AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Identification and Authentication Failures

Anna Ledwoń
Jakub Dyrcz

Wydział Informatyki, Elektroniki i Telekomunikacji
Instytut Informatyki

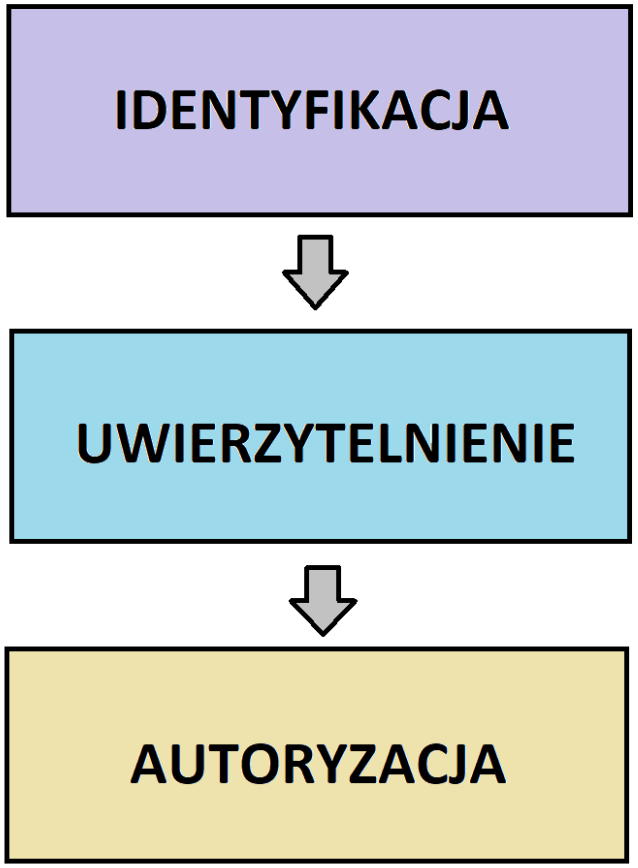
2021-12-10, Kraków

Identyfikacja, Uwierzytelnie i Autoryzacja

Identyfikacja (*identification*) - podmiot deklaruje swoją tożsamość

Uwierzytelnienie (*authentication*) - weryfikacja tożsamości danego podmiotu

Autoryzacja (*authorization*) - określenie zasobów, do których podmiot może mieć dostęp oraz uprawnień, jakie mu przysługują



Czynniki uwierzytelnienia

Można wyróżnić 3 podstawowe czynniki, które mogą posłużyć do uwierzytelnienia:

1. **Knowledge factors** - coś, co wiemy, jak np. hasło czy odpowiedź na specjalne pytanie

2. **Possession factors** - coś, co posiadamy, jak security token

3. **Inherence factors** - coś związanego z naszym zachowaniem bądź tym kim jesteśmy, jak dane biometryczne tudzież pewne szczególne wzorce zachowań

Password-based login

W tym przypadku, aby pomyślnie przejść proces uwierzytelnienia wystarczy znajomość sekretnego hasła

=>

Cel ataku: odgadnięcie bądź zdobycie danych logowania

Rodzaje ataków:

1. **Brute-forcing usernames & passwords**
2. **Username enumeration**

Sposoby ochrony – Brute-force protection:

1. **Account locking**
2. **User rate limiting**

Username enumeration

Z podatnością tą mamy do czynienia wówczas, gdy atakujący ma możliwość stwierdzenia, czy dany username jest poprawny czy też nie

=> skrócenie czasu poświęconego na zbrute-forcowanie danych logowania poprzez uzyskanie listy poprawnych użytkowników

Gdzie szukać wskazówek?

1. **Kod statusu** *(Status code)*
2. **Wiadomość o błędzie** *(Error message)*

Valid User, but invalid password

```
<p>
    The username or password is not valid.
</p>
```

Invalid User

```
<p>

    The username or password is not valid.
</p>
```

3. **Czas odpowiedzi** *(Response times)*

Lab 1 – Uwierzytelnianie hasłem

https://github.com/LittleBigKiller/bawim-auth

Account locking

Zablokowanie danego konta, jeśli spełnione zostaną wyspecyfikowane warunki, najczęściej - przekroczona zostanie liczba nieudanych prób uwierzytelnienia.

- ✓ Zapewnie pewien poziom ochrony przed próbą ataku brute-force na **konkretne** konto
- Brak ochrony w przypadku próby brute-force'owania losowego konta:

1. Ustal listę nazw użytkowników, które powinny być poprawne (np. dzięki username enumeration)
2. Ustal listę często używanych haseł (maksymalna ich liczba, która nie spowoduje zablokowania konta)
3. Dla każdego username wypróbuj każde z haseł

- Brak ochrony przed credential stuffing attacks

User rate limiting

W tym przypadku zbyt duża liczba wysłanych w krótkim przedziale czasu żądań HTTP skutkuje zablokowaniem adresu IP, z którego je wysyłano.

Najczęściej odblokowanie adresu następuje:

- automatycznie po pewnym czasie
- ręcznie, przez administratora
- ręcznie, przez użytkownika po pomyślnym wypełnieniu CAPTCHA

Jeden ze sposobów obejścia tegoż zabezpieczenia:

= opracowanie metody na zgadnięcie wielu haseł poprzez pojedyncze żądanie

Lab 2 – Rate limit i blokada konta

https://github.com/LittleBigKiller/bawim-auth

Multi-factor authentication

By w pełni skorzystać z możliwości wielopoziomowego uwierzytelnienia należy weryfikować **różne** czynniki.

Np. W przypadku Email-based 2FA dwukrotnie sprawdzamy ten sam czynnik - 'something you know'.

Najczęściej korzysta się z dwu-poziomowego uwierzytelnienia na podstawie: sth you **know** & **have**

- ✓ **Tokeny uwierzytelniania** *(Authentication tokens)* - bezpośrednie
- **Wiadomość SMS** *(Text message)* - pośrednie, możliwość przejęcia kodu podczas transmisji czy podmiany karty SIM

Ataki w przypadku dwupoziomowego uwierzytelniania

1. Możliwość **całkowitego** obejścia dwupoziomowego uwierzytelniania wynika z błędnej implementacji tegoż mechanizmu -> można przejść na strony "logged-in only" po pomyślnym zakończeniu pierwszego etapu bez konieczności wykonania drugiego.

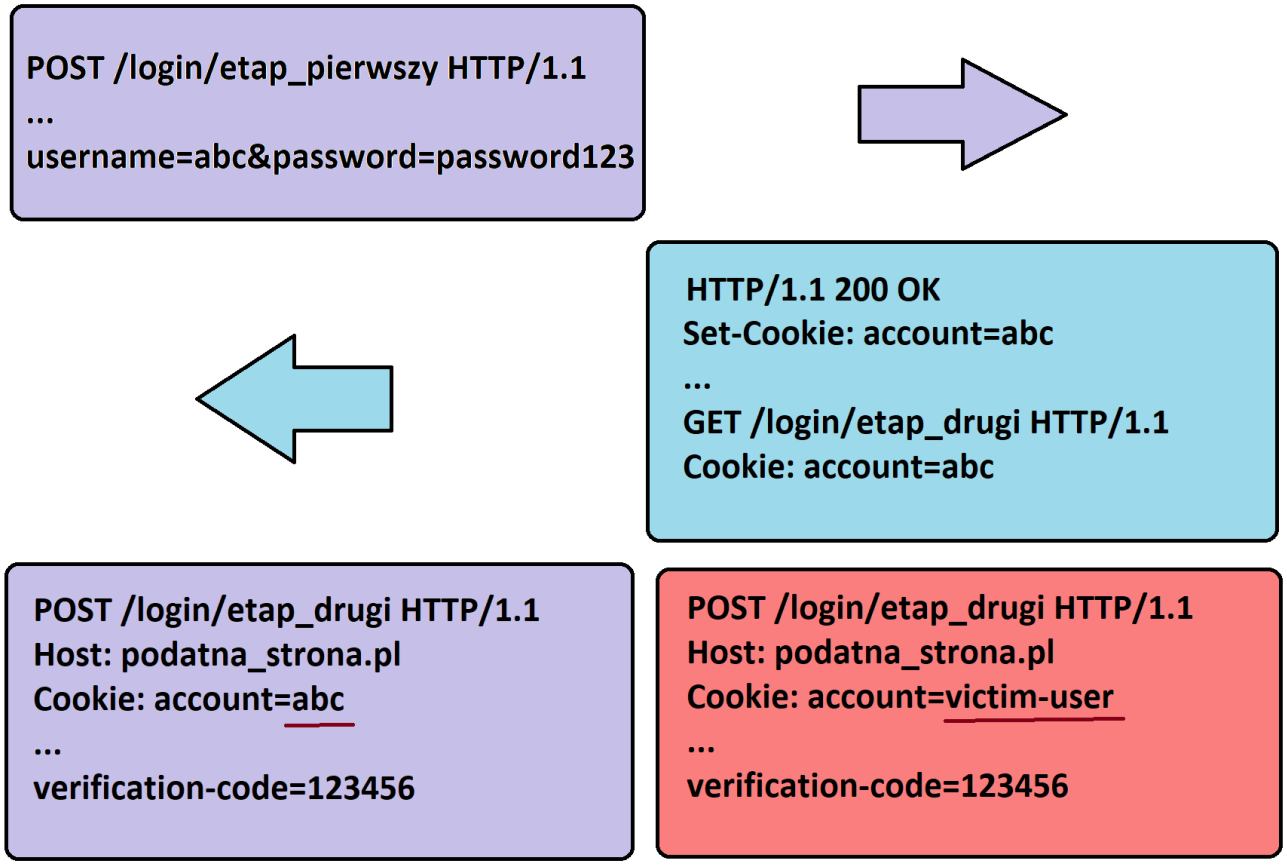
2. **Flawed two-factor verification logic**

Po pomyślnym przejściu pierwszego etapu logowania, strona nie sprawdza odpowiednio, czy to **ten sam** użytkownik wykonuje drugi etap.

Lab 3.1 – Proste ominięcie 2FA

https://github.com/LittleBigKiller/bawim-auth

Flawed two-factor verification logic



Lab 3.2 – Błędna implementacja 2FA

https://github.com/LittleBigKiller/bawim-auth

Keeping users logged in

Guess cookie

Naïve encryption

No salt

Password	MD5	SHA1
123456	e10adc3949ba59abbe56e057f20f883e	7c4a8d09ca3762af61e59520943dc26494f8941b
password÷NO PASSWORDS!	320157b0a9d971845c5b0a0796058c79	d8be2ef007bd17d46b3cd126861cd58655a40c33
12345678	25d55ad283aa400af464c76d713c07ad	7c222fb2927d828af22f592134e8932480637c0d
qwerty	d8578edf8458ce06fbc5bb76a58c5ca4	b1b3773a05c0ed0176787a4f1574ff0075f7521e

Źródło: https://hashtoolkit.com/common-passwords/

Resetting user passwords

1. Wysłanie nowego hasła poprzez **email**
   - Wygaśnięcie hasła po bardzo krótkim czasie
   - Natychmiastowa zmiana hasła

Atak: man-in-the-middle

2. Reset hasła poprzez **URL**

http://podatna-strona.pl/reset-hasla?user=ofiara-user

http://podatna-strona.pl/reset-hasla?token=a0ba0dc8eff1cd2257e69a8

Należy ponownie **zweryfikować token**, gdy formularz jest przesyłany!

Lab 4 – Password resets

https://github.com/LittleBigKiller/bawim-auth

Jeszcze inne typowe ataki

1. **Credential stuffing** - dostęp do skompromitowanych danych

2. **Domyślne ustawienia** - “admin/admin”

3. ...

Podstawowe zasady zapobiegania atakom

1. Należy przesyłać dane logowania jedynie przez połączenia **szyfrowane**

2. Należy pamiętać, że użytkownicy rzadko dbają o zasady bezpieczeństwa, jeśli strona tego od nich nie wymaga

3. Należy pamiętać o implementacji **brute-force protection**

4. Należy wziąć pod uwagę **wszelkie** funkcjonalności dotyczące uwierzytelnienia, nie tylko proces logowania, np. również bezpieczeństwo podczas resetu hasła

5. Dobrze zaimplementowane uwierzytelnianie **wielopoziome** jest znacznie bezpieczniejsze od samego password-based login

Dziękujemy za uwagę.

Jakub Dyrcz
Anna Ledwoń