

Wireless Intrusion Detection System Pada STMIK Bina Insani

Desi Puspasari¹, Hero Suhartono², Herlawati^{2*}

¹ Teknik Informatika; STMIK Bina Insani; Jl. Siliwangi No.6 Rawa Panjang Bekasi Timur 17114 Indonesia, Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: desypuspasari30@gmail.com, hero.debian@gmail.com

² Sistem Informasi; STMIK Bina Insani; Jl. Siliwangi No.6 Rawa Panjang Bekasi Timur 17114 Indonesia, Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: herlawati@binainsani.ac.id

* Korespondensi: e-mail: herlawati@binainsani.ac.id

Diterima: 08 April 2018; Review: 16 April 2018; Disetujui: 23 April 2018

Cara sitasi: Puspasari D, Suhartono H, Herlawati. 2018. *Wireless Intrusion Detection System Pada STMIK Bina Insani*. Information Management For Educators And Professionals. 2 (2): 199 – 208.

Abstrak: Jaringan *wireless* yang berada pada *frekuensi* terbuka seringkali dimanfaatkan oleh *attacker* sebagai jalur masuk untuk menyusup ke jaringan infrastruktur utama (*production network*) pada suatu instansi. Oleh karena itu keamanan jaringan *wireless* menjadi sangat penting agar tidak berakibat terganggunya suatu kegiatan tertentu. Maka diperlukan suatu sistem keamanan yang mampu mendeteksi serangan-serangan yang tertuju pada jaringan *wireless*. Implementasi *Wireless Intrusion Detection System* menggunakan metode NDLC menjadi salah satu solusi untuk keamanan jaringan *wireless*. Hasil dari implementasi *Wireless Intrusion Detection System* setiap serangan yang terjadi pada jaringan *wireless* dapat terdeteksi oleh sistem IDS dan dapat memberikan pesan peringatan kepada *administrator*.

Kata kunci: keamanan, *wireless intrusion detection system*, *wids.py*

Abstract: *Wireless networks that are on an open frequency are often used by attackers as the entry point to infiltrate the main network infrastructure (production network) in an agency. Therefore, wireless network security becomes very important so as not to disrupt a particular activity. Then required a security system that is able to detect attacks directed at wireless networks. Implementation of Wireless Intrusion Detection System to be one solution for wireless network security. The results of the Wireless Intrusion Detection System implementation of any attacks that occur on wireless networks can be detected by the IDS system and can provide a warning message to the administrator.*

Keywords: *security, wireless intrusion detection system, wids.py*

1. Pendahuluan

Perkembangan teknologi jaringan komputer membuat semua infrastruktur jaringan beralih ke jaringan *wireless* hal itu disebabkan karena sifat jaringan *wireless* yang fleksibel serta lebih mudah dalam perancangan dan penggunaannya. Pengguna jaringan *wireless* dapat bergerak bebas selama masih terhubung melalui gelombang radio yang ditangkap oleh *wireless adaptor* yang ada pada komputer, *notebook*, *smartphone*. Penggunaan teknologi *wireless* yang diimplementasikan dalam suatu jaringan lokal sering dinamakan WLAN (*Wireless Local Area Network*).

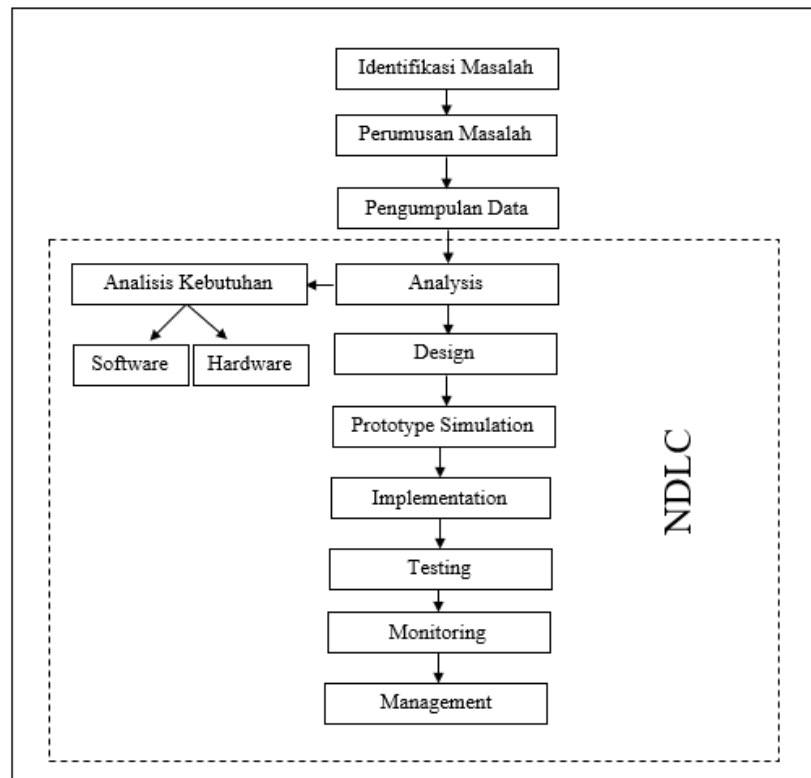
Wireless, pada dasarnya terdiri dari dua kata, yaitu *wire* yang artinya kawat atau kabel dan *less* yang bermakna tiada, tidak ada, tanpa. Jadi, jika diartikan *wireless* berarti tanpa kabel, atau tidak menggunakan kabel. Secara lengkap jaringan *wireless* merupakan sebuah teknologi komunikasi yang tidak menggunakan kabel untuk menghubungkan antar perangkat, melainkan dengan memanfaatkan gelombang radio sebagai media yang digunakan [Zam, 2014].

Sistem deteksi intrusi dapat didefinisikan sebagai alat atau metode yang digunakan untuk memantau semua *inbound* dan *outbound host* atau aktivitas jaringan dengan mengidentifikasi pola mencurigakan. Pola yang mencurigakan dapat mengindikasikan adanya serangan atau kesalahan komputer. Sistem deteksi intrusi dapat mengidentifikasi dan memblokir IP yang terkait dengan pola yang mencurigakan [Council, 2011].

Wireless IDS (WIDS) secara khusus dibuat untuk memantau jaringan *wireless*. WIDS menganalisis aktivitas pengguna dan sistem, mendeteksi aktivitas jaringan tidak normal, dan mendeteksi pelanggaran kebijakan untuk WLAN. WIDS menonton semua transmisi *wireless* lokal untuk mengetahui tanda tangan konten berbahaya yang diketahui [Council, 2011].

2. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode *Network Development Life Cycle (NDLC)*, metode ini terbagi menjadi enam tahapan. Gambaran mengenai tahapan dari metode *Network Development Life Cycle* dipaparkan pada Gambar 1.



Sumber: Hasil Penelitian (2018)

Gambar 1. Metode *Network Development Life Cycle*

Berdasarkan gambar diatas, dapat dijelaskan bahwa metode tersebut memiliki enam tahapan, yaitu: **Pertama**, Tahap *Analysis*, pada tahap ini yang dilakukan adalah analisa kebutuhan *software* dan *hardware* untuk memastikan setiap perangkat yang digunakan memberikan hasil yang diinginkan dalam implementasi *wireless intrusion detection system*. **Kedua**, Tahap *Design*, setelah mendapatkan data-data dari tahap analisa, maka tahap selanjutnya adalah membuat gambar desain topologi jaringan yang akan dibangun. **Ketiga**, Tahap *Simulation Prototype*, pada tahap ini membangun *wireless Intrusion Detection System* menggunakan sistem operasi Kali Linux dan menggunakan *script open source wireless IDS script python (wids.py)* sebagai simulator untuk mencegah terjadinya kesalahan yang mungkin terjadi pada proses implementasi. **Keempat**, Tahap *Implemetation*, pada tahap ini akan dilakukan penerapan dari hasil analisa kebutuhan, desain topologi, dan simulasi prototipe jaringan yang akan dilakukan sesuai dengan bentuk yang sudah terimplementasi pada tahapan simulasi. **Kelima**, Tahap *Monitoring*, pada tahap monitoring ini dilakukan pada *wireless Intrusion Detection System* yang telah diterapkan dengan memanfaatkan *script open wireless*

IDS *script python* (*wids.py*) sehingga apabila terjadi penyerangan pada jaringan *wireless* akan memunculkan pesan peringatan dan semua jenis serangan akan tercatat pada file log. **Keenam**, Tahap *Management*, tahap *management* atau pengaturan dengan melakukan pemeliharaan dan pengelolaan yang baik secara berkala sehingga sistem yang telah dibangun dapat berlangsung lama.

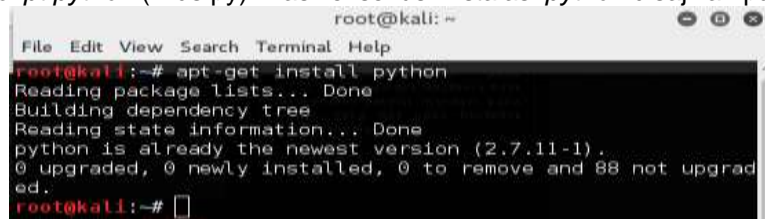
3. Hasil dan Pembahasan

Dalam implementasi *wireless intrusion detection system* yang dilakukan, penelitian ini menggunakan tool yang ada pada sistem operasi Kali Linux dengan menggunakan *open source Wireless IDS Script Python* (*wids.py*). Berikut beberapa tahapan yang dilakukan:

3.1. Konfigurasi *Wireless Intrusion Detection System*

a. Instalasi *Python*

Pada tahap ini dilakukan instalasi *python* dimana terlihat sudah terinstal dengan versi terbaru yaitu *python 2.7.11-1*. *Python* ini digunakan untuk membaca bahasa pemrograman *wireless IDS script python* (*wids.py*). Hasil eksekusi instalasi *python* disajikan pada gambar 2.



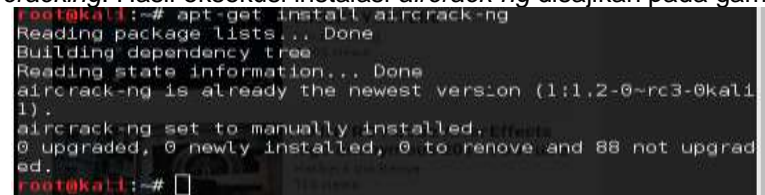
```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# apt-get install python  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python is already the newest version (2.7.11-1).  
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.  
root@kali:~#
```

Sumber: Hasil Penelitian (2018)

Gambar 2. Instalasi *Python*

b. Instalasi *Aircrack-ng*

Selanjutnya instalasi *Aircrack-ng* yang biasanya sudah terinstal manual pada Kali Linux. *Aircrack-ng* berfokus pada keamanan jaringan *wireless* seperti pemantauan, penyerangan, pengujian dan *cracking*. Hasil eksekusi instalasi *aircrack-ng* disajikan pada gambar 3.



```
root@kali:~# apt-get install aircrack-ng  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
aircrack-ng is already the newest version (1:1.2-0~rc3-0kali1).  
aircrack-ng set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.  
root@kali:~#
```

Sumber: Hasil Penelitian (2018)

Gambar 3. Instalasi *Aircrack-ng*

c. Instalasi *Tshark*

Selanjutnya melakukan instalasi *Tshark*, fungsi dari *Tshark* untuk melakukan *capture* paket data yang masuk maupun keluar pada jaringan secara *real time* dan *filter* paket data. Hasil eksekusi instalasi *Tshark* adalah sebagai berikut:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# apt-get install tshark  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
tshark is already the newest version (2.0.1+g59ea390-3).  
6 upgraded, 6 newly installed, 6 to remove and 88 not upgraded.  
root@kali:~#
```

Sumber: Hasil Penelitian (2018)

Gambar 4. Instalasi *Tshark*

d. Konfigurasi *wids.py*

Wireless IDS Script Python (*wids.py*) merupakan salah satu aplikasi IDS yang bersifat *open source* dan *wids.py* dapat di download pada berbagai sumber di internet. Pada penelitian ini *wids.py* di download pada <https://github.com/SYWorks/wireless-ids> dan simpan file *wids.py* pada */home/lirva32/test_ids*. Sebelum menjalankan *wireless IDS script python* langkah pertama adalah memastikan *device wireless* yang akan digunakan dengan perintah pada gambar 5.

```
root@kali:~# iwconfig
```

Sumber: Hasil Penelitian (2018)

Gambar 5. Perintah Memastikan *Wireless Device*

Setelah melakukan perintah pada gambar 5 maka di dapat hasil pada gambar 6:

```
root@kali:~# iwconfig
wlan0 IEEE 802.11bgn ESSID:"WifiGratisan"
Mode:Managed Frequency:2.437 GHz Access Point: 30:CB:68:ED:41:78
Bit Rate=72.2 Mb/s Tx-Power=20 dBm
Retry short limit:7 RTS thr=2347 B Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=78/79 Signal level=-14 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:1 Missed beacon:0

lo no wireless extensions.

eth0 no wireless extensions.

wlan1 IEEE 802.11bgn ESSID:off/any
Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
```

Sumber: Hasil Penelitian (2018)

Gambar 6. Tampilan *Wireless Device* Yang Digunakan

Setelah ditemukan *device wireless* ditemukan maka langkah selanjutnya adalah melakukan pengaturan *Wifi USB* menjadi *monitor mode* dengan perintah pada Gambar 7.

```
root@kali:~# airmon-ng start wlan0
```

Sumber: Hasil Penelitian (2018)

Gambar 7. Perintah Mengubah *Wireless Device* Menjadi *Monitor Mode* *Web Server*

```
root@DebianGutsy:~# airmon-ng start wlan0

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID      Name
2645    NetworkManager
2768    wpa_supplicant
2808    dhclient
Process with PID 2808 (dhclient) is running on interface wlan0

Interface      Chipset          Driver
wlan0          Intel N iwlwifi - [phy0]
                (monitor mode enabled on mon0)
```

Sumber: Hasil Penelitian (2018)

Gambar 8. Hasil *Wireless Device* Menjadi *Monitor Mode*

Pada gambar 8 dapat dilihat *wireless device* sudah berubah menjadi *monitor mode* dengan *wireless interface mono*. Setelah itu untuk menjalankan *wireless IDS script python* (*wids.py*) gunakan perintah pada Gambar 9:

```
root@DebianGutsy:~/home/lirva32/test_ids# python wids.py -i mon0
```

Sumber: Hasil Penelitian (2018)

Gambar 9. Perintah Menjalankan *wireless IDS script python* (*wids.py*)

Berikut ini tampilan awal *wireless IDS script python* (*wids.py*) setelah menjalankan perintah pada gambar 8:

```
root@DebianGutsy:~/home/lirva32/test_ids# python wids.py -i non0
      SSSSS  S  SS  S  CS  SSSSS  SSSSS  SS  S  SSSSS
    S  SS  S  S  SS  S  SS  S  SS  SS  SS  S
  SSSS  SSSS  S  SS  S  S  SS  SSSS  SSSS  SSSS
    S  SS  S  S  S  SS  S  SS  S  SS  S  SS  S
    S  SS  S  S  SSS  SS  SS  S  SS  S  SS
  SSSSS  SS  SS  SS  SSSS  S  SS  S  SS  SSSS

WIDS 1.0, R.9 - The Wireless Intrusion Detection System
by SY Chua, 07 Jan 2014, Updated 26 Feb 2014

Description :
This application sniff your surrounding wireless traffic and analyse for suspicious packets such as WEP/WPA/WPS attacks, wireless client switched to another access point, detection of possible Rogue AP, displaying AP with the same name and much more..

[!] NAC OUI Database not found!
You can download it @ https://raw2.github.com/SYWorks/wireless_ids/master/na
```

Sumber: Hasil Penelitian (2018)

Gambar 10. Tampilan Awal *Wireless IDS Script Python (wids.py)*

Meskipun tidak terjadi serangan pada jaringan *wireless* tetapi *wireless IDS script python (wids.py)* tetap menampilkan pesan seperti pada gambar 11:

```
Alert : Client [ 54:A0:50:17:84:4F ] Initially associated to [ C8:D7:19:95:BA:54 ]
is now associated to [ 1C:B7:2C:84:F2:80 ] .SAYANGKU
      BSSID [ 1C:B7:2C:84:F2:80 ]'s Name is [ RuangSeminar ]

Alert : Client [ F0:7B:CB:11:AA:C1 ] Initially associated to [ FC:E3:9C:99:77:20 ]
is now associated to [ 1C:B7:2C:84:F2:80 ] .@wifi.id
      BSSID [ 1C:B7:2C:84:F2:80 ]'s Name is [ RuangSeminar ]

Alert : Client [ 28:E3:47:87:6C:04 ] Initially associated to [ 20:AA:48:41:03:38 ]
is now associated to [ 1C:B7:2C:84:F2:80 ] .HackMePlease
      BSSID [ 1C:B7:2C:84:F2:80 ]'s Name is [ RuangSeminar ]

Alert : Client [ DC:85:DE:51:FC:D0 ] Initially associated to [ 00:23:69:4F:D2:82 ]
is now associated to [ 44:C3:49:D6:56:4A ] ..
      BSSID [ 00:23:69:4F:D2:82 ]'s Name is [ BinaInsani_Hotspot ]
      BSSID [ 44:C3:49:D6:56:4A ]'s Name is [ H-G800 ]

Alert : Client [ 54:A0:50:17:85:99 ] Initially associated to [ C8:B3:73:3A:71:A5 ]
is now associated to [ 1C:B7:2C:84:F2:80 ] .RUANG MEETING
      BSSID [ 1C:B7:2C:84:F2:80 ]'s Name is [ RuangSeminar ]
```

Sumber: Hasil Penelitian (2018)

Gambar 11. Tampilan Pesan Normal Tidak Terjadi Serangan

3.2. Pengujian

Pada penelitian ini dilakukan pengujian untuk memastikan bahwa *wireless IDS* dapat berjalan sesuai dengan kebutuhan. Berikut ini pengujian yang telah dilakukan:

a. Beacon Flooding

Tindakan penyerangan dengan mengirimkan "*Beacon Frames*" dengan membuat banyak *Fake AP*. Tindakan ini akan mengakibatkan jaringan *wireless* menjadi "*crash*" untuk beberapa saat. Untuk menjalankan pengujian *Beacon Flooding* gunakan perintah pada Gambar 12:

```
root@DebianGutsy:~# mdk3 wlan1 b -n BinaInsani_Hotspot
```

Sumber: Hasil Penelitian (2018)

Gambar 12. Perintah Melakukan *Beacon Flooding*

Berdasarkan dari pengujian di atas, di dapat hasil seperti gambar dibawah ini. Pada gambar tersebut dijelaskan bahwa terdapat permintaan untuk mengakses *wireless Binalnsani_Hotspot* yang terlalu tinggi dari *MAC address 00:4F:77:00:04:BF* pada gambar 13.

```
[!] Unusual high amount of association sent from [ 00:4F:77:00:04:BF ] to [ 00:23:69:4F:D2:82 ] with 14 association request detected
Note: If amount is too high, likely to be Association flood.
[ 00:23:69:4F:D2:82 ]'s SSID Name is [ BinaInsani_Hotspot ] and Privcy=OPN
Cipher= Authentication= Power=32
[ 00:23:69:4F:D2:82 ]'s NAC OUI is not found in NAC OUI Database.
[ 00:4F:77:00:04:BF ] is associated with access point [ 00:23:69:4F:D2:82 ]
[ 00:23:69:4F:D2:82 ]'s NAC OUI is not found in NAC OUI Database.
[ 00:23:69:4F:D2:82 ]'s SSID Name is [ BinaInsani_Hotspot ],
[ 00:4F:77:00:04:BF ]'s NAC OUI is not found in NAC OUI Database.
[i] 22/02/2018 12:40:17 - 2 concerns found...
```

Sumber: Hasil Penelitian (2018)

Gambar 13. Hasil Pengujian *Beacon Flooding*

b. Authentication DoS Mode

Tindakan penyerangan dengan mengirimkan "*Authentication Frame*" ke semua AP yang terdeteksi dalam jaringan *wireless*. Tindakan ini akan mengakibatkan *client (user wireless)*

tidak bisa melakukan koneksi ke AP, bahkan *client* (*user wireless*) yang sudah terkoneksi ke AP akan ter-reset dan putus koneksi. Berikut perintah untuk melakukan serangan DOS:

```
root@DebianGutsy:~# mdk3 wlan1 a -a 00:23:69:4F:D2:B2
```

Sumber: Hasil Penelitian (2018)

Gambar 14. Perintah Melakukan DoS Attack

Setelah melakukan perintah penyerangan seperti gambar diatas maka aplikasi wids.py akan menampilkan pesan seperti pada gambar 15.

```
[.] Detected authentication sent from [ CC:2D:83:B8:76:FB ] to [ 90:C7:D8:84:35:41 ] with 6 authentication request detected
[ 90:C7:D8:84:35:41 ]'s SSID Name is [ SALSA ] and Privicy=WPA2 Cipher=CCMP
/TKIP Authentication=PSK Power=-62
[ 90:C7:D8:84:35:41 ]'s MAC OUI is not found in MAC OUI Database.
[ CC:2D:83:B8:76:FB ] is associated with access point [ 90:C7:D8:84:35:41 ]
[ 90:C7:D8:84:35:41 ]'s MAC OUI is not found in MAC OUI Database.
[ 90:C7:D8:84:35:41 ]'s SSID Name is [ SALSA ].
[ CC:2D:83:B8:76:FB ]'s MAC OUI is not found in MAC OUI Database.
[i] 22/02/2018 12:12:11 - 2 concerns found...
Possibility : Authentication DOS attacks.
```

Sumber: Hasil Penelitian (2018)

Gambar 15. Hasil Pengujian DOS Attack

c. Penyerangan WPS Dengan Reaver

Serangan WPS relatif mudah menggunakan alat *open source* yang disebut *Reaver*. *Reaver* bekerja dengan mengeksekusi serangan *brute force* terhadap pin WPS. Berikut perintah yang digunakan untuk melakukan penyerangan WPS dengan *Reaver*:

```
root@DebianGutsy:~# reaver -i mon0 -b 00:23:69:4F:D2:B2 -vv
```

Sumber: Hasil Penelitian (2018)

Gambar 16. Perintah Serangan WPS Reaver

Terdeteksi *wireless* BinaInsani_Hotspot telah di serang oleh MAC address 00:EC:0A:4D:70:68 dengan mengirimkan permintaan *autentikasi*. Berikut hasil pengujian WPS terlihat pada gambar 17.

```
[.] Detected authentication sent from [ 00:EC:0A:4D:70:68 ] to [ 00:23:69:4F:D2:B2 ] with 23 authentication request detected
[ 00:23:69:4F:D2:B2 ]'s SSID Name is [ BinaInsani_Hotspot ] and Privicy=OPEN
Cipher= Authentication= Power=-58
[ 00:23:69:4F:D2:B2 ]'s MAC OUI is not found in MAC OUI Database.
[ 00:EC:0A:4D:70:68 ] is associated with access point [ 00:23:69:4F:D2:B2 ]
[ 00:23:69:4F:D2:B2 ]'s MAC OUI is not found in MAC OUI Database.
[ 00:23:69:4F:D2:B2 ]'s SSID Name is [ BinaInsani_Hotspot ].
[ 00:EC:0A:4D:70:68 ]'s MAC OUI is not found in MAC OUI Database.
[i] 26/02/2018 13:14:58 - 1 concerns found...
```

Sumber: Hasil Penelitian (2018)

Gambar 17. Hasil Pengujian WPS Dengan Reaver

d. WIDS/WIPS/WDS Confusion

Serangan ini merupakan tindakan pengacauan terhadap IDS dan IPS jaringan *wireless* serta WDS *routing*. Berikut ini perintah untuk melakukan penyerangan WIDS/WIPS/WDS *confusion* terlihat pada gambar 18.

```
root@DebianGutsy:~# mdk3 mon0 w -e BinaInsani_Hotspot -c 9 -z
```

Sumber: Hasil Penelitian (2018)

Gambar 18. Perintah Pengujian WIDS/WIPS/WDS Confusion

Pada gambar 19 terdapat persamaan nama *Access Point* (AP) dengan nama (ESSID) Redmi, banyak kemiripan antara AP satu dengan satunya namun yang membedakan MAC address kedua AP tersebut:

```
[i] Access Point Using The Same Name
BSSID : 0C:98:38:1E:D7:D1 Privacy : WPA2 Cipher : CCMP
Auth : PSK ESSID : Redmi
Client : 0 client Channel : 1 Speed : 54 MB
Power : 28
BSSID : 0C:98:38:0D:5F:D5 Privacy : WPA2 Cipher : CCMP
Auth : PSK ESSID : Redmi
Client : 2 client Channel : 1 Speed : 54 MB
Power : 34
```

Sumber: Hasil Penelitian (2018)

Gambar 19. Hasil Pengujian WIDS/WIPS/WDS

e. WPA Downgrade Test

Serangan ini merupakan tindakan dengan menggunakan *deauthenticates station* AP untuk mengirimkan paket WPA yang menyebabkan WPA down. Berikut ini perintah untuk melakukan serangan WPA downgrade terlihat pada gambar 20.

```
root@DebianGutsy:~# mdk3 mon0 g -t 00:23:69:4F:D2:B2
```

Sumber: Hasil Penelitian (2018)

Gambar 20. Perintah Pegujian WPA Downgrade

Pada hasil pengujian gambar dibawah telah terjadi penyerangan WPA attack tanpa handshake. MAC address EC:D0:9F:A5:52:8D membanjiri paket *deauthentication* kepada MAC address 00:23:69:4F:D2:B2 dengan nama BinaInsani_Hotspot. Artinya ada serangan yang ditujukan kepada jaringan wireless STMIK Bina Insani dengan jenis serangan WPA attack.

```
[.] Death Flood detected calling from [ EC:D0:9F:A5:52:8D ] to [ 00:23:69:4F:D2:B2 ] with 12 deauth packets
[ 00:23:69:4F:D2:B2 ]'s SSID Name is [ BinaInsani_Hotspot ] and Privicy=OPN
Cipher= Authentication= Power=-25
[ 00:23:69:4F:D2:B2 ]'s MAC OUI is not found in MAC OUI Database.
[ EC:D0:9F:A5:52:8D ] is associated with access point [ 00:23:69:4F:D2:B2 ]
[ 00:23:69:4F:D2:B2 ]'s MAC OUI is not found in MAC OUI Database.
[ 00:23:69:4F:D2:B2 ]'s SSID Name is [ BinaInsani_Hotspot ].
[ EC:D0:9F:A5:52:8D ]'s MAC OUI is not found in MAC OUI Database.
Handshake Found [ 0 ]

[i] 22/02/2018 12:06:07 - 1 concerns found...
Possibility : WPA attacks.
```

Sumber: Hasil Penelitian (2018)

Gambar 21. Hasil Pengujian WPA Downgrade

f. WPA and WEP All Cracking Method

WPA cracking salah satu jenis serangan mdk3 dengan mencoba masuk atau mendapatkan akses dari jaringan wireless. Berikut perintah untuk melakukan WPA cracking terlihat pada gambar 22.

```
root@kali:~# mdk3 wlan1 g -g -t 1C:B7:2C:84:F2:B0
```

Sumber: Hasil Penelitian (2018)

Gambar 22. Perintah Serangan WPA Cracking

Terdapat pesan peringatan dengan membanjiri paket *deauthentication* dari MAC address 01:00:5E:7F:FF:FA menuju 1C:B7:2C:84:F2:B0 dimana MAC address tersebut milik wireless RuangSeminar dengan kemungkinan serangan WPA, MDK3-WPA Downgrade.

```
[.] Deauth Flood detected calling from [ 01:00:5E:7F:FF:FA ] to [ 1C:B7:2C:84:F2:80 ] with 28 deauth packets
[.] Dissassociation Flood detected calling from [ 01:00:5E:7F:FF:FA ] to [ 1C:B7:2C:84:F2:80 ] with 37 disassociation packets
[ 1C:B7:2C:84:F2:80 ]'s SSID Name is [ RuangSeminar ] and Privicy=WPA2 Cipher=CCMP Authentication=PSK Power=-38
[ 1C:B7:2C:84:F2:80 ]'s MAC OUI is not found in MAC OUI Database.
[ 01:00:5E:7F:FF:FA ] is associated with access point [ 1C:B7:2C:84:F2:80 ]
[ 1C:B7:2C:84:F2:80 ]'s MAC OUI is not found in MAC OUI Database.
[ 1C:B7:2C:84:F2:80 ]'s SSID Name is [ RuangSeminar ].
[ 01:00:5E:7F:FF:FA ]'s MAC OUI is not found in MAC OUI Database.
Possible NDK3 WPA Downgrade..

[i] 26/02/2018 12:59:36 - 3 concerns found...
Possibility : WPA , NDK3 - WPA Downgrade Test attacks.
```

Sumber: Hasil Penelitian (2018)

Gambar 23. Hasil Pengujian WPA Cracking

3.3. Monitoring

Setelah melakukan implementasi *wireless IDS script python (wids.py)* maka langkah selanjutnya dalam metode NDLC melakukan tahapan *monitoring* terhadap jaringan *wireless* STMIK Bina Insani melalui (*wids.py*) tersebut. Berikut ini adalah hasil monitoring-nya terdapat pada gambar 24.

```
File Edit View Search Terminal Help
root@DebianGutsy: /home/lirva32/test_ids

Wireless Device [ DA:A1:19:FA:00:84 ] is not associated to any network and did not probe for any SSID ..
[ DA:A1:19:FA:00:84 ]'s MAC OUI belongs to [ ].
Wireless Device [ DA:A1:19:CB:4C:00 ] is not associated to any network and did not probe for any SSID ..
[ DA:A1:19:CB:4C:00 ]'s MAC OUI belongs to [ ].
Wireless Device [ 94:D0:29:41:DA:65 ] is not associated to any network and did not probe for any SSID ..
[ 94:D0:29:41:DA:65 ]'s MAC OUI belongs to [ ].
Wireless Device [ DA:A1:19:2B:27:DB ] is not associated to any network and did not probe for any SSID ..
[ DA:A1:19:2B:27:DB ]'s MAC OUI belongs to [ ].
Wireless Device [ A4:17:31:35:95:75 ] is not associated to any network and did not probe for any SSID ..
[ A4:17:31:35:95:75 ]'s MAC OUI belongs to [ ].
Wireless Device [ DA:A1:19:CE:A6:04 ] is not associated to any network and is probing for [ Lantai5 / Lantai3 / Lantai4 ] ..
[ DA:A1:19:CE:A6:04 ]'s MAC OUI belongs to [ ].

[i] 22/02/2018 11:46:42 - Did not detect any suspicious activity ...

[18] Refreshing after 20.0 seconds... please wait..
```

Sumber: Hasil Penelitian (2018)

Gambar 24. Hasil Monitoring Jaringan Wireless

3.4. Management

Tahap terakhir dalam metode NDLC yaitu *management*. Pada tahap ini dilakukan *backup file wids.py*. Backup dapat digunakan ketika *script* tersebut mengalami kerusakan. Sedangkan untuk menciptakan *file log* dapat menjalankan *script* dengan model *result teks*. Berikut ini perintah yang dapat dilakukan untuk membuat *file log*.

```
root@DebianGutsy: /home/lirva32/test_ids# python wids.py -i mon0 > log_file.txt
```

Sumber: Hasil Penelitian (2018)

Gambar 25. Perintah Pembuatan File Log

Maka *file log* akan tersimpan dengan nama "log_file" dengan ekstensi ".txt"

```
root@DebianGutsy: /home/lirva32/test_ids# ls
log_file.txt  lo.x  wids.py
```

Sumber: Hasil Penelitian (2018)

Gambar 26. File Log Yang Tersimpan

Untuk dapat melihat hasil *log* yang tersimpan, dapat menggunakan perintah pada Gambar 27:

```
root@DebianGutsy: /home/lirva32/test_ids# vi log_file.txt
```

Sumber: Hasil Penelitian (2018)

Gambar 27. Perintah untuk membuka file log

Setiap serangan yang terjadi pada jaringan *wireless* dapat terekam dalam log yang telah dibuat. *File log* ini berguna agar *administrator* dapat melakukan evaluasi terhadap serangan-serangan yang terjadi pada jaringan *wireless* terlihat pada gambar 28.

```

log_file.txt (/home/Urva32/test_ids) - VIM
File Edit View Search Terminal Help
[0:37m This application sniff your surrounding wireless traffic and analyse fo
r suspicious packets such as
[0:37m WEP/WPA/WPS attacks, wireless client switched to another access point,
detection of possible Rogue AP,
[0:37m displaying AP with the same name and much more..

[0:37m [1:31m [0:37m [1:31mMAC OUI Database not found !
[0:37m [1:32m [0:32mYou can download it @ [0:34mhttps://raw2.github.co
m/5YWorks/wireless-ids/master/mac-oui.db

[1:33mParameter set:
[1:37mSelected interface      : [1:31mmon0

[0:37m [1:34mi [0:37m [1:37m [1:36mEntering Semi-Interactive Mode..
[0:37m [1:32m [1:32mStarted      : [0:32m2018-02-26 16:50:12

[0:37m [1:34mi [0:37m [1:37mMonitor Selection
[0:37m [1:32m [0:37mSelected Monitoring Interface ==> [1:31mmon0

@
@
@
12,1 57%

```

Sumber: Hasil Penelitian (2018)

Gambar 28. Tampilan *File Log Monitoring*

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa jaringan *wireless* sangat rentan dengan berbagai serangan. Untuk itu dibutuhkan suatu sistem keamanan yang mampu mendeteksi serangan-serangan yang tertuju pada jaringan *wireless*. salah satu solusi dapat diterapkan untuk meningkatkan keamanan yaitu dengan menggunakan *Intrusion Detection System* (IDS). Mengimplementasikan *Intrusion Detection System* dapat membantu *Administrator* jaringan komputer untuk mengetahui apabila terjadi serangan-serangan pada jaringan *wireless*. Adanya serangan-serangan pada jaringan *wireless* dapat terlihat melalui pesan peringatan (*alert*) yang diberikan oleh sistem intrusi deteksi. Adanya kerentanan pada jaringan *wireless* dapat dilihat dari beberapa pengujian yang telah dilakukan untuk memastikan bahwa sistem intrusi deteksi sudah dapat berjalan sebagaimana seharusnya. Penelitian yang dilakukan mengenai implementasi *wireless intrusion detection system* ini selanjutnya dapat dikembangkan dari sisi pesan peringatan (*alert*) yang dapat dikirimkan melalui *email* atau SMS agar *Administrator* jaringan komputer tidak harus terus berada di depan komputer untuk memantau setiap aktivitas yang terjadi pada jaringan *wireless*.

Referensi

- Aditya A. 2011. Mahir Membuat Jaringan Komputer. Jakarta: Dunia Komputer.
- Beaver K. 2016. Hacking For Dummies 5th Edition. New Jersey: John Wiley & Sons Inc. 172.
- Budiman SA, Catur I, Sholeh M. 2014. Implementasi Intrusion Detection System (IDS) Pada Server Debian Menggunakan Jejaring Sosial Sebagai Media Notifikasi. Jurnal JARKOM. 2(1): 36-45.
- Course Technology Cengage Learning. 2011. 12065-2919. Network Defense Security Policy and Threats Vol 2. New York: EC-Council Press.
- Course Technology Cengage Learning. 2011. 12065-2919. Penetration Testing Network & Perimeter Testing Vol 3. New York: EC-Council Press.
- Edwards J, Bramante R. 2009. Networking Self-Teaching Guide: OSI, TCP/IP, LANs, MANs, WANs, Implementation, Management, and Maintenance. Indianapolis: Wiley Publishing, Inc. 241.

- Kizza JM. 2015. Guide to Computer Network Security Third Edition. London: Springer Verlag. 43.
- Masse FA, Hidayat AN, 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.
- Mentang R, Sinsuw AAE, Najoo XBN. 2015. Perancangan Dan Analisis Keamanan Jaringan Nirkabel Menggunakan Wireless Intrusion Detection System. E-Journal Teknik Elektro dan Komputer. 5(7): 35-44.
- Nurhidayat A, Bata J, Sihombing D. 2015. Wireless Intrusion Detection System Using Open Source Tool. Seminar Nasional Teknologi Informasi dan Komunikasi 2015 (SENTIKA 2015). ISSN: 2089-9815.
- Odom W. 2012. 1624. CCNA ICND2 640-816 Official Cert Guide Third Edition. Indianapolis: Cisco Press.
- Perkasa MG, Ismail SJI. 2015. Implementasi Wireless IDS (Intrusion Detection System) Untuk Monitoring Keamanan Jaringan Berbasis Kismet. e-Proceeding of Applied Science. Bandung. 2170-2174.
- Piper S, CISSP, SFCP. 2011. Intrusion Prevention System For Dummies. Indianapolis: Wiley Publishing, Inc. 3.
- Rehim R. 2016. Effective Python Penetration Testing. Birmingham: Packt Publishing. 1.
- Sadikin R. 2012. Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java. Yogyakarta: C.V Andi Offset.
- Sahala A. 2014. Konsep dan Implementasi Jaringan dengan Linux Ubuntu. Semarang: Wahana Komputer. 26.
- Sofana I. 2013. Membangun jaringan komputer, mudah membuat jaringan komputer (wire&wireless) untuk pengguna windows dan linux. Bandung: Informatika Bandung.
- Velu VK. 2017. Mastering Kali Linux for Advanced Penetration Testing Second Edition. Birmingham: Packt Publishing. 12.
- Wagito. 2007. Jaringan Komputer Teori dan Implementasi Berbasis Linux. Yogyakarta: Gava Media.
- Warman I, Andrian A. 2017. Analisis Kerja Load Balancing Dua Line Koneksi Dengan Metode Nth. Jurnal TEKNOIF. 5(1): 57-58.
- Wiyanto P, Hamzah A, Sholeh M. 2014. Aplikasi Monitoring Keamanan Jaringan Dengan Menggunakan IDS dan Router Mikrotik. Jurnal JARKOM. 2(1): 89-98.
- Zam EZ. 2014. Cara Mudah Membuat Jaringan Wireless. Jakarta: PT Elex Media Komputindo. 1.