JURNAL KHATULISTIWA INFORMATIKA
p-ISSN: 2339-1928 e-ISSN: 2579-633X

VOL. 11 NO. 2 Desember 2023
Halaman : 92 - 102

Analisis Kerentanan Aplikasi Akademik Berbasis Website XYZ Menggunakan OWASP

Sabariman [1]; Haeruddin [2]; **Deven Lee** [3]

Teknologi Informasi, Fakultas Ilmu Komputer
Universitas Internasional Batam
devenlee4@gmail.com

INFO ARTIKEL

Diajukan :
18 Oktober 2023

Diterima :
01 November 2023

Diterbitkan:
01 Desember 2023

Kata Kunci :
OWASP, Aplikasi Berbasis Website; Kerentanan; Teknologi Informasi dan Komunikasi; Keamanan

INTISARI

Perkembangan teknologi informasi dan komunikasi (TIK) merupakan alasan utama sebuah instansi atau perusahaan perlu beradaptasi. Perkembangan TIK mengharuskan pengelolanya menerapkan sistem keamanan termasuk pada aplikasi berbasis website. Penerapan keamanan pada aplikasi berbasis website ditujukan untuk mengatasi kemungkinan serangan *cyber*. Aplikasi akademik berbasis website XYZ merupakan layanan validasi data yang diberikan universitas XYZ kepada mahasiswa. Untuk memeriksa keamanan aplikasi ini diperlukan proses pengujian kerentanan. Dalam penelitian ini pengujian dilakukan dengan menggunakan metode *Open Web Application Security Project* (OWASP). Beberapa *tools* yang digunakan diantaranya *SSL Scan*, *Whois*, *Nmap*, *Shodan.io*, *Google Chrome*, *Metasploit Framework*, dan *OWASP ZAP* untuk mengidentifikasi kerentanan sesuai dengan standar Top OWASP 2021 dan Top OWASP API Security Risk 2023. Melalui proses pengujian kerentanan tersebut, ditemukan sebanyak delapan belas bentuk kerentanan dengan tingkat risiko *high*, *medium*, *low*, dan *informational* dengan bentuk kerentanan yang paling *critical* adalah *broken object level authorization* untuk tahun 2023.Sementara untuk 2021 kerentanan yang paling *critical* adalah *broken access control.* Oleh karena itu diperlukan tindak lanjut pada aplikasi akademik XYZ berdasarkan penyebab yang ditemukan. Adapun solusi yang diberikan dalam penelitian ini sesuai dengan tingkat risiko kerentanan.

I. PENDAHULUAN

Saat ini perkembangan teknologi informasi dan komunikasi (TIK) sangat pesat (Huda, 2020). Dilansir Badan Pusat Statistik, perkembangan dan pengguna teknologi informasi dan komunikasi di Indonesia bertumbuh dengan cepat dimana pada tahun 2018 jumlah pengguna internet baru sebesar 39,90% dari jumlah penduduk Indonesia. Kemudian pada tahun 2022 mengalami peningkatan hingga 66,48%. Teknologi informasi dan komunikasi tersebut umumnya dimanfaatkan untuk memenuhi kebutuhan sehari-hari di mana akses dapat dilakukan dari mana saja, oleh siapa saja, dan kapan saja melalui jaringan internet (Asmawi et al., 2019).

Penggunaan jaringan internet yang mudah dapat menimbulkan dampak positif atau negatif tergantung pada pemanfaatannya. Informasi yang diperoleh melalui internet merupakan informasi dari sumber global. Dengan banyaknya kebutuhan akses informasi maka banyak instansi ataupun perusahaan berlomba-lomba membangun aplikasi berbasis website untuk meningkatkan produktivitasnya (Kusuma, 2022).

Universitas XYZ merupakan lembaga pendidikan tinggi yang telah menerapkan penggunaan teknologi informasi dan komunikasi untuk meningkatkan produktivitas layanan kepada mahasiswa. Penerapan aplikasi berbasis website pada XYZ sebagai wadah informasi umum sudah digunakan sejak tahun 2003. Pada tahun 2010, XYZ memfokuskan pengembangan efisiensi dan relevansi sehingga merambah pada penerapan aplikasi berbasis website dalam lingkup akademik. Perkembangan penggunaan aplikasi berbasis website yang signifikan terhitung mulai tahun akademik 2020/2021. Dalam implementasinya, semua informasi akademik dapat diakses melalui aplikasi berbasis website. XYZ menyediakan sebuah aplikasi berbasis website sebagai wadah validasi data mahasiswa.

Aplikasi berbasis website merupakan sebuah situs internet yang memberikan informasi berupa tulisan, gambar, suara, ataupun animasi sumber data yang dimuat melalui HTTP (Hypertext Transfer Protocol). Perancangan aplikasi berbasis

website, memerlukan kerangka seperti HTML (Hypertext Markup Language) dan bahasa pemrograman seperti PHP, Javascript, Java, Phyton, dan lain-lain. Untuk memaksimalkan penampilan dari aplikasi berbasis website, digunakanlah *Cascading Style Sheet* atau yang disebut CSS (Putra et al., 2022). Penggunaan bahasa pemrograman atau teknologi yang meningkatkan performa dari aplikasi berbasis website dapat disesuaikan dengan perkembangannya.

Aplikasi berbasis website berkembang cepat sehingga mengharuskan pemiliknya dapat beradaptasi dengan perkembangan teknologi. Sah satunya yang berkenaan dengan aspek keamanan pada aplikasi berbasis website yang dibuat untuk menghindari tindakan *hacking* (Priyawati et al., 2022). Meluasnya popularitas penggunaan internet menyebabkan peningkatan aktivitas serangan siber. *Hacking* dapat didefinisikan sebagai akses khusus atau istimewa ke suatu sistem dengan menggunakan perangkat lunak melalui sebuah teknologi komputer untuk menemukan sistem yang rentan untuk ditembus (Ul Haq et al., 2022). Jika *hacker* telah terhubung dengan jaringan komputer, ia akan mencari kerentanan pada sebuah sistem atau jaringan dengan peralatan yang terkontrol untuk mencuri data penting orang lain dan menjualnya ataupun menghancurkan data targetnya yang di kemudian hari dapat menyebabkan masalah kepada korbannya. Tindakan seperti ini umumnya dilakukan oleh *black hat hacker* atau disebut *hacker* jahat (Smith et al., 2022). Oleh sebab itu, untuk mengantisipasi resiko tersebut diperlukan peran *network security*.

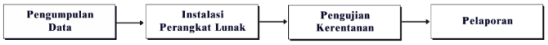
Network security team akan memberikan layanan pengembangan dan pertahanan untuk jaringan komputer yang terkoneksi internet sehingga komputer dapat bekerja dan bertukar data secara aman dan terpercaya (I. F. Ashari et al., 2022). Pemeriksaan rutin yang dilakukan *network security team* terhadap aplikasi berbasis website merupakan proses yang penting dalam pengembangan keamanan aplikasi dari serangan *hacker* dengan *self-test* yang bertujuan untuk mengukur kualitas dan memastikan tidak ada kerentanan pada aplikasi berbasis website. Untuk memeriksa kerentanan yang ada pada aplikasi berbasis website, dapat menggunaan OWASP (I. F. A. Ashari et al., 2023).

OWASP (*Open Web Application Security Project*) ialah sebuah organisasi nirlaba berbasis komunitas yang melakukan promosi keamanan *software* melalui *open-source software*, materi pendidikan, dan inisiatif lainnya (Wen & Katt, 2023). OWASP ialah *framework* yang digunakan untuk mengamankan website oleh ahli teknologi dan pengembang. OWASP memberikan wadah untuk pengembang dalam peningkatkan keamanan sistem dengan merancang dan mengembangankan *tools* berbasis *open-source* yang digunakan sebagai pendukung dalam pengujian sistem (Kuncoro & Rahma, 2021). OWASP ZAP merupakan salah satu *tools* yang dikembangkan oleh OWASP untuk menemukan kerentanan dalam aplikasi berbasis website dengan *scanner* otomatis. ZAP (*Zad Attack Proxy*) bersifat mudah diinstal, *open source*, *community based*, *intercepting proxy*, *active scanner*, *growing add-ons*, *forced browsing*, *traditional & ajax spider*, *fuzzer*, *dynamic,* SSL certificates, *integrated, websocket support,* dan *smard card support* (Riadi et al., 2020). Pada penelitian Mu'min et al. (Mu'min et al., 2022) dengan judul Analisis Keamanan Sistem Informasi Akademik Menggunakan OWASP framework, memiliki tujuan untuk menganalisa kerentanan berdasarkan top 10 OWASP 2021 dengan menggunakan tools OWASP ZAP, SSL Scan, Nmap, dan WhoIS. Pada penelitian Priambodo et al. (Priambodo et al., 2023) dengan judul Penetration Testing Web XYZ Berdasarkan OWASP Risk Rating, memiliki tujuan untuk mencari sistem kerentanan pada aplikasi layanan pembuatan dokumen kependudukan dengan menggunakan dua *tools* pemeriksaan kerentanan (OWASP ZAP dan Vega) dengan acuan top 10 OWASP 2021. Pada penelitian Idris et al. (Idris et al., 2022) dengan judul Web Application Security Education Platform Based on OWASP API Security Project, memiliki tujuan untuk menganalisa sistem kerentanan API-*based system* pada aplikasi berbasis website mengacu pada top 10 OWASP 2021. Pada penelitian Ariyadi et al. (Ariyadi et al., 2023) dengan judul Analisis Kerentanan Keamanan Sistem Informasi Akademik Universitas Bina Darma Menggunakan OWASP, memiliki tujuan mencari celah keamanan dari website dengan *action research method*. Pada penelitian Ashari et al. (I. F. Ashari et al., 2022) dengan judul Analysis of Cross Site Request Forgery (CSRF) Attacks on West Lampung Regency Websites Using OWASP ZAP Tools, memiliki tujuan untuk memanfaatkan OWASP ZAP *tools* untuk menganalisa kerentanan website dengan serangan CSRF. Berdasarkan penelitian-penelitian terdahulu di atas yang memanfaatkan teknologi OWASP ZAP untuk mengidentifikasi kerentanan hanya mengacu pada top 10 OWASP 2021.

Untuk itu, pada penelitian ini selain mengacu pada top OWASP 2021 juga pada top OWASP API Security Risk 2023 dengan menggunakan metode OWASP. Penelitian ini dilakukan dengan menggunakan *tools* OWASP ZAP, Shodan.io, Metasploit Framework, SSL Scan, Nmap dan Whois dengan tujuan untuk mengetahui kerentanan dan memberikan solusi terkait permasalahan kerentanan yang ditemukan.

II. BAHAN DAN METODE

Pada penelitian ini, digunakan metode OWASP yang tujuan untuk menganalisis kerentanan aplikasi akademik XYZ yang berbasis website. Terdapat tahap-tahap penelitian yang dirancang seperti ditunjukkan gambar 1:



Sumber: Data Hasil Pengolahan (2023)

Gambar 1. Tahapan Penelitian OWASP

Berdasarkan gambar 1, terdapat tahapan utama dalam metode penelitian mulai dari pengumpulan data, instalasi perangkat lunak, pengujian kerentanan, analisis, dan pelaporan. Proses penelitian ini dibuat berdasarkan metode yang digunakan yaitu OWASP. Penjelasan lebih lanjut untuk setiap tahapan adalah sebagai berikut:

2.1.1 Tahap Pengumpulan Data

Melakukan pengumpulan informasi terkait target penelitian. Pengumpulan data tersebut meliputi *domain*, *subdomain*, dan informasi terkait aplikasi berbasis website melalui tahap wawancara dengan pihak terkait/*staff IT* universitas XYZ.

2.1.2 Instalasi Perangkat Lunak

Persiapan dengan melakukan instalasi *software* yang diperlukan dalam penelitian seperti Kali Linux OS, Nmap, dan OWASP ZAP yang berbasis *open source*. *Open Source* merupakan aplikasi yang dapat diakses dengan mudah dan gratis.

2.1.3 Pengujian Kerentanan

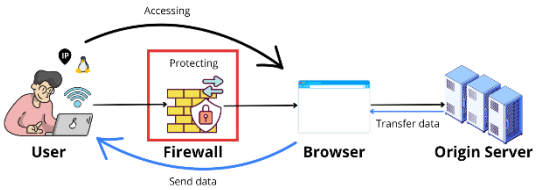
Melakukan *penetration testing* atau pengujian kerentanan dengan tahap *information gathering*, *Vulnerable Analysis*, dan *Exoploiting*. Di dalam tahap *information gathering* dilakukan pengumpulan informasi lebih lanjut melalui pengamatan terhadap aktivitas pemilik target atau langsung melakukan pencarian pada sistem aplikasi berbasis website.

Untuk *vulnerable analysis*, dilakukan analisis terkait faktor kerentanan yang ada pada aplikasi berbasis website dengan menggunakan *tools* dari Kali Linux, Nmap, Shodan.io, dan OWASP ZAP. Selanjutnya, *exploiting* dilakukan sebagai tindak lanjut terhadap kerentanan yang ditemukan pada aplikasi berbasis website, umumnya dengan melakukan penyerangan dan mencari tahu dampak yang akan diterima oleh pemilik aplikasi berbasis website. Penyerangan khusus tersebut dilakukan jika ada beberapa faktor atau informasi yang menyebabkan kerentanan tidak dapat dieksploitasi.

2.1.4 Pelaporan

Pada tahap terakhir, dilakukan penguraian hasil dari analisis penelitian yang dilakukan dan dimasukkan ke dalam laporan.

2.1 Skenario Penelitian



Sumber: Data Hasil Pengolahan (2023)

Gambar 2. Skenario Penelitian OWASP

Pada tahap ini skenario serangan pada penelitian ini disusun. Ilustrasinya ditunjukkan oleh Gambar 2.Diawali dengan menghubungkan perangkat keras pada jaringan internet. Kemudian melakukan penginstalan perangkat lunak seperti Virtual Box dan Kali Linux. Dilanjutkan dengan pencarian celah kerentanan pada aplikasi Akademik berbasis website XYZ dengan memanfaatkan *tools* OWASP ZAP, Nmap, dan Shodan.io sehingga kerentanan pada *website* yang terlindungi oleh Firewall pada *web server* akan didapatkan.Eksekusi kerentanan dilakukan dengan penyerangan yang menggunakan Metasploit Framework dan Google Chrome.

2.2 Alat Penelitian

Untuk mendukung penelitian ini, beberapa alat digunakan untuk mencari dan menghitung tingkat risiko kerentanan pada aplikasi akademik berbasis website XYZ, seperti dinyatakan pada Tabel 1 berikut:

Tabel 1. Alat-alat Penelitian

Perangkat Keras	Perangkat Lunak
Laptop ROG GL553vd • Processor I7 Gen7 • Sodimm 32 GB • *Solit State Drive* 500 GB • *Hard Disk Drive* 1 TB • Wifi External (SolNet)	Sistem operasi Windows 10 Home Virtual Box Kali Linux 2023.3 Google chrome ver. 119.0.6045.105 Shodan.io Nmap Metasploit Framework OWASP ZAP Whois SSL Scan

Sumber: Data Hasil Pengolahan (2023)

Pada proses pengujian kerentanan, terdapat tahap-tahap yang mengklasifikasikan beberapa alat yang digunakan pada setiap tahapnya, seperti ditunjukkan pada tabel 2 berikut:

Tabel 2. Klasifikasi Alat Tahapan Pengujian Kerentanan

Tahapan	Alat	Keterangan
Information Gathering	SSL Scan dan Whois	Mencari informasi terkait *domain* ataupun *sub-domain*.
Vulnerable Analysis	Nmap, Shodan.io, dan OWASP ZAP	Memeriksa port jaringan dan kerentanan.
Exploiting	Google chrome, OWASP ZAP, dan Metasploit Framework	Eksekusi kerentanan.

Sumber: Data hasil pengolahan (2023)

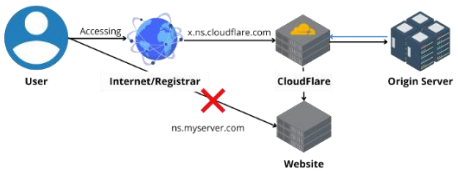
III. HASIL DAN PEMBAHASAN

Pada bagian hasil, pembahasan secara berurutan dimulai dari pengumpulan data, instalasi perangkat lunak, pengujian kerentanan, dan pelaporan, sebagai berikut:

3.1 Pengumpulan Data

Melalui tahap pengumpulan data, peneliti telah melewati proses wawancara dengan pihak terkait atau *staff IT* universitas XYZ untuk menggali informasi lebih dalam terkait aplikasi akademik berbasis website XYZ. Melalui proses ini dihasilkan data sebagai berikut;

a. Topologi jaringan



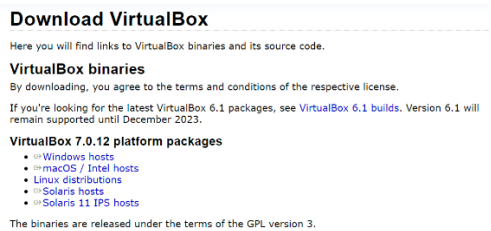
Gambar 3. Topologi Jaringan

Gambar 3. merupakan topologi jaringan aplikasi akademik berbasis website XYZ yang diperoleh dari proses wawancara dan analisis. Topologi jaringan ini menggunakan teknologi *Cloudflare* untuk mengantisipasi akses langsung terhadap *origin server* dan aplikasi berbasis website. Pengarahan ke server *Cloudflare* terdekat oleh pengakses ditujukan untuk mengatasi kepenuhan *traffic* yang masuk ke dalam web server dan menghindari akses yang tidak diinginkan memasuki *origin server* untuk pencurian data. Teknik ini merupakan teknik yang memanfaatkan pihak ketiga untuk mengatasi keamanan server utama.

3.2 Instalasi Perangkat Lunak

Setelah mendapatkan data yang diperlukan, selanjutnya dilakukan analiisa kebutuhan alat yang akan digunakan dalam mencari kerentanan pada aplikasi akademik berbasis website XYZ. Kemudian dilakukan instalasi perangkat lunak. Instalasi perangkat lunak tersebut mencakup:

a. Virtual Box

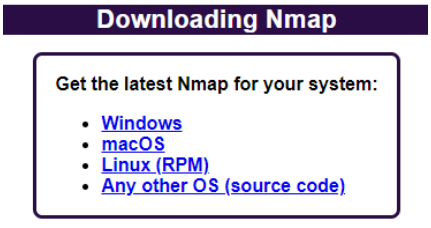


Sumber: https://www.virtualbox.org/wiki/Downloads

Gambar 4. Unduh Virtual Box

Sebelum proses instalasi virtual box, perlu dipastikan bahwa perangkat lunak komputer yang digunakan harus sesuai. Pada penelitian ini, digunakan laptop dengan sistem operasi Windows 10 sehingga dilakukan penginstalan virtual box dengan file type untuk Windowsn seperti ditunjuk pada gambar 4. Penggunaan virtual box dimaksudkan untuk mengantisipasi kesalahan penginstalan pada sistem operasi laptop dan menjauhkan bahaya *malware* atau virus pada laptop.

b. Nmap



Sumber: https://nmap.org/download.html

Gambar 5. Unduh Nmap

Pada proses instalasi Nmap, perlu dipastikan bahwa *installer* yang diunduh sesuai dengan sistem operasi pada laptop yang digunakan. Pada penelitian ini, Nmap yang digunakan berbasis sistem operasi Windows dan diunduh seperti ditunjuk pada gambar 5.

Penggunaan Nmap untuk mengetahui *internet security port* yang terbuka pada *dns/domain* dari aplikasi akademik Berbasis website XYZ.

c. Kali Linux

Sumber: https://www.kali.org/get-kali/#kali-platforms

Gambar 6. Unduh Kali Linux

Sebelum proses instalasi Kali Linux, perlu dipastikan penginstalan sistem operasi sesuai dengan platform yang dipakai. Penginstalan yang dimaksud ialah Kali Linux *images*. Pemilihan *image* tergantung pada tipe *processor* yang digunakan. Untuk penelitian ini, digunakan *installer image* dengan tipe *processor* 64-bit, seperti ditunjuk pada gambar 6. Setelah mengunduh *Installer image*, *import* file *image* ke dalam virtual box. Pada Kali Linux, sudah terdapat berbagai jenis *tools* untuk pengujian seperti Nmap, Metasploit Framework, Amass, Dmitry, dan lain-lain. Namun, dalam penelitian tools yang digunakan hanya Nmap, Metasploit Framework.

d. OWASP ZAP

Sumber: Data Hasil Pengolahan (2023)

Gambar 7. OWASP ZAP

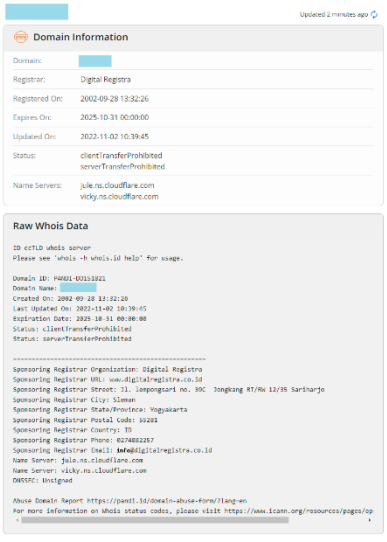
Pada tahap penginstalan OWASP ZAP, file terlebih dahulu diunduh dari website https://www.zaproxy.org/download/ dan menyesuaikan sistem operasi. Pada penelitian ini, digunakan *installer* pada sistem operasi Linux. Setelah diinstal, dapat dilakukan pengecekan sukses atau tidaknya proses penginstalan melalui fitur pencarian. Jika berhasil maka akan ditemukan file zap dan jika diklik maka akan muncul tampilan seperti Gambar 7.

3.3 Pengujian Kerentanan

Pengujian kerentanan merupakan sebuah upaya yang dilakukan seseorang untuk mengidentifikasi kerentanan pada suatu sistem. Pelaksanaannya dilakukan seperti peretas sedang melakukan eksploitasi.Tindakan ini dilakukan untuk menguji kelayakan keamanan sistem untuk mengurangi risiko peretasan (Mu'min et al., 2022). Pada pengujian kerentanan aplikasi akademik berbasis website XYZ digunakan tiga tahapan sebagai berikut:

3.3.1 Information Gathering

Pada tahap *information gathering* dilakukan *scanning* tentang website menggunakan Whois dan SSL Scan. Pencarian ini mencakup informasi detail terkait *domain* dan uji kelayakan SSL pada website. Dengan menggunakan *tools* Whois, menghasilkan lampiran hasil scanning, sebagai berikut:



Sumber: Data Hasil Pengolahan (2023)

Gambar 8. Tool Whois

Berdasasarkan Gambar 8. terlihat hasil dari pencarian mengenai asal-usul *domain* pada aplikasi akademik berbasis website XYZ lengkap dengan informasi *domain*, tempat pendaftaran, tanggal awal pendaftaran, tanggal kadaluarsa, *domain update*, status, dan *name servers*. Melalui pencarian tersebut didapatkan target penelitian merupakan

sub-domain dan terdapat peran pihak ketiga dalam menangani server pada *domain* target.

Selanjutnya dilakukan pemeriksanaan *Secure Socket Layer* atau SSL yang merupakan rancangan yang berfungsi untuk memberikan jaminan layanan dan keamanan data yang terenkripsi (Sahren, 2021). Pada penelitian ini, pemeriksaan SSL dilakukan dengan menggunakan website https://www.sslshopper.com/ssl-checker.html. Hasil dari pemeriksaan SSL adalah sebagai berikut:

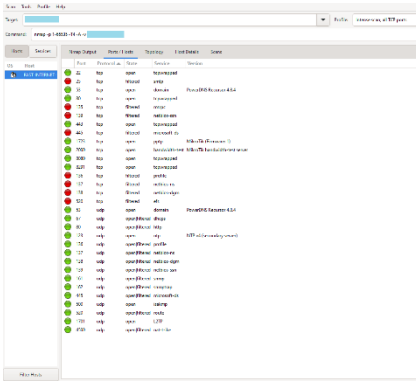
Sumber: Data Hasil Pengolahan (2023)

Gambar 9. SSL Scan

Gambar 9. menampilkan bahwa SSL yang dimiliki oleh aplikasi akademik berbasis website XYZ telah memiliki layanan SSL sehingga data yang diambil oleh pengakses bukan merupakan data yang asli. Melalui pemeriksaan ini diketahui bahwa layanan yang digunakan berasal dari pihak ke tiga, yaitu *Cloudflare*.

3.3.2 Vulnerable Analysis

Pada tahap *vulnerable Analysis* dilakukan pemeriksaan kerentanan menggunakan perangkat lunak Nmap, Shodan.io, dan OWASP ZAP. Pemeriksaan menggunakan Nmap ataupun Shodan.io adalah untuk mendeteksi apakah ada *internet security port* terbuka atau tertutup. Sementara penggunaan OWASP ZAP adalah untuk memeriksa apakah aplikasi akademik berbasis website XYZ memiliki kerentanan dengan tingkat risiko tertentu . Untuk lampiran pemeriksaan tekait *port protocol* sebagai berikut:



Sumber: Data Hasil Pengolahan (2023)

Gambar 10. Network Mapping Using Nmap

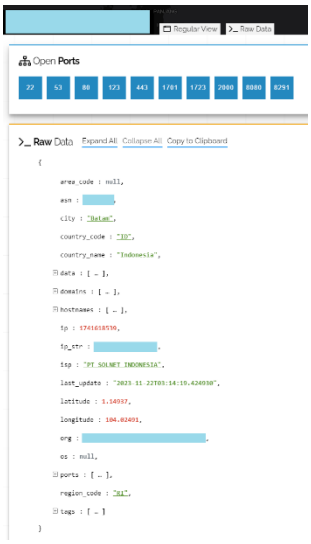
Gambar 10 menampilkan *protocol port* yang ada pada aplikasi akademik berbasis website XYZ. Terdapat delapan *port* yang tertutup, dua puluh tiga *port* terbuka termasuk *port* TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*) dan penggunaan perangkat keras Mikrotik yang pada umumnya digunakan sebagai perangkat jaringan internet. Hasil pemeriksaan *protocol port* ditunjukkan dalam tabel 3 sebagai berikut:

Tabel 3. Hasil Scanning Protocol Port

Protocol Port Terbuka	
TCP	22, 53, 80, 443, 1723, 2000, 8080, 8291
UDP	53, 67, 80, 123, 136, 137, 138, 139, 161, 162, 445, 500, 520, 1701, 4500
Protocol Port Tertutup	
TCP	25, 135, 139, 445, 136, 137, 138, 520

Sumber: Data Hasil Pengolahan (2023)

Untuk memastikan pencarian valid, digunakan layanan yang berbeda menggunakan Shodan.io. Untuk keterangan pencarian menggunakan Shodan.io adalah sebagai berikut:

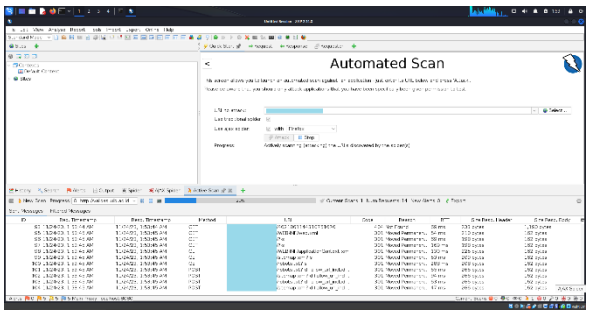


Sumber: Data Hasil Pengolahan (2023)

Gambar 11. Scanning Using Shodan.io

Pada Gambar 11. dibuktikan bahwa terdapat *protocol port* yang terbuka yaitu *port* TCP. *Port* UDP tidak terdeteksi karena untuk mencari *port* UDP memerlukan perintah khusus dan pencarian lebih mendalam. Penggunaan Shodan.io dapat menampilkan spesifikasi teknologi, lokasi ISP (*Internet Service Provider*), ASN (*Autonomous System Number*), dan versi dari teknologi yang digunakan oleh aplikasi akademik berbasis website XYZ. Fungsi dari Nmap dan Shodan.io pada dasarnya mirip. Perbedaannya Shodan.io dapat melacak perangkat jaringan dan lokasi, sedangkat penggunaan Nmap lebih ditujukan pada *mapping network*.

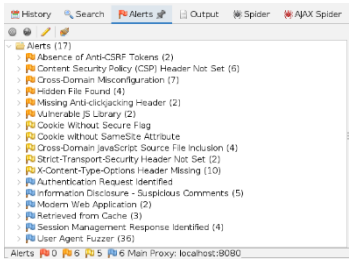
Setelah menemukan *protocol port* yang terbuka, dilakukan pemeriksaan fungsi dari aplikasi akademik berbasis website XYZ yang digunakan oleh mahasiswa. Kemudian aplikasi akademik berbasis website XYZ ini diperiksa kerentanannya menggunakan OWASP ZAP. Untuk keterangan pencarian kerentanan menggunakan OWASP ZAP adalah sebagai berikut:



Sumber: Data Hasil Pengolahan (2023)

Gambar 12. Scanning Memakai OWASP ZAP

Gambar 12menampilkan proses *scanning* menggunakan OWASP ZAP. Dengan menggunakan perangkat lunak ini, penelitian yang dilakukan dapat mengetahui kerentanan yang ada pada aplikasi akademik berbasis website XYZ. Untuk kategori kerentanan di dalam perangkat lunak OWASP ZAP terdapat empat jenis, yaitu level *high*, *medium*, *low*, dan *informational*. Level *high* merupakan kondisi kerentanan pada aplikasi berbasis website berada dalam fase serius, yang berarti dapat dieksploitasi dengan mudah. Pada kondisi ini, diperlukan tindak lanjut yang serius dalam memperbaiki celah kerentanan tersebut karena dampak yang akan diterima dapat berupa kehilangan data ataupun akses terhadap aplikasi berbasis website tersebut. Level *medium* merupakan kondisi kerentanan pada aplikasi berbasis website berada dalam fase waspada. Hal ini memungkinkan aplikasi berbasis website dapat terkena serangan melalui celah kerentanan yang ditemukan. Penyerangan yang dilakukan pada level *medium* memerlukan analisis kerentanan lebih lanjut untuk mengetahui apakah kerentanan dapat diserang dengan teknik khusus. Level *low* merupakan kondisi kerentanan pada aplikasi berbasis website yang berada dalam fase cukup aman. Dengan kemungkinan kecil untuk ditembus. Level *low* bisa menjadi sumber informasi penting untuk penyerangan pada level *high* atau *medium*. Level *informational* merupakan kondisi kerentanan aplikasi berbasis website yang berada pada fase aman. Pada umumnya, level ini hanya menunjukkan kemungkinan kesalahan dalam sistem pengodingan. Hasil dari *scanning* dapat dilihat pada lampiran berikut:



Sumber: Data Hasil Pengolahan (2023)

Gambar 13. Hasil Scanning OWASP ZAP

Gambar 13. menampilkan hasil *Scanning* kerentanan aplikasi akademik berbasis website XYZ dengan OWASP ZAP. Kerentanan yang dihasilkan dapat dikategorikan menjadi tiga level, yaitu *medium*, *low*, dan *informational*. Hasil pemeriksaan kerentanan dengan *tools* OWASP ZAP dapat dilihat pada tabel 4 berikut:

Tabel 4. Hasil Pemeriksaan Kerentanan Berdasarkan Tingkat Risiko

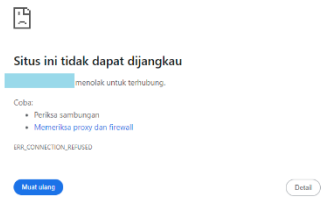
Tingkat Kerentanan	Jenis Kerentanan
Medium	*Content Security Policy (CSP) Header Not Set, Absence of Anti-CSRF Tokens,*

	Vulnerable JS Library, *Hidden File Found*, *Missing Anti-clickjacking Header*, dan *Cross-Domain Misconfiguration*.
Low	*Cookie Without SameSite Attribute*, *Strict-Transport-Security Header Not Set*, *X-Content-Type-Options Header Missing*, *Cross-Domain Java Script Source File Inclusion*, dan *Cookie Without Secure Flag*.
Informational	*Modern Web Application*, *User Agent Fuzzer*, *Retrieved from Cache*, *Authentication Request Identified*, *Session Management Response Identified*, dan *Information Disclosure – Suspicious Comments*.

Sumber: Hasil Penelitian (2023)

3.3.3 Exploiting

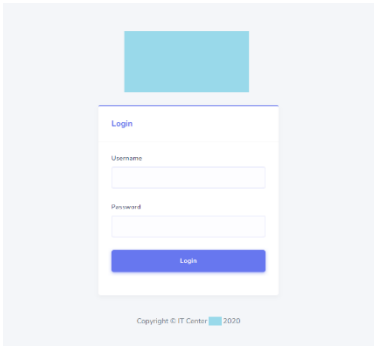
Pada tahap *exploiting*, dilakukan eksploitasi menggunakan Metasploit Framework dan Google Chrome. Pada proses eksploitasi, IP *Address* menjadi terblokir karena penyerangan yang dilakukan terlalu *massive*. Sebagai alternatif, digunakan *mobile hotspot* untuk melanjutkan proses ekploitasi. Berikut merupakan tampilan jaringan terblokir:



Sumber: Data Hasil Pengolahan (2023)

Gambar 14. Penggunaan Jaringan Wifi Yang Terblokir

Gambar 14. menunjukkan bahwa akses laptop ke aplikasi akademik berbasis website XYZ tidak terjangkau. Hal ini disebabkan setelah terjadi penyerangan secara beruntun atau signifikan. Penyerangan tersebut terblokir setelah melakukan *information gathering* hingga *vulnerable analysis*. Berikut tampilan setelah penggunaan jaringan *mobile hotspot*:

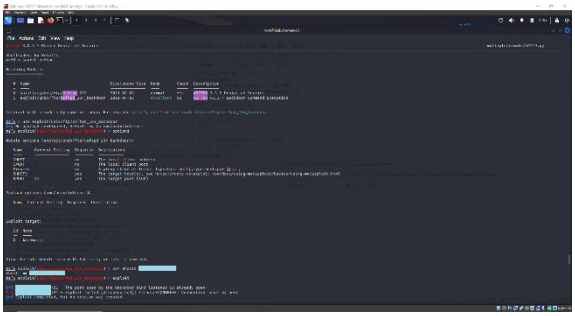


Sumber: Data Hasil Pengolahan (2023)

Gambar 15. Penggunaan Jaringan Mobile Hotspot

Gambar 15. menunjukkan bahwa akses laptop ke aplikasi akademik berbasis website XYZ dapat terhubung kembali. Penyebab dari terblokirnya jaringan wifi ialah aktivasi *firewall* yang memang dipasang untuk melindungi aplikasi akademik berbasis website XYZ. Pemblokiran tersebut hanya mencakup jaringan internet, namun tidak sampai memblokir *mac address*.

Kemudian penelitian dapat dilanjutkan dengan eksploitasi menggunakan Metasploit Framework dan Google Chrome. Eksploitasi pertama menggunakan perangkat lunak Metasploit Framework. Penggunaan perangkat lunak ini melalui sistem operasi Kali Linux dengan tampilan melalui terminal. Tampilan eksploitasi menggunakan Metasploit Framework adalah sebagai berikut:

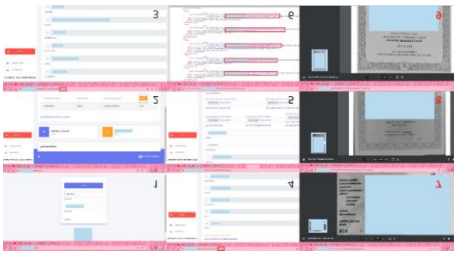


Sumber: Data Hasil Pengolahan (2023)

Gambar 16. Eksploitasi Dengan Metasploit Framework

Gambar 16menunjukkan penyerangan terhadap aplikasi akademik berbasis website XYZ dengan menghindari sistem deteksi dan pencegahan intrusi. Melalui eksploitasi ini, penyerangan tidak dapat tembus ke dalam *datastore* aplikasi akademik berbasis website XYZ. Hal ini disebabkan oleh peran *firewall* yang memblokir *traffic* jaringan mencurigakan yang ingin masuk ke sistem jaringan aplikasi akademik berbasis website XYZ.

Setelah tidak menemukan celah kerentanan pada Metasploit Framework, dilanjutkan dengan penggunaan Google Chrome. Tampilan penyerangan menggunakan Google Chrome adalah sebagai berikut:



Sumber: Data Hasil Pengolahan (2023)

Gambar 17. Eksploitasi Menggunakan Google Chrome

Gambar 17. menampilkan penyerangan dalam enam langkah pada aplikasi berbasis website XYZ yang digunakan sebagai tempat validasi data mahasiswa. Langkah pertama, memasuki aplikasi akademik berbasis website XYZ melalui akun salah satu mahasiswa ABC yang didapat. Langkah kedua, melakukan pemeriksaan pada bagian *button* ubah data. Langkah ketiga, memasuki bagian validasi ubah data untuk data mahasiswa yang akunnya digunakan untuk memasuki aplikasi akademik berbasis Website XYZ. Langkah keempat, melakukan teknik pengubahan *value* pada bagian *url*. Langkah kelima, memeriksa bagian data yang bisa diambil atau dicuri. Langkah keenam, memeriksa bagian *source code* dan menemukan bahwa ada fitur penyimpanan *file* penting seperti KTP, Ijazah, Akte Kelahiran, dan Kartu Keluarga. Hal ini membuktikan kerentanan yang ada pada aplikasi akademik XYZ berbasis website harus ditangani secepatnya. Berdasarkan eksploitasi ini, ditemukan sebanyak 5.582 mahasiswa XYZ yang dapat dicuri data privasinya.

3.4 Pelaporan

Setelah menemukan bagian kerentanan dan berhasil mendapatkan celah untuk menembus aplikasi akademik berbasis website XYZ, selanjutntya dilakukan analisis terkait faktor dan solusi terhadap kerentanan yang ditemukan pada aplikasi akademix berbasis website XYZ. Melalui Gambar 13 diketahui bahwa terdapat kerentanan yang terjadi pada aplikasi akademik berbasis website XYZ. Berikut penjelasan terkait kerentanan yang ada pada aplikasi akademik berbasis website XYZ ditunjukkan dalam Tabel 4:

Tabel 5. Rekapitulasi Kerentanan

Tingkat Kerentanan	Jumlah Peringatan
Level *High*	1
Level *Medium*	6
Level *Low*	5
Level *Informational*	6

Sumber: Data Hasil Pengolahan (2023)

Melalui proses analisis kerentanan, selanjutnya dilakukan perhitungan jumlah kerentanan yang harus diwaspadai sesuai dengan tingkat risiko dalam bentuk persentase. Pada tingkat risiko *high* mendapatkan nilai 5,55% yang ditemukan melalui proses *exploiting* dengan perangkat lunak Google Chrome. Pada tingkat risiko *medium*, *low*, dan *informational* didapatkan melalui proses *vulnerable analysis* dengan menggunakan OWASP ZAP. Untuk tingkat risiko *medium* mendapatkan nilai sebesar 33,3%. Untuk tingkat risiko *low* mendapatkan nilai sebesar 27,7%. Untuk tingkat risiko *informational* mendapatkan nilai sebesar 33,3%.

Analisis kerentanan top OWASP 2021 dan top OWASP API Security Risk 2023 yang didapatkan melalui proses pengujian kerentanan ditabulasikan pada Tabel 4, sebagai berikut:

Tabel 6. Analisa Hasil Kerentanan Top OWASP 2021 dan Top OWASP API Security Risk 2023

Kerentanan	Penyebab	Solusi
A01:2021 – Broken Access Control	1. Tidak terdapat token Anti-CSRF pada HTML. 2. Data *loading* pada aplikasi berbasis website yang disebabkan oleh kesalahan konfigurasi *Cross Origin Resource Sharing* (CORS). 3. Salah satu *cookie* disetel tidak menggunakan *SameSite attribute*. 4. Beberapa respon mengandung komen yang mendukung penyerang.	1. Pada fase desain dan arsitektur aplikasi berbasis website, gunakan sebuah *library* atau *framework* yang sudah terverifikasi. 2. Memastikan bahwa data yang sensitif tidak dapat diakses melalui proses tidak diautentikasi. Menghapus seluruh *header* CORS atau konfigurasi *header* HTTP "Access-Control-Allow-Origin" dengan kumpulan *domain* yang terisolasi aman. 3. Memastikan setelan *SameSite attribute* secara longgar atau memperketat semua *cookie*. 4. Menghapus komentar yang mengandung informasi privasi yang mendukung penyerang.
(A05:2021) Security Misconfiguration	1. Tidak menemukan *Content Security Policy* (CSP) pada *header* 2. *File* yang sensitive dapat diakses secara bebas. *File* tersebut berkemungkinan memberikan informasi terkait *administrative*, *configuration*, dan *credential*. 3. Pada halaman	1. Memastikan bahwa *application server*, *web server*, *load balancer*, dan lainnya terkonfigurasi menggunakan *Content-Security-Policy header*. 2. Mempertimbangkan apakah *file hidden* tersebut merupakan *file* yang digunakan. Jika tidak, sebaiknya dihilangkan atau dimatikan. Jika iya, pastikan akses pada *file* itu memiliki fitur *authentication* and

		aplikasi berbasis website tidak ada tanggapan yang memiliki *Content-Security-Policy dengan frame-ancestor* dan *X-Frame-Options* untuk mengatasi serangan *clickjacking*. 4. Setelan *cookie* tidak memiliki keamanan yang pasti, di mana *cookie* tersebut dapat di akses via *unencrypted connections*. 5. Tidak terdapat *Strict-Transport-Security* pada bagian *header*. 6. Pada bagian *Anti-MME-Sniffing header X-Content-Type-Options* tidak tersetel sebagai *nosniff*. Hali ini menyebabkan veri lama Internet Exploler atau Google Chrome dapat melakukan *MIME-Sniffing*.	*authorization* yang sesuai, atau membatasi aksesnya hanya untuk sistem internal atau IP khusus. 3. Mengaktifkan fitur *Content-Security-Policy* dan *X-Frame-Options* HTTP *header* pada semua halaman aplikasi berbasis website. 4. Memastikan bahwa *cookies* memiliki *secure flag*. 5. Memastikan *application server*, *web server*, *load balancer*, dan lainnya terkonfigurasi dapat melaksanakan *Strict-Transport-Security*. 6. Memastikan aplikasi berbasis website bahwa penyetelan *Content-Type header* dengan benar. Lalu setel *nosniff* pada fitur *X-Content-Type-Options* untuk semua halaman aplikasi berbasis website.eq2
(A06:2021) Vulnerable and Outdated Components	1. Teridentifikasi bahwa *library jquery* memiliki versi yang lama.	1. Melakukan *upgrade* pada versi *jquery* yang lama.
(A08:2021) Software and Data Integrity Failures	1. Pada halaman aplikasi berbasis website menampilkan satu atau lebih *script*	1. Memastikan bahwa sumber *file JavaScript* dimuat dari sumber terpercaya dan sumber tersebut tidak boleh
	file melalui *domain* pihak ketiga.	dikontrol oleh *end users* pada aplikasi berbasis website.
(API1:2023) Broken Object Level Authorization	1. Kesalahan konfigurasi API yang dapat menyebabkan *user* bisa mengakses *value* atau *ID* orang lain.	1. Melakukan konfigurasi API untuk memeriksa identitas yang sah dalam mengakses sebuah data.

Sumber: Data Hasil Pengolahan (2023)

IV. KESIMPULAN

Penelitian kerentanan yang dilakukan pada aplikasi akademik berbasis website XYZ dengan metode OWASP berhasil menemukan kerentanan yang harus segera ditindaklanjuti. Penelitian ini melalui empat tahap pengujian kerentanan dengan menggunakan tujuh *tools* atau perangkat lunak. Pada tahap information gathering ditemukan bahwa aplikasi akademik berbasis website XYZ menggunakan layanan pihak ketiga berupa *cloudflare.* Selanjut pada *vulnerable analysis* ditemukan delapan *port* yang tertutup, dua puluh tiga *port* terbuka termasuk *port* TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*) serta informasi penggunaan perangkat keras Mikrotik sebagai perangkat jaringan internet. Pada tahap *vulnerable analysis* ini, ditemukan 17 kerentanan dengan enam level *medium*, lima level *low*, dan enam level *informational*. Pada langkah selanjutnya dilakukan eksploitasi dengan Google Chrome dan ditemukan kerentanan level *high* dalam bentuk broken object level authorization. Dari total kerentanan yang ditemukan, terdapat 5,55% kategori risiko high, 33,3% risiko medium, 27,7% resiko low dan 33,3% bersifat informational. Aplikasi akademik berbasis website XYZ masuk dalam kategori tidak aman dari serangan *hacker*. Terdapat satu masalah yang wajib segera diatasi, berupa kerentanan API1:2023 – Broken Object Level Authorization yang dapat menyebabkan pencurian data privasi. Pada penelitian selanjutnya dapat dilakukan identifikasi serangan dengan menggunakan *tools* lain yang tidak mudah terdeteksi secara cepat oleh *firewall.* Hal ini penting supaya pengelola sistem dapat mendesain sistem keamanan yang dapat menangkal bentuk serangan yang menggunakan berbagai tools yang berbeda.

V. REFERENSI

Ariyadi, T., Widodo, T. L., Apriyanti, N., & Kirana, F. S. (2023). Analisis Kerentanan Keamanan

Sistem Informasi Akademik Universitas Bina Darma Menggunakan OWASP. *Techno.Com*, *22*(2), 418–429. https://doi.org/10.33633/tc.v22i2.7562

Ashari, I. F. A., Affandi, M., Putra, H. T., & Nur, M. T. (2023). Security Audit for Vulnerability Detection and Mitigation of UPT Integrated Laboratory (ILab) ITERA Website Based on OWASP Zed Attack Proxy (ZAP). *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, *7*(1), 24–34. https://doi.org/10.35870/jtik.v7i1.657

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

Asmawi, Syafei, & Yamin, M. (2019). Pendidikan Berbasis Teknologi Informasi Dan Komunikasi. *Prosiding Seminar Nasional Pendidikan*, *3*, 50–55.

Huda, I. A. (2020). Perkembangan Teknologi Informasi Dan Komunikasi (Tik) Terhadap Kualitas Pembelajaran Di Sekolah Dasar. *Jurnal Pendidikan Dan Konseling (JPDK)*, *2*(1), 121–125. https://doi.org/10.31004/jpdk.v1i2.622

Idris, M., Syarif, I., & Winarno, I. (2022). Web Application Security Education Platform Based on OWASP API Security Project. *EMITTER International Journal of Engineering Technology*, *10*(2), 246–261. https://doi.org/10.24003/emitter.v10i2.705

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

Kusuma, G. (2022). Implementasi Owasp Zap Untuk Pengujian Keamanan Sistem Informasi Akademik. *Jurnal Teknologi Informasi: Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, *16*(2), 178–186. https://doi.org/10.47111/jti.v16i2.3995

Mu'min, M. A., Fadlil, A., & Riadi, I. (2022). Analisis Keamanan Sistem Informasi Akademik Menggunakan Open Web Application Security Project Framework. *Jurnal Media Informatika Budidarma*, *6*(3), 1468. https://doi.org/10.30865/mib.v6i3.4099

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

Priyawati, D., Rokhmah, S., & Utomo, I. C. (2022). Website Vulnerability Testing and Analysis of Internet Management Information System Using OWASP. *International Journal of Computer and Information System (IJCIS) Peer Reviewed-International Journal*, *3*(3), 143–147. https://doi.org/10.29040/ijcis.v3i3.90

Putra, W. A., Fitri, I., & Hidayatullah, D. (2022). Implementasi Waterfall dan Agile dalam Perancangan E-Commerce Alat Musik Berbasis Website. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, *6*(1), 56–62. https://doi.org/10.35870/jtik.v6i1.380

Riadi, I., Umar, R., & Lestari, T. (2020). Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *5*(3), 146–152. https://doi.org/10.14421/jiska.2020.53-02

Sahren. (2021). Implementasi SSL Untuk Pencegahan Man in The Middle Attack Pada FTP Server. *Journal of Science and Social Research*, *4*(1), 28–33. http://jurnal.goretanpena.com/index.php/JSSR

Smith, L. A., Chowdhury, M., & Latif, S. (2022). Ethical hacking: Skills to fight cybersecurity threats. *EPiC Series in Computing*, *82*, 102–111. https://doi.org/10.29007/vwww

Ul Haq, H. B., Hassan, M. Z., Hussain, M. Z., Khan, R. A., Nawaz, S., Khokhar, H. R., & Arshad, M. (2022). The Impacts of Ethical Hacking and its Security Mechanisms. *Pakistan Journal of Engineering and Technology*, *5*(4), 29–35. https://doi.org/10.51846/vol5iss4pp29-35

Wen, S. F., & Katt, B. (2023). A quantitative security evaluation and analysis model for web applications based on OWASP application security verification standard. *Computers and Security*, *135*(October), 103532. https://doi.org/10.1016/j.cose.2023.103532