UWAGA! W tym rozdziale zignoruję kwestię *odzyskiwania hasła/konta*, ponieważ mają one niewiele wspólnego z kryptografią. Zapamiętajmy po prostu, że często są one związane z tym, w jaki sposób dokonywaliśmy rejestracji. Na przykład jeśli rejestrowaliśmy się za pośrednictwem działu IT w naszym miejscu pracy, prawdopodobnie będziemy musieli go odwiedzić, jeśli utracimy hasło. Mogą też być najsłabszym ogniwem naszego systemu, jeśli tylko nie będziemy ostrożni. I rzeczywiście, jeśli można odzyskać nasze konto dzięki zadzwonieniu na jakiś numer i podaniu komuś naszej daty urodzenia, wtedy cała ta wymyślna kryptografia stosowana w momencie logowania na nic się nie przyda.

Naiwny sposób implementacji tego poprzedniego przepływu uwierzytelniania użytkownika polega na zachowaniu jego hasła w momencie rejestracji, a następnie na poproszeniu o jego podanie w czasie logowania. Jak widzieliśmy w rozdziale 3, po pomyślnym uwierzytelnieniu użytkownik zwykle otrzymuje ciasteczko, które może zostać przesłane przy każdym kolejnym zapytaniu zamiast nazwy użytkownika i hasła. Ale momencik! Jeśli serwer przechowuje nasze hasło w jawnej postaci, wtedy każde włamanie do bazy danych kończy się ujawnieniem hasła napastnikom. Następnie będą oni mogli wykorzystać je do zalogowania się do dowolnej witryny, w której użyliśmy tego samego hasła do rejestracji.

Lepszym sposobem na przechowanie haseł byłoby wykorzystanie algorytmu *haszowania hasła*, takiego jak znormalizowany Argon2, które poznaliśmy w rozdziale 2. Skutecznie zapobiegłoby to „kradzieży z włamaniem" do bazy danych mającej na celu doprowadzenie do wycieku haseł, choć napastnik, który nadmiernie przedłużyłby swoją bytność w systemie, nadal byłby w stanie podpatrzyć nasze hasło za każdym razem, gdy się logujemy. Mimo to w dalszym ciągu wiele witryn i przedsiębiorstw przechowuje hasła w jawnej postaci.

Ćwiczenie

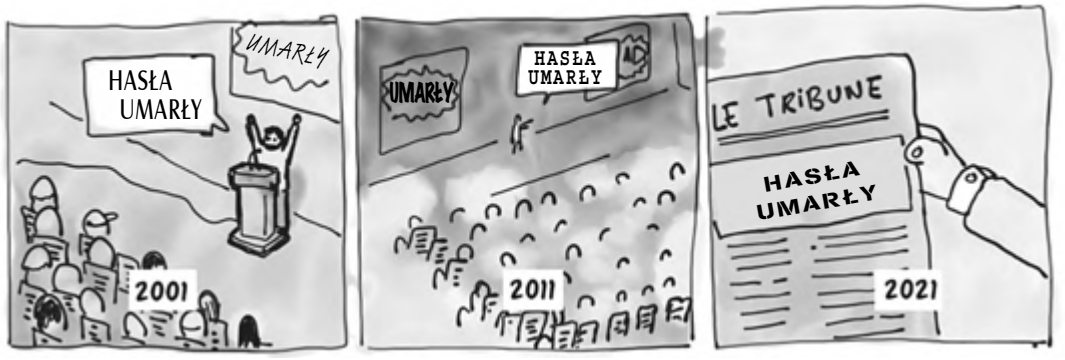
Niekiedy aplikacje usiłują naprawić problem związany z tym, że serwer ma możliwość poznania haseł użytkowników w momencie rejestracji, haszując hasło po stronie klienta (możliwe, że za pomocą algorytmu haszowania hasła), nim zostanie ono przesłane na serwer. Czy takie rozwiązanie rzeczywiście działa?

Co więcej, ludzie z natury rzeczy źle radzą sobie z hasłami. Najlepiej czujemy się z takimi, które są krótkie i łatwe do zapamiętania. A jeśli to możliwe, wolelibyśmy po prostu wszędzie używać tego samego hasła.

81% wszystkich włamań hakerskich wykorzystuje wykradzione lub słabe hasła.

Verizon Data Breach Report (2017)

Problem słabych haseł oraz problem ponownego wykorzystywania haseł doprowadziły do powstania wielu mało poważnych i denerwujących wzorców projektowych, które usiłują wymusić na użytkownikach bardziej poważne podejście do haseł. Na przykład niektóre witryny wymagają użycia w hasłach znaków specjalnych, wymuszają zmianę hasła co 6 miesięcy i tak dalej. Co więcej, wiele protokołów usiłuje „naprawić" hasła lub pozbyć się ich całkowicie. Wydaje się, że każdego roku nowi eksperci zaczynają żywić przekonanie, że oto „hasła" umarły, a mimo to pozostają one najszerzej stosowanym mechanizmem uwierzytelniania użytkowników.



Jak widać, hasła prawdopodobnie z nami zostaną. Mimo to istnieje wiele protokołów, które starają się ulepszyć lub zastąpić hasła. Pora im się przyjrzeć.

11.2.1. Jedno hasło, by rządzić wszystkimi. Pojedyncze logowanie (SSO) i menedżery haseł

W porządku, wielokrotne używanie tych samych haseł jest złe. Co możemy z tym zrobić? Przy naiwnym podejściu moglibyśmy oczekiwać, że użytkownicy będą tworzyć różne hasła dla różnych witryn. To podejście rodzi jednak dwa problemy:

- użytkownicy kiepsko sobie radzą z tworzeniem wielu różnych haseł;
- wysiłek umysłowy konieczny do zapamiętania wielu haseł sprawia, że jest to niepraktyczne.

Aby złagodzić te niedogodności, szeroko stosowane są dwa rozwiązania.

- *Pojedyncze logowanie* (ang. *single sign-on*, w skrócie SSO) – pomysł polega tu na tym, by umożliwić użytkownikom łączenie się z wieloma różnymi serwisami przez wykazanie, że mają konto w jednym serwisie. W ten sposób użytkownik musi zapamiętać tylko hasło powiązane z tym jednym serwisem, aby móc połączyć się z wieloma. Pomyślmy o przyciskach typu „Zaloguj przez Facebook", takich jak te pokazane na rysunku 11.3.