ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

MITIGASI SERANGAN DNS *CACHE POISONING* PADA *LOCAL AREA NETWORK* BERBASIS *ROUTERBOARD* MIKROTIK

Dian Novianto[1)*], Lukas Tommy[2)], Yohanes Setiawan Japriadi[3)], Sujono[4)]
[1), 2), 3)] Program Studi Teknik Informatika, ISB Atma Luhur
[4)] Program Studi Sistem Informasi, ISB Atma Luhur
Email :diannovianto@atmaluhur.ac.id[1)], lukastommy@atmaluhur.ac.id[2)], ysetiawanj@atmaluhur.ac.id[3)], sujono@atmaluhur.ac.id[4)]*

ABSTRACT

Domain Name System (DNS) enables users to access websites via domain names, offering convenient navigation. However, criminals can exploit this convenience to redirect connections from user devices to fake servers for a variety of purposes. A form of attack known as DNS cache poisoning exploits vulnerabilities in the Domain Name System (DNS) to redirect connections from a legitimate website address to an illegitimate one. As a consequence of the fact that users of networked systems are typically unaware that they are accessing an illegitimate site, this attack can have a particularly damaging impact. Such incidents can give rise to several issues, including the theft of data, the distribution of malware, and other security threats. To address this issue, this study employs a firewall in conjunction with DoH (DNS over HTTPS) and the utilization of registered certificates. The Domain Name System over HTTPS (DoH) protocol encrypts DNS requests and responses, preventing third parties (such as attackers who manipulate DNS) from reading or modifying DNS requests. The data employed in this study is derived from a review of existing literature. Furthermore, this study employs the PPDIOO model (Preparation, Planning, Design, Implementation, Operation, and Optimization) for the development of the network. Furthermore, the Mikrotik RB951ui-2Hnd routerboard is employed in this study. The outcome is the implementation of a multifaceted security strategy that effectively mitigates DNS cache poisoning attacks by 100%, while simultaneously reducing CPU usage to 11.5%. This approach enhances the security and reliability of user search activities on the internet.
Keywords : *DNS cache poisoning , DNS over https, Mikrotik.*

ABSTRAK

DNS memungkinkan pengguna menggunakan nama domain untuk mengakses sebuah website, tetapi kemudahan ini dapat digunakan oleh pelaku kejahatan untuk mengalihkan koneksi dari perangkat pengguna ke server palsu dengan berbagai tujuan. Jenis serangan yang disebut DNS Cache Poisoning menggunakan kerentanan dalam Domain Name System (DNS) untuk mengarahkan koneksi dari suatu alamat website yang sah menuju alamat website yang tidak sah. Karena pengguna jaringan tidak menyadari bahwa mereka sedang mengunjungi situs yang salah, serangan ini dapat sangat merugikan. Hal ini dapat menyebabkan banyak masalah, seperti pencurian data, penyebaran malware, dan ancaman keamanan lainnya. Untuk memecahkan masalah ini, penelitian ini menggunakan firewall yang dikombinasikan dengan DoH (DNS over HTTPS) dan penggunaan sertifikat yang terdaftar. DoH mengenkripsi permintaan dan respons DNS menggunakan protokol HTTPS, sehingga pihak ketiga (seperti penyerang yang mencampur DNS) tidak dapat membaca atau mengubah permintaan DNS. Data yang digunakan dalam penelitian ini berbasis studi literatur. Dan pada penelitian ini, pengembangan di jaringan mengikuti model PPDIOO (Persiapan, Perencanaan, Desain, Implementasi, Operasi, dan Optimalisasi) digunakan. Selain itu, routerboard mikrotik RB951ui-2Hnd digunakan dalam penelitian ini. Hasilnya adalah penerapan kombinasi beberapa teknik keamanan tersebut menghasilkan sebuah sistem yang bisa membatasi serangan DNS cache Poisoning 100%, dengan hanya 11,5% penggunaan CPU sehingga membuat aktifitas pencarian yang dilakukan pengguna di internet jadi lebih aman.
Kata Kunci : *DNS cache poisoning , DNS over https, Mikrotik.*

1. Pendahuluan

Di zaman digital sekarang, internet berfungsi sebagai fondasi untuk berbagai kegiatan sehari-hari, baik dalam bisnis, pendidikan, atau komunikasi pribadi. Salah satu elemen penting yang mendukung konektivitas internet adalah Sistem Nama Domain (DNS), yang berfungsi mengubah nama domain yang mudah diingat oleh manusia menjadi alamat IP yang dipakai perangkat untuk berkomunikasi. Sistem DNS ini dirancang untuk memberikan resolusi nama secara cepat dan efisien melalui penggunaan *cache*, atau penyimpanan sementara, yang menyimpan data DNS yang sering diakses. DNS *cache* yaitu penyimpanan sementara yang memuat data DNS yang sering diakses. *Cache* ini memungkinkan resolusi DNS yang lebih cepat tanpa perlu menghubungi server DNS asli setiap kali permintaan diajukan.

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

Namun, seperti sistem jaringan lainnya, DNS juga memiliki sejumlah kerentanan yang dapat dieksploitasi oleh penyerang. Salah satu bentuk serangan yang signifikan adalah DNS *Cache Poisoning* atau dikenal juga sebagai DNS *Spoofing*. DNS *Cache Poisoning* merupakan serangan yang berpotensi merusak, di mana penyerang menyuntikkan data palsu ke dalam *cache* DNS server. Hal ini menyebabkan server DNS memberikan respons yang salah, sehingga pengguna diarahkan ke alamat IP yang berbeda dari yang sebenarnya. Efek dari serangan ini dapat sangat merugikan, mulai dari pengalihan lalu lintas ke situs berbahaya, pencurian data sensitif, hingga penyebaran *malware* secara massal. Salah satu akibat dari meracuni DNS *Cache* yang paling berbahaya adalah karena dapat menyebar dari server DNS ke server DNS (P. Cisar and R. Pinter, 2019). Menurut *Global DNS Threat Report*, pada tahun 2021 serangan dns *hijacking* ini sebanyak 27% dan pada tahun 2022 sebanyak 28%, itu artinya kejahatan *cyber* ini terus meningkat dari tahun ke tahun (R. Fouchereau, 2022).

Serangan DNS *Cache Poisoning* dapat berdampak serius, karena memungkinkan penyerang untuk mengalihkan lalu lintas internet, mencuri data sensitif, menyebarkan *malware*, hingga melakukan serangan phishing. Ketika server DNS telah terinfeksi dengan data palsu ini, pengguna yang meminta akses ke situs tertentu dapat diarahkan ke situs berbahaya yang dikendalikan oleh penyerang tanpa sepengetahuan mereka. Oleh karena itu, memahami cara kerja DNS *Cache Poisoning* serta metode mitigasi, seperti DNS *over* HTTPS (DoH), sangat penting dalam memperkuat keamanan jaringan dan melindungi pengguna dari serangan siber.

Perkembangan teknologi keamanan jaringan, seperti DNS *over* HTTPS (DoH), bertujuan untuk mengatasi serangan DNS *Cache Poisoning* . DoH adalah protokol yang mengenkripsi permintaan dan respons DNS menggunakan protokol HTTPS. Dengan DoH, permintaan DNS dikirim melalui koneksi terenkripsi menggunakan port HTTPS standar (443), sehingga tidak mudah dibaca atau dimodifikasi oleh pihak ketiga, seperti penyerang yang mencoba melakukan serangan DNS *spoofing* atau *man in the middle* (MitM).

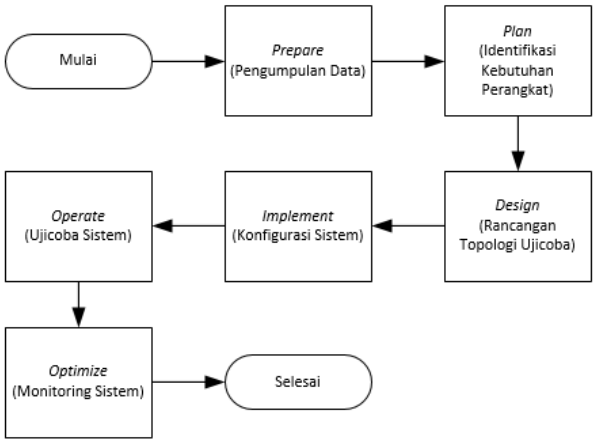
Selain itu *Firewall* pada mikrotik dapat dimanfaatkan untuk melakukan pengamanan tambahan. *Firewall* adalah sebuah sistem atau perangkat keamanan, khususnya dalam jaringan komputer, yang bertanggung jawab untuk menjamin keamanan transmisi data pada jaringan komputer (A. Noviriandini et al, 2022). *Firewall* ini nantinya diharapkan bisa memberikan keamanan pada jaringan terhadap serangan dns *cache poisoning* dengan cara dilakukannya *filtering* permintaan alamat dns selain dari dns *resolver* yang telah ditentukan.

Studi ini menerapkan model pengembangan jaringan PPDIOO (Persiapkan, Rencanakan, Rancang, Terapkan, Operasikan, dan Optimalkan). PPDIOO merupakan pendekatan untuk mendesain dan mengembangkan jaringan yang diciptakan oleh Cisco. (Imam Solikin, 2017), dan akan memberikan langkah-langkah penting untuk keberhasilan perancangan jaringan (Dian, 2020).

Studi ini diharapkan mampu menunjukkan sebuah gambaran baru mengenai metode mitigasi DNS *Cache Poisoning* yang lebih efektif, sehingga dapat meningkatkan keamanan jaringan dan melindungi pengguna dari ancaman siber yang terus berkembang.

A. Metode Penelitian

Studi ini menerapkan metode kualitatif, di mana peneliti berfungsi sebagai instrumen utama dalam proses pengumpulan data (Dian, 2022). Metode ini diterapkan sebagai metode pengumpulan data dan pencarian solusi untuk permasalahan (Marinu, 2023). Peneliti mengumpulkan data dengan mencari sumber dari jurnal akademik dan buku. Dengan demikian, peneliti dapat memahami serta mengerti metode atau teknik yang akan diterapkan, sehingga diharapkan pengembangan sistem jaringan akan berjalan dengan lancar di masa mendatang. Menyiapkan, merancang, membangun, menerapkan, menjalankan, dan mengoptimalkan adalah beberapa langkah yang perlu diikuti oleh peneliti saat mengembangkan sistem menggunakan model PPDIOO ini, adapun alurnya dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

1. *Prepare* (Persiapan)

Pada tahap ini, penulis akan menyiapkan peralatan dan bahan-bahan atau studi literatur dengan mengumpulkan referensi jurnal dari lima tahun terakhir untuk membantu mereka menjalankan penelitian. Judul penelitian yang direferensikan antara lain:

a. Penelitian pada tahun 2021 dengan judul *Implementation of a Network Security System Using the Simple Port Knocking Method on a Mikrotik-Based Router* (Dian, 2021).
b. Penelitian pada tahun 2018 mengenai *DNS Cache Poisoning : A Review on its Technique and Countermeasures* (Dissanayake, 2023).
c. Penelitian pada tahun 2020 dengan judul *DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels* (Man, keyu dkk, 2020).

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

d. Penelitian pada tahun 2020 dengan judul Poison *over* troubles forwarders: A *cache poisoning* attack targeting DNS forwarding devices (Zheng, 2020).
e. Penelitian pada tahun 2022 dnegan judul Analisis Keamanan Jaringan Pada Jaringan Wireless Dari Serangan Man In The Middle Attack DNS *Spoofing* (Pangestu,dkk. 2022).

2. *Plan* (Perencanaan)
Tabel 1 dan 2 akan menampilkan spesifikasi dari perangkat keras maupun dari perangkat lunak yang diperlukan oleh peneliti untuk melakukan studi tentang *domain name system cache poisoning* selama proses pengembangan sistem pada penelitian berlangsung:

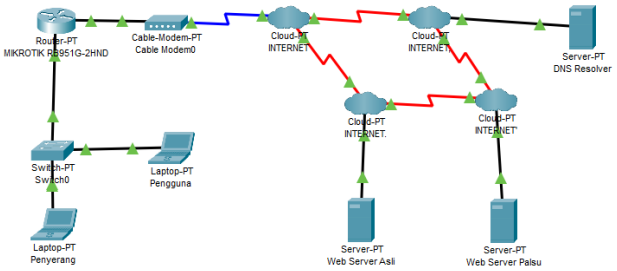
Tabel 1. Kebutuhan *Hardware*

No	Perangkat Keras	Keterangan	Jumlah
1	*Routerboard* Mikrotik RB951G-2HND	Router yang digunakan pada jaringan LAN	1
2	Laptop	Perangkat yang digunakan pengguna dan penyerang	2
3	*Switch*	Digunakan pada LAN sebagai penghubung	1
4	Kabel UTP siap pakai	Media koneksi di jaringan LAN	4

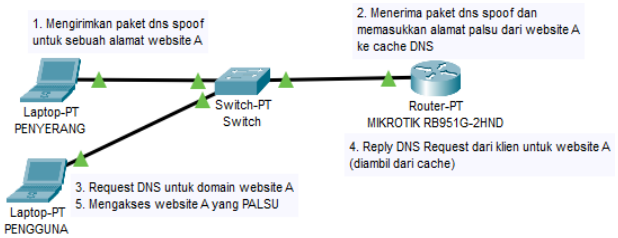
Tabel 2. Kebutuhan *Software*

No	Perangkat Lunak	Keterangan	Jumlah
1	Sistem operasi *windows* 8.1	Sistem operasi pengguna	1
2	Sistem operasi *Kali linux* 2023.1	Sistem operasi penyerang	1
3	*Cisco Packet Tracer* 7.3.0	Digunakan untuk mendesain topologi jaringan	1
4	*Winbox* v3.41	Digunakan untuk melakukan *remote access* ke router	1
5	*MikroTik RouterOS* v6.48	Sistem operasi *routerboard* mikrotik	1
6	*Ettercap* DNS	Aplikasi dns *spoofing*	1

3. *Design* (Desain)
Setelah proses perencanaan rampung, langkah berikutnya adalah desain, yang dilakukan dengan menerapkan model ppdioo. Peneliti memanfaatkan perangkat lunak simulator dari cisco dengan versi 7.3.0 untuk merancang simulasi topologi jaringan.
Gambar 2 menunjukkan topologi dari jaringan yang berjalan saat dilakukannya penelitian ini.

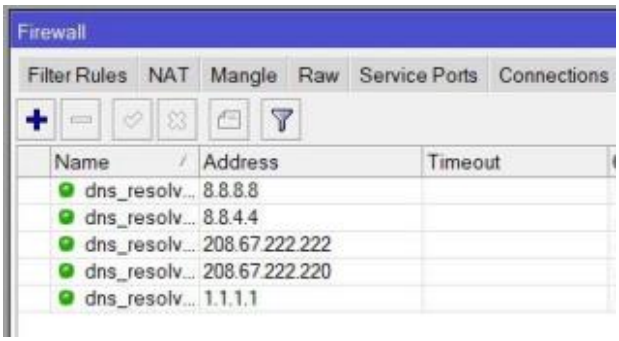


Gambar 2. Topologi Simulasi Berlangsung



Gambar 3. serangan *cache poisoning*

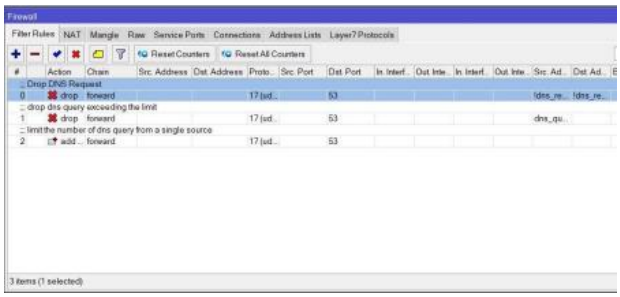
Dari gambar nomor 3 terlihat berupa serangan yang terjadi dimulai dari perangkat yang mengirimkan dns *spoofing* ke router, sehingga router akan mengupdate dns *cache* untuk alamat tertentu, lalu saat pengguna meminta informasi mengenai dns dari sebuah website yang sudah ditargetkan oleh penyerang, router akan membalas dengan mengirimkan informasi yang diminta, dan pengguna akan mengunjungi website tersebut yang ternyata merupakan website palsu.
4. *Implement* (*Implement*asi)
Dalam fase urutan keempat ini, pengaturan pengamanan terhadap *router* diimplementasikan berdasar pada analisa di bagian sebelumnya, dengan cara menerapkan penyaringan paket yang lewat pada *firewall* untuk memblokir proses permintaan dns yang sumber maupun tujuannya selain dari *resolver* yang sudah ditetapkan. Proses ini dimulai dari menambahkan alamat dns *resolver* yang akan digunakan pada *AddressList Mikrotik*. Dalam penelitian ini dns *resolver* yang digunakan dalam jaringan LAN merupakan dns dari google dengan alamat 8.8.8.8 dan 8.8.4.4, lalu dari *cloudflare* dengan alamat 1.1.1.1 dan dari OpenDNS dengan alamat 208.67.222.222 dan 208.67.220.220.



Gambar 4. Konfigurasi *Address List*

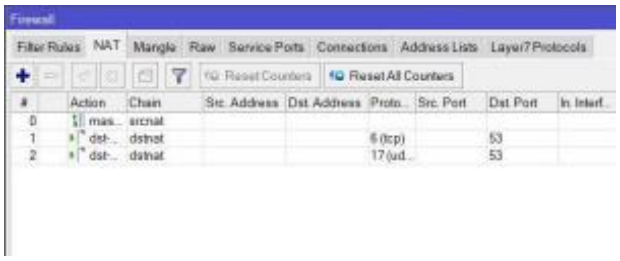
Gambar 4 menunjukkan tampilan setelah penambahan alamat dns *resolver* pada router yang digunakan, adapun nama yang digunakan pada address list adalah dns_*resolver*. Penambahan address lists untuk DNS *resolver* ini dapat membantu meningkatkan keamanan jaringan dan memberikan kontrol lebih besar atas lalu lintas DNS.
Selanjutnya pada firewall ditambahkan tiga *filter rule* untuk memblokir permintaan dns yang datang dari sumber yang tidak diizinkan, menambahkan filter *rule* untuk membatasi jumlah permintaan dns yang diterima dari satu sumber dalam waktu tertentu.

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786



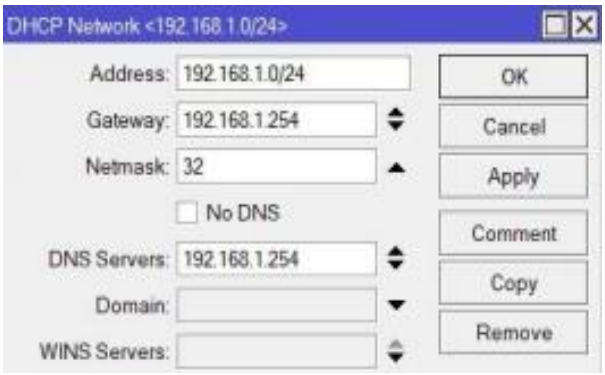
Gambar 5. *Filter Rule* DNS

Dari Gambar 5 dapat terlihat *rule* yang telah dibuat pada firewall mikrotik, Tujuan dari *rule* pertama dimana saat ada permintaan *resolve* domain yang sumber dan tujuannya selain dari dns server yang sudah ditentukan, maka permintaan tersebut akan ditolak dan paket *request* akan di *drop*. Lalu untuk *rule* kedua bertujuan untuk membatasi paket *query* dns dari satu sumber, dan apabila melebihi normal maka paket akan di *drop*, hal ini dilakukan agar tidak terjadi paket *flood* di jaringan. Sedangkan *rule* ketiga bertujuan untuk memasukkan sumber berupa alamat ip yang melakukan *query* dns melebihi normal ke dalam *address list* untuk kemudian di blok oleh *rule* kedua, dengan demikian admin jaringan tidak harus selalu memantau trafik *resolve* dns yang ada di dalam router. Selanjutnya dibuat dua *rule* pada NAT dengan *chain dstnat*, yang berarti alamat tujuan akan diubah setelah memasuki router. Adapun *rule dstnat* bisa di lihat di tampilan pada gambar 6.



Gambar 6. Destination NAT

Dari gambar 6, terlihat dua protokol yang akan masuk dalam *rule* dstnat, yaitu protokol tcp dan udp dengan port 53, dimana port 53 merupakan *port* bagi lalu lintas dns di jaringan. Tujuan dari *rule* ini adalah memaksa pengguna menggunakan alamat dns lokal dari router, sehingga meskipun pengguna jaringan menggunakan dns *server* lain yang diatur secara manual pada perangkat, maka secara otomatis akan diarahkan ke alamat yang telah diatur pada *action* dstnat.
Selanjutnya dilakukan modifikasi *netmask* yang digunakan di jaringan LAN, seperti yang terlihat pada gambar 7.



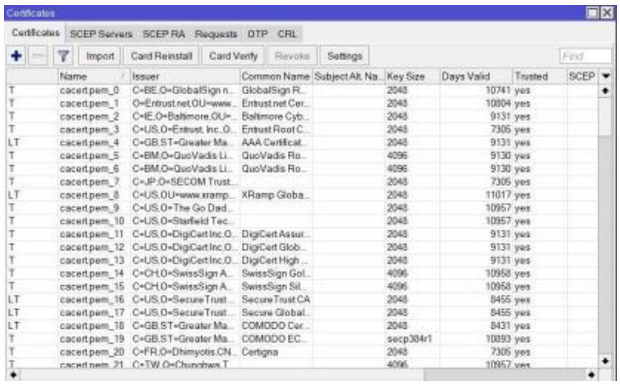
Gambar 7. *Netmask*

Pada gambar 7 terlihat *netmask* yang digunakan adalah *prefix* 32, hal ini berarti trafik jaringan akan berjalan secara individual pada masing – masing perangkat pengguna. Trafik dari perangkat akan berhubungan secara langsung dengan alamat *gateway* pada router yang bertujuan untuk keamanan jaringan dengan cara mengisolasi koneksi perangkat klien dari sistem lainnya yang ada di jaringan. Hal ini hanya akan mengizinkan lalu lintas ke tujuan yang ditetapkan secara eksplisit oleh rute yang ada pada sistem.
Selnjutnya dilakukan konfigurasi untuk mengaktifkan fitur DoH dan *verify certificate* pada router mikrotik yang bertujuan untuk meningkatkan keamanan dan privasi saat menggunakan DNS, dan dapat memastikan bahwa lalu lintas DNS aman dan tidak disusupi oleh pihak yang tidak berwenang karena sudah terenkripsi. Adapun menu DOH dapat diakses dari DNS Server seperti gambar 8.

Gambar 8. DOH (DNS *over* HTTPS)

Dari gambar 8 dapat terlihat bahwa parameter yang digunakan antara lain: dns *server*, *server* DoH, dan *verify certificate*. DoH yang digunakan dalam ujicoba ini milik *cloudflare*, yang memang menyediakan fasilitas DoH secara gratis, dengan alamat ip dns *server* di 1.1.1.1, dan alamat doh di https://cloudflare-dns.com/dns-query.
Lalu memastikan sertifikat keamanan dari DoH valid, maka perlu ditambahkan list sertifikat dari penyedia yang sah, dan di *import* kedalam router seperti yang terlihat pada gambar 9.

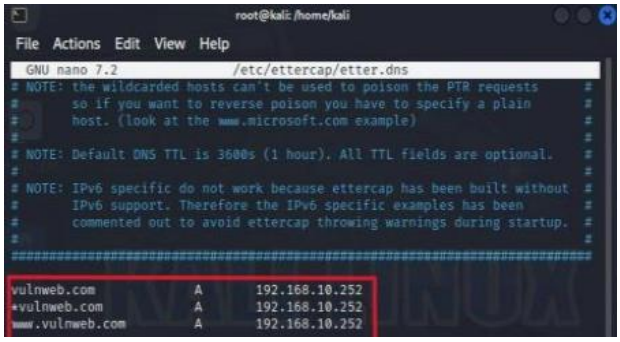
ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786



Gambar 9. List Sertifikat DoH

Dari gambar 9 dapat terlihat bahwa sertifikat berhasil di import kedalam router, dengan demikian konfigurasi DoH pada router telah selesai.

5. *Operate* (Operasi)
Pada fase atau tahap operasional, penulis melaksanakan simulasi dan pengaturan terhadap teknik pencegahan yang telah dipilih.. Simulasi dan pengaturan ini dilaksanakan melalui aplikasi *ettercap* yang telah ada dalam sistem operasi kali linux untuk melakukan serangan *man in the middle* dan dns *spoofing* atau meracuni *cache* dns, tahapan pertama dalam proses ini adalah membuat tampilan dari web yang berhasil dilakukan dns *spoofing*, lalu yang selanjutnya adalah menargetkan halaman web yang akan di *spoofing*, adapun tampilan konfigurasi web target pada ettercap terlihat seperti pada gambar 10.

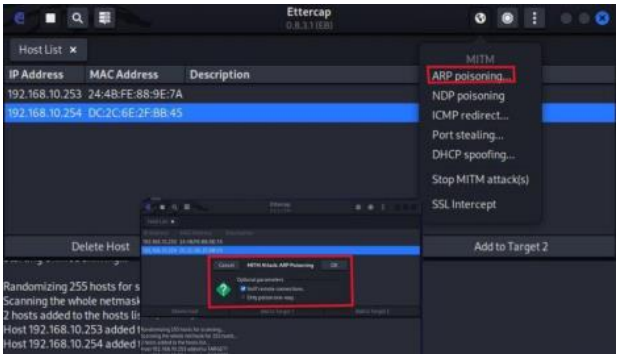


Gambar 10. Konfigurasi dns *poison*

Pada gambar 10 terlihat web yang akan menjadi target *spoofing* dalam simulasi ini adalah vulnweb.com, website ini memang digunakan oleh banyak penguji keamanan sistem sebagai target. Beberapa parameter yang di isi pada file konfigurasi adalah domain, type *record* dan alamat ip.
Selanjutnya untuk melakukan dns *poisoning* menggunakan *ettercap*, dapat dibuka aplikasinya yang berbasis GUI untuk mempermudah dalam melakukan ujicoba. Adapun tampilan utama dari aplikasi *ettercap* dapat dilihat pada gambar 11.

Gambar 11. Tampilan *Ettercap*

Dari gambar 11 dapat dilihat ada beberapa parameter yang bisa diatur, dalam kasus ini *interface*/adapter jaringan yang digunakan adalah eth0 dan sniffing dilakukan saat aplikasi dijalankan.
Selanjutnya dilakukan serangan MITM (*Man in the middle attack*) untuk merubah arah koneksi dari perangkat klien menuju ke *server* web palsu, seperti yang terlihat pada gambar 12.



Gambar 12. *Man in the middle attack*

Dari gambar 12 terlihat proses serangan ini memanfaatkan arp *poisoning* sebelum dilakukannya dns *spoofing*. Meracuni arp pada ettercap memiliki tujuan agar dapat memantau lalu lintas data yang dikirim oleh perangkat pengguna ke alamat *gateway* atau router, dengan demikian ettercap dapat mengubah isi dari data yang dikirim dalam jaringan sebelum meneruskan data tersebut ke server yang sebenarnya. Selanjutnya setelah arp *poisoning* , dilakukan serangan dns *poisoning* seperti yang terlihat pada gambar 13.


Gambar 13. Proses Serangan DNS *Poisoning*

Dari gambar 13 terlihat bahwa dns *poisoning* di *plugin* ettercap telah dijalankan, dengan keterangan *activating dns poisoning* . Dengan aktifnya serangan ini, maka saat pengguna mengakses alamat web vulnweb.com, kemudian *ettercap* yang menjalankan arp *poisoning*

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

melihat informasi tersebut, maka fitur dns *poisoning* akan bekerja dengan cara mengarahkan data tersebut ke server palsu.

6. *Optimize* (Optimasi)
Tahap Optimasi dilakukan setelah jaringan mulai berfungsi. Langkah ini akan melibatkan pemantauan kinerja pada sumber daya router setelah penerapan teknik keamanan.

2. Hasil dan Pembahasan

Pada bagian ini akan ditampilkan hasil dari simulasi serangan *dns cache poisoning* yang dilakukan oleh laptop penyerang ke jaringan LAN dengan tujuan agar pengguna diarahkan menuju ke web server palsu. Adapun hasilnya sebelum konfigurasi keamanan di terapkan dapat dilihat pada gambar 14.

Gambar 14. Hasil percobaan serangan

Dari gambar 14 terlihat bahwa ettercap berhasil merubah trafik pada jaringan, dimana saat pengguna mengakses alamat vulnweb.com, maka tampilannya menunjukkan tampilan hasil buatan penyerang, tentunya hal ini akan sangat berbahaya jika tampilan yang dibuat menyerupai website aslinya karena dapat digunakan untuk tindak kejahatan seperti pencurian data.
Selanjutnya saat konfigurasi DoH berhasil dilakukan, untuk melihat hasilnya dapat melalui alamat ip 1.1.1.1/help, hasilnya dapat dilihat pada gambar 15.

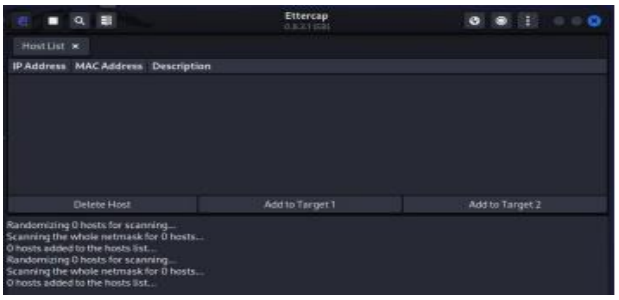
Gambar 15. Hasil verifikasi DoH

Dari gambar 15 terlihat keterangan bahwa jaringan yang digunakan telah berhasil menggunakan DoH (DNS *over* HTTPS) milik c*loudflare* dengan keterangan *yes*, dengan demikian, pengguna yang akan melakukan resolve alamat dns di jaringan akan lebih aman, karena trafik tersebut telah dienkripsi.

Lalu saat konfigurasi keamanan dan DoH telah diterapkan, pengguna dapat mengakses alamat web tanpa dibelokkan ke web server palsu, hal ini dapat dilihat pada gambar 16.

Gambar 16. Website vulnweb asli

Dari gambar 16, pengguna dapat mengakses website asli dengan tampilan yang berbeda saat pengguna berhasil di arahkan menuju web server palsu. Dan DoH tidak membuat ada yang berbeda saat pengguna melakukan aktifitas *browsing* meskipun trafik dns telah di enkripsi.
Selanjutnya saat dilakukan percobaan menggunaan aplikasi ettercap setelah konfigurasi keamanan diterapkan, ettercap tidak berhasil menemukan target seperti yang terlihat pada gambar 17.



Gambar 17. *Scanning host*

Dari gambar 17, terlihat dari keterangan ettercap yang telah melakukan scanning tidak berhasil menemukan host yang merupakan pengguna di jaringan tersebut, sehingga tidak ada yang dapat dijadikan target dari arp dan dns *spoofing*. Kemudian pada penggunaan sumber daya pada router yang dipantau selama proses pengujian terlihat adanya peningkatan signifikan pada penggunaan CPU saat serangan berlangsung dan berkurang saat teknik keamanan diterapkan. Hasilnya dapat dilihat pada tabel 3.

Tabel 3. Penggunaan *Resource*

Profile	*Resource* Sebelum Penerapan Keamanan	*Resource* Sesudah Penerapan Keamanan
CPU	51,5 %	11,5 %
DNS	13 %	1 %
Firewall	7,5 %	3 %
Firewall-mgmt	1 %	0 %

Dari tabel 3 dapat diketahui saat pengujian serangan dns *cache poisoning* berlangsung penggunaan *resource* CPU pada router meningkat cukup tinggi, mencapai 51,5%,

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

dimana DNS mengambil sebesar 13%. Namun saat sudah diterapkan konfigurasi keamanan sehingga serangan dns *spoofing* atau dns *poisoning* tidak dapat dilakukan, penggunaan CPU menjadi lebih rendah, hanya sebesar 11% dan dns mengambil resource hanya 1%.

Hal Ini menunjukkan bahwa sumber daya pada router tetap stabil ketika teknik pencegahan ini diterapkan. Tentu saja, hal ini juga dipengaruhi oleh aktivitas lain, tetapi dalam skenario simulasi ini, router berfungsi dengan baik dan pengguna dapat menjelajahi internet dengan lebih aman tanpa gangguan dari DNS *cache poisoning*.

3. Kesimpulan

Setelah proses pada penelitian ini selesai dilaksanakan, penulis dapat menarik beberapa kesimpulan dari penelitian ini, antara lain sebagai berikut: Berhasil menerapkan *firewall* sebagai pelindung jaringan dari serangan dns *cache poisoning* dengan menambahkan aturan *filter*, daftar alamat, NAT, serta mengubah *netmask* pada server dhcp di *routerboard mikrotik*. Kombinasi dari penerapan DoH (DNS *over* HTTPS) untuk memverifikasi sertifikat keamanan bersama *firewall* pada *routerboard mikrotik*, telah berfungsi dengan baik, dapat diverifikasi dan tidak mengganggu aktivitas *browsing* pengguna. Sumber daya router tidak terpengaruh secara substansial ketika metode keamanan yang dipilih diterapkan pada jaringan yang aktif, hal ini terlihat dari hasil pemantauan pada bagian hasil.

Daftar Pustaka

A. Noviriandini et al. (2022). Analisis Management Bandwidth Dan Firewall Dengan Router Mikrotik Pada Pt . Bca Multifinance. Vol. 1 (No. 3), 40–45.

Dian Novianto, Tri Sugihartono. (2020). Sistem Deteksi Kualitas Buah Jambu Air Berdasarkan Warna Kulit Menggunakan Algoritma Princ*ip*al Component Analysis (Pca) dan K-Nearest Neigbor (K-NN). Jurnal Ilmiah Informatika Global Vol.11(2).

Dian Novianto, dkk. (2022). Implementasi Keamanan Akses Terhadap Website Menggunakan Wireguard VPN Di *Router*board Mikrotik. Jurnal Ilmiah Informatika Global, Vol. 13(2).

Imam Solikin. (2017). Penerapan Metode PPDIOO dalam Pengembangan LAN dan WLAN. Teknomatika, Vol.07 (No.01), 65-73.

I. M. M. Dissanayake. (2018). *DNS Cache Poisoning: A Review on its Technique and Countermeasures*. Natl. Inf. Technol. Conf. NITC 2018, pp. 1–6, doi: 10.1109/NITC.2018.8550085.

K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan. (2020). *DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels*. Proc. ACM Conf. Comput. Commun. Secur, pp. 1337–1350, doi: 10.1145/3372297.3417280.

Marinu Waruwu. (2023). Pendekatan Penelitian Pendidikan: Metode Penelitian Kualitatif, Metode Penelitian Kuantitatif dan Metode Penelitian Kombinasi (*Mixed Method*). Jurnal Pendidikan Tambusai: Vol. 7 (1), 2896-2910.

Novianto, Dian dkk. (2021). *Implementation of a Network Security System Using the Simple Port Knocking Method on a Mikrotik-Based Router*. Jurnal Komputer, Informasi dan Teknologi vol.1 (No.2).

P. Cisar and R. Pinter. (2019). *Technical and Educational Sciences jATES Some ethical hacking possibilities in Kali Linux environment*. Vol. 9(4), pp. 129–149.

R. Fouchereau. (2022). *Securing Anywhere Networking*. June, pp.1–18, [Online]. Available: https://efficientip.com/wpcontent/uploads/2022/10/IDC-EUR149048522-EfficientIPinfobrief_FINAL.pdf

T. Pangestu and R. Liza. (2022). Analisis Keamanan Jaringan Pada Jaringan Wireless Dari Serangan Man In The Middle Attack DNS Spoofing. JiTEKH, vol. 10 (No.2), 60–67, doi: 10.35447/jitekh.v10i2.571.

X. Zheng et al. (2020). *Poison over troubled forwarders: A cache poisoning attack targeting DNS forwarding devices*. Proc. 29th USENIX Secur. Symp., pp. 577–593.