

Pengujian Penetrasi Json Web Token Terhadap Spring Boot Suite Menggunakan Algoritma SHA256

Yiyi Supendi*¹

Program Studi Teknik Informatika, Universitas Langlangbuana, Bandung

Email: *¹yiyi.supendi@gmail.com

Abstrak

Pada sebuah sistem teknologi informasi terdapat banyak data yang tersaji mulai dari data yang bersifat umum sampai data yang bersifat sangat privat, jika terdapat suatu data informasi yang privat maka sudah pasti data itu haruslah terlindungi dari orang yang tidak mempunyai otoritas untuk membaca dan memanipulasi data informasi tersebut. Data privat suatu individu sangat krusial bagi individu tersebut, dan data privat suatu individu tidak boleh diketahui oleh individu lain, yang artinya suatu data hanya dapat dilihat oleh individu yang telah terotentifikasi dan mempunyai hak akses atau mempunyai otoritas atas data tersebut. JSON Web Token yang disingkat JWT adalah sistem otentifikasi dan otorisasi *simple* yang populer belakangan ini, JWT adalah alat yang cocok untuk melakukan perlindungan proses otentifikasi dan otorisasi. Implementasi yang *simple* dan bentuknya yang kecil membuat JWT ini populer dikalangan aplikasi yang mempunyai arsitektur *microservice*. *Microservice* sendiri adalah sebuah arsitektur aplikasi dimana rancangannya yang bersifat Modular yang dimana sangat cocok bila disandingkan dengan sistem keamanan otentikasi dan otorisasi JSON Web Token. Namun dibalik kepopuleran JSON WebToken terdapat suatu kelemahan yang bila tidak diperhatikan akan berdampak pada data.

Kata Kunci: JSON Web Token, *icroservice*, *Penetration Test*

arsitektur program yang semakin populer pada setengah dekade atau lebih (Newman, 2021) *microservice* adalah sebuah *service* yang memiliki otoritas independent yang bekerja sama dengan *microservice* lainnya untuk mendukung sebuah fungsi dari aplikasi. Keunggulan utama dari penggunaan *microservice* adalah, pengembangan yang lebih cepat, pemeliharaan lebih mudah, skalabilitas dan ketahanan yang dapat ditingkatkan (Algimantas Venčkauskas, 2023, p. 1).

Saat mengintegrasikan aplikasi pada arsitektur *microservice*, terdapat banyak permasalahan yang menyoroti permasalahan pada kerahasiaan dan integritas data yaitu komunikasi sistem, entitas, atau pada proses. Jika permasalahan-permasalahan tersebut gagal terdeteksi dapat mengarah pada infrastruktur, seperti *spoofing*, *illegal access* dan *replay attack*. Pada permasalahan integritas terdapat masalah seperti *data interception*, manipulasi dan kebocoran data. (Lenin Leines- Vite, 2021). Namun selain pada *Confidential integrity*, terdapat konsep tambahan yaitu *Authentication* yaitu proses verifikasi pada orang yang mengakses.

Otentikasi dan otorisasi adalah sebuah sistem keamanan yang terletak paling terdepan dalam hal pengamanan. JSON Web Token adalah salah satu sistem otentikasi dan otorisasi yang cukup banyak digunakan, keunggulan otentikasi dan otorisasi berbasis token adalah proses otentikasi yang terpisah, dimana data otentikasi dan otorisasi disimpan pada *service* yang terpisah dan dapat secara mudah dibagikan antara *microservice* lainnya. (Algimantas Venčkauskas, 2023, p. 3)

Namun sayangnya JWT mempunyai kelemahan pada keamanan, diantaranya adalah pembajakan, pemalsuan, *replay* atau pemutar ulangan *request*, masa berlakutoken dan pada penyimpanan (Algimantas

1. PENDAHULUAN

Microservice telah menjadi sebuah

Venčkauskas, 2023, p. 3) maka dari itu untuk memitigasi hal itu haruslah dilakukan pengujian kelemahan yang dapat berpotensi penyerangan kepada JWT ini.

Dilihat dari beberapa paragraf di atas maka dalam penelitian ini akan dilakukan analisis terhadap keamanan JSON Web Token (JWT) dalam sebuah program berarsitektur microservice dan dirancang menggunakan *framework* Spring Boot Suite berbasis Java. JSON Web token akan dilakukan uji coba kerentanan dengan melakukan Breach/Attack pada JSON Web Token dan dilakukan pendokumentasian hasil uji coba kerentanan.

2. METODOLOGI PENELITIAN

2.1 Metode Penelitian

Untuk melakukan uji coba ketahanan pada aplikasi yang terimplementasi JSON Web Token, maka metode yang akan dipakai adalah metode penelitian eksperimental, karena pendekatan yang dilakukan adalah memahami hubungan sebab-akibat antara dua atau lebih variabel. Metode eksperimental memiliki keunggulan karena memberikan kontrol yang lebih tinggi atas variabel-variabel yang diteliti.

2.2 Metode Penterasi

Metode untuk melakukan asesemen pengujian keamanan yang dilakukan adalah *Penetration Testing Execution Standard* (PTES). Agar pengujian penetrasi terukur dan terorganisir karena tahapan-tahapan pengujian penetrasi mengikuti standar yang telah terukur dan teruji. Berikut tahapan-tahapan metode PTES yang akan dilakukan pada penelitian ini

1. *Pre-engagement Interactions*: pada tahapan pertama yaitu pendeskripsian persiapan alat yang akan digunakan yaitu kali linux, Burp Suite dan JWT_toolkit V2
2. *Intelligent Gathering*: pada tahap ini dilakukan pengumpulan informasi tentang Aplikasi yang akan diuji penetrasi
3. *Threat Modelling*: tahap ini akan mengidentifikasi tujuan dan kelemahan dari JSON Web Token lalu mendefinisikan cara untuk menghentikan atau memitigasi efek dari ancaman.
4. *Vulnerability Analysis*: pada tahap ini dilakukan analisis pada JSON Web Token untuk menemukan dan memvalidasi kelemahan.

5. *Exploitation*: pada bagian ini, penguji mencoba untuk mencapai melampaui sistem keamanan target sistem menggunakan kelemahan yang sebelumnya teridentifikasi dan tervalidasi.
6. *Post Exploitation*: Tahap ini mempertahankan control pada sistem target dan mengumpulkan data.
7. *Reporting*: mendokumentasikan seluruh proses kedalam bentuk yang dapat mudah dipahami. Laporan tersebut membahas mengenai keamanan pada sistem target.

2.3 Tahapan Penelitian

2.3.1 Identifikasi Masalah

Tahapan ini merupakan tahapan dimana peneliti menentukan permasalahan yang akan diteliti Adapun masalah yang akan diteliti meliputi kelemahan dalam mekanisme otorisasi, masa kadaluarsa token setelah logout, keamanan dalam penyimpanan token, dan kekurangan dalam implementasi. Pada penelitian ini akan menganalisis dan mengevaluasi kelemahan yang ada serta memberikan rekomendasi untuk meningkatkan keamanan otorisasi dalam sistem informasi akademik terpadu menggunakan JSON Web Token (JWT).

2.3.2 Tinjauan Pustaka

Pada tahapan ini peneliti akan melakukan studi Pustaka untuk memahami dasar teori yang terkait dengan keamanan *previlage*, JSON Web Token. Dan pada tahapan ini penulis juga akan melakukan studi pada sistem yang terimplementasi JSON Web Token.

2.3.3 Metode PTES

1. *Pre-Engagement Interaction*
Setelah melakukan studi Pustaka pada tahapan sebelumnya maka Langkah selanjutnya adalah melakukan pendeskripsian alat atau tools yang akan dipakai untuk menguji keamanan JSON Web Token
2. *Intelligent gathering*
Pada ini akan dilakukan pengumpulan informasi terhadap aplikasi yang akan dilakukan pengujian keamanan, mulai dari platform, Bahasa pemrograman yang digunakan, data flow, proses bisnis, framework, endpoint dan parameter parameter yang terdapat

- pada target sistem.
3. **Threat Modeling**
Pada tahap ini dilakukan perancangan pendekatan untuk proses penyerangan sistem target, perancangan ini dibuat sebagai acuan saat melakukan eksekusi penetrasi. Bentuk dari rancangan ini berupa proses bisnis dari sistem dan proses/langkah-langkah dari uji coba penyerangan.
 4. **Vulnerability Analysis**
Pada tahap ini dilakukan pencarian atau pengidentifikasian Kelemahan dan pendalaman terhadap kelemahan JSON Web Token, lalu memahami resiko yang dapat terjadi pada kelemahan yang telah teridentifikasi.
 5. **Exploitation**
Tahap ini dilakukan serangkaian eksekusi penyerangan terhadap kelemahan yang telah teridentifikasi sebelumnya, dimana semua kelemahan yang telah ditemukan diuji pada sistem target dan sistem keamanan JSON Web Token
 6. **Post Exploitation**
Tahap post exploitation dilakukan pemeliharaan sistem dari serangan yang telah berhasil dilakukan agar target sistem dapat bertahan dari serangan serangan yang berhasil dilakukan. Dan pada tahap ini pengujian penetrasi melakukan penilaian tingkat bahaya dari suatu serangan.
 7. **Reporting**
Reporting adalah tahap pendokumentasian semua Langkah-langkah yang telah dilakukan. Pada dokumentasi ini akan berisi tentang apa saja yang telah dilakukan, tahap-tahap proses pengujian, kelemahan-kelemahan yang teridentifikasi, dan bagaimana pengujian penetrasi menemukan kelemahan tersebut.

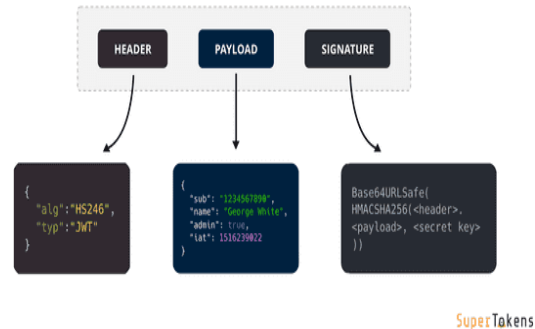
3. HASIL DAN PEMBAHASAN

3.1 Pre-Engagement Interaction

1. Proses pembuatan Token
JSON Web Token terdiri dari tiga bagian yaitu *header*, *payload* dan *signature*. *Header* memuat informasi tentang jenis data yang dikirim dan algoritma yang digunakan untuk

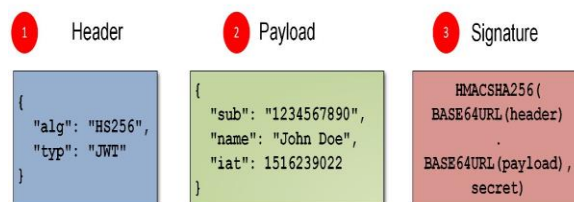
mengenkripsi *signature*, sedangkan *payload* berisi klaim atau *datarequest*, sedangkan *signature* adalah bagian yang dipakai untuk memverifikasi integritas data (Irwan Darmawan, 2023).

Structure of a JSON Web Token (JWT)



Gambar 1 Bagian JSON Web Token

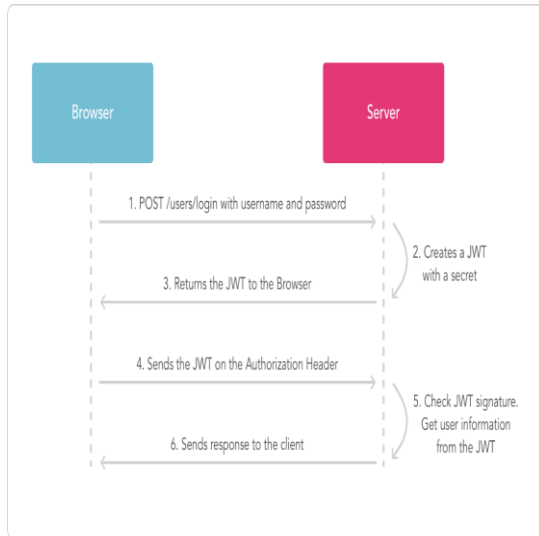
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJkZW50ODkwlwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5M00.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJkZW50ODkwlwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5M00.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJkZW50ODkwlwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5M00.



Gambar 2 Bentuk JSon Web Token

Dari gambar 2 dapat dilihat bentuk dari JSON Web Token yang telah terbentuk. Pada indikator no 1 dan nomor dua adalah header dan payload yang telah di enkripsi dengan *base64*, dan dipisahkan oleh sebuah titik(.). Setelah *header* dan *payload* terenkrpsi dimulai proses pembentukan bagian *signature* yaitu bagian *header* dan *payload* yang telah terenkrpsi dienkripsi lagi dengan algoritma yang tertera pada header, pada saat pengenkripsian ditambahkan sebuah *secret key*, lalu hasil nya disimpan dapa bagian akhir seperti yang tertera pada gambar 2 (Sajdak, 2023).

2. Proses Bisnis JSON Web Token



Gambar 3 Proses Bisnis JSON Web Token
Proses dari otentikasi JSON Web Token terdapat 6 langkah yang pertama yaitu pengguna melakukan *request login* dengan metode POST dengan mengirim objek JSON yang berisi data *username* dan *password*, lalu server membuat token, setelah itu mengirim token Kembali ke *browser* pengguna untuk digunakan *request* selanjutnya (Pooja Mahindrakar, 2020), setelah token didapatkan oleh pengguna maka saat pengguna membuat sebuah *request* lagi, browser mengirim token pada *server* pada *header* bernama *Authorization*. Dan setelah server memvalidasi token maka selanjutnya token melanjutkan proses *request* yang diminta.

a. JSON Web Token Toolkit

JSON Web Toolkit adalah sebuah *tools* atau alat untuk melakukan uji coba khusus untuk JSON Web Token, *tool* ini dibuat menggunakan Bahasa python versi 3, pada *tool* ini terdapat metode-metode penyerangan terhadap JSON Web Token (ticarpi, 2023). Berikut beberapa metode penyerangan yang tersedia :

- *The alg=none signature-bypass vulnerability*
- *The RS/HS256 public key mismatch vulnerability*
- *Key injection vulnerability*
- *Blank password vulnerability*
- *Null signature vulnerability*

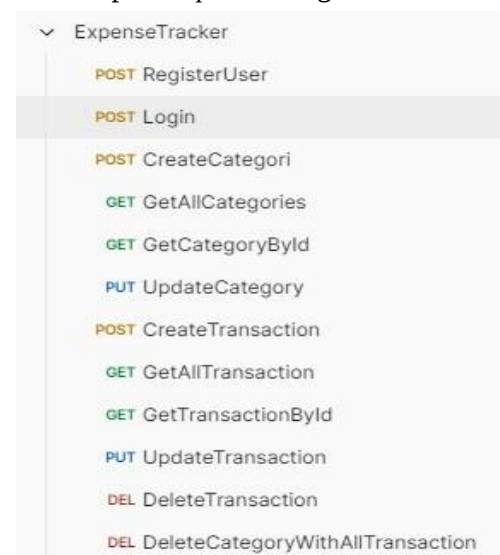
b. Jhon The Reaper (JTR)

Jhon the reaper adalah sebuah *open-source software* yang didisain untuk melakukan uji coba kekuatan *password* (Openwall, 2023), pada umumnya JTR digunakan oleh *pentester* atau *ethical hacker* untuk serangan pada *password* menggunakan metode *bruteforce* yaitu metode serangan *password* dengan cara menebak *password* satu persatu dimana *password* yang akan dicoba telah dipersiapkan terlebih dahulu.

c. Intelligence Gathering

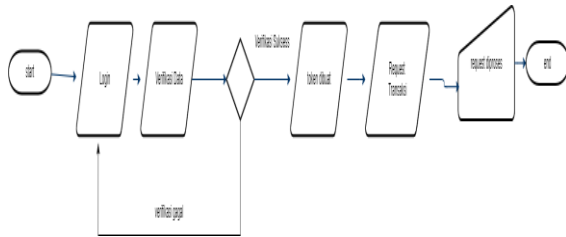
Untuk aplikasi yang akan menjadi target penetrasi adalah sebuah program API berarsitektur *microservice* yang dibangun menggunakan Bahasa pemrograman *Javada* framework *spring boot* tujuan aplikasi ini adalah mencatat setiap transaksi yang dilakukan. JSON Web Token pada aplikasi ini diimplementasi dengan library *jwt.io*, dan pada aplikasi ini terdapat 3 class controller yaitu *UserController*, *TransactionController*, *CategoryController*.

pada aplikasi *ExpenseTracker* terdapat *endpoint* sebagai berikut :



Gambar 4 ExpendTracker app Endpoint
Seperti yang tertera pada gambar 4 pada aplikasi *expenseTracker* mempunyai fungsi CRUD pada entitas *category* dan

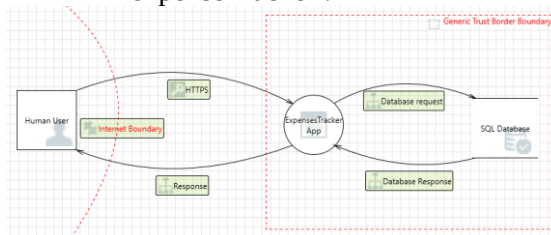
transaction, selain itu terdapat fungsi *registerUser* untuk mendaftarkan pengguna, *login* untuk otentikasi dan otorisasi.



Gambar 5 Flowchart ExpenseTracker app

d. Threat Modelling

Pada tahap threat modeling dilakukan analisis pada aplikasi expenseTracker, dimana dibuat gambaran dari arsitektur dari target sistem lalu melakukan analisis celah yang mungkin terjadi penyerangan pada sistem. pada kasus ini analisis dilakukan scanning menggunakan Microsoft threat modeling tools, berikut hasil dari tahap threat modeling dari aplikasi expenseTracker :



Gambar 6 Threat modeling ExpenseTracker

Tabel 1 Threat Modeling Report

No	Kategori	Threat	Deskripsi	Prioritas
1	Tampering	SQL Injection for SQL Database	Metode penyerangan dimana <i>malicious code</i> dimasukan sebagai tipe <i>string</i> yang selanjutnya digunakan untuk <i>sql Execution</i>	High
2	Spoofing	Spoof destinasi penyimpanan pada SQL Database	Penyerang dapat menipu <i>Sql database</i> yang mengarah pada pengiriman data pada penyerang	High
3	Information Disclosure	Authorization Bypass	Terdapat potensi untuk mengakses <i>database</i> tanpa melakukan proses otorisasi dan otentikasi	High
4	Tampering	Authenticated Data Flow Compromised	Penyerang dapat memodifikasi data yang ditransmisi pada proses data yang Terverifikasi	High

d. Vulnerability-analysis

Metode yang digunakan untuk vulnerability assessment adalah dengan cara manual, yaitu dengan mengubah variabel-variabel pada JSON Web Token seperti header, payload, dan pada signature. berikut beberapa kelemahan yang berpotensi terjadi sebuah exploit :

Tabel 2 Vulnerability Assesment

No	Kategori	Vulnerability	Deskripsi	Tingkat bahaya
1	Replay token	Token Expire check	Menguji apakah token yang dipakai setelah masa berlaku habis masih dapat Digunakan	Low
2	Token Check	Token signature check	Menguji validasi integritas signature	Low
3	alg=none signature-bypass	CVE-2015-2951	Informasi algoritma yang dipakai untuk hashing dapat mudah direkayasa	High
4	Null signature	CVE-2020-28042	Uji coba terhadap implementasi sistem validasi <i>signature</i> pada JWT	Medium
5	Bruteforce	secretkey bruteforcing	Penebakan <i>secretkey</i> dengan menggunakan Teknik <i>bruteforce</i> untuk megenerasi <i>signature</i>	High
6	X-HTTP-METHOD-Override	CVE-2023-30845	Mengubah <i>http method</i> untuk eskalasi otentikasi	High
	Blank password	CVE-2019-20933	Database me	High

e. Reporting

Berikut laporan dari hasil ujicoba pentes pada JSON Web Token pada aplikasi expenseTracker.

Tabel 3 Report Penetration

No	Kategori	Vulnerability	Hasil Uji Coba	Tingkat bahaya
1	Replay token	Token Expire check	Tidak menembus	Low
2	Token Check	Token signature check	Tidak menembus	Low
3	alg=none signature-bypass	CVE-2015-2951	Tidak menembus	High
4	Null signature	CVE-2020-28042	Tidak menembus	Medium
5	Bruteforce	secretkey bruteforcing	Tidak menembus	High
6	X-HTTP-METHOD-Override	CVE-2023-30845	Tidak menembus	High

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah diuraikan pada bab-bab sebelumnya, dapat ditarik kesimpulan sebagai berikut:

1. Setelah melakukan beberapa uji coba penetrasi dengan Teknik yang berbeda maka disimpulkan tidak ditemukan kelemahan yang diuji dapat dieksploitasi pada JSON Web Token pada aplikasi spring boot expenseTracker.
2. Berdasarkan hasil dari penelitian yang telah dilakukan, JSON Web Token tetap

dapat diserang dengan berbagai Teknik penyerangan namun hasil tergantung pada implementasi JSON Web Token pada aplikasi yang dijadikan objek Implementasi.

5. SARAN

Karena pengujian hanya pada JSON Web Token saja dan tidak ditemukan vulnerability yang diuji dapat dieksploitasi. Maka penelitian selanjutnya harus dilakukan pada JWT yang terimplementasi pada aplikasi yang telah berjalan.

DAFTAR PUSTAKA

- Algimantas Venčkauskas, D. K. (2023). Enhancing Microservices Security with Token-Based Access. *Sensors*, 1.
- Irwan Darmawan, M. U. (2023). Evaluasi Keamanan Privilege Terintegrasi JSON Web Token pada Sistem Informasi Akademik. *Jurnal Informasi dan Teknologi*, 3.
- Lenin Leines-Vite, J. C.-A. (2021). INFORMATION AND COMMUNICATION. *International Journal of Network Security & Its Applications (IJNSA)*, 1.
- Microsoft. (2022, Agustus 24). *Microsoft Threat Modeling Tool*. Retrieved from [learn.microsoft.com:https://learn.mi](https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool)
- crosoft.com/en-us/azure/security/develop/threat-modeling-tool
- Microsoft. (2023, Agustus 26). *What is the Windows Subsystem for Linux?* Diambil kembali dari [learn.microsoft.com:https://learn.microsoft.com/en-us/windows/wsl/about](https://learn.microsoft.com/en-us/windows/wsl/about)
- Newman, S. (2021). *Building Microservices designing Fine-Grained System*. Canada: O'Reilly Media, Inc.
- Openwall. (2023, Agustus 20). *John the Ripper password cracker*. Diambil kembali dari [www.openwall.com:https://www.openwall.com/john/](https://www.openwall.com/john/)
- Penetration Execution Standard. (2014, Agustus 16). *Main Page*. Diambil kembali dari [pentest-standard: http://www.pentest-standard.org/index.php/Main_Page](http://www.pentest-standard.org/index.php/Main_Page)
- Pooja Mahindrakar, U. P. (2020). Insights of JSON Web Token. *International Journal of Recent Technology and Engineering (IJRTE)*, 2.
- Sajdak, M. (2023, agustus 20). *JWT (JSON Web Token) (in)security*. Diambil kembali dari [research.securitum.com:https://research.securitum.com/jwt-json-web-token-security/ticarpi](https://research.securitum.com/jwt-json-web-token-security/ticarpi).
- (2023, agustus 20). *The JSON Web Token Toolkit v2*. Diambil kembali dari [github.com:https://github.com/ticarpi/jwt_tool](https://github.com/ticarpi/jwt_tool)