🔥 Awaria Cloudflare 2025

Jak jeden plik "położył" 16% internetu

Geeks Club - Spotkanie Programistów

📅 10 grudnia 2025 r.

📋 Agenda

1. 🌐 **Dlaczego Cloudflare jest ważny?**
2. 💥 **Co się stało?** - Timeline awarii
3. 🔧 **Analiza techniczna** - ClickHouse, Rust, unwrap()
4. 🎭 **Czynniki mylące** - Dlaczego myśleli, że to atak DDoS
5. 📝 **Wnioski i działania naprawcze**
6. 💭 **Komentarz** - Co my z tego wyciągamy?

🌐 Co to jest Cloudflare?

Middleware między klientem a Twoją aplikacją

📊 Skala Cloudflare

~16% całego ruchu internetowego 🌍

> Każdy co szósty request w internecie przechodzi przez Cloudflare

Znani użytkownicy:

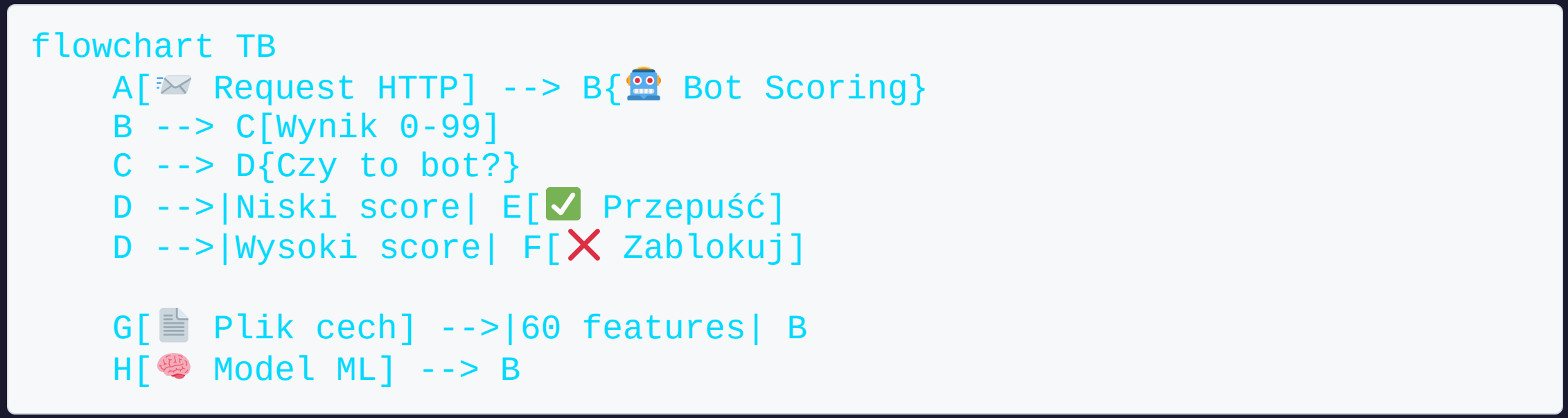
Kategoria	Firmy
🏛️ Technologia	Mozilla, Microsoft Azure, Office 365, IBM
🛒 E-commerce	Nike, H&M, Shopify
💬 Social	Reddit, Digital Ocean

⏰ Timeline awarii

```
timeline
    title 18 listopada 2025 - Awaria Cloudflare (UTC)
    11:05 : Wdrożono zmianę uprawnień w ClickHouse
    11:20 : 🔴 Początek problemów - błędy 5xx
    11:28 : Wdrożenie dociera do produkcji
    11:32 : Analiza - początkowo podejrzenie DDoS
    13:05 : Obejście dla Workers KV i Access
    14:24 : Identyfikacja przyczyny - plik bot managementu
    14:30 : 🟢 Wdrożenie poprawnego pliku
    17:06 : 🟢 Pełna normalizacja
```

🤖 Bot Management - Źródło problemu

Jak działa ocena botów?



Bot Score: 0-99 (im wyżej = większe prawdopodobieństwo bota)

🗃️ Architektura ClickHouse

Bazy danych i shardy

```
flowchart TB
    subgraph Przed awarią
        A1[Zapytanie SQL] --> B1[Baza 'default']
        B1 --> C1[~60 cech]
    end

    subgraph Po zmianie uprawnień
        A2[Zapytanie SQL] --> B2[Baza 'default']
        B2 --> C2[Cechy zagregowane]
        A2 --> D2[Baza 'R0']
        D2 --> E2[Cechy z shardów]
        C2 & E2 --> F2[❌ >200 cech!]
    end
```

🔍 Zapytanie bez dyskryminatora bazy

```sql
SELECT
  name,
  type
FROM system.columns
WHERE
  table = 'http_requests_features'
ORDER BY name;
```

⚠️ Problem:

- Brak `WHERE database = 'default'`
- Po zmianie uprawnień → widoczne obie bazy
- **60 cech × 2 = 120+** cech

🦀 Rust i fatalne `unwrap()`

```
// Uproszczony kod który spowodował panikę
fn load_features(config: &Config) -> Features {
    let features = append_with_names(&config)
        .unwrap();  // 💥 BOOM!

    features
}
```

Problem z prealokacją pamięci:

- **Limit:** 200 cech (bufor bezpieczeństwa)
- **Oczekiwane:** ~60 cech
- **Otrzymane:** >200 cech (duplikaty)
- **Rezultat:** `Result::unwrap()` on `Err` → **PANIKA** 💀

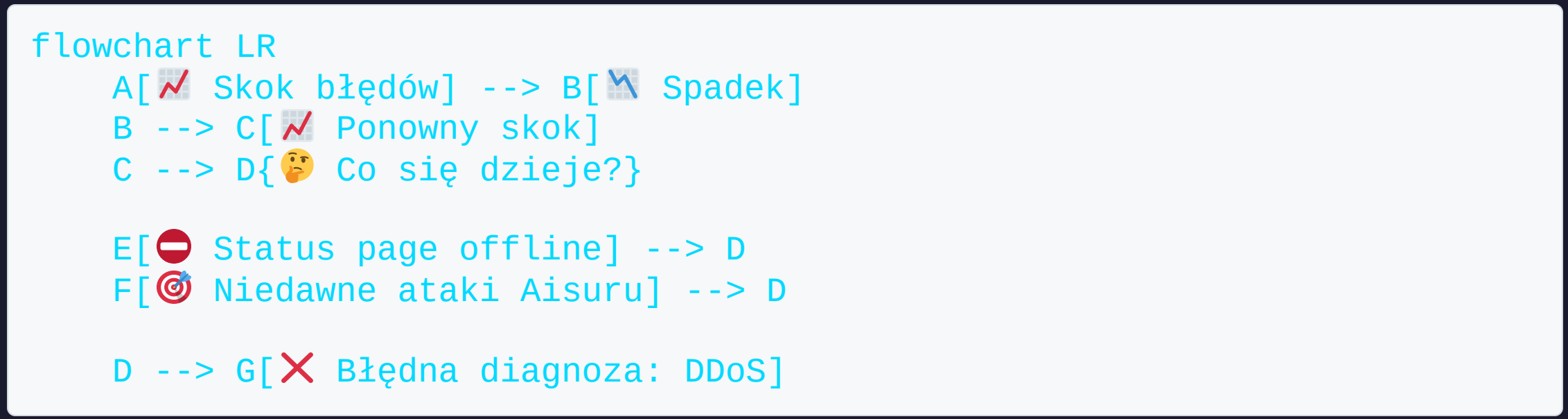
💥 Mechanizm awarii

```
sequenceDiagram
    participant CH as ClickHouse
    participant Gen as Generator pliku
    participant FL2 as Proxy FL2
    participant User as 👤 Użytkownik

    CH->>Gen: Zmienione uprawnienia
    Gen->>Gen: Generuj plik cech
    Note over Gen: >200 cech (duplikaty)
    Gen->>FL2: Propaguj plik
    FL2->>FL2: append_with_names()
    Note over FL2: 💥 unwrap() PANIC!
    FL2->>User: ❌ HTTP 500
```

🎭 Czynniki mylące

Dlaczego myśleli o ataku DDoS?



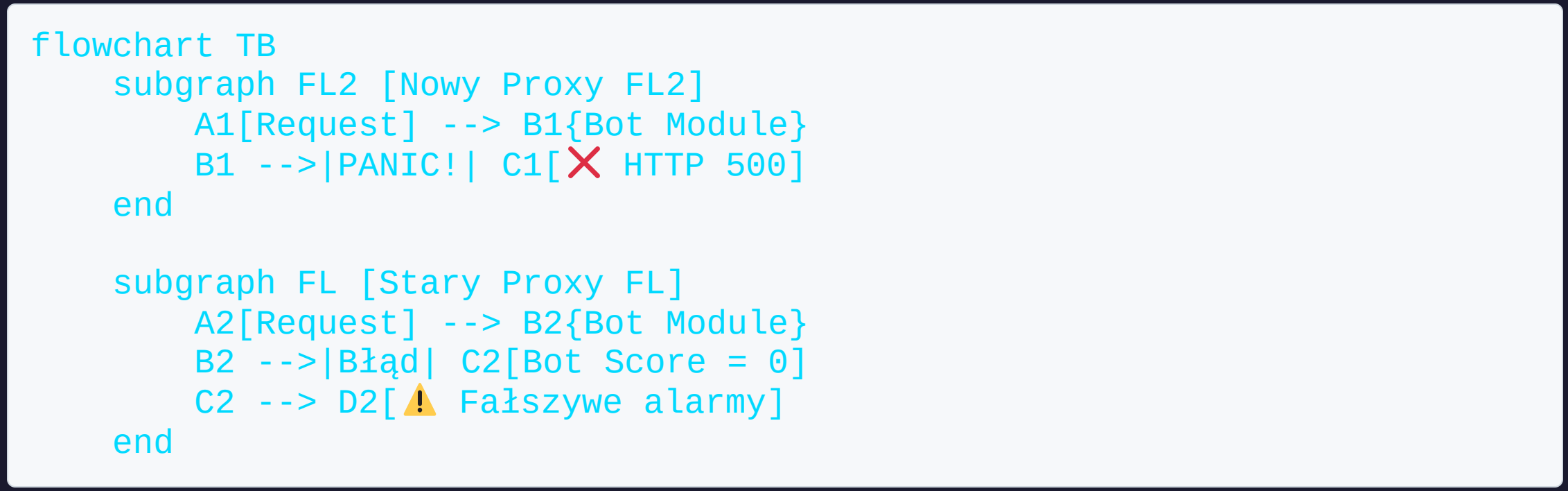
Nietypowe zachowanie:

- Fluktuacje: stare nody miały poprawny cache
- Status page (niezależna infra) też offline → **zbieg okoliczności!**

Wpływ na usługi

Usługa	Wpływ
CDN / Bezpieczeństwo	HTTP 5xx dla wszystkich klientów
Turnstile	Całkowity brak działania
Workers KV	Podwyższony poziom błędów
Dashboard	Brak możliwości logowania
Access	Błędy uwierzytelniania
Email Security	Obniżone wykrywanie spamu

🔧 FL vs FL2 - Różny wpływ



FL2: Twarde błędy 500

FL: Wszystko = "nie-bot" → problemy z regułami blokowania

📝 Działania naprawcze Cloudflare

Oficjalna lista:

1. 🔒 **Hardening** konfiguracji wewnętrznej (jak dane od użytkowników)
2. 🔵 **Kill-switches** - globalne wyłączniki funkcji
3. 💾 **Core dumps** - nie mogą przeciążać systemu
4. 🔍 **Przegląd trybów awarii** wszystkich modułów proxy

> *"Dzisiejsza awaria była najpoważniejszym incydentem od 2019 roku"*
> — Matthew Prince, CEO

💡 Nasze wnioski techniczne

Co można było zrobić lepiej?

```
flowchart TB
    A[Pobrano >200 cech] --> B{Sprawdź limit}
    B -->|Przekroczono| C[Weź pierwsze 200]
    C --> D[Log warning]
    D --> E[✅ Kontynuuj działanie]

    B -->|OK| E

    style C fill:#2d5016
    style D fill:#2d5016
```

Zamiast:

```
.unwrap()  // ❌ PANIC!
```

🏙️ Problem organizacyjny

```
flowchart LR
    subgraph Team_A [Zespół A - ClickHouse]
        A1[Modernizacja uprawnień]
    end

    subgraph Team_B [Zespół B - Bot Management]
        B1[Kod od lat działa]
        B2[Założenie: tylko baza 'default']
    end

    A1 -.->|Brak komunikacji| B1
    B1 --> C[💥 Awaria]

    style C fill:#8b0000
```

🎯 Kluczowy problem:

Zmiana w jednym miejscu → eksplozja w innym

🧪 A co ze środowiskiem testowym?

Możliwe wyjaśnienie:

```
flowchart TB
    subgraph PreProd [🧪 Pre-produkcja]
        A1[10 cech] --> B1[× 2 = 20]
        B1 --> C1[✅ < 200 - OK!]
    end

    subgraph Prod [🏭 Produkcja]
        A2[60 cech] --> B2[× 2+ = >200]
        B2 --> C2[❌ Przekroczony limit!]
    end
```

Skala produkcji ≠ Skala testów

🔥 Kluczowe lekcje

1️⃣ Defensywne programowanie

> Nigdy nie ufaj, że dane wejściowe będą poprawne

2️⃣ Graceful degradation

> System powinien działać ograniczenie, nie crashować

3️⃣ Komunikacja między zespołami

> Zmiany w jednym systemie mogą wpłynąć na inne

📈 Wizualizacja awarii

```
xychart-beta
    title "Błędy HTTP 5xx podczas incydentu"
    x-axis [11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 15:30, 16:00, 17:00]
    y-axis "Wolumen błędów" 0 --> 100
    line [5, 85, 70, 90, 60, 50, 45, 20, 15, 10, 8, 5]
```

Fluktuacje = różne nody z różnymi wersjami pliku cech

🤔 Do dyskusji

Pytania dla zespołu:

1. 🔍 **Czy mamy podobne "ukryte zależności"** w naszych systemach?
2. 🦀 **Jak obsługujemy błędy** w krytycznych ścieżkach kodu?
3. 📊 **Czy nasze środowiska testowe** odzwierciedlają skalę produkcji?
4. 🔔 **Jak szybko wykryjemy** awarię przed użytkownikami?
5. 📝 **Czy robimy post-mortemy** i czy są publiczne?

🎯 Podsumowanie

```
mindmap
  root((Awaria Cloudflare))
    Przyczyna
      Zmiana uprawnień ClickHouse
      Brak dyskryminatora bazy
      Duplikaty cech >200
    Błąd
      Prealokacja pamięci
      unwrap() w Rust
      Brak graceful degradation
    Skutek
      16% internetu offline
      ~6h do pełnego recovery
    Lekcje
      Defensywne programowanie
      Komunikacja zespołów
      Testy na skali prod
```

📚 Źródła

Oficjalne Post-Mortem:

🔗 blog.cloudflare.com/pl-pl/18-november-2025-outage

Video:

🎬 IT News #25 - DevMentors

🙏 Dziękuję!

Pytania?

Kontakt: [Twój email/Slack]