BAB 1
Pendahuluan

1.1 Pendahuluan

Protokol Domain Name System (DNS) merupakan protokol yang digunakan untuk menerjemahkan alamat Internet Protocol (IP) sebuah *host* menjadi sebuah nama domain yang mudah diingat, dan sebaliknya untuk mengubah alamat domain menjadi alamat IP dari sebuah *host*. Protokol DNS diciptakan bukan untuk bertukar data, oleh sebab itu protokol DNS jarang dimonitoring dengan baik oleh Administrator. Celah inilah yang dimanfaatkan oleh penyerang untuk mengelabui Administrator. Kejahatan yang biasanya dilakukan oleh penyerang yaitu pencurian data melalui protokol DNS sering disebut DNS *tunneling* atau DNS Data Exfiltration.

Peneliti keamanan baru-baru ini menemukan sebuah *malware* yang berjenis *Remote Access Trojan* (RAT) yang menggunakan teknik baru yang sulit dideteksi oleh Administrator jaringan. Malware ini melakukan *Command and control* untuk mengendalikan komputer korban melalui protokol DNS, *malware* ini dinamakan *DNSMessenger* (Brumaghin & Grady, 2017).

Selain DNS*Messenger*, ada *malware* lain yang memanfaatkan protokol DNS untuk *command and control* ataupun *tunneling*, seperti yang terlihat pada gambar berikut:

Year	Malware name	Targets	Domains used
2011	Morto (Mullaney, 2011)	RDP / RAT	ms[.]jifr[.]co[.]cc, ms[.]jifr[.]net (Symantec, 2011)
2011	FeederBot (Dietrich et al., 2011)	Botnet	images[.]moviedyear[.]net (Dietrich, 2016)
2014	PlugX (Perigaud, 2014)	RDP / RAT	ns4[.]msftncsl[.]com (Vasilenko, 2013)
2014	FrameworkPOS (Rascagneres, 2016)	POS	a23-33-37-54-deploy- akamaitechnologies[.]com (Mendleta, 2016)
2015	Wekby (Josh Grunzweig and Lee, 2016)	Targeted	ns1[.]logitech-usa[.]com (Josh Grunzweig and Lee, 2016)
2015	BernhardPOS (Morphick, 2015)	POS	29a[.]de (Morphick, 2015)
2015	JAKU (Andy Settle and Toro, 2015)	Botnet	LS4[.]com (Andy Settle and Toro, 2015)
2016	MULTIGRAIN (Lynch, 2016)	POS	dojfgj[.]com (Security, 2016)
2017	DNSMessenger (Brumaghin and Grady, 2017)	Targeted	cspg[.]pw, algew[.]me (Brumaghin and Grady, 2017)

The domains in use by previously detected malware.

Gambar 1.1 Daftar Malware yang menggunakan DNS *tunneling*.

DNS *tunneling* juga sering digunakan untuk melewati *captive portal* atau *login hotspot* internet di tempat umum (Farnham, 2013). Namun yang lebih berbahaya dan merugikan daripada *bypass captive portal* adalah exfiltrasi data melalui protokol DNS. Data yang di-exfiltrasi dapat berupa rahasia dagang, kekayaan intelektual, data karyawan, data pelanggan dan sebagainya. DNS *tunnel* biasanya membutuhkan *software* yang

terpasang pada komputer korban untuk bekerja, jika terpasang pelaku kejahatan dapat mem-*bypass* seluruh kontrol keamanan perusahaan.

DNS *tunneling* juga membuat kualitas jaringan menjadi buruk. Sebuah penelitian menyimpulkan bahwa penggunaan DNS *tunneling* dapat meningkatkan *delay* pada keseluruhan jaringan sampai 140-1500 *ms, jitter* sampai 8-57 *ms,* dan DNS *Overhead* 200-2000% (Leijenhorst, Chin, & Lowe, 2008). Implikasinya adalah jika ada client di jaringan tersebut sedang melakukan panggilan VOIP atau *videocall*, nilai *jitter* yang tinggi menyebabkan suara dan video putus-putus, *delay* yang tinggi menjadikan suara dan video terlambat sampai ke tujuan, dan *header* DNS yang lebih besar atau DNS *overhead* akan membuat ukuran paket lebih besar sehingga memakan *bandwidth* lebih banyak. Untuk mendapatkan kualitas jaringan yang baik, Cisco merekomendasikan nilai *jitter* dibawah 30 *ms*, dan *delay* kurang dari 150 *ms* ("Acceptable Jitter and Latency," n.d.).

Administrator jaringan biasanya tidak terlalu memperhatikan trafik DNS pada jaringan mereka, karena protokol DNS merupakan sebuah protokol yang digunakan untuk mentranslasikan Alamat IP menjadi sebuah nama *domain*, sehingga kita dapat mengakses sebuah komputer atau *server* dengan nama, tanpa perlu mengingat-ingat alamat IP komputer atau *server* tersebut, bukan untuk bertukar data. Inilah yang dimanfaatkan oleh penyerang untuk melakukan *command and control* ataupun mencuri data dengan DNS *tunneling*.

Oleh karena itulah *traffic* DNS dalam sebuah jaringan perlu dipantau untuk mencegah terjadinya DNS *tunneling*. Administrator jaringan bisa memblokir semua *traffic* DNS untuk mencegah *tunneling*, namun itu bukan solusi yang ideal karena dapat menyulitkan pengguna jaringan untuk mengakses alamat *host* tujuan, pendekatan lain adalah dengan menggunakan DNS *Sinkhole* (Bruneau, 2010). DNS *Sinkhole* merupakan sebuah DNS *server* yang memberikan jawaban alamat IP yang salah *(spoofing)* dari sebuah *request* DNS, sehingga nama *domain* yang dituju tidak akan bisa diakses. Hal ini bisa dimanfaatkan untuk mencegah *malware* atau DNS *tunnel* untuk mengontak *server*-nya.

DNS *sinkhole* menggunakan daftar domain yang diblokir yang bisa didapatkan dari beberapa situs seperti urlblacklist.com, malwaredomains.com dan lainnya. Pendekatan seperti ini kurang efektif karena hanya akan memblokir nama *domain* yang ada dalam daftar. Oleh karena itu perlu dilakukan *monitoring* dan *logging* semua *traffic* DNS di jaringan kita. *Log* DNS dapat diambil dari berbagai sumber, seperti DNS *server* itu sendiri, *Intruder Detection Systems* (IDS), *Proxy*, maupun dari *log* pada komputer. Untuk mendeteksi adanya DNS *tunneling* dari *log* tersebut, dilakukan analisis secara manual

menggunakan *packet capture analyzer* seperti *wireshark*. Pendekatan seperti ini sulit dilakukan dan memakan waktu, apalagi jika kita ingin memvisualisasikan hasil analisis tersebut, kita akan membutuhkan *tools* lain. Pendekatan lain adalah dengan menggunakan metode payload analysis, dan traffic analysis (Farnham, 2013). Payload analysis dapat mendeteksi DNS tunneling tertentu saja, sedangkan *traffic analysis* dapat mendeteksi DNS *tunneling* secara universal.

Pendekatan yang kami lakukan dalam permasalahan ini adalah mendeteksi DNS *tunneling* memanfaatkan metode *traffic analysis* dengan jumlah *unique hostname* sebagai *indicator of compromise* menggunakan Elasticsearch. Elasticsearch mempunyai beberapa komponen yang dapat digunakan pada penelitian ini, yaitu Packetbeats, Kibana, dan Watcher. Packetbeats berperan sebagai *sniffer realtime* yang akan melakukan *capture traffic* DNS, Watcher akan memberikan notifikasi berupa *email* kepada Administrator jaringan jika terjadi DNS *tunneling*, dan Kibana akan digunakan sebagai *dashboard panel* visualisasi dan akan menampilkan grafik *bar* nama *domain* yang paling banyak jumlah *unique hostnamenya*. Kombinasi tersebut diharapkan dapat membantu Administator dalam memantau dan mengamankan jaringan.

1.2 Rumusan Masalah

Merujuk dari latar belakang, maka dapat diambil rumusan masalah didalam penelitian ini, yaitu:

a. Bagaimana cara mendeteksi DNS *tunneling* dengan metode *traffic analysis* menggunakan Elasticsearch?
b. Bagaimana penggunaan Elasticsearch dapat membantu Administrator dalam meningkatkan keamanan jaringan?
c. Bagaimana pengaruh DNS *tunneling* pada kualitas jaringan?
d. Bagaimana mengatasi terjadinya DNS *tunneling*?

1.3 Batasan Masalah

Beberapa batasan masalah yang ditetapkan dalam penelitian ini adalah sebagai berikut:

a. *Log* DNS *query* hanya merupakan simulasi
b. Pendeteksian dilakukan dengan mengamati jumlah *unique hostname* sebagai *indicator of compromise*, semakin banyak jumlah *unique hostname* merupakan indikasi adanya DNS *tunneling*.
c. Penelitian ini tidak akan membongkar data apa yang dikirimkan melalui DNS *tunneling*.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini yaitu:

a. Melakukan simulasi untuk mendeteksi adanya DNS *tunneling*.
b. Membantu Administrator dalam meningkatkan keamanan jaringan.
c. Mengetahui pengaruh DNS *tunneling* terhadap performa jaringan.
d. Mengetahui cara mengatasi DNS *tunneling*.

1.5 Manfaat Penelitian

Manfaat dilakukannya penelitian ini adalah sebagai berikut:

a. Melengkapi penelitian-penelitian sebelumnya tentang bagaimana cara mendeteksi DNS *tunneling*.
b. Memberikan kemudahan bagi Administrator jaringan dalam mendeteksi sebuah DNS *tunneling*.

1.6 Metode Penelitian

Adapun langkah-langkah yang ditempuh untuk menyelesaikan penelitian ini adalah sebagai berikut:

a. Studi Literatur

   Studi literatur merupakan kegiatan untuk mempelajari literatur-literatur dan teori yang mendukung dalam mengerjakan penelitian ini. Studi literatur melalui paper, jurnal, artikel, buku, website yang terkait dengan DNS *tunneling, malware,* dan Elasticsearch.

b. Perancangan dan Topologi

   Merupakan tahapan dimana kami melakukan perancangan topologi, identifikasi kebutuhan perangkat lunak dan perangkat keras untuk melakukan simulasi DNS *tunneling*.

c. Simulasi dan Pengumpulan *Log*

   Pada tahap ini dilakukan implementasi dari perancangan untuk mensimulasikan DNS *tunneling*, dan menjalankan *packet sniffer* untuk mendapatkan *log* paket DNS.

d. Deteksi dan Analisis

   Kami akan melakukan analisis dari setiap *tools* DNS *tunneling* dan *malware* yang kami gunakan. Kami juga menganalisis kinerja dari Elasticsearch dalam mendeteksi adanya DNS *tunneling* dari beberapa *tools* dan *malware.*

e. Laporan

Merupakan tahapan pembuatan laporan terhadap hasil pengujian untuk memberikan gambaran menyeluruh mengenai topik penelitian ini, serta dapat memberikan rekomendasi yang bermanfaat untuk penelitian-penelitian selanjutnya.

1.7 Sistematika Penulisan

Laporan penelitian ini disusun dengan sistematika penulisan yang dapat mempermudah proses pembahasan penelitian. Adapun sistematika penulisan yang dimaksud adalah sebagai berikut:

BAB 1 PENDAHULUAN

Pendahuluan merupakan pengantar terhadap permasalahan yang akan dibahas. Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini memuat literatur review dan teori-teori penunjang yang digunakan sebagai dasar penelitian deteksi DNS tunneling dengan Elasticsearch.

BAB 3 METODE PENELITIAN

Bab ini menjelaskan bagaimana cara penelitian ini dilakukan, sehingga dapat memberikan rincian tentang alur atau langkah-langkah yang dibuat secara sistematis serta dapat digunakan dijadikan pedoman dengan jelas dalam menyelesaikan masalah, membuat analisis terhadap hasil penelitian, serta kesulitan yang dihadapi.

BAB 4 HASIL DAN PEMBAHASAN

Bab ini membahas tentang pengolahan dan analisis data untuk menghasilkan temuan serta analisis dari hasil temuan tersebut.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil penelitian serta saran dan rekomendasi untuk penelitian selanjutnya.