

## Lekcja

# Temat: Wprowadzenie do cyberbezpieczeństwa

**Cyberbezpieczeństwo (ang. cybersecurity)** to zbiór praktyk, procesów, technologii i środków organizacyjnych mających na celu ochronę systemów informatycznych, sieci, programów, urządzeń i danych przed atakami, uszkodzeniem lub nieautoryzowanym dostępem. Jego celem jest zapewnienie poufności, integralności i dostępności informacji (tzw. triada CIA):

- **C (Confidentiality)** – Poufność: dane dostępne tylko dla uprawnionych.
- **I (Integrity)** – Integralność: dane są kompletne, dokładne i niezmienione bez autoryzacji.
- **A (Availability)** – Dostępność: systemy i dane są dostępne dla uprawnionych użytkowników, gdy są potrzebne.

## Zastosowanie w informatyce

**Cyberbezpieczeństwo znajduje zastosowanie w każdej dziedzinie IT, m.in.:**

- **Ochrona infrastruktury sieciowej** (routery, przełączniki, firewalle).
- **Bezpieczeństwo aplikacji** (testy penetracyjne, bezpieczny kod, ochrona przed atakami na aplikacje webowe).
- **Bezpieczeństwo danych** (szifrowanie, zarządzanie dostępem, backup).
- **Bezpieczeństwo chmury obliczeniowej** (ochrona danych i aplikacji hostowanych w chmurze).
- **Bezpieczeństwo operacyjne** (monitoring, zarządzanie incydentami, polityki bezpieczeństwa).
- **Bezpieczeństwo urządzeń końcowych** (komputery, smartfony, IoT).

- **Bezpieczeństwo tożsamości i dostępów** (uwierzytelnianie wieloskładnikowe – MFA, zarządzanie tożsamością).

## Ochrona infrastruktury sieciowej (routery, przełączniki, firewalle).

Infrastruktura sieciowa to m.in.: **routery, przełączniki, firewalle**.  
Każde z tych urządzeń ma inną rolę w ochronie sieci.

### 2 Router – pierwsza linia obrony w sieci

**Router** łączy sieć lokalną (LAN) z Internetem (WAN).

**Jak chroni:**

#### 1. Filtrowanie ruchu (Access Control Lists – ACL)

- a. Router może blokować ruch przychodzący lub wychodzący na podstawie adresu IP, portu lub protokołu.
- b. Przykład:
  - i. Blokujemy wszystkie połączenia przychodzące na port 23 (Telnet)
  - ii. Pozwalamy tylko na porty 80 (HTTP) i 443 (HTTPS)

#### 2. Network Address Translation (NAT)

- a. Ukrywa wewnętrzne adresy IP przed Internetem
- b. Haker nie widzi dokładnie, jakie urządzenia są w sieci wewnętrznej

#### 3. Routing bezpieczny

- a. Routery mogą używać protokołów, które **chronią przed podszywaniem się w sieci** (np. OSPF authentication)

### 3 Przełącznik (Switch) – ochrona wewnętrznej sieci LAN

**Switch** rozdziela ruch w sieci lokalnej.

**Jak chroni:**

## **1. Segmentacja sieci (VLAN)**

- a. Można podzielić sieć na różne grupy np. IT, księgowość, goście
- b. Atak z jednego VLANu **nie przechodzi automatycznie** do innych VLANów

## **2. Port Security**

- a. Ogranicza dostęp do portu tylko dla zdefiniowanych adresów MAC,
- b. Jeśli ktoś podłączy obcy komputer, nie dostanie dostępu do sieci

## **3. Ochrona przed ARP spoofing i MAC flooding**

- a. Zapobiega podszywaniu się hakerów pod inne urządzenia w LAN

## **4 Firewall – ochrona przed atakami z zewnątrz**

**Firewall** to strażnik między siecią wewnętrzną a Internetem.

**Jak chroni:**

### **1. Blokowanie nieautoryzowanego ruchu**

- a. Firewall przepuszcza tylko ruch zgodny z regułami
- b. Przykład: blokuje połączenia przychodzące spoza zaufanych IP

### **2. Ochrona przed atakami typu DoS / DDoS**

- a. Limituje liczbę połączeń z jednego IP, żeby **nie zablokować serwera przeciążeniem**

### **3. Analiza ruchu i filtrowanie pakietów**

- a. Sprawdza, czy pakiet jest „legalny”
- b. W niektórych firewallach NGFW analizuje **aplikacje i złośliwe wzorce ruchu**

**Bezpieczeństwo aplikacji** (testy penetracyjne, bezpieczny kod, ochrona przed atakami na aplikacje webowe).

## 1 Testy penetracyjne (Penetration Testing)

**Cel:** sprawdzić, czy aplikacja ma luki bezpieczeństwa.

**Przykład:**

- Strona logowania do panelu administracyjnego.
- Tester próbuje wprowadzić w polu hasła:

' OR '1'='1

- Jeśli aplikacja **nie sprawdza poprawnie danych**, tester może się zalogować bez prawidłowego hasła → luka SQL Injection.
- Raport z testu zawiera opis luki i zalecenia: **użycie zapytania parametryzowanego**.

## 2 Pisanie bezpiecznego kodu (Secure Coding)

**Cel:** tworzyć aplikacje w taki sposób, aby były odporne na ataki.

**Przykłady:**

### 1. Weryfikacja danych wejściowych

```
# zamiast wstawiać dane bezpośrednio do SQL  
cursor.execute("SELECT * FROM users WHERE username = ?",  
(username,))
```

- a. Użycie **zapytania parametryzowanego** chroni przed SQL Injection.

### 2. Szyfrowanie wrażliwych danych

```
import hashlib  
password_hash = hashlib.sha256(password.encode()).hexdigest()
```

- a. Hasła przechowywane w bazie są w formie **hasha**, nie w czystym tekście.

### 3. Używanie bezpiecznych protokołów

- a. HTTPS zamiast HTTP

b. TLS/SSL dla komunikacji z API

### 3 Ochrona przed atakami na aplikacje webowe

Typ ataku	Przykład ochrony
SQL Injection	Zapytania parametryzowane, ORM (np. Django ORM)
Cross-Site Scripting (XSS)	Escapowanie danych wyświetlanych w HTML, Content Security Policy (CSP)
Cross-Site Request Forgery (CSRF)	Tokeny CSRF w formularzach
Brute Force	Limitowanie prób logowania, CAPTCHA, MFA
Session Hijacking	Bezpieczne ciasteczka (HttpOnly, Secure), szyfrowanie sesji

### 4 Przykład praktyczny – strona logowania

1. Formularz logowania:
  - a. Użytkownik wpisuje login i hasło
2. Testy penetracyjne wykrywają lukę SQL Injection
3. Programista poprawia kod:
  - a. Używa zapytań parametryzowanych
  - b. Dodaje limit prób logowania i MFA
4. Strona jest teraz odporna na najczęstsze ataki

### 5 Podsumowanie

- **Testy penetracyjne** → sprawdzają luki w aplikacji
- **Bezpieczny kod** → minimalizuje ryzyko ataków (walidacja, szyfrowanie, HTTPS)
- **Ochrona przed atakami webowymi** → XSS, CSRF, SQL Injection, brute force

**Bezpieczeństwo danych** to zestaw działań mających na celu ochronę informacji przed nieautoryzowanym dostępem, utratą lub uszkodzeniem. Obejmuje m.in.:

1. **Szyfrowanie** – dane są kodowane w taki sposób, że bez odpowiedniego klucza nie da się ich odczytać.
  - a. *Przykład:* wiadomości w komunikatorach takich jak Signal czy WhatsApp są szyfrowane end-to-end.
2. **Zarządzanie dostępem** – kontrola, kto i w jakim zakresie może korzystać z danych.
  - a. *Przykład:* w firmie pracownicy działu księgowości mają dostęp do faktur, ale nie do danych kadrowych.
3. **Backup (kopie zapasowe)** – tworzenie kopii danych, aby można je było przywrócić w razie awarii lub utraty.

**Bezpieczeństwo chmury obliczeniowej** polega na ochronie danych, aplikacji i infrastruktury hostowanej w chmurze przed nieautoryzowanym dostępem, utratą lub atakami cybernetycznymi. Obejmuje kilka kluczowych obszarów:

1. **Szyfrowanie danych w chmurze** – chroni informacje zarówno w czasie przesyłania (transmisji), jak i przechowywania (w spoczynku).
  - a. *Przykład:* dane przechowywane w Amazon S3 lub Google Cloud Storage mogą być szyfrowane AES-256.
2. **Kontrola dostępu i uwierzytelnianie** – ogranicza, kto może logować się do chmury i korzystać z aplikacji.
  - a. *Przykład:* użycie uwierzytelniania wieloskładnikowego (MFA) dla kont administracyjnych w Microsoft Azure.
3. **Monitoring i wykrywanie zagrożeń** – systemy monitorują aktywność w chmurze i wykrywają nietypowe działania lub próby włamań.
  - a. *Przykład:* AWS CloudTrail rejestruje wszystkie operacje na zasobach chmurowych, a Amazon GuardDuty analizuje podejrzane zachowania.

4. **Backup i odzyskiwanie danych** – tworzenie kopii zapasowych i planów odzyskiwania po awarii.
  - a. *Przykład:* Google Cloud Backup & DR pozwala przywrócić dane aplikacji w razie przypadkowego usunięcia lub awarii.
5. **Zabezpieczenie aplikacji** – ochrona przed atakami typu SQL injection, XSS, DDoS itp.
  - a. *Przykład:* korzystanie z Web Application Firewall (WAF) w chmurze do filtrowania niebezpiecznych żądań HTTP.

**Bezpieczeństwo operacyjne** to działania i procedury zapewniające ochronę systemów informatycznych w codziennej pracy organizacji. Skupia się na tym, aby procesy, infrastruktura i ludzie pracowali w bezpieczny sposób. Obejmuje m.in.:

1. **Monitoring** – ciągłe śledzenie systemów i sieci w celu wykrywania zagrożeń lub nieprawidłowości.
  - a. *Przykład:* system SIEM (Security Information and Event Management) analizuje logi serwerów i powiadamia administratora o podejrzanych próbach logowania.
2. **Zarządzanie incydentami** – proces reagowania na zdarzenia bezpieczeństwa, tak aby zminimalizować szkody i przywrócić normalne działanie systemu.
  - a. *Przykład:* po wykryciu ataku ransomware dział IT izoluje zainfekowane komputery i przywraca dane z kopii zapasowej.
3. **Polityki bezpieczeństwa** – zestaw zasad i procedur, które określają, jak pracownicy mają postępować, aby chronić dane i systemy.
  - a. *Przykład:* polityka wymaga zmiany hasła co 90 dni i zakazuje korzystania z niezaufanych nośników USB w firmie.

W skrócie: **bezpieczeństwo operacyjne to codzienna „tarcza” organizacji**, która chroni systemy i dane dzięki monitorowaniu, reagowaniu na incydenty i ustalonym zasadom postępowania.

**Bezpieczeństwo urządzeń końcowych** (ang. *endpoint security*) polega na ochronie komputerów, smartfonów, tabletów czy urządzeń IoT przed zagrożeniami takimi jak malware, kradzież danych czy

nieautoryzowany dostęp. Chodzi o to, żeby każde urządzenie podłączone do sieci było bezpieczne, bo stanowi potencjalne wejście dla atakującego.

1. **Ochrona antywirusowa i antymalware** – instalacja programów, które wykrywają i usuwają złośliwe oprogramowanie.

*Przykład:* Windows Defender na komputerach z Windows lub Avast na laptopach i smartfonach.

2. **Aktualizacje systemu i aplikacji** – regularne łatanie luk bezpieczeństwa, które mogą być wykorzystane przez atakujących.

*Przykład:* automatyczne aktualizacje Androida, iOS czy Windowsa, które naprawiają podatności.

3. **Szyfrowanie danych na urządzeniu** – zabezpiecza dane w razie kradzieży lub zgubienia sprzętu.

*Przykład:* BitLocker na komputerach z Windows lub szyfrowanie iPhone (FileVault w MacOS).

4. **Kontrola dostępu i uwierzytelnianie** – blokowanie dostępu osobom nieupoważnionym.

*Przykład:* logowanie do smartfona za pomocą PIN-u, odcisku palca lub Face ID.

5. **Bezpieczeństwo IoT** – zabezpieczenie inteligentnych urządzeń domowych lub przemysłowych.

*Przykład:* aktualizacja oprogramowania inteligentnych kamer, zablokowanie domyślnych haseł w smart TV lub czujnikach IoT.