Analisis Keamanan Website Leads UPNVJ Terhadap Serangan SQL Injection & Sniffing Attack

Dimas Perdana Putranto[1], Jayanta[2], Bayu Hananto[3]
[1,2,3]Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta
[1,2,3]Jl. RS. Fatmawati Raya, Pd. Labu, Depok , Jawa Barat 12450
dimaspp@upnvj.ac.id[1], jayanta@upnvj.com[2], bayuhananto@upnvj.ac.id[3]

Abstrak. Leads UPNVJ adalah sistem yang dirancang untuk keperluan pembelajaran elektronik bagi mahasiswa, yang dapat diakses secara *online* menggunakan *Website*. Pada *website* LEADS UPNVJ terdapat data yang penting dan harus dijaga keamanannya karena terdapat informasi-informasi yang bersifat privasi. Metode serangan yang digunakan pada penelitian ini adalah SQL *injection* & *Sniffing Attack*. SQL *injection* adalah sebuah serangan injeksi SQL yang dapat menimbulkan ancaman keamanan serius terhadap sebuah website, yang mana SQL *injection* mengizinkan penyerangnya untuk mendapatkan akses ke *database* sebuah *website* yang dapat menyebabkan kebocoran data yang membuat hilangnya kerahasiaan data terutama untuk informasi yang bersifat sensitif. *Sniffing Attack* adalah penyadapan dan pencurian data dengan cara menangkap dan memonitor lalu lintas paket data jaringan internet. Yang bertujuan untuk memperoleh data dan informasi yang bersifat sensitif. Oleh karena itu melalui penelitian ini, diharapkan dapat diperoleh analisis yang baik untuk mengukur tingkat keamanan dari *website* LEADS UPNVJ. Hasil dari penelitian ini menunjukkan tingkat keamanan *website* LEADS UPNVJ sudah cukup baik dalam menanggulangi serangan SQL *injection & sniffing attack*, pada serangan SQL *injection website* dapat mencegah serangan dikarenakan terdapat *Web Application Firewall* (WAF) yang dapat mencegah serangan kueri SQL dijalankan oleh *server website,* lalu pada serangan *sniffing attack website* dapat mencegah serangan terjadi dikarenakan terdapat protokol *Transport Layer Security* (TLS) yang dapat mengenkripsi isi dari paket data yang dikirimkan dari *client* menuju *server* maupun sebaliknya.

Kata Kunci: Leads UPNVJ, *website,* serangan, keamanan, SQL *Injection, Sniffing attack, database,* dan paket data.

1 Pendahuluan

Pada saat ini kebutuhan akan suatu *website* menjadi sangat penting, *website* bukan lagi hanya menjadi sarana penyedia informasi, melainkan sudah menjadi media komunikasi, media transaksi, media pembelajaran, dan lain-lain. Karena peran *website* sangat penting dan mencakup banyak aspek, maka banyak terjadi kasus pencurian data, penyadapan, dan lain-lain yang dilakukan oleh *hacker*. untuk itu sistem keamanan pada suatu *website* menjadi hal yang sangat penting untuk menghindari terjadinya hal-hal yang tidak diinginkan yang disebabkan oleh serangan *hacker*.

Serangan yang sering digunakan dan yang berbahaya adalah SQL *injection* & *Sniffing attack*. serangan SQL *injection* adalah teknik yang dapat mengeksploitasi kueri dari *structured query language* (SQL) untuk bisa menembus menuju *back-end* dari *database*, jika sudah menembus *database*, penyerang mendapat keleluasaan dalam mendapatkan informasi sensitif yang terdapat pada database tersebut, seperti *username*, *password*, nama, alamat, nomor telepon, dan lain-lain. Lalu *sniffing attack* adalah ketika *packet* melakukan lalu lintas melalui jaringan *internet*, lalu lintas data tadi dapat dianalisis untuk mendapatkan data dan informasi yang bersifat sensitif. Lalu lintas data tadi bisa ditangkap menggunakan bantuan alat *sniffing*.

LEADS UPNVJ merupakan sistem pembelajaran elektronik berbasis *web* yang dapat diakses kapan saja dan dimana saja. LEADS UPNVJ digunakan oleh seluruh mahasiswa & dosen di UPNVJ. Kegiatan-kegiatan pembelajaran yang bisa dilakukan di LEADS UPNVJ meliputi pemberian materi, pelaksanaan ujian, absensi, penugasan, penilaian, dan lain-lain. Mengetahui bahwa SQL *injection* & *Sniffing attack* merupakan serangan yang berbahaya dan dapat mengancam privasi dari penggunanya. maka penulis memutuskan untuk mengangkat tema ini dengan mengambil judul "Analisis Keamanan Website LEADS UPNVJ Terhadap Serangan SQL Injection & Sniffing Attack".

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

2 Landasan Teori

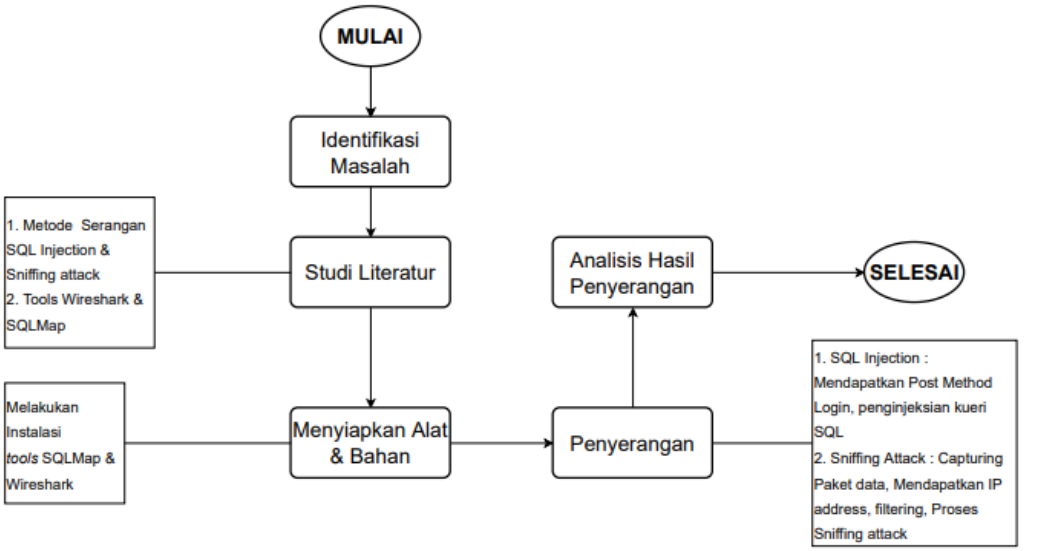
Pada penelitian ini memperlukan teori-teori untuk memperkuat dasar dari penelitian yang dilakukan, berikut merupakan landasan teori yang digunakan pada penelitian ini.

a. SQL (*Structured Query Language*) adalah bahasa database komputer yang dirancang untuk mengelola data di dalam sebuah sistem manajemen basis data relasional [1]. SQL merupakan bahasa komputer standar yang dikembangkan oleh IBM, banyak hal yang bisa dilakukan oleh SQL seperti, menambahkan database baru, melakukan update pada database baru, menghapus data pada database, dan lain-lain. Walaupun SQL merupakan bahasa yang menjadi standar untuk sistem database, tetapi banyak sistem database yang mengimplementasikan bahasa SQL versinya sendiri-sendiri seperti, Microsoft SQL Server, MySQL, Microsoft Access, Sybase, dan lain-lain.
b. SQL Injection merupakan suatu teknik eksploitasi dengan cara melakukan modifikasi perintah sql pada form input suatu aplikasi yang nantinya akan memungkinkan penyerang untuk mengirimkan sintaks atau perintah kepada database suatu aplikasi [2].
c. Sniffing Attack, teknik dimana data paket yang mengalir melalui jaringan terdeteksi dan diamati. Administrator jaringan menggunakan alat sniffing paket untuk memantau dan memvalidasi lalu lintas jaringan, sementara peretas dapat menggunakan alat serupa untuk tujuan jahat [3]. Dapat disimpulkan bahwa Sniffing attack adalah teknik penyadapan melalui proses penangkapan aliran paket data yang melalui jaringan tertentu dengan menggunakan alat sniffing. Informasi yang ditangkap bisa berupa informasi yang sensitif seperti username, password, dan lain-lain.
d. Kerentanan adalah kelemahan suatu sistem yang dieksploitasi oleh penyerang, biasanya dilakukan untuk mendapatkan akses ke beberapa aset. banyaknya keadaan yang secara tidak sengaja menciptakan kelemahan di dalam sistem, mereka dibagi menjadi tiga area : layanan, aplikasi, tindakan yang dilakukan oleh user [4].
e. Website merupakan apa yang anda lihat melalui browser, sedangkan definisi dari web adalah sebuah aplikasi web, karena disana kita akan melakukan perintah tertentu dan membantu anda dalam melakukan aktifitas tertentu [5].
f. Serangan Siber (cyber attack) adalah serangan dunia maya, baik yang ditujukan untuk menyerang maupun bertahan yang menjadi alasan sebagai penyebab kematian seseorang atau kerusakan suatu objek yang dituju [6].
g. Basis Data adalah kumpulan informasi yang disimpan pada komputer dengan cara yang sistematik sehingga dapat diperiksa dengan menggunakan program komputer agar dapat memperoleh informasi [7].
h. Web Application Firewall atau firewall aplikasi web membantu melindungi aplikasi web dengan memfilter dan memantau lalu lintas HTTP antara aplikasi web dan Internet. Ini biasanya melindungi aplikasi web dari serangan seperti *cross-site forgery*, *cross-site-scripting* (XSS), dan SQL *injection* [9]. Dapat disimpulkan bahwa *Web Application Firewall* (WAF) adalah bentuk lain dari *firewall* yang bertugas mengidentifikasi, menyaring, dan menahan aliran data yang dianggap mencurigakan dari *client* menuju *server* dari suatu *website*. Berikut merupakan gambaran dari cara kerja dari WAF.
i. Transport Layer Security (TLS) Protokol kriptografi yang dirancang untuk menyediakan komunikasi yang aman melalui jaringan komputer. Beberapa versi protokol TLS banyak digunakan di *browser web*, *email*, pesan instan, dan *voice over IP* (VoIP). Situs *web* dapat menggunakan TLS untuk mengamankan semua komunikasi antara *server* dan *browser web* [10]. Dapat disimpulkan bahwa *Transport layer security* (TLS) merupakan protokol kriptografi yang berfungsi mengamankan komunikasi paket data yang dikirimkan antara *client* dan *server*, sehingga isi dari paket data yang dikirimkan tadi bisa dilindungi privasi dan kerahasiaannya agar terjalin komunikasi yang aman pada jaringan *internet*.

3 Hasil

Pada pengujian ini akan dilakukan sesuai dengan alur penelitian yang sudah dibuat. Berikut merupakan gambar dari alur penelitian yang akan dilakukan.

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)



Gambar. 1. Gambar kerangka pikir penelitian

Berikut ini merupakan tahapan-tahapan yang akan dilakukan dalam penelitian ini :

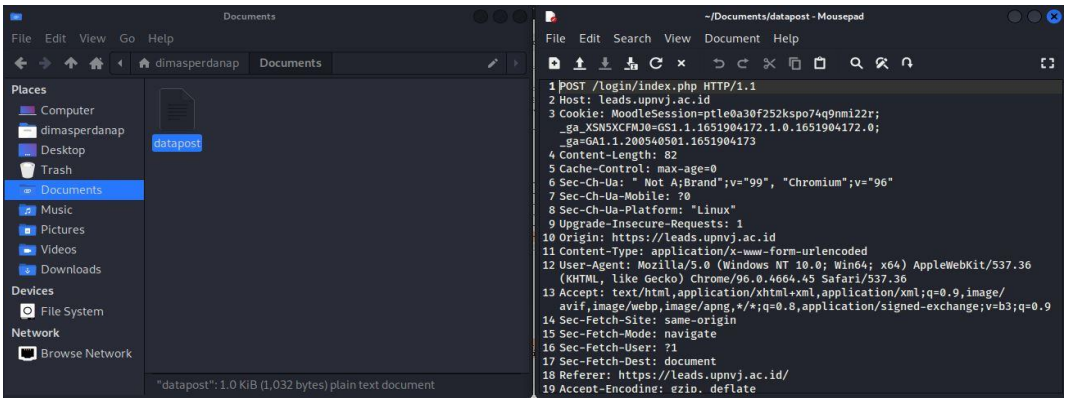
a. Identifikasi Masalah, dalam tahap ini penulis mencoba mencari masalah dan menegaskan masalah yang akan diangkat pada penelitian ini.
b. Studi Literatur, dalam tahap ini penulis mengumpulkan sumber literatur dari mulai buku, jurnal, artikel, dan lain-lain untuk menunjang penelitian ini.
c. Menyiapkan Alat dan bahan, dalam tahap ini penulis menyiapkan alat dan bahan berupa perangkat keras dan perangkat lunak yang akan digunakan untuk membantu penelitian ini.
d. Penyerangan, dalam tahap ini penulis akan mencoba melakukan penyerangan SQL *Injection* dan *Sniffing Attack* untuk serangan SQL *injection* memiliki tahapan seperti berikut, tahapan pertama adalah untuk mendapatkan *POST Method login*, lalu tahap kedua adalah tahap penginjeksian kueri SQL. Untuk *sniffing attack* tahapan dari serangannya adalah seperti berikut, tahapan pertama adalah *Capturing* paket data, lalu tahapan kedua adalah mendapatkan IP *address*, selanjutnya tahapan ketiga adalah *filtering*, dan tahapan terakhir adalah melakukan proses *sniffing attack* pada paket data.
e. Analisis Hasil Penyerangan, dalam tahap ini penulis akan melakukan analisis terhadap hasil dan mengambil kesimpulan dari hasil penyerangan tersebut.

3.1 Serangan SQL *Injection*

SQL *injection* merupakan suatu serangan yang memanfaatkan kerentanan pada suatu lapisan *database* sebuah *website*. Mekanisme dari serangan ini adalah mencoba menginjeksikan perintah-perintah SQL pada *form input* suatu aplikasi sehingga penyerang bisa mengirimkan perintah ke *database website* tersebut, yang pada akhirnya penyerang nanti dapat menguasai *database* pada *website* tersebut. Jika penyerang sudah menguasai *database* maka penyerang bisa mencuri data-data pada *database* yang biasanya bersifat pribadi dan rahasia seperti *username*, *password*, tanggal lahir, dan lain-lain. Pada penelitian kali ini penulis akan melakukan percobaan serangan SQL *injection* menggunakan *tools* SQLMap dan Burp Suite. Berikut merupakan hasil dari pengujiannya.

a. Mendapatkan POST *Method Login*

Sebelum melakukan penyerangan kita harus mendapatkan dulu data POST *method login* pada saat kita melakukan penginputan pada *form login website* LEADS UPNVJ. Pengambilan data *POST method login* menggunakan *tools Burp Suite*. Tujuan kita mendapatkan *POST method* tersebut adalah agar nanti *SQLMap* dapat menganalisa *POST method login* yang sudah kita dapatkan tadi lalu selanjutnya *SQLMap* akan mencoba menemukan kerentanan pada *form input website* LEADS UPNVJ dengan cara menginjeksikan perintah-perintah SQL. Berikut adalah hasil dari pengambilan data *POST method login* dengan *tools Burp Suite.*

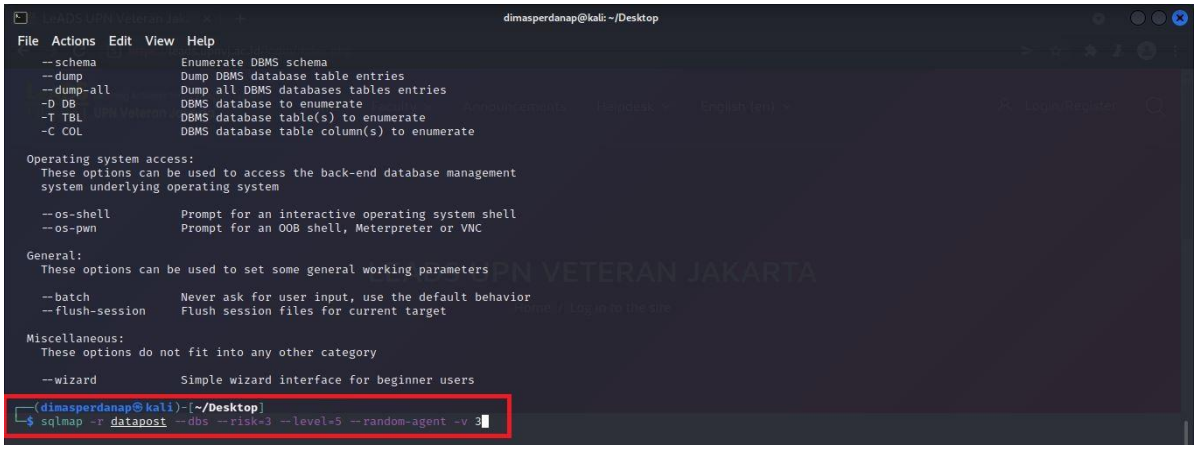


Gambar. 2. File POST *method login*

Pada gambar 2 adalah jika kita telah berhasil menyimpan file *POST method login*. disini saya memberikan nama file tersebut dengan nama datapost. Selanjutnya adalah kita akan membuka *SQLMap* untuk melakukan proses serangan *SQL injection*.

b. **Penginjeksian Kueri SQL**

Selanjutnya adalah melakukan serangan SQL *injection* menggunakan *tools* SQLMap. Pada tahap ini SQLMap akan melakukan penginjeksian kueri-kueri SQL secara otomatis untuk mencari kerentanan pada layer database dari *website* tersebut. Berikut adalah tampilan dari perintah atau *command* yang digunakan untuk melakukan serangan SQL *injection.*



Gambar. 3. Perintah/*command* yang digunakan pada *tools* SQLMap untuk melakukan serangan SQL *injection*

Pada Gambar 3 adalah langkah yang melakukan serangan *SQL injection* menggunakan tools SQLMap dengan mengetikan perintah atau *command.*

"sqlmap -r datapost –dbs –risk=3 –level=5 –random-agent -v 3 ".

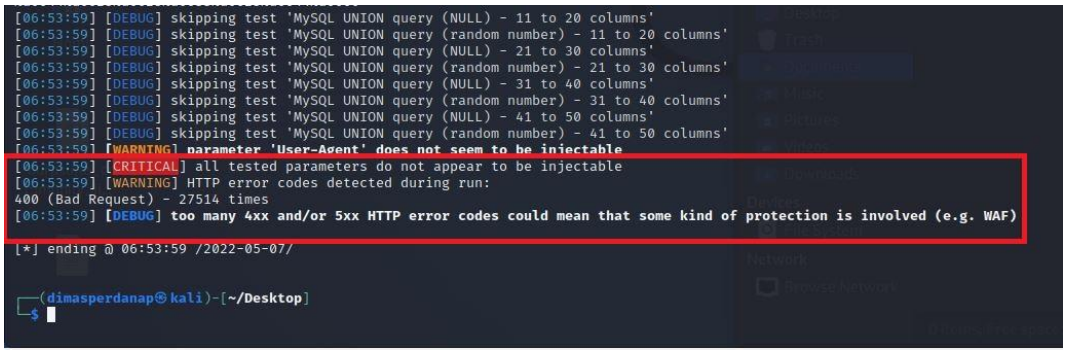
Berikut merupakan fungsi dari perintah atau *command* yang penulis gunakan:

- -r = perintah atau *command* ini berfungsi untuk membaca file yang akan kita eksekusi.
- datapost = merupakan nama *file request POST method login* yang tadi kita sudah simpan dan yang akan dieksekusi.
- –dbs = perintah atau *command* ini berfungsi untuk mendapatkan nama-nama dan daftar *database* yang berada pada *target*
- –risk = perintah atau *command* ini berfungsi untuk mengatur tingkat dari tipe kueri *SQL* yang akan diinjeksikan ke suatu website. Pada perintah –risk ini dibagi menjadi 3 *level* (1-3), semakin tinggi *level* yang dipilih maka akan semakin besar juga tingkat keberhasilan serangan *SQL injection.* Tapi proses yang dilakukan akan berlangsung lama karena akan lebih banyak mengandung perintah kueri *SQL* yang berat untuk dieksekusi, disini penulis memutuskan menggunakan *level 3* untuk meningkatkan kemungkinan berhasilnya serangan.

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

- –level = perintah atau *command* ini berfungsi untuk mengatur tingkat pencarian tools *SQLMap* dalam mencari atau *scan* kerentanan yang ada pada *website* tersebut. Pada perintah –level ini dibagi menjadi 5 *level* (1-5), semakin tinggi level yang dipilih maka akan semakin tinggi tingkat keberhasilan dalam menemukan celah kerentanan pada website tersebut, karena lebih dalam dan lebih banyak yang di *scan.* Tapi semakin tinggi *level* yang dipilih maka akan semakin lama prosesnya, karena akan semakin dalam pencarian dan semakin banyak perintah yang dieksekusi. Disini penulis memutuskan menggunakan level 5 untuk memperdalam pencarian celah kerentanan agar meningkatkan tingkat keberhasilan serangan.
- -random-agent = perintah atau *command* ini berfungsi untuk memberikan informasi user-agent palsu yang akan diterima oleh server, misal seperti penyerang mencoba melakukan request ke server melalui *browser* Firefox lalu data yang diterima oleh server bahwa penyerang mengakses menggunakan *browser* safari. Jadi dalam dunia *web security* informasi user-agent itu sebenarnya tidak dapat dipercaya karena bisa dengan mudah dimanipulasi.
- -v = perintah atau *command* ini disebut juga *verbosity* berfungsi untuk memberikan penyerang informasi kueri atau perintah-perintah *SQL* apa saja yang sedang diinjeksikan dan untuk memberi tahu detail tentang apa saja yang sedang dijalankan oleh *tools SQLMap*. *level Verbosity* ini dibagi menjadi 6 level, semakin tinggi *level verbosity* yang digunakan maka detail apa saja yang sedang dikerjakan oleh *SQLMap* akan semakin jelas dan detail. Tetapi akan membuat kinerja dari SQLMap semakin berat, disini penulis memutuskan untuk menggunakan level 3 saja karena tidak menambah efek apapun dalam pencarian kerentanan dan penginjeksian kueri *SQL*. Dan hanya membuat lebih berat kinerja SQLMap. Selanjutnya adalah melakukan konfirmasi untuk melakukan penyerangan.

Setelah melakukan proses penginjeksian kueri SQL maka akan keluar hasil dari serangan yang dilancarkan oleh penyerang. Berikut merupakan hasil dari penyerangan yang sudah dilakukan oleh penyerang.



Gambar. 4 Hasil dari pengujian SQL *injection*

Pada gambar 4 dapat dilihat bahwa SQLMap tidak dapat menemukan kerentanan pada *website* LEADS UPNVJ dan serangan gagal menembus WAF (Web Application Firewall) yang digunakan oleh *website* LEADS UPNVJ.

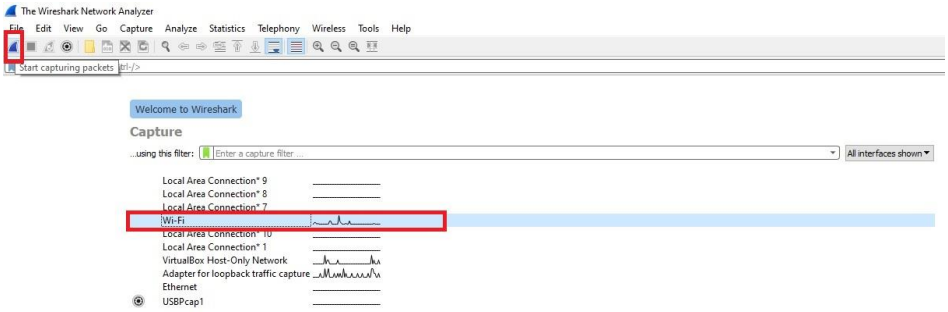
3.2 Serangan S*niffing Attack*

Sniffing attack merupakan suatu skema serangan menggunakan teknik penyadapan aliran paket data melalui jaringan tertentu yang dikirim dari *client* menuju *server* maupun sebaliknya dengan menggunakan bantuan alat *sniffing.* Informasi yang dikirimkan dari client menuju server ataupun sebaliknya bisa berupa informasi yang sensitif dan privat seperti *username, password*, alamat email, dan lain-lain.

Pada penelitian kali ini penulis akan melakukan percobaan serangan *sniffing attack* menggunakan *tools* wireshark. Berikut merupakan hasil dari pengujiannya.

a. Melakukan *Capturing* Paket Data

Pertama kita akan menjalankan *tools* Wireshark yang akan membantu kita dalam menyadap aliran paket data yang dikirimkan dari client menuju server maupun sebaliknya, seperti pada gambar berikut.

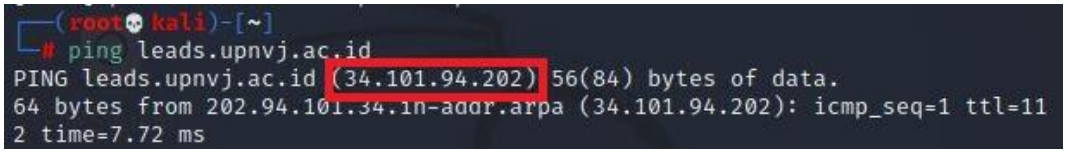


Gambar. 5. Proses *Capturing* paket data

Pada gambar 5 merupakan tampilan awal pada menu wireshark, disini kita bisa memilih jaringan mana yang ingin kita *capture* aliran paket data nya. karena disini komputer client yang mengirimkan paket data menuju server terhubung dengan jaringan wifi. Maka penulis mengkonfigurasi *tools* wireshark untuk menangkap aliran paket data yang dikirimkan maupun diterima melalui jaringan wifi, setelah itu kita bisa mulai melakukan menangkap paket data.

b. Mendapatkan IP *address*

Selanjutnya adalah mendapatkan alamat IP dari *website* LEADS UPNVJ untuk membantu kita dalam melakukan tahapan selanjutnya yaitu proses *filtering*.

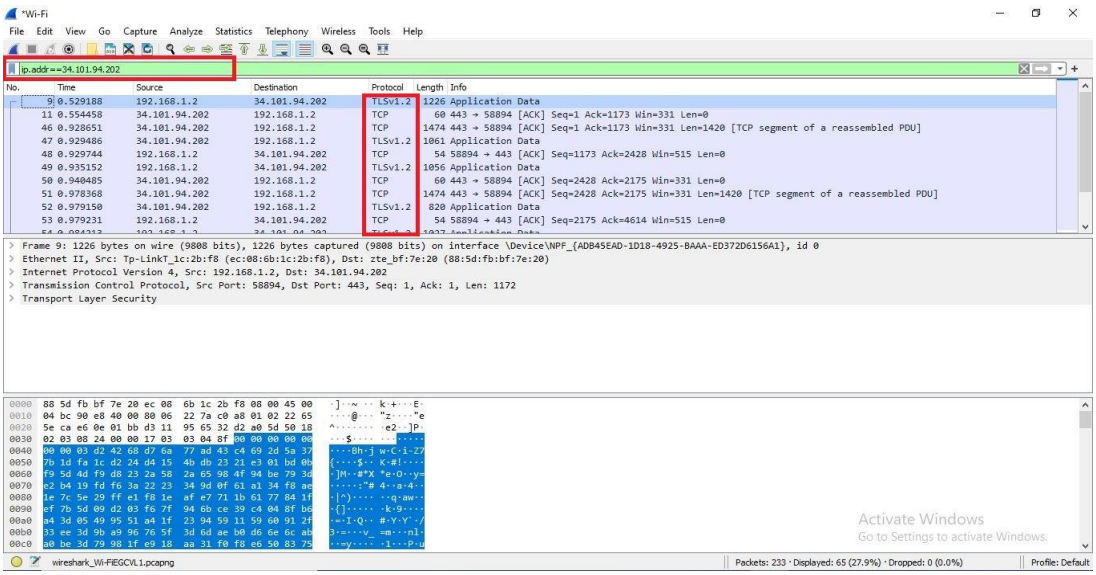


Gambar. 6. Proses mendapatkan alamat IP address website LEADS UPNVJ

Pada gambar 6 untuk mendapatkan alamat IP address dari website LEADS UPNVJ kita bisa mengetikkan perintah atau *command* "ping leads.upnvj.ac.id". Setelah itu maka akan muncul alamat IP dari *website* LEADS UPNVJ yaitu 34.101.94.202.

c. *Filtering*

Selanjutnya penulis melakukan proses *filter* yang berfungsi untuk menyaring aliran paket data yang dikirimkan oleh *website* LEADS UPNVJ untuk memudahkan kita dalam menganalisis aliran paket data pada alamat IP yang dituju yaitu alamat IP website LEADS UPNVJ.

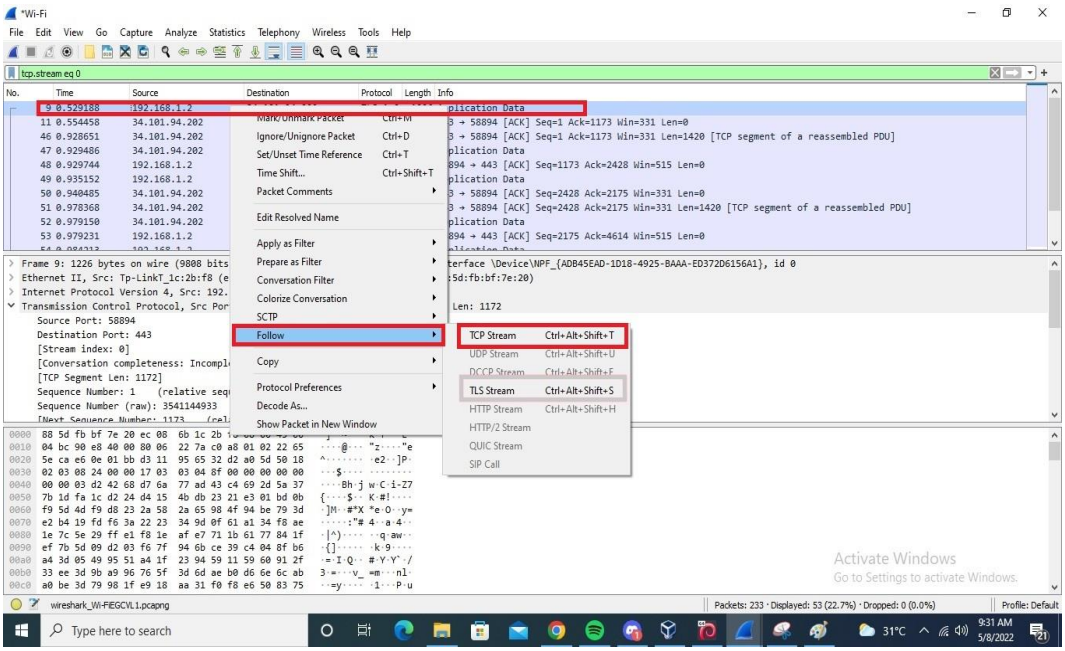


Gambar. 7. Proses filter paket data dari website LEADS UPNVJ

Pada gambar 7 disini penulis melakukan proses *filtering* yang berfungsi untuk menyaring aliran paket data yang dikirimkan oleh *website* LEADS UPNVJ untuk memudahkan kita dalam menganalisis aliran paket data pada alamat IP yang dituju yaitu alamat IP website LEADS UPNVJ dengan cara menggunakan perintah atau *command* "ip.addr==34.101.94.202". Setelah proses *filter* selesai kita dapat melihat bahwa paket data itu berasal darimana dan tujuannya kemana pada menu *source* dan *destination*, dapat dilihat yang meminta paket data dan yang mengirimkan paket data adalah alamat IP 192.168.1.2 dan alamat IP 34.101.94.202, yang artinya terjadi komunikasi data antara alamat IP komputer korban dengan alamat IP *website* LEADS UPNVJ.

d. Proses *Sniffing Attack*

Selanjutnya penulis akan melakukan proses *sniffing attack* pada aliran paket data yang dikirimkan dari komputer *client* menuju *server* pada saat melakukan *login* pada *website* LEADS UPNVJ yang dimana pada paket data tersebut terdapat informasi mengenai *username* dan *password* yang dikirimkan dari client menuju server.

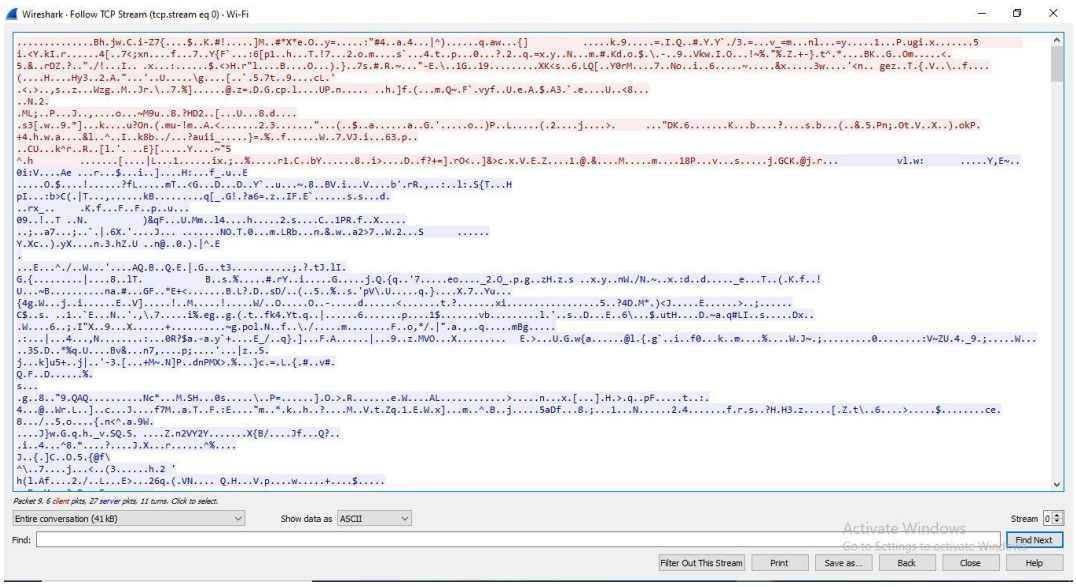


Gambar. 8. Proses sniffing attack pada paket data website LEADS UPNVJ

Pada gambar 8 kita akan melakukan proses *sniffing attack* pada aliran paket data yang dikirimkan dari komputer *client* menuju *server* pada saat melakukan *login* pada *website* LEADS UPNVJ yang dimana pada paket data tersebut terdapat informasi mengenai *username* dan *password* yang dikirimkan dari client menuju server, dengan cara menggunakan mengklik kanan mouse pada paket data yang ingin kita ambil informasi paket datanya lalu klik menu *follow* dan pilih TCP *stream* untuk menganalisis isi paket data yang dikirimkan dari client menuju server.

Berikut merupakan hasil dari *sniffing attack* yang dilakukan oleh penyerang kepada isi dari paket data yang dikirimkan dari *client* menuju *server*. Seperti pada gambar berikut.

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)



Gambar. 9. Hasil dari *sniffing attack* pada aliran paket data

Pada gambar 9 berikut merupakan tampilan dari TCP stream yang sudah di capture menggunakan wireshark. Dapat dilihat bahwa pada protokol TCP bahwa isi dari paket data tidak dapat dianalisis oleh penyerang dikarenakan pesan yang dikirimkan sudah dienkripsi oleh protokol TLS. Jadi *tools* wireshark tidak bisa menyadap aliran paket data yang dikirimkan dari *client* menuju *server website* LEADS UPNVJ sehingga informasi yang sensitif seperti *username* dan *password* dapat terlindungi dengan baik.

3.3 Hasil Keseluruhan Dari Penyerangan

Berikut merupakan tabel rekap hasil serangan SQL *injection* & *sniffing attack* terhadap *website* LEADS UPNVJ.

Tabel. 1 Hasil dari *sniffing attack* pada aliran paket data

No	Jenis Serangan	Tools	Waktu Penyerangan	Hasil Serangan
1	SQL Injection	Burp Suite & SQLMap	6 Jam	*Website LEADS UPNVJ aman dari serangan SQL Injection karena terdapat Web Application Firewall (WAF)*
2	*Sniffing Attack*	Wireshark	15 Menit	Website LEADS UPNVJ aman dari serangan sniffing attack karena terdapat protokol TLS yang mengenkripsi paket data yang dikirim dari client menuju server

Pada tabel 1 merupakan rekap hasil percobaan serangan SQL *injection* & *Sniffing attack* terhadap *website* LEADS UPNVJ, dari percobaan yang sudah penulis lakukan mulai dari serangan SQL *injection* dan *sniffing attack* dapat disimpulkan bahwa keamanan *website* LEADS UPNVJ mampu mengatasi serangan-serangan tersebut. Dalam mengatasi serangan SQL *injection*, *website* LEADS UPNVJ juga memasang *Web application firewall* (WAF), WAF merupakan sebuah *firewall* yang melakukan memantau aliran data, melakukan penyaringan data, dan memblokir aliran data yang dianggap berbahaya dari komputer *client* menuju *web server*, dalam hal serangan SQL *injection* nanti WAF akan memblokir semua usaha penginjeksian perintah SQL, pada suatu sistem WAF sendiri bisa berbasis *host*, *cloud*, dan jaringan lokal. Dalam mengatasi serangan *sniffing attack website* LEADS UPNVJ sudah menggunakan protokol *Transport layer security* (TLS). TLS merupakan protokol yang bertugas untuk melakukan enkripsi pada perpindahan paket data dari *client* menuju ke *server* maupun sebaliknya. Seperti pada percobaan yang sudah dilakukan bahwa isi dari paket data yang dikirimkan dari *client* menuju *server website* LEADS UPNVJ tidak bisa dianalisis isi pesannya oleh penyerang karena pesan diberikan enkripsi sehingga pesan hanya bisa dipecahkan oleh web server yang menerimanya.

4 Kesimpulan dan Saran

4.1 Kesimpulan

Setelah melakukan penelitian dapat ditarik beberapa kesimpulan seperti berikut:

a. Tingkat keamanan pada *Website* LEADS UPNVJ sudah cukup baik dalam menangani serangan SQL *injection* dan *sniffing attack.*
b. Keamanan *Website* LEADS UPNVJ cukup baik menghadapi serangan SQL *injection* karena sudah menggunakan *Web application firewall* (WAF) yang berfungsi melakukan pemantauan, penyaringan, dan pemblokiran data yang terindikasi berbahaya yang dikirimkan oleh *client* menuju *server* dalam hal SQL *injection* adalah memblokir usaha penginjeksian perintah kueri SQL menuju *database website*.
c. Keamanan *Website* LEADS UPNVJ cukup baik menghadapi serangan *sniffing attack* dikarenakan sudah menggunakan protokol *Transport layer security* (TLS) yang berfungsi untuk melakukan enkripsi pada isi paket data yang dikirimkan oleh *client* menuju *server* maupun sebaliknya sehingga penyerang yang ingin menyadap jalur komunikasi antara *client* dan *server* tidak dapat membaca isi dari paket data yang bersifat sensitif dan rahasia seperti *username*, *password*, nomor telepon, tanggal lahir, dan lain-lain.

4.2 Saran

Berdasarkan kesimpulan diatas, memang *website* LEADS UPNVJ sudah cukup baik dalam menangani serangan SQL *injection* dan *sniffing attack* tetapi untuk tetap menjaga tingkat keamanan yang tinggi, maka saran yang dapat dipertimbangkan untuk kedepannya antara lain:

a. Terus melakukan perpanjangan masa aktif secara berkala pada sertifikat TLS yang digunakan oleh *website* agar terhindar dari hal-hal yang tidak diinginkan.
b. Selalu melakukan *update* terbaru terhadap sistem *Web application firewall* (WAF) agar sistem WAF dapat mengidentifikasi dan menanggulangi ancaman atau serangan yang baru.
c. Terus melakukan *penetration testing* secara berkala untuk memeriksa celah-celah kerentanan pada suatu *website*.
d. Selalu melakukan *update* terbaru dari *browser* yang digunakan oleh pengguna karena pada *update* terbaru biasanya terdapat pembaruan terhadap *bug* atau celah keamanan yang dapat dieksploitasi.

Referensi

[1] H.-P. Halvorsen, “Structured Query Language,” 2017.
[2] Kementerian PPN/Bappenas, “SQL Injection.” https://csirt.bappenas.go.id/layanan/detail/180a9a4e-7e0a-49f2-ac92-92702ac094f5.
[3] Netscout, “What is packet sniffing?” https://www.netscout.com/what-is/sniffer#:~:text=Packet sniffing is a technique,similar tools for nefarious purposes.
[4] C. P. Pfleeger dan S. L. Pfleeger, “Analyzing computer security: a threat/vulnerability/countermeasure approach,” *Upper Saddle River*, NJ: Prentice Hall.
[5] Tim EMS, “Teori Dan Praktik PHP-MySQL Untuk Pemula,” Jakarta: PT Elec Media Komputindo, 2014.
[6] M. N. Schmitt, “Tallinn Manual on the International Law Applicable to Cyber Warfare,” *Cambridge University Press*, 2013.
[7] R. Abdulloh, “7 in 1 Pemrograman Web untuk Pemula,” *Jakarta: Elex Media Komputindo*, 2018.
[8] A. Lubis, “Basis data dasar,” *Yogyakarta: Deepublish*, 2016.
[9] CloudFlare, “What is a Web Application Firewall (WAF)?,” https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/.
[10] Wikipedia, “Transport Layer Security,” https://id.wikipedia.org/wiki/Transport_Layer_Security.