Identification

CVE tracking number: CAN-2003-0352
Vulnerability Name: Microsoft W32/Blaster Worm. Key references include: CERT CA-2003-20 http://www.cert.org/advisories/CA-2003-20.html and Microsoft Security Bulletin MS03-026 http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-026.asp

Overview

W32/Blaster worm exploits known vulnerabilities in the Microsoft Remote Procedure Call (RPC) Interface. A remote attacker could exploit these vulnerabilities to execute arbitrary code with Local System privileges or to cause a denial-of-service condition. It impacts Microsoft Windows NT 4.0, 2000, XP and Server 2003 systems.

Description

The W32/Blaster worm exploits a vulnerability in Microsoft's DCOM RPC interface as described in VU#568148 and CA-2003-16. Upon successful execution, the worm attempts to retrieve a copy of the file msblast.exe from the compromising host. Once this file is retrieved, the compromised system then runs it and begins scanning for other vulnerable systems to compromise in the same manner. In the course of propagation, a TCP session to port 135 is used to execute the attack. However, access to TCP ports 139 and 445 may also provide attack vectors and should be considered when applying mitigation strategies. Microsoft has published information about this vulnerability in Microsoft Security Bulletin MS03-026.

The worm includes the ability to launch a TCP SYN flood denial-of-service attack against windowsupdate.com.

Sites that do not use windowsupdate.com to manage patches may wish to block outbound traffic to windowsupdate.com. In practice, this may be difficult to achieve, since windowsupdate.com may not resolve to the same address every time.

Impact

Although the OmniSwitch is not a target of the worm, OmniSwitch configurations can provide some assistance in minimizing the impact for those networks that have not patched all applicable Microsoft systems.

OS-7700/7800/8800 has Hardware Route Cache (HRE) on each NI. Whenever a request comes in from an end-station for an unknown destination, it is handled in software. Once the destination is learned by software a route cache entry is created and then the traffic is routed in hardware.

This DoS attack starts scanning for other vulnerable systems, so each infected system will start generating hundreds of random destination addresses resulting in high CPU utilization of the NI processor. This can be observed by using the command “show health <slot>”, or “show health all cpu”. To see the number of entries currently learned in the HRE use the command "show hre pcam utilization slot/0" where slot is the NI in question and 0 is the slice (7800’s only have one slice per NI where 8800 can have up to 4). When a DoS attack is occurring this command will show that the HRE utilization is increasing.

Solution of AOS-based OmniSwitch Products

ACLs can be configured on the OmniSwitch 6600, 7700, 7800 and 8800 to stop the propagation of the virus (worm) to other workstations in the network. Network administrators must note that implementing the recommended measures to protect the network from this attack blocks services using the affected ports.

The following is an example of how to deny all the routed traffic to the TCP and UDP ports advised by CERT. These policies use a combination of destination network address, which is for the internal network, and the destination TCP/UDP ports to drop traffic so that other systems are not infected.
Please note that blocking ports 137 and 138 will affect Microsoft Windows Shares including WINS services, etc.

```
Policy network group internal_network  169.10.64.0 mask 255.255.240.0
169.10.32.0 mask 255.255.224.0 169.10.80.0 mask 255.255.240.0 169.10.176.0
mask 255.255.240.0 169.10.160.0 mask 255.255.240.0

policy service t135 destination tcp port 135
policy service t137 destination tcp port 137
policy service t138 destination tcp port 138
policy service t139 destination tcp port 139
policy service t69 destination tcp port 69
policy service t445 destination tcp port 445
policy service t4444 destination tcp port 4444

policy service u135 destination udp port 135
policy service u137 destination udp port 137
policy service u138 destination udp port 138
policy service u139 destination udp port 139

policy service group tcp_udp_group t135 t137 t138 t139 t69  t445  t4444
u135 u137 u138 u139

policy condition c1 service group tcp_udp_group destination network group
internal_network
policy action deny disposition deny
policy rule r1 condition c1 action deny

qos apply
```

Deleting a Network from ACLs
The following command will delete the network 223ip from the network group internal_network to allow access to all of the ports in that subnet.

```
Policy network group internal_network no 169.10.64.0 mask 255.255.240.0
Qos apply
```

Adding a Network in ACLs
The following command will add the network 223ip into the network group internal_network to block access to all of the ports in that subnet.

Policy network group internal_network 169.10.64.0 mask 255.255.240.0
Qos apply

Note that the feature “qos classifyl3 bridged” can be used to match layer 3 IP headers on bridged traffic if desired. It is recommended that this only be used if traffic is bridged and not routed.

The following are additional enhancements made in code build 5.1.4.113r03 and 5.1.5.398r01 and above. It is recommended to use the latest 5.1.4r03 or 5.1.5r01 or above when using the features below. Contact customer support if there are any questions. The features below only apply to the 7700/7800 and the 8800.

Early drop function for tcp/udp ports which helps prevent PCAM entry exhaustion when subjected to attack which targets a specific TCP port.

This feature is enabled on a per slice/NI (or vlan) basis. All of the traffic that matches the tcp/udp port will be dropped.

The keyword word "DropServices" must be used for the feature above to work.
To do this make a policy that specifies where to do the "DropServices". QOS will accept two kinds of policies using the DropServices group:
policy condition c1 source port group <portgroup> service group DropServices
policy condition c2 source vlan <vlan> service group DropServices

This allows two conditions that apply to Drop Services one based on source physical ports and a second based on the source VLAN. To apply Drop services to a particular set of port groups use a source port group condition. To exempt certain VLANs on those physical ports then assign exception VLANs using the condition source VLAN. For example:
> policy rule r1 condition c1 action drop
> policy rule r2 condition c2 action accept

The restrictions are:
When using a port or port group with the DropServices, only a drop action can be used.
When using a source VLAN with the DropServices, only an allow action can be used.
The accept rule takes precedence over the drop rule. The DropServices rules are essentially the highest precedence rule a magic service group is present. Other types of qos policies can not be implemented to allow packets that already match specified DropServices.

Here is an example of implementation on a per slice basis, as seen below it does not require a rule to be in effect.

policy port group UserPorts 1/1-12 2/1-12 3/1-12 4/1-12 5/1-12 11/1-2 12/1-2 13/1-2 (these are the ingress NI's where you want the traffic dropped)

policy service t135 destination tcp port 135
policy service t139 destination tcp port 139
policy service t445 destination tcp port 445
policy service t1025 destination tcp port 1025
policy service t2745 destination tcp port 2745
policy service t3127 destination tcp port 3127
policy service t5000 destination tcp port 5000
policy service t6129 destination tcp port 6129
policy service u135 destination udp port 135

policy service group DropServices t135 t139 t445 t1025 t2745 t3127 t5000 t6129 u135

Preventing users from spoofing addresses that are not on the local network.

Users could be identified by the source MAC address, but this address is easily spoofed. The one thing that users cannot spoof and is easily identifiable is the port on which the user is entering the switch/router. User ports should never legitimately have a source IP address that is not in the defined subnet for that port as opposed to router ports, which could. User ports are defined in a QOS port group rule using the keyword "UserPorts"

Example:
policy port group UserPorts <slot>/<port-port>
For example:
policy port group UserPorts 2/5 3/1-10
Defines ports 2/5 and ports 1 thru 10 on slot 3 as user ports.

"UserPorts" is a magic port group. The "UserPorts" groups works like any other port group as far as the CLI is concerned. DHCP packets are never checked for spoofing since the will typically have the source IP set to 0.0.0.0.
There is no command for turning on spoofing prevention, all ports that are identified as "UserPorts" will automatically do spoofing prevention.
If QOS is not enabled on the switch, no spoofing prevention will be done.
A count is kept of packets dropped because of spoofing as seen at the bottom of the "show ip traffic" output. This count does not indicate all packets dropped, but will give an idea of what addresses are spoofing.

Preventing User pings and reducing DOS exposure from Pings.

In order to reduce Ping traffic on a network without excessively impacting NI traffic a ping drop QOS command is defined. The effect of this command is that an NI will drop all ICMP echo request and ICMP echo reply packets.
The example below will drop all ICMP echo request/reply on that VLAN.

Example configuration:
policy condition ping10 source vlan 10 ip protocol 1
policy action drop disposition drop
policy rule noping10 condition ping10 action drop

Combined Example

Here is an example of the above rules combined for the 7700/7800 and 8800.
This would accomplish the following items -

1. Drop icmp for vlan's 2,3 and 4
2. Early drop for TCP ports 69,445, and 4444 - the IP flows would not be learned.
3. Drop TCP/UDP ports 135,137,138 and 139 - the IP flows will be learned as an HRE PCAM entry.
4. Antispoofing on the specified UserPorts
5. Allows communication to server IP's 169.10.64.10, and 169.10.64.11 for TCP/UDP ports 135, 137, 138, and 139, but drops traffic to these ports when no communicating with these IP's.
6. Allows all traffic on vlan 2 except icmp.

```
policy service t135 destination tcp port 135
policy service t137 destination tcp port 137
policy service t138 destination tcp port 138
policy service t139 destination tcp port 139
policy service t4444 destination tcp port 4444
policy service t445 destination tcp port 445
policy service t69 destination tcp port 69
policy service u135 destination udp port 135
policy service u137 destination udp port 137
policy service u138 destination udp port 138
policy service u139 destination udp port 139
policy service group DropServices t4444 t445 t69
policy service group tcp_udp_group t135 t137 t138 t139 u135
policy service group tcp_udp_group u137 u138 u139
policy network group internal_network 169.10.64.0 mask 255.255.240.0
169.10.32.0 mask 255.255.224.0 169.10.80.0 mask 255.255.240.0 169.10.176.0
mask 255.255.240.0 169.10.160.0 mask 255.255.240.0
policy network group servers 169.10.64.10 169.10.64.11
policy port group UserPorts 1/1-12 2/1-12 3/1-12 4/1-12 5/1-12
policy port group UserPorts 11/1-2 12/1-2 13/1-2
policy condition allow_vlan source vlan 2 service group DropServices
policy condition c1 destination network group internal_network service
group tcp_udp_group
policy condition icmp2 source vlan 2 ip protocol 1
policy condition icmp3 source vlan 3 ip protocol 1
policy condition icmp4 source vlan 4 ip protocol 1
policy condition servers source network group servers
policy condition users_to_servers destination network group servers service
group tcp_udp_group
policy action allow
policy action drop disposition drop
policy rule servers precedence 200 condition servers action allow
policy rule users_to_servers precedence 200 condition users_to_servers
action allow
policy rule allow_vlan precedence 100 condition allow_vlan action allow
policy rule r1 condition c1 action drop
policy rule vlan2 condition icmp2 action drop
policy rule vlan3 condition icmp3 action drop
policy rule vlan4 condition icmp4 action drop
qos apply
```

Tracking the Source

In order to track the most chatty device generating all the traffic in the network, the following command can be used"

debug ip-packet [start] [timeout seconds] [stop] [direction {in | out | all}] [format {header | text | all}] [output {screen | switchlog}] [board {cmm | ni [1-16] | all | none} [ether-type {arp | ip | hex [hex] | all}] [ip-address ip_address] [ip-pair [ip1] [ip2]] [protocol {tcp | udp | icmp | igmp | num [integer] | all}] [show-broadcast {on | off}] show-multicast {on | off}]

There are several options available which helps to classify the kind of traffic one may be interested in.

```
start              Starts an IP packet debug session.
timeout            Sets the duration of the debug session, in seconds. To
                   specify a duration for the
debug              session, enter timeout, then enter the session length.
seconds            The debug session length, in seconds.
stop               Stops IP packet debug session.
direction          Specifies the type of the packets you want to debug:
in -               Debugs incoming packets
out -              Debugs outgoing packets.
all -              Debugs both incoming and outgoing packets.
format             Specifies the area of the packet you want to debug:
header -           Debugs the packet header.
text -             Debugs the packet text.
all -              Debugs the entire packet.
output             Specifies where you want the debug information to go:
screen -           Output will appear on screen.
switchlog -        Output will be saved to a log file.
board              Specifies the slot (board) that you want to debug:
cmm -              Debugs CMM packets.
ni -               Debugs packets for an Network Interface (NI). To debug a
                   specific inter-face, enter ni, then enter the slot number
                   of the NI.
all -              Debugs packets for all CMMs and NIs on the switch
none -             Clears the previous "board" settings.
```

The output is available on the console or telnet of the switch. This can help to identify the most chatty device which can be taken off the network to rectify the situation. Please note that if used over telnet with the default of showing all traffic in software, that telnet traffic will also be shown in the debug which may not be desirable.

Example –

-> debug ip packet start timeout 1
C R 5/10 00d0957c6188->00d0957c6189 IP 169.12.1.254->169.12.1.3 TCP 28198,100, S, s 28988, a 0, w 16384
C R 5/10 00d0957c6188->00d0957c6189 IP 169.12.1.254->169.12.1.3 TCP 28198,100, S, s 28988, a 0, w 16384
C R 5/10 00d0957c6188->00d0957c6189 IP 169.12.1.254->169.12.1.3 TCP 28198,100, S, s 28988, a 0, w 16384

-> debug ip packet ip-address 169.12.1.100 start timeout 1
-> 5 R 5/2 (00003939600e)->(00d0958eccf4) IP 169.12.1.100->169.12.1.10 ICMP 8,0 seq=26367.
5 S 5/2 00d0958eccf4->00003939600e IP 169.12.1.10->169.12.1.100 ICMP 0,0 seq=26367.
5 R 5/2 (00003939600e)->(ffffffffffff) ARP Request 169.12.1.100->169.12.1.2
5 R 5/2 (00003939600e)->(ffffffffffff) ARP Request 169.12.1.100->169.12.1.3

Solution for XOS-based Omni Switches

Most XOS-based switches cannot filter on TCP or UDP port numbers. Filtering can only be done for routed traffic on the OmniAccess 512 or OS/R. OS/R's need to run 4.1 code or above with an HRE-X installed on the MPX or on the NIs in a distributed manner to filter destination TCP ports. The OS/R can not filter traffic based on the UDP ports. The OmniSwitch can not do filtering.

From a LAN switch's perspective, the "Sasser" worm, is a denial of service attack. The worm generates enough traffic to your site that it denies service to the site's legitimate users. See http://www.cert.org/tech_tips/denial_of_service.html for more info. A DOS attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include:

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person

It has also been determined that the XOS OmniSwitch and OmniSwitch/Router products are not affected by IP DOS attacks when configured as a layer 2 device with routing disabled (one IP address for Mgmt interface is allowable) on all configured vlans and 802.1q tagging enabled. This configuration has been implemented in multiple locations with no performance degradation to the switch.

How to Identify a DOS Attack from an OmniSwitch

This is best accomplished by capturing statistics from an operational switch that has an MPM-III and HRE-VX. To do this, from the console, run hdstat, systat, taskstat, prn_rc_dbg, and if_vbDebug=33;taskDelay 600;if_vbDebug=0. Consider increasing the taskDelay to 900 or 1200 or take it out entirely. Make sure you turn it off when you are done or reboot the switch.

One resolution to a DOS attack is to patch the offending operating system (Microsoft in this case). A second would be to remove the offending workstation from the network until patched.

An Alcatel customer stated: "We had identified two possible IP addresses which could be infected with the virus. There may be other possible IP addresses that could be infected. We analyzed the captured file and deduced the following:

```
202.160.24.16 (OK)
202.160.24.33 (OK)
202.160.24.156 (X)
202.160.24.164 (OK)
202.160.24.178 (OK)
202.160.26.7 (X)
202.160.28.4 (OK)
202.160.28.7 (OK)
202.160.28.246 (--- Cannot test cos' modem is not power up)
202.160.29.163 (OK)
202.160.29.180 (OK)
```

"But we had tested and deduced that only the above two IP addresses are infected PCs. They are 202.160.24.156 and 202.160.26.7. They are currently disabled.

"We are simultaneously monitoring the hreRouteCacheCount and CPU utilization while these IP addresses are enabled and disabled. When we enabled one of the above infected IP address, both reRouteCacheCount and CPU utilization shot up, and when we disabled it, both count and utilization revert to normal condition"

hdstat Results

```
Device 1 Min 1 Hr 1 Hr
Resources Limit Curr Avg Avg Max
---------------- ----- ---- ----- ---- ----
Receive 80 01 01 01 01
Transmit/Receive 80 01 01 01 01
```

```
 Backplane 80 01 01 01 01
 CAM [MPM] 80 100* 131 131 131
 CAM [HRE] 80 100* 99 100 100
 Collisions [HRE] 80 04 04 02 08
 CPU 80 53 55 65 100
 Memory 80 26 26 26 26
 Temperature 62 31 31 30 31
 Virtual Ports 80 21 21 21 21
```

systat Results

```
 System Uptime : 0 days, 12:56:24.37
 MPM Transmit Overruns : 0
 MPM Receive Overruns : 0
 Excessive Ping Requests : 0 in the last 0 days, 01:34:53
 MPM total memory : 64MB
 MPM free memory : 46965176 bytes
 MPM CPU Utilization ( 5 sec) : 56% ( 0% intr 2% kernel 53% task 44% idle)
 MPM CPU Utilization ( 60 sec) : 55% ( 0% intr 1% kernel 51% task 45% idle)
 Power Supply 1 State : OK
 Power Supply 2 State : Bad
 Temperature Sensor : OK - Under Threshold

 Temperature: 31.00c 87.80f
 Temperature Alarm Masking : Disabled
```

Dshell Results

```
 -> prn_rc_dbg
 The current router cache timer is set to 30 seconds.
 The current HREX router cache timer is set to 30 seconds.
 157 MPM IP route cache entries aged out in the last 30 seconds
 0 MPM IPX route cache entries aged out in the last 30 seconds
 7457 HREX IP route cache entries aged out in the last 30 seconds
 0 HREX IPX route cache entries aged out in the last 30 seconds
 All IP cache entries cleared 3 times. (Last cleared by tRip, 3453 seconds
ago
 IP network cache entries cleared 25 times. (Network 202.160.12.0
(255.255.255.128) cleared by tif_vbPMH, 19220 seconds ago
 Individual IP cache entries have been cleared 1492 times.
 (Last entry cleared by tNetTask, 25 seconds ago
 All IPX cache entries cleared 0 times.
 IPX network cache entries cleared 0 times.
 Individual IPX cache entries have been cleared 0 times.
 There are 3968 MPM route cache entries.
 There are 40960 HREX route cache entries.
 value = 0 = 0x

 -> hreRouteCacheCount
 _hreRouteCacheCount = 0x308586f0: value = 40960 = 0xa000
 -> exit
 HRE-8205 / %
```

Additional dshell Results

```
-> if_vbDebug=33;taskDelay 300;if_vbDebug=0.

-> S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.168.132.121 TCP 3992,80
```

```
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.168.202.38 TCP 3993,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.168.26.80 TCP 3994,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.28.170.94 TCP 3995,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.215.146.153 TCP 3996,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.44.1.156 TCP 3997,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->133.1.96.231 TCP 3278,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->202.94.179.3 TCP 3279,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.39->24.70.145.252 TCP 2110,6346
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->202.160.47.233 TCP 3280,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.251.132.178 TCP 4089,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.7->28.40.55.199 TCP 4125,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->212.112.103.219 TCP 4729,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->151.167.221.95 TCP 4049,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.7->134.237.193.109 TCP 4110,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->212.112.121.231 TCP 4736,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->32.208.166.207 TCP 4141,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->212.163.187.253 TCP 4737,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.7->22.240.204.174 TCP 4113,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->147.37.249.248 TCP 4738,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->202.160.228.66 TCP 3387,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->146.75.19.73 TCP 4739,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.156->213.106.169.156 UDP 63212,1055
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.180->93.77.58.72 TCP 4816,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.156->24.67.13.160 UDP 63127,4721
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.77->213.89.36.241 TCP 1065,1214
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.156->129.22.169.93 UDP 63143,2074
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.28.4->192.168.195.224 TCP 4427,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->212.144.223.48 TCP 1245,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.180->202.160.63.171 TCP 4446,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->202.248.38.196 TCP 4288,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->202.89.90.14 TCP 4290,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.29.163->212.252.173.88 TCP 1668,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.26.7->222.52.49.250 TCP 4357,80
S:00d095:4e4ec3->00000c:07ac01 IP 202.160.24.33->29.30.27.100 TCP 4872,80
```

If you have any questions, the following contact information should be used:

Web Links
Customers: http://eservice.ind.alcatel.com/
EMEAI Business Partners: http://www.businesspartner.alcatel.com/

Email
support@ind.alcatel.com
EMEAI Business Partners: mailto:support.center@alcatel.fr)

Phone
North America 1 800-995-2696
Latin America 1 877-919-9526
Europe (EMEAI) +33-388-55-69-04
Asia Pacific +65-394-7933
Other International +1-818-878-4507

Alcatel customers under maintenance contracts have access to troubleshooting tools on the Intranet found at the web addresses listed above.