Cyberspace: Jurnal Pendidikan Teknologi informasi
Volume 9, Nomor 1, Maret 2025, hal. 27 - 41
ISSN 2598-2079 (print) | ISSN 2597-9671 (online)

IMLEMENTASI PENETRATION TESTING PADA APLIKASI WEB SISTEM EVALUASI DATA BIDANG TIK POLDA ACEH MENGGUNAKAN METODE OWASP DAN NIST SP 800-115

[1]Syahril Handaya, [2]Raihan Islamadina

[1,2]Pendidikan Teknologi Informasi, Fakultas Tarbiyah dan Keguruan, Universitas Islam Negeri Ar-Raniry, Jl. Syeikh Abdul Rauf Darussalam, Banda Aceh 23111, Indonesia
Email: 210212050@student.ar-raniry.ac.id

Abstract

One factor that must be taken into account is information system security, which is comprised of three components confidentiality, integrity, and availability. As a result, in order to preserve information security, an information system must be evaluated through penetration testing, which attempts to strengthen the system's security. In this context, the Aceh Regional Police's Information and Communication Technology department has created an information system in the form of a Web-based application called the Data Evaluation System Web Application. However, in order to transmit the security site, the web application's security must be examined, which calls for penetration testing. The Aceh Police Information and Communication Technology Data Evaluation System Web Application was subjected to penetration testing using the National Institute of Standards Technology Securely Provision 800-115 (NIST SP 800-115) method and the Open Web Applications Security Project (OWASP) method. Specifically for the National Institute of Standards Technology Securely Provision 800-115 (NIST SP 800-115) method, these two approaches employ the same information gathering technique, the Black Box technique, along with additional steps, such as vulnerability scanning and the attack phase. Numerous vulnerabilities in the web application were discovered as a consequence of the penetration testing, and these reports were used to strengthen the information system's security.

Keywords: *Penetration Testing, OWASP, NIST 800-115, Information Systems.*

Abstrak

Keamanan sistem informasi merupakan aspek yang perlu diperhatikan. Keamanan informasi terdiri dari 3 aspek yaitu Kerahasiaan, Integritas, Ketersediaan maka dari itu untuk menjaga keamanan informasi tersebut perlu dilakukannya evaluasi dari suatu sistem informasi berupa pengujian penetrasi. Pengujian penetrasi ini bertujuan untuk meningkatkan keamanan sitem informasi. Barkaitan dengan hal tersebut bidang teknologi Informasi dan Komunikasi Polda Aceh telah merancang sebuah sistem informasi berupa Aplikasi berbasis Web yaitu Aplikasi Web Sistem Evaluasi Data. Namun keamanan dari Aplikasi Wseb tersebut perlu diuji untuk mengevaluasi sitem keamananya, maka dari itu perlu dilakukan pengujian penetrasi pada Aplikasi Web tersebut. Pengujian penetrasi pada Aplikasi Web Sistem Evaluasi Data Bidang Teknologi Informasi dan Komunikasi Polda Aceh ini dengan menggunakan metode *Open Web Aplications Scurity Project* (OWASP) dan metode *National Institut Standart Technology Securely Provision* 800-115 (NIST SP 800-115). Didalam proses kedua metode ini menggunakan teknik yang sama dalam *Information Gathring* yaitu teknik *Black Box* dan terdapat tahapan lainnya yaitu *Vulnerabelity*

Scanning serta Fase Attack khusus untuk metode *National Institut Standart Technology Securely Provision* 800-115 (NIST SP 800-115). Dari hasil pengujian penetrasi tersebut berhasil ditemukannya beberapa kerentanan pada aplikasi web tersebut dan dijadikan pelaporan untuk meningkatkan sistem keamanan pada sistem informasi tersebut.

Kata Kunci: Pegujian Penetrasi, OWASP, NIST 800-115, Sistem Informasi.

1. Pendahuluan

Penggunaan Internet telah merambah hampir ke semua sektor pekerjaan, termasuk pada bidang Pendidikan, Perkantoran, dan Industri. Ketergantungan manusia pada fasilitas internet yang terus berkembang ini sangatlah besar, terutama di era perkembangan IPTEK saat ini. Perkembangan IPTEK yang sangat pesat telah mengakibatkan lonjakan akses terhadap informasi digital, didukung oleh infrastruktur jaringan internet yang sangat cepat. Di lain sisi, keunggulan dari IPTEK ini juga sangat membawa ancaman terhadap keamanan suatu sistem informasi , yang dapat mengakibatkan kejahatan *Cyber*.[1] Oleh karena itu suatu sistem informasi perlu diperhatikan sistem keamanannya.

Keamanan Informasi terdiri dari 3 aspek yaitu kerahasian, Integritas dan Ketersediaan. Untuk mengukur sejauh mana tingkat keamanan sistem informasi tersebut perlu dilakukan pengujian berupa pengujian penetrasi.[2] Terdapat beberapa metode dalam pengujian penetrasi tersebut. Pengujian penetrasi berguna untuk mengevaluasi keamanan aplikasi web, memastikan area yang berpotensi rentan telah diperiksa, dan diberikan rekomendasi yang relevan untuk meningkatkan keamanan suatu sistem informasi. [3]

Berhubungan dengan hal tersebut Polda Aceh merupakan suatu instansi pemerintah yang berada di bawah naungan Kepolisian Negara Republik Indonesia dan Polda Aceh memiliki peran penting dalam menjaga keberlangsungan kesatuan negara Republik Indonesia khususnya di wilayah Aceh, Polda Aceh sendiri memiliki banyak personil-personilnya yang ditempatkan baik di satuan wilayah (Polres) maupun satuan kerja (Satker) dan memiliki perannya masing-masing. Dari hal tersebut Bidang Teknologi Informasi dan Komuikasi Polda Aceh membangun sebuah sistem informasi berupa Aplikasi berbasis Web yang bertujuan untuk memonitor kinerja baik di satuan wilayah maupun di satuan kerja, Aplikasi berbasis Web ini Bernama “Sistem Evaluasi Data” . Namun Aplikasi Web ini masih perlu dievaluasi sistem keamanannya.[4]

Oleh karena itu, Pada penelitian ini peneliti melakukan pengujian penetrasi tersebut pada suatu sistem informasi yaitu Aplikasi Web Sistem Evaluasi Data Bidang Teknologi Informasi dan Komuikasi Polda Aceh untuk mengevaluasi sistem keamanan pada sistem informasi tersebut dengan menggunkan metode *Open Web Aplication Security Project* (OWASP) dan metode *National Institut Standart Technology Securely Provision* 800-115 (NIST 800-115).[5]

2. Kajian Pustaka

2.1 Penetration Testing

Pentesting, juga dikenal sebagai pengujian penetrasi, adalah teknik yang digunakan untuk menilai risiko kemungkinan pelanggaran keamanan dengan mensimulasikan serangan sebenarnya. Penguji tidak hanya mengidentifikasi kerentanan yang dapat dieksploitasi oleh penyerang, namun mereka juga memanfaatkan kerentanan tersebut untuk menentukan kemungkinan konsekuensi dari keberhasilan eksploitasi.[6]

Salah satu teknik untuk memperingatkan keamanan sistem komputer atau jaringan tentang kemungkinan serangan peretasan adalah pengujian penetrasi. Langkah ini sangat

penting dalam mengembangkan pertahanan yang kuat untuk server komputer yang terhubung dalam jaringan, karena tidak hanya menilai operasi tetapi juga implementasi dan desain sistem. Pengujian penetrasi dilakukan secara resmi dan terjadwal, yang membedakan mereka dari penyerang, dan meskipun penting, tidak diadopsi secara luas oleh berbagai organisasi dan institusi.[7]

2.2 OWASP

Sebuah kelompok nirlaba bernama OWASP berdedikasi untuk meningkatkan keamanan perangkat lunak dan aplikasi web. Aplikasi *open source* OWASP ZAP (Zed Attack Proxy) dikembangkan oleh grup OWASP untuk mengevaluasi keamanan aplikasi web.[8] Untuk menemukan dan memperbaiki kelemahan keamanan dalam aplikasi web, alat ini menyertakan fitur penting seperti SQL Injection, Cross Site Scripting (XSS), dan Cross Site Request Forgery (CSRF).[9]

Pengguna juga dapat melakukan eksplorasi sistem dan fuzzing manual dengan OWASP ZAP.[10] Antarmuka pengguna grafis (GUI) alat ini memudahkan pengembang dan peneliti keamanan untuk menggunakannya. Selain itu, OWASP ZAP dapat berkomunikasi dengan prosedur pengujian keamanan terkini dengan mendukung otomatisasi melalui baris perintah[11].

2.3 NIST SP 800-115

NIST menciptakan standar sebagai standar pengembangan untuk menjamin kualitas dan kesesuaian barang dan jasa. Hal ini mencakup berbagai bidang teknis, seperti keamanan siber dan standar pengukuran.[12] NIST melakukan penelitian dan pengembangan ilmu pengukuran untuk menjamin presisi dan keseragaman secara global. NIST menciptakan standar dan pedoman untuk melindungi data dan sistem komputer dari intrusi dan serangan.[13] NIST menciptakan kerangka keamanan siber yang membantu bisnis mengelola risiko keamanan siber dengan cara yang terorganisir dan metodis. Untuk memastikan fasilitas pengujian dan kalibrasi mematuhi persyaratan yang ketat, NIST menjalankan program akreditasi. NIST menerbitkan standar keamanan informasi dan teknologi, seperti seri NIST SP 800. [14]

NIST melakukan penelitian ilmiah dan teknologi dibidang ilmu material, nanoteknologi, dankomuikasi untuk mendorong inovasi dan pengembangan. NIST mengembangkan dan mempromosikan teknologi dan standar dalam kemitraan dengan bisnis, akademisi, dan organisasi pemerintah lainnya.[15] Selain itu, NIST menawarkan materi pengajaran dan pelatihan tentang berbagai mata pelajaran teknis.[16] NIST telah memberikan dampak signifikan pada berbagai industri, termasuk manufaktur, teknologi informasi, kesehatan, dan energi, dengan mendorong inovasi, efisiensi, dan keamanan.[17]

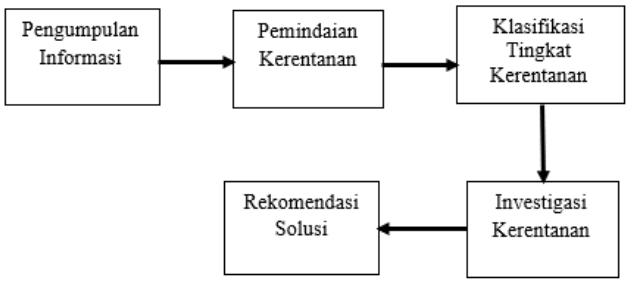
3. Metode Penelitian

3.1 Sistem Operasi

TABLE 1. *SPESIFIKASI HADWARE*

Komponen	
Processor	13th Gen Intel(R) Core(TM) i9-13900H (20CPUs), ~2.6GHz
Random Acces memory (RAM)	32 GB
Storage Memory	1,5 TB
Vidio Graphics Array (GVA)	GVA 0: Intel(R) Iris® Xe Graphics, GVA 1: NVIDIA GeForce RTX 4050
Operating System (OS)	Windows 11 Home Sinle Languange 64-bit (10.0, Build 22631)

3.2 Rancangan Penelitian Metode OWASP



Gambar 1. *Flowchart* Metode OWASP.

Untuk mengubah data OWASP menjadi wawasan berharga, berikut tahapan yang dapat dilakukan:

1. Pengumpulan Informasi *(Information Gathring)*
   Megumpulkan segala informasi terkait aplikasi web yang akan dianalisis dengan menggunakan teknik Black Box.
2. Pemindaian Kerentanan *(Vulnerability Scanning)*
   Melakukan analisis mendalam pada data OWASP untuk mengidentifikasi celah keamanan yang mungkin ada dalam aplikasi web.
3. Klasifikasi tingkat kerentanan
   Mengelompokkan dan mengurutkan kerentanan berdasarkan tingkat bahayanya. Kerentanan yang paling berpotensi merugikan harus menjadi prioritas utama
4. Investigasi kerentanan
   Mendeskripsikan atau menjelaskan kerentanan yang ditemukan pada saat *Vulnerability Scanning*
5. Rekomendasi Solusi.
   Menindaklanjuti dengan membuat laporan berupa Solusi untuk perbaikan pada sistem. Hal ini bisa berupa pembaruan perangkat lunak, penyesuaian konfigurasi, perbaikan kode,

atau penguatan kebijakan keamanan. Berikut Komponen yang digunakan pada metode OWASP.

TABLE 2. *SPESIFIKASI SOFTWARE* METODE OWASP

Komponen	Komponen	Versi
Dual OS	Virtual Box	7.1.4
	Kali Linux	2024.3
Information Gathering	Ping	-
	Whois	5.5.23
	SSLScan	1.0.2
Vulnerability Scanning	Nmap	7.94
	Zap	2.15.0
Reporting	Word	2021

3.3 Rancangan Penelitian Metode NIST SP 800-115

Gambar 2. *Flowchart* Metode NIST SP 800-115.

Metode NIST SP 800-115 memiliki beberapa tahapan diantaranya sebagai berikut:

1. Fase *Planning*

   Pada tahap Perencanaan, Anda akan merencanakan pengujian yang akan dilakukan. Beberapa hal yang termasuk dalam perencanaan ini adalah penentuan sasaran, tujuan, ruang lingkup, dan metode yang digunakan dalam pengujian.

2. Fase *Discovery*

   Penulis mengumpulkan dan memeriksa data mengenai tujuan tes sepanjang langkah ini. Dalam Fase Serangan Kali Linux, teknik *Black Box* digunakan untuk pengumpulan dan pemindaian.

3. Fase *Attack*

   Sampel penelitian dilakukan penulis pada fase *Attack*. Setelah serangan dilakukan, kerentanan akan dieksploitasi menggunakan alat seperti Metasploit untuk mengonfirmasi temuan.

4. Fase *Reporting*

   Tiga fase sebelumnya diselesaikan bersamaan dengan fase pelaporan. Strategi penilaian dikembangkan selama tahap perencanaan. *Log* tertulis kemudian disimpan selama tahap Penemuan dan Serangan, dan administrator serta manajemen sistem menerima laporan secara teratur. Pendekatan NIST SP 800-115 memanfaatkan komponen berikut.

TABLE 3. *SPESIFIKASI SOFTWARE* METODE NIST SP 800-115

Komponen	Komponen	Versi
Dual OS	Virtual Box	7.1.4
	Kali Linux	2024.3
Information Gathering	Ping	-
	Whois	5.5.23
	SSLScan	1.0.2
Vulnerability Scanning	NMap	7.94
	Zap	2.15.0
Attack	Hydra	9.5
	Metasploit	6.4.34
	Wireshark	4.4.2
Reporting	Microsoft Word	2021

4. Hasil dan Pembahasan

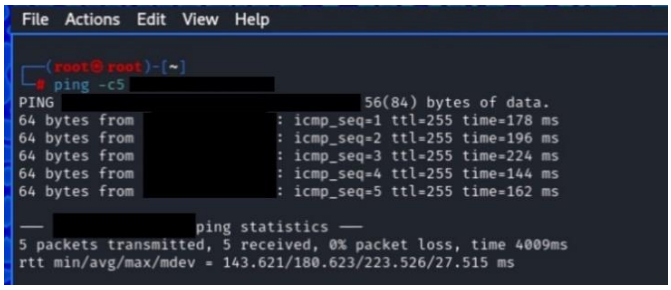
4.1 *Planning*

Pada tahapan ini membahas mengenai ruang lingkup penelitian, *hardware* yang digunakan pada penelitian, *software* yang digunakan pada peelitian serta teknik yang digunakan pada saat penelitian.

4.2 *Discovery*

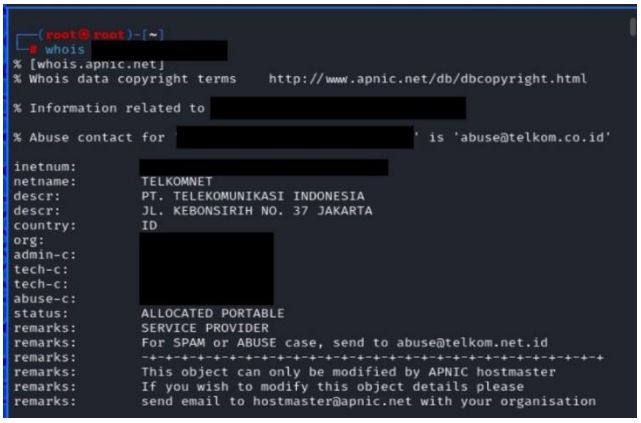
4.2.1 *Information Gathring*

Pemahaman Data pada metode ini maksudnya yaitu untuk menggali informasi dari sample penelitian, dalam hal ini menggunakan teknik Black box. Langkah yang pertama yaitu dengan melakukan perintah **ping -c5** “Ip Adrees” untuk melihat bagaimana kecepatan sistem informasi tersbut dalam mengirim paket data, seperti pada gambar 3.



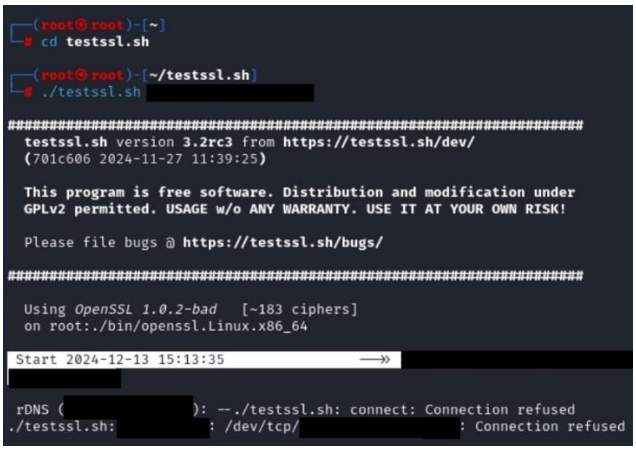
Gambar 3. Hasil nmap pada DNS.

Waktu adalah waktu respons host dalam milidetik (ms), hasil *ping* yang bagus adalah kurang dari 100 ms. Time to live (TTL) adalah lamanya suatu paket data pada jaringan dalam satuan detik. Kemudian seperti yang terlihat pada gambar di bawah, jalankan perintah ***whois*** "Alamat IP" untuk melihat informasi sistem yang lebih spesifik, seperti pada gambar 4.



Gambar 4. Hasil *Whois* Pada *IP Address.*

Dari hasil perintah tersebut berhasil menunjukkan informasi detail mnegenai aplikasi tersebut, perintah selanjutnnya yang dilakukan adalah **cd testssl.sh** lalu perintah **./testssl.sh** "*IP Adress*", Seperti pada gambar 5.

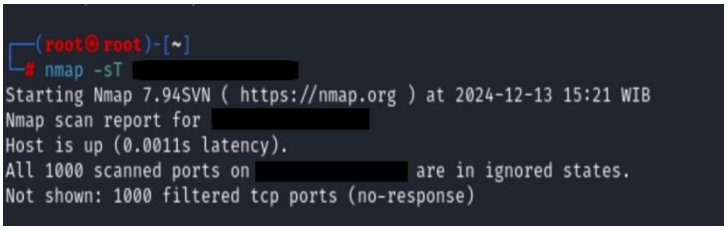


Gambar 5. Hasil *Testssl* Pada *IP Addres.*

Dari gambar tersebut menyatakan bahwa tidak terdapatnya ssl pada aplikasi web tersebut, ssl adalah standar keamanan jaringan dari suatu sistem informasi.

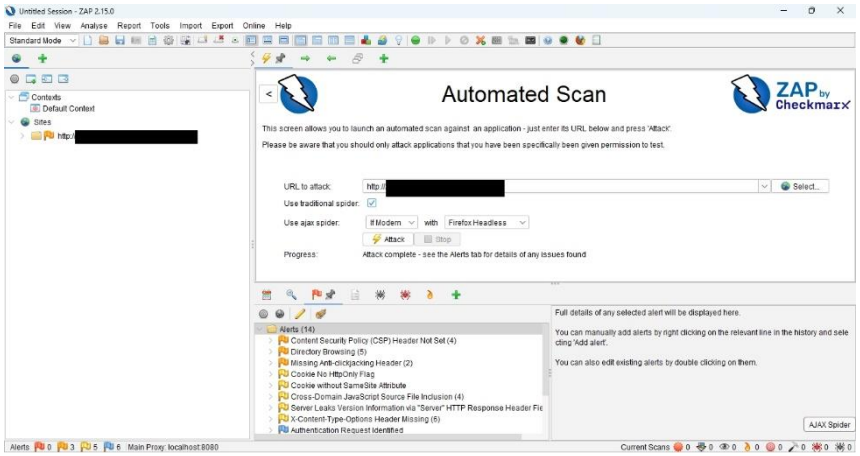
4.2.2 *Vulnerability Scanning*

Pada titik ini, nmap digunakan untuk memindai program. Di sini, perintah **nmap -sT** *"IP Address*" digunakan untuk melihat port aplikasi web yang terbuka, seperti yang ditunjukkan pada gambar di bawah.



Gambar 6. Hasil *nmap* Pada *IP Addres.*

Dari hasil Scanning manggunkan tools nmap tersebut menyatakan bahwa tidak ada ports pada server yang terbuka. Dan langkah selanjutya yaitu melakukan *scanning* menggunkan *framework* Zap, seperti gambar di bawah ini.



Gambar 7. Proses *Scanning* menggunakan Zap.

Dari hasil scanning menggunakan *framework* OWASP Zap berhasil ditemukannya beberapa kerentanan yang terdapat pada aplikasi web tersebut dan dari hasil *scanning* tersebut menunjukan beberapa tingkatat kerentanan, berikut meupakan hasil *scanning* menggunkan *framework* OWASP Zap.

TABLE 4. HASIL PROSES SCANNING BERDASARKAN *LEVEL*

Risk Level	Number of Allert
High	0
Medium	3
Low	5
Informational	6

Berikut merupakan detail dari penjelasan *Scanning* menggunakan *framework* Zap.

TABLE 5. HASIL *VULENERABILITY SCANNING* ZAP.

No	*Allert*	*Description*	*Risk level*	*Solution*
1	*Content Security Policy (CSP) Header Not Set*	Serangan ini dapat digunakan untuk segala hal mulai dari penyebaran virus hingga perusakan situs atau pencurian data.	*Medium*	Verifikasi bahwa *Header* Kebijakan Keamanan Konten telah diatur pada penyeimbang beban, server web, server aplikasi, dll.
2	*Directory Browsing*	Daftar direktori dapat mengungkapkan skrip tersembunyi, termasuk file, file sumber cadangan, dll. yang dapat diakses untuk membaca informasi sensitif.	*Medium*	Nonaktifkan penjelajahan direktori. Jika perlu, pastikan file yang tercantum tidak menimbulkan risiko.

3	*Missing Anti-clickjacking Header*	Serangan 'ClickJacking' tidak dapat dilindungi. Ini karena tidak ada konfigurasi pada Kebijakan Keamanan Konten dengan arahan *'frame-ancestors'* atau *X-Frame-Options.*	*Medium*	Header HTTP X-Frame-Options dan Content-Security-*Policy* didukung oleh browser web kontemporer. Pastikan setiap halaman web yang ditampilkan situs atau aplikasi telah menetapkan salah satu halaman tersebut.Sebagai alternatif, pertimbangkan untuk menerapkan arahan *"frame-ancestor"* dari Kebijakan Keamanan Konten.
4	*Cookie No Http Only Flag*	JavaScript dapat mengakses cookie karena telah disetel tanpa *flag HttpOnly.* Cookie dapat diakses dan ditransfer ke situs web lain jika skrip berbahaya dapat dijalankan di halaman ini.	*Low*	Verifikasi bahwa setiap cookie memiliki pengaturan Http *Only* yang disetel.
5	*Cookie without SameSite Attribute*	Permintaan 'lintas situs' dapat mengakibatkan cookie dikirimkan karena telah disetel tanpa atribut SameSite.	*Low*	Pastikan semua cookie memiliki atribut SameSite yang disetel ke 'longgar' atau 'ketat'.
6	*Cross-Domain JavaScript Source File Inclusion*	Terdapat halaman yang berisi satu atau lebih file skrip dari situs web eksternal.	*Low*	Verifikasi bahwa hanya sumber terpercaya yang digunakan untuk memuat file sumber JavaScript, dan pengguna akhir aplikasi tidak memiliki kendali atas sumber tersebut.
7	*Server Leaks Version Information via "Server" HTTP Response Header Field*	Informasi versi dibocorkan oleh server web/aplikasi melalui *header respons* HTTP "Server". Penyerang mungkin akan lebih mudah menemukan lebih banyak kelemahan di server web/aplikasi jika penyerang memiliki akses ke informasi ini.	*Low*	Verifikasi bahwa header Kebijakan Keamanan Konten telah diatur pada penyeimbang beban, server web, server aplikasi, dll.
8	*X-Content-Type-Options Header Missing*	Opsi Tipe Konten Anti-MIME *Sniffing* Title X tidak dikonfigurasi. Oleh karena itu, sniffing MIME dapat mengakibatkan isi respons dibaca dan	*Low*	Pastikan server atau aplikasi web memuat *header X-Content-Type-Options* ke *'nosniff'* untuk setiap halaman web dan

		disajikan sebagai jenis konten yang berbeda dari yang ditentukan di versi Chrome dan Internet *Explorer* yang lebih lama.		memuat header Tipe Konten dengan benar.
9	*Authentication Request Identified*	Kumpulan baris *key=value* yang ditemukan di bidang 'Info Lainnya'. Yang di indentifikasi adalah permintaan otentikasi.	*Informational*	Tidak ada yang perlu diperbaiki karena ini hanyalah pemberitahuan dan bukan kerentanan.
10	*GET for POST*	*GET* juga diterima untuk permintaan yang awalnya dianggap sebagai *POST.* Meskipun masalah ini bukan merupakan kelemahan keamanan, hal ini mungkin membuat serangan lain lebih mudah dilakukan. Misalnya, penelitian ini mungkin menyarankan bahwa XSS yang lebih sederhana (berbasis GET) juga layak dilakukan jika *Critical POST* awal rentan terhadap *Critical Critical oss-Site Scripting (XSS).*	*Informational*	Pastikan bahwa ketika *POST* diharapkan, hanya *POST* yang diterima.
11	*Information Disclosure - Suspicious Comments*	Pernyataan mencurigakan yang dapat membantu penyerang tampaknya disertakan dalam tanggapan. Catatan: Konten secara keseluruhan, bukan hanya komentar, bertentangan dengan kecocokan yang dibuat dalam blok skrip atau file.	*Informational*	Hilangkan komentar apa pun yang memberikan informasi yang dapat membantu penyerang dalam mengidentifikasi dan memperbaiki sumber masalah yang mereka maksudkan.
12	*Modern Web Application*	Hapus semua komentar yang memberikan detail yang mungkin membantu penyerang dan mengatasi masalah mendasar yang mereka ajukan.	*Informational*	Tidak ada yang perlu diubah, ini hanyalah pemberitahuan informasi.
13	*Session Management Response Identified*	Telah ditentukan bahwa respons yang diberikan berisi token manajemen sesi. Kumpulan token header yang berlaku untuk Metode Manajemen Sesi	*Informational*	Tidak ada yang perlu diperbaiki karena ini hanyalah pemberitahuan dan bukan kerentanan.

		Berbasis Header terdapat di bidang 'Info Lainnya'.		
14	*User Agent Fuzzer*	Variasi tanggapan tergantung pada Agen Pengguna yang tidak jelas (seperti crawler mesin pencari atau situs web seluler). Bandingkan kode hash dan kode status respons dengan respons awal.	*Informational*	Gunakan *Monitor Header* HTTP, Analisis Log Server, dan Validasi Agen Pengguna.

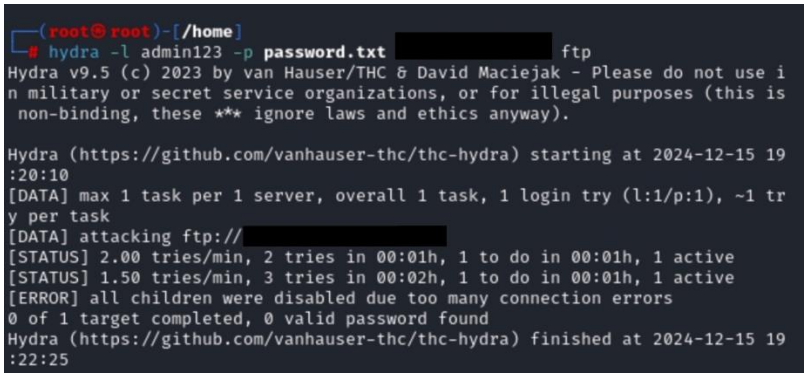
Dari hasil *Scanning* menggunakna *framework* Zap tersebut sebagian besar kerentanan yang ditemukan disebabkan karena kesalahan dalam proses pemrograman atau *insicure design* yang berdampak pada keamanan website tersebut, hal ini bisa terjadi karena kurangnya kesadaran tentang keamanan selama fase desain.

4.3 *Attack*

Langkah utama metode NIST SP 800-115 untuk melakukan pengujian petrasi pada hasil yang diperoleh melalui Discovery adalah langkah ini. Berikut serangan yang bisa digunakan:

4.3.1 *Brute For Attack*

Saat mencoba melakukan serangan *brute force*, Anda harus memberikan tebakan acak untuk sesi pencarian nama pengguna dan kata sandi yang berbeda. Gunakan Hydra pada saat ini, lalu jalankan perintah seperti yang terlihat pada gambar 8.



Gambar 8. Serangan *Brute For Attack.*

Tidak ada hasil sah yang dicapai, seperti yang ditunjukkan oleh hasil tes *Brute for Attack*, yang menunjukkan 0 kata sandi valid diambil. Oleh karena itu, menggunakan SSH untuk menguji *login brute force* tidaklah rentan.

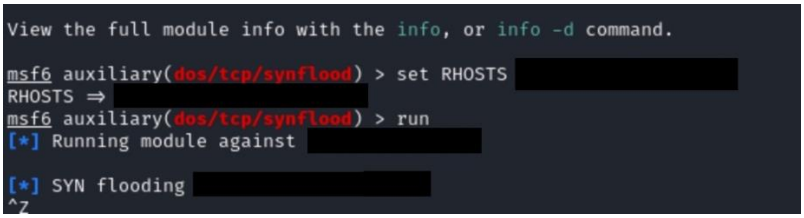
4.3.2 *Denial of Service Synflood*

Gunakan eksploitasi *Synflood* DoS untuk memanfaatkan *Denial of Service* (DoS). DoS Synflood adalah serangan di mana jaringan kelebihan beban dengan lalu lintas fiktif. Artinya jaringan atau server yang diserang tidak akan mampu menyuplai trafik sehingga menyebabkan sistem rusak dan tidak dapat beroperasi secara efektif. Menggunakan perintah, kemudian menggunakan ux/dos/tcp/synflood, dan terakhir menampilkan opsi, pengujian ini memanfaatkan kerangka Metasploit untuk

mensimulasikan serangan *Synflood* DoS dan menganalisis detail lalu lintas jaringan selama proses eksploitasi DoS menggunakan *Wirshark.*

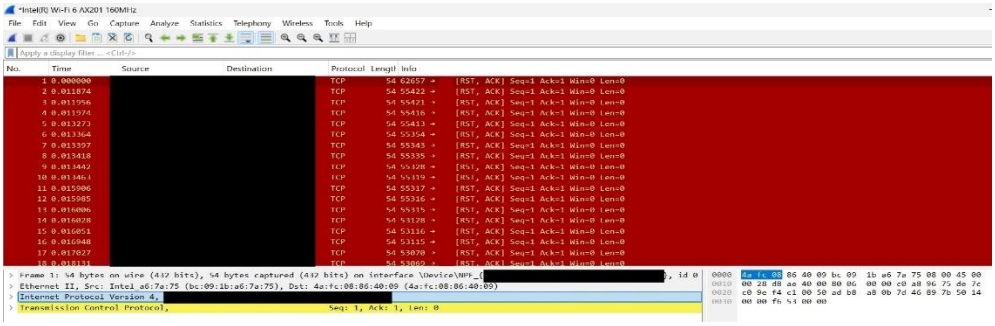
Gambar 9. Modul serangan *Denial of Service Synflood*

Selanjutnya konfigurasikan menggunakan perintah yang tersedia pada modul seperti pada gambar di bawah.



Gambar 10. Serangan *Denial of Service Synflood.*

Serangan telah diimplementasikan seperti yang ditunjukkan pada gambar 10. Selanjutnya *Wireshark* diperlukan untuk mencatat aktivitas paket data grafik di seperti pada gambar 11.

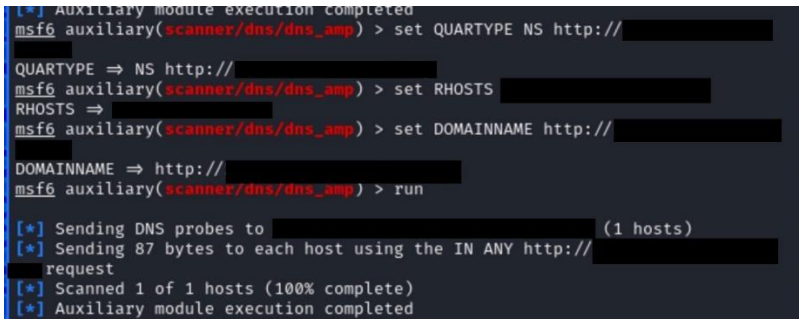


Gambar 11. Serangan *Denial of Service Synflood.*

Tampilan *Wireshark* menunjukkan bahwa sistem sedang sibuk karena lalu lintas jaringan yang sangat padat. TCP dapat dieksploitasi dengan mengirimkan paket SYN dan memalsukan alamat IP, yang menyebabkan server mengklarifikasi koneksi namun tidak pernah membuatnya. Akibatnya kapasitas server terlampaui oleh proses yang berjalan di dalamnya.

4.3.2 DNS Server *Request Amplification*

Untuk mematikan server DNS sebagai bagian dari serangan *Distributed Denial of Service* (DDoS), eksploitasi amplifikasi DNS mengubah kueri sederhana menjadi muatan yang lebih besar. Setelah menampilkan opsi dengan perintah bantu/scanner/dns/dns_amp, konfigurasikan menggunakan perintah modul yang tersedia seperti yang ditunjukkan pada gambar 12 di bawah.



Gambar 12. Serangan *Distributed Denial of Service.*

Ini telah secara efektif menyusup ke modul DNS Amp, sesuai dengan hasil eksploitasi. Mengirimkan 87 byte permintaan untuk menguji paket menunjukkan efek amplifikasi DNS.

5. Kesimpulan

Dari hasil pengujian pada Aplikasi Web Sistem Evaluasi Data Bidang Tik Polda Aceh berhasil ditemukannya kerentanan-kerentanan yang terdapat pada Aplikasi Web tersebut seperti:

1. Sebaiknya menggunakan DNS karena aplikasi web tersebut masih menggunakan *IP Adress* server sebagai DNS sehingga dapat memungkinkan peyerang mengakses situs yang lain pada server tersebut.
2. Sebaiknya ditambahkan standar pengamanan pada web server tersebut berupa *Secure Socket Layer* (SSL) karena pada web server tersebut tidak menggunakan SSL yang menjadi standar dalam suatu sistem informasi.
3. Sebaiknya mengkonfigurasi *Header X Type Options* karena penyerang dapat membaca informasi sensitif melalui respon pada Aplikasi Web tersebut.
4. Sebaiknya memperbaiki kesalahan dalam pemrograman atau *insecure design* yang menjadi sumber kerentanan yang dapat meberikan informasi mengenai kerentanan-kerentanan yang lain.
5. Sebaiknya mengkonfigurasi CSP untuk meningkatkan keamanan karena jika tidak dikonfigurasi dapat mengakibatkan perusakan situs dan penyebaran malware.
6. Untuk menangani serangan DoS
    1) Sebaiknya server menggunkan *Firewall* untuk menghindari serangan,
    2) Melakukan *Blocking* terhadap IP yang terliht mencurigakan,
    3) Menolak paket data dan mematikan service UDP (User Diagram Protocol),
    4) Menggunakan antivirus yang dapat menangkal serangan data seperti Kapersky,
    5) Melakukan *Filtering* pada permintaan ICMP *echo* pada *Firewall.*
7. Untuk menangani serangan DDoS

1) Sebaiknya Menggunakan *Firewall* yang kuat,
2) Sebaiknya Menggunakan *Load Blancer* untuk mendistribusikan beban jaringan secara merata ke beberapa server.,
3) Sebaiknya memasang Sertifikasi SSL sebagai standar keamanan suatu sistem informasi..

Sebaiknya dilakukan perbaikan untuk meningkatkan keamanan sistem baik pada Aplikasi Web maupun Server.

Referensi

[1] B. P. Sembiring, M. F. Sidiq, and W. A. Prabowo, "Analisis Keamanan Sistem Informasi Menggunakan Metode Open Web Application Security Project (Owasp)," vol. 8, no. 3, pp. 3049–3054, 2024.

[2] L. S. Sign-on, S. Ulfa, and D. Surya, "Implementasi Penetration Testing Execution Standard Untuk Uji Penetrasi Pada," vol. 8, no. 1, pp. 48–56, 2021.

[3] M. Rozali and M. Dayan Sinaga, "DIAGNOSIS KEAMANAN WEB MENGGUNAKAN METODE UJI PENETRASI WEBSITE SEKOLAH Web Security Diagnosis Using School Website Penetration Test Method," *JID (Jurnal Info Digit.*, vol. 2, no. 1, pp. 248–262, 2024, [Online]. Available: http://kti.potensi-utama.ac.id/index.php/JID

[4] W. Ma, M. T. S. Husnul, and K. Kuningan, "1262-Article Text-3020-1-10-20230116," vol. 8, no. 3, pp. 138–145, 2022.

[5] S. Serangan, "Ancaman dan Solusi Serangan Siber di Indonesia," vol. 1, no. 2, pp. 85–92, 2021.

[6] A. Bastian, H. Sujadi, and L. Abror, "Analisis Keamanan Aplikasi Data Pokok Pendidikan (Dapodik) Menggunakan Penetration Testing Dan Sql Injection," *INFOTECH J.*, vol. 6, no. 2, pp. 65–70, 2020.

[7] A. Fatihah and P. Dinarto, "Analisis Keamanan Aplikasi Website Menggunakan Metode Penetration Testing Berdasarkan Framework ISSAF Pada Perusahaan Daerah XYZ," *Innov. J. Soc. Sci. Res.*, vol. 4, pp. 4536–4549, 2024.

[8] A. Jakobsson, "Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications," 2022.

[9] K. Nisa, M. A. Putra, R. A. Siregar, and M. D. Irawan, "Analisis Website Tapanuli Tengah Menggunakan Metode Open Web Application Security Project Zap (Owasp Zap)," vol. 3, no. 4, pp. 308–316, 2022.

[10] Sunardi, I. Riadi, and P. A. Raharja, "Vulnerability analysis of E-voting application using open web application security project (OWASP) framework," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 135–143, 2019, doi: 10.14569/IJACSA.2019.0101118.

[11] J. Jtik, J. Teknologi, I. F. Ashari, M. Affandi, H. T. Putra, and M. T. Nur, "Security Audit for Vulnerability Detection and Mitigation of UPT Integrated Laboratory (ILab) ITERA Website Based on OWASP Zed Attack Proxy (ZAP)," vol. 7, no. 1, 2023.

[12] Y. A. Pohan, "Meningkatkan Keamanan Webserver Aplikasi Pelaporan Pajak Daerah Menggunakan Metode Penetration Testing Execution Standar," *J. Sistim Inf. dan Teknol.*, vol. 3, pp. 1–6, 2021, doi: 10.37034/jsisfotek.v3i1.36.

[13] M. Nist, S. P. Dan, O. Di, and J. D. No, "Analisis Kerentanan Website Menggunakan Diskominfo Kabupaten Bandung".

[14] S. S. Anelia and B. Hananto, “Uji Penetrasi Server Universitas PQR Menggunakan Metode National Institute Of Standards And Technology (NIST SP 800-115),” vol. 07, no. 01, pp. 35–43, 2023, doi: 10.22441/jitkom.2023.v7i1.005.
[15] M. D. Purnomo, A. Chusyairi, U. B. Insani, S. Jaya, and K. Bekasi, “Pengujian Keamanan Sistem Mengunakan Metode Penetration Testing di Website Diskominfostandi Kota Bekasi,” vol. 1, no. 1, pp. 92–101, 2024.
[16] S. A. Maherza, B. Hananto, and I. W. W. Pradnyana, “Penetration Testing Terhadap Website Sekolah Menengah Atas ABC dengan Metode NIST SP 800-115,” *Inform. J. Ilmu Komput.*, vol. 19, no. 1, pp. 11–27, 2023, doi: 10.52958/iftk.v19i1.4697.
[17] D. Metode, N. Sp, and D. A. N. Owasp, “Analisa Kualitas Keamanan Pada Aplikasi Slims Akasia,” pp. 500–506, 2024.