

IMPLEMENTASI DAN ANALISIS *NETWORK INTRUSION DETECTION SYSTEM (NIDS)* UNTUK MONITORING JARINGAN INTRANET

Meisi Sulfarita

Prodi Rekayasa Sistem Komputer Fakultas Ilmu Komputer Universitas Dehasen Bengkulu
meisisulfarita1@gmail.com

ABSTRACT

This research aims to build a network intrusion detection system based on the Network Intrusion Detection System (NIDS) and implement a notification system on Android using a telegram bot application to receive monitoring results from NIDS quickly. This research is using experimental method. In this study, the implementation of the Network Intrusion Detection System (NIDS) and notification system using the Telegram application on an Android smartphone was carried out. The experimental results are then documented to conduct analysis so that appropriate recommendations are made for NIDS with notifications using the Telegram application on Android. From the results of the analysis, conclusions will be obtained regarding the benefits, functions and advantages of the system that has been built. The results of this study indicate that the Network Intrusion Detection System (NIDS) can detect ICMP flooding and UDP flooding attacks on the Intranet network. The results of the attack are sent to the network admin using a telegram bot notification system so that the admin can receive notifications of ICMP and UDP flooding attacks using a bash shell-based script. From the results of attack detection, there is a time difference of 5 minutes for the detection of UDP flooding attacks by the NIDS server, while for ICMP flooding attacks there is no time difference.

Keywords : *Linux, NIDS, android, network, flooding data.*

ABSTRAK

Penelitian ini bertujuan untuk membangun sistem pendeteksi gangguan dalam jaringan berbasis *Network Intrusion Detection System (NIDS)* serta menerapkan sistem notifikasi pada Android menggunakan aplikasi bot telegram untuk menerima hasil pemantauan dari NIDS secara cepat. Penelitian ini menggunakan metode eksperimen. Pada penelitian ini dilakukan Implementasi *Network Intrusion Detection System (NIDS)* serta sistem notifikasi menggunakan aplikasi telegram pada smartphone android. Hasil eksperimen selanjutnya didokumentasikan untuk melakukan analisis sehingga dihasilkan rekomendasi yang tepat untuk NIDS dengan notifikasi menggunakan aplikasi telegram pada Android. Dari hasil analisis tersebut nantinya akan mendapatkan kesimpulan mengenai manfaat, fungsi serta kelebihan dari sistem yang sudah dibangun. Hasil dari penelitian ini menunjukkan bahwa *Network Intrusion Detection System (NIDS)* dapat mendeteksi serangan *ICMP flooding* dan *UDP flooding* pada jaringan Intranet. Hasil serangan tersebut dikirimkan kepada admin jaringan menggunakan sistem notifikasi telegram bot sehingga admin dapat menerima notifikasi serangan *ICMP* dan *UDP flooding* menggunakan script berbasis *bash shell*. Dari hasil deteksi serangan, terdapat selisih waktu sebesar 5 menit untuk deteksi serangan *UDP flooding* oleh *NIDS server*, sedangkan untuk serangan *ICMP flooding* tidak ada selisih waktu.

Kata Kunci : *Linux, NIDS, android, network, flooding data.*

1. Pendahuluan

Penggunaan jaringan komputer sebagai media komunikasi, baik dalam jaringan intranet maupun internet, telah mengalami perkembangan yang pesat dalam beberapa tahun terakhir. Jaringan komputer telah menjadi komponen yang sangat penting dalam perkembangan teknologi

informasi, karena semua aspek dalam dunia teknologi informasi mengandalkan jaringan komputer sebagai media komunikasi antara pengguna teknologi informasi tersebut.

Dalam konteks penerapan jaringan komputer, aspek keamanan memiliki peran yang sangat krusial. Keamanan jaringan komputer menjadi fokus utama guna mencegah potensi eksploitasi sumber daya jaringan yang tidak sah dan mengantisipasi ancaman yang dapat merusak jaringan, baik secara langsung maupun tidak langsung. Upaya ini bertujuan untuk menjamin kelancaran arus data dalam jaringan komputer, dengan tujuan utama menghindari serangan yang dapat mengakibatkan peningkatan *bandwidth*, keterlambatan (*delay*), variabilitas dalam waktu pengiriman (*jitter*), serta tingkat kinerja (*throughput*) yang tidak optimal dalam jaringan tersebut.

Upaya pencegahan yang paling sering dilakukan adalah dengan menempatkan seorang administrator jaringan, namun seorang administrator tidak mungkin melakukan pengawasan secara langsung terus menerus, ada kalanya administrator jaringan membutuhkan istirahat. Untuk mengatasi masalah ini, dapat diterapkan suatu sistem deteksi terhadap ancaman ataupun gangguan di dalam jaringan komputer yang disebut dengan *Network Intrusion Detection System* (NIDS). NIDS ini merupakan teknik yang dapat digunakan untuk melihat ataupun memantau *traffic* keluar dan masuk ataupun *traffic* di antara *host* atau di antara segmen jaringan lokal.

Dalam melakukan pemantauan menggunakan NIDS ini, seorang administrator jaringan dapat membuka *log* dari NIDS yang terletak pada NIDS *server* baik menggunakan perintah *Command Line* (CLI) ataupun *log* yang sudah di integrasikan pada aplikasi berbasis web. Hal ini juga menuntut seorang administrator untuk selalu siap di depan komputer yang tentunya ini tidak bisa dilakukan oleh seorang administrator jaringan. Untuk memudahkan dalam memantau hasil dari NIDS *server* ini dapat menggunakan sistem notifikasi yang dapat dikirimkan langsung kepada administrator jaringan. Mengingat saat ini hampir semua kalangan menggunakan *smartphone* dengan sistem operasi Android, maka sistem notifikasi dari NIDS ini dapat dikirimkan ke *smartphone* Android dari administrator jaringan.

Salah satu aplikasi Android yang dapat digunakan sebagai media untuk menerima notifikasi adalah bot telegram, dimana bot telegram ini adalah salah satu fitur *Application Programming Interface* (API) dari aplikasi telegram, dimana dengan memanfaatkan fasilitas ini dapat memudahkan user untuk melakukan proses otomatisasi terhadap sebuah sistem, sebab bot telegram ini merupakan *platform* aplikasi tambahan yang memiliki berbagai fungsi tersendiri yang bisa dimanfaatkan oleh pengguna Telegram dengan mengirimkan perintah melalui format tersendiri. Sehingga dengan menggunakan bot telegram ini dapat mengoptimalkan notifikasi dari pemantauan NIDS ke *smartphone* Android administrator jaringan.

2. Tinjauan pustaka

A. Sistem Keamanan Jaringan

Menurut Riza (2016:1), sistem keamanan jaringan adalah upaya untuk mencegah dan mengidentifikasi pengguna yang tidak sah atau penyusup dalam jaringan komputer. Tujuannya adalah untuk mengurangi risiko yang dapat berasal dari ancaman fisik (kerusakan perangkat keras komputer) maupun ancaman logis (pencurian data atau intrusi ke dalam akun seseorang). Ikhwan dan Elfitri (2014:119) menyatakan bahwa inti dari keamanan jaringan adalah mengendalikan akses terhadap sumber daya jaringan sehingga hanya orang yang berhak dapat mengaksesnya, sementara yang tidak berhak diblokir untuk menghindari potensi masalah keamanan.

Prinsip keamanan jaringan di klasifikasikan menjadi 3 bagian :

a. Confidentiality (Kerahasiaan)

Confidentiality adalah menjaga kerahasiaan objek atau data dari akses yang tidak sah. Ini mencakup melindungi data pribadi seperti nama, nomor kartu kredit, dan lainnya. Beberapa alat yang digunakan termasuk enkripsi, kontrol akses, otentikasi, otorisasi, dan keamanan fisik.

b. Integrity (Integritas)

Integrity berarti memastikan bahwa suatu objek atau data tetap asli dan tidak mengalami perubahan selama proses pengiriman atau komunikasi. Perubahan ini dapat terjadi melalui serangan seperti virus, trojan horse, atau campur tangan pihak ketiga. Untuk melindungi integritas data, digunakan alat seperti *checksums*, *data correcting codes*, dan *backup*.

c. **Availability (Ketersediaan)**

Availability berarti memastikan bahwa sumber daya atau layanan tersedia secara tepat waktu tanpa terganggu. Salah satu serangan yang dapat mengganggu ketersediaan adalah serangan *Distributed Denial of Service* (DDoS Attack) yang bertujuan untuk membanjiri sumber daya yang dibutuhkan oleh pengguna sehingga pengguna tidak dapat mengaksesnya dengan normal.

Pengiriman data yang berlebihan, seperti melalui serangan *Flooding Data*, dapat mengganggu jalur lalu lintas (*traffic*) dalam jaringan, menyebabkan lambatnya pengiriman dan masalah lainnya. Pada jam sibuk, lalu lintas data dalam jaringan juga dapat menjadi padat, yang mengakibatkan antrian data dan keterlambatan dalam pengiriman dan penerimaan data. (Hambali and Nurmiati 2018). Adapun macam-macam dari *flooding attack* yang sering dilakukan dalam jaringan adalah sebagai berikut:

- 1) *Ping of Death*
- 2) *Smurf Attack*
- 3) *SYN Flooding*
- 4) *UDP Flooding*

B. **Intrusion Detection System (IDS)**

Menurut penelitian Alamsyah et al.(2020:18), IDS (*Intrusion Detection System*) adalah sistem yang memantau lalu lintas jaringan dan kegiatan yang mencurigakan dalam jaringan. Jika IDS mendeteksi aktivitas mencurigakan terkait dengan lalu lintas jaringan, sistem atau administrator jaringan akan menerima peringatan. Sama halnya, menurut Muqorobin et al. (2019:2), IDS adalah aplikasi perangkat keras atau perangkat lunak yang secara otomatis memonitor kejadian dalam jaringan komputer dan menganalisis masalah keamanan. Tujuannya adalah mendeteksi perilaku yang tidak biasa, tindakan yang tidak semestinya, serta menghentikan serangan atau memberikan informasi yang berguna untuk melacak penyerang. Atmojo (2018:176) menambahkan bahwa IDS memberikan informasi tentang serangan yang terjadi sehingga administrator sistem dapat mengambil tindakan pencegahan. IDS dapat ditempatkan baik dalam perangkat keras maupun perangkat lunak, dan perlu pengawasan terus-menerus untuk efektif melindungi jaringan.

Terdapat dua jenis IDS berdasarkan penempatannya, yaitu:

a. **Host Based IDS (HIDS)**

HIDS hanya memantau peristiwa tertentu pada perangkat komputer, seperti kesalahan login dan file. Implementasi IDS perlu memperhatikan *false positive* (peringatan palsu) dan *false negative* (kegagalan mendeteksi serangan). IDS bisa melewatkan serangan jika tidak mengenalinya atau jika penyerang berhasil menghindari deteksi.

b. **Network Based IDS**

Network Based IDS akan melakukan pemantauan terhadap seluruh bagian pada jaringan dengan mengumpulkan paket-paket data yang terdapat pada jaringan tersebut serta melakukan analisa dan menentukan apakah paket paket tersebut merupakan paket normal atau paket serangan.

Menurut penelitian Muqorobin et al. (2019:2), *Network Intrusion Detection System* (NIDS) adalah jenis IDS yang ditempatkan di titik-titik strategis dalam jaringan untuk memantau dan menganalisis lalu lintas data. NIDS efektif untuk memonitor lalu lintas masuk dan keluar, serta lalu lintas di antara host atau segmen jaringan lokal. Biasanya, NIDS ditempatkan di depan dan di

belakang *firewall* dan VPN *gateway* untuk mengukur efektivitas perangkat keamanan tersebut dan berinteraksi dengan mereka untuk memperkuat keamanan jaringan. Namun, seperti yang disebutkan oleh Masse, dkk (2015:5), NIDS memiliki kelemahan dalam implementasinya pada jaringan yang menggunakan switch ethernet. Ini karena kompleksitas implementasi NIDS dalam jaringan dengan switch ethernet, meskipun beberapa produsen *switch ethernet* telah mulai mengintegrasikan fungsi IDS di dalam *switch* mereka untuk memantau port atau koneksi secara lebih efisien.

C. Monitoring

Monitoring adalah proses pengumpulan data untuk mengukur kemajuan program, sedangkan evaluasi adalah tempat data tersebut digunakan untuk belajar, membuat rekomendasi, dan perbaikan. Tanpa monitoring, evaluasi akan kurang dasar dan terbatas pada spekulasi, oleh itu keduanya harus berjalan bersama. Menurut Hikmat dalam Mardiani (2013:36), monitoring adalah pengumpulan dan analisis informasi berdasarkan indikator program secara sistematis dan terus-menerus, untuk memungkinkan tindakan koreksi yang diperlukan.

D. Android

Menurut Wiranda dan Adri (2020:85), Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan antara perangkat dan penggunaannya, sehingga pengguna dapat berinteraksi dengan *device*-nya dan menjalankan aplikasi yang tersedia pada *device*. Sedangkan menurut Tahel dan Ginting (2019:115), Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *Middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

E. Snort

Menurut Ekel dalam Masse, dkk (2015:6), Snort adalah *open source network intrusion detection system* (NIDS) yang memiliki kemampuan untuk memonitoring paket-paket sekaligus menjadi *security tools* yang berguna untuk mendeteksi berbagai serangan, sebagai contoh ddos, MITM *attack* dll. Snort dapat dioperasikan dengan tiga mode yaitu:

1. Paket *Sniffer*: untuk melihat paket yang lewat di jaringan.
2. Paket *Logger*: untuk mencatat semua paket yang lewat di jaringan untuk di analisis di kemudian hari.
3. NIDS: pada mode ini snort akan berfungsi untuk mendeteksi serangan yang dilakukan melalui jaringan komputer.

Komponen - komponen snort meliputi:

1. *Rule Snort*. Merupakan *database* yang berisi pola-pola serangan berupa *signature* jenis-jenis serangan. *Rule snort* IDS ini, harus di-*update* secara rutin agar ketika ada suatu teknik serangan yang baru.
2. *Snort Engine*. Merupakan program yang berjalan sebagian proses yang selalu bekerja untuk membaca paket data dan kemudian membandingkannya dengan *rule snort*.
3. *Alert*. Merupakan catatan serangan pada deteksi penyusupan. Untuk kebutuhan analisa, *alert* dapat disimpan dalam *database*, sebagai contoh ACID (*Analysis Console for Intrusion Database*) sebagai modul tambahan pada snort.

Dalam penerapannya, snort dapat ditambahkan beberapa *plugin* yang dapat membantu dalam menganalisis data yang berhasil di *capture*. *Plugin* yang biasa digunakan seperti *acid base*. *Acid base* digunakan sebagai tampilan agar hasil pembacaan snort bisa dibaca dengan mudah. Sebelum melakukan instalasi *Acid Base*, terlebih dahulu di-*install* *baryard2* agar hasil pembacaan snort bisa disimpan dalam *mysql database*. *Base* menunjukkan beberapa informasi tentang *network intrusion detection system* yang terpasang pada komputer *server*. Informasi itu

adalah jumlah alert yang dibangkitkan snort, jumlah sensor, jenis *alert*, *source* IP dan *destination* IP.

F. Bot Telegram

Menurut Utomo, dkk (2017:82), Telegram adalah sebuah sistem perpesanan yang lintas platform dan berpusat pada keamanan kerahasiaan pribadi penggunanya, sedangkan bot adalah program komputer yang melakukan pekerjaan tertentu secara otomatis. Bot adalah sebuah mesin, dibuat memudahkan kehidupan keseharian kita tanpa harus terpaku di depan komputer. Jika ingin membuat bot telegram, ia perlu komunikasi utama dengan peladen (*server*) telegram dilakukan melalui protokol MTProto, sebuah protokol biner buatan telegram sendiri. Bot yang paling terkenal adalah telegram-bot buatan Yugo Perez. Bot-telegram cli bekerja layaknya akun pribadi dan manfaat bot ini diamini juga oleh telegram yang kemudian meluncurkan bot API (*Application Programming Interface*) agar orang banyak dapat membangun bot menggunakan bahasa pemrograman yang mereka kuasai tanpa harus berhubungan dengan telegram-cli atau MTProto.

3. Metode penelitian

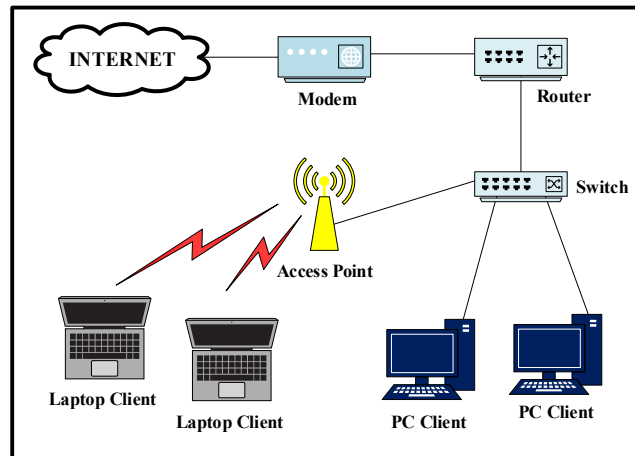
Metode penelitian yang digunakan adalah metode Eksperimen. Pada penelitian ini dilakukan Implementasi *Network Intrusion Detection System* (NIDS) serta sistem notifikasi menggunakan aplikasi telegram pada smartphone android. Hasil eksperimen selanjutnya didokumentasikan untuk melakukan analisis sehingga dihasilkan rekomendasi yang tepat untuk NIDS dengan notifikasi menggunakan aplikasi telegram pada Android. Dari hasil analisis tersebut nantinya akan mendapatkan kesimpulan mengenai manfaat, fungsi serta kelebihan dari sistem yang sudah dibangun.

A. Perangkat Lunak dan Perangkat Keras

- a. Perangkat Lunak (Software)
 - 1) Sistem Operasi Linux Ubuntu Server
 - 2) Snort
 - 3) Acidbase
 - 4) Telegram
- b. Perangkat Keras (Hardware)
 - 1) 1 unit PC sebagai NIDS Server
 - 2) 1 unit RouterBoard RB 750
 - 3) 1 Unit Laptop sebagai Attacker (Penyerang)
 - 4) 1 unit Smartphone Android

B. Diagram Blok Sistem Lama

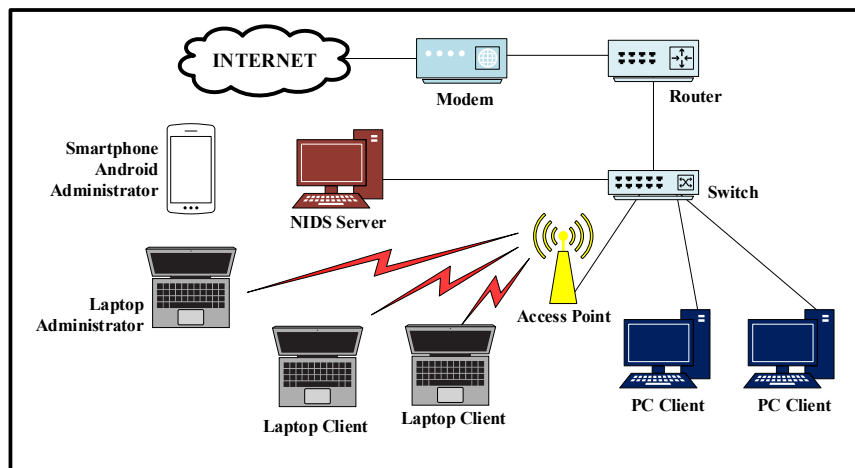
Berdasarkan dari data yang penulis peroleh dari studi observasi yang dilakukan pada tempat penelitian, yaitu laboratorium komputer UPT. Puskom, saat ini tidak ada sistem yang digunakan secara khusus untuk melakukan pemantauan gangguan di dalam jaringan komputer. Adapun skema diagram blok sistem yang ada saat ini adalah sebagai berikut.



Gambar 1. Diagram Blok Sistem Lama

C. Diagram Blok Sistem Baru

Pada penelitian ini dilakukan pengembangan terhadap jaringan yang sudah ada dengan menerapkan *Network Intrusion Detection System* (NIDS) serta notifikasi menggunakan aplikasi Telegram pada Android. Adapun topologi yang akan digunakan adalah sebagai berikut.



Gambar 2. Diagram Blok Sistem Baru

Pada Gambar 1 dapat dilihat bahwa terdapat penambahan server yang akan digunakan sebagai *Network Intrusion Detection System* (NIDS) server yang akan memantau ataupun melakukan deteksi di dalam satu segmen jaringan, selain itu juga akan diterapkan sistem notifikasi yang terhubung ke perangkat android dengan menggunakan bot telegram.

Prinsip kerja dari *Network Intrusion Detection System* (NIDS) pada jaringan LAN adalah dengan melakukan pemantauan terhadap *Traffic* pada semua bagaian jaringan, baik *Traffic* yang normal ataupun *Traffic* yang tidak normal, dimana pengetahuan tentang *Traffic* itu nantinya akan dibuat dalam sebuah *rule* ataupun aturan yang diterapkan pada NIDS server, selain itu NIDS server ini nantinya juga akan mengirimkan notifikasi ke *smartphone* Android melalui aplikasi bot telegram.

D. Rencana Pengujian

Pengujian ini dilakukan dengan metode *blackbox*, yaitu sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional sistem saat dioperasikan, apakah *input* diterima dengan benar dan *output* yang dihasilkan telah sesuai dengan yang diharapkan, sehingga dapat membuktikan kebenarannya. Adapun rancangan pengujian dapat dilihat seperti Tabel 1.

Tabel 1. Rencana Jenis dan Kriteria Pengujian

No	Jenis Pengujian	Kriteria
1.	Pengujian ICMP <i>Flooding</i> (<i>Ping Of Death</i>)	Pemantauan dengan NIDS <i>Server</i> Notifikasi Melalui Android dengan Bot Telegram
2.	Pengujian UDP <i>Flooding</i>	Pemantauan dengan NIDS <i>Server</i> Notifikasi Melalui Android dengan Bot Telegram
3.	Pengujian Waktu yang dibutuhkan dalam deteksi	ICMP <i>Flooding</i> UDP <i>Flooding</i>
4.	Pengujian Penggunaan Sumberdaya	Melihat seberapa besar penggunaan <i>disk space</i> , memori dan cpu dari setiap deteksi yang dilakukan oleh NIDS <i>Server</i>

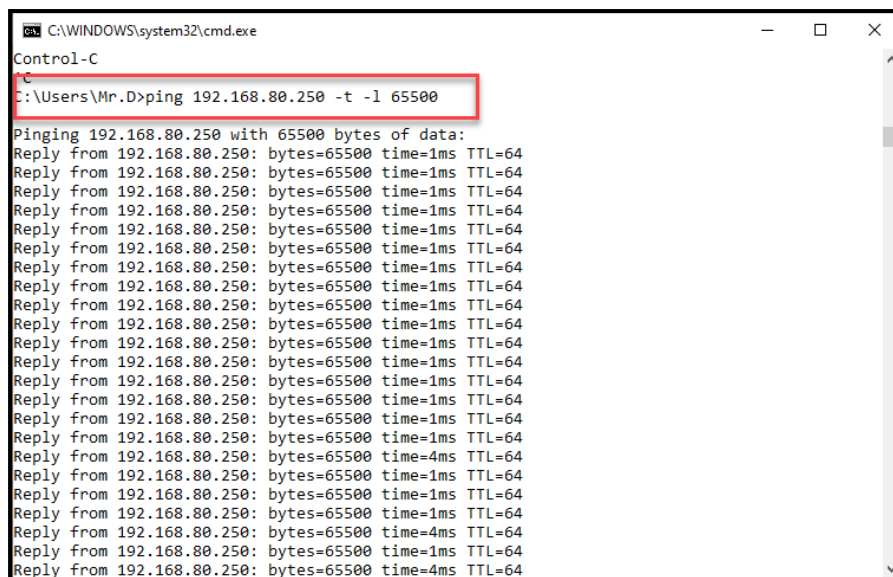
4. Hasil dan pembahasan

A. Pengujian ICMP *Flooding*

Pengujian ICMP flooding atau yang sering disebut juga dengan *Ping of Death* ini dapat dilakukan dengan cara melakukan perintah ping melalui command prompt dengan menambahkan byte tinggi untuk membanjiri jalur lalu lintas pada objek penyerangan, yang mana dalam penelitian ini objek penyerangan adalah IP address 192.168.80.250. Adapun percobaan yang penulis lakukan antara lain sebagai berikut.

a. Melakukan Percobaan PING dengan CMD

Untuk membanjiri jalur lalu lintas data pada objek penyerangan dapat dilakukan dengan menggunakan PING address “ping 192.168.80.250 -t -l 65500” dimana besaran byte yang digunakan adalah 65.500 byte seperti yang terlihat pada Gambar 3.



```

C:\WINDOWS\system32\cmd.exe
Control-C
C
C:\Users\Mr.D>ping 192.168.80.250 -t -l 65500

Pinging 192.168.80.250 with 65500 bytes of data:
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=4ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=4ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=4ms TTL=64

```

Gambar 3. Ping dengan Byte 65500

b. Monitoring Pada NIDS Server

Setelah pengujian dijalankan pada laptop penyerang dengan besaran byte 65500, server NIDS dapat memantau serangan tersebut dengan menjalankan script bot telegram pada server NIDS dengan mengetikkan perintah “./bot-tele.sh”, hasilnya dapat dilihat pada Gambar 4

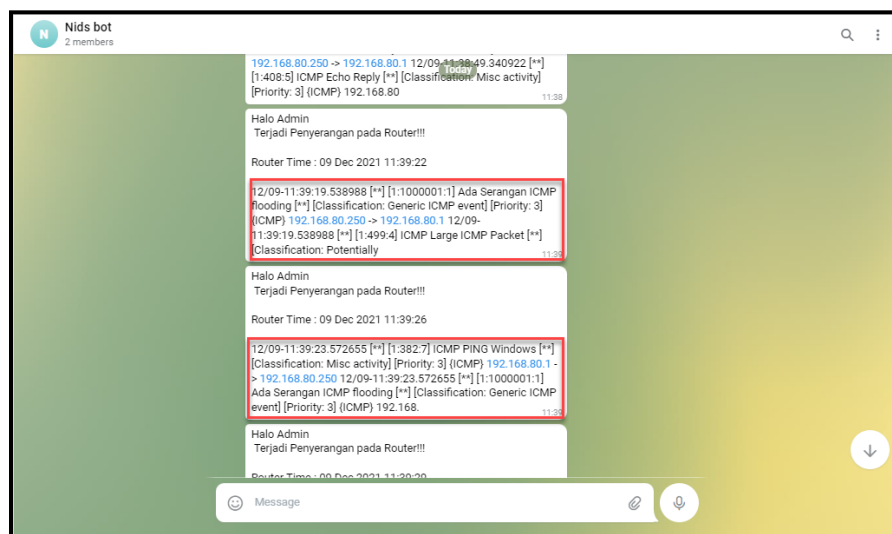


Gambar 4. Monitoring ICMP Flooding pada NIDS Server

Pada Gambar 4 dapat dilihat NIDS server dapat memantau ataupun memonitoring serangan ICMP flooding yang dilakukan oleh address 192.168.80.1.

c. Notifikasi Melalui Android dengan Bot Telegram

Selain monitoring yang dilakukan pada NIDS server, notifikasi serangan juga akan terkirim ke telegram bot yang sudah dikonfigurasi sebelumnya, seperti yang terlihat pada Gambar 4.



Gambar 4. Notifikasi Serangan pada Bot Telegram

Pada Gambar 4.3 diatas, dapat dilihat bahwa adanya serangan yang terjadi pada address 192.168.80.250 dengan pesan “Ada Serangan ICMP Flooding”.

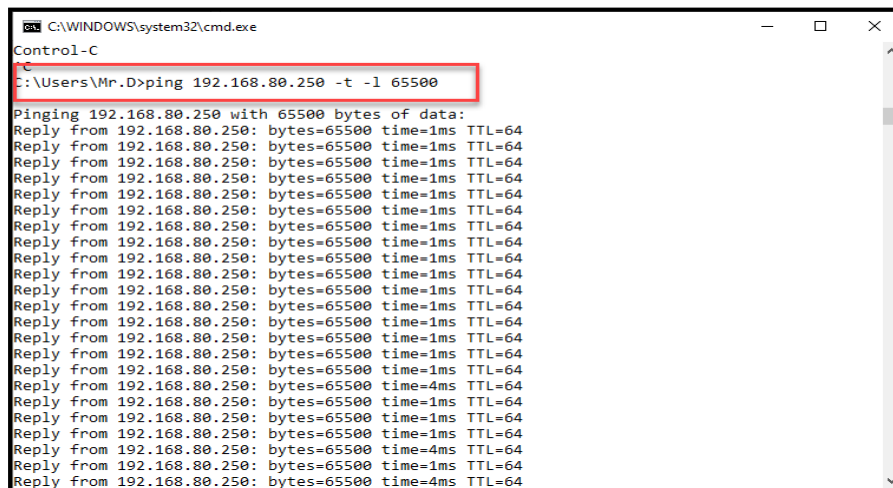
B. Pengujian UDP Flooding

Serangan UDP ini memanfaatkan protokol UDP yang bersifat *connectionless* untuk menyerang target. Karena sifatnya itulah *UDP flood* cukup mudah untuk dilakukan. Sejumlah paket data yang besar dikirimkan begitu saja kepada target. Target yang tidak siap menerima serangan ini tentu akan bingung, dan pada beberapa kasus komputer *server* tersebut akan hang karena besarnya paket data yang dikirimkan. Penyerang dapat menggunakan tehnik spoofed untu menyembunyikan identitasnya. Pengujian *UDP flooding* dapat dilakukan dengan cara melakukan perintah ping melalui *command promt* dengan menambahkan byte tinggi untuk

membanjiri jalur lalu lintas pada objek penyerangan sama halnya dengan ICMP flood, hanya saja UDP flooding ini terjadi pada waktu tertentu saja, yang mana dalam penelitian ini objek penyerangan adalah IP address 192.168.80.250. Adapun percobaan yang penulis lakukan antara lain sebagai berikut.

a. Melakukan Percobaan PING dengan CMD

Untuk mengirimkan paket data yang besar pada objek penyerangan dapat dilakukan dengan menggunakan PING address “ping 192.168.80.250 -t -l 65500” dimana besaran byte yang digunakan adalah 65.500 byte seperti yang terlihat pada Gambar 5.



```

C:\WINDOWS\system32\cmd.exe
Control-C
C:\Users\Mr.D>ping 192.168.80.250 -t -l 65500

Pinging 192.168.80.250 with 65500 bytes of data:
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=4ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=4ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=1ms TTL=64
Reply from 192.168.80.250: bytes=65500 time=4ms TTL=64

```

Gambar 5. Ping dengan paket data 65500 b

b. Monitoring Pada NIDS Server

Setelah pengujian dijalankan pada laptop penyerang dengan besaran byte 65500, server NIDS akan menerima paket data secara mendadak dengan ukuran yang besar, sehingga hasil dari pemantauan serangan tersebut berupa UDP flooding, untuk melihat hasil monitoring dapat dilakukan dengan menjalankan script bot telegram pada server NIDS dengan mengetikkan perintah “./bot-tele.sh”, sehingga hasilnya dapat dilihat pada Gambar 6.



```

root@nids_server: /home/Snort-Bot-Telegram-Shell
root@nids_server: /home/Snort-Bot-Telegram-Shell# ./bot-tele.sh
{"ok":true,"result":{"message_id":42,"from":{"id":2144804111,"is_bot":true,"first_name":"meisi_ids","username":"meisinids_bot"},"chat":{"id":-723401403,"title":"Nids bot","type":"group"},"all_members_are_administrators":true},"date":1638948970,"text":"Halo Admin\n Terjadi Penyerangan pa a Router!!!\n\nRouter Time : 08 Dec 2021 14:36:08\n\n12/08-14:30:34.533998 [**] [1:1000003:1] A a Serangan UDP flooding [**] [Priority: 0] {UDP} 8.8.8.8:53 -> 192.168.80.250:53663 12/08-14:31 40.001392 [**] [1:1000003:1] Ada Serangan UDP flooding [**] [Priority: 0] {UDP} 192.168.80.1:13 -> 192.168.80.255:138","entities":[{"offset":176,"length":10,"type":"url"},{"offset":190,"length":20,"type":"url"},{"offset":303,"length":16,"type":"url"},{"offset":323,"length":18,"type":"url"}]}Alert Terkirim

^C
root@nids_server: /home/Snort-Bot-Telegram-Shell# clear
root@nids_server: /home/Snort-Bot-Telegram-Shell# ./bot-tele.sh

{"ok":true,"result":{"message_id":43,"from":{"id":2144804111,"is_bot":true,"first_name":"meisi_ids","username":"meisinids_bot"},"chat":{"id":-723401403,"title":"Nids bot","type":"group"},"all_members_are_administrators":true},"date":1638949374,"text":"Halo Admin\n Terjadi Penyerangan pa a Router!!!\n\nRouter Time : 08 Dec 2021 14:42:50\n\n12/08-14:42:48.790431 [**] [1:1000001:1] A a Serangan ICMP Flooding [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168 80.250 -> 192.168.80.1 12/08-14:42:48.790431 [**] [1:499:4] ICMP Large ICMP Packet [**] [Classi ication: Potentially Bad Traffic] [Prioriti","entities":[{"offset":215,"length":14,"type":"url"}, {"offset":233,"length":12,"type":"url"}]}Alert Terkirim

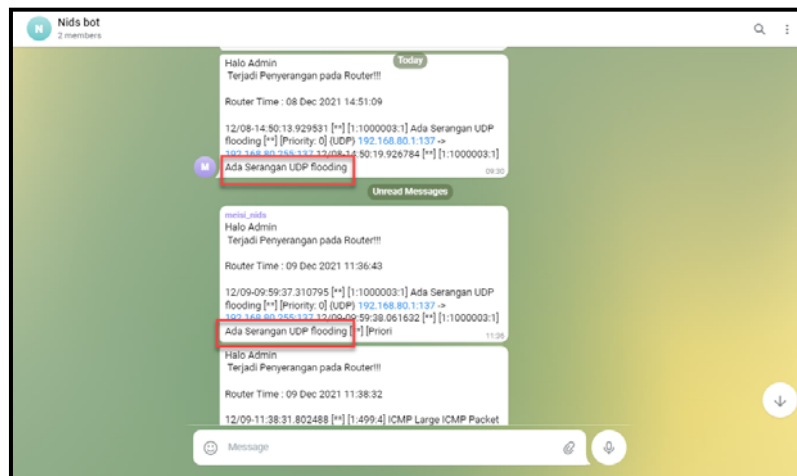
```

Gambar 6. Monitoring UDP Flooding pada NIDS Server

Pada Gambar 6 dapat dilihat NIDS server dapat memantau ataupun memonitoring serangan UDP flooding yang dilakukan oleh address 192.168.80.1.

c. Notifikasi Melalui Android dengan Bot Telegram

Selain monitoring yang dilakukan pada NIDS server, notifikasi serangan juga akan terkirim ke telegram bot yang sudah dikonfigurasi sebelumnya, seperti yang terlihat pada Gambar 7.



Gambar 7. Notifikasi Serangan pada Bot Telegram

Pada Gambar 7 dapat dilihat bahwa adanya serangan yang terjadi pada address 192.168.80.250 dengan pesan “Ada Serangan UDP Flooding”.

C. Pengujian Waktu yang dibutuhkan dalam deteksi

Pengujian ini dilakukan dengan memperhatikan waktu dari monitoring oleh NIDS dan notifikasi yang dikirimkan ke bot telegram. Dari pengujian yang sudah dilakukan, didapatkan perbedaan waktu yang cukup jauh antara deteksi oleh NIDS *server* saat adanya serangan UDP *flooding* yaitu 5 menit sedangkan untuk ICMP flooding tidak ada selisih waktu seperti yang terlihat seperti Gambar 8.



Gambar 8. Perbandingan Waktu Deteksi Serangan

Sedangkan untuk notifikasi telegram bot tidak ada perbandingan waktu antara deteksi oleh NIDS *server* dan notifikasi terkirim ke group telegram bot.

D. Pengujian Penggunaan Sumberdaya

Pengujian ini dilakukan dengan menggunakan *tool default* dari linux ubuntu yaitu top dengan cara mengetikkan perintah “top” pada terminal linux, dimana fungsi dari tool top ini adalah untuk menampilkan informasi yang berhubungan dengan performansi dari komputer seperti yang terlihat pada Gambar 9.

```

root@nids_server: /home/Snort-Bot-Telegram-Shell
top - 13:20:49 up 6:19, 2 users, load average: 0.14, 0.04, 0.01
Tasks: 99 total, 2 running, 52 sleeping, 3 stopped, 0 zombie
%Cpu(s): 43.0 us, 56.6 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
KiB Mem : 1008732 total, 246412 free, 320348 used, 441972 buff/cache
KiB Swap: 1902588 total, 1902320 free, 268 used, 536468 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 760 root        20   0 316580 63212 4744 R  98.7   6.3   0:04.73 snort
20602 root        20   0   0      0   0   I  0.7   0.0   0:10.60 kworker/0:1
 8 root        20   0   0      0   0   I  0.3   0.0   0:04.55 rcu_sched
434 root        20   0 42768 3884 3300 R  0.3   0.4   0:00.49 top
 1 root        20   0 159792 8980 6608 S  0.0   0.9   0:02.98 systemd
 2 root        20   0   0      0   0   S  0.0   0.0   0:00.00 kthreadd
 4 root        0 -20   0      0   0   I  0.0   0.0   0:00.00 kworker/0:0H
 6 root        0 -20   0      0   0   I  0.0   0.0   0:00.00 mm_percpu_wq
 7 root        20   0   0      0   0   S  0.0   0.0   0:01.16 ksoftirqd/0
 9 root        20   0   0      0   0   I  0.0   0.0   0:00.00 rcu_bh
10 root        rt   0   0      0   0   S  0.0   0.0   0:00.00 migration/0
11 root        rt   0   0      0   0   S  0.0   0.0   0:00.09 watchdog/0
12 root        20   0   0      0   0   S  0.0   0.0   0:00.00 cpuhp/0
13 root        20   0   0      0   0   S  0.0   0.0   0:00.00 kdevtmpfs
14 root        0 -20   0      0   0   I  0.0   0.0   0:00.00 netns
15 root        20   0   0      0   0   S  0.0   0.0   0:00.00 rcu_tasks_kthre
16 root        20   0   0      0   0   S  0.0   0.0   0:00.00 kauditd

```

Gambar 4.8 Penggunaan Sumber Daya Server NIDS

Dari Gambar 9 dapat dilihat penggunaan cpu dan memory oleh snort saat mendeteksi adanya serangan, dimana untuk penggunaan memory sebesar 6.3% dan penggunaan CPU sebesar 98.7% yang diakses oleh user root.

4.1 Hasil Pengujian

Tabel 2. Hasil Pengujian

No	Jenis Pengujian	Kriteria	Hasil	Keterangan
1.	Pengujian ICMP Flooding (Ping Of Death)	Pemantauan dengan NIDS Server	NIDS server dapat memonitoring serangan ICMP flooding	ICMP Flooding terdeteksi
		Notifikasi Melalui Android dengan Bot Telegram	Notifikasi ICMP Flooding terkirim ke bot telegram dengan waktu yang sama saat deteksi oleh NIDS server	
2.	Pengujian UDP Flooding	Pemantauan dengan NIDS Server	NIDS server dapat memonitoring serangan UDP flooding	UDP Flooding terdeteksi
		Notifikasi Melalui Android dengan Bot Telegram	Notifikasi UDP Flooding terkirim ke bot telegram dengan waktu yang sama saat deteksi oleh NIDS server	
3.	Pengujian Waktu yang dibutuhkan dalam deteksi	ICMP Flooding	Tidak ada selisih waktu monitoring ICMP Flooding	Terjadi selisih waktu deteksi antara serangan UDP dan deteksi oleh NIDS server
		UDP Flooding	Terdapat selisih Waktu 5 menit untuk Deteksi Serangan UDP Flooding	

4.	Pengujian Penggunaan Sumberdaya	Melihat seberapa besar penggunaan memori dan cpu dari setiap deteksi yang dilakukan oleh NIDS Server	Saat serangan sedang berlangsung, terdapat penambahan yang cukup besar pada penggunaan CPU yaitu sebesar 98.7% sedangkan untuk memori hanya sebesar 6.3 %	Cukup Baik
----	---------------------------------	------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------	------------

5. Kesimpulan

Dari hasil penelitian yang dilakukan, dapat disimpulkan bahwa :

- Network Intrusion Detection System* (NIDS) dapat mendeteksi serangan ICMP *flooding* dan UDP *flooding* pada jaringan Intranet.
- Sistem notifikasi menggunakan telegram bot dapat menerima notifikasi serangan ICMP dan UDP *flooding* menggunakan *script* berbasis *bash shell*.
- Terdapat selisih waktu sebesar 5 menit untuk deteksi serangan UDP *flooding* oleh NIDS server, sedangkan untuk serangan ICMP *flooding* tidak ada selisih waktu.
- NIDS *server* mampu menjalankan sistem pendeteksi dan juga notifikasi secara bersamaan dengan performa *server* cukup baik, dimana penggunaan CPU sebesar 98.7% dan *Memory* 6.3 % saat terjadi serangan.

Limitasi dan studi lanjutan

- Sistem yang dibuat hanya mampu mendeteksi ICMP dan UDP *flooding*, untuk memastikan performa server yang lebih baik lagi, coba dengan jenis serangan lainnya atau menggunakan aplikasi yang lebih baik lagi.
- Untuk monitoring yg lebih baik, coba tambahkan aplikasi untuk menampilkan log secara realtime.
- Masih terdapat selisih waktu dalam deteksi serangan UDP *flooding*, analisis rules yang sudah digunakan agar tidak ada selisih waktu.

REFERENSI

- Alamsyah, Hendri, Riska, and Al Akbar, Abdussalam. 2020. "Analisa Keamanan Jaringan Menggunakan Network Intrusion Detection and Prevention System." *JOINTECS (Journal of Information Technology and Computer Science)* 5(1): 17.
- Atmojo, Yohanes Priyo. 2018. "Bot Alert Snort Dengan Telegram Bot API Pada Intrusion Detection System: Studi Kasus IDS Pada Server Web." *Proceeding Seminar Nasional Sistem Informasi dan Teknologi Informasi* 12(1): 176–80.
[https://api.telegram.org/bot\\$apiToken/sendMessage](https://api.telegram.org/bot$apiToken/sendMessage).
- Hambali, Achmad, and Siti Nurmiati. 2018. "Implementasi Intrusion Detection System (IDS) Pada Keamanan PC Server Terhadap Serangan Flooding Data." *Sainstech: Jurnal Penelitian dan Pengkajian Sains dan Teknologi* 28(1): 35–43.
- Ikwan Syariful dan Elfitri E, 2014. *Analisa Delay Yang Terjadi Pada Penerapan Demilitarized Zone (Dmz) Terhadap Server Universitas Andalas*. Jurnal Nasional Teknik Elektro Vol: 3 No. 2. Universitas Andalas Padang.
- Masse, Fitriyanti A, Andi Nurul Hidayat, and Badrianto. 2015. "Penerapan Network Intrusion Detection System Menggunakan Snort Berbasis Database MySQL Pada Hotspot Kota." *Jurnal Elektronik Sistem Informasi Dan Komputer* 1(2): 1–16.
- Muqorobin, Muqorobin et al. 2019. "Implementasi Network Intrusion Detection System (NIDS) Dalam Sistem Keamanan Open Cloud Computing." *Majalah Ilmiah Bahari Jogja* 17(2): 1–9.
- Riza Muhammad, 2016. *Sistem Keamanan Jaringan Komputer, Artikel Microcyber2*.
<https://webdev-id.com/berita/sistem-keamanan-jaringan/>. diakses tgl 27 Agustus 2021.

- Tahel, Fithry, and Erwin Ginting. 2019. "Perancangan Aplikasi Media Pembelajaran Pengenalan Pahlawan Nasional Untuk Meningkatkan Rasa Nasionalis Berbasis Android." *Teknomatika* 09(02): 113–20. <http://ojs.palcomtech.com/index.php/teknomatika/article/view/467>.
- Utomo, Dias, Muchammad Sholeh, and Arry Avorizano. 2017. "Membangun Sistem Mobile Monitoring Keamanan Web Aplikasi Menggunakan Suricata Dan Bot Telegram Channel." *Seminar Nasional Teknoka 2(2502)*: 1–7.
- Wiranda, Tio, and Muhammad Adri. 2020. "Rancang Bangun Aplikasi Modul Pembelajaran Teknologi Wan Berbasis Android." *Voteteknika (Vocational Teknik Elektronika dan Informatika)* 7(4): 85.