Available *online* at : http://jitce.fti.unand.ac.id/
JITCE (Journal of Information Technology and Computer Engineering)
| ISSN (Online) 2599-1663 |

Research Paper

Vulnerability Testing and Analysis on Websites and Web-Based Applications in the XYZ Faculty Environment Using Acunetix Vulnerability

Mifthahul Rahmi [1], Yuhandri Yunus [2], Sumijan [3]

[1,2,3] *Universitas Putra Indonesia (UPI) YPTK, Lubuk Begalung, Padang, 25145, Indonesia*

ARTICLE INFORMATION

Received: March 29th, 2023
Revised: November 18th, 2024
Available online: December 30th, 2024

KEYWORDS

Vulnerability, Website, Aplikasi, Fakultas XYZ, Acunetix Web Vulnerability Scanner

CORRESPONDENCE

Phone: -

E-mail: hellorahmi28@gmail.com

A B S T R A C T

The internet's continuous evolution has profoundly impacted society through the advancement of website technology and applications, reshaping contemporary ways of life. These digital platforms offer unrestricted information access, overcoming spatial and temporal limitations. In the realm of software development, Vulnerability Assessment is essential for producing high-quality products, as seemingly minor errors can create dangerous vulnerabilities that malicious actors may exploit to pilfer information from websites or applications. This study examines the security level of the Integrated website and application within the Faculty of Medicine, Universitas Andalas (Fakultas XYZ) environment, utilizing the Acunetix Web Vulnerability Scanner tool. The initial scan revealed a threat level of 3 (high) for the Fakultas XYZ website and level 2 (medium) for the Integrated application. Following a recapitulation process, several web alerts were identified for optimization, including Cross-Site Scripting (XSS), Blind SQL Injection, Application error message, HTML form without CSRF protection, Development configuration file, Directory listing, Error message on page, and User credentials sent in clear text. The optimization process involved source code review and enhancement to improve website features. A subsequent scan post-optimization demonstrated a reduction in threat levels for both the website and the UNAND FK Symphony application, with both achieving threat level 1 (low).

PENDAHULUAN

Perkembangan internet dan teknologi situs web serta aplikasi telah mengubah cara hidup masyarakat secara signifikan [1]. Situs web kini berfungsi sebagai pusat transaksi dan sumber informasi global yang dapat diakses publik secara bebas [2]. Perubahan teknologi yang cepat sering mengabaikan pengujian aplikasi, menciptakan celah keamanan yang dapat dieksploitasi. Karena itu, evaluasi kerentanan sangat penting dalam pengelolaan situs web dan aplikasi, meskipun sering diremehkan[3].

Celah keamanan dalam jaringan komputer adalah titik lemah yang bisa dieksploitasi penyerang, mengancam kerahasiaan, integritas, dan ketersediaan sistem serta aset [4]. Aplikasi berbasis website sering menjadi target hacker dan cracker karena pertumbuhannya yang pesat. Serangan dapat berupa XSS, CSRF, SQL injection, dan lainnya [5].

Vulnerability Assessment (VA) merupakan bagian dari penilaian risiko yang mencakup analisis risiko, pengembangan kebijakan, pelatihan, serta pengujian kerentanan dan *Penetration Testing*. VA melibatkan pemindaian sistem dan jaringan untuk mengidentifikasi kelemahan yang dapat dieksploitasi oleh penyerang. Oleh karena itu, sistem harus dilengkapi dengan kontrol akses untuk mengatasi kerentanan, seperti validasi input, pengaturan konfigurasi, dan penanganan pengecualian [6]. Hal ini sangat relevan bagi Fakultas XYZ, salah satu fakultas terbesar dan tertua di Universitas ABC, yang telah berkembang pesat dalam bidang Teknologi Informasi dan Komunikasi. Fakultas tersebut mengelola beberapa situs web dan aplikasi berbasis web untuk mendukung aktivitas Tri Dharma Perguruan Tinggi. Namun, seringkali pengelola dan staf IT fakultas tersebut mengabaikan pentingnya keamanan pada situs dan aplikasi yang mereka kelola [7].

Oleh karena itu, penelitian tentang Pengujian dan Analisis Vulnerability pada Website dan Aplikasi Berbasis Web di Fakultas XYZ menggunakan *Acunetix Vulnerability* sangat diperlukan. Penelitian ini bertujuan untuk mengidentifikasi dan mendokumentasikan celah keamanan dalam website dan aplikasi Fakultas XYZ. Selain itu, penelitian ini akan menganalisis tingkat kerentanan yang ditemukan untuk menilai dampaknya terhadap integritas, kerahasiaan, dan ketersediaan aset fakultas. Penelitian ini juga akan memberikan rekomendasi perbaikan untuk meningkatkan keamanan sistem Terakhir, penelitian ini akan mengatasi beberapa celah keamanan yang teridentifikasi dalam

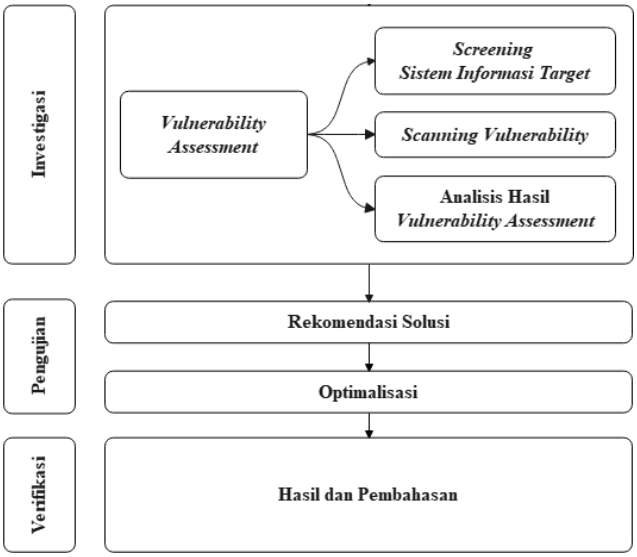
https://doi.org/10.25077/jitce.8.2.83-96.2024

assessment untuk memastikan sistem tetap terlindungi dari ancaman di masa mendatang. Dengan tujuan ini, penelitian diharapkan dapat memberikan kontribusi signifikan terhadap keamanan website dan aplikasi di Fakultas XYZ.

Dalam konteks keamanan siber, pemahaman mengenai kerentanan sangat penting untuk melindungi aset digital. Kerentanan merupakan kelemahan yang mengancam integritas, kerahasiaan, dan ketersediaan aset [8]. Beberapa celah yang sering dieksploitasi oleh hacker meliputi *HTTP Header, HTML Injection, metode GET, Cookie, Shell injection*, dan perintah *include* [9]. *Vulnerability Assessment (VA*) merupakan analisis keamanan komprehensif yang mencakup dokumen keamanan, hasil pemindaian jaringan, konfigurasi sistem, kesadaran pengguna, dan keamanan fisik untuk mengidentifikasi kerentanan penting [10]. Audit keamanan sistem pada website atau aplikasi berbasis web penting untuk mengurangi risiko kehilangan data dan mencegah kesalahan pada sistem berbasis teknologi informasi [11]. Hasil eksploitasi dalam proses VA akan mengklasifikasikan celah keamanan berdasarkan tingkat kerentanannya dan selanjutnya akan ditindaklanjuti pada tahap optimalisasi untuk perbaikan celah tersebut [12].

METODE

Metode penelitian ini disusun secara sistematis dalam bentuk kerangka kerja, sehingga hasil yang diperoleh dapat digunakan sebagai pedoman analisis untuk mencapai tujuan yang telah ditetapkan. Secara umum, kerangka kerja penelitian ini terdiri dari beberapa tahap, yaitu investigasi, pengujian, dan verifikasi. Setiap tahap tersebut mencakup sub-tahapan yang lebih rinci untuk memastikan penelitian dilakukan secara terstruktur dan sistematis. Kerangka kerja penelitian ini ditampilkan pada Gambar 1.



Gambar 1. Kerangka Kerja Penelitian

Gambar 1 menggambarkan bahwa proses *vulnerability assessment* dimulai dengan tahapan *screening* sistem informasi target, yang bertujuan untuk memahami lingkungan teknologi yang digunakan. Pemahaman ini akan membantu dalam merumuskan rekomendasi solusi dan optimalisasi untuk menangani *web alert* yang terdeteksi. Pada tahap *Scanning Vulnerability*, digunakan *Acunetix Vulnerability Web Scanner* untuk mengidentifikasi celah keamanan dalam sistem informasi target. Hasil *vulnerability assessment* oleh *Acunetix* secara otomatis akan dikelompokkan berdasarkan *level severity*, dan analisis lebih lanjut akan difokuskan pada *web alert* dengan *level severity medium* hingga *high*.

Pada tahap pengujian, serangkaian *penetration testing* akan dilakukan terhadap celah keamanan dengan *level severity medium* hingga *high* untuk mengidentifikasi, mengeksploitasi, dan mengevaluasi kerentanannya. Berdasarkan hasil pengujian tersebut, rekomendasi solusi akan diberikan. Penelitian kemudian dilanjutkan ke tahap optimalisasi guna meningkatkan keamanan sistem.

Selanjutnya, pada tahap akhir, verifikasi akan dilakukan untuk menilai sejauh mana optimalisasi yang dilakukan berdampak positif pada sistem target. Verifikasi ini dilakukan dengan melakukan Scanning Vulnerability pasca-optimalisasi untuk memastikan bahwa web alert yang terdeteksi sebelum optimalisasi telah berhasil diatasi.

Analisis Data

Sistem informasi target yang digunakan dalam penelitian ini adalah Website resmi Fakultas XYZ dan Aplikasi Terintegrasi Fakultas XYZ . Pemilihan Website dan aplikasi berbasis Website di lingkungan Fakultas XYZ untuk mewakili keanekaragaman system teknologi informasi yang ada di lingkungan Fakultas XYZ. *Website dan Aplikasi* Terintegrasi Fakultas XYZ dibangun dengan *detail* sistem informasi sebagaimana yang ditampilkan pada Tabel 1 dan Tabel 2.

Tabel 1. Sistem Informasi *Website* Fakultas XYZ

SETTING	VALUE
PHP Built On	Linux xyz.ac.id 4.9.0-15-amd64 #1 SMP Debian 4.9.258-1 (2021-03-08) x86_64
Database Type	mysql
Database Version	5.5.5-10.1.48-MariaDB-0+deb9u2
Database Collation	latin1_swedish_ci
Database Connection Collation	utf8mb4_general_ci
PHP Version	7.4.4
Web Server	Apache/2.4.25
Web Server to PHP Interface	cgi-fcgi
Joomla! Version	Joomla! 3.9.14 Stable [Amani] 17-December-2019 15:00 GMT
Joomla! Platform Version	Joomla Platform 13.1.0 Stable [Curiosity] 24-Apr-2013 00:00 GMT
User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36

https://doi.org/10.25077/jitce.8.2.83-96.2024

Tabel 2. Sistem Informasi Aplikasi Terintegrasi Fakultas XYZ

SETTING	VALUE
Analytics	Google Analytics GA4
Web frameworks	CodeIgniter 2+
Miscellaneous	Open Graph
Web servers	Apache HTTP Server
Programming languages	PHP
CDN	Cloudflare
cdnjs	JavaScript libraries
jQuery	V3.2.1
UI frameworks	Bootstrap 4.3.1

Proses *Vulnerability Assessment* dalam penelitian ini dilakukan menggunakan *Acunetix Vulnerability Web Scanner* untuk menguji kerentanan pada website dan aplikasi yang menjadi objek penelitian. *Scanning* menggunakan *Acunetix* dilakukan dalam dua kali iterasi yaitu sebelum dan setelah optimaliasi, dengan masing-masing iterasi terdiri dari dua kali proses *scanning*. Celah keamanan dengan *level severity medium* hingga *high* yang ditemukan pada sistem informasi target akan dijadikan fokus untuk proses optimalisasi.

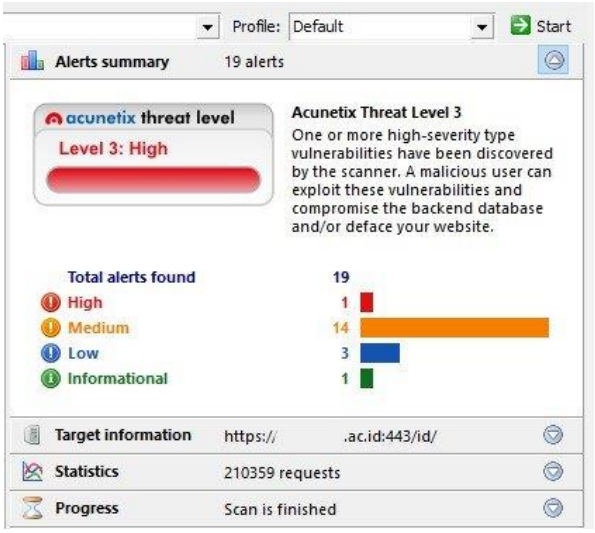
Scanning Iterasi 1 pada Website Fakultas XYZ

Pada website Fakultas XYZ, proses *scanning* iterasi 1 (sebelum optimaliasi) dilakukan sebanyak dua kali, dengan rincian hasil yang ditampilkan pada Tabel 3.

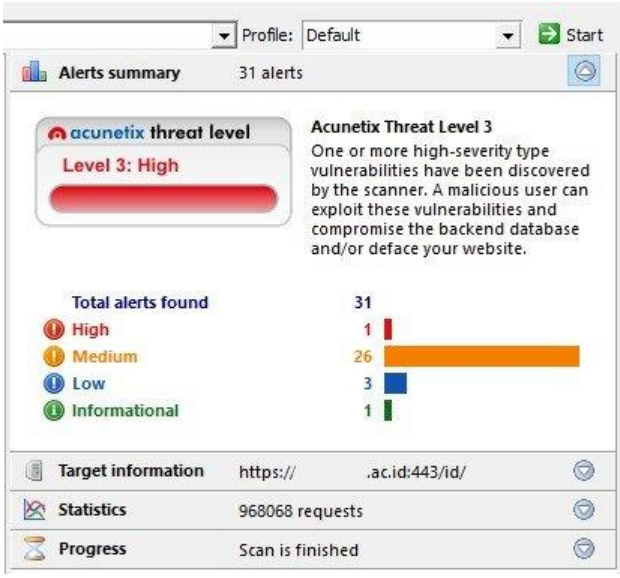
Tabel 3. *Scanning* iterasi 1 *Website* Fakultas XYZ.

No.	Tanggal Pengujian	Durasi Pengujian	Total *Web Alert*
1.	03 Maret 2023	2 Jam 57 Menit	19
2.	05 Maret 2023	10 Jam 30 Menit	31

Berdasarkan hasil *scanning* iterasi pertama, baik pada *scanning* pertama maupun kedua, *website* Fakultas XYZ menunjukkan adanya celah kerentanan yang tinggi. Hal ini terindikasi dari hasil pemindaian *Acunetix*, yang menunjukkan bahwa *threat level website* tersebut berada pada *level* 3 *(high)*, hal ini dapat dilihat pada Gambar 2 dan Gambar 3.



Gambar 2. Hasil *Scanning* 1 (Iterasi 1) *Website* Fakultas XYZ



Gambar 3. Hasil *Scanning* 2 (Iterasi 1) *Website* Fakultas XYZ

Akumulasi sebaran *Web Alert* yang terdeteksi pada *website* Fakultas XYZ selama *scanning* iterasi 1 ditunjukkan dalam Tabel 4.

Tabel 4. Sebaran *Web Alert Website* Fakultas XYZ pada *scanning* Iterasi 1

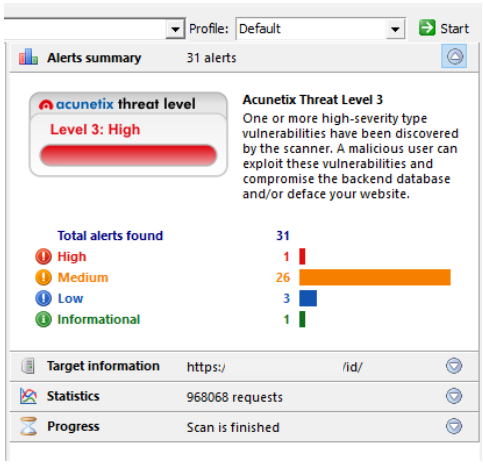
No.	Thread Level	Jenis Web Alert	Level Severity	Jumlah Alert
1.	3 (High)	Blind SQL Injection	High	1
2.		Cross site scripting	High	1
3.		Application error message	Medium	23
4.		HTML form without CSRF protection	Medium	3
5.		Clickjacking: X-Frame-Options header missing	Low	1
6.		Cookie without Secure flag set	Low	1
7.		File upload	Low	1
8.		Broken links	Low	1

Scanning Iterasi 1 pada Aplikasi Terintegrasi Fakultas XYZ

Pada Aplikasi Terintegrasi Fakultas XYZ proses *scanning* iterasi 1 (sebelum optimaliasi) dilakukan sebanyak dua kali sebagaimana ditampilkan pada Tabel 5. Berdasarkan hasil *scanning* iterasi 1, baik pada *scanning* pertama maupun kedua, Aplikasi Terintegrasi Fakultas XYZ menggunakan *Acunetix*, menunjukkan bahwa *threat level* aplikasi tersebut berada pada *level 2 (medium)* namun tidak ada *web alert* dengan tingkat kerentanan tinggi yang terdeteksi, seperti yang ditampilkan dalam Gambar 5 dan Gambar 6.

Tabel 5. *Scanning* iterasi 1 Aplikasi Terintegrasi Fakultas XYZ

No.	Tanggal Pengujian	Durasi Pengujian	Total *Web Alert*
1.	05 Maret 2023	17 Menit 53 Detik	924
2.	06 Maret 2023	18 Menit 58 Detik	920



Gambar 5. Hasil Scanning 2 (Iterasi 1) Terintegrasi Fakultas

Gambar 6. Hasil *Scanning* 2 (Iterasi 1) Terintegrasi Fakultas XYZ

Berdasarkan hasil *scanning* tersebut, didapatkan informasi bahwa hasil *scanning* menampilkan jumlah *web alert* dan tingkat potensi serangan yang berbeda. Rincian dari hasil *scanning* tersebut dapat dilihat pada Tabel 6.

Berdasarkan hasil *scanning* sebelumnya, kedua website tersebut ditemukan memiliki jenis-jenis *alert* atau celah yang serupa. Untuk meningkatkan efisiensi penelitian, berikut adalah pengelompokan jenis-jenis *web alert* dengan tingkat kerentanan dari *high* hingga *medium* yang terdeteksi pada kedua sistem tersebut, sebagaimana dijelaskan dalam Tabel 7.

Tabel 6. Sebaran Web Alert Aplikasi Terintegrasi Fakultas XYZ pada *Scanning* 2 (Iterasi 1)

No.	Thread Level	Jenis Web Alert	Level Severity	Jumlah Alert
1.		Application error message	Medium	1
2.		Development configuration file	Medium	1
3.		Directory listing	Medium	6
4.		Error message on page	Medium	32
5.		User credentials are sent in clear text	Medium	1
6.		Clickjacking: X-Frame-Options header missing	Low	1
7.		Documentation file	Low	1
8.		Insecure response with wildcard	Low	2
9.	2 (Medium)	Login page password-guessing attack	Low	1
10.		Possible sensitive directories	Low	13
11.		Possible sensitive files	Low	13
12.		Broken links	Informational	71
13.		Email address found	Informational	85
14.		Password type input with auto-complete enabled	Informational	1
15.		Possible internal IP address disclosure	Informational	208
16.		Possible server path disclosure (Unix)	Informational	404
17.		Possible username or password disclosure	Informational	79

https://doi.org/10.25077/jitce.8.2.83-96.2024

Tabel 7. Rekapitulasi *Web Alert* yang Akan Dioptimalisasi

No.	Jenis Web Alert	Level Severity	Terdeteksi pada	
			Website Fakultas XYZ	Aplikasi Terintegrasi Fakultas XYZ
1.	Cross Site Scripting (XSS)	High	√	-
2.	Blind SQL Injection	High	√	-
3.	Application error message	Medium	√	√
4.	HTML form without CSRF protection	Medium	√	-
5.	Development configuration file	Medium	-	√
6.	Directory listing	Medium	-	√
7.	Error message on page	Medium	-	√
8.	User credentials are sent in clear text	Medium	-	√

Penelitian ini berfokus untuk melakukan analisa *penetration testing* dan optimalisasi pada celah keamanan yang berada pada *level medium* hingga *level high*. Memilih untuk fokus pada celah keamanan dengan *level high* dan *medium* dari hasil pemindaian *Acunetix* didasarkan pada beberapa alasan kunci:

1. Risiko Terhadap Keamanan : celah keamanan dengan *level high* dan *medium* mencerminkan kerentanan yang memiliki potensi risiko yang signifikan terhadap keamanan aplikasi dan data. Mengatasi celah-celah ini lebih mendesak untuk mencegah serangan yang dapat menyebabkan kerugian besar.
2. Dampak Potensial : kerentanan pada *level high* dapat memungkinkan penyerang untuk melakukan tindakan berbahaya, seperti pencurian data, pengambilalihan akun, atau bahkan pengendalian penuh atas sistem. *Level medium* juga menunjukkan kerentanan yang bisa dieksploitasi dalam

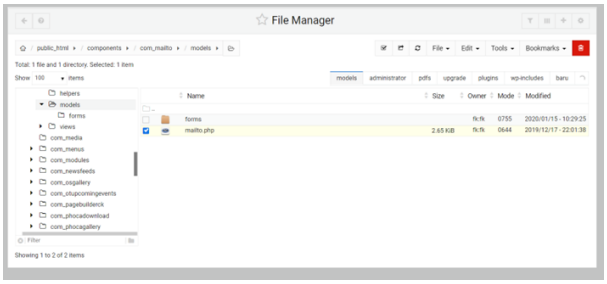
Dengan demikian, memprioritaskan celah keamana *level high* dan *medium* merupakan langkah strategis dalam meningkatkan keamanan sistem dan melindungi aset informasi. Hasil *vulnerability assessment* ini selanjutnya akan diproses pada tahap pengujian untuk dilakukan identifikasi lebih lanjut sehubungan dengan *attack details*, jenis *web alert*, teknik optimalisasi yang akan dilakukan hingga didapatkan kesimpulan atau rekomendasi untuk pengelolaan kedua sistem bagi pengembang aplikasi sistem informasi di institusi tersebut.

HASIL DAN PEMBAHASAN

Cross Site Scripting (XSS)

Berdasarkan hasil *vulnerability assessment* yang dilakukan oleh *Acunetix*, *item* yang terdampak *vulnerability* ini adalah *Website* Fakultas XYZ daoat dilihat pada Gambar 7.

Cross Site Scripting (XSS) merupakan sebuah celah keamanan yang memanfaatkan kerentanan untuk mencuri data, mengendalikan sesi pengguna, menjalankan kode berbahaya, atau mendukung serangan *phishing*. Penyerang dapat melakukan *XSS* dengan menyisipkan kode *HTML* atau skrip ke dalam situs web, sehingga situs tersebut tampak seolah-olah melakukan serangan sendiri.

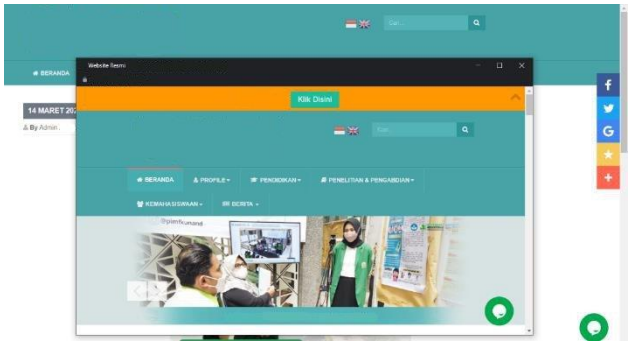


Gambar 7. Lokasi Item Terdampak *Cross Site Scripting (XSS)* pada *Web Server*

Dalam penelitian ini, proses *penetration testing* dilakukan untuk menguji celah kerentanan dengan cara menginputkan kode sumber *JavaScript* yang mengandung metakarakter sensitif ke dalam formulir berbagi artikel melalui email yang ada pada website Fakultas XYZ seperti yang ditampilkan pada Gambar 8.

Berdasarkan hasil *penetration testing*, meskipun skrip sensitif telah dimasukkan ke dalam formulir, sistem tetap memberikan respons positif, yang berarti konten berhasil dikirim. Hal ini ditandai dengan halaman formulir yang langsung dialihkan ke URL https://fakultasxyz.ac.id/id/ melalui halaman *pop-up*, seperti yang ditunjukkan pada Gambar 9.

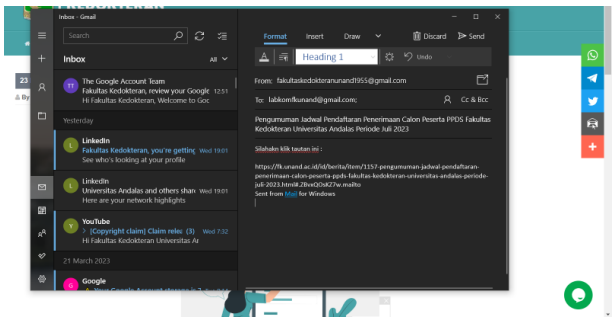
Gambar 8. *Penetration Testing Cross Site Scripting (XSS)*



Gambar 9. Tampilan Halaman Setelah *Penetration Testing*

Dampak serangan *XSS* pada *website* Fakultas XYZ meliputi pencurian data sensitif, seperti kredensial dan informasi pribadi, Serangan ini juga dapat menyebarkan *malware* kepada pengunjung lainnya, menciptakan halaman *phishing* yang menipu pengguna untuk memberikan informasi pribadi, dan merusak reputasi Fakultas XYZ, sehingga mengurangi kepercayaan publik. Selain itu, pengguna dapat dialihkan ke situs berbahaya, dan serangan ini dapat mengakibatkan gangguan layanan, mempengaruhi akses dan fungsionalitas situs *web*.

Proses optimalisasi *Cross Site Scripting (XSS)* pada *Website* Fakultas XYZ akan dilakukan dengan mengganti fitur *plugin email* yaitu menggunakan *plugin addthis*, dimana *plugin* ini nantinya akan melakukan *redirect* pada aplikasi *email* yang ada pada sistem operasi baik pada windows maupun macOS sehingga pengiriman *email* hanya dapat dilakukan jika *user* telah melakukan *login* pada *email* seperti pada Gambar 10.

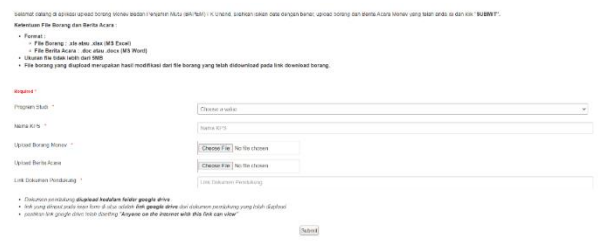


Gambar 10. Hasil Optimalisasi *Cross Site Scripting (XSS)*

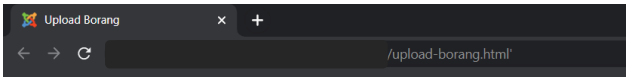
Blind Sql Injection

Blind SQL injection adalah teknik yang mengeksploitasi celah keamanan pada lapisan database suatu aplikasi akibat input yang tidak difilter dengan benar. Penyerang memodifikasi perintah SQL melalui formulir input aplikasi, memungkinkan mereka mengirimkan sintaks ke *database* dan melihat data yang seharusnya tidak mereka akses, termasuk informasi pengguna lain. Dalam beberapa kasus, penyerang dapat memodifikasi atau menghapus data, mengubah konten atau fungsi aplikasi. Oleh karena itu, sangat penting bagi pengembang untuk memastikan bahwa semua input yang diterima telah difilter dengan baik untuk mencegah serangan blind SQL injection.

Berdasarkan hasil eksploitasi celah yang dilakukan oleh *Acunetix*, sistem yang terdampak *vulnerability Blind SQL Injextion* adalah *website* Fakultas XYZ dengan rincian *item* yang rentan akan serangan sebagaimana ditampilkan pada Gambar 11.



Gambar 11. Halaman Terindikasi Celah *Blind SQL Injection Penetration testing* untuk menguji kerentanan *website* Fakultas XYZ terhadap serangan *Blind SQL injection* dilakukan dengan injeksi parameter *stirng (')* setelah akhir link halaman yang terdampak sebagaimana yang ditampilkan pada Gambar 12.



Gambar 11. *Penetration Testing Blind SQL Injection*

Optimalisasi celah *SQL Injection* adalah dengan menerapkan filter metakarakter hingga mengimplementasikan *input* validasi pada *form* yang ada pada sistem.

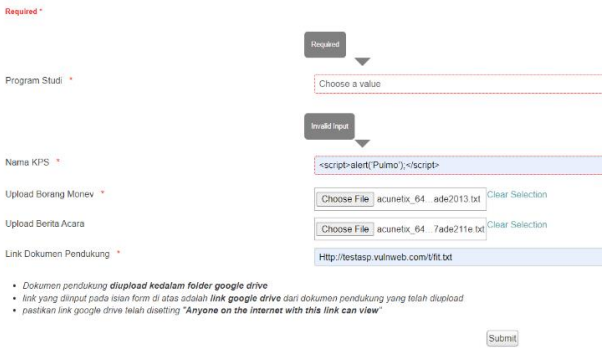
```
118         // Declare a regular expression
119         $regex = "/<b>(.*)<\/b>/U";
```

Gambar 12. Implementasi *Script* untuk mengatasi *Blind SQL Injection*

Selain dengan melakukan review *source code*, untuk mencegah celah *Blind SQL Injection* ini dilakukan juga validasi input method yang ada pada konfigurasi form sebagimana yang terlihat pada Gambar 13.

Gambar 13. Implementasi Konfigurasi Validasi *Input Method*

Optimalisasi pada celah *Blind SQL Injection* yang telah diterapkan membuat formulir otomatis menolak *input* dengan karakter sensitif yang berpotensi memicu serangan *SQL Injection*, seperti ditunjukkan pada Gambar 14.



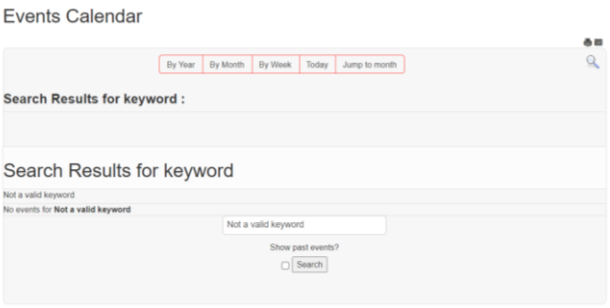
Gambar 14. Hasil Optimalisasi *Alert Blind SQL Injection*

Application Error Message

Hasil *scanning* menggunakan *Acunetix Vulnerability Web Scanner* menunjukkan bahwa terdapat 16 *alert* pada *website* dan 1 *alert* pada aplikasi terintegrasi Fakultas XYZ terkait kerentanan

https://doi.org/10.25077/jitce.8.2.83-96.2024

ini. Pada website Fakultas XYZ, item yang terdampak tersebar di beberapa halaman, seperti terlihat pada Gambar 15.



Gambar 15. Halaman *Website* yang Terdampak *Vulnerability Application Error Message*

Dampak dari *Vulnerability Application Error Message* pada website dan aplikasi terintegrasi Fakultas XYZ adalah munculnya kerentanan keamanan saat pesan *error* mengandung informasi sensitif. Pesan ini biasanya memberi tahu pengguna tentang kesalahan, tetapi jika mencakup data seperti alamat IP, rincian koneksi *database*, atau kata sandi, penyerang dapat memanfaatkannya untuk meretas sistem. Oleh karena itu, optimalisasi diperlukan untuk mencegah kerugian bagi institusi.
Penetration testing untuk menguji celah *Application Error Message* pada Aplikasi Terintegrasi Fakultas XYZ dilakukan oleh *Acunetix* dengan injeksi *script request data* seperti pada Gambar 16.

Gambar 16. *Script Penetration Testing Application Error Message* pada *Aplikasi* Terintegrasi Fakultas XYZ

Hasil dari implementasi *script* tersebut, aplikasi Terintegrasi menampilkan pesan *error* sebagaimana yang ditampilkan pada Gambar 17.

Gambar 17. Hasil *Penetration Testing Application Error Message* oleh A*cunetix* pada *Aplikasi* Terintegrasi Fakultas XYZ

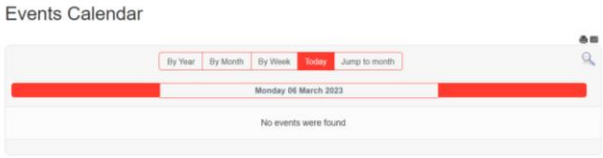
Optmalisasi dilakukan dengan memodifikasi *handling multiple environments* yang terdapat pada *file index.php* yang terdapat pada *web server d*imana *script ENVIRONMENT* yang awalnya diinisalisasikan dengan *'development'* menjadi *'production'* agar pesan *error* yang muncul tidak menampilkan informasi terkait sistem sebagaimana yang ditampilkan pada Gambar 17.

```
56    define('ENVIRONMENT', isset($_SERVER['CI_ENV']) ? $_SERVER['CI_ENV'] : 'production');
```

Gambar 17. *Script Filter* Pada Aplikasi Terintegrasi Fakultas XYZ

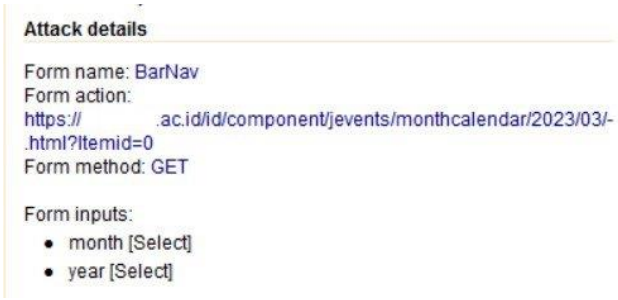
HTML Form Without CSRF Protection

Berdasarkan hasil pemindaian dengan *Acunetix Vulnerability Web Scanner*, kerentanan *HTML Form Without CSRF Protection* ditemukan pada beberapa halaman di website Fakultas XYZ, seperti ditunjukkan pada Gambar 18.



Gambar 18. Halaman Website yang Terdampak *Vulnerability HTML Form Without CSRF Protection*

Cross-site request forgery (CSRF) adalah teknik serangan di mana penyerang mengirim permintaan palsu dari situs tepercaya yang sedang diakses korban, bertujuan untuk melakukan tindakan tidak sah, seperti mengubah data atau mengirim informasi sensitif. Jika form HTML di aplikasi web tidak memiliki perlindungan CSRF, penyerang dapat mengirim permintaan palsu ke *server* yang akan dieksekusi tanpa verifikasi, memungkinkan tindakan tidak diinginkan, seperti perubahan data atau pengiriman informasi sensitif.
Proses *penetration testing* yang dilakukan oleh *Acunetix* untuk membuktikan kerentanan yang disebabkan oleh celah *HTML Form Without CSRF Protection* melibatkan serangkaian uji coba, sebagaimana diperlihatkan pada Gambar 19.



Gambar 19. *Attack Details HTML Form Without CSRF Protection*

Berdasarkan serangan tersebut, maka respon yang diberikan oleh *website* adalah seperti pada Gambar 20.

Gambar 20. Hasil *Penteration Testing HTML Form Without CSRF Protection*

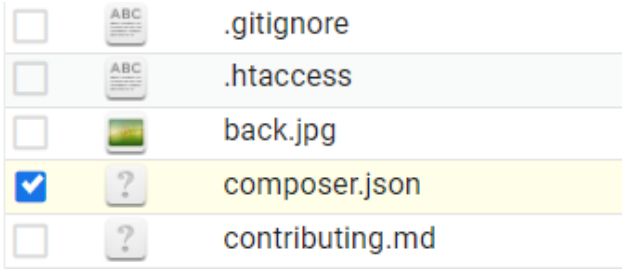
Berdasarkan item yang terdampak oleh *alert* ini, optimalisasi tidak hanya memerlukan *review* kode sumber, tetapi juga dapat dilakukan melalui pemeriksaan komponen dan konfigurasi system, solusi untuk optimalisasi kerentanan ini disajikan pada Tabel 8.

Tabel 8. Rekomendasi Solusi Optimalisasi *HTML Form Without CSRF Protection*

No.	Item alert	Solusi
1	/id/component/jevents/eventsbyday/2023/3/6/-.html	1. Melakukan konfigurasi ulang pada *extension* 2. Review orisinalitas *extension* 3. Jika *extension* tidak lagi dikembangkan secara resmi, direkomendasikan untuk di-*uninstal*
2	/id/component/jevents/search_form/-.html (070379ee25b7b924bbe1090708b06703)	
3	/id/component/jevents/search_form/-.html (72af7383d92a7c2fd0be2df9e914c722)	

Development Configuration File

Hasil *scanning* menunjukkan bahwa *alert* ini ditemukan pada Aplikasi Terintegrasi Fakultas XYZ, dengan item yang terdampak yaitu *file composer.json* pada *web server*. Pada baris kedua file tersebut terdapat deskripsi yang menjelaskan teknologi yang digunakan dalam sistem, sebagaimana terlihat pada Gambar 21 dan Gambar 22.

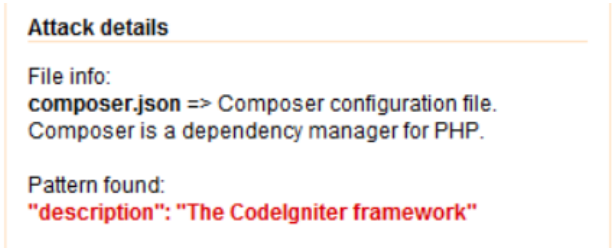


Gambar 21. File Terdampak Celah *Development Configuration File*

Gambar 22. *Baris Code Development Configuration File* pada *composer.json*

Configuration File mengandung informasi yang sangat sensitif seperti *credential database*, kunci enkripsi, kunci rahasia aplikasi, dan konfigurasi khusus lainnya yang tidak seharusnya diketahui oleh pihak yang tidak berwenang. Jika *file* konfigurasi pengembangan tidak diatur dengan benar atau tidak dihapus setelah selesai digunakan, maka informasi sensitif pada *file* tersebut dapat diakses oleh penyerang yang dapat mengakses lingkungan produksi aplikasi. Penyerang akan memanfaatkan informasi tersebut untuk melakukan tindakan yang tidak diinginkan pada aplikasi, seperti mencuri atau mengubah data, atau bahkan mengambil alih kontrol sistem.

Penetration testing yang dilakukan oleh *Acunetix* untuk menguji kerentanan pada *file* konfigurasi melibatkan investigasi menyeluruh terhadap *file composer.json* yang terdapat dalam aplikasi, sebagaimana ditunjukkan pada Gambar 23.



Gambar 23. Hasil *Penetration Testing Development Configuration File* oleh *Acunetix*

Langkah optimalisasi yang dapat dilakukan untuk mengantisipasi celah ini adalah dengan menyembunyikan informasi terkait deskripsi *framework* tersebut. Hal ini dapat dilakukan dengan mengkonfersikan *source code* tersebut menjadi *comment* Proses konfersi *script program* menjadi *comment* ini dilakukan pada *file composer.json* sebagaimana yang ditampilkan pada Gambar 24.

Gambar 24. Optimalisasi *File composer.json*

Optimalisasi yang dilakukan pada file *composer.json* ini akan menonaktifkan *script* yang menginisiasikan terkait teknologi yang digunakan pada aplikasi Terintegrasi. Modifikasi *script* ini dilakukan dengan menjadikan *line* tersebut sebagai *comment* saja sehinggga tidak akan ikut dieksekusi ketika program dijalankan.

Directory Listing

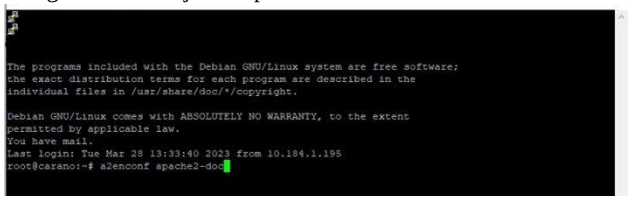
Berdasarkan hasil eksploitasi celah yang dilakukan oleh *Acunetix Vulnerability Web Scanner*, sistem yang terdampak *vulnerability* ini adalah Aplikasi Terintegrasi Fakultas XYZ dimana terdapat beberapa direktori yang terekspos.

Proses identifikasi dan penetration testing celah ini dilakukan oleh Acunetix dimana dari hasil scanning terdapat beberapa direktori yang terekspos, seperti yang terdapat pada Gambar 25.

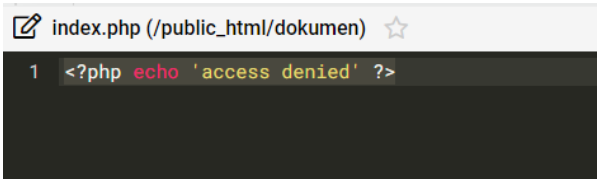
https://doi.org/10.25077/jitce.8.2.83-96.2024

Gambar 25. Direktori Dokumen yang Dapat Diakses Secara Langsung

Optimalisasi yang dapat dilakukan untuk mencegah serangan yang disebabkan oleh *web alert* directory listing antara lain dengan melakukan konfigurasi pada file *apache2* di *Virtualmin*. Hal ini bisa dilakukan dengan menambahkan baris skrip tertentu atau dengan menyertakan file *index.php* di setiap folder direktori yang berisi kode sumber `<?php echo 'access denied'; ?>`, sebagaimana ditunjukkan pada Gambar. 26 dan Gambar 27.

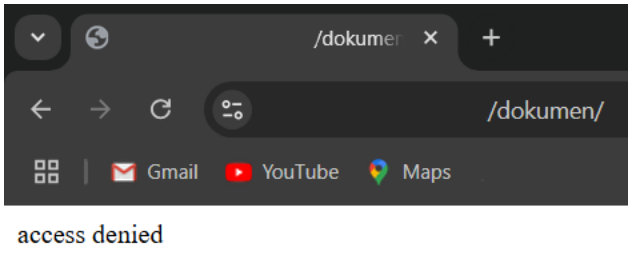


Gambar 26. Konfigurasi *File Apache*



Gambar 26. *Source Code File index.php*

File index.php ini akan ditambahkan pada setiap *folder* direktori yang terdeteksi sebagai celah oleh *Acunetix*.



Gambar 27. Hasil Optimalisasi *Web Alert Directory Listing*

Gambar 27 menunjukkan hasil implementasi optimalisasi, yang dibuktikan dengan tidak dapat diaksesnya folder direktori yang terekspos. Saat folder tersebut diakses melalui *browser*, statusnya menjadi *access denied*.

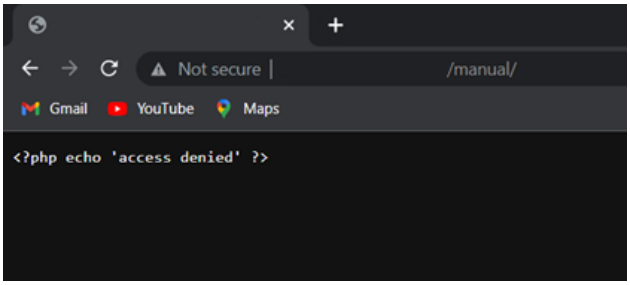
Error Message On Page

Error Message On Page adalah sebuah celah kemanaan pada aplikasi berbasis *web* yang terjadi ketika aplikasi mengalami sebuah *error*, maka akan menampilkan sebuah pesan yang mengandung informasi *debug* yang seharusnya hanya dapat dilihat oleh pengembang namun ditampilkan juga pada halaman pengguna.

Penetration testing celah keamanan *Error Message On Page* ini dilakukan oleh *Acunetix Vulnerability Web Scanner* dengan beberapa detail serangan sehingga menampilkan pesan *error* seperti pada Gambar 28.

Gambar 27. *Error Message On Page* pada Aplikasi Terintegrasi

Optimalisasi celah keamanan terkait *Alert Error Message On Page* dapat dilakukan dengan memodifikasi *file controller index* agar diarahkan ke halaman *access denied*. Selain itu, proses optimalisasi juga dapat dilakukan dengan mencegah munculnya pesan *error*, yaitu dengan mengatur fungsi *error_reporting()* menjadi 0 atau mengubah pengaturan konfigurasi *environment* menjadi *production*, sehingga *URL* halaman tersebut tidak dapat diakses Implementasi optimalisasi tersebut dapat dilihat pada Gambar 28.



Gambar 28. Hasil Optimalisasi *Alert Error Message On Page*

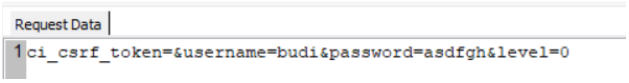
User Credentials Are Sent In Clear Text

Celah keamanan *User Credentials Are Sent In Clear Text* terjadi ketika informasi penting seperti *username* dan *password* pengguna dikirimkan dalam bentuk teks biasa *(plaint text)* tanpa dienkripsi. Jika penyerang yang memiliki akses ke jaringan atau aliran data maka penyerang dapat dengan mudah membaca kredensial tersebut dan mengakses akun pengguna.

Penetration testing untuk melakukan pengujian celah keamanan *User Credentials Are Sent In Clear Text* dilakukan oleh *Acunetix* dengan *attack details* seperti pada Gambar 29.

Gambar 29. *Attack Details Web alert User Credentials Are Sent In Clear Text*

Hasil dari serangan ini menunjukkan bahwa data kredensial yang diinput dan dikirimkan oleh pengguna sepenuhnya ditransmisikan dalam bentuk teks biasa (plain text), seperti yang terlihat pada Gambar 30.



Gambar 30. Hasil Pengujian *Web alert User Credentials Are Sent In Clear Text*

Pada Gambar 30 dapat dilihat bahwa data akun yang diinputkan *user* dikirim apa adanya. Hal ini merupakan kondisi yang rentan kareena pihak ketiga dapat membaca kredensial pengguna dengan cara menangkap koneksi HTTP yang tidak dienkripsi tersebut.

Optimalisasi *web alert User Credentials Are Sent In Clear Text* dilakukan dengan menerapkan *function password hash*. *Password Hash* adalah salah satu fungsi pada *php* untuk melakukan *(one way hashing)* atau *hashing* satu arah untuk merubah *plain text* menjadi suatu kode acak atau kode enkripsi. Implementasi password hashing ini dapat dilakukan dengan menambahkam script seperti pada Gambar 31.

```
$user       = $this->input->post('user');
$pass       = md5($this->input->post('pass'));
$group      = $this->input->post('group');
```

Gambar 31. Implementasi *Script Code Password Hashing*

Script Code ini akan diimplementasikan pada *file M_auth.php* dan *data.php* dalam aplikasi Terintegrasi untuk mengenkripsi kata sandi pengguna saat melakukan *login*. Untuk menilai keberhasilan proses optimalisasi terhadap celah keamanan ini, dapat merujuk pada hasil *scanning* iterasi kedua oleh *Acunetix*, guna menentukan apakah celah keamanan ini masih dapat dieksploitasi atau tidak.

HASIL DAN PEMBAHASAN

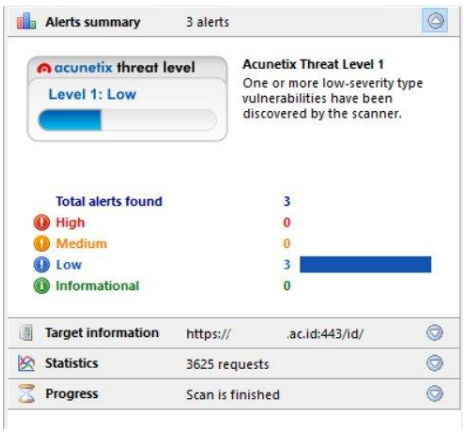
Setelah proses optimalisasi diterapkan pada celah keamanan, dilakukan *scanning* iterasi kedua untuk memverifikasi keberhasilan optimalisasi dalam mengatasi celah keamanan yang terdeteksi pada *scanning* iterasi pertama. Langkah ini bertujuan memastikan bahwa sistem telah beroperasi dengan aman.

Pada *Website* Fakultas XYZ proses *scanning* iterasi kedua dilakukan sebanyak 2 kali dengan rincian pada Tabel 9.

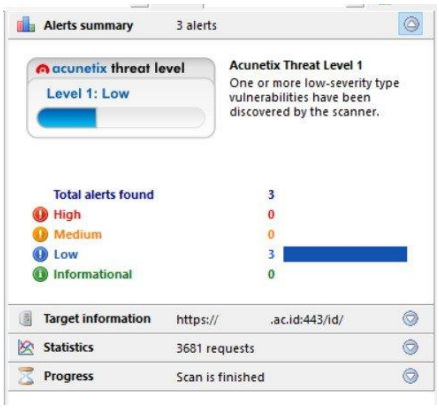
Tabel 9. *Website* Fakultas XYZ *Scanning* Iterasi 2

No.	Tanggal Pengujian	Durasi Pengujian	Total *Web Alert*
1.	22 Maret 2023	1 Jam 55 Menit	3
2.	22 Maret 2023	2 Jam 31 Menit	3
	Thread Level		***Level 1 (Low)***

Hasil *scanning* iterasi kedua menunjukkan bahwa total celah keamanan pada website Fakultas XYZ menurun dibandingkan dengan *scanning* iterasi pertama. Selain itu, celah keamanan pada *website* Fakultas XYZ juga mengalami perubahan; jika pada iterasi pertama *threat level* berada di kategori *high*, pada iterasi kedua telah turun menjadi *level 1: low*, sebagaimana ditunjukkan pada Gambar 32 dan Gambar 33.



Gambar 32. Hasil *Scanning* 1 (Iterasi 2) *Website* Fakultas XYZ



Gambar 33. Hasil *Scanning* 2 (Iterasi 2) *Website* Fakultas XYZ

Gambar 32 dan Gambar 33 menunjukkan hasil *scanning* pertama dan kedua pada iterasi kedua menggunakan *Acunetix* pada website Fakultas XYZ. Berdasarkan hasil tersebut, terlihat adanya penurunan jumlah web alert dengan rincian informasi yang disajikan pada Tabel 10. Perbandingan data *web alert* pada website Fakultas XYZ setelah dilakukan optimalisasi ditampilkan pada Tabel 11.

https://doi.org/10.25077/jitce.8.2.83-96.2024

Table 10. Sebaran Web Alert Website Fakultas XYZ pada Scanning (Iterasi 2)

No.	Thread Level	Jenis Alert	Level Alert	Jumlah Web Alert
1.		Clickjacking: X-Frame-Options header missing	Low	1
2.	2 (Medium)	Cookie without Secure flag set	Low	1
3.		File upload	Low	1

Tabel 11. Perbandingan Data *Web Alert* Wesbite Fakultas XYZ Setelah Dioptimalisasi

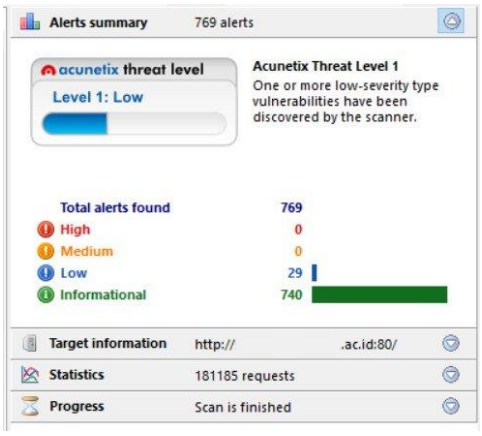
No.	Web Alert	Jumlah Case	
		Sebelum Optimalisasi	Setelah Optimalisasi
1.	*Cross Site Scripting (XSS)*	1	0
2.	*Blind SQL Injection*	1	0
3.	*Application error message*	11	0
4.	*HTML form without CSRF protection*	3	0
Jumlah Case		**16**	**0**

Proses serupa juga dilakukan pada aplikasi Terintegrasi Fakultas XYZ, di mana *scanning* iterasi kedua dilaksanakan untuk memverifikasi efektivitas optimalisasi dalam mengatasi celah keamanan yang terdeteksi pada *scanning* iterasi pertama. Langkah ini bertujuan untuk memastikan bahwa sistem telah beroperasi dengan aman. Pada aplikasi Terintegrasi Fakultas XYZ, proses *scanning* iterasi kedua dilakukan sebanyak dua kali dengan rincian ditunjukkan pada Tabel 12.

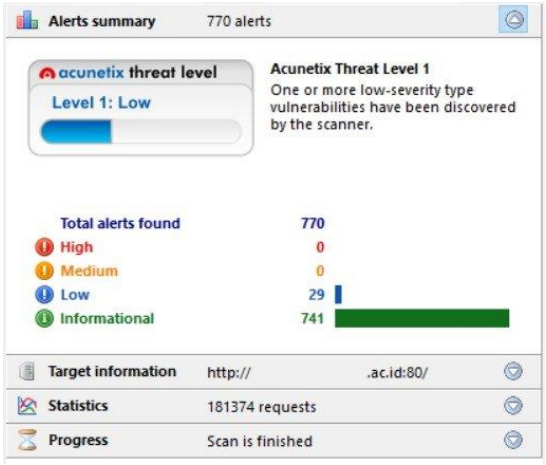
Tabel 12. Aplikasi Terintegrasi Fakultas XYZ Scanning Iterasi 2

No.	Tanggal Pengujian	Durasi Pengujian	Total *Web Alert*
1.	22 Maret 2023	33 Menit 47 Detik	769
2.	22 Maret 2023	35 Menit 15 Detik	770

Berdasarkan hasil scanning iterasi kedua, baik pada *scanning* pertama maupun kedua, jumlah celah keamanan pada aplikasi Terintegrasi Fakultas XYZ menunjukkan penurunan dibandingkan dengan hasil *scanning* iterasi pertama. Namun, pada iterasi kedua ini, *threat level* aplikasi Terintegrasi Fakultas XYZ tetap tidak berubah dibandingkan hasil scanning sebelumnya, sebagaimana terlihat pada Gambar 34 dan Gambar 35.



Gambar 34. Hasil *Scanning* 1 (Iterasi 2) Aplikasi Terintegrasi Fakultas XYZ



Gambar 35. Hasil *Scanning* 2 (Iterasi 2) Aplikasi Terintegrasi Fakultas XYZ

Berdasarkan hasil tersebut, didapatkan informasi bahwa hasil *scanning* yang menampilkan penurunan yang tidak begitu siginifikan pada jumlah *web alert* level medium dengan detail informasi seperti pada Tabel 13. Perbandingan data *web alert* pada aplikasi Terintegrasi Fakultas XYZ setelah dilakukan optimalisasi ditampilkan pada Tabel 14.

Optimalisasi pada *Website* dan Aplikasi Terintegrasi Fakultas XYZ tidak hanya dilakukan dengan meninjau source code tetapi juga melalui pengaturan konfigurasi sistem. Sebagaimana dijelaskan pada bagian teknologi sistem informasi yang digunakan, *Website* dan Aplikasi Terintegrasi Fakultas XYZ dibangun menggunakan *CMS* dan *Framework*, dengan sebagian besar kontennya dikembangkan melalui plugin, komponen, dan ekstensi pihak ketiga. Penggunaan plugin, komponen, dan ekstensi pihak ketiga ini berisiko mengandung *malware* yang dapat mengeksploitasi celah keamanan, terutama jika diunduh atau diinstal dari sumber yang tidak resmi.

Tabel 13. Sebaran *Web Alert* Terintegrasi Fakultas XYZ pada *Scanning* 1 (Iterasi 2)

No.	Thread Level	Jenis Alert	Level Alert	Jumlah Alert
1.		Clickjacking: X-Frame-Options header missing	Low	1
2.		Documentation file	Low	1
3.		Login page password-guessing attack	Low	1
4.		Possible sensitive directories	Low	13
5.	2	Possible sensitive files	Low	13
6.	(Medium)	Broken links	Informational	64
7.		Password type input with auto-complete enabled	Informational	1
8.		Possible internal IP address disclosure	Informational	196
9.		Possible server path disclosure (Unix)	Informational	402
10.		Possible username or password disclosure	Informational	77
		Total Web Alert		769

Tabel 14. Perbandingan Data Web Alert Aplikasi Terintegrasi Fakultas XYZ Setelah Dioptimalisasi

No.	Web Alert	Jumlah Case	
		Sebelum Optimalisasi	Setelah Optimalisasi
1.	Application error message	1	0
2.	Development configuration file	1	0
3.	Directory listing	6	0
4.	Error message on page	32	0
5.	User credentials are sent in clear text	1	
	Jumlah Case	41	0

KESIMPULAN

Berdasarkan analisis permasalahan, rancangan, implementasi, dan analisis data, dapat disimpulkan bahwa penilaian kerentanan pada Website dan Aplikasi Terintegrasi Fakultas XYZ menggunakan Acunetix Vulnerability Web Scanner berhasil mengidentifikasi level ancaman awal pada *website* adalah pada *level high* (3) dan aplikasi pada *level low* (1). Proses ini melibatkan pengkajian celah keamanan dari hasil pemindaian pertama, diikuti dengan eksploitasi yang mengungkap berbagai ancaman mulai dari level tinggi hingga sedang. Selanjutnya, dilakukan optimalisasi keamanan melalui tinjauan source code, konfigurasi, dan penerapan rekomendasi solusi untuk mengatasi celah keamanan yang ditemukan. Optimalisasi ini terbukti efektif, sebagaimana ditunjukkan oleh hasil pemindaian kedua yang menunjukkan penurunan level ancaman pada website dan aplikasi ke *level low* rendah (1), sehingga keamanan sistem meningkat secara signifikan.

REFERENSI

[1] Andriyan, W., dkk. (2020). Perancangan Website sebagai Media Informasi dan Peningkatan Citra Pada SMK Dewi Sartika Tangerang. Jurnal Teknologi Terpadu. Vol. 6 No. 2 2020, 79-88. https://doi.org/10.54914/jtt.v6i2.289

[2] Mayasari, R., dkk. (2020). Analisis Vulnerability pada Website Universitas Singaperbangsa Karawang menggunakan Acunetix Vulnerability. SYSTEMATICS. Vol. 2, No. 1, April 2020, pp 33-38. https://doi.org/10.35706/sys.v2i1.3450

[3] Setiawan, MF., dkk (2022). Penutupan Celah Keamanan Menggunakan Metode Hardening Studi Kasus: *Cloudfri Closing Security Vocations. e-Proceeding of Engineering :* Vol.9, No.2 April 2022, pp 656-663. ISSN : 2355-9365

[4] Alwi, EI., dkk (2021). *Analisis KeamananSistem Informasi Akademik (SIAKAD) Universitas XYZ Menggunakan Metode Vulnerability Assessment*. Informatics Journal (INFORMAL) : Vol. 6 No. 3. pp 131-135. DOI : https://doi.org/10.19184/isj.v6i3.27053

[5] Fajar, FA., dkk/ (2020). Analisis Keamanan Aplikasi Web Prodi Teknik Informatika UIKA Menggunakan *Acunetix Web Vulnerability*. Jurnal INOVA-TIF. Vol. 3 No. 2, pp 110-120. http://dx.doi.org/10.32832/inova-tif.v3i2

[6] Riadi, I., dkk. (2021). Optimasi Keamanan *Web Server* terhadap Serangan *Broken Authentication* Menggunakan Teknologi *Blockchain*. JISKa, Vol. 6, No. 3, September, 2021, pp. 139 – 14. https://doi.org/10.14421/jiska.2021.6.3.139-148

[7] Zirwan, A., dkk. (2022). Pengujian dan Analisis Keamanan *Website* Menggunakan *Acunetix Vulnerability Scanner*. Jurnal Informasi dan Teknologi. Vol. 4 No. 1, pp 70-7. https://doi.org/10.37034/jidt.v4i1.190

[8] Guntoro., dkk (2020). *Analisis Keamanan Web Server Open Journal System (OJS) Menggunakan Metode Issaf dan OWASP (Studi Kasus OJS Universitas Lancang Kuning)*. Jurnal Ilmiah Penelitian dan Pembelajaran Informatika (JIPI). Vol 5, No 1. https://doi.org/10.29100/jipi.v5i1.1565

[9] Listartha, IME., dkk. (2021). Pengujian Kerentanan dan Penetrasi Keamanan pada Aplikasi Web Manajemen Skripsi Prodi XYZ. *ScientiCO : Computer Science and Informatics Journal*. Vol. 4, No. 2, (2021). E-ISSN: 2620-4118

[10] Sandy, Solihin, HH. (2021). Audit Keamanan dan Manajemen Risiko pada e-Learning Universitas Sangga Buana . Jurnal Manajemen Informatika (JAMIKA). Vol. 11 Nomor 1 Edisi April 2021. https://doi.org/10.34010/jamika.v11i1.3641

[11] Ashar, R. (2022). Analisis Keamanan Open Website Menggunakan Metode OWASP dan ISSAF. Jurnal Informasi dan Teknologi. Vol. 4 No. 4 pp 211-218. https://doi.org/10.37034/jidt.v4i4.233

[12] Kritianto, F., dkk. (2022). Analisis Kerentanan pada Website Servio Menggunakan *Acunetix Web Vulnerability*. Journal of Technology Research in Information System and Engineering (JTRISTE). Vol. 9, No. 1, Maret 2022, pp 46~55. https://doi.org/10.55645/jtriste.v9i1.363

[13] Aziz, Muhammad. (2022). Vulnerability Assesment Untuk Mencari Celah Keamanan Web Aplikasi E-Learning Pada Universitas XYZ. *Journal of Engineering, Computer Science and Informatics Technology (JECSIT).* Vol. 1, No. 1, 2021, pp 101-109. https://doi.org/10.33365/jecsit.v2i1

[14] Nasir, SWN., dkk. (2021). *Vulnerability Testing Analysis* of XYZ *Regional Government Site Using PTES*. Jurnal Teknik Informatika dan Sistem Informasi. Vol. 8, No. 3, September 2021, pp 1543-1556. https://doi.org/10.35957/jatisi.v8i3.1224

[15] Orisa, M., Ardita, M. (2021). *Vulnerability Assesment* untuk Meningkatkan Kualitas Kemanan *Web*. MNEMONIC Jurnal Teknik Informatika. Vol 4, No. 1, Februari 2021, pp 16-19. https://doi.org/10.36040/mnemonic.v4i1.3213

[16] Suputri, KA., dkk. (2022). Perbandingan *Tools Vulnerability Scanning* Pada Pengujian Sebuah *Website*. Informatik : Jurnal Ilmu Komputer. Vol 18 No 3 (2022): Desember 2022, pp 269-277. https://doi.org/10.52958/iftk.v18i3.5133

[17] Raazi, IM,. Dkk. (2023). Uji *Vulnerability Assessment* dalam Mengetahui Tingkat Keamanan *Web* Aplikasi Sistem Informasi Laporan Diskominfo Dan Sandi Aceh. JINTECH: *Journal of Information Technology*. Vol. 4, No. 1. Februari 2023, pp : 1 – 15. https://doi.org/10.22373/jintech.v4i1.2409

[18] Budiman, A., dkk. (2021). Analisis Celah Keamanan Aplikasi Web E-Learning Universitas ABC dengan Vulnerability Assesment. Jurnal Komputasi. Vol 9 No. 2 , 2021, pp 1-10. http://dx.doi.org/10.23960%2Fkomputasi.v9i2

[19] Zen, BP., dkk. (2020). Analisis *Security Assessment* Menggunakan Metode Penetration Testing dalam Menjaga Kapabilitas Keamanan Teknologi Informasi Pertahanan Negara. Jurnal Teknologi Penginderaan. Vol 2, No 1 (2020), pp 105-122

[20] Ardita, IKAO., dkk. (2022). Analisis Keamanan Aplikasi Android Dengan Metode *Vulnerability Assessment*. Jurnal Elektronik Ilmu Komputer Udayana. Volume 10, No 3. February 2022, pp 279-286. e-ISSN: 2654-5101

[21] Rahardian, RL. (2022). Analisis Keamanan *Web New* Kuta Golf Menggunakan Metode *Vulnerability Assessments* Dan Perhitungan *Security Metriks*. Jurnal Informatika Dan Teknologi Komputer (JITEK). Vol. 2 No. 3 (2022): November, pp 256-265. https://doi.org/10.55606/jitek.v2i3.582

[22] Ramadhan, RA., dkk. (2022). Edukasi Pemrograman WEB Fundamental Sebagai Ilmu Wajib Era Industri 4.0. Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan. Volume 03, No. 01, 2022, pp 11-15. https://doi.org/10.25299/jpmpip.2022.10591

[23] Mahardika, BT. (2020). Perancangan Sistem Informasi Management Siswa Berprestasi Berbasis Android Pada SMK Pgri Rawalumbu. Jurnal Sains dan Teknologi (JST). Vol. 10 No. 2 (2020). ISSN 2088-060X

[24] Ibrahim, AM., dkk. (2022). Analisis Keamanan Sistem pada Website Perusahaan CV. Kazar Teknologi Indonesia dengan Metode *Vulnerability Assesment and Penetration Testing (VAPT)*. Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA). Jakarta-Indonesia, 14 April 2022. e-ISSN 2962-6129

[25] Soebijono, T., Martinus, SE. (2022). *Audit* Sistem Informasi Menggunakan *Framework Cobit* Pada Sekolah Tinggi "X" Surabaya. Jurnal Riset Mahasiswa Akuntansi (JRMA). Volume X, No. 1, Tahun 2022, pp 71-81. e-ISSN : 2715 – 7016

[26] Zuraidah, E., Sulthon, BM. (2022). Audit Sistem Informasi Penjualan Pada UMKM MAM Menggunakan Framework Cobit 5. JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 5, Oktober 2022. http://dx.doi.org/10.30865/jurikom.v9i5.4985

BIOGRAFI PENULIS

Mifthahul Rahmi, M.Kom
Lulusan S2 Teknik Informatika dari Universitas Putra Indonesia YPTK Padang (2023) dan S1 dari Universitas Andalas (2015). Ia bekerja sebagai staf TIK di Fakultas Kedokteran Universitas Andalas dan memiliki pengalaman mengembangkan berbagai website, termasuk untuk organisasi dokter forensik dan kelompok studi medis. Ia juga pernah mempublikasikan karya ilmiah dalam Prosiding tentang teknologi navigasi robot menggunakan sensor Kinect.

Prof. Dr. Yuhandri, S.Kom, M.Kom
Akademisi dan peneliti dengan gelar Doktor Teknologi Informasi dari Universitas Gunadarma (2017). Saat ini, ia menjabat sebagai Dekan Fakultas Ilmu Komputer di Universitas Putra Indonesia YPTK Padang. Ia memiliki banyak pengalaman dalam publikasi ilmiah, terutama di bidang segmentasi citra, deep learning, dan pengembangan algoritma. Selain itu, ia aktif dalam kegiatan pengabdian masyarakat, seperti pelatihan teknologi dan aplikasi kecerdasan buatan untuk pendidikan serta pengembangan keterampilan teknologi informasi.

Dr. Ir. H. Sumijan, M.Sc, OCA
Akademisi dan peneliti dengan gelar Doktor Teknologi Informasi dari Universitas Gunadarma (2015) dan Master bidang Ilmu Komputer dari Universiti Teknologi Malaysia (1998). Saat ini, ia menjabat sebagai

Pembantu Rektor I di Universitas Putra Indonesia YPTK Padang. Ia aktif meneliti teknologi informasi, keamanan web, dan pengolahan citra digital, serta telah menerbitkan jurnal dan prosiding internasional. Selain itu, ia juga berkontribusi dalam pelatihan teknologi digital dan pengembangan sistem informasi untuk komunitas pendidikan.

https://doi.org/10.25077/jitce.8.2.83-96.2024