CoSciTech
Computer Science and Information Technology

p-ISSN: 2723-567X
e-ISSN: 2723-5661

Jurnal Computer Science and Information Technology (CoSciTech)

http://ejurnal.umri.ac.id/index.php/coscitech/index

String Matching untuk Mendeteksi Serangan Sniffing (ARP Spoofing) pada IDS Snort

Ilham Firdaus*[1], Januar Al Amien[2], Soni[3]
Email: [1]150401045@student.umri.ac.id, [2]januaralamien@umri.ac.id, [3]soni@umri.ac.id

[123]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

Diterima: 14 Oktober 2020 | Direvisi: - | Disetujui: 29 Oktober 2020

Abstrak

Teknik *Sniffing* (ARP Spoofing) adalah serangan yang mengirimkan paket ARP palsu atau paket ARP yang sudah di modifikasi sesuai alamat jaringan penyerang untuk meracuni ARP *cache table* korban. Serangan ARP *spoofing* merupakan serangan yang berbahaya karena dapat memonitor aktifitas korban dalam melakukan pencarian pada *browser* serta dapat mencuri akun *login* sosmed,kantor, dan akun lainnya. Serangan ini mendukung terjadinya serangan jaringan komputer lainnya seperti *Denial of service, Man in the middle attack, host impersonatting* dan lain lain . Serangan *sniffing* pada umumnya didapati pada tempat – tempat yang menyediakan wifi publik seperti Kampus, Perpustakaan, kafe, dan lain lain. IDS *Snort* dapat mendeteksi serangan *sniffing* (Arp *Spoofing*). Metode *String Matching* algoritme KMP diterapkan untuk mendeteksi serangan pada *file logging snort* untuk memberikan *alert* (pesan) ke pengguna. Pengujian yang dilakukan adalah pengujian *black box* untuk menguji fungsionalitas aplikasi, dan uji akurasi. Seluruh fungsionalitas aplikasi berhasil, dan pengujian akurasi kecocokkan antara perhitungan manual pencarian *string matching* dengan aplikasi akurat.

Kata kunci: *ARP, Sniffing, ARP Spoofing, String Matching, KMP*

String Matching to Detect Sniffing Attacks (ARP Spoofing) on IDS Snort

Abstract

Sniffing technique (ARP Spoofing) is an attack that sends fake ARP packets or ARP packets that have been modified according to the network address attacker's to poison the victim's ARP cache table. ARP spoofing attack is a dangerous attack because it can monitor the activities of victims in searching the browser and can steal social logins, office and other accounts. This attack supports the occurrence of other computer network attacks such as Denial of service, Man in the middle attack, host impersonating and others. Sniffing attacks are generally found in places that provide public Wi-Fi such as campus, libraries, cafes, and others. IDS Snort can detect sniffing attacks (Arp Spoofing). String Matching Method KMP algorithm is applied to detect attacks on snort logging files to provide alerts (messages) to users. Tests carried out are black box testing to test application functionality, and accuracy testing. All application functionality was successful, and testing the accuracy of the match between manual calculations for string matching and accurate application.

Keywords: *ARP, Sniffing, ARP Spoofing, String Matching, KMP*

1. PENDAHULUAN

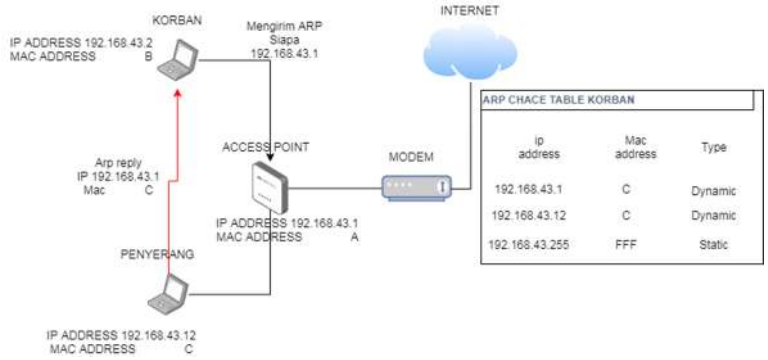
Berdasarkan *Norton Cyber Insight Report* tahun 2016, lebih dari 26% orang indonesia mengakui mereka terbiasa memanfaatkan jaringan wifi publik. Secara umum ada sekedar mengecek e-mail, mengirimkan dokumen, mengakses akun kantor untuk bekerja, dan mengakses akun media sosial. "mereka mengirim email, dokumen, masuk ke akun kantor, akun media sosialnya menggunakan komputer. Banyak informasi ada di ponsel tentunya itu sangat berharga dan privasi. Kalau terhubung wifi publik bisa saja dicuri informasi – informasinya (disadap)", ujar *security Advocate*, *Consumer Business Unit, Symantec*, Nick Savvides di Jakarta, Jumat (3/2/2017) [3].

Pada tanggal 16 maret 2020 penulis telah melakukan percobaan simulasi serangan wifi publik bertempat di laboraturium komputer UMRI dengan cara mengumpulkan 5 orang untuk menggunakan jaringan wifi publik yang memiliki SSID lab fasilkom dan melakukan penyerangan dengan teknik ARP *spoofing*, Penulis berhasil melakukan penyerangan *sniffing* (arp *spoofing*) pada jaringan labor, dan penulis mewawancarai ke pengguna wifi lab fasilkom tentang kesadaran akan disadap jaringan yang digunakannya.

ARP *Spoofing* adalah serangan yang mengirimkan paket ARP palsu atau paket ARP yang sudah di modifikasi untuk meracuni ARP *cache table korban.* Serangan ARP *spoofing* merupakan serangan yang berbahaya karena dapat mendukung terjadinya serangan jaringan komputer lainnya seperti *Denial of service, Man in the middle attack, host impersonatting* dan lain lain.

Salah satu cara yang dapat digunakan untuk mengatasi hal tersebut adalah dengan menggunakan *Intrusion detection system* (IDS), IDS adalah sistem pendeteksian dan pencegahan penyusup dengan menggunakan perangkat lunak (*software*) atau perangkat keras (*hardware*) yang bekerja secara otomatis untuk memonitor keadaan pada jaringan komputer dan dapat menganilisis masalah keamanan jaringan. *Software* yang menyediakan fitur IDS yaitu sebagai berikut, *Snort, Suricata, OSSEC, Zeek, Fail2Ban,* Dan masih banyak lainnya. Pada penelitian ini peneliti menggunakan *Snort* sebagai IDS [6].

Tujuan dari penelitian ini yang telah dibahas dari latarbelakang tersebut adalah Untuk menginformasikan ke pengguna jaringan wifi publik apakah jaringan wifi yang sedang digunakan terdapat pencurian informasi data atau serangan *sniffing*(ARP *spoofing*) dengan cara melakukan *String matching* untuk mendeteksi serangan *sniffing* (arp *spoofing*) pada ids snort untuk menampilkan alert ke pengguna. Adapun rekayasa skenario pada penelitian ini tampak pada gambar 1 sebagai berikut :



Gambar 1. Topologi dari penyerangan *sniffing* (ARP *spoofing*)

Pada Gambar diatas Pc Korban dengan ip 192.168.43.2 mengirim sebuah *request broadcast* arp siapa 192.168.43.1 akan tetapi Pc Penyerang melakukan sebuah serangan ARP *spoofing* jadi si Penyerang mengirim arp *reply* ke komputer korban bahwa ip 192.168.43.1 adalah mac *address* C yaitu mac *address* Penyerang, maka Arp *Chace Table* korban berubah secara *dynamic* menjadi mac *address* Penyerang.

2. METODE PENELITIAN

Adapun Tahapan Alur Metode Penelitian yang digunakan yakni tampak pada gambar 2 sebagai berikut:.

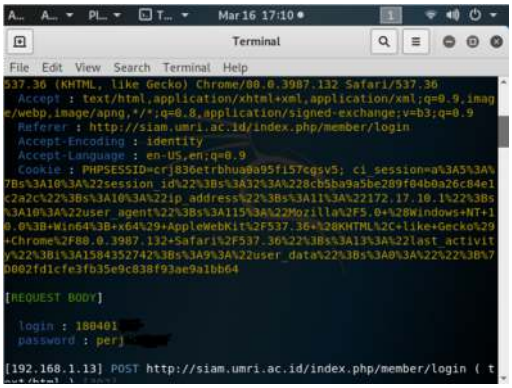
Gambar 2. Metodologi Penelitian

Pada Gambar 2 diatas, Metodologi Penelitian ini menggunakan diagram alir (*waterfall*) yang dilakukan secara bertahap yakni pertama Identifkasi Masalah & Pengumpulan Data, Ke-2 Analisa Kebutuhan Aplikasi, Ke-3 Perancangan Aplikasi, Ke-4 Implementasi Aplikasi dan tahap terakhir adalah Pengujian Aplikasi.

3. HASIL DAN PEMBAHASAN

3.1. Identifikasi Masalah

Data pada penelitian ini diperoleh dari beberapa sumber Web Edukasi, Website berita dan hasil simulasi yang telah dilakukan di salah satu laboratorium Universitas Muhammadiyah Riau yang memiliki wifi publik, berdasarkan hasil simulasi dapat disimpulkan bahwasanya pengguna/korban dari penyerangan *sniffing* (ARP *Spoofing*) tidak menyadari akan penyerangan tersebut. Jadi dibutuhkanlah aplikasi yang dapat mendeteksi serangan *sniffing* sehingga pengguna lebih aman untuk mengakses internet tanpa takut data pengguna akan disadap.



Gambar 3. Hasil Simulasi Penyerangan *Sniffing*

Pada Gambar 3 didapat sebuah data hasil *sniffing* pada salah satu pengguna jaringan wifi publik , pengguna A sedang mengakses siam.umri.ac.id dan login dengan menggunakan NIM dan password

3.2. Algoritme *Knuth Morris Pratt*

Algoritme *Knuth Morris Pratt* merupakan proses pencocokkan *string*. Bila terjadi ketidakcocokan pada saat *pattern* sejajar dengan teks [i...i + n-1], kita bisa menganggap ketidakcocokan pertama terjadi diantara teks [i+j] dan *pattten* [j], dengan <j<n. Berarti, teks [i..i + j] = *pattern* [0..j +1] dan a = teks [i + j] tidak sama dengan b = *pattern* [j], ketika kita menggeser.

Dengan kata lain, pencocokan string akan berjalan secara efesien bila kita mempunyai tabel yang menentukan berapa panjang seharusnya menggeser seandainya terdeteksi ketidakcocokkan di karakter ke-j dari *pattern*. Tabel itu harus memuat *next* [j] yang merupakan posisi karakter *pattern* seteleah digeser, sehingga kita menggeser *pattern* secara besar j-*next*[j] relatif terhadap teks.

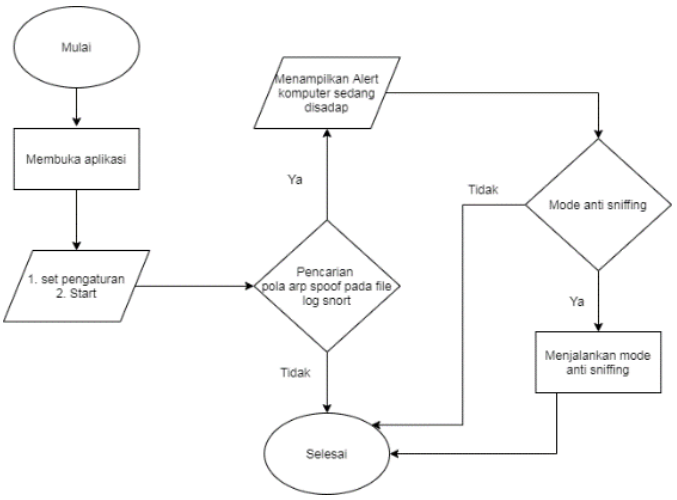
Secara sistematis, langkah langkah yang dilakukan dalam *Knuth Morris Pratt* pada saat mencocokkan string, sebagai berikut :

1. Algoritme *Knuth Morris Pratt* mulai mencocokkan *pattern* pada awal teks.
2. Dari kiri kekanan, algoritme ini akan mencocokkan karakter per karakter *patten*, dengan karakter di teks yang bersesuai sampai salah satu kondisi berikut terpenuhi:
   a. Karakter di pattern dan teks yang dibandingkan tidak cocok (mismatch).
   b. Semua karakter di *pattern* cocok. Kemudian algoritme akan memberitahukan penemuan diposisi ini.
3. Algoritme kemudian menggeser *pattern* berdasarkan *table next*, lalu menghitung langkah 2 sampai pattern berada di ujung teks[15].

3.3. Desain Model UML

Dalam tahapan desain ini akan mengartikan gambaran aplikasi menggunakan model perancangan UML*(Unified Model Language).*

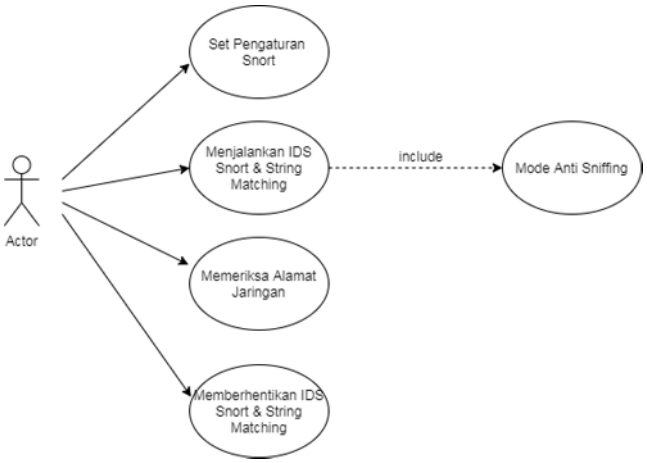
1. *Flowchart*



Gambar 4. Hasil Simulasi Penyerangan *Sniffing*

Pada gambar diatas Pengguna ketika menajalankan aplikasi *string matching* untuk deteksi serangan *sniffing* (ARP *spoofing*) pertama pengguna *set* pengaturan setiap menggunakan jaringan yang baru lalu kemudian menjalankan aplikasi, jika terdapat serangan maka akan tampil *alert* komputer sedang disadap, jika tidak maka program akan tetap berjalan sampai menemukan penyerangan atau memberhentikan aplikasi.

2. Use Case Diagram



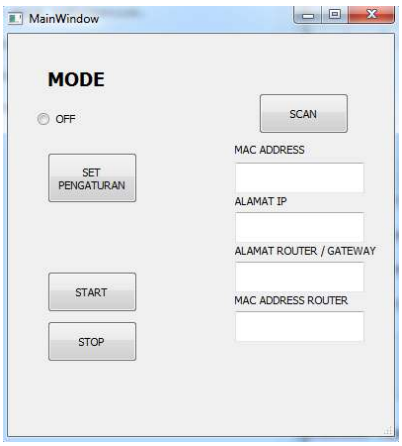
Gambar 5. Hasil Simulasi Penyerangan *Sniffing*

Pada Gambar 5 *Use Case* digambarkan menjadi 5 *Use Case* sesuai dengan kebutuhan user yaitu *Set* Pengaturan *Snort*, Menjalankan IDS *Snort*, *String Matching*, Memeriksa Alamat Jaringan, Memberhentikan IDS *Snort* & *String Matching* serta *Mode Anti Sniffing* digambarkan *include* dari *Use Case IDS Snort & String Matching*

3.4 Implementasi Aplikasi

Berdasarkan rancangan yang telah dibuat, maka akan diimplementasikan ke bentuk aplikasi menggunakan bahasa pemrograman *Python* dan menerapkan metode *String Matching* algoritme KMP (*Knuth Morris Pratt*) untuk mencari pola teks pada *file snort.conf* dan *file logging snort*, sehingga dapat mempermudah konfigurasi IDS *snort* dan memberikan *alert* bahwa komputer pengguna telah disadap.

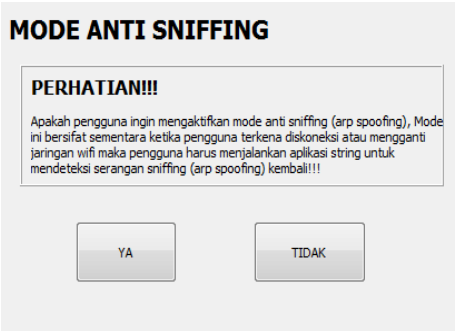
1. Menu Utama
   Menu utama sebagai tampilan awal dari aplikasi. Pada halaman ini user memiliki beberapa tombol untuk digunakan yaitu tombol *set* pengaturan untuk *set* pengaturan IDS *snort* sesuai dengan alamat jaringan yang sedang digunakan , tombol *Start* digunakan untuk menjalankan aplikasi *snort* dan aplikasi *string matching* untuk deteksi serangan *sniffing* (ARP *spoofing*), *Stop* digunakan untuk memberhentikan aplikasi *snort* serta aplikasi *string matching* kemudian menyimpan *log* baru dari *snort* berdasarkan tanggal dan waktu ketika awal mula dijalan sampai memberhentikan aplikasi *snort*, dan *scan* untuk mencek alamat jaringan yang sedang digunakan. Untuk lebih lengkap dapat dilhat pada gambar 6 Implementasi Menu Utama



Gambar 6. Implementasi Menu Utama

Ketika Pengguna membuka aplikasi maka akan menampilkan tampilan utama dari aplikasi yang memiliki beberapa tombol. Sebelum menjalankan aplikasi pengguna harus memiliki aplikasi *snort* dan mengklik tombol *set* pengaturan setiap ingin menjalankan aplikasi.

2. Mode *Anti Sniffing*
   Mode *Anti Sniffing* (ARP *spoofing*) akan tampil ketika pengguna menjalankan aplikasi dan ketika pengguna terdeteksi serangan *sniffing*. Setelah menampilkan *alert* di komputer pengguna maka akan tampil menu mode *anti sniffing* (ARP *spoofing*).



Gambar 7. Implementasi Menu Utama

Pada gambar 7 merupakan tampilan mode *anti sniffing*, aplikasi meminta persetujuan ke pengguna apakah pengguna ingin mengaktifkan mode *anti sniffing* jika iya maka aplikasi merubah arp *cache table* pada komputer menjadi seperti semula dan *mode static* agar tidak dapat di *sniffing*.

3.5 Pengujian Aplikasi

Tahapan pengujian aplikasi ini merupakan bentuk dari teknik menjalankan aplikasi yang telah dirancang dan bertujuan untuk menguji komponen aplikasi yang telah dibuat dapat berfungsi dengan baik. Pengujian pada aplikasi ini menggunakan metode *Black Box* yaitu pengujian yang berfokus pada kebutuhan fungsional dari aplikasi yang telah dibuat. *Text case* ini bertujuan untuk menunjukkan fungsi aplikasi tentang cara beroperasinya, Pengujian telah dilakukan semua fungsi pada menu aplikasi berhasil berjalan.

4. KESIMPULAN

Setelah aplikasi *string matching* untuk deteksi *sniffing* (arp *spoofing*) pada ids snort ini selesai diimplementasikan maka dapat ditarik kesimpulan bahwa *string matching* dengan penerapan algoritme KMP mampu untuk mendeteksi serangan *sniffing* (arp *spoofing*) pada ids *snort* dan menyampaikan informasi ke pengguna bahwa jaringan yang sedang digunakan sedang disadap(*sniffng*), adapun beberapa fitur dari aplikasi yaitu:

1. Aplikasi ini dapat memberikan kemudahan terhadap pengguna untuk mendeteksi serangan *sniffing* (arp *spoofing*).
2. Aplikasi dapat mendeteksi serangan sniffing pada penggunaan semua jaringan wifi publik yang berbeda.
3. Aplikasi dapat mencegah terjadinya serangan *sniffing* terhadap pengguna ketika pendeteksian serangan terjadi.
4. Aplikasi dapat menampilkan alamat jaringan sekarang

DAFTAR PUSTAKA

[1] Vishwa Modi and Asst. Prof. Chandresh Parekh, "*Detection of Rogue Access Point to Prevent Evil Twin Attack in Wireless Network*," Int. J. Eng. Res., vol. V6, no. 04, pp. 23–26, 2017, doi: 10.17577/ijertv6is040102.

[2] S. Susanto, B. A. Pramono, and S. Handayani, "Analisis *Sniffing Password* Menggunakan Aplikasi *Cain Dan Abel* Pada Jaringan Wifi Universitas Semarang," J. Transform., vol. 16, no. 1, p. 67, 2018, doi: 10.26623/transformatika.v16i1.787.

[3] A. T. Haryanto, "Awas, Ini Bahayanya Gunakan Wifi publik," detikinet, Jakarta, Mar. 2017.

[4] BSSN, "Pakai WiFi Gratisan Boleh, Asal Jangan Akses 5 Hal Ini," https://kliksbssn.id/, 2019. .

[5] F. Syahrulah, A. Bhawiyuga, and M. Data, "Implementasi Sistem Pendeteksi *Rogue Access Point* Dengan Metode Perhitungan Nilai *Round Trip Time*," J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 2, no. 12, pp. 7367–7373, 2018.

[6] B. Sudradjat, "Sistem Pendeteksian Dan Pencegahan Penyusup Pada Jaringan Komputer Dengan Menggunakan *Snort* Dan *Firewall*," vol. 1, no. November, pp. 10–24, 2017.

[7] X. Hou, "*The detection and prevention for ARP Spoofing based on Snort*," no. Iccasm, pp. 137–139, 2010.

[8] V. C. B. Ginting, M. Data, and D. P. Kartikasari, "Deteksi Serangan ARP *Spoofing* berdasarkan Analisis Lalu Lintas Paket," vol. 3, no. 5, pp. 5049–5057, 2019.

[9] D. Kurnia, "Pemanfaatan *Bettercap* Sebagai Teknik *Sniffing* Pada Paket Trafik Jaringan WIFI," pp. 83–85, 2019.

[10] A. Ginting, J. Napitupulu, and Jamaluddin, "Sistem Monitoring Pendeteksian Penyusup Menggunakan *Snort* pada Jaringan Komputer Fakultas Ekonomi Universitas Methodist Indonesia," Semin. Nas. Teknol. Inf. dan Komun. 2015, vol. 6, no. September 2015, pp. 1–3, 2015, doi: 10.17605/OSF.IO/W5GT7.

[11] D. Mualfah and I. Riadi, "*Network Forensics For Detecting Flooding Attack On Web Server*," IJCSIS) Int. J. Comput. Sci. Inf. Secur., vol. 15, no. 2, pp. 326–331, 2017, doi: 10.1016/j.ecss.2004.08.013.

[12] E. K. Dewi, "Analisis Log Snort Menggunakan Network Forensic," JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 2, no. 2, 2017, doi: 10.29100/jipi.v2i2.370.

[13] S. Team, "Snort Users Manual 2.9.15," 2019.

[14] R. I. Borman, "Penerapan *String Matching* Dengan Algoritma *Boyer Moore* Pada Aplikasi *Font Italic* Untuk Deteksi Kata Asing," J. Teknoinfo, vol. 10, no. 2, p. 39, 2016, doi: 10.33365/jti.v10i2.9.

[15] A. Fau, Mesran, and G. L. Ginting, "Analisa Perbandingan Boyer Moore Dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial (Studi Kasus : Perpustakaan STMIK Budi Darma)," J. Times (Technology Informatics Comput. Syst., vol. 6, no. 1, pp. 12–22, 2017.

[16] R. K. Hondro, Z. A. Hsb, and R. D. Sianturi, "Aplikasi Penerjemahan Bahasa Mandailing-Indonesia," JURIKOM (Jurnal Ris. Komputer), vol. 3, no. 4, pp. 49–53, 2016.

[17] A. Ariyanto and Asmunin, "Deteksi Paet Sniffing Pada Wirelles Menggunakan ARP Watch," J. Manaj. Inform., vol. 8, no. 2, pp. 178–181, 2018.

[18] Chandra Dirgantara, Rifan Ramadhan, and I Made Suartana, "Implementasi Arp Watch dengan Pfsense untuk mekanisme Pengamanan Acess Point," pp. 67–76, 2020.

[19] D. Srinath, S. P. S.Panimalar, A. J. Simla, and J. D. J.Deepa, "Detection and Prevention of ARP spoofing using Centralized Server," Int. J. Comput. Appl., vol. 113, no. 19, pp. 26–30, 2015, doi: 10.5120/19935-1931.

[20] G. A. V. Chennai, "Detection and Prevention of Arp-Spoofing Attacks," The Times, vol. 8, no. 10, pp. 472–478, 1952.