MANAJEMEN *DOMAIN NAME SERVER* MENGGUNAKAN STANDAR *NATIONAL INSTITUTE OF STANDARD AND TECHNOLOGY (NIST) 800-81r1*

I Gede Putu Krisna Juliharta[1)], I Gede Oka Antara[2)], Made Henny Aryani[3)]

[1)] Sistem Informasi STMIK STIKOM Bali
[2)3)] Sistem Komputer STMIK STIKOM Bali
Jl. Raya Puputan Renon No. 86 (0361) 24445
Email : krisna@stikom-bali.ac.id *[1)],* okanta43@gmail.com *[2)],* henny.aryani@gmail.com *[3)]*

Abstrak

Internet merupakan jaringan komputer terbesar didunia, sehingga untuk menghubungi komputer atau perangkat dalam komputer berkomunikasi memerlukan nama yang unik. Salah satu dapat memiliki nama yang unik adalah menggunakan domain name system (DNS). DNS dikelola oleh mesin yang bernama DNS server dan memiliki ancaman keamanan yang cukup besar. Dalam penelitian yang dibuat dilakukan perbandingan antara DNS server standar dan DNS menggunakan NIST Standar menggunakan tiga alat yaitu nessus, openVAS, dan nmap. Untuk DNS standar NIST ada enam tahapan yang dilakukan yaitu manajemen versi, manajemen spoofing, manajemen zona tranfer, manajemen dynamic update, manajemen query, dan manajemen access list. Hasil yang diperoleh telah dibuktikan bahwa DNS dengan standar NIST memiliki performa yang lebih baik.

Kata kunci: *Manajemen, DNS, NIST, Server*

1. Pendahuluan

Internet merupakan jaringan komputer terbesar di dunia, dengan jumlah pengguna lebih dari 580 juta. Dari perspektif pengguna, setiap resource atau jaringan yang lebih kecil tentunya memiliki nama yang unik sehingga bisa dicari untuk proses komunikasi data, salah satu nama unik tersebut disebut sebagai domain name. domain name dikelola oleh perangkat komputer yang disebut sebagai domain name server (DNS)[1].

DNS merupakan salah satu layanan yang berisiko mengalami serangan, beberapa serangan DNS yang ditemukan adalah DNS cache snooping, DNS poisoning, DNS spoofing, dan sebagainya tentunya tujuannya berbagai macam seperti membelokkan arah browsing menuju website yang salah. Salah satu ancaman yaitu Cache snooping bisa digunakan untuk menentukan situs/host, siapa klien dan penggunanya, dan informasi lain yang berguna bagi penyerang. Selain banyaknya ancama seperti yang sudah disebutkan sebelumnya ancaman terhadap DNS bisa juga digunakan untuk melihat software yang digunakan sebuah host dari resource record yang berisi alamat update software.

Metodenya bisa dengan cara mengirim request non-rekursif ke server DNS tentang resource record. Jika record ada di cache, server akan memberikan jawabannya dalam query. Cache snooping bisa dilakukan juga dengan mengirimkan request rekursif lalu menganalisis TTL yang dikembalikan query dan jumlah waktu yang diperlukan server untuk merespon[2] .

Berdasarkan besarnya ancaman tersebut tentunya sudah seharusnya dilakukan proses manajemen keamanan jaringan dengan salah satu cara melakukan pengukuran terhadap jaringan yang digunakan. Ada beberapa alat ukur yang dapat digunakan seperti nessus, openVAS, dan nmap.

Nessus sendiri merupakan sebuah program yang memang dirancang untuk security scanner yang berfungsi untuk mengaudit keamanan sebuah sistem. Menurut survey membuktikan pada tahun 2000, 2003, dan 2006 yang dilakukan oleh sectools.org, Nessus merupakan vulnerability scanning yang memiliki license yang sangat mahal namun memiliki juga yang bersifat free untuk kepentingan non komersial. Walaupun versi free namun nessus tetaplah memiliki kemampuan yang baik sehingga wajar sectool menempatkan nessus sebagi yang terbaik sebanyak tiga kali. Nessus sendiri dibangun oleh Renuad Deraison pada tahun 1998 dengan tujuan awal memberikan referensi mengenai *security tools* terkait remote *security scanner* yang mampu mendeteksi kerentatan potensial yang ada di dalam sistem. Beberapa fitur yang dapat digunakan dalam versi free saat ini adalah audit configurasi, vulnerability analysis, mobile device audit dan memberikan laporan[3]. Hal ini yang menjadikan dasar untuk menggunakan nessus dalam penelitian ini, selain penggunaan openVAS yang memiliki sama namun dikelola oleh komunitas openscource. Tools yang ketiga yang digunakan adalah Network Mapper (Nmap). Nmap memiliki fungsi yang sedikit berbeda. Ia dirancang untuk memeriksa jaringan besar secara cepat, meskipun ia dapat pula bekerja terhadap host tunggal.

Untuk mendukung penelitian dilakukan kajian pustakan mengenai pengelolaan DNS. Dari beberapa kajian selain dari NIST dan Infosec, ada dua artikel ilmiah yang juga dijadikan fokus kajian. Yang pertama berjudul, Pengelolaan dinamik domain name server menggunakan DJNDNS oleh Arif kurniawan dan rekan [4], serta yang kedua yang berjudul implementasi dan optimasi switching DNS[5]. Kedua penelitian memiliki keunikan masing masing namun sama sama

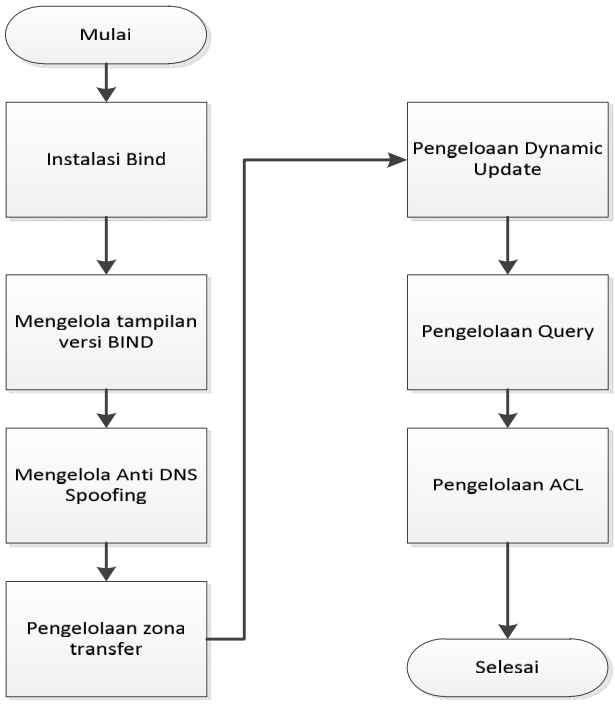
menggunakan DNS sebagai topik penelitian, namun dari kedua penelitian tersebut belum ada yang melakukan manajemen DNS dengan mengukur tingkat kerentanan yang dimiliki oleh DNS tersebut.

Atas dasar kajian pustaka, ancaman dan kemampuan alat ukur keamanan tersebut maka dilakukan penelitian yang berjudul manajemen Domain Name Server menggunakan National Institute Standard and Technology (NIST) 800-81r1. Penelitian melakukan perbandingan antara DNS standar dengan DNS yang menggunakan tahapan manajemen sebanyak tujuh langkah. Hasilnya berupa tabel perbandingan tingkat keamanan antara DNS standar dan DNS yang dikonfigurasi menggunakan NIST.

2. Pembahasan

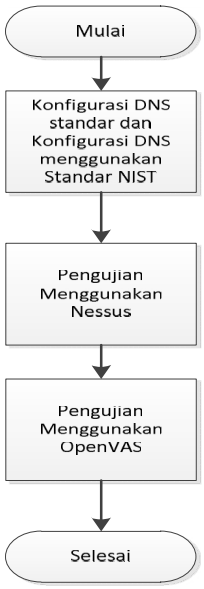
2.1 Metode Penelitian

Proses awal penelitian adalah dengan menggambarkan alur dari manajemen DNS menggunakan NIST. Dengan tapan manajemen sebanyak 7 langkah.



Gambar 1. Flow Chart Proses Pengelolaan DNS

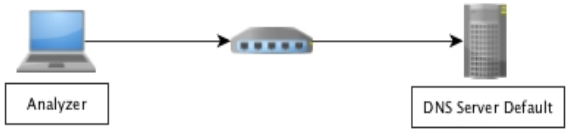
Proses manajemen Domain Name Server (DNS) menggunakan tahapan seperti pada gambar 1 Flow chart proses pengelolaan DNS. Dimulai dengan instalasi software DNS bernama BIND, dilanjutkan dengan mengelola versi dari BIND hingga pengelolaan Access List (ACL). Tahapan pada gambar 1 diatas berdasarkan manajemen BIND menggunakan standar NIST.



Gambar 2 Proses pengujian

Untuk mendapatkan hasil dalam penelitian manajemen *DNS Server* Menggunakan Standar Keamanan *National Institute of Standards and Technology (NIST)* diperlukan dilakukan langkah pengujian seperti pada gambar 2. Selanjutnya dilakukan proses pengujian dengan cara seperti gambar 3 dan gambar 4 skenario untuk mengetahui performa dari *DNS Server* tanpa keamanan dan *DNS Server* menggunakan standar keamanan *NIST* dimana terdapat komputer *analyzer* masing – masing skenario agar dapat menemukan performansi pada kedua aplikasi tersebut dan melihat kekurangan dan kelebihan pada kedua *DNS Server* tersebut. Skenario yang akan diterapkan dalam penelitian ini sebagai berikut :

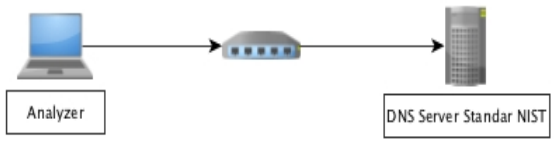
1. Skenario pertama analisa pada *DNS Server* Default :



Gambar 3. Skenario pertama

Pada Gambar 3. skenario pertama dalam proses pengukuran menggunakan 1 komputer sebagai *DNS Server*, 1 komputer sebagai *analyzer* dan satu *switch, switch* berfungsi untuk menghubungkan komputer *server* ke komputer *analyzer* yang akan mengakses *server* tersebut. Untuk menguji *DNS Server* adalah ketika komputer *analyzer* terhubung dengan *DNS Server* kemudian akan di uji tingkat keamanannya oleh komputer *analyzer* dengan cara melakukan *scanning vulnerability DNS server* menggunakan *tools / software Nessus, Openvas, dan Nmap.*

2. Skenario kedua analisa pada *DNS Server* menggunakan standar keamanan *NIST* :



Gambar 4. Skenario kedua

Pada gambar 4. skenario kedua dalam proses pengukuran menggunakan 1 komputer sebagai *DNS Server*, 1 komputer sebagai *analyzer* dan satu *switch, switch* berfungsi untuk menghubungkan komputer *server* ke komputer *analyzer* yang akan mengakses *server* tersebut. Untuk menguji *DNS Server* adalah ketika komputer *analyzer* terhubung dengan *DNS Server* kemudian akan di tes keamanannya oleh komputer *analyzer* dengan cara melakukan *scanning vulnerability DNS server* menggunakan *tools / software Nessus, Openvas, dan Nmap*

2.2 Manajemen Domain Name Server.

Tahap pertama, menggunakan versi *BIND* terbaru, dimana sebagian besar versi *BIND* lama mempunyai *bug* dan dapat dieksploitasi sehingga akan menimbulkan serangan. Dengan menggunakan versi terbaru diharapkan dapat meminimalkan adanya *bug* dan eksploitasi yang bisa dilakukan terhadap *name-server* karena *bug* pada versi sebelumnya biasanya sudah diperbaiki pada versi yang lebih baru.

```
apt-get install bind9
```

Tahap kedua, pengeloaan Permintaan versi perangkat lunak *BIND* biasanya dilakukan penyerang untuk mengetahui versi dari *BIND* dan akan membandingkan dengan daftar versi yang mempunyai kelemahan. Selanjutnya si penyerang akan menyerang dengan serangan atau eksploitasi yang spesifik pada versi *BIND* tertentu. Kita harus menolak permintaan versi dan mencatat permintaan tersebut, yaitu dengan mengedit */etc/bind/named.conf* dan *zone*.

```
options {
      version none;
};
```

Tahap Ketiga, Pengelolaan permintaan rekursif pada server DNS dengan tujuan meminimalkan ancaman *DNS spoofing*/*cache poisoning*. Konfigurasi dilakukan pada file named.conf dan menambahkan perintah dibawah ini

```
options {
      recursion  no;
};
```

Tahap keempat, melakukan pembatasan *zone-transfer* hanya kepada *secondary name-server* atau mesin yang benar-benar mempunyai *authority* terhadap informasi atau data pada *zone* yang bersangkutan. Pada */etc/bind/named.conf* tambahkan baris allow transfer { [ip-atau-jaringan-yang-boleh-melakukan-zone-transfer]}; di bagian options atau zone. Pada penelitian ini dibuat sebuah kelompok ip dengan nama xfer. Kelompok ip ini akan dijelaskan pada pembahasan pengelolaan acl.

```
options {
      allow-transfer  { xfer; };
};
```

Tahap Kelima, *Dynamic update* harus dibatasi pada *server DNS* kita dari permintaan *update* informasi *host*/domain. Pada */etc/bind/named.conf* tambahkan baris allow update { [ip-atau-jaringan-yang-boleh-melakukan-update]}; di bagian options atau zone.

```
options {
      allow-update  { xfer; };
};
```

Tahap keenam, membatasi *query* atau permintaan terhadap data atau informasi yang ada dalam *database-cache* atau berkas *zone server DNS*. Pada */etc/bind/named.conf* tambahkan baris allow query { [ip-atau-jaringan-yang-boleh-melakukan-query]}; di bagian options atau zone.

```
options {
      allow-query  { xfer; };
};
```

Pada tahap ini juga dilakukan pembagian pengelolaan name server menjadi dua yaitu jaringan internal dan eksternal dengan tujuan untuk menghindari informasi atau data dimanfaatkan oleh orang yang tidak bertanggung jawab.

```
view "internal-in" in {
      match-clients { trusted; };
      recursion no;
      additional-from-auth no;
      additional-from-cache no;
      zone "." in {
            type hint;
            file
"/etc/bind/db.cache";
      };
      zone   "0.0.127.in-addr.arpa"
in {
            type master;
            file
"/etc/bind/db.127";
            allow-query { any; };
            allow-transfer  { none;
};
      };
      zone "stikombali.ac.id" in {
            type master;
            file
"/etc/bind/db.stikom";
      };
      zone            "2.168.192.in-
```

Tahap Ketujuh, Manajemen Access List (*ACL).* *ACL* dapat membatasi layanan DNS kepada *host*/jaringan yang dipercaya yaitu dengan memasukkan data *host* dan jaringan tersebut ke dalam berkas konfigurasi. Data atau informasi *database-cache* server DNS hanya dapat diakses oleh *host* dan jaringan yang terpercaya sehingga mempersempit kesempatan seorang penyusup untuk melakukan pengambilan data *host* dan jaringan pada suatu perusahaan.

```
acl "xfer" {
      192.168.2.0/24;
};
acl "trusted" {
      192.168.2.0/24;
};
acl "bogon" {
      0.0.0.0/8;
      1.0.0.0/8;
      2.0.0.0/8;
      192.168.3.0/24;
      192.168.4.0/24;
};
```

2.3 Pengujian Manajemen Domain

Proses terakhir dalam manajemen Domain Name Server adalah proses pengujian untuk menghasilkan bukti bahwa dengan menggunakan standar NIST performa DNS, lebih baik daripada menggunakan konfigurasi secara default. Proses Pengujian diantaranya melakukan *scanning* antara *DNS Server Default* dan *DNS Server* Standar *NIST* serta menggunakan *tools/software* pengujian seperti *Nessus, Openvas dan Nmap*, terdapat perbedaan pada tingkat keamanan[6].

Perbandingan hasil *scanning* antara *DNS Server Default* dan *DNS Server Standar NIST* menggunakan *tool Nessus* dijelaskan pada tabel 1.

Tabel 1. Perbandingan menggunakan tools Nessus

Perbandingan Scanning DNS Server menggunakan tool Nessus	
DNS Server Default	**DNS Server Standar NIST**
DNS Server Zone Transfer Information Disclosure (AXFR)	-
DNS Server Detection	-
DNS Server BIND version Directive Remote Version Detection	-
DNS Server hostname.bind Map Hostname Disclosure	-
DNS Server Version Detection	-
Common Platform Enumeration (CPE)	-
Device Type	-
Host Fully Qualified Domain Name (FQDN) Resolution	Host Fully Qualified Domain Name (FQDN) Resolution
ICMP Timestamp Request Remote Date Disclosure	ICMP Timestamp Request Remote Date Disclosure
OS Identification	-
TCP/IP Timestamps Supported	-
Traceroute Information	Traceroute Information
Ethernet Card Manufacturer Detection	Ethernet Card Manufacturer Detection

Pada tabel 1 adalah hasil dari perbandingan *DNS Server Default* dan *DNS Standar NIST*, pada hasil *scanning DNS Server Default* masih terdapat banyak *vulnerability* yang terkait dengan *DNS* diantaranya DNS Server Zone Transfer Information Disclosure (AXFR) (memiliki tingkat severity Medium), DNS Server Detection, DNS Server BIND version Directive Remote Version Detection, DNS Server hostname.bind Map Hostname Disclosure, DNS Server Version Detection. Sedangkan hasil scanning *DNS Server Standar NIST vulnerability* dari *DNS* sudah berkurang.

Perbandingan hasil scanning antara *DNS Server Default* dan *DNS Server Standar NIST* menggunakan tool *Openvas* dijelaskan pada tabel 2.

Tabel 2. Perbandingan Menggunakan Tools OpenVAS

Perbandingan Scanning DNS Server menggunakan tool Openvas	
DNS Server Default	**DNS Server Standar NIST**
TCP timestamps	-
CPE Inventory	CPE Inventory
ICMP Timestamp Detection	ICMP Timestamp Detection
OS fingerprinting	OS fingerprinting
Traceroute	Traceroute
DNS Server Detection	-
DNS Server Detection	-
Determine which version of BIND name daemon is running	-

Pada tabel 2 adalah hasil dari perbandingan *DNS Server Default* dan *DNS Standar NIST*, pada hasil *scanning DNS Server Default* masih terdapat banyak *vulnerability* yang terkait dengan *DNS* diantaranya TCP timestamps DNS Server Detection (memiliki tingkat serverity Low), Determine which version of BIND name daemon is running. Sedangkan hasil *scanning DNS Server Standar NIST vulnerability* dari *DNS* sudah berkurang.

Perbandingan hasil scanning antara *DNS Server Default* dan *DNS Server Standar NIST* menggunakan *tool Nmap* dijelaskan pada tabel 3

Tabel 3. Perbandingan Menggunakan Tools Nmap

Perbandingan Vulnerability DNS Server menggunakan tool Nmap	
DNS Server Default	**DNS Server Standar NIST**
Port 53 Open	-
OS Detection	-

Pada tabel 3 adalah hasil dari perbandingan *DNS Server Default* dan *DNS Standar NIST*, pada hasil *scanning DNS Server Default port* 53 masih terbuka dan SIstem Operasi masih bisa terdeteksi oleh *tool Nmap*. Sedangkan hasil *scanning DNS Server Standar NIST*

port 53 sudah tertutup dan Sistem Operasi dari *DNS Server* tersebut sudah tidak terdeteksi oleh *tool Nmap*.

3. Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Konfigurasi dan manajemen *DNS* telah dilakukan dengan baik
2. Terlihat perbedaan antara *DNS Default* dan *DNS* Standar *NIST* dari hasil *scanning vulnerability* menggunakan *tools Nessus*, dan *Openvas*.
3. Pengujian DNS Default masih terdapat banyak *vulnerability* mengenai DNS Server.
   a. Pengujian menggunakan Nesus terdapat 13 *vulnerability,* dari 13 *vulnerability* terdapat 5 *vulnerability* terkait dengan *DNS* diantaranya 1 *vulnerability* tingkat *MEDIUM* dan 4 INFO mengenai *DNS*.
   b. Pengujian menggunakan Openvas terdapat 8 *vulnerability*, dari 8 *vulnerability* terdapat 3 *vulnerability* terkait dengan *DNS* dan 1 *vulnerability level Low*.
4. Dengan menggunakan *DNS* Standar *NIST*, *vulnerability* yang ada pada *DNS Default* dapat berkurang.

Daftar Pustaka

[1] Ramaswamy Chandramouli, Scott Rose, "Secure Domain Name Server Deployment Guide" NIST Special Publication 800-81r1, 2010.
[2] R. Arend, R. Austein, R. Bolles, M. Larson, "DNS Security Introduction and Requirements,". RFC 4033, 2005.
[3] NN, "Network Scanning Using Nessus".Infosec Institute,2012.
[4] Arif Kurniawan, Sujoko Sumaryono, Addin Suwaswanto, "Pengelolaan Domain Name System Berbasis DJBDNS", Jurnal Penelitian Teknik Elektro. Vol. 3, No. 3. 2010.
[5] Moh. Nuril Rohman, "Implementasi dan Optimasi Switching Domain Name Server Untuk Filtering Konten Dengan Mikrotik Scheduler" UIN Sunan kalijaga, 2013.
[6] Priandoyo, Anjar. (2006). *Vulnerablity Assessment untuk Meningkatkan Kesadaran Pentingnya Keamanan Informasi*. Jurnal Sistem Informasi. Vol. 1, No. 2, pp.73-83.

Biodata Penulis

I Gede Putu Krisna Juliharta,memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Informatika UPN Veteran Yogyakarta, lulus tahun 2007. Memperoleh gelar Magister Teknik (M.T) Program Pasca Sarjana Magister Teknik Elektro Universitas Udayana Bali, lulus tahun 2010.Saat ini menjadi Dosen di STMIK STIKOM Bali.

I Gede Oka Antara, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Komputer STMIK STIKOM Bali, lulus tahun 2015.

Made Henny Aryani, memperoleh gelar Sarjana Teknik (S.T.), JurusanTeknik informatika Universitas Atmajaya Yogyakarta, lulus tahun 2006. Saat ini sedang menempuh gelar Magister Teknik (M.T) Program Pasca Sarjana Magister Teknik Elektro Universitas Udayana Bali, dan aktif menjadi Dosen di STMIK STIKOM Bali.

ISSN : 2302-3805