Architektura Cyberbezpieczeństwa

Podstawowe Zasady Bezpieczeństwa

Enterprise Security Architecture

6 Zasad Cyberbezpieczeństwa

1

Obrona Warstwowa

Defense in Depth

2

Najmniejsze Uprawnienia

Least Privilege

3

Rozdzielność Obowiązków

Separation of Duties

4

Bezpieczny przez Projekt

Secure by Design

5

Nie Komplikuj

KISS Principle

X

NIE: Bezp. przez Ukrycie

Security by Obscurity

1

Obrona Warstwowa

Defense in Depth

Nie polegaj na jednym mechanizmie

Każda warstwa ochrony działa niezależnie — awaria jednej nie kompromituje całego systemu.

Nowoczesny przykład warstwowy

MFA → MDM/EDR na urządzeniu → Firewall sieciowy → Testy podatności → Szyfrowanie bazy danych.

Cel: brak pojedynczego punktu awarii

System powinien zawodzić w bezpieczny sposób (fail safe), nawet gdy jeden element zostanie przełamany.

2

Najmniejsze Uprawnienia

Least Privilege

Dostęp tylko wtedy, gdy naprawdę potrzebny

Nadawaj uprawnienia wyłącznie osobom z uzasadnioną potrzebą biznesową i tylko na czas trwania tej potrzeby.

Utwardzanie systemów (hardening)

Usuń niepotrzebne usługi (FTP, SSH), zmień domyślne nazwy kont i hasła — każda zbędna usługa to powiększona powierzchnia ataku.

Zapobiegaj pełzaniu uprawnień (privilege creep)

Coroczne kampanie recertyfikacyjne przeglądają i usuwają nieaktualne prawa dostępu. Żadnych uprawnień 'na wszelki wypadek'.

3

Rozdzielność Obowiązków

Separation of Duties

Żaden pojedynczy użytkownik nie kontroluje krytycznej akcji

Celem jest wymuszenie zmowy co najmniej dwóch osób — skoordynowane działanie jest trudne do ukrycia.

Przykład fizyczny: drzwi z dwoma zamkami

Każda z dwóch osób ma klucz do innego zamka. Żadna nie może otworzyć drzwi samodzielnie.

Przykład IT: wnioskodawca ≠ zatwierdzający

Użytkownik składa wniosek o dostęp, inna osoba go zatwierdza. Samo-zatwierdzanie niszczy zasadę rozdzielności.

4

Bezpieczny przez Projekt

Secure by Design

Bezpieczeństwo od pierwszego dnia, nie na końcu

Tak jak budynek w strefie sejsmicznej projektuje się od razu jako odporny, bezpieczeństwo musi być wbudowane od wymagań.

Każdy etap SDLC zawiera aspekty bezpieczeństwa

Wymagania → Projekt → Kodowanie → Instalacja → Testy → Produkcja — bezpieczeństwo jest obecne wszędzie.

Cel: system bezpieczny 'z pudełka'

Odpowiedzialność ponoszą wszyscy — projektant, administrator i użytkownik — ale zaczyna się od projektanta.

5

Zasada KISS

Keep It Simple, Stupid — Nie Komplikuj

Złożoność to wróg bezpieczeństwa

Jeśli zabezpieczenia są zbyt skomplikowane dla uprawnionych użytkowników, będą je omijać — i to jest gorsze niż ich brak.

Przykład: skomplikowane polityki haseł

Zmuszeni do tworzenia niemożliwych do zapamiętania haseł, użytkownicy zapisują jedno i używają wszędzie — efekt odwrotny do zamierzonego.

Przeszkody dla atakującego, nie dla użytkownika

Obrona warstwowa i KISS nie są sprzeczne — labirynt powinien być zaprojektowany dla złoczyńcy, a nie dla pracownika.

Czego NIGDY nie robić

Bezpieczeństwo przez Ukrycie · Security by Obscurity

Tajemnica ≠ Bezpieczeństwo

Poleganie na tym, że nikt nie pozna mechanizmu działania systemu, nie jest prawdziwym zabezpieczeniem.

Zasada Kerckhoffsa

System kryptograficzny powinien być bezpieczny nawet jeśli wszystko o nim jest znane — jedyną tajemnicą jest klucz.

Szklana skrzynka zamiast czarnej skrzynki

AES i RSA są w pełni opublikowane, a mimo to są bezpieczne. Bezpieczeństwo oparte wyłącznie na tajności mechanizmu jest zawodne.

Podsumowanie

1. **Obrona Warstwowa —**
   Wiele warstw — brak pojedynczego punktu awarii
2. **Najmniejsze Uprawnienia —**
   Tylko to, czego potrzebujesz, tylko gdy potrzebujesz
3. **Rozdzielność Obowiązków —**
   Wymuś zmowę — jeden to za mało
4. **Bezpieczny przez Projekt —**
   Bezpieczeństwo od pierwszego dnia
5. **KISS —**
   Prostota chroni — złożoność szkodzi

Zasada, której nigdy nie stosuj: Bezpieczeństwo przez Ukrycie