyjaśniania protokołów i operacji:
  - Pomagają w projektowaniu protokołów.
  - Sprzyjają konkurencji, ponieważ produkty różnych dostawców mogą współpracować.
  - Zapobiegają wpływowi zmian technologii lub możliwości w jednej warstwie na inne warstwy powyżej i poniżej.
  - Zapewniają wspólny język do opisywania funkcji i możliwości sieciowych.
- **OSI (Open Systems Interconnection):** Model referencyjny składający się z 7 warstw (Fizyczna, Łącza Danych, Sieciowa, Transportowa, Sesji, Prezentacji, Aplikacji).

- **TCP/IP (Transport Control Protocol/Internet Protocol):** Praktyczny model używany w Internecie, składający się z 4 warstw (Dostępu do Sieci, Internetowa, Transportowa, Aplikacji). Modele OSI i TCP/IP różnią się strukturą, ale opisują podobne funkcje.

8 Modele IoT

8.1 Model Referencyjny IoT (IoT Reference Model)

Model ten składa się z 7 poziomów (levels):

1. **Urządzenia Fizyczne i Kontrolery (Physical Devices & Controllers):** "Rzeczy"w IoT; szeroka gama urządzeń końcowych, które wysyłają i odbierają informacje.
2. **Łączność (Connectivity):** Komunikacja i jednostki przetwarzające; odpowiedzialne za niezawodną i terminową transmisję danych między urządzeniami a siecią, między sieciami oraz między siecią a przetwarzaniem danych na Poziomie 3.
3. **Przetwarzanie Brzegowe (Edge/Fog Computing):** Analiza elementów danych i transformacja; konwertuje dane na informacje odpowiednie do przechowywania i przetwarzania na wyższym poziomie.
4. **Akumulacja Danych (Data Accumulation):** Przechowywanie; dane w ruchu (data in motion) są konwertowane na dane w spoczynku (data at rest). Dane są również transformowane, aby mogły być konsumowane przez wyższe warstwy.
5. **Abstrakcja Danych (Data Abstraction):** Agregacja i dostęp; skupia się na renderowaniu danych i ich przechowywaniu w sposób umożliwiający rozwój aplikacji.
6. **Aplikacja (Application):** Raportowanie, analityka, kontrola; interpretacja informacji w oparciu o naturę danych urządzenia i potrzeby biznesowe.
7. **Współpraca i Procesy (Collaboration & Processes):** Zaangażowanie ludzi i procesów biznesowych; wykracza poza pojedyncze aplikacje, aby objąć komunikację i współpracę wymaganą między ludźmi a procesami biznesowymi.

8.2 Bezpieczeństwo w Modelu Referencyjnym IoT

- Środki bezpieczeństwa muszą być stosowane na **wszystkich poziomach** modelu.
- Obejmują one:
  - Zabezpieczenie sprzętu i oprogramowania każdego urządzenia lub systemu podłączonego do sieci IoT.
  - Zapewnienie bezpieczeństwa dla wszystkich procesów zachodzących na każdym poziomie sieci.
  - Zabezpieczenie przepływu danych i komunikacji między poszczególnymi poziomami.
- Wizualizacja: Bezpieczeństwo (Security) jako pionowy filar przechodzący przez wszystkie warstwy, łączący Brzeg (Edge) z Centrum (Center).

8.3 Standaryzowana Architektura ETSI M2M

- W 2008 roku Europejski Instytut Norm Telekomunikacyjnych (**ETSI - European Telecommunications Standards Institute**) stworzył architekturę dla komunikacji **maszyna-maszyna (M2M)**.
- Celem modelu jest zapewnienie **wspólnych ram** dla zrozumienia umiejscowienia różnych standardów i protokołów w systemie IoT.
- Model obejmuje **trzy domeny**:
  - **Domena Aplikacji (Application Domain):** Mogą tu występować funkcje zarządzania, takie jak analityka danych, zarządzanie łącznością, zarządzanie inteligentną energią, zarządzanie flotą itp.
  - **Domena Sieci (Network Domain):** Miejsce, gdzie dane opuszczają sieć lokalną i są transportowane do Domeny Aplikacji za pomocą protokołów przewodowych i bezprzewodowych (obejmuje M2M Core i M2M Gateway).
  - **Domena Urządzeń M2M (M2M Device Domain):** Gdzie urządzenia końcowe, takie jak czujniki, aktuatory i kontrolery, łączą się z siecią poprzez bramy M2M (obejmuje urządzenia i sieć obszarową M2M).

8.4 Inne Modele IoT

- **Model Purdue dla Hierarchii Sterowania (Purdue Model for Control Hierarchy):** Używany w przemyśle produkcyjnym, segmentuje urządzenia i sprzęt według funkcji hierarchicznych (Poziom 0: Proces, Poziom 1: Kontrola podstawowa, Poziom 2: Kontrola nadzorcza obszaru, Poziom 3: Operacje produkcyjne i kontrola obiektu, Poziom 4: Planowanie biznesowe i logistyka obiektu, Poziom 5: Sieć korporacyjna). Uwzględnia również Strefę Zdemilitaryzowaną (DMZ) i Strefę Bezpieczeństwa (Safety Zone).
- **Referencyjna Architektura Przemysłowego Internetu (IIRA - Industrial Internet Reference Architecture):** Framework oparty na standardach, używany przez architektów systemów do projektowania systemów przemysłowych.
- **Architektura Internetu Rzeczy (IoT-A - Internet of Things - Architecture):** Bardziej formalnie znana jako Architektoniczny Model Referencyjny (ARM) dla Internetu Rzeczy.

8.5 Prosty Model IoT i Warstwy Bezpieczeństwa

- Uproszczony model IoT z warstwami funkcjonalnymi i warstwami zarządzania danymi.
- **Domeny/Warstwy Funkcjonalne:**
  - **Warstwa Urządzeń (Device):** np. w systemie nawadniania: pojedyncze zraszacze, czujniki wilgotności, czujniki temperatury, aktuatory.
  - **Warstwa Komunikacji (Communication):** Urządzenia mogą być podłączone do lokalnego panelu sterowania nawadnianiem, który monitoruje stan systemu.
  - **Warstwa Aplikacji (Application):** Panel sterowania może być podłączony do zdalnego centrum danych, gdzie agregowane są panele sterowania z wielu systemów nawadniania.
- **Zarządzanie Danymi** - Gdzie i kiedy dane są przetwarzane:
  - **Warstwa Mgły Rozproszonej (Mist layer):** Blisko ziemi, gdzie rzeczy są podłączone do sieci. Minimalne przetwarzanie.

- **Warstwa Mgły (Fog layer):** Na lokalnym urządzeniu o większej mocy, np. panelu sterowania systemu nawadniania. Przetwarzanie bliżej źródła danych.
- **Chmura (Cloud):** Zdalne centrum danych; umożliwia np. zdalne nadpisanie autonomicznych działań panelu sterowania przez nadzorcę za pomocą aplikacji mobilnej lub desktopowej.

8.6 Model Bezpieczeństwa IoT (zorientowany na protokoły)

- Ten kurs wykorzystuje kombinację warstw funkcjonalnych uproszczonego modelu IoT nałożonych na model TCP/IP.
- Mapowanie protokołów na warstwy funkcjonalne:
  - **Aplikacja (Application):** ZigBee (część aplikacyjna), HTTP/HTTPS, MQTT, CoAP.
  - **Komunikacja (Communication):** Thread, TCP, UDP, RPL, IPv6 (jako warstwy transportowe i sieciowe).
  - **Urządzenie (Device):** 6LoWPAN, IEEE 802.15.4, Bluetooth Low Energy (BLE), Wi-Fi, Near Field Communication (NFC), Cellular (jako warstwy dostępu do sieci i fizyczne).

9 Ramy Kompetencji Cybersecurity (NICE) a Systemy IoT

9.1 NICE Cybersecurity Workforce Framework

- Narodowy Instytut Standardów i Technologii (**NIST**) opublikował Narodową Inicjatywę na rzecz Edukacji w zakresie Cyberbezpieczeństwa (**NICE**) Cybersecurity Workforce Framework w listopadzie 2017.
- Doskonałe źródło informacji o identyfikacji, rekrutacji, rozwoju i zatrzymywaniu talentów w dziedzinie cyberbezpieczeństwa.
- Publikacja definiuje **role zawodowe**, które obejmują wymaganą wiedzę, umiejętności i zdolności (**KSAs - Knowledge, Skills, and Abilities**) oraz zadania wykonywane przez osobę na danej roli.
- Role zawodowe są podzielone na **siedem kategorii**. W kontekście tego kursu interesują nas kategorie: **Bezpieczne Dostarczanie (Securely Provision)** oraz **Ochrona i Obrona (Protect and Defend)**.
- Pozostałe kategorie: Nadzór i Zarządzanie (Oversee and Govern), Obsługa i Utrzymanie (Operate and Maintain), Badanie (Investigate), Zbieranie i Operowanie (Collect and Operate), Analiza (Analyze).

9.2 Kategoria: Bezpieczne Dostarczanie (Securely Provision)

- Role zawodowe w tej kategorii są odpowiedzialne za **konceptualizację, projektowanie, pozyskiwanie i wdrażanie** bezpiecznych systemów technologii informacyjnej (IT).
- W tym kursie skupiamy się na obszarze specjalizacji **Zarządzanie Ryzykiem (Risk Management)**.
- Obejmuje wszystkie procesy niezbędne do zapewnienia, że istniejące i nowe systemy IT spełniają wymagania organizacji dotyczące cyberbezpieczeństwa i ryzyka.

- Rola: **Oceniający Kontrole Bezpieczeństwa (Security Control Assessor):** Osoby na tej roli przeprowadzają kompleksowe oceny zarządczych, operacyjnych i technicznych kontroli bezpieczeństwa w celu określenia ich ogólnej skuteczności.

9.3 Kategoria: Ochrona i Obrona (Protect and Defend)

- Role zawodowe w tej kategorii zajmują się **identyfikowaniem, analizowaniem i mitygowaniem zagrożeń** dla systemów IT.
- W tym kursie skupiamy się na obszarze specjalizacji **Ocena i Zarządzanie Podatnościami (Vulnerability Assessment and Management)**.
- Obejmuje przeprowadzanie ocen zagrożeń i podatności; określanie odchyleń od akceptowalnych konfiguracji lub polityk; ocenę poziomu ryzyka; oraz opracowywanie lub rekomendowanie odpowiednich środków zaradczych (mitygacji).
- Rola: **Analityk Oceny Podatności (Vulnerability Assessment Analyst):** Przeprowadza oceny systemów IT i identyfikuje miejsca, w których systemy te odbiegają od akceptowalnych konfiguracji lub polityk.

10 Analiza Modelu Zagrożeń dla Systemu IoT

10.1 Wprowadzenie do Modelowania Zagrożeń

- **Modelowanie zagrożeń** to narzędzie używane do przeprowadzania zadań związanych z zarządzaniem ryzykiem i oceną podatności.
- Jest to **ustrukturyzowane podejście** do analizy bezpieczeństwa i podatności systemu, niezależnie od tego, czy systemem jest sprzęt urządzenia, oprogramowanie, czy sieci używane do komunikacji z innymi urządzeniami.
- Ten kurs wykorzystuje **adaptację Analizy Modelu Zagrożeń Microsoftu** i stosuje ją do systemu IoT.

10.2 Pięcioetapowy Proces Modelowania Zagrożeń

10.2.1 Krok 1: Identyfikacja Celów Bezpieczeństwa (Identify Security Objectives)

Należy użyć następujących kategorii do określenia celów bezpieczeństwa dla systemu IoT:

- **Tożsamość (Identity):** Dokumentowanie kontroli zapewniających zbieranie dowodów na tożsamość użytkowników uzyskujących dostęp i korzystających z systemu IoT.
- **Finanse (Financial):** Dokumentowanie ryzyk finansowych związanych z różnymi aspektami systemu IoT, aby zarząd mógł określić akceptowalny poziom ryzyka (np. ryzyko utraty kontrolera vs. czujnika).
- **Reputacja (Reputation):** Dokumentowanie możliwego wpływu na reputację organizacji w przypadku ataku na system IoT (np. reputacja firmy sprzedającej kamery internetowe po ataku DDoS z ich udziałem).
- **Prywatność i Regulacje (Privacy and Regulation):** Dokumentowanie wpływu obaw dotyczących prywatności i wymagań regulacyjnych (np. dane z czujnika temperatury w systemie nawadniania mogą nie podlegać takim obawom).

- **Gwarancje Dostępności (Availability Guarantees):** Dokumentowanie oczekiwanej dostępności i gwarantowanego czasu działania systemu IoT (np. tolerancja na przestoje w systemie ICS może być bardzo niska).
- **Bezpieczeństwo Fizyczne (Safety):** Dokumentowanie potencjalnego wpływu na dobrostan fizyczny ludzi oraz fizyczne uszkodzenia sprzętu i obiektów.

10.2.2 Krok 2: Dokumentowanie Architektury Systemu IoT (Document the IoT System Architecture)

Należy stworzyć dokumenty opisujące architekturę systemu IoT, w tym:

- Komponenty systemu IoT na warstwach aplikacji, komunikacji i urządzeń.
- Przepływ danych między komponentami i między warstwami.
- Technologie, protokoły i standardy użyte do implementacji systemu IoT.

10.2.3 Krok 3: Dekompozycja Systemu IoT (Decompose the IoT System)

- Zagłębienie się w poszczególne komponenty i funkcje, które wpływają na cele bezpieczeństwa systemu IoT.
- Stworzenie profilu bezpieczeństwa, który pomoże zidentyfikować zagrożenia i podatności w projekcie, implementacji i wdrożeniu systemu IoT.
- Podczas tego kroku należy zebrać informacje o systemie IoT, wykonując następujące zadania:
  - Zidentyfikować **granice zaufania** między zaufanymi a niezaufanymi komponentami.
  - Zidentyfikować **przepływ danych** między urządzeniami, siecią komunikacyjną i aplikacjami.
  - Zidentyfikować **punkty wejścia**, gdzie dane są wprowadzane do systemu.
  - Zidentyfikować **dane wrażliwe** w systemie IoT, gdzie przechowywane i manipulowane są bezpieczne zasoby.
  - Udokumentować **profil bezpieczeństwa**, aby uwzględnić podejścia do walidacji danych wejściowych, uwierzytelniania, autoryzacji, konfiguracji i wszelkich innych obszarów systemu IoT, które są podatne na zagrożenia.

10.2.4 Krok 4: Identyfikacja i Ocena Zagrożeń (Identify and Rate Threats)

- **Zagrożenie (Threat):** Potencjalne niebezpieczeństwo dla dowolnego zasobu, takiego jak dane lub komponenty systemu IoT.
- **Aktor Zagrożenia (Threat Actor):** Osoby lub podmioty wykorzystujące podatności.
- **Podatność (Vulnerability):** Słabość w systemie IoT lub jego projekcie, która może zostać wykorzystana przez zagrożenie.
- **Powierzchnia Ataku (Attack Surface):** Kombinacja podatności; opisuje różne punkty, w których aktor zagrożenia może dostać się do systemu i skąd może wydobyć dane.
- W tym kursie używane są dwa narzędzia do identyfikacji i oceny zagrożeń i podatności:

- **STRIDE:** Narzędzie do oceny podatności używane do identyfikacji zagrożeń. Akronim kategorii: **S**poofing Identity (Podszywanie się pod tożsamość), **T**ampering with Data (Manipulacja danymi), **R**epudiation (Zaprzeczenie), **I**nformation Disclosure (Ujawnienie informacji), **D**enial of Service (Odmowa usługi), **E**levation of Privilege (Podniesienie uprawnień).
- **DREAD:** Narzędzie do oceny ryzyka używane do oceny zagrożeń odkrytych w procesie STRIDE. Akronim zmiennych używanych do kwantyfikacji, porównania i priorytetyzacji ryzyka: **D**amage (Potencjał szkód), **R**eproducibility (Odtwarzalność), **E**xploitability (Podatność na wykorzystanie), **A**ffected Users (Dotknięci użytkownicy), **D**iscoverability (Wykrywalność).
- Formuła oceny ryzyka DREAD: **Ocena Ryzyka DREAD = (D + R + E + A + D) / 5**

10.2.5 Krok 5: Rekomendowanie Technik i Technologii Mitygacji (Recommend Mitigations Techniques and Technologies)

- Po zidentyfikowaniu i ocenie zagrożeń, należy określić techniki mitygacji dla każdego zagrożenia i wybrać najbardziej odpowiednią technologię, która zredukuje lub wyeliminuje zagrożenie.
- Podczas tej oceny należy pamiętać o tym, co ma sens z **perspektywy biznesowej**, w tym o istniejących politykach w organizacji.

11 Podsumowanie

- Korzyści z używania modelu warstwowego: pomoc w projektowaniu protokołów, wspieranie konkurencji, izolacja zmian między warstwami, wspólny język.
- Celem modelu referencyjnego IoT jest zapewnienie wspólnej terminologii i wyjaśnienie przepływu i przetwarzania informacji dla zunifikowanej branży IoT.
- Bezpieczeństwo musi przenikać wszystkie poziomy modelu referencyjnego IoT.
- Model ETSI obejmuje trzy domeny: urządzenia M2M, sieci i aplikacji.
- Inne modele IoT to Model Purdue, IIRA i IoT-A.
- Warstwy bezpieczeństwa IoT w uproszczonym modelu to: urządzenia, sieci i aplikacji. Nakładanie warstw na TCP/IP pomaga zrozumieć umiejscowienie protokołów.
- Publikacja NICE NIST to doskonałe źródło wiedzy o identyfikacji, rekrutacji, rozwoju i zatrzymywaniu talentów w cyberbezpieczeństwie.
  - Kategoria Robocza - Zarządzanie Ryzykiem: obejmuje procesy zapewniające spełnienie wymagań cyberbezpieczeństwa i ryzyka.
  - Kategoria Robocza - Ocena i Zarządzanie Podatnościami: obejmuje oceny zagrożeń/podatności, identyfikację odchyleń, ocenę ryzyka i rekomendowanie mitygacji.
- Analiza modelu zagrożeń: Krok 1 - identyfikacja celów bezpieczeństwa, Krok 2 - dokumentowanie architektury systemu IoT, Krok 3 - dekompozycja systemu IoT, Krok 4 - identyfikacja i ocena zagrożeń (STRIDE, DREAD), Krok 5 - rekomendowanie mitygacji.

12 Komponenty Sprzętowe Urządzeń IoT

12.1 Podatności Sprzętowe wg OWASP

Open Web Application Security Project (OWASP) wymienia podatności dla różnych powierzchni ataku sprzętowego:

- **Czujniki Sprzętowe (Hardware Sensors):**
  - Manipulacja środowiskiem (Environment manipulation)
  - Manipulacja fizyczna (Tampering)
  - Uszkodzenie (Damage)
- **Pamięć Urządzenia (Device Memory):**
  - Domyślna nazwa użytkownika i hasło
  - Dane wrażliwe (Sensitive data)
  - Nazwy użytkowników i hasła w postaci jawnego tekstu (Plaintext usernames and passwords)
  - Klucze szyfrujące (Encryption keys)
- **Fizyczne Interfejsy Urządzenia (Device Physical Interfaces):**
  - Usunięcie nośnika pamięci (Removal of storage media)
  - Reset do stanu niezabezpieczonego (Reset to insecure state)
  - ID urządzenia/Numer seryjny (Device ID/Serial number)
  - Połączenia interfejsu szeregowego (Serial interface connections)
  - Dostęp użytkownika i administracyjny (User and Administrative access)
  - Eskalacja uprawnień (Privilege escalation)
- **Oprogramowanie Układowe Urządzenia (Device Firmware):**
  - Konta tylnych drzwi (Backdoor Accounts)
  - Zaszyte na stałe poświadczenia (Hardcoded credentials)
  - Klucze szyfrujące (Encryption keys)
  - Wyświetlanie wersji firmware (Firmware version display)
  - Data ostatniej aktualizacji firmware (Firmware version last update date)
  - Podatne usługi (Vulnerable services)
  - Ekspozycja API funkcji związanych z bezpieczeństwem (Security related function API exposure)
- **Mechanizm Aktualizacji Firmware (Firmware Update Mechanism):**
  - Aktualizacja wysyłana bez szyfrowania
  - Aktualizacje niepodpisane (Updates not signed)
  - Lokalizacja aktualizacji możliwa do zapisu (Update location writable)
  - Weryfikacja i uwierzytelnienie aktualizacji (Update verification and authentication)
  - Złośliwa aktualizacja (Malicious update)
  - Brakujący mechanizm aktualizacji (Missing update mechanism)
  - Brak mechanizmu ręcznej aktualizacji (No manual update mechanism)

12.2 Urządzenia o Ograniczonych Zasobach (Constrained Devices)

- IoT składa się z urządzeń o ograniczonych zasobach, które zazwyczaj mają bardzo **ograniczoną moc, pamięć i cykle przetwarzania**.
- Możliwości komunikacyjne są ograniczone i jest mało prawdopodobne, że zaimplementowano szyfrowanie (jedna z podatności OWASP).
- **Klasy urządzeń o ograniczonych zasobach:**

Nazwa	Rozmiar Danych (RAM)	Rozmiar Kodu (Flash)
Klasa 0, C0	$\leq$ 10 Kilobajtów	$\leq$ 100 Kilobajtów
Klasa 1, C1	$\approx$ 10 Kilobajtów	$\approx$ 100 Kilobajtów
Klasa 2, C2	$\approx$ 50 Kilobajtów	$\approx$ 250 Kilobajtów

- **Urządzenia o ograniczonych zasobach w tym kursie:**
  - **Inteligentne Czujniki (Smart Sensors):** Centrum urządzeń IoT. Zdolne do komunikacji z systemem monitorującym za pomocą mikroprocesora i mają zdolność do autodiagnostyki problemów.
  - **Urządzenia Wbudowane (Embedded Devices):** Zawierają system komputerowy zaprojektowany do specjalnego celu, zwykle do uruchamiania jednej aplikacji. Produkty mogą zapewniać łączność internetową i są uważane za inteligentne.
  - **Prototypowanie (Prototyping):** Raspberry Pi i Arduino to urządzenia prototypowe dla systemów wbudowanych. Raspberry Pi potrzebuje pełnego systemu operacyjnego. Arduino to jednopłytkowy mikrokontroler, który można skonfigurować, pisząc kod programu, aby instruować go do wykonywania różnych funkcji. Program jest następnie kompilowany i wysyłany do nieulotnej pamięci flash Arduino.

12.3 Typy Procesorów (CPU) w IoT

- Główne typy CPU używane w IoT to **ARM, MIPS i x86**.
- Dwie kategorie: **RISC (Reduced Instruction Set Computing)** i **CISC (Complex Instruction Set Computing)**.
- **Procesory RISC:**
  - Mają mniej tranzystorów niż procesory CISC.
  - Dominują na rynku komputerów mobilnych.
  - Mniejsza liczba tranzystorów przekłada się na niższy koszt, mniejsze zużycie energii i mniejszą produkcję ciepła.
  - Dwaj główni dostawcy procesorów RISC:
    - **ARM (Advanced RISC Machine):** Architektura zazwyczaj licencjonowana innym firmom do projektowania własnych procesorów. Architektury 32-bitowe i 64-bitowe. Raspberry Pi jest procesorem ARM.
    - **MIPS (Microprocessor without Interlocked Pipeline Stages):** Używany w wielu procesorach w systemach wbudowanych, a także w urządzeniach sieciowych, mobilnych i IoT. Implementacje 32-bitowe i 64-bitowe.

- **Procesory CISC:**
  - Zdolność do wykonywania kilku operacji za pomocą jednej instrukcji.
  - Więcej tranzystorów jest potrzebnych do przechowywania bardziej złożonych instrukcji, co generuje więcej ciepła, wymaga więcej mocy i zwiększa koszt procesora.
  - Użycie złożonych instrukcji zmniejsza rozmiar kodu programu, co stanowi korzyść przy ładowaniu i przechowywaniu aplikacji.
  - Główni dostawcy to **Intel** i **Advanced Micro Devices (AMD)**. Obie firmy aktywnie wchodzą na rynek IoT, próbując zmniejszyć zużycie energii i ciepło w swoich procesorach.
- **Obliczenia Heterogeniczne (Heterogeneous Computing):**
  - Polega na użyciu więcej niż jednego rodzaju procesora o różnych możliwościach.
  - Powszechnym podejściem jest wykorzystanie **GPU (Graphics Processing Unit)** do wykonywania złożonych obliczeń matematycznych lub obsługi zadań szyfrowania i deszyfrowania.
- **Obliczenia big.LITTLE (ARM):**
  - Technologia ARM wykorzystująca procesory (rdzenie) o różnych możliwościach przetwarzania i wymaganiach energetycznych.
  - "Duży"procesor zapewnia największą wydajność obliczeniową i ma wyższe wymagania energetyczne.
  - Użycie big.LITTLE może wydłużyć żywotność baterii w urządzeniach znajdujących się w odległych lokalizacjach.

12.4 Pamięć (Memory)

- Urządzenia IoT mają różne typy pamięci używane do przechowywania danych, oprogramowania układowego i przetwarzania.
- **Typowe typy pamięci i ich zastosowania:**
  - **Karta SD (SD Card):** Używana do przechowywania danych niezbędnych do działania IoT lub do przechowywania zebranych danych. Musi być chroniona przed usunięciem.
  - **Pamięć Nieulotna (Non-Volatile Memory):** EPROM (kasowalna programowalna pamięć tylko do odczytu) i EEPROM (elektrycznie kasowalna programowalna pamięć tylko do odczytu). Zachowują przechowywane informacje nawet po wyłączeniu zasilania. Używane do przechowywania firmware, bootloadera i innych krytycznych informacji wymaganych do działania urządzenia IoT. Atakujący może być w stanie odczytać komunikację między pamięcią a mikrokontrolerem.
  - **Pamięć Ulotna (Volatile Memory):** SRAM (statyczna pamięć o dostępie swobodnym) i DRAM (dynamiczna pamięć o dostępie swobodnym). Używane do przechowywania kodu operacyjnego i zapewnienia tymczasowego magazynu podczas działania urządzenia. Po wyłączeniu urządzenia wszystkie dane w pamięci są tracone.

12.5 Porty Fizyczne (Physical Ports)

- Urządzenia IoT mogą mieć porty takie jak **USB** i **Ethernet**.
  - Należy stosować standardowe procedury ochrony portów USB i Ethernet.

  - Porty te mogą być używane do ekstrakcji informacji z urządzenia poprzez podłączenie innego systemu komputerowego.

- **Inne dostępne porty komunikacyjne:**
  - **UART (Universal Asynchronous Receiver-Transmitter):** Interfejs do komunikacji z innymi urządzeniami peryferyjnymi. Zazwyczaj trzy piny: Tx (Transmit), Rx (Receive) i Ground. Istnieje możliwość, że dane są przesyłane na tych pinach, dając atakującemu możliwość ich przechwycenia. Gdy piny nie są potrzebne, powinny być wyłączone w konfiguracji, jeśli to możliwe.
  - **I2C (Inter-Integrated Circuit):** Protokół danych szeregowych używany do komunikacji na krótkie dystanse, często między układami na tej samej płycie. Może być używany do komunikacji między mikrokontrolerem a układami EEPROM. Atakujący może potencjalnie uszkodzić lub wyodrębnić przesyłane dane.
  - **SPI (Serial Peripheral Interface):** Protokół szeregowy na krótkie dystanse używający czteroprzewodowej magistrali szeregowej. Umożliwia podłączenie wielu urządzeń do tych samych przewodów i obsługuje komunikację w pełnym dupleksie. Używany do komunikacji z urządzeniami na tej samej płycie lub z EEPROM, flash, itp. znajdującymi się w odległości kilku stóp. Ekstrakcja wrażliwych informacji jest bardzo realną możliwością.
  - **JTAG (Joint Test Action Group):** Nie jest protokołem komunikacyjnym, lecz protokołem do testowania i debugowania. Zapewnienie dostępu do portu JTAG może pozwolić atakującemu na inżynierię wsteczną logiki mikrokontrolera. Atakujący może również wyodrębnić firmware i potencjalnie załadować złośliwe oprogramowanie układowe.

13 Komponenty Programowe Urządzeń IoT

13.1 Systemy Wbudowane (Embedded Systems)

- Systemy wbudowane są zaprojektowane do **określonej funkcji** w ramach większego systemu.
- **Przykład: Urządzenia bezpieczeństwa domowego**
  - Wszystkie operacje są kontrolowane przez mikrokontroler zaprojektowany specjalnie do tego celu.
  - Mikrokontroler może być zaprogramowany dla czujników unikalnych dla danej instalacji.
  - Czujniki mogą obejmować czujniki dymu, ruchu, gazu i temperatury, które dostarczają dane do mikrokontrolera, który uruchomi alarm, jeśli coś przekroczy ustawione progi.
  - Mikrokontroler może mieć możliwość wyświetlania informacji na ekranie lub komunikowania się z innym sprzętem komputerowym w celu monitorowania.
- Niektóre systemy wbudowane używają mikroprocesorów. Mikroprocesor i mikrokontroler mogą mieć ten sam wbudowany CPU.
- System oparty na mikrokontrolerze jest **samowystarczalny** i może zawierać pamięć flash, RAM, komunikację szeregową i inne peryferia w ramach zintegrowanego układu.
- Systemy wbudowane mogą używać **wbudowanego systemu operacyjnego** lub być programowane bezpośrednio za pomocą **kodu maszynowego** dla CPU. Czasami używane są okrojone wersje Linuksa.
- **Debugowanie** programowania dla systemów wbudowanych różni się od typowego debugowania oprogramowania PC.

- Na PC oprogramowanie jest tworzone na tym samym procesorze, na którym będzie działać program. Umożliwia to korzystanie z wbudowanych narzędzi w środowisku programistycznym do debugowania.
- W systemie wbudowanym oprogramowanie jest tworzone poza środowiskiem, w którym będzie działać. Aby debugować oprogramowanie wbudowane, systemy wykorzystują **port JTAG** do śledzenia problemów z oprogramowaniem.

13.2 Kod Kompilowany vs. Interpretowany (Compiled or Interpreted Code)

- Programiści mają wybór co do środowiska podczas tworzenia oprogramowania aplikacyjnego.
  - Dostępne są środowiska **emulacji** na PC do tworzenia aplikacji przeznaczonych na inne platformy. Dla urządzeń mobilnych dostępny jest emulator, dzięki czemu układ ekranu odzwierciedla wygląd i działanie aplikacji na urządzeniu mobilnym.
- Programiści mają również wybór co do **typu języka programowania**.
  - **Kod Kompilowany:**
    - Kod źródłowy jest pisany w formacie czytelnym za pomocą edytora tekstu, a następnie konwertowany (kompilowany) na kod maszynowy, który jest odczytywany i wykonywany przez procesor.
    - Programista musi zakończyć proces kompilacji, zanim program będzie użyteczny. W razie potrzeby zmian, kod tekstowy jest zmieniany, a następnie ponownie kompilowany.
    - Przykłady: C, C++, Rust, Visual Basic.
  - **Kod Interpretowany:**
    - Każda instrukcja jest wykonywana jedna po drugiej. Interpreter tłumaczy instrukcję na formę kodu maszynowego, który może być wykonany przez procesor. Jeśli wystąpi błąd, program zatrzyma się w tym punkcie i można dokonać poprawek.
    - Przykłady: Python, JavaScript, Perl, PHP.
- W zależności od używanego systemu operacyjnego, możliwe jest również używanie **skryptów** do wykonywania różnych zadań (np. skrypty powłoki Linux, PowerShell Windows; oba działają w trybie interpretowanym).
- Kod interpretowany jest **łatwy do modyfikacji** przez atakującego, ponieważ jest przechowywany w formacie tekstowym.
- Kod kompilowany może zostać zmieniony przez atakującego za pomocą debuggera i zastąpienia instrukcji kodu maszynowego złośliwym kodem. W przypadku kodu kompilowanego możliwe jest **cyfrowe podpisanie** pliku wykonywalnego binarnego, aby zweryfikować, że nie został on zmieniony.

13.3 Tryb Debugowania/Rozruchu (Debug/Boot Mode)

- Powszechne jest, że systemy oferują specjalny tryb debugowania/rozruchu na wypadek napotkania problemu podczas uruchamiania.
- Czasami dostęp do trybu debugowania/rozruchu można uzyskać za pomocą kombinacji klawiszy.
- Jest to również możliwe w przypadku, gdy atakujący mają dostęp do płyty urządzenia. Mogą być w stanie użyć **portu JTAG**.

- Podczas pracy w trybie debugowania/rozruchu istnieje możliwość **ominięcia uwierzytelniania**.
- Jeśli atakujący uzyskają dostęp do trybu debugowania/rozruchu, możliwe byłoby dla nich dokonanie innych zmian w systemie lub nawet zainstalowanie **tylnych drzwi (backdoor)**. Zapewniłoby to dostęp do systemu, jeśli system jest dostępny w sieci.

13.4 Typowe Systemy Operacyjne IoT

- Urządzenia IoT zazwyczaj używają **okrojonej wersji** systemu operacyjnego.
- Programiści mogą wybierać spośród opcji **open source** i **komercyjnych**.
- **Busybox:** Open source, używa jądra Linux. Zapewnia zestaw programów, które można wykonać z wiersza poleceń. Programista powinien wyłączyć niepotrzebne programy podczas kompilacji (np. Telnet).
- **Android Embedded:** Lekka wersja Linuksa, używana głównie w urządzeniach mobilnych, ale może być używana w urządzeniach IoT. Zaprojektowany w celu zmniejszenia zużycia energii i współpracuje z popularnymi procesorami używanymi w urządzeniach IoT.
- **Opcje Komercyjne:** Dostępne są produkty takie jak VxWorks, Windows 10 IoT i ARM Mbed.

14 Bezpieczeństwo Sprzętu (Hardware Security)

14.1 Podatności Fizyczne Urządzeń o Ograniczonych Zasobach

- Urządzenia o ograniczonych zasobach są często umieszczane w **odległych lokalizacjach**, gdzie bezpieczeństwo fizyczne może być trudne do wdrożenia.
- Potencjalne podatności fizyczne:
  - Kradzież urządzenia.
  - Uszkodzenie fizyczne urządzenia.
  - Wyłączenie urządzenia, usunięcie źródła zasilania.
  - Wyłączenie komunikacji, odłączenie kabli lub inne środki zakłócenia.
- **Środki zaradcze:**
  - Zapewnij jakiś rodzaj nadzoru wideo, jeśli to możliwe.
  - Zapewnij obudowę odporną na manipulacje (tamper proof / tamper resistant).

14.2 Bezpieczeństwo Urządzeń Fizycznych

- Urządzenie takie jak czujnik może zostać przesunięte, co spowoduje utratę kalibracji.
- Wiele inteligentnych czujników ma możliwość wywołania alarmu, gdy nie są prawidłowo ustawione.
- Urządzenia z pamięcią masową, takie jak karta SD, mogą potencjalnie mieć skradzione lub zniszczone dane przez atakującego.
- Standardowe protokoły nadzoru i bezpieczeństwa powinny być wdrożone jako pierwsza warstwa obrony.

- Bezpieczeństwo fizyczne urządzenia obejmuje również zapewnienie, że zawsze masz dostęp do urządzenia. Rozważ wdrożenie **zasilania awaryjnego (baterii)** dla urządzeń IoT na wypadek przerw w dostawie prądu.

14.3 Podatności Sprzętowe - Przykłady

- Podatności sprzętowe są bardzo powszechne w wielu urządzeniach IoT.
- Artykuł w magazynie Wired (sierpień 2017) opisywał, jak uzyskać dostęp do firmware na licznych urządzeniach IoT za pomocą pamięci flash eMMC i czytnika kart SD za 10 USD.
  - Poprzez przylutowanie pięciu przewodów do układu flash eMMC i użycie standardowego czytnika kart SD, atakujący był w stanie odzyskać firmware, system operacyjny i oprogramowanie z układu i zapisać je na PC. Po skopiowaniu oprogramowania można je zbadać pod kątem podatności kodu. Takie ataki ilustrują potrzebę wdrożenia bezpieczeństwa fizycznego urządzeń.
- **Inne przykłady podatności sprzętowych:**
  - Dostęp do powłoki (shell) przez połączenie UART do dzwonka IoT.
  - Hack UART - inteligentna lodówka zapewniała dostęp do powłoki roota po ponownym uruchomieniu systemu.
- Inne urządzenia z wykorzystanymi podatnościami sprzętowymi: odtwarzacze Blu-Ray, kamery, urządzenia automatyki domowej, odtwarzacze multimedialne/muzyczne, urządzenia NAS, drukarki, telewizory, sprzęt VoIP, urządzenia medyczne, urządzenia sieciowe, urządzenia Android TV.

15 Podatności Firmware

15.1 Wprowadzenie

- Urządzenia IoT wymagają do działania oprogramowania układowego (firmware).
- Firmware to zasadniczo **oprogramowanie wbudowane**, które zawiera minimalny system operacyjny i powiązane programy do sterowania urządzeniem IoT.
- Firmware urządzeń IoT może zawierać **podatności bezpieczeństwa**, które są odkrywane po ich wydaniu. Podatności związane z firmware dla urządzeń IoT są podobne do tych w innych komputerach lub urządzeniach sieciowych.

15.2 Typowe Podatności Firmware

- **Domyślne Poświadczenia Logowania (Default Login Credentials):**
  - Większość ataków IoT ma miejsce, ponieważ domyślne poświadczenia logowania nie zostały zmienione.
  - Ważne jest, aby nazwy użytkowników i hasła zostały zmienione, aby spełniały silne kryteria przed podłączeniem jakiegokolwiek urządzenia IoT do internetu.
- **Ataki Rozproszonej Odmowy Usługi (DDoS - Distributed Denial of Service):**
  - Ataki DDoS wymagają botnetów zainfekowanych systemów z całego świata.

  - Gdy znane są słabe informacje logowania, haker może napisać zautomatyzowany skrypt do logowania się do zdalnych urządzeń IoT i kopiowania zainfekowanego oprogramowania do ich systemów.

- **Nieaktualne Oprogramowanie Układowe (Out-of-Date Firmware):**
  - Jeśli haker próbuje zaatakować określone urządzenie IoT lub grupę urządzeń, zazwyczaj sprawdzi, czy oprogramowanie układowe jest nieaktualne lub poszuka jakichkolwiek exploitów, które nie zostały jeszcze zaadresowane za pomocą poprawki.

- **Ataki Przepełnienia Bufora (Buffer Overflow Attacks):**
  - Mogą wystąpić w podatnym oprogramowaniu, gdy programista nie uwzględni rozmiaru danych wejściowych, które użytkownik może wprowadzić.
  - Atak przepełnienia bufora może spowodować uszkodzenie danych, odmowę usługi lub umożliwić uruchomienie złośliwego kodu w systemie docelowym.

- **Instalacja Tylnych Drzwi (Backdoor Installation):**
  - Instalacja backdoor zwykle następuje po uzyskaniu przez atakującego zdalnego dostępu do urządzenia IoT.
  - W systemie operacyjnym opartym na Linuksie atakujący może uruchomić polecenie `netcat` w tle i wykonywać złośliwe polecenia w tym systemie zdalnie z dowolnego miejsca na świecie.
  - Dodatkowo, narzędzia diagnostyczne i testowe sieci są czasami pozostawiane w firmware przez producenta urządzenia IoT. Narzędzia te mogą uczynić urządzenia bardziej podatnymi na wykorzystanie, jeśli nastąpi nieautoryzowane wejście.

15.3 Problemy z Aktualizacją Firmware (Firmware Update Issues)

- Aktualizacja oprogramowania układowego IoT i instalowanie poprawek w celu naprawy podatności bezpieczeństwa są **krytycznymi komponentami** bezpieczeństwa sieci.
- Bezpieczeństwo IoT **nie nadąża** za tempem wzrostu urządzeń IoT.
- W niektórych przypadkach **łatki nie istnieją** dla podatności bezpieczeństwa urządzeń.
- W innych przypadkach urządzenie może **nawet nie być aktualizowalne ani możliwe do załatania**.
- Liczba urządzeń IoT w organizacji może sięgać **tysięcy lub dziesiątek tysięcy**.
  - Instalowanie aktualizacji i poprawek na takiej liczbie urządzeń stanowi własne wyzwania.
  - Ważne jest, aby zweryfikować, czy wszystkie aktualizacje i poprawki pochodzą ze **zweryfikowanego źródła**.
- **Lista kontrolna aktualizacji firmware:** Czy jest aktualizowalne/łatane? Jak obsługiwać dla dużej liczby urządzeń? Czy aktualizacje mogą być zautomatyzowane? Czy aktualizacje są szyfrowane i dostarczane są hashe? Czy aktualizacje są podpisane i zweryfikowane? Czy pochodzą z zaufanego źródła?

15.4 Rozwiązania Aktualizacji Firmware (Firmware Update Solutions)

- Konieczne jest prowadzenie **bazy danych** wszystkich urządzeń IoT i informacji o ich firmware.
- Firmware powinno być aktualizowane **jak najszybciej** po pojawieniu się nowych wersji, ponieważ prawdopodobnie zawierają one poprawki bezpieczeństwa.
- Powinien istnieć **plan regularnego sprawdzania** strony internetowej producenta pod kątem aktualizacji.
- Ważne jest **monitorowanie lub subskrybowanie** usług informujących o podatnościach bezpieczeństwa (np. US-CERT).
- Najlepszym rozwiązaniem jest posiadanie **automatycznego systemu** do aktualizacji firmware i instalowania poprawek bezpieczeństwa na urządzeniach IoT w organizacji. Ważne jest, aby wszelkie aktualizacje lub poprawki firmware były **cyfrowo podpisane i zweryfikowane** przed instalacją.

15.5 Rootowanie Systemu Operacyjnego (Rooting an OS)

- Rootowanie urządzenia IoT - atakujący postępuje zgodnie z procesem, który skutecznie przyznaje mu dostęp roota.
- Dostęp roota zapewnia atakującemu **pełną kontrolę** nad tym urządzeniem.
- Interfejsy **JTAG** i **UART** są powszechnymi wektorami ataku w celu uzyskania dostępu roota do urządzenia.
- Po uzyskaniu dostępu można odczytać pamięć urządzenia i zmodyfikować firmware.
- Po uzyskaniu dostępu do firmware można poszukać podatności i wprowadzić nowe dziury bezpieczeństwa.

16 Koncepcje Kontroli Dostępu do Sieci

16.1 Modele Kontroli Dostępu (Access Control Models)

- Analityk bezpieczeństwa powinien znać różne podstawowe modele kontroli dostępu, aby lepiej rozumieć, jak atakujący mogą przełamać kontrole dostępu.
- **Mandatory Access Control (MAC):** Stosuje najściślejszą kontrolę dostępu, typowo używana w wojsku lub aplikacjach krytycznych. Przypisuje etykiety poziomu bezpieczeństwa do informacji i zapewnia użytkownikom dostęp na podstawie ich poświadczeń poziomu bezpieczeństwa.
- **Discretionary Access Control (DAC):** Pozwala użytkownikom kontrolować dostęp do ich danych jako właściciele tych danych.
- **Non-Discretionary Access Control (aka RBAC - Role-Based Access Control):** Decyzje o dostępie opierają się na rolach i obowiązkach jednostki w organizacji.
- **Attribute-Based Access Control (ABAC):** Pozwala na dostęp na podstawie atrybutów obiektu (zasobu), do którego uzyskuje się dostęp, podmiotu (użytkownika) uzyskującego dostęp do zasobu oraz czynników środowiskowych dotyczących sposobu uzyskiwania dostępu do obiektu, takich jak pora dnia.

- **Zasada Najmniejszych Uprawnień (Principle of Least Privilege):** Użytkownikom należy przyznawać minimalną ilość dostępu wymaganą do wykonywania ich funkcji zawodowych.
- **Exploit Eskalacji Uprawnień (Privilege Escalation Exploit):** Podatności w serwerach lub systemach kontroli dostępu są wykorzystywane do przyznania nieautoryzowanemu użytkownikowi lub procesowi oprogramowania wyższych poziomów uprawnień, niż powinni mieć.

16.2 Framework Autoryzacji OAuth 2.0

- **Zarządzanie Tożsamością i Dostępem (IAM - Identity and Access Management):** Zasada bezpieczeństwa definiująca, kto może uzyskać dostęp do jakich zasobów i jakie uprawnienia ma po uzyskaniu dostępu.
- **OAuth 2.0 Authorization Framework:** Standaryzowany protokół do uwierzytelniania i autoryzacji opartej na internecie.
- Używany do kontroli dostępu urządzeń IoT, aby uczynić je bezpieczniejszymi, poprzez obsługę autoryzacji zasobów przez serwer autoryzacji.
- **Przepływ protokołu OAuth 2.0:**
  1. Klient (np. aplikacja mobilna) wysyła żądanie autoryzacji do Właściciela Zasobu (użytkownika).
  2. Właściciel Zasobu odsyła zgodę na autoryzację (authorization grant) do Klienta.
  3. Klient wysyła zgodę na autoryzację do Serwera Autoryzacji, żąda tokena dostępu i próbuje się uwierzytelnić.
  4. Jeśli uwierzytelnienie się powiodło, Serwer Autoryzacji waliduje zgodę i odsyła token dostępu do Klienta.
  5. Klient wysyła token dostępu do Serwera Zasobów (np. API urządzenia IoT), aby złożyć żądanie dostępu do zasobu.
  6. Po zwalidowaniu tokena dostępu, Serwer Zasobów zezwala na dostęp do żądanego zasobu.

16.3 Zarządzanie Tożsamością Urządzeń IoT (IoT Device Identity Management)

- Zarządzanie tożsamością odnosi się również do identyfikacji szerokiej gamy urządzeń oraz zarządzania ich dostępem do danych.
- Istnieje szeroka gama urządzeń IoT komunikujących się z innymi urządzeniami. Nierzadko przesyłane są wrażliwe informacje, a dostęp do tych danych powinien być kontrolowany.
- Zarządzanie tożsamością urządzeń IoT powinno obsługiwać dostęp urządzeń IoT do innych informacji z innych zasobów, oprócz obsługi dostępu do zasobów tego urządzenia.
- W miarę wykładniczego wzrostu liczby urządzeń IoT, wykładniczo rosną również relacje między urządzeniami.
- **Zarządzanie Zasobami Tożsamości (IRM - Identity Resource Management):** Pomaga organizacjom zarządzać większą liczbą tożsamości i relacji, jednocześnie utrzymując bezpieczeństwo zasobów.

17 Szyfrowanie (Encryption)

17.1 Bezpieczeństwo Danych i Haseł

- **Szyfrowanie:** Mechanizm używany do zapewnienia poufności danych.
- **Szyfrowanie (proces):** Zastosowanie algorytmu do danych, który uczyni je nieczytelnymi dla osób nieupoważnionych do ich zobaczenia.
- **Hasła** powinny być zawsze szyfrowane.
- Szyfrowanie danych IoT jest **krytyczne**, ponieważ przesyłane informacje mogą zawierać dane wrażliwe.
- Urządzenia IoT są podatne, ponieważ wiele starszych urządzeń IoT obecnie w produkcji **nie obsługuje szyfrowania**.
- Urządzenia IoT zazwyczaj wymagają komunikacji bezprzewodowej, co ułatwia przechwytywanie transmisji danych, jeśli nie ma szyfrowania. (Protokoły wspierające szyfrowanie: Zigbee, LoRa, LTE-M, White-Fi(?)).

17.2 Szyfrowanie w Systemach o Ograniczonych Zasobach

- Większość urządzeń IoT **nie ma mocy obliczeniowej ani zasobów** niezbędnych do bardziej niezawodnych algorytmów szyfrowania.
- Można używać **lekkich algorytmów szyfrowania (lightweight encryption)**.
  - Algorytmy te mogą być implementowane w oprogramowaniu lub sprzęcie.
  - Obecnie **nie ma standardu**, a wiele urządzeń IoT w ogóle nie obsługuje szyfrowania.
- NIST niedawno rozpoczął inicjatywę **„lightweight cryptography"**. Jej celem jest opracowanie standardowego algorytmu kryptograficznego, który może być używany w małych urządzeniach IoT przy minimalnych zasobach.

17.3 Kryptografia Klucza Publicznego (Public Key Cryptography)

- **Kryptografia (Symetryczna):** Opiera się na tym, że nadawca i odbiorca wiadomości znają i używają tego samego tajnego klucza. Nadawca używa tajnego klucza do zaszyfrowania wiadomości, a odbiorca używa tego samego tajnego klucza do jej odszyfrowania. Wyzwaniem jest bezpieczne zarządzanie kluczami, ponieważ wszystkie klucze muszą pozostać tajne.
- **Kryptografia Klucza Publicznego (Asymetryczna):** Wprowadzona w 1976 roku przez Whitfielda Diffiego i Martina Hellmana w celu rozwiązania problemu bezpiecznego zarządzania kluczami.
  - Każda osoba otrzymuje parę kluczy: jeden zwany kluczem publicznym, a drugi kluczem prywatnym.
  - Klucz publiczny każdej osoby jest publikowany, podczas gdy klucz prywatny jest trzymany w tajemnicy.
  - W tym systemie każdy może wysłać poufną wiadomość, używając informacji publicznej (klucza publicznego odbiorcy), ale wiadomość może być odszyfrowana tylko za pomocą klucza prywatnego zamierzonego odbiorcy.
  - Może być używana nie tylko do zapewnienia prywatności (szyfrowanie), ale także do uwierzytelniania (podpisy cyfrowe).

17.4 Urzędy Certyfikacji i System Zaufania PKI (Authorities and the PKI Trust System)

- Infrastruktura Klucza Publicznego (**PKI - Public Key Infrastructure**) wraz z Urzędem Certyfikacji (**CA - Certificate Authority**) jest potrzebna do obsługi dystrybucji na dużą skalę i identyfikacji publicznych kluczy szyfrujących. PKI służy do **udowodnienia tożsamości** urządzenia IoT.
- **Elementy Struktury PKI:**
  - **Certyfikat PKI:** Certyfikaty zawierają klucz publiczny podmiotu lub osoby.
  - **Magazyn Certyfikatów (Certificate Store):** Znajduje się na lokalnym komputerze i przechowuje wydane certyfikaty oraz klucze prywatne.
  - **Urząd Certyfikacji PKI (CA):** Zaufana strona trzecia, która wydaje certyfikaty. Podpisuje te certyfikaty za pomocą swojego klucza prywatnego.
  - **Baza Danych Certyfikatów (Certificate Database):** Przechowuje wszystkie certyfikaty wydane przez CA.
- **Przykład działania PKI:** Bob otrzymał swój certyfikat cyfrowy od CA. Certyfikat ten jest używany, gdy Bob komunikuje się z innymi stronami. Bob komunikuje się z Alicją. Gdy Alicja otrzymuje cyfrowy certyfikat Boba, komunikuje się z zaufanym CA, aby zweryfikować tożsamość Boba.
- Używanie Urzędu PKI z urządzeniami IoT jest **wyzwaniem**. Zarządzanie certyfikatami dla dużej liczby urządzeń IoT jest czasochłonne i może być niemożliwe do zarządzania w miarę dodawania kolejnych urządzeń.

18 Podsumowanie

- **Komponenty sprzętowe urządzeń IoT:** OWASP ma listę podatności dla każdej powierzchni ataku. Szyfrowanie jest mało prawdopodobne w urządzeniach o ograniczonych zasobach (zwłaszcza Klasy 0). CPU, pamięć i porty fizyczne mogą być skompromitowane.
- **Komponenty programowe urządzeń IoT:** Systemy wbudowane mogą używać OS lub być programowane bezpośrednio. Kod interpretowany jest łatwy do modyfikacji. Kod kompilowany może być zmieniony, ale podpis cyfrowy może pomóc. Tryb debugowania/rozruchu stanowi ryzyko (ominięcie uwierzytelniania, backdoor).
- **Bezpieczeństwo sprzętu:** Trudne wdrożenie bezpieczeństwa fizycznego w odległych lokalizacjach. Standardowe protokoły nadzoru/bezpieczeństwa jako pierwsza linia obrony. Dostęp przez UART/JTAG stwarza ryzyko.
- **Podatności Firmware:** Mogą zawierać luki odkryte po wydaniu. Łatki mogą nie istnieć lub urządzenia mogą nie być aktualizowalne. Aktualizacje powinny być robione jak najszybciej i pochodzić ze zweryfikowanego, podpisanego źródła.
- **Koncepcje kontroli dostępu do sieci:** Analityk powinien znać modele MAC, DAC, RBAC, ABAC. OAuth 2.0 może być używany do kontroli dostępu w IoT. Zarządzanie tożsamością urządzeń IoT (IRM) jest kluczowe przy dużej liczbie urządzeń.
- **Szyfrowanie:** Krytyczne dla danych IoT. Większość urządzeń IoT nie ma zasobów na silne szyfrowanie; potrzebne są lekkie algorytmy (inicjatywa NIST). Implementacja kryptografii klucza publicznego jest zalecana. PKI służy do udowodnienia tożsamości urządzenia.

19 Funkcje i Podatności Warstwy Komunikacyjnej IoT

19.1 Funkcje Warstwy Komunikacyjnej

- Odpowiedzialna za **transport danych** między urządzeniami, obiektami (facilities) i aplikacjami.
- Komunikacja często odbywa się w **chmurze**.
- Bezpieczeństwo sieci musi być uwzględnione dla wszystkich elementów **powierzchni ataku** systemu IoT.
- Dane w ruchu (**data in motion**) mogą być przechwycone, uszkodzone lub zmienione.
- Ponieważ głównym celem IoT jest zbieranie danych, ataki na systemy transportujące dane mogą **sparaliżować cały system IoT**.

19.2 Podatności Warstwy Komunikacyjnej wg OWASP

OWASP wymienia podatności związane z warstwą komunikacyjną, dzieląc je według powierzchni ataku:

- **Usługi Sieciowe Urządzeń (Device Network Services):**
  - Ujawnienie informacji (Information disclosure)
  - Wstrzykiwanie (Injection)
  - Odmowa usługi (Denial of Service - DoS)
  - Nieszyfrowane usługi (Unencrypted Services)
  - Słabo zaimplementowane szyfrowanie (Poorly implemented encryption)
  - Usługi testowe/deweloperskie (Test/Development Services)
  - Podatne usługi UDP (Vulnerable UDP Services)
  - Atak typu DoS
  - Atak typu Replay (Replay attack)
  - Brak weryfikacji ładunku (Lack of payload verification)
  - Brak sprawdzania integralności wiadomości (Lack of message integrity check)
- **Ruch Sieciowy (Network Traffic):**
  - Ruch LAN
  - Ruch LAN do Internetu
  - Komunikacja krótkiego zasięgu (Short range)
  - Niestandardowe protokoły (Non-standard protocols)
  - Protokoły bezprzewodowe (Wi-Fi, Z-wave, XBee, Zigbee, Bluetooth, LoRa)
  - Manipulacja pakietami (fuzzing protokołów - Packet manipulation (protocol fuzzing))

20 Kanały Komunikacyjne IoT

20.1 Typy Sieci Bezprzewodowych

Istnieje wiele typów sieci bezprzewodowych wykorzystywanych w IoT:

- **WBAN (Wireless Body Area Network):** Sieć czujników noszonych na ciele lub implantowanych. Używa różnych protokołów do komunikacji z bramą w celu wysyłania danych do chmury.
- **WPAN (Wireless Personal Area Network):** Często wykorzystuje Bluetooth do łączenia urządzeń audio, trackerów fitness, smartwatchy z telefonem komórkowym (bramą).
- **WHAN (Wireless Home Area Network):** Używa Bluetooth lub innych protokołów do łączenia urządzeń AGD, systemów alarmowych, aktuatorów z bramami i Internetem.
- **WFAN (Wireless Field/Factory Area Network):** Wzmocnione komponenty sieciowe łączą czujniki i aktuatory w rozproszonych lokalizacjach w trudnych warunkach produkcyjnych.
- **WNAN (Wireless Neighborhood Area Network):** Sieć dla sieci energetycznych (power grid) na ograniczonym obszarze geograficznym, często obsługiwana przez routery polowe (field area router) na zewnątrz.

Każda sieć może używać różnych protokołów do komunikacji między urządzeniami IoT, między urządzeniami a bramami oraz między urządzeniami a Internetem/chmurą.

20.2 Kanały Komunikacyjne i Powierzchnia Ataku

- Kanały komunikacyjne istnieją między urządzeniami o niskiej mocy a bramą (gateway).
- **Brama** tłumaczy ruch z bezprzewodowej sieci czujników na ruch protokołu IP, który może podróżować po sieciach danych.
- Z powodu ograniczeń mocy, węzły mogą używać tylko radiów **bardzo krótkiego zasięgu**.
- Protokoły pozwalają danym z czujników podróżować **od węzła do węzła**, aż dotrą do bramy.
- Kanały komunikacyjne i umożliwiające je protokoły tworzą **powierzchnię ataku komunikacji danych IoT**.
- Model ETSI pokazuje podział na domenę aplikacji, domenę sieciową (z M2M Core i M2M Gateway) oraz domenę urządzeń M2M (z siecią obszarową M2M).

20.3 Scenariusze Komunikacji IoT (Topologie)

Protokoły bezprzewodowe IoT działają w różnych topologiach:

- **Topologia Siatki (Mesh):**
  - Inteligentne obiekty przekazują dane do innych inteligentnych obiektów, aby dotrzeć do bramy (która może być poza zasięgiem).
  - Umożliwia rozmieszczenie węzłów czujnikowych i inteligentnych obiektów na większym obszarze, niż gdyby każdy węzeł musiał komunikować się bezpośrednio z bramą.
- **Topologia Gwiazdy (Hub-and-Spoke / Star):**
  - Urządzenia ("things") muszą być wdrożone w zasięgu bramy IoT.

  - Ogranicza zasięg sieci i wymaga zakupu, instalacji i konfiguracji dodatkowych urządzeń bramowych oraz połączeń do WAN/Internetu.

- **Konwersja przez Bramę:** Brama konwertuje ruch (np. z protokołów 802.15.4) na Wi-Fi lub Ethernet i enkapsuluje dane w pakiety IP do transmisji.

- **Komunikacja Bezpośrednia IP (IP-capable things):**
  - Scenariusz, w którym urządzenia ("things") mogą komunikować się bezpośrednio z chmurą lub centrum danych.
  - Każde urządzenie ma swój unikalny adres IPv6.
  - Urządzenia mają własne stosy protokołów IPv6 i protokoły komunikacyjne.
  - Umożliwia wysyłanie danych czujników przez sieć IP bez konieczności translacji na IP przez bramę IoT.
  - Większość urządzeń nie używa Wi-Fi z powodu ograniczeń mocy/przetwarzania, więc brama konwertuje ruch do odpowiedniej enkapsulacji Warstwy 2, podczas gdy enkapsulacja Warstwy 3 (IP) najprawdopodobniej pozostaje niezmieniona.

21 Protokoły Bezprzewodowe

21.1 Przegląd Protokołów Bezprzewodowych w Stosie IoT

Stos protokołów IoT obejmuje następujące protokoły bezprzewodowe (na warstwach komunikacji/dostępu do sieci):

- **IEEE 802.15.4:** Standard dla bezprzewodowych sieci osobistych (WPAN) o niskiej przepływności, przeznaczony dla tanich urządzeń o niskiej prędkości (np. ZigBee, Thread, 6LoWPAN).

- **Bluetooth Low Energy (BLE):** Sieć osobista (WPAN) wykorzystująca częstotliwość radiową 2.4 GHz.

- **Wi-Fi:** Zbiór standardów IEEE 802.11 dla sieci WLAN działających na częstotliwościach 2.4 GHz i 5 GHz.

- **Near Field Communication (NFC):** Protokoły do komunikacji urządzenie-do-urządzenia w zasięgu do 4 cm (1.6 cala).

- **Cellular:** Obejmuje wszystkie technologie komórkowe (3GPP, 4G, LTE, 5G).

21.2 Bluetooth i Wi-Fi

- Najpopularniejsze protokoły bezprzewodowe w połączonych domach, automatyce domowej i aplikacjach bezpieczeństwa.

- Wielu dostawców tworzy urządzenia łatwe w konfiguracji, aby zwiększyć sprzedaż, ale **kosztem bezpieczeństwa**.

- Problem podłączenia nowego urządzenia (np. smart plug):
  - Jeśli ma być sterowane przez internet, musi połączyć się z domową siecią Wi-Fi, nawet nie znając SSID.
  - Niektóre urządzenia tworzą własne, tymczasowe **hotspoty**, które można zidentyfikować za pomocą aplikacji mobilnej lub oprogramowania PC dostawcy.

  - Umożliwia to kupującemu połączenie się z urządzeniem bez konfigurowania go dla sieci domowej.
  - Te **minimalnie zabezpieczone hotspoty Wi-Fi** działają, ale są zapomniane przez właściciela domu.
  - W połączeniu ze znanymi danymi logowania, stają się podatne na kontrolę przez nieautoryzowanych użytkowników.
- Słabo zabezpieczone urządzenia mogą stanowić **drogę wejścia** do reszty sieci dla atakujących i złośliwego oprogramowania.

21.3 Przegląd IEEE 802.15.4

- Ograniczenia mocy obliczeniowej i operacyjnej wielu urządzeń IoT wymagały opracowania nowych protokołów bezprzewodowych.
- **Protokół IEEE 802.15.4:**
  - Pierwotnie opracowany do użytku w osobistych sieciach obszarowych (PAN).
  - Popularny w szerokim zakresie zastosowań.
  - Składa się ze specyfikacji warstwy dostępu do medium (**MAC**) i warstwy fizycznej (**PHY**).
  - Wykorzystuje architekturę warstwową, co pozwala programistom tworzyć protokoły wyższych warstw na tej samej podstawie.
  - Protokoły **ZigBee, Thread i 6LoWPAN** działają na bazie 802.15.4.
- **Warstwy 802.15.4:**
  - *Warstwy Wyższe (Sieciowa i Aplikacji):* Niesprecyfikowane; istnieją różne implementacje.
  - *Podwarstwa MAC:* Odpowiedzialna za niezawodną komunikację w Warstwie 2 między dwoma urządzeniami, ramkowanie danych, adresowanie urządzeń, zarządzanie dostępem do kanału (CSMA/CA), asocjację/dysasocjację urządzeń.
  - *Warstwa Fizyczna (PHY):* Zapewnia transmisję strumienia bitów drogą radiową, aktywację/deaktywację transceivera, nasłuchiwanie nośnej (CSMA/CA), wskaźnik siły odbieranego sygnału (RSSI), wskaźnik jakości łącza (LQI), kodowanie danych i modulację, korekcję błędów.

21.4 Role Urządzeń IEEE 802.15.4

- **FFD (Full Function Device):**
  - Może działać jako koordynator PAN (przydziela lokalne adresy, brama do innych PAN).
  - Może komunikować się z dowolnym innym urządzeniem (FFD lub RFD).
  - Może przekazywać wiadomości (funkcja koordynatora PAN).
  - Jeden koordynator PAN na topologię.
- **RFD (Reduced Function Device):**
  - Bardzo proste urządzenie, skromne wymagania zasobowe.
  - Może komunikować się tylko z FFD.
  - Przeznaczone do bardzo prostych zastosowań (np. czujnik, przełącznik).
- **P (PAN Coordinator):** Specjalna rola FFD.

21.5 Topologie IEEE 802.15.4

Działa na Warstwie 2.

- **Topologia Gwiazdy (Star):** Komunikacja odbywa się między urządzeniami a jednym centralnym kontrolerem (koordynatorem PAN). Każda sieć gwiazdy wybiera unikalny identyfikator PAN, co pozwala wielu sieciom działać niezależnie.
  - Wszystkie urządzenia komunikują się z koordynatorem PAN (często zasilanym z sieci).
  - Inne urządzenia mogą być zasilane bateryjnie/zbierać energię.
- **Topologia Siatki (Mesh):** Również jeden koordynator PAN. Aplikacje takie jak sterowanie przemysłowe, monitorowanie, sieci czujników bezprzewodowych, śledzenie zasobów/inwentarza skorzystałyby na tej topologii. RFD muszą mieć możliwość połączenia się z FFD lub PAN.
  - Urządzenia mogą komunikować się bezpośrednio, jeśli są w zasięgu.
- **Topologia Drzewa Klastrów (Cluster Tree):** Specjalny przypadek sieci siatkowej, w której większość urządzeń to FFD. RFD może połączyć się z siecią drzewa klastrów jako węzeł liścia na końcu gałęzi.
  - Protokoły wyższych warstw, jak RPL, mogą tworzyć własne topologie niezależne od 802.15.4.

  We wszystkich topologiach istnieje jeden koordynator PAN.

21.6 Bezpieczeństwo IEEE 802.15.4

- Ponieważ 802.15.4 działa na warstwach fizycznej i łącza danych OSI, bezpieczeństwo ramek jest ważne.
- **Cztery podstawowe usługi bezpieczeństwa** realizowane na warstwie łącza danych (MAC):
  - **Kontrola Dostępu (Access control):** Zapobiega dołączaniu nieautoryzowanych urządzeń do sieci.
  - **Integralność Wiadomości (Message integrity):** Chroni przed zmianą danych w tranzycie za pomocą szyfrowanego klucza kryptograficznego (kod uwierzytelniania wiadomości - MAC).
  - **Poufność Wiadomości (Message confidentiality):** Zapobiega odczytaniu przesyłanych danych przez atakujących. Ładunki danych wiadomości są szyfrowane.
  - **Ochrona przed Powtórzeniem (Replay protection):** Legalne wiadomości mogą być przechwycone i wysłane ponownie później. Ponieważ są uwierzytelnione, mogą zostać zaakceptowane. Częste powtarzanie może degradować wydajność sieci.
- 802.15.4 używa **szyfrów klucza symetrycznego** do szyfrowania. Klucze symetryczne są **mniej bezpieczne** niż kryptografia klucza asymetrycznego (publicznego).
- Protokół 802.15.4 zapewnia funkcjonalność bezpieczeństwa w postaci **zestawów bezpieczeństwa (security suites)**, które mogą być określone przez nakładające się warstwy aplikacji.
- Każdy zestaw bezpieczeństwa oferuje różne schematy szyfrowania i uwierzytelniania z różnymi długościami kluczy (np. AES-CBC-MAC-32/64/128, AES-CTR, AES-CCM-32/64/128).
  - 'Null': Brak bezpieczeństwa.
  - 'AES-CBC-MAC': Tylko uwierzytelnianie/integralność.
  - 'AES-CTR': Tylko szyfrowanie (poufność).
  - 'AES-CCM': Szyfrowanie i uwierzytelnianie/integralność.

21.7 Protokoły Siatkowe (Mesh) Wykorzystujące 802.15.4

- **6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks):** Zapewnia usługi IPv6 dla urządzeń małej mocy w sieciach PAN. Może być dodany do innych stosów protokołów, takich jak Zigbee i Thread.
- **Zigbee:** Prosta i tania grupa protokołów komunikacji bezprzewodowej wyższych warstw, implementująca małe sieci PAN o niskiej mocy. Szeroko stosowana w automatyce domowej, zbieraniu danych medycznych itp. Dodaje opcjonalną warstwę bezpieczeństwa, warstwę sieciową (w tym routing) i warstwę aplikacji do warstw PHY i MAC IEEE 802.15.4.
- **Thread:** Protokół siatki bezprzewodowej zbudowany specjalnie dla domowych sieci IoT, który wykorzystuje IPv6 do adresowania i 6LoWPAN jako fundament.
- **WirelessHART:** Międzynarodowa specyfikacja bezprzewodowa dla Przemysłowego Internetu Rzeczy (IIoT). Zbudowana na sieci siatkowej 802.15.4.
- **ISA 100.11a:** Standard amerykański dla komunikacji w IoT.

21.8 Inne Opcje Bezprzewodowe

- Opracowano inne protokoły bezprzewodowe obsługujące sieci rozległe małej mocy (**LPWAN - Low Power Wide Area Networks**).
- **LoRa:** Jedna z technologii LPWAN, popularna ze względu na niski koszt i szerokie wdrożenie.
- **LoRaWAN:** Sieć The Things Network to międzynarodowa organizacja umożliwiająca rozwój systemów proof-of-concept IoT przy użyciu warstwy fizycznej radia LoRa oraz elementów warstwy łącza danych i warstwy sieciowej LoRaWAN. Zachęca do tworzenia i utrzymywania własnych bram LoRaWAN.
- **Cellular (Komórkowe):**
  - Standardy danych komórkowych (**3GPP**) są implementowane w celu rozszerzenia sieci IoT, ale dla urządzeń ze stałym zasilaniem.
  - Obecne usługi danych komórkowych nie są dobrze dostosowane do aplikacji IoT ze względu na ograniczenia mocy.
  - Specyfikacje komórkowe piątej generacji (**5G**) obejmują **LTE Advanced for Machine-Type Communication (LTE MTC)**. Ta technologia zawiera funkcje znacznie poprawiające zużycie energii, jednocześnie zapewniając uproszczoną obsługę urządzeń dla małych, okresowych transmisji danych.
  - **Narrowband IoT (NB-IoT):** Protokół małej mocy i niskiej przepustowości, szczególnie do zastosowań wewnętrznych. Wykorzystuje część widma częstotliwości operatora LTE.

22 Podatności Warstwy IP

22.1 Powierzchnia Ataku Warstwy Komunikacyjnej IoT

- Bezpieczeństwo IP jest poważnym problemem, ponieważ głównym celem IoT jest przesyłanie, przechowywanie i analizowanie danych przez sieci IP.
- Powierzchnia ataku IoT, która będzie miała podatności IP, obejmuje (przykład):
  - Sieć czujników

- Bramę IoT
- Sieć IT przedsiębiorstwa (np. router Cisco 829 na rysunku)
- Łącze do Internetu (Uplink)

Czerwone okręgi na diagramie wskazują podatne interfejsy lub urządzenia.

22.2 Typowe Ataki Związane z IP

- **Ataki DoS (Denial of Service):** Atakujący próbują uniemożliwić legalnym użytkownikom dostęp do informacji lub usług.
- **Ataki DDoS (Distributed DoS):** Podobne do DoS, ale charakteryzują się jednoczesnym, skoordynowanym atakiem z wielu maszyn źródłowych.
- **Ataki ICMP:** Atakujący używają pakietów echa ICMP (ping) do odkrywania podsieci i hostów w chronionej sieci, generowania ataków zalewowych DoS (flood) i zmiany tablic routingu hostów.
- **Ataki Spoofingu Adresów (Address Spoofing):** Atakujący umieszcza fałszywy adres IP źródłowy w pakiecie, aby podszyć się pod inne źródło, oszukując odbiorcę, że pakiet pochodzi z legalnego źródła.
- **Atak Man-in-the-Middle (MITM):** Atakujący pozycjonują się między źródłem a miejscem docelowym, aby transparentnie monitorować, przechwytywać i kontrolować komunikację. Mogą po prostu podsłuchiwać, przeglądając przechwycone pakiety, lub zmieniać pakiety i przesyłać je dalej do pierwotnego miejsca docelowego.
- **Przejmowanie Sesji (Session Hijacking):** Atakujący uzyskują dostęp do sieci fizycznej, a następnie używają ataku MITM do przechwycenia ważnego tokenu dostępu do serwera WWW.

22.3 Ataki DoS

- Celem jest uniemożliwienie legalnym użytkownikom dostępu do usług.
- Dwa główne źródła ataków DoS:
  - **Złośliwie Sformatowane Pakiety:** Atakujący tworzą złośliwie sformatowany pakiet i przesyłają go do podatnego hosta, powodując jego awarię lub ekstremalne spowolnienie.
  - **Przytłaczająca Ilość Ruchu:** Atakujący przytłaczają docelową sieć, hosta lub aplikację, powodując ich awarię lub ekstremalne spowolnienie.
- Atak **DDoS** jest większy, ponieważ pochodzi z wielu źródeł (botnet).
- **Scenariusz ataku DDoS (np. Mirai Botnet):**
  1. Atakujący (botmaster) buduje lub kupuje użycie botnetu składającego się z zainfekowanych hostów (zombie).
  2. Komputery zombie kontynuują skanowanie i infekowanie kolejnych celów, aby stworzyć więcej zombie.
  3. Gdy jest gotowy, botmaster używa systemów zarządzających (handler systems), aby botnet zombie przeprowadził atak DDoS na wybrany cel.

22.4 Ataki Wzmocnienia i Odbicia (Amplification and Reflection)

- Technika używana do przytłoczenia hosta docelowego. Przykład: **Atak Smurf**.
- **Kroki ataku Smurf:**
  1. **Wzmocnienie (Amplification):** Atakujący wysyła (forward) wiadomości żądania echa ICMP (ICMP echo request), które zawierają **sfałszowany adres IP źródłowy ofiary**, do dużej liczby hostów (np. na adres rozgłoszeniowy sieci).
  2. **Odbicie (Reflection):** Wszystkie hosty odpowiadają (wysyłając ICMP echo reply) na sfałszowany adres IP ofiary, przytłaczając ją ruchem.

22.5 Ataki ICMP

- **ICMP (Internet Control Message Protocol):** Opracowany do przenoszenia komunikatów diagnostycznych i raportowania błędów, gdy trasy, hosty lub porty są niedostępne.
- Polecenie **Ping** używa generowanego przez użytkownika komunikatu ICMP zwanego żądaniem echa (echo request) do weryfikacji łączności z miejscem docelowym. Odpowiedź to echo reply.
- Atakujący używają ICMP do **rekonesansu i skanowania**.
- Atakujący często używają ICMP do tworzenia **ataków DoS** (np. ICMP Flood, gdzie wysyłane są liczne żądania echo, często ze sfałszowanym adresem źródłowym).

22.6 Ataki Spoofingu Adresów

- **Spoofing adresu IP:** Występuje, gdy atakujący tworzy pakiety z fałszywymi informacjami o adresie IP źródłowym.
- **Spoofing adresu MAC (Media Access Control):** Używany, gdy atakujący mają dostęp do sieci wewnętrznej.
  - Atakujący zmienia adres MAC swojej karty sieciowej, aby pasował do adresu MAC legalnego serwera.
  - Gdy atakujący wyśle ramkę, przełącznik zaktualizuje swoją tablicę CAM (Content Addressable Memory), błędnie kojarząc adres MAC serwera z portem, do którego podłączony jest atakujący. Ruch przeznaczony dla serwera zostanie wysłany do atakującego.

23 Podatności TCP i UDP

23.1 Podatności TCP

- TCP (Transmission Control Protocol) świadczy następujące usługi:
  - **Niezawodne dostarczanie (Reliable delivery):** Wykorzystuje potwierdzenia (acknowledgments - ACK) do gwarantowania dostarczenia. Jeśli potwierdzenie nie nadejdzie na czas, nadawca retransmituje dane.
  - **Kontrola przepływu (Flow control):** Implementuje kontrolę przepływu, aby radzić sobie z opóźnieniami. Zamiast potwierdzać każdy segment osobno, wiele segmentów może być potwierdzonych jednym segmentem ACK.
  - **Komunikacja stanowa (Stateful communication):** Zanim dane zostaną przesłane, **trójfazowe uzgadnianie (three-way handshake)** otwiera połączenie TCP. Jeśli obie strony zgadzają się na połączenie, dane mogą być wysyłane i odbierane.

- Protokoły warstwy transportowej (jak TCP i UDP) używają **adresowania portów**, aby umożliwić śledzenie wielu konwersacji i łączenie ich z właściwymi aplikacjami.
- **Dobrze znane numery portów (Well-known port numbers)** identyfikują powszechnie używane aplikacje.
- Aplikacja taka jak Telnet może być przypisana do dowolnego numeru portu w zakresie otwartych portów. Ponieważ Telnet **nie jest bezpieczny**, nie powinien być uruchomiony na urządzeniach IoT.
- Ważne jest, aby ocenić każdy inteligentny obiekt pod kątem domyślnie włączonych protokołów komunikacyjnych i otwartych portów nasłuchujących.
- Protokół TCP jest podatny na **skanowanie portów**.
  - Atakujący przeprowadzają skanowanie portów docelowych urządzeń, aby odkryć, jakie usługi oferują.
  - Skanery portów mogą dostarczyć bardzo szczegółowych informacji o usługach działających na urządzeniu sieciowym. Te usługi mogą być podatne na wykorzystanie przez atakujących.
- **Atak TCP SYN Flood:** Wykorzystuje trójfazowe uzgadnianie TCP. Atakujący wysyła wiele żądań SYN (często ze sfałszowanym adresem IP), ale nie odpowiada na otrzymane SYN-ACK. Powoduje to, że serwer utrzymuje wiele półotwartych połączeń, zużywając zasoby i potencjalnie uniemożliwiając obsługę legalnych żądań.
- **Atak TCP Reset (RST):** Połączenie TCP może zostać zerwane, gdy otrzyma bit RST. Jest to gwałtowny sposób zerwania połączenia. Atakujący może uruchomić atak resetujący TCP i wysłać sfałszowany pakiet zawierający TCP RST do jednego lub obu punktów końcowych.
- **Przejmowanie sesji TCP (TCP session hijacking):** Inna podatność TCP. Chociaż trudna do przeprowadzenia, umożliwia atakującemu przejęcie już uwierzytelnionego hosta podczas jego komunikacji z celem.

23.2 Podatności UDP

- UDP (User Datagram Protocol) jest protokołem **bezpołączeniowym**. Używany przez DNS, TFTP, NFS, SNMP, media strumieniowe, VoIP itp.
- Ma **mniejszy narzut** niż TCP, ponieważ nie oferuje mechanizmów retransmisji, sekwencjonowania i kontroli przepływu, które zapewniają niezawodność.
- UDP **nie jest chroniony żadnym domyślnym szyfrowaniem**. Możliwe jest dodanie szyfrowania (np. DTLS), ale nie jest ono dostępne domyślnie.
- Zmiana danych w ruchu zmieni 16-bitową **sumę kontrolną UDP**, która jest obowiązkowa przy przesyłaniu przez IPv6.
- Atakujący może **utworzyć nową sumę kontrolną** na podstawie nowych danych ładunku i zapisać ją w nagłówku. Urządzenie docelowe stwierdzi, że suma kontrolna pasuje do danych, nie wiedząc, że dane zostały zmienione.
- Bardziej powszechny jest atak UDP, w którym zużywane są wszystkie zasoby sieciowe - **UDP flood attack**.

24 Bezpieczeństwo Komunikacji IoT

24.1 Ogólne Zasady Bezpieczeństwa Protokołów Komunikacyjnych

- Urządzenia IoT muszą być **bezpieczne fizycznie** i chronione przed uszkodzeniem, zniszczeniem i manipulacją.
- Fizyczne oprogramowanie układowe (firmware) urządzenia i interfejsy **nie mogą być podatne**.
- Pierwszym etapem tworzenia modelu zagrożeń bezpieczeństwa systemu IoT jest **identyfikacja technologii komponentów i ich cech** (jak obieranie płatków karczocha bezpieczeństwa).
- Komunikacja IoT musi być również **bezpieczna**.
- Urządzenia dołączające do sieci czujników i aktuatorów IoT muszą używać **solidnych protokołów bezprzewodowych**.
- **Podejścia kryptograficzne**, które chronią poświadczenia, szyfrują ładunki danych i tworzą bezpieczne kanały dla konwersacji protokołów, są również ważną częścią bezpieczeństwa IoT.

24.2 Izolacja Ruchu IT i OT

- Ważne jest, aby **zmniejszyć rozmiar całkowitej powierzchni ataku** poprzez ustanowienie mniejszych **stref zaufania** za pomocą zapór ogniowych (firewalls) i innych technologii bezpieczeństwa.
- Pomaga to zapewnić, że udany atak na jedną część sieci **nie pozwoli na dostęp** do wszystkich części sieci.
- Dane z wielu obszarów sieci mogą potrzebować przejść przez te strefy zaufania. Dlatego techniki **izolacji ruchu**, takie jak używanie **VLANów**, są ważne.
- Przykład architektury z izolacją i strefami: Strefa Przedsiębiorstwa (Poziom 5, 4), Strefa Zdemilitaryzowana (DMZ), Strefa Produkcyjna (Poziom 3, 2, 1, 0), Strefa Bezpieczeństwa (Safety-Critical).

24.3 Studium Przypadku: Brak Izolacji Ruchu IT i OT

- Duży detalista odkrył naruszenie bezpieczeństwa, w którym skradziono dane osobowe, w tym numery kart kredytowych, ponad 50 milionów klientów.
- Naruszenie nastąpiło z powodu **słabego bezpieczeństwa u podwykonawcy** zajmującego się ogrzewaniem, wentylacją i klimatyzacją (HVAC).
- Podatności w sieci podwykonawcy umożliwiły atakującym zainfekowanie sieci punktów sprzedaży (POS) detalisty złośliwym oprogramowaniem.
- Dwa ważne wnioski z tego przypadku:
  - Bezpieczeństwo urządzenia, sieci i aplikacji jest tak silne, jak **najsłabsze ogniwo**, którym w tym przypadku było bezpieczeństwo podwykonawcy mającego dostęp do sieci.
  - Podwykonawca **nie powinien był mieć dostępu** do tej samej sieci, która przenosiła dane poufne. W tym przypadku sieć podwykonawcy stała się częścią powierzchni ataku detalisty.
- Na poziomie protokołu TCP/IP komunikacji powinny być używane **tylko bezpieczne protokoły warstwy aplikacji**.

- Bezpieczeństwo transportu, w postaci **TLS** lub **DTLS**, powinno być zaimplementowane do uwierzytelniania i ochrony danych IoT.

24.4 Model Zagrożeń dla Technologii Komunikacyjnych IoT

- OWASP zaleca wytyczne do mitygacji podatności kanałów komunikacyjnych w systemach IoT:
  - *I3: Niezabezpieczone Usługi Sieciowe:* Minimalna liczba aktywnych portów, brak portów/usług dostępnych z internetu, przegląd usług pod kątem podatności (np. przepełnienie bufora, DoS).
  - *I4: Brak Szyfrowania Transportowego:* Szyfrowanie całej komunikacji między komponentami systemu oraz między systemem/urządzeniem a internetem, używanie zalecanych praktyk szyfrowania, unikanie protokołów własnościowych, aktualne i poprawnie skonfigurowane implementacje SSL/TLS, rozważenie opcji zapory ogniowej.
- Model zagrożeń tych systemów wymaga dokładnego zrozumienia przepływów danych w sieci i lokalizacji granic zaufania względem tych przepływów.
- **Granice zaufania** to obszary w systemie komunikacyjnym, skąd dane mogą pochodzić z niezaufanego źródła.
- Tworzenie modelu zagrożeń obejmuje użycie modelu **STRIDE** do identyfikacji zagrożeń i modelu **DREAD** do oceny ryzyka każdego zagrożenia.

24.5 Lista Kontrolna Bezpieczeństwa Komunikacji IoT

- **Weryfikacja bezpieczeństwa 802.15.4:**
  - Tryb AES-CTR nie jest używany (samodzielnie, bo nie zapewnia integralności).
  - Używane jest zarówno szyfrowanie, jak i uwierzytelnianie.
  - Potwierdzenia nie są używane jako jedyny mechanizm niezawodności w tym kanale.
  - Integralność zarządzania kluczami jest zapewniona.
  - Listy kontroli dostępu (ACL) są utrzymywane w trybie niskiego poboru mocy.
  - Używana jest najnowsza wersja protokołu.
- **Weryfikacja bezpieczeństwa komunikacji TCP/IP:**
  - Wszystkie przepływy danych są dokładnie i precyzyjnie udokumentowane.
  - Przepływ danych jest odpowiednio chroniony na granicach zaufania.
  - Zapory ogniowe lub inne systemy segregują ruch w różnych strefach zaufania.
  - Wykorzystywane jest szyfrowanie transportowe (TLS/DTLS).

25 Podsumowanie

25.1 Funkcje Warstwy Komunikacyjnej IoT

- Odpowiedzialna za transport danych między urządzeniami, obiektami i aplikacjami, często w chmurze.
- Wiele węzłów czujników IoT ma ograniczone zasoby, moc i przetwarzanie.
- Kanały komunikacyjne będą istnieć między urządzeniami małej mocy a bramą.

25.2 Protokoły Bezprzewodowe

- Stos protokołów IoT obejmuje IEEE 802.15.4, BLE, Wi-Fi, NFC i Cellular.
- Bluetooth i Wi-Fi są łatwe w konfiguracji, ale często odbywa się to kosztem bezpieczeństwa.
- ZigBee, Thread, WirelessHART, ISA 100.11a i 6LoWPAN działają na bazie urządzeń 802.15.4 (FFD, PAN coordinator, RFD).
- Protokół 802.15.4 zapewnia funkcjonalność bezpieczeństwa w postaci zestawów bezpieczeństwa określanych przez warstwy aplikacji.
- LoRa jest jedną z popularnych technologii LPWAN ze względu na niski koszt i szerokie wdrożenie.
- Obecne usługi danych komórkowych nie są dobrze dostosowane do aplikacji IoT ze względu na ograniczenia mocy.

25.3 Podatności IP

- Obejmują sieć czujników, bramę IoT, sieć IT przedsiębiorstwa i łącze do internetu.
- Ataki celujące w IP to DoS, DDoS, ICMP, spoofing adresów, MITM i przejmowanie sesji.
- Atakujący często używają technik wzmocnienia i odbicia do tworzenia ataków DoS.
- Atakujący używają ICMP do rekonesansu i skanowania.
- Ataki spoofingu mogą być przeprowadzane jako ślepe (blind) lub nieślepe (non-blind).

25.4 Podatności TCP i UDP

- Protokół TCP jest podatny na skanowanie portów.
- Ataki TCP obejmują TCP SYN flood, TCP reset i przejmowanie sesji TCP.
- Możliwe jest dodanie szyfrowania do UDP, ale nie jest ono dostępne domyślnie.

25.5 Bezpieczeństwo Komunikacji IoT

- Podejścia kryptograficzne chronią poświadczenia, szyfrują ładunki danych i tworzą bezpieczne kanały.
- Zmniejszenie powierzchni ataku poprzez ustanowienie mniejszych stref zaufania za pomocą zapór i innych technologii.
- Na poziomie protokołu TCP/IP powinny być używane tylko bezpieczne protokoły warstwy aplikacji.
- Tworzenie modelu zagrożeń obejmuje użycie modeli STRIDE i DREAD do identyfikacji zagrożeń i oceny ryzyka.

Notatki: Luki w Zabezpieczeniach Aplikacji IoT Na podstawie prezentacji Cisco chapter 5 28 marca 2025

Spis treści

26 Luki w Lokalnych Aplikacjach IoT

26.1 Luki Aplikacyjne wg OWASP

Projekt Open Web Applications Security Project (OWASP) wymienia najczęściej eksponowane luki w zabezpieczeniach aplikacji IoT:

- **Wyliczanie nazw użytkowników (Username enumeration):** Atakujący może odkryć prawidłowe nazwy użytkowników poprzez interakcję z mechanizmem uwierzytelniania aplikacji.
- **Słabe hasła:** Atakujący wykorzystuje domyślne hasła, które nie zostały zmienione przez użytkownika.
- **Blokada konta (Account lockout):** Atakujący próbuje się uwierzytelnić wielokrotnie, powodując zablokowanie konta (choć może to być również celem ataku typu DoS na użytkownika).
- **Brak uwierzytelniania wieloskładnikowego (MFA):** Brak dodatkowej warstwy weryfikacji tożsamości użytkownika.
- **Niebezpieczne komponenty stron trzecich:** Luki odkrywane w zewnętrznych bibliotekach lub komponentach wymagają instalacji poprawek.

26.2 Popularne Eksploity Lokalne i Zdalne

26.2.1 Eksploity Lokalne

Wymagają fizycznego dostępu lub bliskości urządzenia:

- Zamiana oprogramowania układowego (Firmware Replacement)
- Klonowanie (Cloning)
- Odmowa usługi (Denial of Service - DoS)
- Ekstrakcja parametrów bezpieczeństwa (Extraction of Security Parameters)

26.2.2 Eksploity Zdalne

Mogą być wykonane przez sieć:

- **Atak Man-In-the-Middle (MITM):** Atakujący podsłuchuje lub modyfikuje komunikację między dwoma urządzeniami.
- **Podsłuchiwanie (Eavesdropping Attack):** Atakujący przechwytuje dane (np. klucze bezpieczeństwa) podczas instalacji lub komunikacji urządzeń, zwłaszcza tych o ograniczonych zasobach.
- **Wstrzykiwanie SQL (SQL Injection - SQLi):** Atakujący wykorzystuje lukę w aplikacji do wykonania złośliwych zapytań SQL, co pozwala na dostęp, modyfikację danych lub uzyskanie uprawnień administracyjnych.
- **Atak na routing (Routing Attack):** Atakujący umieszcza w sieci fałszywe urządzenie routujące w celu przechwycenia lub przekierowania ruchu.

27 Luki w Aplikacjach Mobilnych

Najczęściej eksponowane luki w zabezpieczeniach urządzeń mobilnych (które często kontrolują urządzenia IoT):

- **Niezabezpieczona komunikacja:** Kanał i technologia komunikacji muszą być odpowiednio zabezpieczone (np. przez szyfrowanie).
- **Niezabezpieczone przechowywanie danych:** Wiele aplikacji ma dostęp do obszarów pamięci masowej urządzeń mobilnych, nawet jeśli tego nie potrzebuje, co stwarza ryzyko wycieku danych.
- **Niezabezpieczone uwierzytelnianie:** Sesje użytkowników muszą być prawidłowo zarządzane, aby zapewnić bezpieczne uwierzytelnianie (np. unikanie długotrwałych sesji, bezpieczne zarządzanie tokenami).
- **Niewłaściwe użycie platformy:** Nadużywanie wbudowanych mechanizmów bezpieczeństwa systemu mobilnego może prowadzić do kompromitacji urządzenia lub innych aplikacji.
- **Niewystarczająca kryptografia:** Użycie słabych algorytmów szyfrowania, nieprawidłowe zarządzanie kluczami lub brak szyfrowania tam, gdzie jest to wymagane.

28 Luki w Aplikacjach Webowych i Chmurowych

28.1 Najczęstsze Luki wg OWASP (Web/Cloud)

- **Wstrzykiwanie (Injection):** Atakujący wysyła złośliwe polecenia wraz z danymi do interpretera. Dotyczy m.in. SQL, NoSQL, LDAP, OS command injection.
- **Zewnętrzne encje XML (XML External Entities - XXE):** Wykorzystanie zewnętrznych encji XML może prowadzić do ujawnienia plików, skanowania portów, wykonania kodu lub ataków DoS.
- **Narażenie wrażliwych danych (Sensitive Data Exposure):** Interfejsy API i aplikacje webowe nie zawsze prawidłowo chronią dane wrażliwe (np. brak szyfrowania danych w spoczynku lub w tranzycie).
- **Naruszenie kontroli dostępu (Broken Access Control):** Jeśli kontrola dostępu nie jest prawidłowo skonfigurowana, atakujący może uzyskać dostęp do nieautoryzowanych danych lub funkcji.
- **Naruszenie uwierzytelniania (Broken Authentication):** Nieprawidłowe wdrożenie zarządzania sesją i uwierzytelniania może umożliwić atakującemu przejęcie tożsamości użytkownika.

28.2 Zarządzanie Urządzeniami i Aplikacje Danych

- Dane IoT mogą być przechowywane na brzegu sieci (edge), centralnie lub w chmurze.
- Przetwarzanie danych odbywa się na różnych poziomach: mgły (fog), mgły rozproszonej (mist) lub w chmurze.
- **Przetwarzanie w chmurze:** Aplikacje chmurowe wykonują złożone obliczenia na dużych ilościach danych sensorycznych. Mogą:
  - Wyzwalać akcje na aktuatorach.

- Powiadamiać ludzi.
- Aktualizować bazy danych.
- Umożliwiać analizy kognitywne i predykcyjne.
- Być używane w czasie rzeczywistym (np. sterowanie ruchem, warunkami fabrycznymi, reagowanie na sytuacje awaryjne).

28.3 Wytyczne dla Bezpiecznych Aplikacji Webowych i Chmurowych

- Przeglądaj znane luki w zabezpieczeniach.
- Zapobiegaj używaniu słabych haseł.
- Zapobiegaj atakom typu brute-force (np. przez blokowanie kont, CAPTCHA).
- Używaj uwierzytelniania dwuskładnikowego (2FA).
- Stosuj szyfrowanie transportowe (np. TLS/SSL).
- Testuj pod kątem luk SQLi, XSS, CSRF.
- Ustawiaj hasła tak, aby wygasały.
- Zmieniaj domyślne nazwy użytkowników i hasła.

28.4 Luki Związane z Hasłami

- Największym problemem jest użytkownik (skłonność do prostych, powtarzalnych haseł).
- Hasła powinny być łatwe do zapamiętania, ale trudne do złamania.
- Wiele haseł jest łatwych do odgadnięcia, rzadko zmienianych, używanych wielokrotnie i zapisywanych w łatwo dostępnych miejscach.
- Wielu użytkowników nie zmienia domyślnych haseł.
- Atakujący wykorzystują te słabości do kompromitacji systemów.

28.5 Luki Interfejsu Użytkownika (Frontend)

Dotyczą aplikacji, API i usług webowych. Trzy najczęstsze:

- **Cross-Site Scripting (XSS):** Atakujący wstrzykuje złośliwy kod (zwykle JavaScript) do danych wyjściowych aplikacji webowej, który jest następnie wykonywany w przeglądarce ofiary. Może to prowadzić do kradzieży sesji, przekierowań, itp.
- **SQL Injection (SQLi):** Atakujący wstrzykuje kod SQL do pól wejściowych, które są używane do konstruowania zapytań do bazy danych. Umożliwia to manipulację bazą danych.
- **Broken Authentication:** Atakujący może przejąć sesję użytkownika, jeśli tokeny sesji nie wygasają lub są łatwe do przewidzenia/przechwycenia.

29 Modelowanie Zagrożeń na Warstwie Aplikacji

Modelowanie zagrożeń pomaga zapewnić bezpieczeństwo aplikacji. Dla warstwy aplikacji IoT, koncentruje się na krokach 3, 4 i 5 ogólnego procesu:

- **Krok 3: Dekompozycja Aplikacji:** Zrozumienie aplikacji, jej komponentów i interakcji z zewnętrznymi encjami.
- **Krok 4: Identyfikacja i Ocena Zagrożeń:** Kategoryzacja zagrożeń (np. przy użyciu metodologii STRIDE lub DREAD) z perspektywy atakującego. Ocena prawdopodobieństwa i wpływu.
- **Krok 5: Rekomendacja Środków Zaradczych (Mitigation):** Proponowanie środków zaradczych (countermeasures) w celu zminimalizowania lub wyeliminowania zagrożeń. Czasami ryzyko może być akceptowalne, jeśli jego wpływ jest mniejszy niż koszt środka zaradczego.

30 Protokoły Warstwy Aplikacji IoT

30.1 Rola Protokołów Komunikacyjnych

- Urządzenia IoT używają protokołów komunikacyjnych do uzgodnienia sposobu wymiany informacji.
- **HTTP jest często nieefektywne dla IoT** z następujących powodów:
  - Urządzenia IoT mają ograniczone zasoby (pamięć, moc obliczeniowa) i mogą nie być w stanie uruchomić pełnych usług HTTP.
  - HTTP jest protokołem "ciężkim", zużywającym więcej zasobów (nagłówki tekstowe, połączenia).
  - Nie zawiera natywnego modelu publikacji/subskrypcji (publish/subscribe).
  - Brak mechanizmu przechowywania lub buforowania wiadomości dla urządzeń offline.
  - Brak standardowego sposobu informowania o niedostępności urządzenia.

30.2 Popularne Protokoły Komunikacyjne IoT

- **MQTT (Message Queuing Telemetry Transport):**
  - Używa TCP.
  - Wymaga brokera wiadomości.
  - Model publikacji/subskrypcji (publish-subscribe).
  - Opracowany przez IBM dla komunikacji M2M (Machine-to-Machine).
  - Klient publikuje wiadomości na określone "tematy"(topics) lub subskrybuje tematy.
  - Tematy są zorganizowane hierarchicznie.
  - Zaprojektowany do zbierania danych z wielu urządzeń i dostarczania ich do infrastruktury IT.
  - **Zalety dla IoT:** Odporność na rozłączenia klientów (broker może buforować wiadomości), minimalizacja ruchu sieciowego i zużycia energii.
- **CoAP (Constrained Application Protocol):**
  - Używa UDP.

  - Protokół transferu dokumentów (podobny do HTTP, ale lżejszy).
  - Zaprojektowany dla komunikacji M2M i urządzeń o ograniczonych zasobach.
  - Model klient-serwer (żądania/odpowiedzi).
  - Obsługuje metody podobne do HTTP: GET, POST, PUT, DELETE.
  - Posiada mechanizm "obserwacji"zasobów, pozwalający serwerowi informować klienta o zmianach stanu (ważne dla IoT).
  - Często używany do raportowania zmian stanu urządzeń IoT.
- **XMPP (Extensible Messaging and Presence Protocol):**
  - Używa TCP i XML.
  - Pierwotnie zaprojektowany dla komunikatorów internetowych (np. Jabber).
  - Używany do łączenia domowych urządzeń IoT z serwerem webowym w celu monitorowania (np. przez smartfon).
- **DDS (Data Distribution Service):**
  - Używany do połączeń M2M, szczególnie tam, gdzie urządzenia bezpośrednio wykorzystują dane od siebie nawzajem (bez centralnego brokera).
  - Stosowany w obrazowaniu medycznym, testach motoryzacyjnych, handlu finansowym, kontroli ruchu lotniczego, sieciach IoT o dużym natężeniu danych.
- **AMQP (Advanced Message Queuing Protocol):**
  - Protokół kolejkowania wiadomości, używany, gdy niezawodność dostarczania jest kluczowym czynnikiem.

30.3 Ważne Charakterystyki Protokołów IoT

Przy wyborze protokołu należy wziąć pod uwagę:

- Zużycie energii
- Prędkość transmisji
- Opóźnienie (Latency)
- Bezpieczeństwo

30.4 Uwaga o UPnP (Universal Plug and Play)

- Zestaw protokołów pozwalający urządzeniom na automatyczne wykrywanie się w sieci lokalnej bez interwencji użytkownika.
- Zaprojektowany głównie dla sieci domowych.
- Multicastowy charakter UPnP zużywa dużo zasobów w sieciach z wieloma urządzeniami (np. korporacyjnych).
- **Posiada poważne luki w zabezpieczeniach:**
  - Mogą umożliwić atakującemu zdalne sterowanie urządzeniami (kamery, światła itp.).
  - Routery domowe z UPnP mogą zostać oszukane przez złośliwe oprogramowanie do przekierowania ruchu DNS na fałszywy serwer za pomocą pojedynczego żądania UPnP.
- UPnP zakłada, że wszystkie urządzenia w sieci lokalnej są przyjazne i godne zaufania, co jest niebezpiecznym założeniem.

31 Łagodzenie Problemów Bezpieczeństwa w Protokolach Komunikacyjnych

31.1 Zabezpieczanie MQTT

31.1.1 Uwierzytelnianie Klienta

Trzy główne metody:

1. **Client ID:** Każdy klient musi mieć unikalny identyfikator. Broker używa go do powiązania subskrypcji z klientami. (Podstawowy, ale nie silny mechanizm uwierzytelniania sam w sobie).
2. **Nazwa użytkownika i hasło:** Kombinacja wysyłana w postaci jawnego tekstu. **Wymaga szyfrowania transportowego (SSL/TLS)** do ochrony poświadczeń.
3. **Certyfikaty klienta (x.509):** Mogą być wdrożone, gdy wymagany jest wysoki poziom bezpieczeństwa. Klient przedstawia certyfikat brokerowi.

31.1.2 Zabezpieczanie Wiadomości

Dwa sposoby:

1. **Szyfrowanie SSL/TLS:** Zabezpiecza całe połączenie między klientem a brokerem (szyfrowanie na warstwie transportowej). Używa tej samej technologii co HTTPS.
2. **Szyfrowanie ładunku (Payload encryption):** Wykonywane na warstwie aplikacji. Zapewnia szyfrowanie end-to-end, chroniąc treść wiadomości nawet przed samym brokerem.

Wybór implementacji bezpieczeństwa zależy od możliwości brokera i klientów.

31.2 Zabezpieczanie CoAP

- CoAP używa **DTLS (Datagram Transport Layer Security)** - odpowiednika TLS dla UDP. Zabezpieczony CoAP jest czasem nazywany CoAPs.
- Istnieją proste implementacje DTLS dla urządzeń wbudowanych (np. `tinydtls`).
- **DTLS zapewnia:**
  - Ochronę danych za pomocą kluczy i algorytmów szyfrujących.
  - Mechanizm wymiany kluczy.
  - Uwierzytelnianie stron komunikacji.

31.3 Wyłączanie UPnP

- Najlepszą praktyką jest **wyłączenie UPnP** na wszystkich urządzeniach (routerach, urządzeniach końcowych), gdy jest to możliwe.
- Może to wymagać ręcznej konfiguracji niektórych usług (np. przekierowania portów).
- Większość routerów domowych umożliwia włączenie/wyłączenie UPnP w interfejsie administracyjnym.
- Jeśli urządzenie w sieci zostanie zainfekowane złośliwym oprogramowaniem, może ono wykorzystać UPnP (jeśli jest włączone na routerze) do otwarcia portów i uzyskania dostępu z zewnątrz.

31.4 Problemy z Hasłami - Wzmocnienie

- **Zawsze zmieniaj domyślną nazwę użytkownika i hasło** na nowym urządzeniu.
- Stosuj następujące wytyczne:
  - **Hasła frazowe (Passphrases):** Używaj dłuższych fraz zamiast pojedynczych słów. Są łatwiejsze do zapamiętania i trudniejsze do złamania niż krótkie, skomplikowane hasła.
  - **Wzmocnione hasła:** Minimum 8 znaków, ale zalecane co najmniej 64 znaki dla obsługi długich haseł frazowych. Powinny zawierać mieszankę liter (małych i wielkich), cyfr i symboli.
  - **Menedżery haseł:** Używaj menedżerów haseł do generowania i bezpiecznego przechowywania silnych, unikalnych haseł dla różnych usług, unikając zapisywania ich.
  - **Uwierzytelnianie wieloskładnikowe (MFA):** Używaj więcej niż jednej formy weryfikacji (np. hasło + kod z SMS/aplikacji, klucz sprzętowy).
- U.S. National Institute for Standards and Technology (NIST) publikuje aktualne wytyczne dotyczące haseł.

31.5 Utwardzanie Interfejsów Administracyjnych

Zapobieganie atakom na interfejsy webowe/API:

- **SQLi (SQL Injection):**
  - Używaj **bezpiecznych API** (np. Prepared Statements, Parameterized Queries), które oddzielają dane wejściowe od poleceń SQL.
  - Waliduj i sanityzuj dane wejściowe.
- **XXE (XML External Entity injection):**
  - Najbezpieczniejszym sposobem jest **wyłączenie przetwarzania zewnętrznych encji XML i DTD** w parserze XML aplikacji.
- **XSS (Cross-Site Scripting):**
  - **Ucieczka (Escaping):** Koduj specjalne znaki HTML/JS w danych wyjściowych, aby przeglądarka interpretowała je jako tekst, a nie kod (np. '<' staje się '<').
  - **Walidacja danych wejściowych (Validating Input):** Używaj białych list (whitelisting) do akceptowania tylko znanych, bezpiecznych znaków/formatów. Odrzucaj lub sanityzuj wszystko inne.
  - **Sanityzacja (Sanitizing):** Usuwaj potencjalnie szkodliwe znaczniki HTML/JS z danych wejściowych przed ich przetworzeniem lub wyświetleniem.
- Stosuj kombinację przeglądu kodu, testów statycznych (SAST) i dynamicznych (DAST) do wykrywania i naprawy luk.

32 Podsumowanie

- **Lokalna luka w aplikacji IoT** to słabość, którą atakujący może wykorzystać do naruszenia bezpieczeństwa aplikacji.
- Systemy mobilne (iOS, Android) są stosunkowo bezpieczne, ale mogą zostać skompromitowane, zwłaszcza przez zainstalowane aplikacje i ich interakcje z siecią.

- Zabezpieczenie aplikacji chmurowych wymaga narzędzi takich jak analiza statyczna kodu, skanery aplikacji webowych i ochrona w czasie rzeczywistym (runtime protection).
- Aplikacje webowe, chmurowe i ich interfejsy muszą przestrzegać ścisłych wytycznych bezpieczeństwa.
- **Modelowanie zagrożeń** jest kluczowe dla zapewnienia bezpieczeństwa aplikacji.
- **Protokół komunikacyjny** definiuje funkcje i zasady transmisji wiadomości między urządzeniami.
- **MQTT** jest używane do monitorowania wielu małych urządzeń i udostępniania ich danych w chmurze do podejmowania decyzji M2M (model pub/sub).
- **CoAP** używa modelu klient-serwer, gdzie klient żąda danych od serwera (często do raportowania zmian stanu).
- Inne protokoły IoT to **XMPP, DDS, AMQP** i problematyczny **UPnP**.
- Bezpieczeństwo **MQTT** zależy od możliwości brokera i klientów (Client ID, user/pass + TLS, certyfikaty, szyfrowanie payloadu).
- Do zabezpieczenia **CoAP** używa się **DTLS**.
- Zainfekowane urządzenie może wykorzystać **UPnP** do uzyskania nieautoryzowanego dostępu (dlatego zaleca się wyłączenie UPnP).
- Należy wzmacniać i chronić **hasła** zgodnie z wytycznymi (zmiana domyślnych, passphrases, menedżery haseł, MFA).
- **Utwardzanie interfejsów administracyjnych** wymaga połączenia przeglądu kodu, testów statycznych i dynamicznych oraz stosowania technik zapobiegania (np. escaping, walidacja, bezpieczne API).

32.1 Wprowadzenie: Programy Bug Bounty

- **Łowcy Nagród za Błędy (Bug Bounty Hunters):** Utalentowani etyczni hakerzy zatrudniani przez firmy (często za pośrednictwem platform crowdsourcingowych) do testowania sieci i systemów ich klientów w poszukiwaniu luk w zabezpieczeniach.
- Firmy zyskują dostęp do szerokiego grona kreatywnych talentów hakerskich.
- **HackerOne** jest jedną z pierwszych firm świadczących takie usługi.
- Testowane są również podstawowe technologie internetowe (np. OpenSSL), serwery (Nginx, Apache) oraz języki programowania (PHP, Python, Perl).
- Platformy te angażują setki tysięcy hakerów i wypłacają miliony dolarów nagród za znalezione błędy.

32.2 Definicja i Cele Oceny Podatności

- **Ocena podatności** identyfikuje słabości (podatności) w systemach, które prawdopodobnie zostaną wykorzystane przez atakujących (threat actors).
- Może być przeprowadzana rutynowo, regularnie lub być ukierunkowana na konkretne komponenty systemu IoT.
- Często wykonywana przy użyciu gotowych narzędzi, np. dostępnych w dystrybucji **Kali Linux**.

32.3 Typy Oceny Podatności

Klasyfikacja oparta na poziomie wiedzy testera o systemie:

- **White Box (Biała Skrzynka):**
  - Oceniający mają pełną wiedzę o systemach sieciowych, często działają od wewnątrz organizacji.
  - Zazwyczaj skupiają się na konkretnych aspektach systemu.
  - Pełna wiedza o architekturze i kodzie źródłowym.
- **Black Box (Czarna Skrzynka):**
  - Ocena najbardziej zbliżona do rzeczywistego ataku.
  - Oceniający (często strona trzecia) nie mają żadnej wiedzy o architekturze sieciowej celu.
  - Symuluje atak zewnętrznego hakera.
- **Gray Box (Szara Skrzynka):**
  - Testerzy mają częściową wiedzę o systemach sieciowych, np. dostęp do dokumentacji architektury wewnętrznej.
  - Cel: weryfikacja podatności, określenie łatwości ich wykorzystania oraz potencjalnego wpływu eksploitów.
  - Połączenie perspektywy wewnętrznej i zewnętrznej.

32.4 Testy Penetracyjne (Penetration Testing - Pen Testing)

- Polegają na przeprowadzaniu **rzeczywistych, ukierunkowanych ataków** w celu odkrycia potencjalnego wpływu znanych podatności.
- Wykwalifikowani etyczni hakerzy wcielają się w rolę atakujących i przeprowadzają ataki, które mają naśladować działania złośliwych hakerów.
- Często używane w testach typu **black box** (hakerzy działają bez wiedzy o wewnętrznym działaniu systemu).
- Używane również do potwierdzenia istnienia podatności zidentyfikowanych podczas innych ocen (testy **gray box**).
- Służą do potwierdzenia skuteczności wdrożonych środków zaradczych w eliminowaniu podatności.

32.5 Narzędzia do Oceny Podatności

32.5.1 Narzędzia do Mapowania Portów

- Służą do odkrywania otwartych portów na systemach końcowych i urządzeniach sieciowych.
- Przykłady: **Nmap**, **Netcat**, SolarWinds Port Scanner.
- **Zenmap** (graficzny interfejs Nmapa):
  - Może dostarczyć szczegółowych informacji o pojedynczym systemie lub segmencie sieci.
  - Odkrywa hosty w sieci.
  - Raportuje otwarte porty.

- Identyfikuje systemy operacyjne działające na hostach.
- Ujawnia szczegóły usług działających na otwartych portach, w tym wersje oprogramowania (proces znany jako **fingerprinting**).

32.5.2 Narzędzia do Oceny Podatności Haseł

- Słabe hasła na portalach aplikacji IoT stanowią poważne zagrożenie.
- Popularne metody ataku na hasła:
  - **Brute force:** Czasochłonna, zautomatyzowana metoda próbowania wszystkich możliwych kombinacji liter, cyfr i symboli.
  - **Atak słownikowy (Dictionary attack):** Wykorzystuje listy słów (słowniki), które mogą być używane jako hasła.
  - **Podsłuchiwanie i łamanie haseł (Password sniffing and cracking):**
    * Analizatory protokołów mogą przechwytywać ruch uwierzytelniający zawierający haszowane hasła.
    * Haszowane hasła mogą być również znalezione w systemach plików urządzeń IoT.
    * Narzędzia takie jak **John the Ripper** i **Aircrack-NG** mogą być używane do próby złamania szyfrowania haszy.
- **Zapobieganie atakom brute force:** Ograniczenie liczby nieudanych prób uwierzytelnienia przed zablokowaniem konta.
- **Bezpieczeństwo nazwy użytkownika:** Blokada konta może być wykorzystana do złośliwego ataku DoS na legalnych użytkowników.
- **Zaawansowane metody uwierzytelniania:** Konieczne w systemach o wysokim ryzyku (np. szkody fizyczne), jak **Przemysłowe Systemy Sterowania (IICSs)**.
- **FIDO Alliance (Fast IDentity Online):** Rozwija nowe technologie i standardy uwierzytelniania dla IoT.
- **NIST (National Institute of Standards and Technology):** Posiada wytyczne dotyczące tożsamości cyfrowych.

32.5.3 Narzędzia do Oceny Podatności Aplikacji Webowych

Oprócz narzędzi z Kali Linux, inne popularne to:

- **OWASP ZAP (Zed Attack Proxy):** Darmowe narzędzie open-source do oceny podatności aplikacji webowych, często używane do testów black box. OWASP jest głównym źródłem wiedzy o podatnościach webowych.
- **OpenVAS (Open Vulnerability Assessment System):** Framework łączący wiele narzędzi skanujących w jedną aplikację z przechowywaniem danych o podatnościach, planowaniem skanów i raportowaniem.
- **Burp Suite:** Kompleksowy zestaw narzędzi do testowania podatności aplikacji webowych. Może identyfikować luki z listy OWASP Top 10. Zawiera skaner, konfigurowalne narzędzie do automatycznych ataków i web crawlera do mapowania systemu plików aplikacji.

32.6 Usługi Oceny Podatności (SECaaS - Security as a Service)

- Firmy SECaaS oferują szeroki zakres zarządzanych usług bezpieczeństwa, w tym skanowanie podatności.
- Przykłady dostawców: AlienVault, Qualys, Mandiant.
- **Cisco** oferuje ocenę penetracji sieci jako część swojego portfolio produktów i usług bezpieczeństwa.

32.7 Źródła Informacji o Podatnościach

- Wiele źródeł analizy zagrożeń (threat intelligence) pracuje nad odkrywaniem, badaniem i rozpowszechnianiem informacji o zagrożeniach.
- **Cisco Talos Intelligence Group:** Jeden z największych komercyjnych zespołów analizy zagrożeń na świecie, chroniący klientów Cisco przed znanymi i nowymi zagrożeniami.
- **NIST National Vulnerability Database (NVD):** Rozszerza bazę **Common Vulnerabilities and Exposures (CVE)** o dodatkową analizę, bazę danych i zaawansowaną wyszukiwarkę.
- Dostawcy sprzętu i oprogramowania powinni informować klientów i opinię publiczną o podatnościach w swoich produktach oraz udostępniać poprawki.

33 Ocena Ryzyka i Podejścia (Risk Assessment Concepts and Approaches)

33.1 Ocena Ryzyka w IoT

- Wiele podejść do bezpieczeństwa **Technologii Operacyjnej (OT - Operational Technology)** koncentruje się na bezpieczeństwie ludzi i sprzętu.
- Komunikacja OT to często M2M (Machine-to-Machine) z ludźmi monitorującymi i kontrolującymi systemy przemysłowe, energetyczne, środowiskowe, smart city itp.
- IoT wprowadza **nowy zestaw ryzyk bezpieczeństwa** ze względu na ogromną powierzchnię ataku (attack surface).
- Przykład: Kompromitacja krytycznych komponentów infrastruktury przemysłowej podłączonych do internetu.

33.2 Podatność a Ryzyko

- **Podatności (Vulnerabilities):** Słabości w oprogramowaniu i systemach, które mogą być wykorzystane przez atakujących w cyberatakach.
- **Ryzyka (Risks):** Podatności ocenione w kontekście konkretnej organizacji.
  - Dana podatność może mieć większy wpływ na jeden typ organizacji niż na inny.
- **Poziom ryzyka** zależy od:
  - Wartości zasobu (asset).
  - Podatności tego zasobu w kontekście używanego oprogramowania i systemów.

  - Prawdopodobieństwa, że zagrożenia zostaną skutecznie wykorzystane przeciwko temu zasobowi.

- Formuła: **Zagrożenie (Threat) + Zasób (Asset) + Podatność (Vulnerability) = Ryzyko (Risk)**

33.3 Myślenie o Ryzyku (Kluczowe Pytania)

Określenie ryzyka wymaga odpowiedzi na następujące pytania w ramach oceny ryzyka:

- Kim są atakujący, którzy chcą nas zaatakować?
- Jakie podatności mogą wykorzystać atakujący?
- Jak udany atak wpłynąłby na organizację? (Konsekwencje: utrata reputacji, danych, przewagi, sprzętu, bezpieczeństwa, inne)
- Jakie jest prawdopodobieństwo wystąpienia różnych ataków?
- Co organizacja może zrobić, aby zaradzić ryzyku?

33.4 Common Vulnerability Scoring System (CVSS)

33.4.1 Cel i Definicja

- **CVSS** to system oceny ryzyka zaprojektowany do przekazywania wspólnych atrybutów i dotkliwości podatności w systemach komputerowych (sprzęt i oprogramowanie).
- **CVSS 3.0** jest neutralnym dla dostawców, branżowym standardem i otwartym frameworkiem do ważenia ryzyka podatności przy użyciu różnych metryk.
- Wynik liczbowy CVSS może być użyty do określenia **pilności** podatności i **priorytetu** jej zaadresowania.
- Pierwotnie zaprojektowany dla bezpieczeństwa IT, **nie obejmuje metryk związanych z bezpieczeństwem fizycznym (safety)**. Przyszłe systemy oceny powinny uwzględniać dodatkowe metryki specyficzne dla implementacji IoT.

33.4.2 Grupy Metryk CVSS

Narzędzie CVSS wymaga od oceniającego wybrania wartości w trzech grupach metryk dla każdej zidentyfikowanej podatności:

1. **Grupa Metryk Podstawowych (Base Metric Group):** Reprezentuje cechy podatności, które są **stałe w czasie i niezależne od kontekstu**.
   - *Metryki Wykorzystania (Exploitability):* Cechy eksploita, takie jak wektor ataku, złożoność, wymagane uprawnienia, interakcja użytkownika.
   - *Metryki Wpływu (Impact):* Skutki eksploita związane z triadą CIA (Poufność, Integralność, Dostępność).
2. **Grupa Metryk Czasowych (Temporal Metric Group):** Mierzy cechy podatności, które **mogą zmieniać się w czasie**, ale nie w różnych środowiskach użytkownika (np. dostępność exploita, poziom poprawek).
3. **Grupa Metryk Środowiskowych (Environmental Metric Group):** Mierzy aspekty podatności, które są **specyficzne dla środowiska danej organizacji** (np. wymagania dotyczące bezpieczeństwa, zmodyfikowane metryki podstawowe).

33.4.3 Metryki Grupy Podstawowej (Base Metric Group)

- **Metryki Wykorzystania (Exploitability):**
  - *Wektor Ataku (Attack Vector - AV):* Odzwierciedla bliskość atakującego (np. sieć, sąsiednia sieć, lokalny, fizyczny).
  - *Złożoność Ataku (Attack Complexity - AC):* Wyraża liczbę komponentów (oprogramowanie, sprzęt, sieci), które są poza kontrolą atakującego i muszą być obecne.
  - *Wymagane Uprawnienia (Privileges Required - PR):* Określa poziom dostępu wymagany do przeprowadzenia ataku.
  - *Interakcja Użytkownika (User Interaction - UI):* Wyraża obecność lub brak wymogu interakcji użytkownika.
  - *Zakres (Scope - S):* Wyraża, czy podatność w jednym komponencie może wpłynąć na zasoby w innym zakresie uprawnień (authority scope).
- **Metryki Wpływu (Impact):** (mierzone w odniesieniu do komponentu, którego dotyczy podatność)
  - *Wpływ na Poufność (Confidentiality Impact - C):* Mierzy wpływ na poufność danych.
  - *Wpływ na Integralność (Integrity Impact - I):* Mierzy wpływ na integralność danych.
  - *Wpływ na Dostępność (Availability Impact - A):* Mierzy wpływ na dostępność zasobu.

33.4.4 Proces CVSS

- Grupa Metryk Podstawowych (Base Metrics Group) służy do oceny podatności w oprogramowaniu i sprzęcie. Opisuje dotkliwość podatności na podstawie cech udanego jej wykorzystania.
- Pozostałe grupy metryk (Czasowa i Środowiskowa) modyfikują podstawową ocenę dotkliwości, uwzględniając, jak na ocenę wpływa czas i czynniki środowiskowe.
- Proces CVSS wykorzystuje narzędzie zwane **Kalkulatorem CVSS v3.0**.
- Kalkulator działa jak kwestionariusz, w którym dokonuje się wyborów opisujących podatność dla każdej grupy metryk, a następnie generowany jest wynik.
- Oprócz wyniku liczbowego generowany jest również **ciąg wektorowy (vector string)**, który podsumowuje dokonane wybory.
- Wartości metryk Czasowych i Środowiskowych modyfikują wyniki metryk Podstawowych, dając ostateczny, całościowy wynik.

34 Ocena Ryzyka za pomocą Modelowania Zagrożeń (Assessing Risk with Threat Modeling)

34.1 Modelowanie Zagrożeń - Dogłębnie

- **Modelowanie zagrożeń** to proaktywne podejście do oceny bezpieczeństwa systemów i oprogramowania. Najlepiej stosować je **podczas całego procesu rozwoju**.
- Trzy podejścia do modelowania zagrożeń:
  - **Skoncentrowane na ataku (Attack-centric):** Z punktu widzenia atakującego.

  - **Skoncentrowane na obronie (Defense-centric):** Analizuje architekturę systemu w celu identyfikacji zagrożeń dla różnych elementów. **(Używane w tym kursie)**
  - **Skoncentrowane na zasobach (Asset-centric):** Skupia się na klasyfikacji zasobów i przypisywaniu im wartości.

- Proces modelowania zagrożeń zaczyna się od zrozumienia systemu poprzez odpowiedzi na pytania:
  - Co modelujemy? Jakie są potencjalne zagrożenia?
  - Jakie są ryzyka? Co można zrobić, aby im zaradzić?

34.2 Proces Modelowania Zagrożeń (Kroki)

34.2.1 Krok 1: Cele Bezpieczeństwa Systemu (Identify Security Objectives)

- Określenie celów bezpieczeństwa systemu na podstawie jego przeznaczenia i działania.
- Zrozumienie, jaki rodzaj danych jest przetwarzany przez system i jakie są konsekwencje kradzieży lub zniszczenia danych.
  - Czy utrata danych spowoduje straty finansowe? W jakim stopniu?
  - Czy reputacja firmy zostanie naruszona? Jakie będą skutki biznesowe?
- Rządy i inne organizacje wprowadzają regulacje dotyczące gromadzenia, przesyłania i przechowywania danych.
  - Naruszenie tych regulacji może skutkować poważnymi karami finansowymi i prawnymi.
- Systemy infrastruktury krytycznej muszą być zawsze dostępne; przerwy mogą mieć poważne konsekwencje.

34.2.2 Krok 2: Mapowanie Przepływów Danych (Document IoT System Architecture / Map Data Flows)

- Po zidentyfikowaniu celów bezpieczeństwa, należy stworzyć diagram funkcji architektury systemu.
- **Diagramy Przepływu Danych (Data Flow Diagrams - DFDs)** są niezwykle przydatne do wizualizacji systemu IoT.
- DFD przedstawiają ścieżki, którymi dane będą przemieszczać się między różnymi komponentami funkcjonalnymi systemu, w tym punkty wejścia do systemu oraz urządzenia i osoby korzystające z tych punktów wejścia.

34.2.3 Krok 3: Dekompozycja Systemu (Decompose the IoT System)

(Ten krok jest często połączony z tworzeniem DFD)

- Identyfikacja głównych komponentów systemu IoT w DFD:
  - Urządzenia IoT (czujniki, aktuatory)
  - Bramy IoT (umożliwiają przesyłanie danych z czujników przez sieć IP)
  - Aplikacje lokalne
  - Urządzenia brzegowe (Edge devices - umożliwiają przesyłanie wewnętrznego ruchu IP między lokalizacjami a internetem/chmurą)

  - Aplikacje danych
  - Magazyny danych (Data storage)
  - Aplikacje sterujące (przetwarzają dane w celu podejmowania decyzji i sterowania)
  - Aplikacje mobilne
- DFD używają symboli do reprezentowania tych elementów. Ten kurs używa symboli **Yourdon and Coad**.
- **Podstawowe symbole DFD (Yourdon/Coad):**
  - *Encja Zewnętrzna (External Entity):* Prostokąt - Użytkownicy, systemy poza kontrolą.
  - *Proces (Process):* Okrąg/Elipsa - Transformacja danych (np. odczyt z czujnika, analiza).
  - *Magazyn Danych (Data Store):* Dwie równoległe linie - Dane w spoczynku (np. baza danych).
  - *Przepływ Danych (Data Flow):* Strzałka - Ruch danych między komponentami.
- **Podstawowe zasady DFD:**
  - Każdy proces powinien mieć co najmniej jedno wejście i jedno wyjście.
  - Magazyny danych powinny mieć przepływy do zapisu i odczytu.
  - Dane przechowywane w systemie muszą przejść przez co najmniej jeden proces.

34.2.4 Strefy Systemu (Zones of the System)

(Część dekompozycji i analizy architektury)

- **Strefy (Zones)** można zdefiniować jako obszary systemu wymagające różnej autoryzacji i uwierzytelniania.
- Pomagają ograniczyć ekspozycję różnych części systemu na podatności związane z daną strefą.
- Przykłady stref: obszar czujników sieci, aplikacje webowe, brama IP, brzeg sieci itp.
- Strefy mogą być zagnieżdżone, gdy komponenty znajdują się w innej organizacji.

34.2.5 Granice Zaufania (Determine Trust Boundaries)

(Część dekompozycji i analizy architektury)

- **Granice zaufania (Trust Boundaries)** oddzielają sekcje sieci, w których poziom zaufania między encjami na obu końcach przepływu danych jest różny.
- Przykład: Dane płynące z Bramy IoT do Bramy Chmurowej przekraczają granicę zaufania.
- Uprawnienia dla Bramy IoT są inne niż dla Bramy Chmurowej (która jest wystawiona na internet i dostępna dla wielu użytkowników).
- Ruch danych przekraczający granicę zaufania musi być **autoryzowany i uwierzytelniony** na urządzeniu wejściowym.

34.2.6 Krok 4: Identyfikacja Zagrożeń i Ocena Ryzyka (Identify and Rate Threats)

- Po dekompozycji systemu następuje identyfikacja potencjalnych zagrożeń dla każdego elementu.
- Metodologie takie jak **STRIDE** i **DREAD** pomagają w kategoryzacji i ocenie zagrożeń.

STRIDE

- Podejście STRIDE dostarcza zestaw kategorii pomocnych w identyfikacji potencjalnych zagrożeń w systemach IoT.
- STRIDE to akronim od:
  - **S**poofing (Podszywanie się): Udawanie innego użytkownika lub urządzenia.
  - **T**ampering (Manipulacja): Modyfikacja danych, kodu lub urządzenia.
  - **R**epudiation (Zaprzeczenie): Niemożność udowodnienia lub zaprzeczenia zdarzenia.
  - **I**nformation Disclosure (Ujawnienie informacji): Udostępnienie informacji nieuprawnionym stronom.
  - **D**enial of Service (Odmowa usługi): Uniemożliwienie działania urządzenia lub usługi.
  - **E**levation of Privilege (Podniesienie uprawnień): Uzyskanie wyższych uprawnień niż normalnie autoryzowane.
- STRIDE dostarcza schemat klasyfikacji do identyfikacji zagrożeń dla każdego elementu DFD.
- Zrozumienie, które podatności są istotne dla których elementów systemu, pomaga oszczędzić czas w procesie modelowania zagrożeń.

DREAD

- Każde zagrożenie zidentyfikowane przez STRIDE musi być ocenione pod kątem stopnia ryzyka dla organizacji.
- Model **DREAD** daje ilościowy wynik ryzyka.
- DREAD to akronim od kategorii oceny:
  - **D**amage potential (Potencjał szkód): Jakie są potencjalne szkody, jeśli zagrożenie zostanie wykorzystane?
  - **R**eproducibility (Odtwarzalność): Jak łatwo jest odtworzyć atak?
  - **E**xploitability (Podatność na wykorzystanie): Jak trudno jest wykorzystać podatność?
  - **A**ffected users (Dotknięci użytkownicy): Jaka jest skala ataku? Ilu użytkowników jest dotkniętych?
  - **D**iscoverability (Wykrywalność): Jak trudno jest odkryć podatność?
- Wynik DREAD może być użyty wraz z oceną kosztów ryzyka do oceny zasadności i wykonalności mitygacji zagrożenia.
- **Model Oceny DREAD (przykład skali 1-3):**
  - *Wysokie (3):* np. system nie działa, łatwe do odtworzenia/wykorzystania, dotyczy wielu użytkowników, powszechnie znane.
  - *Średnie (2):* np. utrata ważnych danych, działa w połowie przypadków, wymaga umiejętności, dotyczy niektórych urządzeń, mało znane.
  - *Niskie (1):* np. niewielka utrata danych, trudne do odtworzenia, wymaga dużych umiejętności/specyficznych warunków, dotyczy niewielu użytkowników, mało interesujące.
- Mogą być wymagane niewielkie modyfikacje, aby zastosować te modele (pierwotnie dla oprogramowania) do systemów IoT.
- **Przykład oceny DREAD:** Podatność w rejestratorze DVR pozwalająca na przejęcie kontroli przez spreparowane ciasteczko HTTP. Ocena (w skali 1-3): D=2, R=3, E=3, A=3, D=3. Suma = 14 (Ryzyko Wysokie). Mimo wysokiego ryzyka, firma może nie aktualizować oprogramowania z powodu kosztów.

34.2.7 Krok 5: Rekomendowanie Środków Zaradczych (Recommend Mitigation)

- **Bezpieczny rozwój i wdrożenie** systemów IoT to najskuteczniejszy sposób na mitygację ryzyka.
- Wytyczne dla bezpiecznego projektowania:
  - Rozważ powierzchnię ataku (wdrożenia IoT mają dużą powierzchnię ataku na warstwie urządzeń).
  - Rozważ urządzenia zbudowane z bezpiecznymi domyślnymi konfiguracjami i mechanizmami aktualizacji.
  - Rozważ polityki bezpieczeństwa partnerów i usług stron trzecich.
  - Uwzględniaj bezpieczeństwo we wszystkim ("security in all things").
- Ogólne strategie mitygacji:
  - Utrzymuj aktualne oprogramowanie układowe urządzeń (firmware).
  - Utrzymuj aktualne oprogramowanie aplikacyjne.
  - Oddzielaj ruch IT i IoT w sieciach organizacji.
  - Zapewnij bezpieczeństwo fizyczne, gdy tylko jest to możliwe.
  - Używaj bezpiecznych protokołów komunikacyjnych i szyfrowania.
  - Angażuj personel ds. bezpieczeństwa sieci w organizacji.

35 Zarządzanie Ryzykiem w Systemach IoT (Managing Risk in IoT Systems)

35.1 Strategie Zarządzania Ryzykiem

- **NIST Risk Management Framework (RMF)** to proces cykliczny i ciągły.
- Identyfikacja Ryzyka i Ocena Ryzyka są równoległe do podejścia modelowania zagrożeń. RMF zamyka cykl, włączając **reakcję na ryzyko, ocenę reakcji i działania oceny odpowiedzi**.
- Identyfikacja i dopasowanie zagrożeń do podatności nazywa się **parowaniem zagrożenie-podatność (threat-vulnerability (T-V) pairing)**.
- Pary T-V mogą być używane jako **linia bazowa (baseline)** do wskazania ryzyka przed wdrożeniem kontroli bezpieczeństwa.
  - Linia bazowa może być porównywana z bieżącymi ocenami ryzyka w celu oceny skuteczności zarządzania ryzykiem.
  - Określa to wrodzony profil ryzyka organizacji.
- Ryzyka mogą być oceniane punktowo lub ważone w celu priorytetyzacji strategii redukcji ryzyka.

35.2 Reagowanie na Ryzyko (Risk Response)

Cztery "T"reakcji na ryzyko:

1. **Unikanie Ryzyka (Terminate / Avoidance):** Zaprzestanie działań, które tworzą ryzyko.
2. **Redukcja Ryzyka (Treat / Reduction):** Zmniejszenie ryzyka poprzez podjęcie środków w celu zmniejszenia podatności.
3. **Dzielenie się Ryzykiem (Transfer / Sharing):** Przeniesienie części ryzyka na inne strony (np. ubezpieczyciele, dostawcy SECaaS).
4. **Zatrzymanie Ryzyka (Tolerate / Retention):** Akceptacja ryzyka i jego konsekwencji. Brak działań w celu redukcji ryzyka.

- Wybór odpowiedniej reakcji polega na ważeniu potencjalnego wpływu ryzyka w stosunku do prawdopodobieństwa jego wystąpienia.

36 Wprowadzenie do Blockchain

36.1 Obietnica Blockchain

- Blockchain jest najbardziej znany jako technologia stojąca za **Bitcoinem**.
- Zyskuje duże zainteresowanie jako sposób na bezpieczne przeprowadzanie transakcji, w tym wymiany informacji przez urządzenia IoT.
- Blockchain to technologia, która rozwiązuje **problem zaufania**.
- Jest to **rozproszony rejestr (distributed ledger)**, którego ciągle rosnąca lista rekordów, zwanych **blokami**, jest połączona i zabezpieczona za pomocą kryptografii.
- Kluczowe cechy:
  - **Niezmienność (Immutable):** Po zapisaniu, dane w bloku nie mogą być zmienione.
  - **Dystrybucja/Decentralizacja:** Rejestr jest kopiowany i przechowywany przez wielu uczestników.

36.2 IoT i Blockchain

- Zarówno IoT, jak i blockchain są technologiami **przełomowymi (disruptive)**.
- **Cisco Systems** jest jednym z czołowych członków **Trusted IoT Alliance**, konsorcjum firm pomagających ustanowić standardowy protokół dla rozwiązań bezpieczeństwa IoT opartych na blockchain.

36.3 Obecne Systemy Zaufania

- Aby zrozumieć blockchain, warto najpierw przyjrzeć się, jak działa zaufanie w obecnym systemie monetarnym.
- Przy zakupie towarów/usług obie strony zgadzają się na metodę płatności (gotówka, karta, czek).
- Polegamy na **stronie trzeciej (pośredniku)**, np. banku, aby zagwarantować transakcję finansową między kupującym a sprzedającym (zazwyczaj pobierając opłatę).

- Te transakcje są zazwyczaj rejestrowane w **pojedynczym, scentralizowanym rejestrze**, któremu ufamy, że jest dokładnie prowadzony.
- Zaufanie, jakie zapewniają ci pośrednicy, obejmuje:
  - Uwierzytelnianie osoby dokonującej transakcji.
  - Zapewnienie dokładności wszystkich transakcji zapisanych w rejestrze.
  - Niedopuszczanie do nielegalnych transakcji.

36.4 System Zaufania Blockchain

- Blockchain osiąga zaufanie w zupełnie inny sposób.
- Kryptowaluty takie jak Bitcoin **nie używają pośrednika** do zapewnienia zaufania transakcji.
- Zamiast tego Bitcoin używa samego blockchaina do zapewnienia zaufania między kupującym a sprzedającym.
- Może to być zastosowane do dowolnego typu aplikacji wykorzystującej transakcje lub rejestr.

37 Jak Działa Blockchain

37.1 Cechy Blockchain

- Blockchain to ciągle rosnąca lista transakcji w formie bloków. Bloki te są połączone i zabezpieczone za pomocą kryptografii.
- Blockchain wykorzystuje:
  - **Podpisy cyfrowe (Digital signatures)**
  - **Zdecentralizowany rejestr (Decentralized ledger)**
  - **Algorytm osiągania konsensusu (Algorithm for reaching consensus)**
- Każdy blok zawiera **hash poprzedniego bloku**, tworząc łańcuch bloków znany jako blockchain.

37.2 Podpis Cyfrowy (Digital Signature)

- Schemat matematyczny służący do demonstracji autentyczności informacji cyfrowej.
- Podpis cyfrowy **nie może być skopiowany**, ponieważ jest zawsze inny (zależny od wiadomości).
- Wykorzystuje **wiadomość (lub transakcję)** do wygenerowania podpisu.
- Nawet niewielka zmiana wiadomości powoduje całkowitą zmianę podpisu cyfrowego.
- Podpisy cyfrowe obejmują: **wiadomość (transakcję)**, **klucz prywatny** (używany do podpisania) i **klucz publiczny** (używany do weryfikacji). Weryfikacja potwierdza, że wiadomość została stworzona przez posiadacza klucza prywatnego.

37.3 Zdecentralizowany Rejestr (Decentralized Ledger)

- Blockchain używa zdecentralizowanego rejestru, gdzie **wszystkie zainteresowane strony utrzymują kopię**.
- Zaufanie jest zapewnione przez to, że **każdy otrzymuje i akceptuje (wierzy w) każdą nową transakcję**.
- Wszyscy muszą używać i pracować z **dokładnie tym samym rejestrem**. Odbywa się to za pomocą procesu znanego jako **Proof of Work (Dowód Pracy)**.

37.4 Osiąganie Konsensusu (Reaching Consensus)

- Blok zawiera transakcje wraz z ich podpisami cyfrowymi.
- Walidacja transakcji w bloku wykorzystuje proces znany jako **Proof of Work (PoW)**.
- **PoW** to algorytm (oparty na haszowaniu) wykonywany przez komputery (**górników - miners**), który wymaga dużej mocy obliczeniowej w relatywnie krótkim czasie.
- Blockchain składa się z bloków. Każdy blok to lista transakcji, z hashem poprzedniego bloku i hashem bieżącego bloku (zawierającym jego PoW).
  - Hash jest obliczany przy użyciu hasha poprzedniego bloku (prior PoW) oraz wszystkich transakcji w bieżącym bloku wraz z ich podpisami cyfrowymi.
  - To sprawia, że **obliczeniowo niewykonalne (computationally infeasible)** jest zmodyfikowanie bloku lub zmiana kolejności bloków.

37.5 Zastosowanie Blockchain w Bezpieczeństwie IoT

Blockchain może pomóc rozwiązać wiele wyzwań związanych z bezpieczeństwem i zaufaniem w IoT:

- Śledzenie pomiarów danych z czujników i zapobieganie złośliwym danym.
- Zapewnienie identyfikacji, uwierzytelniania i bezpiecznego transferu danych urządzeń IoT.
- Umożliwienie czujnikom IoT bezpiecznej wymiany danych bezpośrednio między sobą, bez pośrednika.
- Rozproszony rejestr eliminuje pojedynczy punkt awarii (single source of failure) w ekosystemie IoT.
- Uproszczenie wdrażania i redukcja kosztów operacyjnych IoT dzięki braku pośrednika.
- Urządzenia IoT są bezpośrednio adresowalne za pomocą blockchain, zapewniając niezmienną historię.