DAFTAR PUSTAKA

27001, I. (2013). *Information Technology - Security Techniques - Information Security Management System - Requirements.* International Organizatioon for Standardization.

Cappelli, D. M. (2012). *The CERT Guide to Insider Threats: How to Prevent Detect, and Respond to Information Technology Crimes.* Syngress.

Cole, E. (2012). *Advanced Persistent Threat: Understanding the Danger and How to Protect Your Organization.* Syngress.

Grossman, J. (2007). *Cross Site Scripting (XSS) Attacks and Defense.* Syngress.

Halfond, W. G. (2006). *A Classification of SQL-Injection Attacks and Countermeasures.* . International Symposium.

International Organization for Standardization. (2018). *Occupational health and safety management systems—Requirements with guidance for use (ISO Standard No. 45001:2018).* Retrieved from https://www.iso.org/standard/63787.html

Jakobsson, M. &. (2007). *Phising and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft.* Wiley.

Jogiyanto, H. M. (2017). *Analisis dan Desain (Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis).* Andi.

Kozierok, C. M. (2005). *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols* .

Kurose, J. F. (2016). *Computer Networking: A Top-Down Approach (7th Edision).* Pearson.

Mitnick, K. D. (2011). *The Art of Deception: Controlling the Human Element of Security*. Wiley.

Mitton, L. (2024). *The Common Vulnerability Scoring System*. Retrieved from splunk.com: https://splunk.com/en_us/blog/learn/cvss-common-vulnerability-scoring-system.html

Pfleeger, C. P. (2007). *Security in Computing.* Prentice Hall.

Richardson, R. &. (2017). Ransomware: Evolution, Mitigation and Prevention. *Kennesaw State University*, 1.

Simamarta, J. (2006). *Pengenalan Teknologi Komputer dan Informasi.* Yogyakarta: Andi. Retrieved from us-cert.gov: https://www.us-cert.gov/sites/default/files/publications/infosecuritybasics.pdf

Simamarta, J. (2006). *Pengenalan Teknologi Komputer dan Informasi.* Yogyakarta: Andi.

Staallings, W. (2017). *Cryptography and Network Security: Principles and Practice (7th Edision).* Pearson.

Stallings, W. (2007). *Data and Computer Communication 7th Edision.*

Stallings, W. (2017). *Network Security Essentials: Applications and Standards (6th Edision).* Pearson.

Tanenbaum, A. S. (2011). *Computer Networks (5th Edision).* Pearson.

What Is Vulnerability Assessment and How Does It Work? (2023). Retrieved from HackerOne: www.hackerone.com

Whitman, M. E. (2010). *Principles of Information Security (4th Edision).* Cengage Learning.

Lampiran 1 *Penetration Testing Report*

Black Box Penetration Testing

For Sikola Unhas

V2.0

August 04th, 2024

By: Andi Nurainun Anugrah AR

Document Properties

Title	Sikola Unhas (Sistem Kelola Pembelajaran)
Version	V2.0
Author	Andi Nurainun Anugrah AR
Pen-testers	Andi Nurainun Anugrah AR
Reviewed By	Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.
Approved By	Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.
Classification	Public

Version control

Version	Date	Author	Description
V2.0	August 04th, 2024	Andi Nurainun Anugrah AR	Final Draft

Ringkasan

Dokumen pelaporan ini disusun sebagai hasil dari analisis keamanan yang dilakukan pada Aplikasi Sikola V2.0 milik Universitas Hasanuddin. Penelitian ini dilakukan dengan menggunakan metode *Penetration Testing*, sebuah pendekatan yang dirancang untuk mengevaluasi tingkat keamanan sistem dengan mensimulasikan berbagai serangan. Tujuan utama dari penelitian ini adalah untuk mengidentifikasi potensi kerentanan dalam aplikasi dan memberikan rekomendasi untuk peningkatan keamanan. Selain itu, laporan ini juga mencakup penyusunan standarisasi dokumen untuk memastikan bahwa proses dan hasil dari penelitian ini dapat dipahami dan diulang oleh pihak lain di masa mendatang. Penulis berharap bahwa temuan dan saran dalam laporan ini akan membantu dalam meningkatkan keamanan Aplikasi Sikola V2.0.

Catatan: Perlu diketahui bahwasanya pada penelitian ini penulis tidak menggunakan Aplikasi Sikola V2.0 milik Universitas Hasanuddin melainkan membuat cloning sistem yang serupa dengan Aplikasi Sikola V2.0 agar tidak menganggu sistem yang sedang berjalan saat ini.

Ruang Lingkup

Pengujian *Penetration Testing* ini dilakukan pada sistem yang dibangun oleh penulis dan diakses melalui localhost yaitu alamat ip default 127.0.0.1. Penulis melakukan pengujian ini menggunakan metode *Blackbox Testing* yang dimana penulis tidak diberikan informasi terkait sistem yang akan dilakukan pengujian.

Tujuan Pengujian

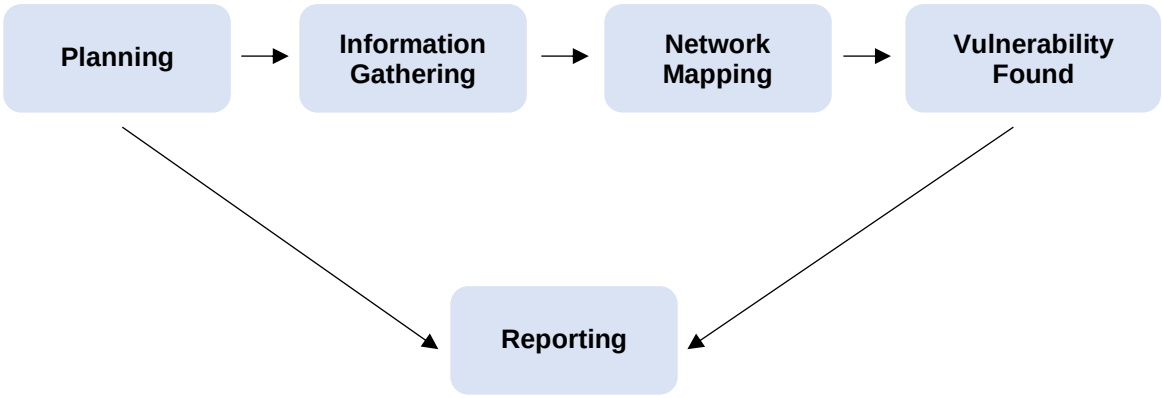
Pengujian *Penetration Testing* ini bertujuan untuk mengidentifikasi kerentanan yang mungkin ada serta memberikan rekomendasi perbaikan terkait kerentanan yang ditemukan pada Aplikasi Sikola V2.0.

Timeline

Waktu pengujian ini secara rinci dijelaskan sebagai berikut:

Penetration Testing	Mulai Tanggal/Waktu	Selesai Tanggal/Waktu
Sikola V2.0	20 Mei 2024	22 Juli 2024

Adapun alur pengujian *Penetration Testing* sebagai berikut:



- *Planning* (Perencanaan) – Identifikasi tujuan dan cakupan dari pengujian dan menetapkan jenis pengujian yang akan dilakukan.
- *Information Gathering* (Pengumpulan Informasi) – Mengumpulkan informasi terkait sistem yang akan diuji.
- *Network Mapping* (Pemindaian Jaringan) – Mengidentifikasi aktifitas jaringan, menemukan sistem dan mencari kerentanan yang mungkin ada.
- *Vulnerability Found* (Temuan Kerentanan) – Menganalisis hasil temuan untuk mengidentifikasi potensi kerentanan yang mungkin dapat dieksploitasi.
- *Reporting* (Pelaporan) – Laporan yang berisi terkait temuan kerentanan, tingkat risiko dan rekomendasi perbaikan untuk mengatasi setiap celah keamanan yang ditemukan.

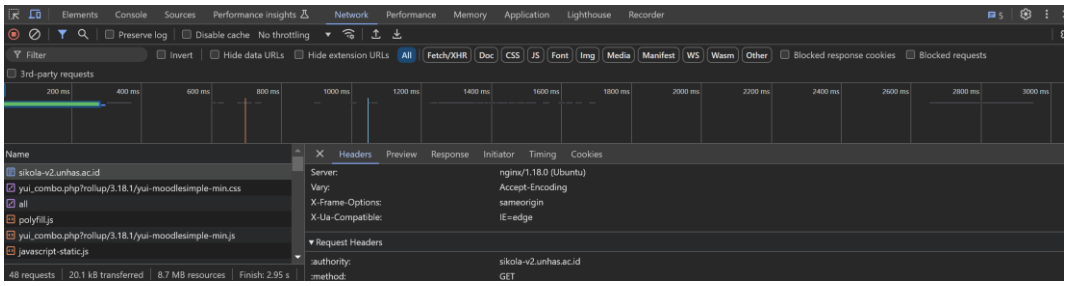
Identifikasi Kerentanan

Information Gathering

Pada tahapan information gathering ini, penulis mencari informasi sebanyak mungkin tentang Sikola V2.0. Adapun hasil pengumpulan informasi sebagai berikut:

Web Inspector

Dengan *Web Inspector* informasi terkait website seperti server serta menggunakan versi berapa dan bahasa pemrograman menggunakan *Web Inspector*. Hasil pengumpulan informasi dapat dilihat pada Gambar 9 sebagai berikut:



Moodle LMS (*Learning Management System*)

Sikola V2.0 menggunakan platform Moodle LMS sebagai sistem manajemen pembelajaran mereka. Oleh karena itu, penulis perlu mengetahui versi Moodle yang digunakan pada Sikola V2.0. Mengetahui versi Moodle ini penting untuk memastikan kompatibilitas fitur dan tingkat keamanan yang ada pada sistem yang digunakan. Berikut adalah versi Moodle yang digunakan pada Sikola V2.0:

Moodle 4.2 (Build: 20230424)
Copyright © 1999 onwards, Martin Dougiamas
and many other contributors.
GNU Public License

Dapat dilihat pada gambar diatas aplikasi Sikola V2.0 menggunakan version Moodle 4.2 (Build: 20230424). Berikut penjelasan terkait version tersebut:

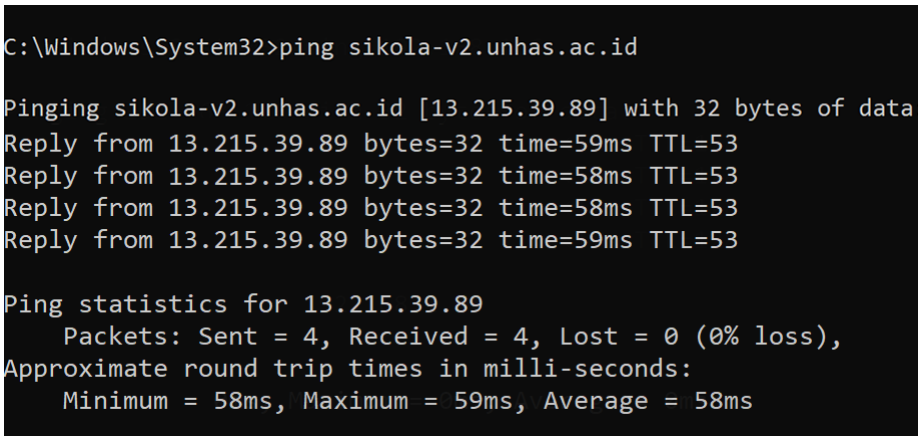
1. Moodle 4.2: Versi utama dari platform *Learning Management System* (LMS) Moodle. Nomor versi 4.2 menunjukkan bahwa versi ini rilis kedua dari generasi keempat Moodle.
2. Build: 20230424: Tanggal rilis dari versi tersebut yaitu pada tanggal 24 April 2023.

Network Mapping

Network scanning tahap dimana penulis mengidentifikasi aktifitas jaringan, menemukan sistem, dan mencari kelemahan yang mungkin ada. Berikut hasil network mapping yang telah dilakukan pada Sikola V2.0:

Ping

Untuk mengetahui respons jaringan dari server Sikola V2.0 dilakukan pengujian dasar untuk memastikan bahwa server dapat dijangkau dan berfungsi menggunakan perintah ping di Commad Prompt (CMD). Berikut adalah hasil ping yang telah dilakukan pada Sikola V2.0:

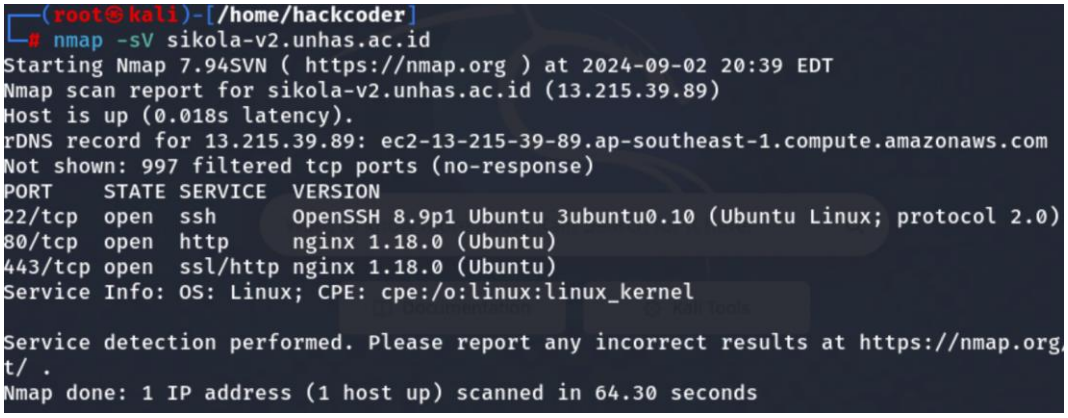


Perintah Ping: ping 192.168.1.6

Dapat dilihat hasil diatas menunjukkan bahwa paket terkirim 4, paket diterima 4, paket hilang 0 hasil tersebut menunjukkan semua paket diterima, tidak ada kehilangan paket dan server berfungsi dengan baik.

Port Scanning

Dalam melakukan *Port Scanning*, peneliti menggunakan NMAP v7.94 dengan perintah nmap -sV ((untuk mendeteksi versi layanan yang berjalan pada port yang terbuka))[Alamat IP]. Adapun informasi terkait Alamat IP (*Internet Protocol*) *Address* yaitu 13.215.39.89. Berikut adalah hasil *port scanning* pada server Sikola V2.0:



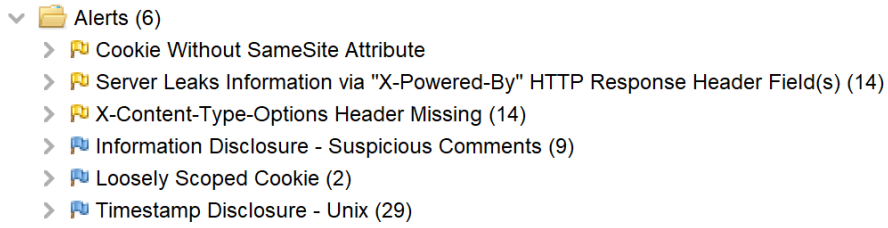
Penjelasan lebih detail dijelaskan dalam Tabel sebagai berikut:

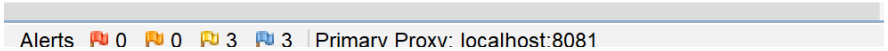
No.	*Ports*	*Status*	*Service*	*Version*
1.	22	Open	OpenSSH	OpenSSH 8.9
2.	80	Open	HTTP	Nginx 1.18.0
3.	443	Open	SSL	Nginx 1.18.0

Vulnerability Found

Pada tahap ini penulis menganalisis hasil dari pengumpulan informasi untuk mengidentifikasi kemungkinan potensi kerentanan yang mungkin dapat dieksploitasi. Berikut hasil vulnerability found yang telah dilakukan pada Sikola V2.0:

OWASP ZAP





Berdasarkan hasil pemindaian menggunakan OWASP ZAP ditemukan 3 kerentanan kategori *low risk* dan 3 kerentanan yang bersifat *informational*. Untuk penjelasan yang lebih rinci, penulis telah merinci hasil pemindaian yang dapat dilihat pada Tabel sebagai berikut:

No.	Celah Keamanan	Tingkat Ancaman	Deskripsi	Dampak
1.	*Cookie Without SameSite Attribute*	*Low Risk* (Rendah)	Cookie telah disetel tanpa atribut SameSite	Cookie dapat dikirim sebagai hasil permintaan 'lintas situs'
2.	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field (s)*	*Low Risk* (Rendah)	Server web/aplikasi membocorkan informasi melalui satu atau lebih header respons HTTP "*X-Powered-By*".	Akses terhadap informasi tersebut dapat memudahkan penyerang mengidentifikasi kerangka kerja/komponen lain yang diandalkan oleh aplikasi web dan kerentanan yang mungkin dimiliki komponen tersebut.

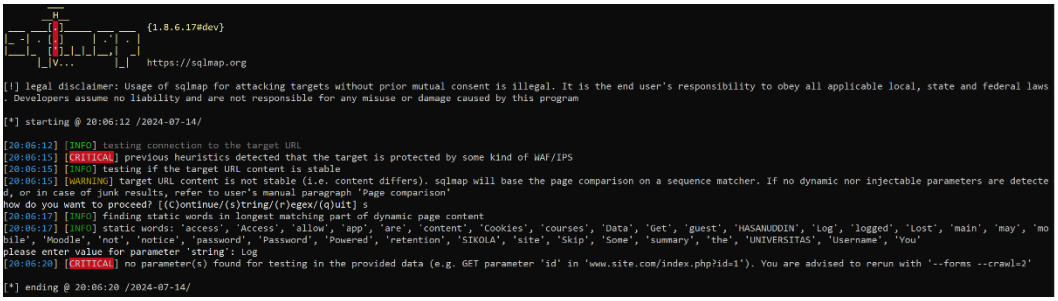
3.	*X-Content-Type-Options Header Missing*	*Low Risk* (Rendah)	Header Anti-MIME-Sniffing X-Content-Type-Options tidak disetel ke 'nosniff'	Hal ini memungkinkan versi Internet Explorer dan Chrome yang lebih lama untuk melakukan MIME-sniffing pada isi respons, yang berpotensi menyebabkan isi respons ditafsirkan dan ditampilkan sebagai tipe konten selain tipe konten yang dinyatakan.
4.	*Information Disclosure – Suspicious Comments*	*Informational*	Ketika komentar yang mencurigakan atau sensitif ditemukan dalam *source code* aplikasi. Komentar ini mungkin berisi informasi penting seperti catatan pengembangan, jalur file sistem, atau petunjuk tentang bagian kode yang rentan.	Komentar yang mencurigakan atau sensitif dapat mengungkapkan informasi tentang kelemahan keamanan atau struktur aplikasi, yang dapat membantu penyerang dalam merencanakan serangan mereka.
5.	*Loosely Scoped Cookie*	*Informational*	Cookie disetel dengan cakupan domain yang luas	Cookie yang memiliki cakupan domain yang luas dapat mempermudah pencurian sesi atau akses tidak sah oleh subdomain yang mungkin tidak aman, yang kemudian bisa

				dieksploitasi lebih lanjut oleh penyerang.
6.	Timestamp Disclosure - Unix	*Informational*	Timestamp ditampilkan oleh aplikasi atau server web	Konfirmasi secara manual bahwa data timestamp tidak bersifat sensitif dan tidak memungkinkan pengumpulan data untuk mengungkap pola yang dapat dieksploitasi.

Simulasi Serangan:

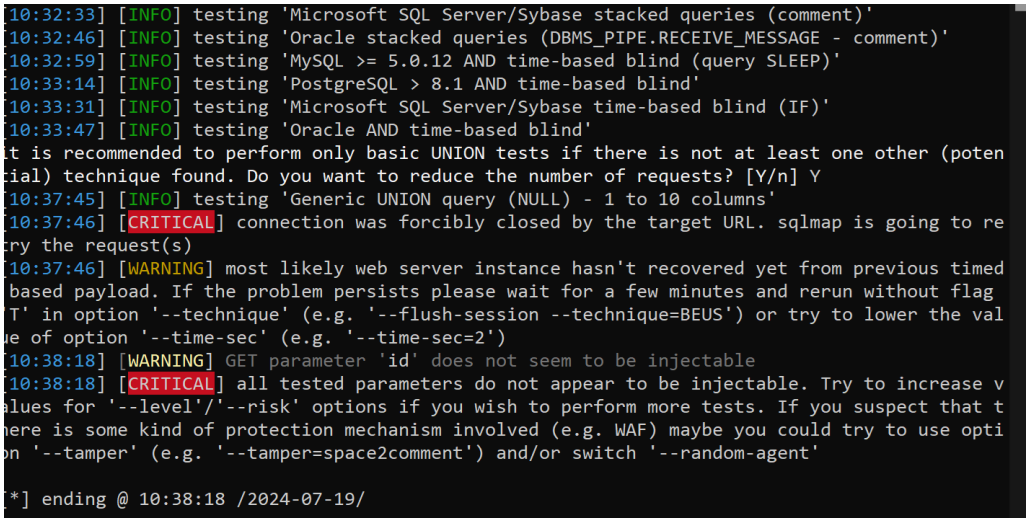
SQL *Injection*

Penulis menggunakan *tools* sqlmap untuk mengidentifikasi dan mengeksploitasi kerentanan SQL *Injection* pada Sikola V2.0. Berikut adalah hasil pemindaian Sqlmap:

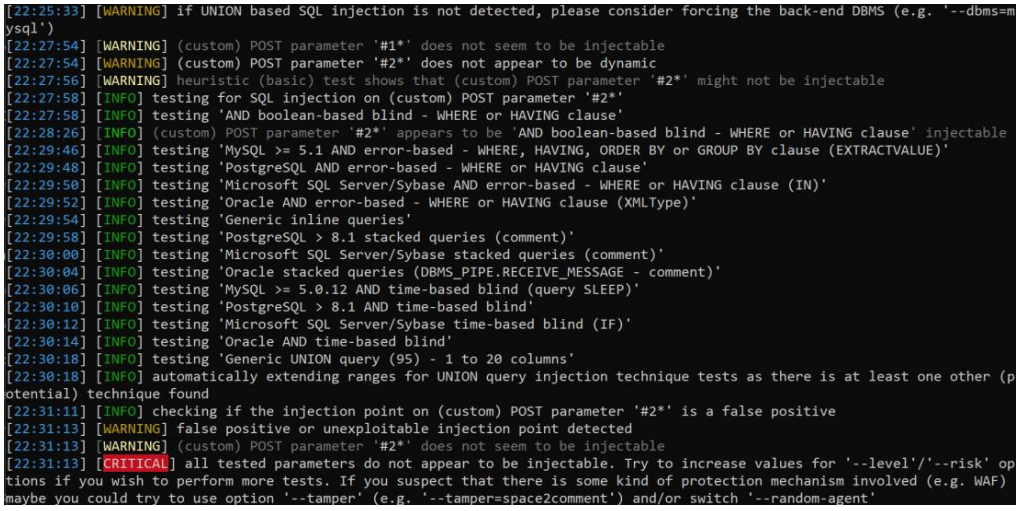


Dapat dilihat pengujian kerentanan SQL *Injection* untuk mencari parameter yang mungkin rentan terhadap SQL *Injection* pada Sikola V2.0 dengan berbagai string seperti ‘access’, ‘Access, ‘allow’, ‘app’, ‘are’, ‘content’, ‘Cookies, ‘courses’, ‘Data’, ‘Get’, ‘guest’, ‘HASANUDDIN’, ‘Log’, ‘logged’, ‘Lost’, ‘main’, ‘may’, ‘mobile’, ‘Moodle’, ‘not’, ‘notice’, ‘password’, ‘Powered’, ‘retention’, ‘SIKOLA’, ‘site’, ‘Skip’, ‘Some’, ‘Some’, ‘summary’, ‘the’, ‘UNIVERSITAS’, ‘You’. Setelah

dilakukan pengecekan untuk setiap string tidak ditemukan adanya parameter yang dapat dilakukan SQL *Injection* pada setiap string tersebut.



Pengujian kerentanan SQL Injection pada method GET parameter "id" dengan pengujian berbagai payload tidak ditemukan adanya kerentanan terhadap SQL Injection pada parameter "id".



Pengujian kerentanan SQL Injection pada method POST parameter "#1" dan "#2" pada parameter "#1" ditemukan adanya kerentanan SQL Injection. Namun pada parameter "#2" percobaan payload '**AND Boolean-based blind – WHERE or Having clause**' injectable atau rentan terhadap SQL Injection. Setelah dilakukan pengecekan kembali pada POST parameter "#2" dinyatakan false positive yang artinya tidak dapat dieksploitasi atau merupakan hasil positif palsu, hal tersebut menunjukkan bahwa meskipun ada indikasi awal terkait kerentanan, namun pada hasil akhir menunjukkan parameter tersebut tidak rentan atau tidak dapat dieksploitasi. Setelah dilakukan pemindaian kerentanan terhadap SQL Injection

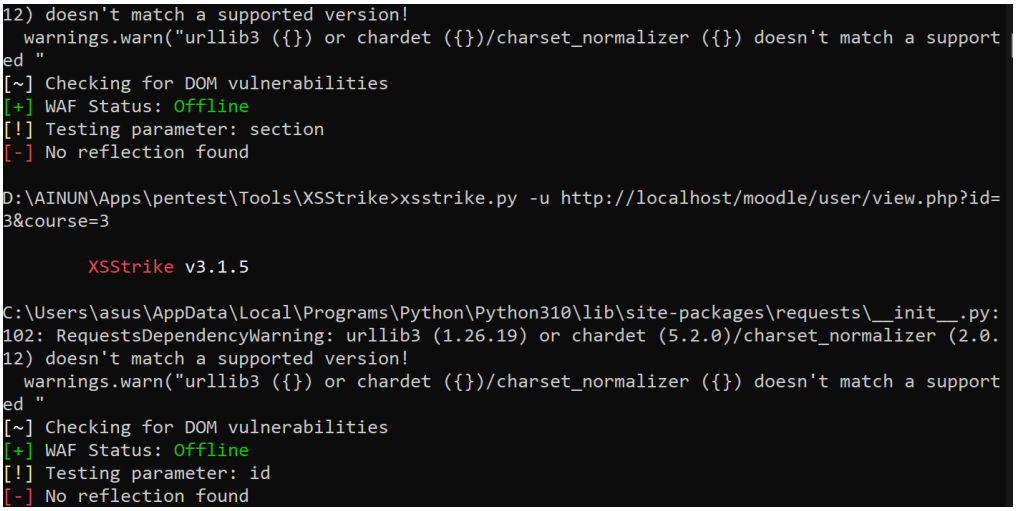
menggunakan sqlmap, penulis menyimpulkan berdasarkan data yang telah dijelaskan di atas bahwa tidak ada kerentanan SQL Injection pada aplikasi Sikola V2.0.

Cross-Site Scripting (XSS)

Pada pengujian kerentanan Cross-Site Scripting (XSS) penulis menggunakan tools dan eksploitasi manual untuk mengidentifikasi kerentanan XSS. Biasanya, kerentanan XSS ini berada pada inputan form, URL parameter, dan area input teks lainnya yang tidak divalidasi atau disanitasi dengan benar. Berikut adalah hasil pemindaian Cross-Site Scripting (XSS) pada Sikola V2.0:

XSStrike

Penulis menggunakan *tools* xsstrike untuk mengidentifikasi kerentanan XSS. Berikut adalah hasil pemindaian *Cross-Site Scripting* (XSS):



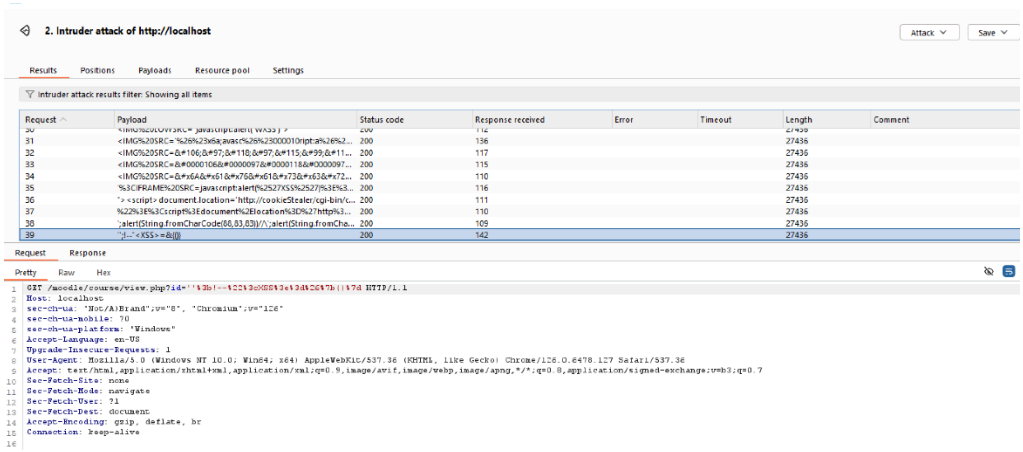
Pengujian kerentanan Cross-Site Scripting (XSS) pada parameter ‘section’ dan ‘id’ menunjukkan bahwa "No reflection found" artinya payload yang disuntikkan ke dalam aplikasi web tidak menghasilkan respons dari server. Ini menunjukkan bahwa tidak ada tempat dalam aplikasi web yang mencerminkan input pengguna tanpa disanitasi atau divalidasi dengan benar. Oleh karena itu, kerentanan XSS tidak ditemukan pada parameter ‘section’ dan ‘id’.

```
D:\AINUN\Apps\pentest\Tools\XSStrike>xsstrike.py -u http://localhost/moodle/report/view.php?c
ourseid=4

        XSStrike v3.1.5

C:\Users\asus\AppData\Local\Programs\Python\Python310\lib\site-packages\requests\__init__.py:
102: RequestsDependencyWarning: urllib3 (1.26.19) or chardet (5.2.0)/charset_normalizer (2.0.
12) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({})/charset_normalizer ({}) doesn't match a support
ed "
[~] Checking for DOM vulnerabilities
[+] WAF Status: Offline
[!] Testing parameter: courseid
[-] No reflection found
```

Pengujian kerentanan *Cross-Site Scripting* (XSS) pada parameter ‘courseid’ menunjukkan bahwa “*No reflection found*”. Oleh karena itu, kerentanan XSS tidak ditemukan pada parameter ‘courseid’.

Burpsuite

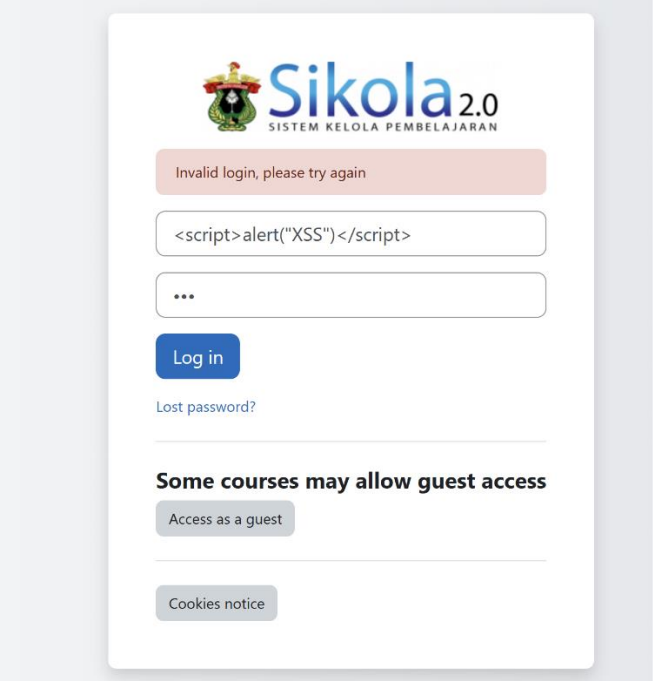
Penulis juga menggunakan Burpsuite untuk mengidentifikasi kerentanan *Cross-Site Scripting* (XSS). Berikut adalah hasil pemindaian Burpsuite:



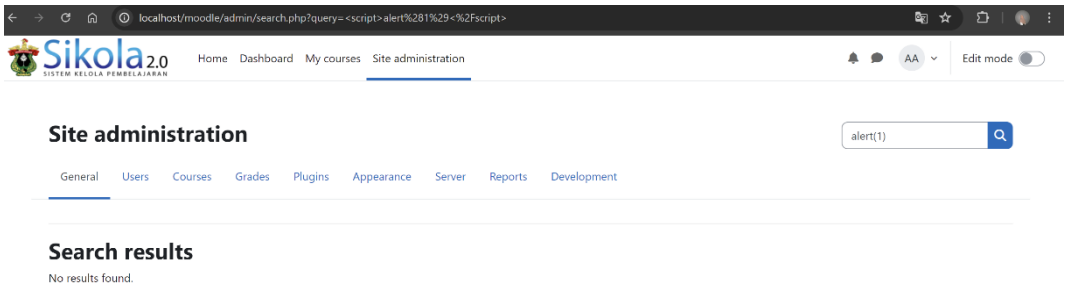
Pengujian kerentanan *Cross-Site Scripting* (XSS) method GET pada path /moodle/course/view.php parameter ‘id’ pengujian dengan 39 payload yang dimasukkan pada parameter tersebut hasilnya tidak menunjukkan adanya kerentanan *Cross-Site Scripting* (XSS).

Eksploitasi Manual

Penulis juga melakukan eksploitasi manual pada setiap input yang terdapat pada Sikola V2.0 untuk mengidentifikasi kemungkinan adanya kerentanan *Cross-Site Scripting* (XSS) pada sistem Moodle tersebut. Adapun input form yang digunakan untuk menguji kerentanan Cross-Site Scripting (XSS) adalah sebagai berikut:



Pada halaman *login page*, inputan *username* penulis memasukkan payload "<script>alert("XSS")</script>". Hasilnya menunjukkan bahwa server merespon script tersebut dengan pesan "*Invalid login, please try again*" mengindikasikan bahwa input *username* disanitasi atau divalidasi dengan benar. Hal ini menunjukkan bahwa tidak ada kerentanan XSS pada input tersebut.

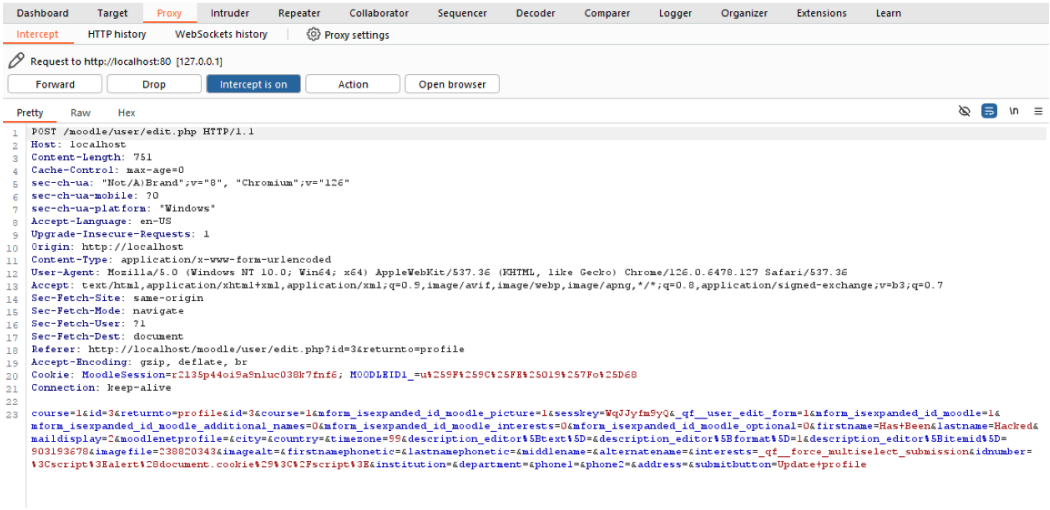


Pada halaman *Site administration*, inputan search penulis memasukkan payload "<script>alert(1)</script>". Hasil menunjukkan bahwa URL server merespons skrip tersebut sebagai teks biasa dan tidak mengeksekusi skrip tersebut sebagai

kode. Ini mengindikasikan bahwa input tersebut divalidasi dengan benar dan tidak rentan terhadap serangan XSS.

Cross-Site Request Forgery (CSRF)

Dalam pengujian kerentanan *Cross-Site Request Forgery* (CSRF), penulis menggunakan Burpsuite untuk mengidentifikasi potensi kerentanan CSRF. Berikut ini adalah hasil dari pemindaian CSRF pada Sikola V2.0:



Pengujian kerentanan *Cross Site Request Forgery* (CSRF) pada halaman edit profil menunjukkan bahwa tidak ada CSRF Token yang ditemukan dalam *method* POST. Ini menunjukkan bahwa halaman tersebut mungkin rentan terhadap serangan CSRF, yang dapat memungkinkan penyerang mengirimkan permintaan tidak sah atas nama pengguna yang telah terotentikasi. Berikut ini adalah hasil dari eksploitasi yang telah dilakukan:

Eksploitasi

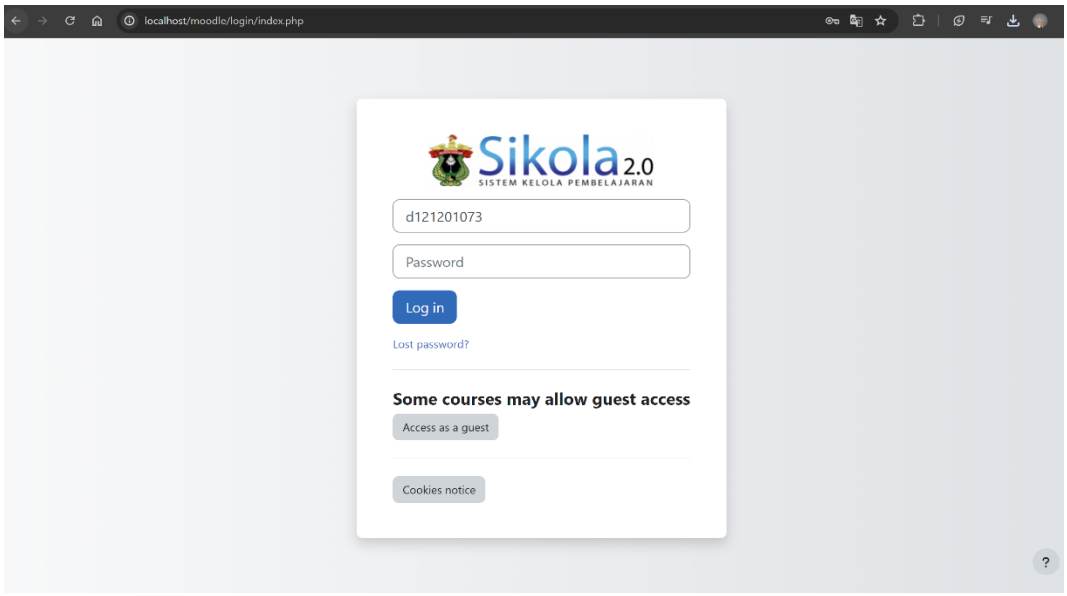
Penulis menggunakan CSRF PoC *Generator* untuk menghasilkan serangan permintaan CSRF dalam format HTML. HTML ini kemudian menggunakan form dan JavaScript untuk menghasilkan permintaan yang diperlukan di browser. CSRF PoC *Generator* yang digunakan oleh penulis dapat ditemukan di situs web https://security.love/CSRF-PoC-Genorator/. Berikut adalah hasil CSRF PoC *Generator*:

mform_isexpanded_id_moodle	1
mform_isexpanded_id_moodle_additional_names	0
mform_isexpanded_id_moodle_interests	0
mform_isexpanded_id_moodle_optional	0
firstname	Has+Been
lastname	Hacked
maildisplay	2
moodlenetprofile	
city	
country	
timezone	99
description_editor%5Btext%5D	
description_editor%5Bformat%5D	1
description_editor%5Bitemid%5D	903193678
imagefile	238820343
imagealt	
firstnamephonetic	
lastnamephonetic	
middlename	
alternatename	
interests	_qf__force_multiselect_sub
idnumber	<script>alert(document.coo
institution	
department	
phone1	
phone2	
address	
submitbutton	Update+profile

http://localhost/moodle/user/edit.php

Form tersebut menampilkan data hasil method POST BurpSuite yang telah diinterpretasikan ulang oleh CSRF PoC Generator. Di sini, penulis mencoba mengubah nama pengguna dari ‘D121201073 Andi Nurainun Anugrah AR’

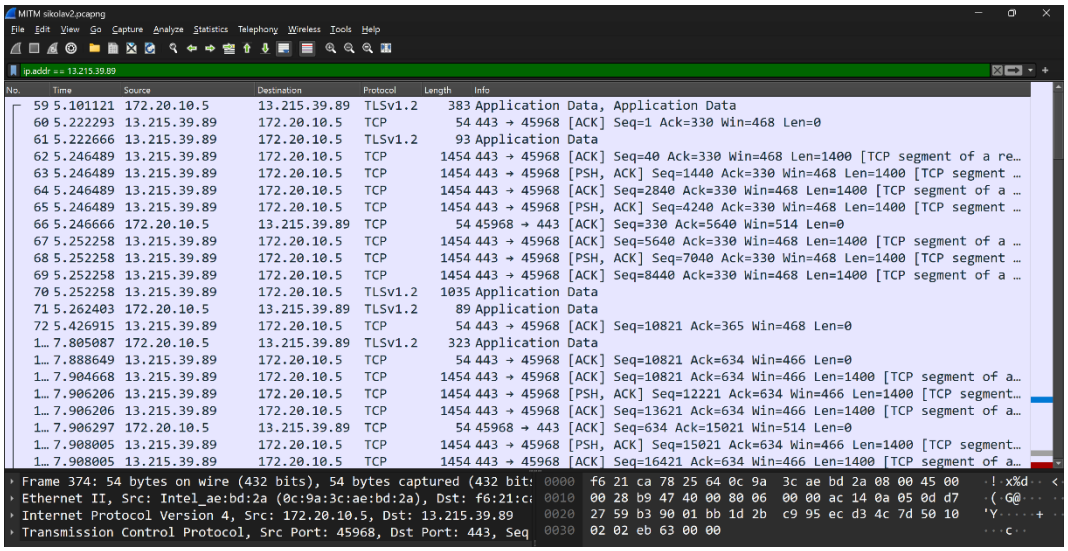
menjadi ‘Has Been Hacked’. Berikut ini adalah hasil dari upaya eksploitasi CSRF tersebut:



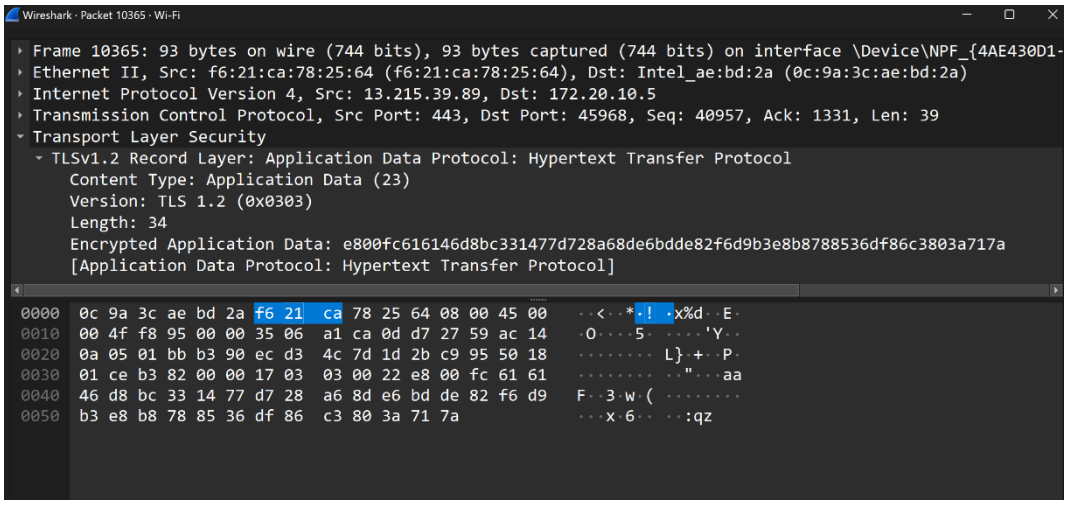
Upaya untuk mengeksploitasi kerentanan CSRF Token menggunakan CSRF PoC *Generator* tidak berhasil. Proses eksploitasi ini berhenti pada halaman login Sikola V2.0.

***Man-in-the-Middle* (MITM)**

Dalam pengujian kerentanan *Man in the Middle* (MITM) penulis menggunakan Wireshark untuk memantau dan menganalisis lalu lintas jaringan secara *real time* untuk mengidentifikasi serangan MITM. Berikut ini adalah hasil dari pemindaian MITM pada Sikola V2.0:



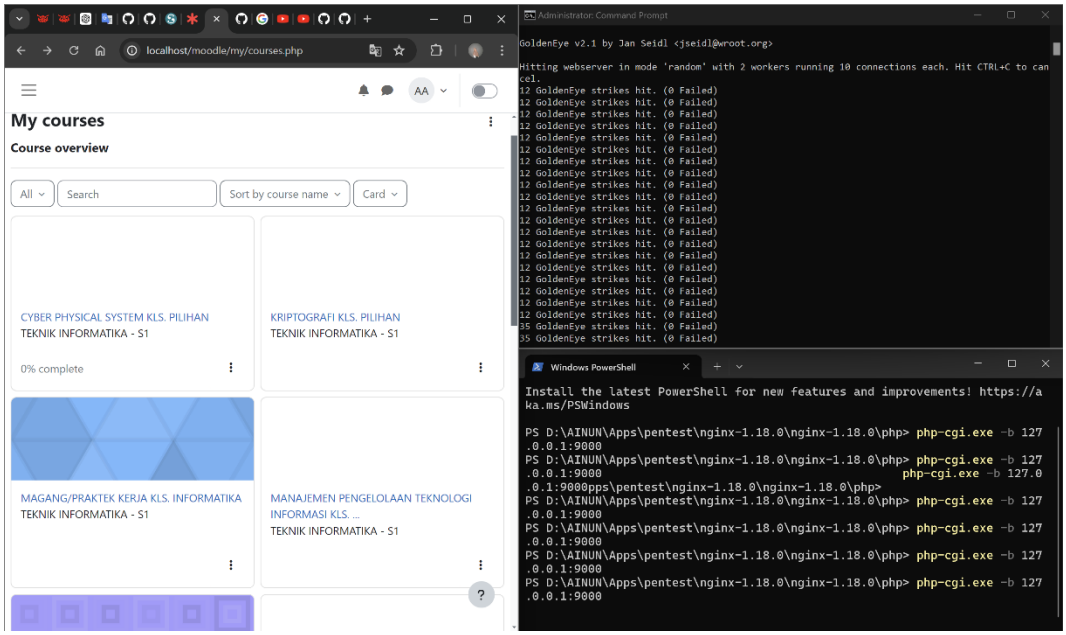
Dalam pengujian *Man in the Middle* (MITM), penulis menemukan bahwa semua aktivitas di Sikola V2.0 menggunakan protokol HTTPS (*Hypertext Transfer Protocol Secure*) untuk komunikasi antara server dan pengguna. Berikut ini adalah hasil *capture* komunikasi dan transfer data menggunakan protokol TLS:



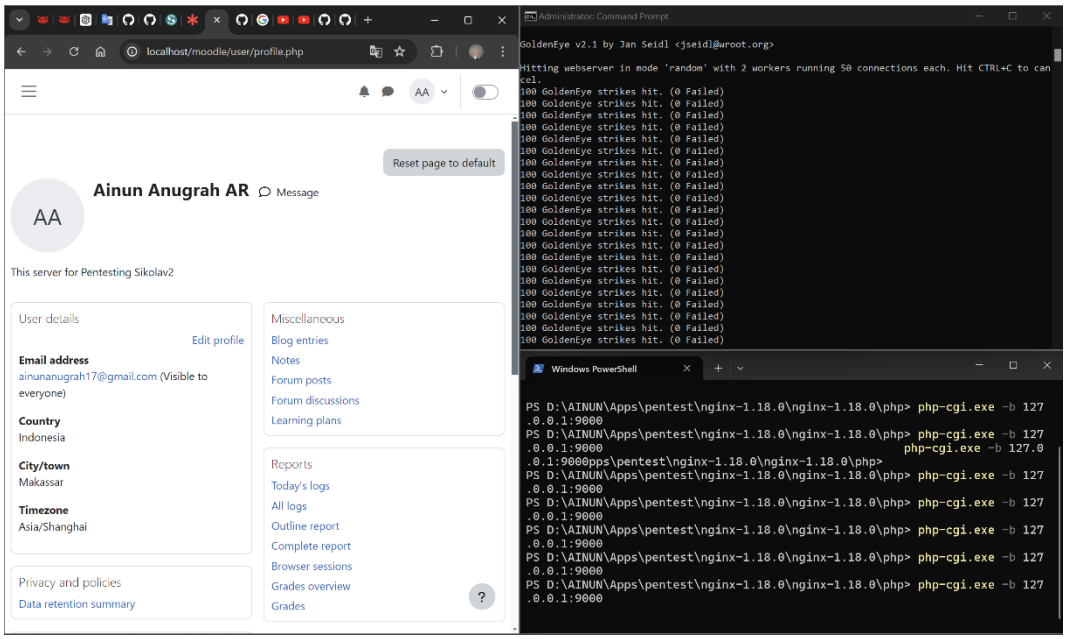
Hasil yang ditampilkan di atas menunjukkan bahwa komunikasi dan transfer data antara pengguna dan server dienkripsi dengan *Transport Layer Security* (TLS). Ini menunjukkan bahwa tidak ada kerentanan yang ditemukan terhadap serangan MITM.

***Denial of Service* (DoS)**

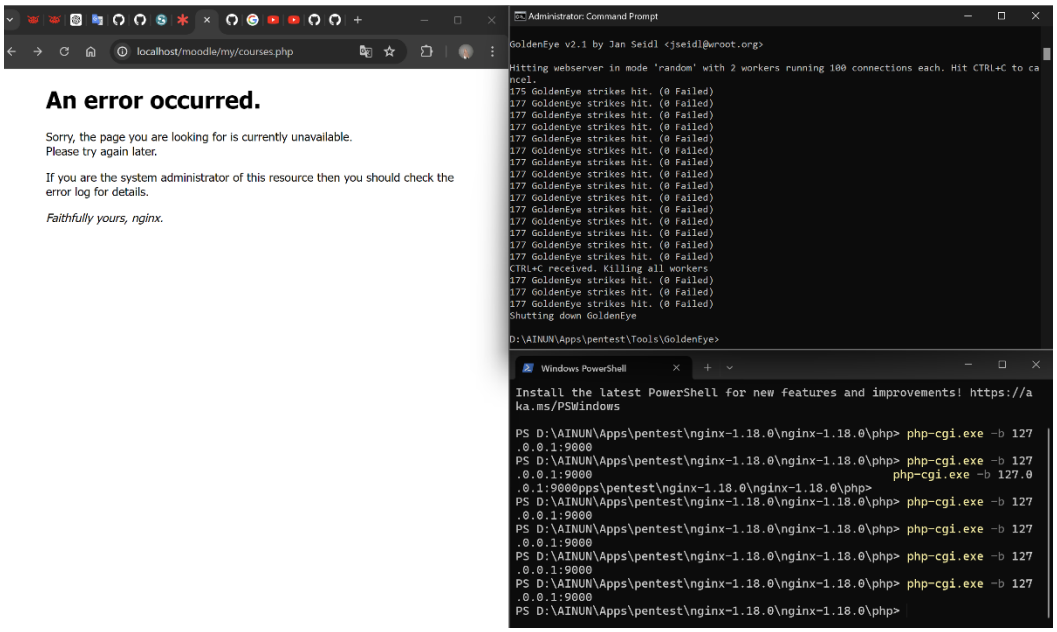
Dalam pengujian kerentanan *Denial of Service* (DoS), penulis menggunakan *tools* GoldenEye untuk menguji kerentanan Sikola V2.0 terhadap serangan DoS. Berikut adalah hasil *Denial of Service* (DoS) pada Sikola V2.0:



Dalam pengujian DoS dengan serangan menggunakan dua pengguna, masing-masing dengan 10 koneksi. Hasil menunjukkan bahwa server tetap stabil dan tidak mengalami gangguan.



Dalam pengujian DoS dengan serangan menggunakan dua pengguna, masing-masing dengan 50 koneksi. Hasil menunjukkan bahwa server masih tetap stabil dan tidak mengalami gangguan.



Dalam pengujian sebelumnya diketahui server tetap stabil dengan masing-masing koneksi selanjutnya penulis mencoba dengan serangan menggunakan dua pengguna, masing-masing dengan 100 koneksi. Hasil menunjukkan bahwa server mengalami down dan tidak dapat diakses.

Berdasarkan pengujian yang telah dilakukan, dapat disimpulkan bahwa server Sikola V2.0 memiliki batas toleransi terhadap serangan Denial of Service (DoS). Ketika serangan dilakukan dengan dua pengguna dan masing-masing memiliki 10 koneksi, server mampu menangani beban dan tetap stabil. Namun, ketika beban ditingkatkan menjadi 100 koneksi per pengguna, server tidak mampu menangani beban tersebut dan akhirnya mengalami gangguan. Ini menunjukkan bahwa meskipun server memiliki beberapa tingkat perlindungan terhadap serangan DoS, masih ada batas atas untuk jumlah koneksi simultan yang dapat ditangani sebelum mengalami gangguan.

Penilaian Kerentanan

Tabel berikut mendefinisikan tingkat keparahan dan rentang skor CVSS yang sesuai untuk menilai kerentanan dan dampak risiko:

Kategori	CVSS V3.1 Rentang skor	Definisi
Critical	9.0 – 10.0	Eksploitasi mudah dilakukan dan biasanya mengakibatkan penyusupan di tingkat sistem.
High	7.0 – 8.9	Eksploitasi lebih sulit tetapi dapat menyebabkan peningkatan hak istimewa dan kemungkinan hilangnya data atau waktu henti.
Medium	4.0 – 6.9	Kerentanan memang ada tetapi tidak dapat dieksploitasi atau memerlukan langkah-langkah tambahan seperti rekayasa sosial.
Low	0.1 – 3.9	Kerentanan tidak dapat dieksploitasi tetapi akan mengurangi permukaan serangan organisasi.
Informational	N/A	Tidak ada kerentanan. Informasi tambahan diberikan terkait item yang ditemukan selama pengujian dan dokumentasi tambahan.

Ringkasan Kerentanan & Pelaporan

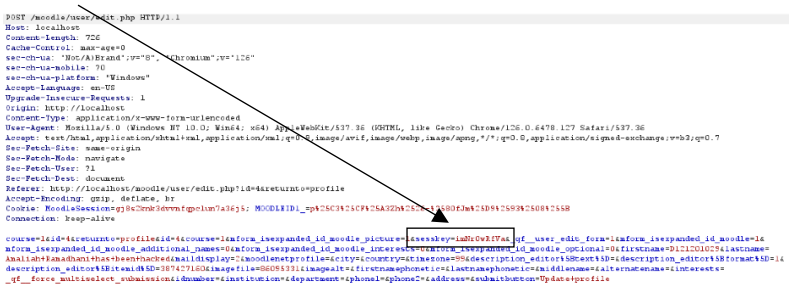
Tabel berikut menggambarkan kerentanan yang ditemukan berdasarkan dampak dan tindakan perbaikan yang direkomendasikan:

Critical	***High***	***Medium***	***Low***	***Informational***
0	0	2	2	4

Temuan	Kategori	Rekomendasi
Penetration Testing		
Cross Site Request Forgery (CSRF)	***Medium***	Implementasikan CSRF Token
Denial of Service (DoS)	***Medium***	- Konfigurasikan server - Gunakan *Firewall*
Cookie Without SameSite Attribute	***Low***	- Tambahkan atribut SameSite - Gunakan secure cookie
X-Content-Type-Options Header Missing	***Low***	- Implementasikan WAF - Tambahkan header *X-Content-Type-Options*
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	***Informational***	Implementasikan WAF (*Web Application Firewall*)
Information Disclosure - Suspicious Comments	***Informational***	Disarankan menghapus semua komentar pada codingan yang mengandung informasi sensitif
Loosely Scoped Cookie	***Informational***	- Setel Cookie - Gunakan atribut HttpOnly dan Secure - Gunakan atribut SameSite
Timestamp Disclosure - Unix	***Informational***	Konfirmasikan secara manual

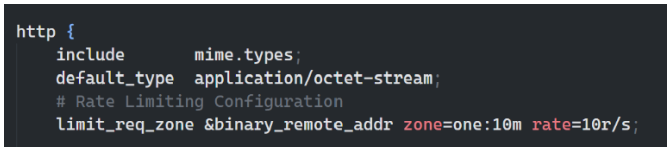
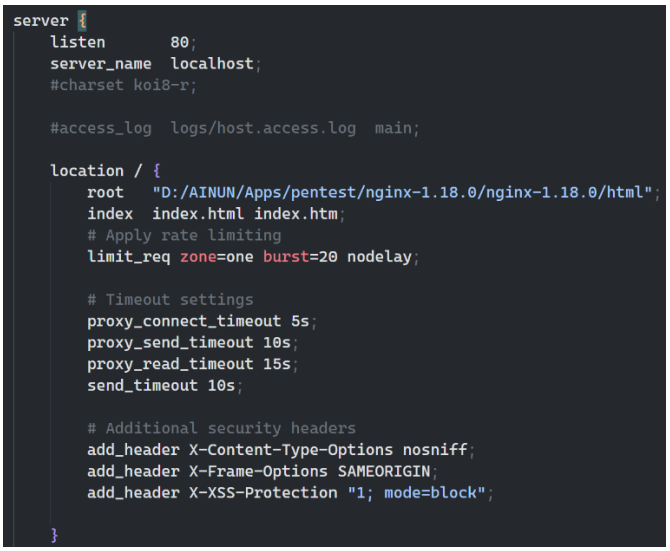
Lampiran: Detail Kerentanan dan Mitigasi

Temuan PT-001: Cross-Site Request Forgery (*Medium*)

Deskripsi	Penyerang dapat melakukan tindakan yang tidak diinginkan oleh pengguna dengan mengelabui peramban web.
Dampak	Tidak ditemukan adanya CSRF Token pada halaman login, memungkinkan rentan. Namun setelah dilakukan eksploitasi tidak menunjukkan keberhasilan maka dinyatakan tidak berhasil.
Rekomendasi	Implementasikan CSRF Token pada setiap permintaan yang mengubah status.
Implementasi	Moodle telah menggunakan CSRF Token dengan value ’sesskey’: 
CVSS Skor	4.3

Temuan PT-002: Denial of Service (*Medium*)

Deskripsi	Serangan dimana penyerang berusaha membuat jaringan tidak tersedia bagi pengguna
Dampak	Server mengalami down dan tidak dapat diakses oleh pengguna yang sah.
Rekomendasi	- Konfigurasikan server agar dapat menangani jumlah koneksi maksimum - Gunakan Firewall
Implementasi	- Implementasikan WAF untuk memfilter dan memantau lalu lintas HTTP. WAF dapat memblokir permintaan yang mencurigakan dan melindungi dari serangan DoS. ModSecurity adalah contoh WAF yang dapat digunakan untuk menambah lapisan keamanan pada server. - Konfigurasi Server Nginx (nginx.conf)

	Dengan menambahkan rate limiting dan timeout settings baris kode http dan server:   - Rate Limiting: Mencegah klien tunggal (atau bot) melakukan permintaan yang berlebihan dan membebani server. - Timeout Settings: Mengatur timeout pada koneksi klien untuk mengurangi dampak serangan DoS yang mencoba menguras sumber daya server dengan menjaga koneksi tetap terbuka terlalu lama.
CVSS Skor	6.6

Temuan PT-003: Cookie Without SameSite (*Low*)

Deskripsi	Cookie diatur tanpa atribut SameSite
Dampak	Cookie dapat dikirim sebagai hasil permintaan 'lintas situs'
Rekomendasi	- Tambahkan Atribut ‘SameSite’ pada cookie - Gunakan Secure Cookie agar dikirim hanya melalui koneksi HTTPS
Implementasi	Konfigurasi Pengaturan di PHP (php.ini) Dengan mengisi ‘session.cookie_samesite=’ dengan value ‘Lax’ agar cookie tidak akan dikirim pada permintaan cross-site biasa (misalnya, mengklik link), tetapi dikirim pada permintaan GET yang memulai navigasi ke URL.

	; Add SameSite attribute to cookie to help mitigate Cross-Site Request Forgery (CSRF/XSRF) ; Current valid values are "Strict", "Lax" or "None". When using "None", ; make sure to include the quotes, as `none` is interpreted like `false` in ini files. ; https://tools.ietf.org/html/draft-west-first-party-cookies-07 session.cookie_samesite = Lax
CVSS Skor	3.0

Temuan PT-004: Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (*Informational*)

Deskripsi	Informasi terkait kerangka kerja atau komponen yang digunakan oleh aplikasi
Dampak	Akses terhadap informasi tersebut dapat memudahkan penyerang mengidentifikasi kerangka kerja/komponen lain yang diandalkan oleh aplikasi web dan kerentanan yang mungkin dimiliki komponen tersebut.
Rekomendasi	Gunakan WAF (Web Application Firewall) untuk memanipulasi dan menyembunyikan header HTTP yang mengungkapkan informasi sensitif.
Implementasi	ModSecurity adalah WAF yang dapat digunakan untuk menambah lapisan keamanan pada server. ModSecurity dapat mendeteksi serangan sebelum mencapai server aplikasi web dan, jika dikonfigurasi dengan benar, dapat memblokir serangan sebelum merusak sistem.
CVSS Skor	0.0

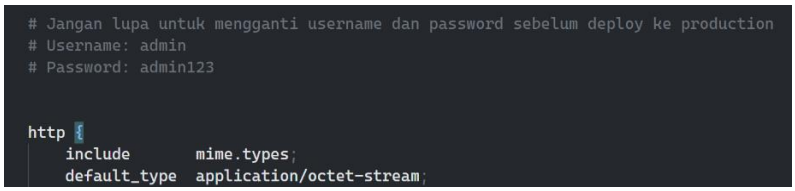
Temuan PT-005: X-Content-Type-Options Header Missing (*Low*)

Deskripsi	Ketika server web tidak menetapkan header 'X-Content-Type-Options' dalam responnya.
Dampak	Hal ini memungkinkan versi Internet Explorer dan Chrome yang lebih lama untuk melakukan MIME-sniffing pada isi respons, yang berpotensi menyebabkan isi respons ditafsirkan dan ditampilkan sebagai tipe konten selain tipe konten yang dinyatakan.
Rekomendasi	- Gunakan WAF untuk menambahkan atau memastikan header X-Content-Type-Options ada di semua respon HTTP.

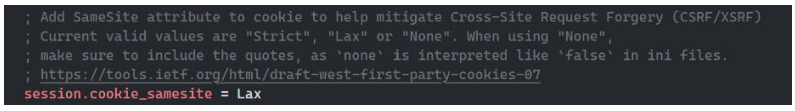
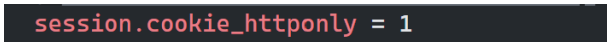
	- Tambahkan header X-Content-Type-Options dengan nilai nosniff di konfigurasi server web
Implementasi	- ModSecurity dapat menjadi WAF yang digunakan untuk menambah lapisan keamanan pada server. - Konfigurasi Pengaturan Nginx (nginx.conf) Dengan menambahkan add_header X-Content-Type-Options “nosniff” always; (Dengan menambahkan header ini, browser hanya akan menggunakan tipe konten yang dikirimkan oleh server, mengurangi risiko serangan MIME type confusion, di mana peramban menganggap suatu file berjenis lain yang dapat mengeksekusi kode berbahaya.). add_header X-XSS-Protection “1; mode=block” always; (Jika browser mendeteksi potensi serangan XSS, dengan mode block, halaman tersebut akan diblokir sepenuhnya daripada mengeksekusi skrip berbahaya. Ini membantu melindungi pengguna dari skrip jahat yang dapat mencuri informasi atau melakukan tindakan berbahaya di situs tersebut). `server { listen 80; server_name localhost; # Additional security headers add_header X-Content-Type-Options "nosniff" always; add_header X-XSS-Protection "1; mode=block" always;`
CVSS Skor	3.0

Temuan PT-006: Information Disclosure – Suspicious Comments (*Informational*)

Deskripsi	Komentar pada codingan
Dampak	Komentar yang sensitif dapat mengungkapkan informasi tentang kelemahan keamanan atau struktur aplikasi, yang

	dapat membantu penyerang dalam merencanakan serangan mereka.
Rekomendasi	Disarankan untuk menghapus semua komentar yang dapat memberikan informasi bagi penyerang.
Implementasi	Contoh Komentar Informasi Sensitif: 
CVSS Skor	0.0

Temuan PT-007: Loosely Scoped Cookie (*Informational*)

Deskripsi	Cookie web diatur dengan cakupan yang terlalu luas
Dampak	Cookie yang memiliki cakupan domain yang luas dapat mempermudah pencurian sesi atau akses tidak sah oleh subdomain yang mungkin tidak aman, yang kemudian bisa dieksploitasi lebih lanjut oleh penyerang.
Rekomendasi	Setel Cookie dengan Scope yang lebih ketat: - Gunakan Atribut 'HttpOnly' dan 'Secure' - Gunakan Atribut 'SameSite'
Implementasi	Konfigurasi Pengaturan di PHP (php.ini) Dengan mengaktifkan: - session.cookie_samesite = Lax - session.cookie_httponly = 1 - session.cookie_secure = On  Session.cookie_samesite: dengan value 'Lax' cookie tidak akan dikirim pada permintaan cross-site biasa (misalnya, mengklik link), tetapi dikirim pada permintaan GET yang memulai navigasi ke URL.  Session.cookie_httponly: kuki hanya dapat diakses melalui protokol HTTP. Ini berarti bahwa kuki tidak dapat diakses

	oleh bahasa skrip, seperti JavaScript. Hal ini secara efektif dapat membantu mengurangi pencurian data melalui serangan XSS.
	`; http://php.net/session.cookie-secure`
	`session.cookie_secure = On`
	Session.cookie_secure: menentukan apakah kuki hanya boleh dikirim melalui koneksi aman, maka koneksi dikirim menggunakan HTTPS. Jika off, maka sesi berfungsi dengan koneksi HTTP dan HTTPS.

CVSS Skor	0.0

Temuan PT-008: Timestamp Disclosure – Unix (*Informational*)

Deskripsi	Timestamp ditampilkan oleh aplikasi/server web
Dampak	Dapat menjadi informasi bagi penyerang untuk mengatur strategi penyerangan.
Rekomendasi	Konfirmasi secara manual bahwa data timestamp tidak bersifat sensitif dan tidak memungkinkan pengumpulan data untuk mengungkap pola yang dapat dieksploitasi.
Implementasi	- Jika timestamp ditemukan maka sebaiknya dihilangkan dari respon HTTP untuk mencegah kebocoran informasi - Periksa pengaturan di server untuk memastikan tidak ada informasi sensitif yang tidak perlu ditampilkan dalam respon HTTP. 
CVSS Skor	0.0

Common Vulnerability Scoring System (CVSS)

No.	*Vulnerability*	Skor
1.	Cookie Without SameSite Attribute	3.0
2.	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	0.0
3.	X-Content-Type-Options Header Missing	3.0
4.	Information Disclosure - Suspicious Comments	0.0
5.	Loosely Scoped Cookie	0.0
6.	Timestamp Disclosure - Unix	0.0
7.	Cross Site Request Forgery (CSRF)	4.3
8.	Denial of Service (DoS)	6.6
Rata – rata skor		**3.4**

Hasil uji penetrasi menunjukkan bahwa keseluruhan kerentanan yang diidentifikasi pada Aplikasi Sikola V2.0 Unhas masuk dalam tingkat risiko *low*/rendah. Meskipun kerentanan tersebut tidak menimbulkan ancaman yang signifikan terhadap keamanan sistem, tetap perlu dilakukan perbaikan untuk mencegah potensi eksploitasi di masa depan. Upaya mitigasi yang tepat akan memastikan bahwa sistem tetap aman dan terlindungi dari ancaman yang mungkin muncul.