ANALISA KEAMANAN WEB SERVER TERHADAP SERANGAN POSSIBILITY SQL INJECTION Studi Kasus: Web Server UMK

Moh Dahlan[1], Anastasya Latubessy[2], Mukhamad Nurkamid[2]
[1]Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muria Kudus
Gondangmanis, PO Box 53, Bae, Kudus 59352
[2]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muria Kudus
Gondangmanis, PO Box 53, Bae, Kudus 59352
[*]E-mail: dahlan.kds@gmail.com

Abstrak

Keamanan merupakan salah satu faktor penting yang harus diperhatikan dalam membangun sebuah website. Hal tersebut menjadi sebuah tantangan tersendiri bagi para pengembang website, karena tidak ada jaminan yang pasti akan defenisi 'aman' itu sendiri. "tidak ada sistem yang benar-benar aman", bukanlah sebuah pernyataan semata, namun telah dirasakan dalam realitas. Website Universitas Muria Kudus yang berada di web server merupakan website yang digunakan sebagai media dan sarana informasi komunikasi kampus. Mengingat website ini dapat diakses secara luas, maka dinilai perlu memperhatikan keamanan website dalam berhubungan dengan lingkungan luar. Terdapat beberapa cara yang dapat digunakan untuk melakukan pengujian terhadap kemanan web server. Salah satunya adalah dengan melakukan SQL injection. SQL injection adalah kerentanan yang terjadi ketika penyerang memiliki kemampuan untuk mempengaruhi Structured Query Language (SQL) query yang melewati suatu aplikasi ke-database back-end. Penelitian ini menerapkan aturan possibility injection pada tools yang dipasang di Web Server Universitas Muria Kudus menggunakan intruder detection system (IDS) Snort sebagai identifikasinya terhadap serangan yang masuk. Hasil penelitian ini berupa alert sebagai alternatif peringatan keamanan dari serangan (intruder) yang masuk ke jaringan.

***Kata kunci**: keamanan web, SQL injection, Server web*

1. PENDAHULUAN

Perkembangan teknologi yang semakin canggih memungkinkan proses kegiatan berjalan sangat cepat. Hadirnya teknologi membuat pekerjaan semain mudah.Kemudahan yang ditawarkan teknologi tentunya seiring-seirama dengan bahaya yang dapat disisipkan melalui berbagai hal.Terlebih, jika bahaya tersebut tersistem sehingga kecenderungan pengguna tidak menyadarinya dengan adanya bahaya yang sudah masuk dan mengintainya.

Sistem dapat didefinisikan sebagai seperangkat komponen (sumber daya) terkait, dengan batas yang jelas dan bekerjasama untuk mencapai tujuan tertentu melalui sebuah inputan dalam proses transfromasi yang teroganisir (Brien dan Marakas, 2010).Sedangkan Sistem Informasi lebih menekankan pada pengelolaan sumber daya (resource) yang adamenjadi produk informasi.

Website UMK (Universitas Muria Kudus) dengan domain umk.ac.id merupakan *website* yang digunakan sebagai media dan sarana informasi kampus. Mengingat *website* ini dapat diakses secara luas, maka dinilai perlu memperhatikan keamanan *website*. Terdapat beberapa cara yang dapat digunakan untuk melakukan pengujian terhadap kemanan *website*. Salah satunya adalah dengan melakukan *SQL Injection.*

SQL injection adalah kerentanan yang terjadi ketika penyerang memiliki kemampuan untuk mempengaruhi *Structured Query Language (SQL) query* yang melewati suatu aplikasi ke *databaseback-end*. Dengan mampu mempengaruhi apa yang akan diteruskan ke *database*, penyerang dapat memanfaatkan sintaks dan kemampuan dari *SQL*, serta kekuatan dan fleksibilitas untuk mendukung fungsi operasi *database* dan fungsionalitas sistem yang tersedia ke *database*. Injeksi *SQL* bukan merupakan kerentanan yang eksklusif mempengaruhi aplikasi *Web*, kode yang menerima masukan dari sumber yang tidak dipercaya dan kemudian menggunakan *input* yang membentuk *SQL* dinamis bisa rentan (Clarke, 2009). Kasus *SQL Injection* terjadi ketika seorang penyerang dapat memasukkan serangkaian pernyataan *SQL* ke *query* dengan memanipulasi data input ke aplikasi (Anley, 2002).

Berdasarkan defenisi tersebut, dapat dikatakan bahwa serangan *SQLInjection* sangat berbahaya karena penyerang yang telah berhasil memasuki *database* sistem dapat melakukan manipulasi data yang ada pada *database* sistem. Proses manipulasi data yang tidak semestinya oleh penyerang dapat menimbulkan kerugian bagi pemilik *website* yang terinjeksi. Kebocoran data dan informasi merupakan hal yang fatal. Data-data tersebut dapat disalahgunakan oleh pihak yang tidak bertanggung jawab.

Keamanan data dan informasi sangat penting dalam menjaga ketahanan sebuah *website.* Berdasarkan uraian-uraian tersebut, maka dinilai perlu untuk menguji keamanan *website* UMK terhadap serangan *SQL Injection,* serta melakukan analisa terhadap kelemahan sistem yang ada, sehingga dapat diperoleh tindakan selanjutnya untuk perbaikan sistem.

2. METODOLOGI

Metodologi yang digunakan pada penelitian ini adalah menggunakan model forensik dan teknik pengumpulan data. Langkah-langkah yang digunakan dalam model forensik antara lain :

1. Identifikasi : yaitu tahapan dengan identifikasi terhadap kebutuhan baik kebutuhan fungsional maupun identifikasi kondisi jaringan Web Server UMK.
2. Pengujian : tahapan pengujian terhadap keamanan web server UMK. Peneliti mulai melakukan SQL Injection terhadap website UMK. Serangan disini hanya dilakukan untuk melihat apakah penyerang dapat memasuki databaseserver UMK tanpa melakukan manipulasi terhadap database yang ada, sehingga tidak akan mengganggu kondisi website yang sedang berjalan.
3. Analisa : tahapan dilakukannya analisa terhadap hasil serangan SQL injection, untuk menemukan kelemahan-kelemahan pada Web Server UMK. Berdasarkan hasil analisa, juga diharapkan dapat diperoleh solusi untuk pengembangan keamanan sistem.
4. Pelaporan : tahap pelaporan, mulai dilakukan dokumentasi terhadap hasil-hasil penelitian.

3. HASIL DAN PEMBAHASAN

Pada bab pembahasan ini terdiri dari tiga disain utama pembahasan, yaitu implementasi penelitian, disain analisa dan topologi jaringandan pembahasan proses forensik.

3.1. Implementasi Penelitian

Pada implementasi penelitian kali ini dilaksanakan oleh tim peneliti internal program studi Teknik Informatika, Fakultas Teknik Universitas Muria Kudus. Objek penelitian dikerjakan di UPT Perencanaan Sistem Informasi (UPT PSI) Universitas Muria Kudus dengan melibatkan tim teknis UPTPSI, dalam hal ini administrator jaringan(*network engineer).*

Kegiatan awal penelitian ini adalah dengan *diintegrasikannya intruder detection systems (IDS) tools (snort)* yang dipasang melalui komputer server PSI UMK pada tanggal 31 Maret 2015 s/d 10 April 2015 (gambar 1). Selanjutnya, hasil paket-paket data yang masuk melalui IDS tools (Snort) di evalusi dan di jelaskan kedalam laporan.

Dalam pelaksanaan kegiatan penelitian ini telah dilakukan identifikasi kebutuhan awal perangkat keras dan perangkat lunak sebagai berikut:

3.1.1. Kebutuhan perangkat keras (*hardware*)
- Server Web UMK

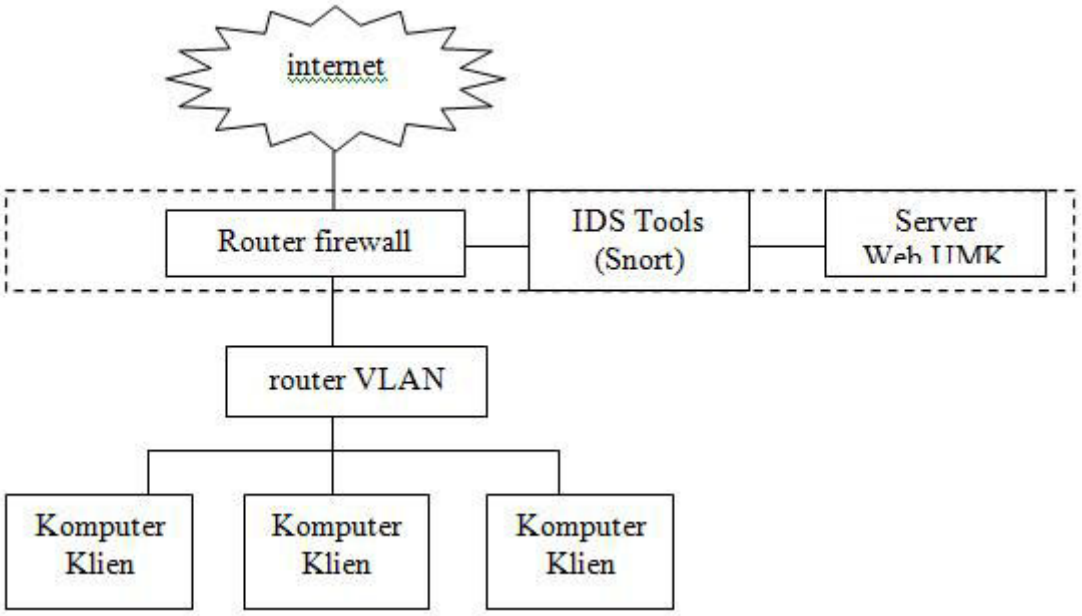
3.1.2. Kebutuhan perangkat lunak (*software*)
1) Opensource IDS (Snort)
2) Internet
3) Linux OS (*operating systems*)

Gambar 1. Komputer server yang telah terpasang *IDS (Snort)*

3.2. Disain Analisa dan Topologi Jaringan

Sebelum penelitian di implementasikan, perlu dibangun sebuah perencanaan yang nantinya membantu memudahkan jalannya proses pelaksanaan identifikasi pada penelitian. Kegiatan ini biasanya disebut sebagai analisa disain topologi (*topology design).* Pada analisa

ini terlibat komponen perangkat yang saling berhubungan, yaitu: jaringan internet, IDS Snort sebagai sistem pendeteksi data yang digunakan mengidentifikasi seluruh aktivitas penyadap (*intruder*) yang masuk ke Server Web UMK (gambar. 2).



Gambar 2. Topologi jaringan analisa *IDS Snort* (Sumber: UPT PSI-UMK, 2015)

Skema yang ditunjukkan pada gambar 2, paket data berawal masuk melalui komputer Client yang masuk dan ingin mengkases server UMK. Setelah itu paket data akan melewati router firewall yang berfungsi meneruskan (*forwarding*) informasi yang akan dituju.

Selanjutnya, melalui IDS Snort inilah sumber informasi akan di deteksi apakah sebagai paket data yang baik atau tidak, dalam istilah jaringan disebut sebagai *alert.*IDS (*intrusion detection systems*) pada prinsipnya sebagai sebuah perangkat yang didisain khusus untuk mengidentifikasi seluruh aktivitas (paket data) yang masuk di jaringan internet.Teknologi IDS ini telah dilengkapi beberapa komponen (software) yang handal, seperti *packet decoder, preprocessor or input plugins, detection engine, loging and alerting systems* dan *output modules* (tabel 1).

Tabel 1. Komponen-komponen IDS

No	Nama	Deskripsi
1.	*Packet decoder*	*Memproses paket-paket data yang masuk*
2.	*Preprocessor dan input plugins*	*Perangkat tambahan yang terpasang di IDS untuk mengenali data yang masuk diluar protokol yang ada*
3.	*Detection engine*	*Megaplikasikan rule ke paket data yang masuk*
4.	*Loging and alerting sytem*	*Pemrosesan pesan log dan tanda-tanda yang berbahaya (tidak dikenali) pada jaringan*
5.	*Output modules*	*Memproses paket data yang masuk sesuai dengan klasifikasi pesan (signature) dan menampilkan hasilnya*

3.3. Pembahasan Proses Forensik

Tahapan berikutnya setelah di lakukan analisa terhadap desian topologi jaringan adalah mengidentifikasi sesuai dengan tahapan proses-proses forensik. Pada tahapan proses forensic terdapat 4 aturan: (1) *collection*, (2) *examination*, (3) *analisys* dan (4) *reporting.*

1. *Collection*

   Mengumpukan seluruh data-data yang masuk untuk dilakukan identifikasi.Data-data yang masuk ditangkap melalui alat yang disebut dengan *intrusion detection system*s (IDS).

2. *Examination*

   Examination adalah pengujian paket-paket data yang telah di tangkap dari IDS sesuai dengan kriterianya klasifikasi pesan (*signatures*) yang ada.Paket data-data IDS biasanya terdiri dari beberapa klasifikasi pesan, seperti alert, logs, fals alarm, dan sensor.Berdasarkan klasifikasi inilah serangan dapat diidentifikasi apakah paket data tersebut berbahaya (merusak) atau tidak.

3. *Analisys*

   Pada tahapan analisa ini adalah menganalisa paket-paket data yang mencurigakan yang masuk ke server internal.Paket-paket data tersebut di analisa dan dilakukan pengujian. Pada tahapan ini di bagi menjadi dua kegiatan:

   a. Identifikasi serangan
   b. Langkah preventif (solusi)

4. *Reporting*

   Proses tahapan akhir dari penelitian yaitu penulisan laporan hasil.

3.4. Konfigurasi Snort

Langkah – langkah dalam melakukan konfigurasi Snort adalah sebagai berikut :

1. Melakukan setting variabel untuk jaringan yang akan di deteksi
2. Melakukan konfigurasi dynamic loaded libraries
3. Melakukan konfigurasi preprocessors
4. Melakukan konfigurasi outputplugins
5. Menambahkan runtime konfigurasi lainnya
6. Mengkostumisasi aturan/rule yang akan ditambahkan.

Variabel-variabel yang disetting pada jaringan yang dideteksi ditunjukan pada gambar 2, dimana terdapat delapan variabel yang diaktifkan pada konfigurasi snort.

```
1  var HOME_NET 192.168.1.0/24
2  var EXTERNAL_NET !$HOME_NET
3  var DNS_SERVERS $HOME_NET
4  var SMTP_SERVERS $HOME_NET
5  var HTTP_SERVERS $HOME_NET
6  var SQL_SERVERS $HOME_NET
7  var TELNET_SERVERS $HOME_NET
8  var SNMP_SERVERS $HOME_NET
```

Gambar 2. Seting variable pada jaringan

Beberapa port yang disetting pada jaringan yang dideteksi ditunjukan pada gambar 3, dimana terdapat beberapa tipe port yang diaktifkan. Baris pertama merupakan baris dimana *web server* jaringan dijalankan. Baris kedua digunakan untuk melihat *shellcode*. *Shellcode* adalah kode yang digunakan dengan *payload* untuk mengeksploitasi komputer target. Sedangkan baris ketiga, digunakan untuk melihat serangan *oracle*.

```
1  portvar HTTP_PORTS 80
2  portvar SHELLCODE_PORTS !80
3  portvar ORACLE_PORTS 1521
```

Gambar 3. Seting port jaringan

Variabel lainya yang disetting adalah variabel AIM Server. Beberapa list dari server yang dapat dimodifikasi ditunjukan pada gambar 4.

```
1  var AIM_SERVERS
2  [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/
3  24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/
4  24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]
```

Gambar 4. Setting AIM Server

Setting path variabel rule snort, path dapat disetting sesuai dengan *path* yang diinginkan. gambar 5, menunjukan **path** yang digunakan pada konfigurasi snort saat ini.

```
1  var RULE_PATH /etc/snort/rules
2  var PREPROC_RULE_PATH /etc/snort/prepro
```

Gambar 5. Seting rule snort pada jaringan

Gambar 6. mengaktifkan include *$RULE_PATH/local.rules*, ditunjukan pada baris ke 822 dengan menghilangkan tanda *hashtag*.

```
822  include $RULE_PATH/local.rules
823  #include $RULE_PATH/bad-traffic.rules
824  #include $RULE_PATH/exploit.rules
825  #include $RULE_PATH/community-exploit.rules
826  #include $RULE_PATH/scan.rules
827  #include $RULE_PATH/finger.rules
828  #include $RULE_PATH/ftp.rules
829  #include $RULE_PATH/telnet.rules
830  #include $RULE_PATH/rpc.rules
831  #include $RULE_PATH/rservices.rules
832  #include $RULE_PATH/dos.rules
833  #include $RULE_PATH/community-dos.rules
834  #include $RULE_PATH/ddos.rules
835  #include $RULE_PATH/dns.rules
836  #include $RULE_PATH/tftp.rules
```

Gambar 6. Seting Path local.rules

Gambar 7, merupakan tambahan *rule*atau aturan pada file *local.rules*. File *local.rule* digunakan untuk menambahkan aturan sendiri terhadap konfigurasi snort. Gambar 8 merupakan aturan untuk *possibility injection.*

```
1  alert icmp any any -> any any (msg:"ada orang yang lagi nyoba ngeping";sid:10000001;rev:0;)
2  alert tcp any any -> any 21 (msg:"FTP server lagi di akses";sid:10000002;rev:1;)
```

Gambar 7. Seting local rules untuk aktifitas ping

```
1  alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"<IN > WEB_SERVER Possible SQL Injection Attempt UNION
   SELECT";flow:established,to_server; content:"UNION"; nocase; http_uri;content:"SELECT"; nocase; http_uri;
   pcre:"/UNION.+SELECT/Ui";classtype:web-application-attack;reference:url,en.wikipedia.org/wiki/SQL_injection;reference:
2  url,doc.emergingthreats.net/2006446;reference:url,www.emergingthreats.net/cgibin/cvsweb.cgi/sigs/WEB_SERVER/WEB_SQL_Inj
   ction_Monster_List;sid:2006446; rev:11;)
```

Gambar 8. Seting Local rules untuk *SQL Injection*

3.5. Hasil Snort

Penelitian dimulai pada tanggal 12 april 2015 sampai dengan 18 april 2015. Berdasarkan hasil penelitian selama kurang lebih satu minggu tersebut, tidak terdapat aktifitas SQL *injection* yang mencoba melakukan injeksi terhadap web server UMK. Namun, mengingat peneliti menambahkan *rule* lainnya untuk mengetahui siapa saja yang mencoba melakukan **'ping'** ke jaringan, maka alert yang muncul pada snort seperti yang ditunjukan pada Gambar 9.

```
[**] [1:10000001:0] ada orang yang lagi nyoba ngeping [**]
[Priority: 0]
04/18-06:31:48.746265 192.168.1.10 -> 224.0.0.1
ICMP TTL:128 TOS:0x0 ID:44048 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:2704  ECHO

[**] [1:10000001:0] ada orang yang lagi nyoba ngeping [**]
[Priority: 0]
04/18-06:35:49.705011 192.168.1.4 -> 180.131.144.144
ICMP TTL:64 TOS:0xC0 ID:9307 IpLen:20 DgmLen:163
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
180.131.144.144:53 -> 192.168.1.4:57488
UDP TTL:61 TOS:0x0 ID:0 IpLen:20 DgmLen:135 DF
Len: 107  Csum: 61226
(107 more bytes of original packet)
** END OF DUMP
```

Gambar 9. Penggalan alert yang dihasilkan snort

4. KESIMPULAN

Berdasarkan pembahasan maka pada penelitian ini dapat disimpulkan bahwa :

(1) Pada Web Server Universitas Muria Kudus (UMK) terdapat beberapa aktifitas yang berusaha masuk ke sistem jaringan melalui notifikasi alert IDS Snort (ping), meskipun aktifitas tersebut hanya sekedar melihat-lihat web yang aktif, namun kegiatan ini perlu di waspadai.

(2) Dengan adanya IDS Snort, seluruh aktifitas jaringan yang berjalan di web server UMK dapat dipantau setiap saat.

(3) Pemberian aturan snort/ rule yang sebagai alternatif yang dapat memberikan peringatan dari serangan (intruder) yang masuk ke jaringan.

(4) Secara umum dengan menggunakan IDS Snort sebagai pemantau jaringan dapat disimpulkan bahwa web server UMK dapat dikatakan relatif aman.

UCAPAN TERIMA KASIH

Pada penelitian ini kami tidak lupa mengucapkan terima kasih yang sebesar-besarnya kepada UPT PSI Universitas Muria Kudus yang telah memberikan ijin melakukan penelitian.

DAFTAR PUSTAKA

Anley, C., 2002, Advanced SQL Injection in SQL Server Applications. An NGS Software Insight Security Research (NISR) Publications: Next Generation Security Software Ltd.

Clarke, J., 2009, SQL Injection Attacks and Defense. Burlington: Syngress Publishing and Elseiver.

Halfond, W.G.J., Orso, A., 2005., AMNESIA: Analysis and Monitoring for Neutralizing SQL Injection Attacks. IEEE and ACM Intern. Conf. On Automated Software Engineering (ASE 2005). Hal. 174–183, Nov. 2005.

Pomeroy, A., Tan, Q., 2011, Effective SQL Injection Attack Reconstruction Using Network Recording. IEEE International Conference on Computer and Information Technology.Canada.

Clarke, J. 2012. SQL Injection Attack and Defense Second Edition, Elsevier, Inc 225 Wyman Street, Waltham, MA 02451, USA

O' Brien J., A., Marakas, G,. M. 2010. Introduction to Information Systems Fifteenth Edition, McGraw-Hill Companies Inc, New York.

Rehman, R.U. 2003.Intrusion Detection Systems with Snort: Advanced Technique Using Snort, Apache, My-SQL, PHP and ACID. Prentice Hall, Inc, USA. Available link URL available : http://ptgmedia.pearsoncmg.com/images/0131407333/downloads/0131407333.pdf

Dahlan, M., Latubessy, A, Nurkamid, M. 2013. Pengujian dan Analisa Keamanan Website terhadap Serangan SQL Injection (Studi Kasus: Website UMK), Laporan Penelitian, Program Studi Tek.Informatika, Fakultas Teknik, Universitas Muria Kudus, Kudus. URL available