نظام كشف تسريب كلمات المرور الذكي

نظام متكامل يُمكّن الأفراد والشركات من اكتشاف تسريب كلمات المرور المحتمل قبل أن يتسبب في اختراق فعلي، من خلال الجمع بين تحليل قواعد بيانات التسريبات العامة، الذكاء الاصطناعي لتحليل سلوك المستخدم، ومراقبة الإنترنت المظلم والويب المفتوح باستخدام تقنيات متقدمة.

المميزات الرئيسية

لوحة تحكم للمستخدم

- عرض الحالة الأمنية الحالية للمستخدم
- عرض آخر التنبيهات والإشعارات
- توصيات لتحسين الأمان
- رسومات بيانية حول النشاط الأمني

التكامل مع قواعد بيانات التسريبات

- التكامل مع موقع HaveIBeenPwned للتحقق من البريد وكلمات المرور
- تتبع منصات مثل Reddit, Pastebin عبر Web Scraping للكشف عن تسريبات جديدة
- فحص دوري للتسريبات الجديدة

تحليل سلوك المستخدم بالذكاء الاصطناعي

- استخدام الذكاء الاصطناعي لتحليل أنماط تسجيل الدخول
- اكتشاف نشاط غير معتاد مثل:
- تسجيل دخول من دولة جديدة
- عدد محاولات خاطئة متكررة
- تسجيل الدخول بأوقات غير معتادة
- تدريب نموذج Machine Learning (Decision Tree أو Random Forest) لتصنيف النشاط

تنبيهات ذكية وفورية

- إرسال إشعارات عبر:
- البريد الإلكتروني
- Telegram
- إشعار مباشر في لوحة التحكم

- تتضمن تفاصيل الحدث، الموقع الجغرافي، والتوصيات الفورية

مولّد ومدير كلمات مرور آمن

- توليد كلمات مرور عشوائية وقوية باستخدام معايير OWASP
- تخزينها بشكل مشفّر باستخدام AES-256
- تذكير دوري لتغيير كلمات المرور
- تقييم قوة كلمات المرور

التقنيات المستخدمة

Backend

- **Python Flask** - إطار عمل الخادم الخلفي
- **SQLAlchemy** - ORM لقاعدة البيانات
- **SQLite** - قاعدة البيانات

الذكاء الاصطناعي والتعلم الآلي

- **Scikit-learn** - مكتبة التعلم الآلي
- **Pandas & NumPy** - معالجة البيانات
- **Random Forest & Isolation Forest** - نماذج الكشف عن الشذوذ

الأمان والتشفير

- **Cryptography (AES-256)** - تشفير كلمات المرور
- **bcrypt** - تشفير كلمات مرور المستخدمين
- **HaveIBeenPwned API** - فحص التسريبات

التنبيهات والإشعارات

- **SMTP** - إرسال البريد الإلكتروني
- **python-telegram-bot** - تكامل مع Telegram
- **APScheduler** - جدولة المهام

أدوات إضافية

- **Requests & BeautifulSoup** - Web Scraping
- **Flask-CORS** - دعم CORS

- **python-dotenv** - إدارة متغيرات البيئة

التثبيت والتشغيل

المتطلبات

- Python 3.11+
- pip

خطوات التثبيت

1. **استنساخ المشروع**

```
git clone <repository-url>
cd password-leak-detection-system
```

1. **إنشاء بيئة افتراضية**

```
python -m venv venv
source venv/bin/activate  # على Linux/Mac
# أو
venv\Scripts\activate  # على Windows
```

1. **تثبيت المتطلبات**

```
pip install -r requirements.txt
```

1. **تشغيل النظام**

```
python src/main.py
```

1. **فتح المتصفح**

```
http://localhost:5000
```

واجهات برمجة التطبيقات (APIs)

إدارة المستخدمين

- `POST /api/users/register` - تسجيل مستخدم جديد
- `POST /api/users/login` - تسجيل الدخول
- `POST /api/users/logout` - تسجيل الخروج
- `GET /api/users/profile` - الملف الشخصي
- `PUT /api/users/profile` - تحديث الملف الشخصي
- `POST /api/users/change-password` - تغيير كلمة المرور

إدارة كلمات المرور

- `POST /api/passwords/generate` - توليد كلمة مرور قوية
- `POST /api/passwords/evaluate` - تقييم قوة كلمة المرور
- `POST /api/passwords/save` - حفظ كلمة مرور
- `GET /api/passwords/list` - قائمة كلمات المرور المحفوظة
- `GET /api/passwords/<id>` - الحصول على كلمة مرور محددة
- `PUT /api/passwords/<id>` - تحديث كلمة مرور
- `DELETE /api/passwords/<id>` - حذف كلمة مرور
- `GET /api/passwords/check-expiry` - فحص كلمات المرور المنتهية الصلاحية
- `POST /api/passwords/bulk-check` - فحص جماعي لكلمات المرور

الأمان والتنبيهات

- `POST /api/security/check-email` - فحص البريد الإلكتروني للتسريبات
- `GET /api/security/breach-history` - تاريخ فحص التسريبات
- `GET /api/security/alerts` - قائمة التنبيهات
- `POST /api/security/alerts/<id>/read` - تحديد التنبيه كمقروء
- `POST /api/security/alerts/<id>/resolve` - تحديد التنبيه كمحلول
- `GET /api/security/alerts/stats` - إحصائيات التنبيهات
- `GET /api/security/behavior/stats` - إحصائيات السلوك
- `POST /api/security/behavior/train` - تدريب نموذج تحليل السلوك
- `GET /api/security/dashboard` - لوحة تحكم الأمان الشاملة
- `POST /api/security/test-notification` - اختبار إرسال التنبيهات

معلومات النظام

- `GET /api/system/info` - معلومات النظام والمطور

هيكل المشروع

```
password-leak-detection-system/
├─── src/
│   ├─── models/
│   │   └─── user.py            # نماذج قاعدة البيانات
│   ├─── routes/
│   │   ├─── user.py            # مسارات المستخدمين
│   │   ├─── passwords.py       # مسارات كلمات المرور
│   │   └─── security.py        # مسارات الأمان والتنبيهات
│   ├─── services/
│   │   ├─── breach_checker.py  # خدمة فحص التسريبات
│   │   ├─── password_manager.py # خدمة إدارة كلمات المرور
│   │   ├─── notification_service.py # خدمة التنبيهات
│   │   └─── behavior_analyzer.py # خدمة تحليل السلوك
│   ├─── static/                # الملفات الثابتة
│   ├─── database/              # قاعدة البيانات
│   └─── main.py                # نقطة دخول التطبيق
├─── venv/                      # البيئة الافتراضية
├─── requirements.txt           # متطلبات المشروع
└─── README.md                  # هذا الملف
```

الأمان والخصوصية

- **تشفير كلمات المرور:** جميع كلمات المرور محفوظة مشفرة باستخدام AES-256
- **تشفير كلمات مرور المستخدمين:** باستخدام bcrypt مع salt
- **جلسات آمنة:** استخدام Flask sessions مع إعدادات أمان محسنة
- **CORS:** دعم CORS للتكامل مع الواجهات الأمامية
- **التحقق من صحة البيانات:** فحص شامل لجميع المدخلات

المساهمة

نرحب بالمساهمات! يرجى:

1. عمل Fork للمشروع
2. إنشاء branch جديد للميزة

3. إجراء التغييرات المطلوبة
4. إرسال Pull Request

الترخيص

هذا المشروع مرخص تحت رخصة MIT.

معلومات المطور

المطور: مطور نظام كشف تسريب كلمات المرور الذكي

الهاتف: 0592774301

البريد الإلكتروني: mabbadi0@icloud.com

الإصدار: 1.0.0
