LABORATORIUM CYBERBEZPIECZEŃSTWO

Data wykonania ćwiczenia:	02.12.2024
Rok studiów:	4
Semestr:	7
Grupa studencka:	2
Grupa laboratoryjna:	2B

Ćwiczenie nr.	5

Temat: Odkrywanie wycieku danych / nieautoryzowanego dostępu

Osoby wykonujące ćwiczenia:

1. Igor Gawłowicz

Katedra Informatyki i Automatyki

Sprawozdanie: Odkrywanie wycieku danych / nieautoryzowanego dostępu

1. Wprowadzenie

Wyciek danych to incydent związany z ujawnieniem informacji w wyniku utraty lub nieautoryzowanego użycia. Takie zdarzenie często wynika z naruszenia jednego z podstawowych atrybutów bezpieczeństwa informacji, takich jak poufność, integralność, dostępność, czy autentyczność. Przyczyny wycieku danych można podzielić na **wewnętrzne** (np. błędy użytkowników, działanie pracowników) i **zewnętrzne** (np. ataki cybernetyczne, klęski żywiołowe).

Technologia **honeytoken** jest jednym ze skutecznych narzędzi do odkrywania nieautoryzowanego dostępu. Honeytoken to celowo umieszczone „pułapki" (np. pliki, konta, adresy URL), których użycie przez nieuprawnionych użytkowników wywołuje natychmiastowe powiadomienie. Serwis **Canarytokens** pozwala łatwo generować i zarządzać różnymi typami tokenów, takimi jak:

- **HTTP Token**: Ukryte zasoby w formie linków.
- **DNS Token**: Nazwy domenowe, które rejestrują próby ich rozwiązania.
- **Cloned Website Token**: Strony internetowe stworzone jako pułapki.
- **Database Token**: Dane w bazach danych, których odczyt jest monitorowany.

Gdy intruz wykorzysta wygenerowany token, narzędzie Canarytokens natychmiast wysyła powiadomienie z informacjami o zdarzeniu, umożliwiając szybkie podjęcie działań.

2. Cel i realizacja zadania

Zadanie polegało na:

1. Wdrożeniu technologii honeytoken z wykorzystaniem serwisu Canarytokens.
2. Obsłudze różnych typów tokenów, takich jak **HTTP Token**, **DNS Token**, czy **Cloned Website Token**.
3. Zintegrowaniu tokenów z funkcjonalnością istniejącego systemu w celu wykrywania prób nieautoryzowanego dostępu.

3. Rozwiązanie

Zaimplementowano funkcję monitorującą próbę nieautoryzowanego dostępu w module panelu administracyjnego aplikacji. Kluczowe elementy rozwiązania to:

- **HTTP Token Notification**: Podczas odwiedzania panelu administracyjnego, aplikacja wykonuje żądanie do adresu URL powiązanego z Canarytoken. Każda próba uzyskania dostępu przez nieautoryzowaną osobę jest rejestrowana, a administrator zostaje o tym powiadomiony e-mailem.
- **Weryfikacja ról użytkowników**: Panel administracyjny jest dostępny tylko dla użytkowników o określonych rolach (np. admin, user_manager). Nieautoryzowane próby dostępu kończą się odpowiedzią HTTP 403.
- **Monitorowanie zmian ustawień**: Wszystkie zmiany w konfiguracji systemu (np. czas sesji) są logowane, co zwiększa rozliczalność działań użytkowników.

4. Mechanizm Canarytoken w kodzie

Kod ilustrujący implementację zawiera:

- Wywołanie HTTP Token Canary przy wchodzeniu na stronę panelu administracyjnego.
- Obsługę błędów związanych z żądaniami do Canarytoken.
- Kontrolę dostępu do panelu na podstawie ról użytkowników.

Poniżej fragment kodu odpowiadającego za wywołanie tokena i kontrolę dostępu:

```
@app.route('/admin/dashboard', methods=['GET', 'POST'])
@jwt_required()
def admin_dashboard():
    # Wywołanie Canary Token
    try:

requests.get("http://canarytokens.com/images/static/2ntxtx7ap5u7yj9uu8e86cc
sp/submit.aspx")
    except requests.RequestException as e:
        # Logowanie błędów wywołania tokena
        print(f"Failed to notify Canary token: {e}")

    current_user = get_jwt_identity()
    allowed_roles = ['admin', 'user_manager', 'session_manager',
'debugger']
    if current_user['role'] not in allowed_roles:
        return {"error": "Unauthorized access"}, 403

    # Reszta kodu obsługującego panel administracyjny...
```

5. Wnioski

Technologia honeytoken, wdrożona za pomocą serwisu Canarytokens, okazała się prostym, ale skutecznym sposobem monitorowania potencjalnych prób wycieku danych lub nieautoryzowanego dostępu. Dzięki automatycznym powiadomieniom, administratorzy mogą szybko reagować na incydenty bezpieczeństwa.

Zastosowanie tej metody w praktyce pozwala na:

- Wczesne wykrywanie naruszeń bezpieczeństwa.
- Zwiększenie kontroli nad dostępem do zasobów.
- Możliwość dostosowania technologii do własnych potrzeb, np. przez hostowanie własnych tokenów.

Możliwości rozwoju obejmują integrację z bardziej zaawansowanymi systemami SIEM (Security Information and Event Management) oraz wykorzystanie tokenów w rozproszonych systemach monitoringu.