JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XII No 1, Juni 2023
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

ANALISIS DAN IMPLEMENTASI SECURE CODE PADA PENGEMBANGAN SISTEM KEAMANAN WEBSITE FIKOM-METHODIST.COM MENGGUNAKAN PENETRATION TESTING DAN OWASP ZAP

Naikson Fandier Saragih [(1)], Reinhard Tamalawe[(2)], Indra M Sarkis[(3)]
Program Studi Informatika
Universitas Methodist Indonesia
Jalan Hang Tuah No. 8 Medan, North Sumatra 20152
email: saragihnaikson@gmail.com[(1)], reinhardtamalawe66@gmail.com[(2)], poetramora@gmail.com[(3)]

Abstrak

Keamanan web menjadi perhatian utama dalam era digital saat ini tidak terkecuali website *fikom-methodist.com* yang merupakan aplikasi untuk administrasi data internal di Fakultas Ilmu Komputer Universitas Methodist Indonesia, dimana portal ini akan terus dikembangkan sesuai kebutuhannya. Sayangnya *fikom-methodist.com* dalam perjalannya masih juga dapat diserang pada tahun 2022. Serangan terhadap aplikasi web menjadi ancaman yang serius bagi organisasi dan pengguna. Penetration testing metode yang dapat digunakan untuk menguji keamanan system sehingga kerentanan dalam aplikasi web dapat teridentifikasi yang selanjutnya digunakan untuk menutup celah keamanan tersebut. Penelitian diawali dengan melakukan assessment menggunakan *Tools Owasp Zap* untuk mendeteksi kerentanan celah CSRF. Dilanjutkan dengan ujicoba serangan pada celah CSRF dan melakukan penambalan menggunakan *secure code* untuk celah yang ada. Terakhir assessment ulang dilakukan untuk melihat tingkat kerentanan setelah penambalan dilakukan untuk memastikan celah CSRF tidak ada lagi. Asesment menggunakan *Tools Owasp Zap pada url https.fikom-methodist.com* terdapat celah kerentanan CSRF 14 celah (absence of Anti-CSRF Token) dan pada Method GET terdeteksi sebanyak 11 serta Method POST sebanyak 3 dengan risiko rrendah (Low). Ujicoba serangan pada celah CSRF dilakukan secara manual pada elemen URL website dengan teknik phising melalui form Register dengan Method POST dimana sumber code form/page diambil dari inspect element/Owasp Zap, yang selanjutnya dimanipulasi dengan menambahkan code berbahaya dengan tujuan menambah akun admin yang seolah-olah bagian dari web fikom-methodist.com. Serangan CSRF one-click, berhasil masuk ke dalam website. Selanjutnya untuk penambalan dengan secure code diimplementasikan menggunakan mekanisme verifikasi permintaan dan token keamanan pada Framework CI dengan mengaktifkan mode *true* pada *$config['csrf_protection']* dan menerapkan fungsi kode *hash* pada setiap formulir untuk memastikan integritas data, mengidentifikasi file dengan unik, dan menyimpan kata sandi dalam database dalam bentuk yang tidak dapat dibaca. Tahapan terakhir dilakukan Penetration Testing ulang untuk memverifikasi efektivitasnya dalam melawan serangan CSRF. Hasil pengujian sistem berhasil melindungi aplikasi web dengan memblok serangan CSRF secara otomatis. Selanjutnya pengujian ulang dengan Owasp Zap, hanya ditemukan 2 kerentanan dengan *Method Get* yang dimana tidak berisiko *(risk low)*. Dua kerentan ini bukan terkait mengolah data tetapi hanya menampilkan suatu data. Sehingga pengembangan system keamanan website fikom-methodist.com dengan secure code telah berhasil diimplementasikan.

Kata Kunci: Keamanan website, *Penetration testing*, *Cross-Site Request Forgery (CSRF)*, Token CSRF, fikom-methodist.com, *secure code*

1. Pendahuluan

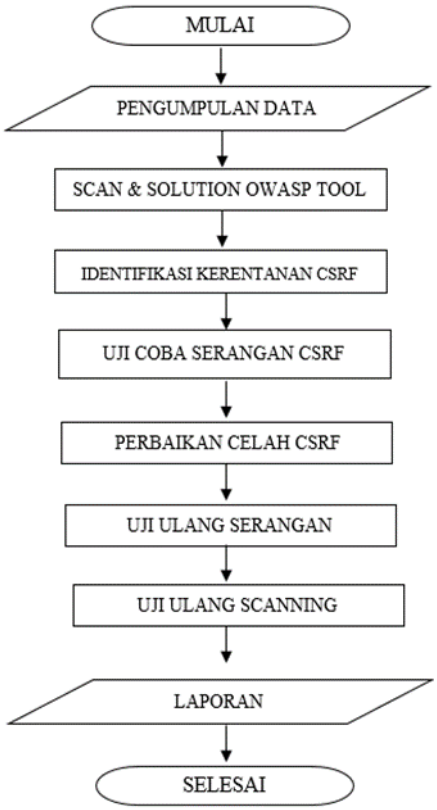
Pembangunan dan pengembangan website harus diikuti dengan adanya kesadaran akan aspek keamanan informasi. Celah kerentanan yang ada pada sebuah website dapat diketahui melalui evaluasi keamanan website, evaluasi keamanan berguna untuk menemukan celah-celah kerentanan yang menjadi kelemahan website tersebut. Kelemahan dalam sebuah sistem baik program, design, ataupun implementasi dinamakan sebagai Vulnerability[].

Dalam laporan itu, terungkap bahwa ada lebih dari 1,6 miliar atau tepatnya 1.637.973.022 anomali trafik atau serangan siber (cyber attack) yang terjadi di seluruh wilayah Indonesia sepanjang tahun 2021 Adapun, serang siber paling banyak terjadi pada bulan Desember 2021 dengan jumlah lebih dari 242 juta anomali. Sementara di bulan Februari, serangan siber paling sedikit terjadi dengan jumlah hampir 45 juta anomali. Menurut perhitungan KompasTekno, bila dirata-rata, tahun lalu, ada lebih dari 136 juta serangan siber yang terjadi setiap bulannya

Gambar 1. Grafik jumlah serangan siber yang di Indonesia
Sumber: https://tekno.kompas.com

Penetration testing merupakan tindakan pengujian sistem dengan cara mensimulasikan bentuk-bentuk serangan terhadap sistem tersebut sehingga akan diketahui tingkat kerentanannya (Saragih & Zebua, 2023).Serangan Cross Site Request Forgery(CSRF) merupakan ancaman aplikasi web yang ditunjukkan untuk mencuri informasi pengguna aplikasi web. Attacker memaksa pengguna untuk mengeksekusi aksi yang tidak di inginkan pada aplikasi web, dimana dia (korban) saat ini terautentikasi. https://fikom-methodist.com adalah website resmi yang dikelola oleh Fakultas Ilmu Komunikasi Universitas Methodist Indonesia (UMI), yang berisi data tugas Akhir(skripsi) Mahasiswa,dan berisi informasi pelaksanaan informasi pengumuman Seminar Proposal,Seminar Hasil dan Seminar meja hijau serta yang lainnya yang akan ditambahkan sesuai kebutuhan.Pada tahun 2022 adanya akses yang dilakukan oleh orang yang tidak dikenal yang dapat merubah dan memodifikasi informasi berupa gambar seperti memasukkan artikel yang tidak berkenan yang tidak berasal dari pengelola (Admin). P Permasalahan tersebut yang menjadi dasar penelitian ini dengan topik "Analisis dan Implementasi Secure Code pada Pengembangan Sistem Keamanan Website fikom-methodist.com Menggunakan Penetration Testing Dan OWASP ZAP".

2. Landasan Teori



Gambar 2.Framework Penelitian

3. Metode Penelitian

Web Penetration Testing

Berdasarkan definisi dalam modul CEH, Web Penetration Testing merupakan metode evaluasi keamanan sistem komputer dengan mensimulasikan serangan dari sumber yang berbahaya dan merupakan bagian dari security audit. Simulasi serangan yang dilakukan dibuat seperti kasus yang bisa dibuat oleh black hat hacker, cracker, dan sebagainya. Tujuannya adalah menentukan dan mengetahui macam – macam serangan yang mungkin dilakukan pada sistem beserta akibat yang bisa terjadi karena kelemahan sistem. Dalam melakukan penetration testing, diperlukan analisa intensif untuk setiap kerentanan yang diakibatkan oleh kelemahan sistem. Nantinya setelah seluruh analisa selesai dilakukan, akan didokumentasikan dan diberikan kepada IT yang bersangkutan beserta solusi dan dampak yang dapat diakibatkan dari celah keamanan yang ada. Hal – hal yang perlu diuji dalam penetration testing ada banyak, hal ini dibutuhkan untuk mengidentifikasi ancaman – ancaman utama seperti kegagalan komunikasi, e- commerce, dan kehilangan informasi rahasia. Penetration testing biasa meliputi Sql Ijection, Cross Site Scripting, Broken Access Control, Unrestricted File Upload (Backdooring), Bruteforce,serta Defacing.

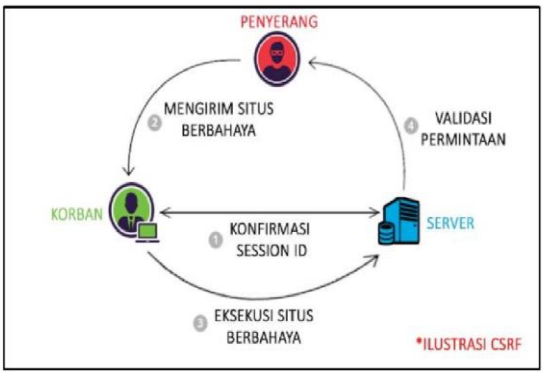
Owasp (*Open Application Security Project)*

Open Web Application Security Project (OWASP) merupakan organisasi non profit berfokus pada peningkatan keamanan perangkat lunak . OWASP menjadi framework yang digunakan oleh pengembang dan ahli teknologi untuk mengamankan website. OWASP memberikan platform bagi pengembang untuk meningkatkan keamanan sistem melalui proyek yang open-source bersama dengan tools dari OWASP sebagai pendukung dalam pengujian sistem.(Kuncoro & Rahma, 2021; Timothy & Daulat, 2021)

Metodologi penilaian risiko OWASP adalah pendekatan sederhana untuk menghitung dan menilai risiko yang terkait dengan aplikasi. dimana dengan metode tersebut dapat diputuskan apa saja yang harus dilakukan terhadap resiko-resiko tersebut . Dengan mengetahui resiko yang akan terjadi maka banyak manfaat yang akan diperoleh diantaranya, menghemat waktu dan mengurangi terjadinya resiko yang lebih serius.

CSRF (*Cross Site Request Forgery)*

Cross Site Request Forgery (CSRF atau XSRF) adalah salah satu celah keamanan yang memanfaatkan program atau task yang digunakan oleh sebuah website. Celah tersebut terjadi karena kurangnya pengamanan pada suatu file yang memproses form tertentu, sehingga attacker dapat melakukan request ke file action tersebut, dengan form yang telah dimodifikasi sebelumnya. CSRF mengeskpolitasi website agar user tetap percaya bahwa website yang digunakan memang benar website yang ingin diakses. CSRF juga dikenal dengan sebutan “one link attack”, karena pada implementasinya sang penyerang hanya perlu menginjeksi sebuah link yang berisi suatu web task URL pada halaman tertentu untuk dibuka oleh calon korban, agar ketika korban membuka halaman tersebut, secara otomatis si korban akan mengeksekusi link URL yang telah diinjeksi sebelumnya. Dalam dokumen OWASP Testing Guide v4 pada subbab Testing for CSRF (OTG-SESS-005), dikatakan bahwa CSRF adalah sebuah serangan yang memaksa pengguna untuk melakukan tindakan yang tidak diinginkan pada aplikasi web saat pengguna tersebut sudah terotentikasi. Dengan merekayasa(phising) seperti mengirim tautan melalui email atau chat), penyerang dapat memaksa pengguna aplikasi web untuk tindakan penyerang. Jika yang ditargetkan untuk dieksploitasi adalah pengguna akun administrator, serangan CSRF dapat membahayakan keseluruhan aplikasi web. Terdapat beberapa hal yang harus terjadi supaya Cross-Site-Request-Forgery berhasil, diantaranya Penyerang harus menemukan formulir disitus target sesuatu yang tidak berguna untuk dia (misalnya, transfer uang, mengubah alamat e-mail, atau password korban); Penyerang harus menentukan nilai-nilai yang tepat untuk segala bentuk masukan. Jika salah satunya diminta untuk menjadi otentikasi nilai rahasia atau ID yang penyerang tidak dapat ditebak penyerang, serangan akan gagal.Para penyerang harus membujuk korban ke halaman Web. CSRF memiliki dua tipe serangan, diantaranya :Stored CSRF, Reflected CSRF. Stored CSRF merupakan serangan dimana penyerang dapat menggunakan aplikasi itu sendiri untuk memberi korban tautan eksploitasi atau konten lainnya yang mengarahkan browser korban untuk melakukan tindakan yang dikendalikan oleh penyerang. Sedangkan Reflected CSRF, merupakan serangan yang memanfaatkan link atau konten diluar sistem aplikasi. Hal ini bisa dilakukan dengan menggunakan email, blog, pesan instan yang terdapat didalam aplikasi tersebut. (Makalalag et al., 2017)



Gambar 3.Implementasi teknik CSRF.
Sumber : Makalalag et al., 2017

1. Tahap pertama : client dan server akan mengkonfirmasi session id dari suatu halaman web tertentu.
2. Tahap kedua : penyerang mengirimkan pesan yang dimana pesan tersebut seolah-olah adalah pesan konfirmasi dari halaman web (contoh.com) Pesan yang dikirimkan berupa link yang di dalamnya sudah diinjeksi perintah untuk menambah admin dari suatu formulir page yang berbeda yang dituju website contoh.com
3. Setelah pesan tersebut terkirim ke korban, selanjutnya korban melakukan klik terhadap link karena mengira pesan tersebut memang benar dari website contoh.com
4. Maka Untuk Penambahan akun admin berhasil dilakukan pada halaman web (contoh.com) dan attacker bisa masuk kedalam website sebagai Administrasi.

4. Hasil Penelitian

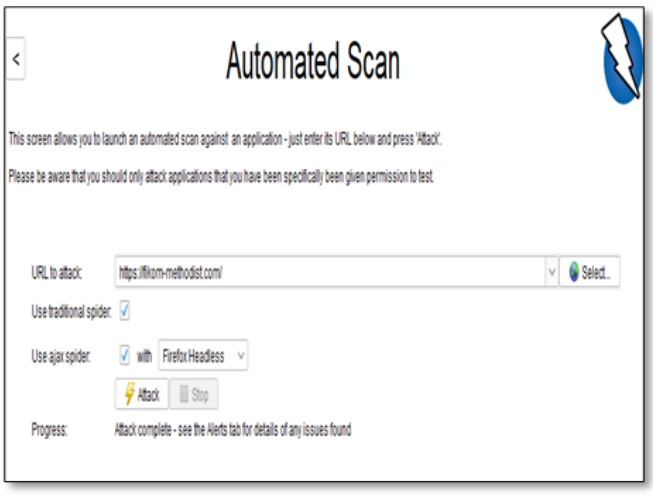
Implementasi Sistem

Tahapan yang pertama yang dilakukan oleh peneliti adalah melakukan instalasi lalu melakukan konfigurasi Proxy pada tool owasp zap versi 2.22.1

Gambar 4. Tampilan Utama Owasp Zap

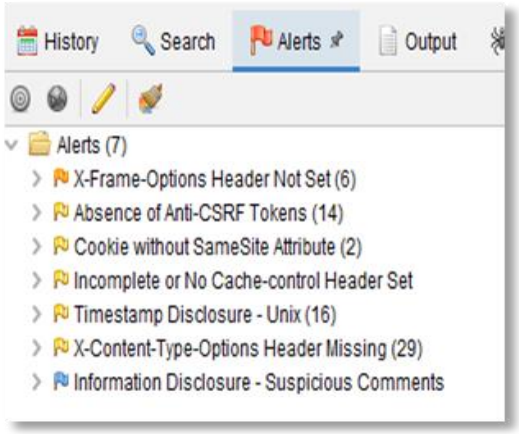
Scanning Domain fikom-methodist.com & Solution Tool Owasp Zap

Setelah melakukan konfigurasi Proxy selanjutnya melakukan Scan pada Web dengan memilih new Session lalu memasukkan url website yang ingin di scan Url to attack : fikom-methodist.com, Ceklis use traditional spider dan selanjutnya lakukan Attack.



Gambar 5. Url To Attack fikom-methodist.com

didalam melakukan scanning web site dengan memasukkan Url fikom-methodistcom pada tool owasp zap akan dibutuhkan waktu kurang lebih 30 menit setelah itu tools owasp zap akan memberikan laporan hasil scanning yang dapat di lihat dengan memilih tab alerts dengan demikian adanya celah Csrf yang dideteksi tools owasp zap yang dimana kerentanannya dengan risk low dan confidence itu medium.



Gambar 6. hasil scanning Tab Alerts

Adanya celah csrf dari hasil scan Attack fikom-methodist.com ditemukan adanya 14 url yang rentan yang bisa ditembus oleh seorang attacker.

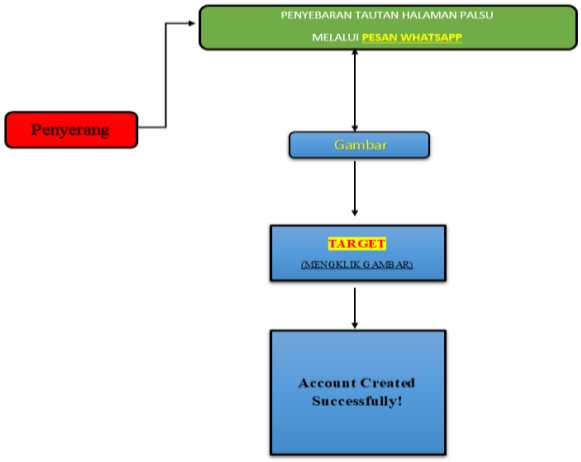
Dan didalam solusi penanganan CSRF diberikan solusi untuk memberikan token acak pada setiap form yang memiliki form action dengan Method Post yang fungsinya Menambah data , menginput data ,menghapus data yang dimana fungsinya akan ditambahkan ke setiap permintaan HTTP yang dikirimkan oleh pengguna ke aplikasi web dari serangan CSRF ,Token ini akan diperiksa oleh server saat permintaan diterima, dan permintaan hanya akan diterima jika token valid.

Identifikasi CSRF

Setelah melakukan Scanning dengan Owasp Zap maka dilakukan identifikasi Serangan *Csrf* yang dimana ketika diperiksa adanya potensi Serangan yang rentan. dilihat bahwa form action dengan *Method Post* yang dimana semua berada pada Form inputan/ouput yang mengolah data, seperti tambah data ,hapus data adupun url nya yakni seperti *<form action=https://fikom-methodist.com/home/register method=post"*

Uji coba serangan CSRF

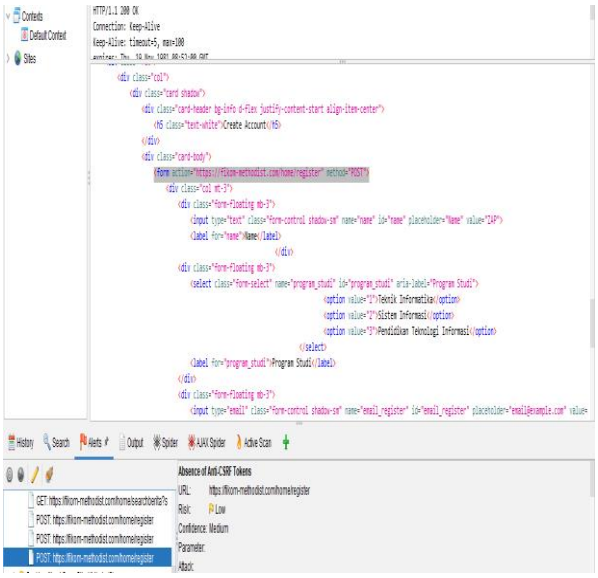
Dalam proses pengujian serangan CSRF dengan cara Membuat sebuah serangan eksploitasi yang membuat pengguna tanpa sadar mengirim sebuah permintaan atau request ke website melalui website yang sedang digunakan saat itu,dari situ aplikasi web akan mengeksekusi request tersebut yang sebenarnya bukan keinginan pengguna..



Gambar 7. Skenario Serangan

Membuat serangan menambah Administrator Baru

Disini peneliti Membuat sebuah konsep serangan yang dimana membuat suatu page yang berisi secure code untuk melakukan penambahan akun administrator yang kodingannya dapat diambil melalui inspect element atau dari tool Owasp Zap fikom-methodist.com mencoba masuk melalui form action method POST: https://fikom-methodist.com/home/register



Gambar 8. *Secure Code Form Action*

***Secure Code* manipulasi serangan Menambah admin**

Penyerang Membuat Halaman web palsu yang seakan akan berasal dari bagian Web Target yang sudah di desain dan Menempatkan Kode HTML pada Situs Palsu yang akan secara otomatis mengirimkan permintaan CSRF ke Aplikasi target saat pengguna mengunjungi situs berikut. yang dimana didalam halaman Web sudah ditanam secure code untuk menambahkan Admin,yang sebelumnya Penyerang sudah mengetahui Level/role keberadaan Administrator .

Yang dimana Administrator :Pada Role 1

untuk Keterangan Penambahan Admin disini Penguji Membuat

Id name : reyi

Untuk Email : rey@gmail.com

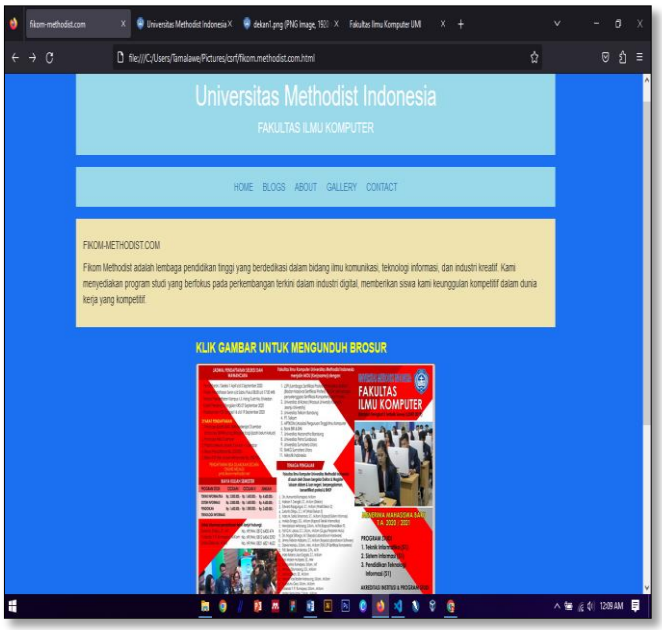
Yang dimana Email tersebut yang akan digunakan dalam Login kedalam Website Targer sebagai Administrator.

Adapun Secure code untuk serangan tersebut adalah sebagai berikut.

```
<form action="fikom-methodist.com/home/register" method="POST">
   <p>
<input type="hidden" id="name" name="name" value="reyi"></p>
<input type="hidden" id="program_studi" name="program_studi" value="1"></p>
<input type="hidden" id="email_register" name="email_register" value="rey@gmail.com"></p>
<input type="hidden" id="role" name="role" value="1"></p>

<button type="register" name="Edit" value="Edit"><img src="methodist_logo.png" width="100%" /></button>
</P>
</div>
</div>
</div>
</form>
<br>
<br>
<br>
```

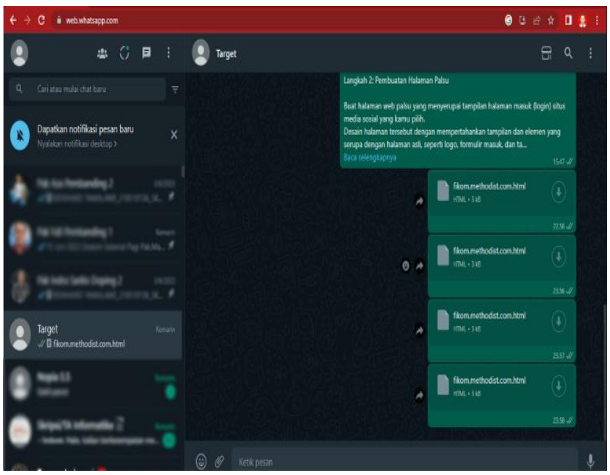
Adapun codingan yang sudah diubah dan dimanipulasi dalam bentuk page yang berisi gambar yang dimana ketika mengklik dengan 1 klik (one click) pada gambar maka otomatis akan menambahkan akun administrator



Gambar 9. halaman palsu yang sudah Buat

Rekayasa Sosial Media(Phising)

Didalam Rekayasa Sosial media dilakukan Penyebaran Tautan Halaman Palsu(phising) yang telah dibuat dan dibagikan kepada si Target melalui Platform sosial media dengan mengirimkan tautan melalui pesan Whatsap kepada Target yang dimana pesan tersebut seolah olah adalah pesan konfirmasi dari halaman fikom-methodist.com



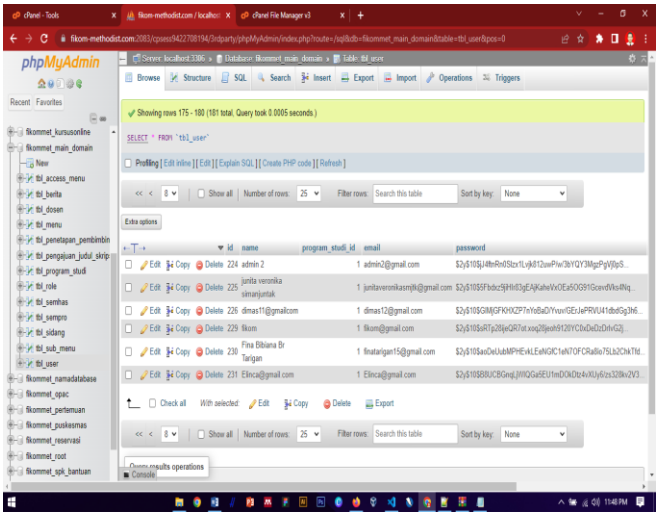
Gambar 10. Kirim tautan melalui Pesan Whatsap

Gambar 11. Tampilan Laman Pengimplementasian CSRF

Ketika Target Membuka Tautan maka akan masuk kedalam sebuah page yang telah di tanam secure code/phising dalam bentuk gambar, yang dimana ketika mengklik dengan 1 klik (one click) akan menambahkan akun administrator secara otomatis.

Database sebelum dilakukan serangan CSRF

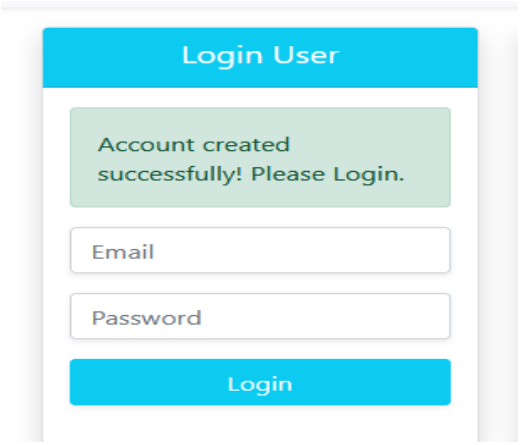
Tampilan Database tabel User dari database fikom-methodist.com sebelum di serang yang terdapat pada database fikom-methodist.com => fikommet_main_domain => tabel user, yang dimana Akun admin belum masuk yang telah masuk kedalam database fikom-methodist.com.



Gambar 12. Tampilan database sebelum diserang

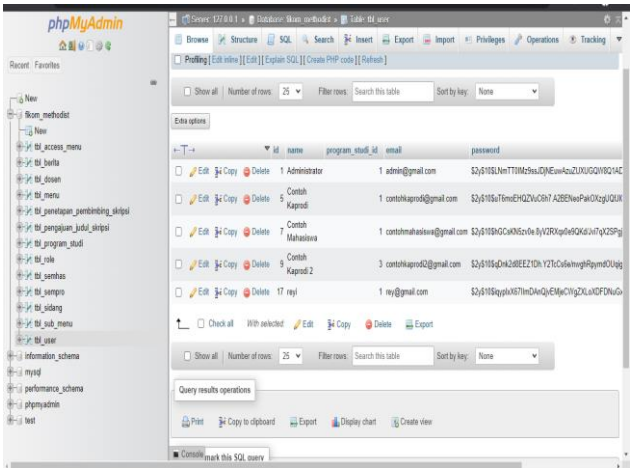
Setelah dilakukan serangan CSRF

Ketika target Mengklik gambar maka otomatis Akan berhasil Menambah akun Admin baru lagi dan akan masuk kedalam Database akun admin yang baru otomatis sukses dibuat.



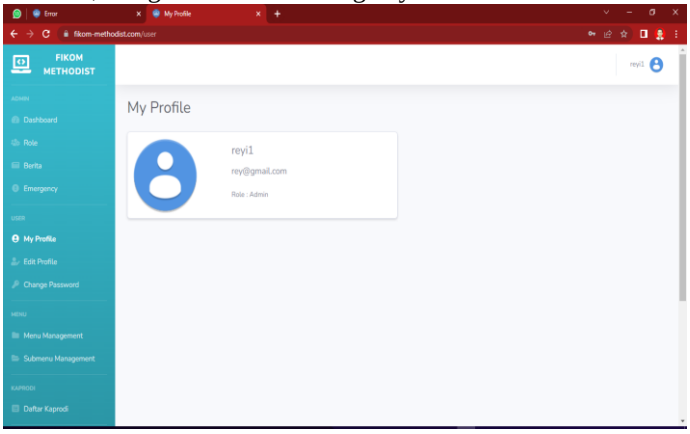
Gambar 13. Accound Created Successfully

Setelah itu maka akan otomatis Akun admin bertambah pada database fikom-methodist.com yang dimana si attacker akan masuk kedalam role admin berhasil memegang sepenuhnya web fikom-methodist.com sehingga si Attacker bebas mengolah data seperti memanipulasi ,mengedit,bahkan menghapus informasi yang berada pada aplikasi fikom-methodist.com



Gambar 14. Tampilan akun admin baru masuk kedalam Database

Oleh sebab itu Penyerang dapat masuk sebagai administrator pada web fikom-methodist.com yang dimana si Penyerang bebas mengupload data,Mengolah dan lain sebagainya.



Gambar 15. Penyerang berhasil Membuat akun sebagai Admin

Perbaikan Celah serangan CSRF

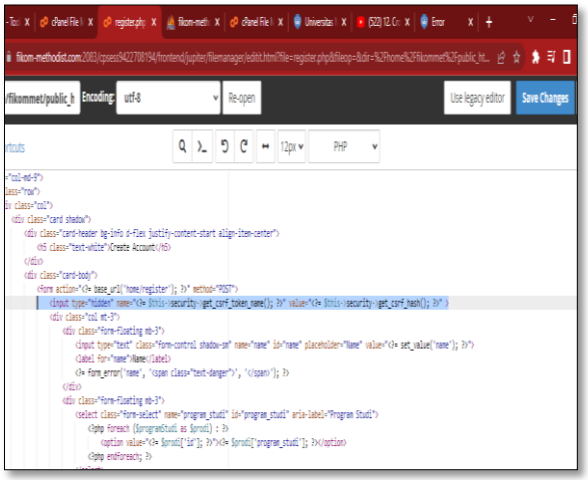
Untuk menangkal serangan CSRF dibutuhkan pemberian token yang random pada Framework Codeigniter yang sudah memiliki fungsi tersendiri untuk menangkal Serangan CSRF ,hanya saja fungsi ini perlu diaktifkan secara manual oleh admin atau programmer Web. letak fungsi tersebut berada pada file config yang terletak didalam folder Application=>config=>config.php adapun tampilan coding dari fungsi csrf ini adalah sebagai berikut. $config['csrf_protection'] adalah sebuah variabel konfigurasi pada framework CodeIgniter yang digunakan untuk mengaktifkan atau menonaktifkan fitur proteksi CSRF (Cross-Site Request Forgery). Ketika $config['csrf_protection'] diatur ke TRUE, CodeIgniter akan otomatis menambahkan token CSRF pada setiap form yang ada di aplikasi web yang menggunakan framework tersebut. Tujuannya adalah untuk mencegah serangan CSRF yang memanipulasi permintaan HTTP yang dikirimkan ke server dari situs web yang berbeda. /home/fikommet/public_html/application/config/config.php

```
438 |--------------------------------------------------------------------------
439 | Cross Site Request Forgery
440 |--------------------------------------------------------------------------
441 | Enables a CSRF cookie token to be set. When set to TRUE, token will be
442 | checked on a submitted form. If you are accepting user data, it is strongly
443 | recommended CSRF protection be enabled.
444 |
445 | 'csrf_token_name' = The token name
446 | 'csrf_cookie_name' = The cookie name
447 | 'csrf_expire' = The number in seconds the token should expire.
448 | 'csrf_regenerate' = Regenerate token on every submission
449 | 'csrf_exclude_uris' = Array of URIs which ignore CSRF checks
450 */
451 $config['csrf_protection'] = TRUE;
452 $config['csrf_token_name'] = 'csrf_test_name';
453 $config['csrf_cookie_name'] = 'csrf_cookie_name';
454 $config['csrf_expire'] = 7200;
455 $config['csrf_regenerate'] = TRUE;
456 $config['csrf_exclude_uris'] = array();
457
```

Gambar 16. Config _csrf_protection diatur ke TRUE

Ketika CSRF Protection diatur ke True maka otomatis website fikom-methodist.com memblok serangan CSRF yang dimana si penyerang tidak bisa menambahkan akun atau masuk kedalam website tersebut. Namun, jika $config['csrf_protection'] diatur ke FALSE, maka fitur proteksi CSRF pada CodeIgniter akan dinonaktifkan. Dalam kondisi ini, aplikasi web tidak akan menambahkan token CSRF pada setiap form, sehingga dapat memudahkan penyerang untuk melakukan serangan CSRF.

Dalam menampilkan token CSRF perlu menambahkan code program disetiap form yang fungsinya memvalidasi semua kode hash disemua halaman yang menggunakan form Action yang dimana fungsi kode hash adalah digunakan untuk menyimpan informasi dalam bentuk yang terenkripsi dan aman, tanpa harus menyimpan data asli. Hash Code biasanya digunakan untuk memastikan integritas data, mengidentifikasi file dengan unik, dan menyimpan kata sandi dalam database dalam bentuk yang tidak dapat dibaca.

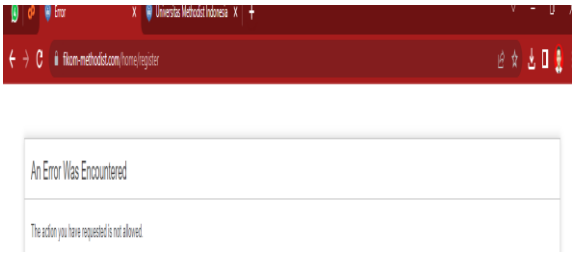


Gambar 17. Penambahan secure code untuk menghasilkan token CSRF

Uji Ulang Serangan

Maka setelah dilakukan pengujian kembali dan keamanan *secure code* Ketika *$config['csrf_protection']* diatur ke *TRUE*,maka *CodeIgniter* akan otomatis menambahkan token CSRF pada setiap form yang ada di aplikasi web yang menggunakan framework tersebut dan otomatis Tujuannya adalah untuk mencegah serangan CSRF yang

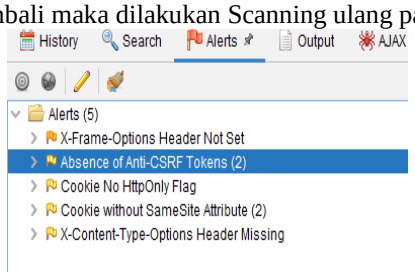
memanipulasi permintaan HTTP yang dikirimkan ke server dari situs web yang berbeda. Gambar hasil blok dari serangan CSRF dapat dilihat pada table



Gambar 18. Hasil Blok *CSRF*

Scanning Ulang

Setelah melakukan Uji serangan kembali maka dilakukan Scanning ulang pada domain web fikom-methodist.



Gambar 19. Scan ulang domain fikom-methodist.com

Maka untuk setiap form action yang menggunakan method Post yang fungsinya menambah,menghapus sudah tidak ada celah lagi ,dan ditemukan 2 url form action yang hanya bermethod Get yang dimana fungsi dari Get hanya menampilkan data.

Laporan

Setelah berhasil melalukan penambalan CSRF dengan menggunakan Token anti CSRF dan pengujian maka setelah itu membuat laporan penyajian ,maka diperlukan penyajian dalam bentuk tabel Resiko Kerentanan.

Tabel 1.Resiko Kerentanan

Jenis Kerentanan	Metode Pengiriman Data	Resiko Kerentanan CSRF
CSRF	Post	Low

Adapun laporan tabel Pengujian CSRF yang dilakukan dengan penyerangan didalam form action denga Dengan Method Post.

Tabel 2. Pengujian Berhasil

Serangan	**Form Action**	**Pengujian**
CSRF *Cross-Site Request Forgery*	Method=Post	Berhasil

5. Kesimpulan

Aassessment menggunakan Tools Owasp Zap untuk mendeteksi kerentanan CSRF berhasil dilakukan.pada web dengan menemukan celah kerentanan CSRF 14 celah (absence of Anti-CSRF Token) dan pada Method GET terdeteksi sebanyak 11 serta Method POST sebanyak 3 dengan risiko rrendah (Low). Ujicoba serangan secara manual menggunakan rekayasa social media(phising). pada elemen URL website melalui form Register dengan Method POST. Serangan CSRF one-click, berhasil masuk ke dalam. Upaya penambalan dengan menambahkan secure code pada framework Codenighter menggunakan mekanisme verifikasi permintaan dan token keamanan pada Framework CI dengan mengaktifkan mode true pada $config['csrf_protection'] dan menerapkan fungsi kode hash pada setiap formulir untuk memastikan integritas data, mengidentifikasi file dengan unik, dan menyimpan kata sandi dalam database dalam bentuk yang tidak dapat dibaca, dan dari pengujian ulang serangan CSRF sudah tidak dapat dilakukan demikian juga pada assessment ulang celah CSRF sudah bersih.

6. Daftar Pustaka

[1] Ashari, I. F., Oktarina, V., Sadewo, R. G., & Damanhuri, S. (2022). Analysis of Cross Site Request Forgery (CSRF) Attacks on West Lampung Regency Websites Using OWASP ZAP Tools. Jurnal Sisfokom (Sistem Informasi Dan Komputer), 11(2), 276–281. https://doi.org/10.32736/sisfokom.v11i2.1393

[2] Elanda, A., & Buana, R. L. (2020). Analisis Keamanan Sistem Informasi Berbasis Website Dengan Metode Open Web Application Security Project (OWASP) Versi 4: Systematic Review. CESS (Journal of Computer Engineering, System and Science), 5(2), 185. https://doi.org/10.24114/cess.v5i2.17149

[3] Kuncoro, A. W., & Rahma, F. (2021). Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review. Automata, 3(1), 1–5. https://www.sciencedirect.com

[4] Makalalag, R., Najoan, X. B. N., Jacobus, A., Studi, P., Informatika, T., Teknik, F., Ratulangi, U. S., Pendahuluan, I., & Courtial, F. (2017). Kajian Implementasi Cross Site Request Forgery (Csrf) Pada Celah Keamanan Website. Jurnal Teknik Informatika, 12(1).

[5] Priambodo, D. F., Rifansyah, A. D., & Hasbi, M. (2023). Penetration Testing Web XYZ Berdasarkan OWASP Risk Rating. Teknika, 12(1), 33–46. https://doi.org/10.34148/teknika.v12i1.571

[6] Saragih, N. F., & Zebua, T. (2023). Analisis Keamanan dan Implementasi secure code pada Pengembangan Keamanan website fikom-methodist . com Menggunakan Penetration Testing dan CVSS. 7(1), 242–253.

[7] Siregar, H. P., & Wijaya, T. (2018). Membangun Keamanan Dari Serangan Cross-Site Request Forgery (CSRF). Information and Communication Technologies in Tourism, 1(1), 58–68. http://sisfotenika.stmikpontianak.ac.id/index.php/enter/article/view/795/559

[8] Timothy, R., & Daulat, B. (2021). Ringkasan Proposal TA. April. https://doi.org/10.13140/RG.2.2.33691.80169

[9] Wardhana, A. W., & Seta, H. B. (2021). Analisis Keamanan Sistem Pembelajaran Online Menggunakan Metode ISSAF pada Website Universitas XYZ. Informatik : Jurnal Ilmu Komputer, 17(3), 226. https://doi.org/10.52958/iftk.v17i3.3653