Projekt "Bezpieczny Interfejs użytkownika bankowości internetowej" - koncepcja

Wojciech Szade

Projekt planuje wykonać z użyciem języka Python z wykorzystaniem framework'u Django.
Wybrałem ten framework ze względu na łatwość przygotowania całej strony webowej (można z jego wykorzystaniem pisać zarówno frontend jak i backend) oraz jego wysoki poziom bezpieczeństwa oraz przydatne wbudowane systemy.
Planuję skorzystać z wbudowanego systemu autoryzacji i kont 'auth', który zajmuje się podstawowym tworzeniem użytkowników, ich grupami i permisjami oraz ich autoryzacją i zabezpieczeniem poprzez token csrf.
Domyślne metody autoryzacji planuję rozszerzyć, tak żeby proces logowania składał się z:
1. Podania nazwy użytkownika i jego hasła - tu wykorzystuję wbudowaną metodę, która haszuje hasła przy użyciu algorytmu PBKDF2 z SHA256 i przechowywuje hasła w bazie w postaci: `'<algorithm>$<iterations>$<salt>$<hash>'`
2. Dodatkowa walidacja przy użyciu własnej metody - zw względu na brak doświadczenia, nie jestem jeszcze pewien z jakiej walidacji skorzystam, ale będzie to: walidacja OTP przy użyciu Google Authenticator, walidacja przy użyciu jednorazowego hasła wysłanego na maila (jak np. steam guard) lub walidacja przy użyciu wybranych cyfr kodu PIN (użytkownik definiuje w takim przypadku hasło + PIN).
Decyzję o wybraniu metody podejmę na etapie implementacji, po ocenieniu trudności i czasochłonności ich implementacji. Metody wymieniłem w kolejności od tych, które najbardziej chciałbym zastosować.
Zrezygnowałem z zadanej walidacji przy użyciu wybranych znaków hasła, ponieważ:
- wymaga ona skomplikowanej, przemyślanej implementacji, która obecnie nie jest też zbyt popularna - przez co brakuje dobrych źródeł,
- uważam, że hasła nie są stworzone do wybierania z nich poszczególnych znaków i takie ich wykorzystanie może prowadzić do stosowania przez użytkowników złych haseł, lub przechowywanie ich w niezabezpieczony sposób - stąd mój pomysł o dodatkowym kodzie PIN - cyfry dużo łatwiej człowiekowi iterować, a także poznanie wybranych cyfr kodu PIN rzadko kiedy pozwala na wydedukowanie całego kodu - w przypadku hasła jest to bardziej możliwe.
- jest to metoda od której się odchodzi (a właściwie to nigdy nie cieszyła się bardzo wysoką popularnością) - w przeciwieństwie do OTP, czy jednorazowych haseł,
- wymaga ona defacto całkowitego porzucenia domyślnych, bezpiecznych metod Django.
Dodatkowymi narzędziami z których planuje korzystać przy projekcie:
- django-defender - cytując autora:
'Super szybka reużywalna aplikacja django, która blokuje próby ataków brutalnej siły na systemy logowania.
- Gunicorn - serwer do pythonowych aplikacji kompatybilny z różnymi frameworkami (międzyinnymi django), - Ngninx - do rozmieszczenia gunicorna.