Bill Young

Intrusion Detection in Depth

GCIA Practical Assignment, v3.3
SANS San Francisco, Dec 2002

06/20/2003

Table of Contents:

Summary:

This paper is a submission under the GIAC Certified Intrusion analyst (GCIA) Practical Assignment version 3.3. It covers several areas of intrusion detection broken into 3 parts. The first section deals with a specific area of intrusion detection, host-based intrusion prevention. This area will cover where host-based intrusion prevention plays within the schema of layered security. The second section will examine 3 network detects. The first two detects were taken from the directory at www.incidents.org/logs/raw. The third detect occurred in the wild on my home network. The last section examines 5 days worth of logs at an unspecified University. Snort Alert logs, Scan logs, and Out of Spec logs were examined for key risks to the University during that time frame.

Section I: A Place for Host-Based Intrusion Prevention in Layered Security

> "The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable"
>
> Sun Tzu, (Chapter VIII. Variation in Tactics)

Background:

At this juncture within the evolution of Intrusion Detection Systems there is much debate on the process and even worth of IDS systems. The efforts to handle intrusions can take many directions. Some take the approach of monitoring for intrusions. Others actively attempt to prevent anomalous traffic.

Tools that monitor traffic and give detailed analysis are generally called IDS systems. Systems that are usually inline with the network flow and attempt to monitor and actively deny traffic are called Intrusion Prevention Systems.

And since nothing is black and white, we see middle of the road efforts that combine both methods. Some split the difference by monitoring and then sending resets to offenders. Others interface with perimeter security devices such as routers and firewalls to "shun" or defeat the intrusions.

In this paper, I make the distinction between network based IDS, Intrusion Prevention systems, and host based systems. Network based systems listen, monitor, and sometimes prune network traffic. Intrusion prevention systems are focused on stopping the bad traffic once detected. Host based systems wait until the traffic actually reaches a host before monitoring,

There are many sectors stridently voicing their opinion. One major analyst firm, Gartner Group recently weighed in. Gartner is questioning the value of Intrusion Detection Systems at all and is in favor of Intrusion Prevention Systems.

Of course the world being what it is, Newton Third Law says, "For every action there is an equal and opposite reaction." and there are many that oppose that view. This is a rich area of discussion and worth much time, analysis and testing. But for the purpose of this paper I would like to fall back to the earlier distinction between network versus host based systems. We will be focusing on host based intrusion prevention and hopefully see some arguments for its use.

Premise:

The premise of this paper is that for some systems, Host Based Intrusion Prevention is a viable option and good alternative to having no protection on a system. Other than examining the security position from the network side, we will not be looking at Network Intrusion Prevention.

Discussion:

Intrusion Detection Systems have many purposes. They enable us to pursue network offenders by providing detailed records of how they have intruded on our networks. They alert us to external or internal attacks on our network. But the most common business justification for deploying an IDS system is Risk Management. Firewalls, IDS Systems, Log analysis, and prevention systems, are all utilized to minimize risk of exposure to attack and in some cases provide the means where we can fight back with our own attack in the legal arena.
.
That potential risks exist depends on what needs to be secured. A weblog operator who posts political commentary on a personal web server in his or her spare time has very little exposure, other than embarrassment, if the system is compromised. A major manufacturing corporation with trade secrets, client data, and medical record information on employees has much more to lose and can be held accountable for compromised data.

So given that the internal systems are worth protecting because of some dollar or intrinsic value, why deploy any security controls. Or better, what does deploying a firewall, a network based IDS, a host based IDS, give to the organization.

For the purpose of discussion let us explore layered network security and dangers of relying on a single firewall to protect a system.

The main idea of layered security is to have backup defenses in case the controls at the forefront are overrun or bypassed. There is also the idea of specificity within this “defense in depth” strategy. For example, having a content server watching HTTP or SMTP traffic for viruses is a specific control for a specific avenue of attack.

Please bear with as I draw parallels between a real world example and our network defense in depth strategy.

In a battle, fighter aircraft control the air, patrol around a carrier battlegroup, and are supported by guided missile cruisers. Destroyers and Frigates would be working the fringes protecting against enemy submarines using a different vector to attack the group.

Ok so how does this apply to the problem at hand of the proposed importance of host based intrusion protection? Carrying the military metaphor further, if a missile is constructed with Stealth technology or an aircraft has codes that fool the aircraft fighters and the missile cruisers, it can get much closer to the battle group heart, the carrier. On each carrier they have an automated system called a CIWS or close in weapon system, that will knock the missile out of the sky if the other systems fail to protect.

This is where the idea of Host Based Intrusion Prevention comes to play. If for example an attacker can bypass the firewall by encrypting the nasty payload so that the firewall or IDS can not see it, the attacker has a better chance of touching the host. Intrusion Prevention can be that last layer of protection that prevents compromise.

No security system is foolproof. What’s more we introduce bypasses in to our security, for the sake of doing business. For example, conduits are placed through a firewall so that customers and partners can access key web and database servers. Attackers are wily and will shift their efforts to the areas of least resistance.

My hypothesis is at that for some organizations and applications it would be optimal to have some sort of system, logging and monitoring for the attacks and best case protecting against well known vulnerabilities at the host. This where Host Based Intrusion Prevention systems might add some benefit.

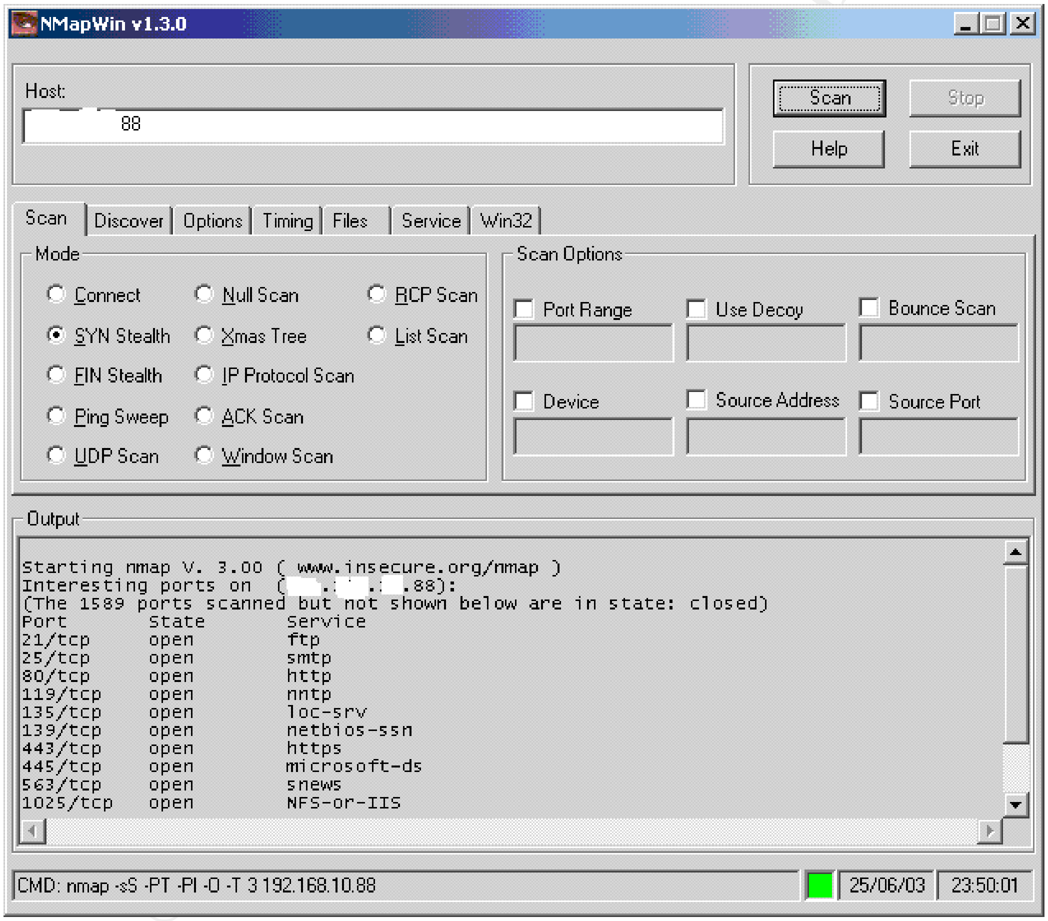
To test out some of my assumptions I used a set of tools that scan and can simulate attacks. These tools consisted of:

- SuperScan – a port scanning tool created by Foundstone Inc.
- NMAP – a free scanning tool created by Fyodor fyodor@insecure.org (Fyodor,Art of Port Scanning)

- Divine Intervention v3– a suite of tools used to create packet flooding, bad packets, and other tools by George D Konidaris. (Divine Intervention III, p 4-6)
- Some practices from the whitepaper, “Testing Entercept Live!” and that can enable buffer overflows on improperly patched IIS systems. (Testing Entercept Live!)

To begin with let’s look at an unpatched system and have some fun with it.

Our test receiving system is a Windows 2000 server running service pack 3. We can port scan the system easily. Below are some results from SuperScan and NMAP.



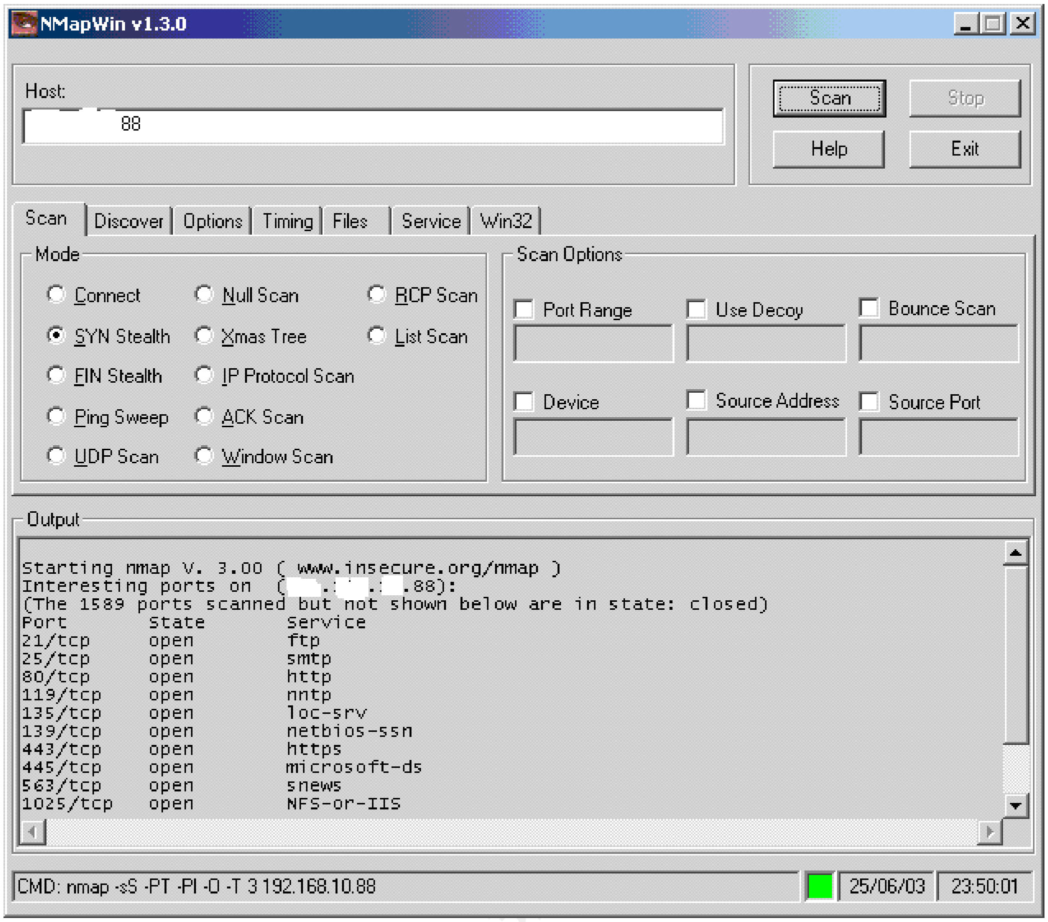
Host: [redacted]88

Mode:
○ Connect ○ Null Scan ○ RCP Scan
● SYN Stealth ○ Xmas Tree ○ List Scan
○ FIN Stealth ○ IP Protocol Scan
○ Ping Sweep ○ ACK Scan
○ UDP Scan ○ Window Scan

Scan Options: □ Port Range □ Use Decoy □ Bounce Scan □ Device □ Source Address □ Source Port

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  ([redacted].[redacted].[redacted].88):
(The 1589 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
25/tcp     open        smtp
80/tcp     open        http
119/tcp    open        nntp
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds
563/tcp    open        snews
1025/tcp   open        NFS-or-IIS
```

CMD: nmap -sS -PT -PI -O -T 3 192.168.10.88 🟩 25/06/03 23:50:01


NMapWin v1.3.0
Scan
Stop
Help
Exit
Scan | Discover | Options | Timing | Files | Service | Win32
Mode
○ Connect ○ Null Scan ○ RCP Scan
○ FIN Stealth ○ IP Protocol Scan
○ Ping Sweep ○ ACK Scan
○ UDP Scan ○ Window Scan
○ Xmas Tree ○ List Scan
Scan Options
□ Port Range □ Use Decoy □ Bounce Scan
□ Device □ Source Address □ Source Port
Output

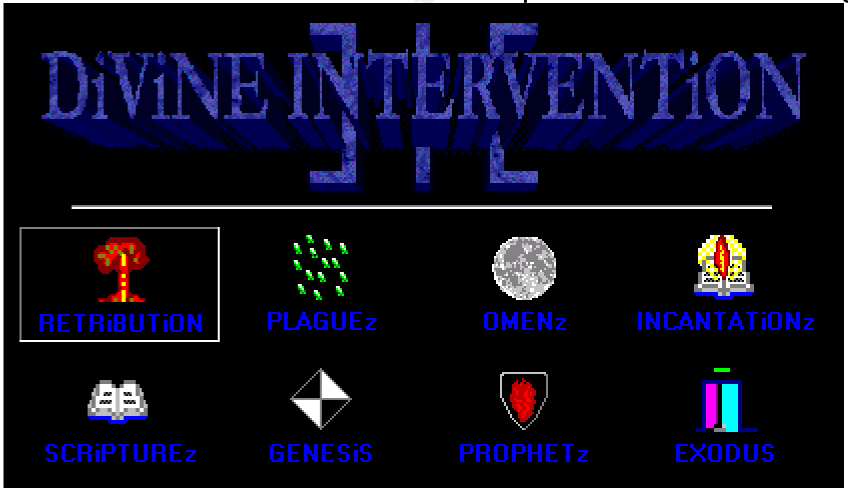
Host: [redacted] 88

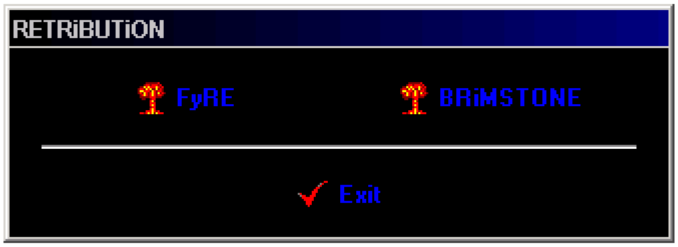
● SYN Stealth

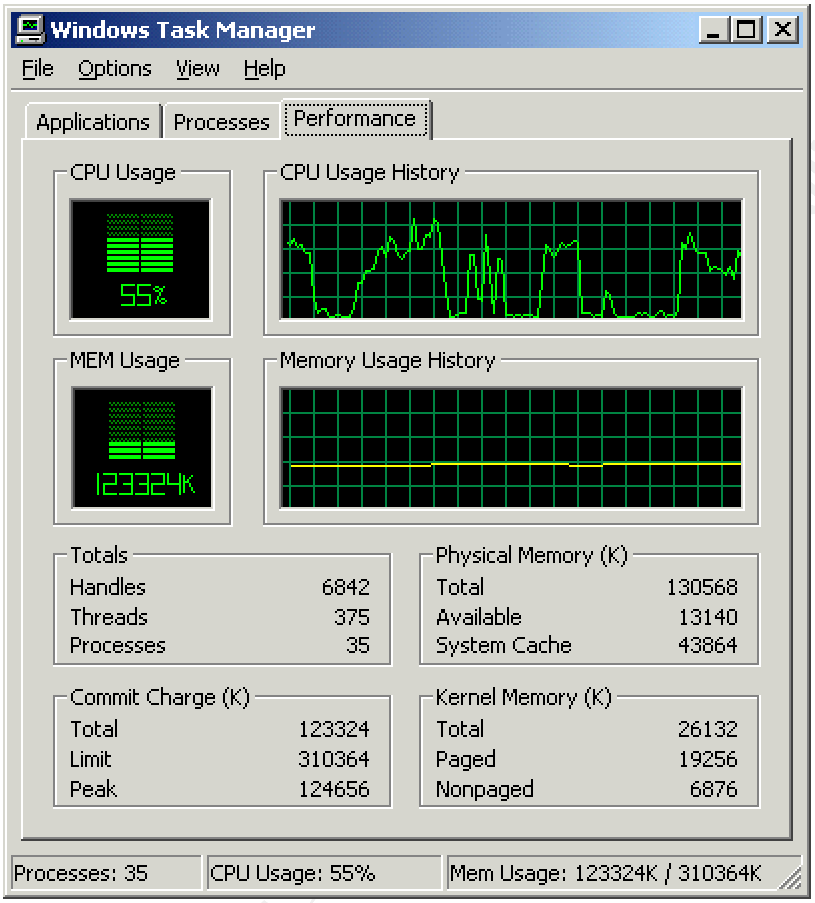
```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  ([redacted].[redacted].[redacted].88):
(The 1589 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
25/tcp     open        smtp
80/tcp     open        http
119/tcp    open        nntp
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds
563/tcp    open        snews
1025/tcp   open        NFS-or-IIS
```

CMD: nmap -sS -PT -PI -O -T 3 192.168.10.88 | 🟩 | 25/06/03 | 23:50:01

To be more vindictive we can employ the FLOODZ applet from Divine Intervention and send a multitude of packets at the receiving system.







```
Totals                          Physical Memory (K)
Handles           6842          Total            130568
Threads            375          Available         13140
Processes           35          System Cache      43864

Commit Charge (K)               Kernel Memory (K)
Total           123324          Total             26132
Limit           310364          Paged             19256
Peak            124656          Nonpaged           6876

Processes: 35   CPU Usage: 55%   Mem Usage: 123324K / 310364K
```

Notice how quickly Floodz ramped up the processor load. The previous hilts on the diagram are 30 sec runs made against the system.

I also used the Fyre tool from Divine Intervention which is an OOB nuker. OOB stands for "Out of Band" and signifies badly constructed packets sent out to confuse or disable susceptible systems. In this case, I did not see the same spike in activity. Perhaps the latest patches enable the system to be smarter about handling these packets or the tool was malfunctioning.

A quick word of warning for those attempting to use this Divine Intervention, be very careful and do not install it on a system with any valuable information. I sampled scores of different sites that had the software and each one had my antivirus system warning of infection with Trojan software.

What does this show us? That from an easily obtained and downloaded tool any would be hacker in your organization could pose a threat to your internal systems. The same goes for hackers external to your system.

Continuing on let us take a look at what happens when we send these same attacks against a system with layers of protection outside

Let us examine when we send the same attacks against a well defended perimeter. I sent the same portscan, packet floods, and OOB packets against a Checkpoint NG Firewall running Feature Pack 3 with a Symantec Manhunt 2.2 IDS Sensor monitoring the attacks. As is seen from the logs below the Firewall easily shrugged off the attacks

Checkpoint NG FP3

```
"143" "25Jun2003" "22:22:44" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"" "EXT.NET.WRK.99" "INT.NET.WRK.68" "icmp" "7" "" "" "icmp-type: 8; icmp-code: 0; "
"144" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"1" "EXT.NET.WRK.99" "INT.NET.WRK.68" "tcp" "7" "48467" "" ""
"145" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"2" "EXT.NET.WRK.99" "INT.NET.WRK.68" "tcp" "7" "48468" "" ""
"146" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"3" "EXT.NET.WRK.99" "INT.NET.WRK.68" "tcp" "7" "48469" "" ""
"147" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"5" "EXT.NET.WRK.99" "INT.NET.WRK.68" "tcp" "7" "48470" "" ""
"148" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"7" "EXT.NET.WRK.99" "INT.NET.WRK.68" "tcp" "7" "48471" "" ""
"149" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log" "Drop"
"9" "EXT.NET.WRK.99" "INT.NET.WRK.68" "tcp" "7" "48472" "" ""
"150" "25Jun2003" "22:22:45" "VPN-1 & FireWall-1" "eth1" "192.168.0.254" "Log"
```

IDS: Symantec Manhunt v2.2

```
EVENT_TIME = Wed Jun 25 22:26:21 EDT 2003
INCIDENT_START = Wed Jun 25 22:26:21 EDT 2003
DEVICE = Lab External Hub
LOCALIFACE = CopyPortHub
INCIDENTFAMILY = availability
INCIDENTTYPE = RCRS/COUNTER_TCP_PORTSCAN
DESC = Portscan
PRIORITY = Critical
ALERT_NUMBER = 1
SRC_IP_LIST = EXT.NET.WRK.99:50212
DST_IP_LIST = INT.NET.WRK.68:1542
FLOWCOOKIE =
TCP%COUNTER,SPOOF,SYNS%EXT.NET.WRK.99:50212/INT.NET.WRK.68:1542#46
08
```

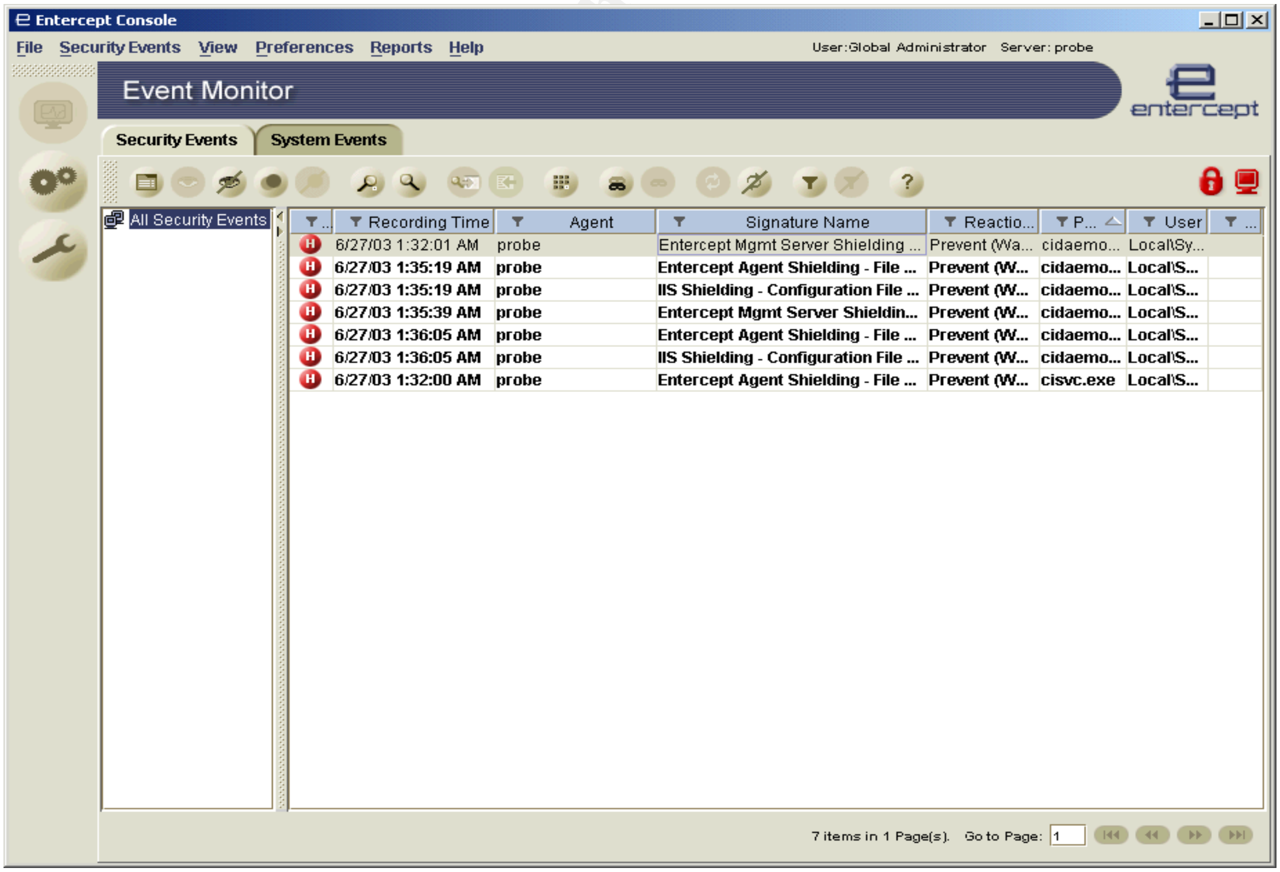
The firewall is dropping the offending packets. From the IDS system we can see who is generating the attack, what sort of attack it is and retain a log to pursue legal action later, if necessary. If we placed the sensor on the inside and the attacker was internal to our network we would be able to track down the user or compromised system that was causing issues with our web server. In the case of Manhunt, a reset could be sent to terminate TCP connections with offending hosts.

So at this juncture we see the Perimeter doing its job of defending the host. Denial of Service attacks are being handled and offenders are being tracked. Why would it be necessary to put up an Intrusion Prevention System?

The story actually becomes grimmer for Host Based Intrusion prevention. With an IPS agent running on a system they in general and in the case of Entercept, will not protect against DDOS attacks. But that really is not their purpose. The IPS protect when the attack actually gets to the host past the network interface card or NIC. Flooding can still fill up the buffer on the NIC and tie up resources. For those avenues of attack host-based intrusion prevention must rely on the network controls.

So again why is it necessary to have Host Based Intrusion prevention? I would offer up for conjecture that the attacks do not just stop at the network but continue into the OS itself. It is there that the IPS systems shine.

Examine the information below.



Entercept Console
File Security Events View Preferences Reports Help
User:Global Administrator Server: probe
Event Monitor
entercept
Security Events
System Events
All Security Events

..	Recording Time	Agent	Signature Name	Reactio...	P...	User	...
H	6/27/03 1:32:01 AM	probe	Entercept Mgmt Server Shielding ...	Prevent (Wa...	cidaemo...	Local\Sy...	
H	6/27/03 1:35:19 AM	probe	Entercept Agent Shielding - File ...	Prevent (W...	cidaemo...	Local\S...	
H	6/27/03 1:35:19 AM	probe	IIS Shielding - Configuration File ...	Prevent (W...	cidaemo...	Local\S...	
H	6/27/03 1:35:39 AM	probe	Entercept Mgmt Server Shieldin...	Prevent (W...	cidaemo...	Local\S...	
H	6/27/03 1:36:05 AM	probe	Entercept Agent Shielding - File ...	Prevent (W...	cidaemo...	Local\S...	
H	6/27/03 1:36:05 AM	probe	IIS Shielding - Configuration File ...	Prevent (W...	cidaemo...	Local\S...	
H	6/27/03 1:32:00 AM	probe	Entercept Agent Shielding - File ...	Prevent (W...	cisvc.exe	Local\S...	

7 items in 1 Page(s). Go to Page: 1

Here we see a Host Based Intrusion Prevention system blocking the changing of key files. Some worms will attempt to take out antivirus and other controls once they land on a host. HIPS will look for processes that are operating outside of their “sandbox” and prevent them from occurring.

Most Host Based Intrusion Prevention systems, NAI’s Entercept or Cisco’s Okena, will do just that; look for anomalous behavior and prevent it from occurring. To speed up that process, Entercept adds a signature base of known attacks.

Another area where Host Based Intrusion Prevention Systems prove useful is in the area of buffer overflow attacks. In a nutshell, these are attacks that attempt to stuff too much information into some variable array and cause it to overflow its buffer. These buffer overflows can then cause erratic behavior on a device. Either the device is shut down or the process fails to another state. If the attacker can get a highly privileged process to fail, then it is possible for the attacker to run code at the privilege level of the process. If this is a high level process then the attacker would have high level permissions. In essence this is a bad thing and should be protected against.

But how are the other layers going to protect us? There may be a signature for the attack that the network can see, but what happens when the exploit is delivered in an encrypted fashion on an approved port? The firewall will not be able to open the encrypted data and neither will the IDS system. It will arrive at the host, be decrypted and possible harm that host, **unless another layer is there for protection**.

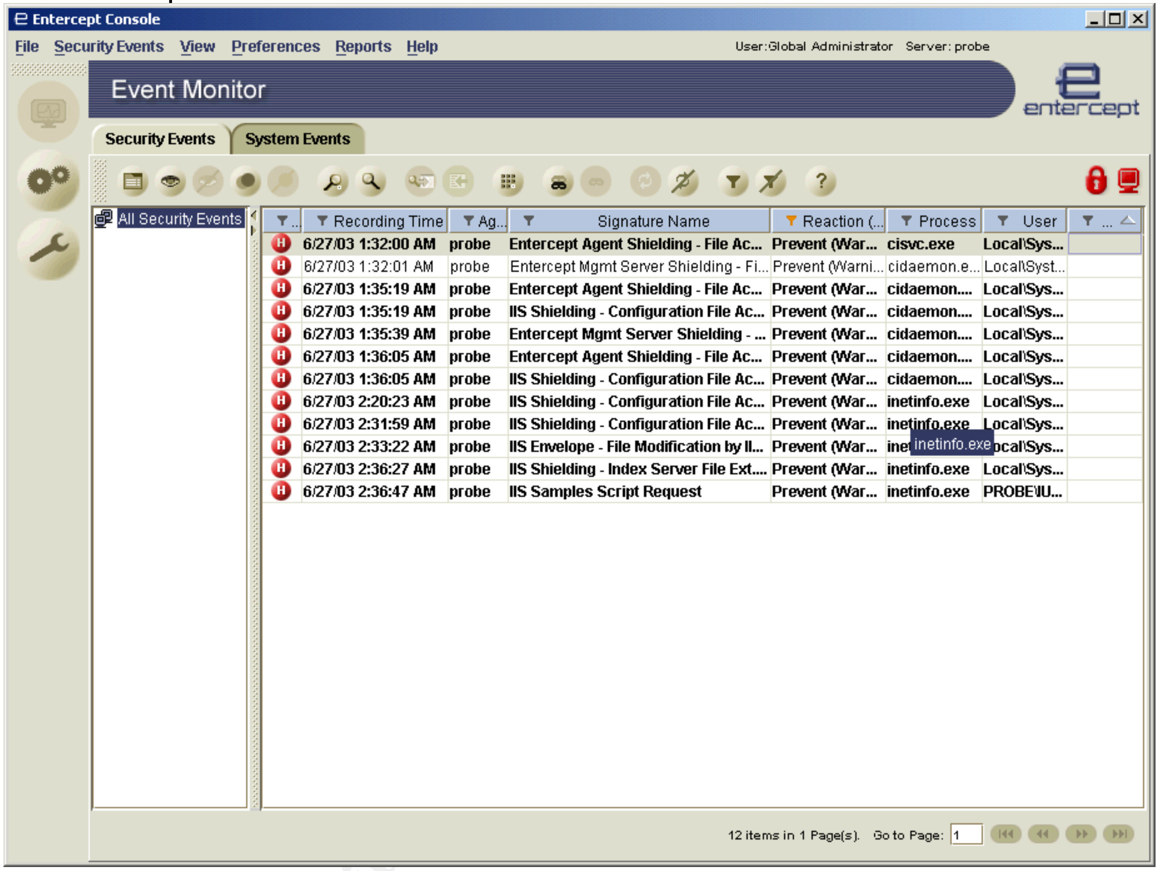
Lets look at another area where having that layer of protection is useful. Another method for an attacker to compromise a system is with invalid escape sequences or extra characters when performing HTTP Get operations to a Web Server. Code Red attacked by providing too many “N”s in during a Get and “overflowing” the web servers, allowing a compromise to occur.

Below we see how a HIPS handles these sort of events. From the CERT Advisory for Code Red, we attempted to get access to the Indexing Service for Microsoft’s Internet Information Server or web server. (CERT Advisory CA-2001-19)

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNN
```

```
NNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%u
cbd3%
u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531
b%u53ff%u0078%u0000%u00=a
```

And the response can be seen a the bottom.



▼ ..	▼ Recording Time	▼ Ag...	▼ Signature Name	▼ Reaction (...	▼ Process	▼ User	▼ ...
H	6/27/03 1:32:00 AM	probe	Entercept Agent Shielding - File Ac...	Prevent (War...	cisvc.exe	Local\Sys...	
H	6/27/03 1:32:01 AM	probe	Entercept Mgmt Server Shielding - Fi...	Prevent (Warni...	cidaemon.e...	Local\Syst...	
H	6/27/03 1:35:19 AM	probe	Entercept Agent Shielding - File Ac...	Prevent (War...	cidaemon....	Local\Sys...	
H	6/27/03 1:35:19 AM	probe	IIS Shielding - Configuration File Ac...	Prevent (War...	cidaemon....	Local\Sys...	
H	6/27/03 1:35:39 AM	probe	Entercept Mgmt Server Shielding - ...	Prevent (War...	cidaemon....	Local\Sys...	
H	6/27/03 1:36:05 AM	probe	Entercept Agent Shielding - File Ac...	Prevent (War...	cidaemon....	Local\Sys...	
H	6/27/03 1:36:05 AM	probe	IIS Shielding - Configuration File Ac...	Prevent (War...	cidaemon....	Local\Sys...	
H	6/27/03 2:20:23 AM	probe	IIS Shielding - Configuration File Ac...	Prevent (War...	inetinfo.exe	Local\Sys...	
H	6/27/03 2:31:59 AM	probe	IIS Shielding - Configuration File Ac...	Prevent (War...	inetinfo.exe	Local\Sys...	
H	6/27/03 2:33:22 AM	probe	IIS Envelope - File Modification by II...	Prevent (War...	inet inetinfo.exe	ocal\Sys...	
H	6/27/03 2:36:27 AM	probe	IIS Shielding - Index Server File Ext....	Prevent (War...	inetinfo.exe	Local\Sys...	
H	6/27/03 2:36:47 AM	probe	IIS Samples Script Request	Prevent (War...	inetinfo.exe	PROBE\IU...	

12 items in 1 Page(s). Go to Page: 1

This particular server was configured in Warn mode versus Prevent mode**.** Entercept makes these distinctions by running in Warn mode first in order to learn what is realm of proper behavior and what is anomalous behavior for a server. After the acceptable behavior is mapped out Entercept can be placed in Prevent mode to protect against attacks.

Summary:

We have seen areas where Host Based Intrusion Prevention is not really the best solution. In the case of Denial Service Attacks it is more effective to have additional layers of network protection such as a firewall and an IDS. However, we have also shown areas where it might not be possible for a firewall or network IDS to screen for attacks such as with encrypted data, rogue users, or compromise a **process on a system**. In addition, acceptable processes coming over an approved channel from a compromised system can slip past network controls, if the access list is not tuned correctly. In those cases where the attack has bypassed or circumvented network controls, it might be prudent to include additional layers at the host. Through the proper use of Network and Host Based controls, we could come close to making our position, in Sun Tzu's words, "unassailable". However, as we have seen in the brief examples above, no one layer of protection will be able to reach that level of security on its own. For some systems, depending on company or organization policy, having Host Based Intrusion Prevention systems available and deployed might make significant impact on their overall risk exposure.

List of References – Section I

1. Sun Tzu on the Art of War Chapter VIII. Variation in Tactics 11, Translated from the Chinese with Introduction and Critical Notes BY LIONEL GILES, M.A. http://www.kimsoft.com/polwar8.htm
2. Testing Entercept –Live!, A Step-by-Step Guide to Testing Entercept with Real Exploits, Entercept Security Technologies. Revision: 08012001
3. Konidaris, George D. Divine Intervention III User's Manual, February 1998
4. CERT Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow in IIS Indexing Service DLL, January 17,2002 http://www.cert.org/advisories/CA-2001-19.html
5. Fyodor Fyodor@insecure.org. The Art of Port Scanning, 1997 http://www.insecure.org/nmap/nmap_doc.html
6. SuperScan v3.0 Copyright 2000 (c) by Foundstone, Inc. http://www.foundstone.com

Section II Network Detects

1. Network detect – TCP data offset is less than 5

{Note this detect was submitted to incidents.org on 6/30/2003 for peer review. No responses were received}

1.1 Source of Trace.

The source of this first trace is from the www.incidents.org/raw 2002.10.16 log. The network information was not made readily available but it is possible to make some qualified assumptions.

My first fix of the network is certain traffic that generally occurs on a local segment.   Please note that having a nautical background, I use the term “fix” in terms of identifying position or location. To me the practice of aligning several landmarks to determine estimated position is very similar to network analysis where we identify features of traffic and attempt to determine intent or risk.

Moving on, in the trace I was able to detect a number of IGMP queries.   These IGMP queries were actually examined in Daniel Wesemann’s network detect for Bad Traffic (Wesemann, Daniel). Whether the traffic is spoofed, which was not determined at the end of his analysis, or innocuous traffic it still serves as good fix on what the area of interest on the local network looks like.

```
09:33:01.856507 IP 170.129.164.3 > 170.129.164.3: igmp query v2 [gaddr 240.0.2.91]
09:33:01.856507 IP 170.129.164.9 > 170.129.164.9: igmp query v2 [gaddr 240.0.2.97]
09:33:01.856507 IP 170.129.164.14 > 170.129.164.14: igmp query v2 [gaddr 240.0.2.102]
```

From the Cisco Whitepaper “Overview of IP Multicast”

> “The Internet Group Multicast Protocol (IGMP) is an IP datagram protocol between routers and hosts that allows group membership lists to be dynamically maintained. The host sends an IGMP "report", or join, to the router to join the group. Periodically, the router sends a "query" to learn which hosts are still part of a group. If a host wants to continue its group membership, it responds to the query with a report. If the host sends no report, the router prunes the group list to minimize unnecessary transmissions. With IGMP V2, a host may send a "leave" message to inform the router that it no longer is participating in a multicast group. This allows the router to prune the group list before the next query is scheduled, minimizing the time period in which wasted transmissions are forwarded to the network.” (Overview of IP Multicast)

This seems to indicate that the local addresses are in the 170.129.0.0 address range as we have hosts reporting with IGMP V2 to a local router.

Another fix we can look at is to see where the range is registered and see if that correlates to any other data. This address range is in the ARIN data base to:

```
OrgName:    Standard Microsystems Corporation
OrgID:      SMC-9
Address:    300 Kennedy Drive
City:       Hauppauge Industrial Park
StateProv:  NY
PostalCode:
Country:    US

NetRange:   170.129.0.0 - 170.129.255.255
```

```
CIDR:       170.129.0.0/16
NetName:    SMCORP
NetHandle:  NET-170-129-0-0-1
Parent:     NET-170-0-0-0-0
NetType:    Direct Assignment
NameServer: NS.PSI.NET
NameServer: NS2.PSI.NET
Comment:
RegDate:    1994-04-29
Updated:    1994-05-25
```

We see a lot of traffic to a midgard.smsc.com and to www.smsc.com. This coupled with the information above seems to have the location of the sensor within the 170.129.0.0 range.

Reviewing the trace with the Ethereal- Network Analyzer version 0.9.6 revealed additional information. Examining the network traffic source and destination showed Mac addresses for Cisco Routers. It would seem to indicate that the sensor is placed between two Cisco routers.

Going out further on a limb, it appears that the sensor is on an internal network and not on the DMZ. If we had Mac addresses from a PIX or something other than a Cisco Router that would open up the possibility of the sensor being on a DMZ segment. That does not appear to be the case here, unless they are utilizing the Firewall Feature set on the external router.

At this juncture we have what appears to be a sensor placed on an internal segment with a valid Internet addresses in the 170.129.0.0 range which is owned by Standard Microsystems Corporation.

1.2 Detect was generated by:

This detect was generated by Snort Version 2.0.0 running on Windows XP with Winpcap 3.0. This is not a rule violation but a warning from the Snort decoder that examines TCP data.

The rule used to review the data was:

```
 snort -d -e  -A console -c snort.conf -r G:\Traces\2002.10.16 -h 170.129.0.0/16
```

The alerts generated were as follows:

```
        11/15-21:11:54.416507  [**] [116:46:1] (snort_decoder) WARNING: TCP
        Data Offset is less than 5! [**] {TCP} 68.41.28.138:0 -> 170.129.23.60:0
        11/15-00:36:10.986507  [**] [116:46:1] (snort_decoder) WARNING: TCP
        Data Offset is less than 5! [**] {TCP} 210.243.145.141:0 ->
        170.129.134.11:0
```

I decided to look deeper into the packets to find any additional data. For that, I employed Windump to the task. Windump is a port of the popular TCPDUMP network analysis tool. (Windump)

I decided to first look if any other traffic was heading to those destinations. To do this I used the WINDUMP filtering options “dst” for destination and –X to see deeper into the packet. The following command was used:

```
>windump -n -X -r 2002.10.16 dst host 170.129.23.60 or dst host 170.129.134.11
```

```
        21:11:54.416507 IP 68.41.28.138.4110 > : . 1531912236:1531912264(28)
        win 28674 (DF)
        0x0000   4500 0030 9bb8 4000 6a06 529f 4429 1c8a        E..0..@.j.R.D)..
        0x0010   aa81 173c 100e 0050 5b4f 202c 0000 0000        ...<...P[O.,....
        0x0020   0000 7002 14f0 14f0 c381 0000 0204 0218        ..p............

        00:36:10.986507 IP 210.243.145.141.3751 > 170.129.134.11.80: R
        704834360:704834368(8) win 0
        0x0000   4500 0028 0000 0000 ec06 39c2 d2f3 918d        E..(......9.....
        0x0010   aa81 860b 0ea7 0050 2a02 eb38 2a02 eb38         .......P*..8*..8
        0x0020   3604 0000 fb65  0000 0000 0000 0000            6....e........
```

Using Ethereal to review the packets for that time we see the following:

```
        Frame 165 (62 on wire, 62 captured)
        Arrival Time: Nov 15, 2002 20:11:54.416507000
        Time delta from previous packet: 0.000000000 seconds
        Time relative to first packet: 13506.810000000 seconds
        Frame Number: 165
        Packet Length: 62 bytes
        Capture Length: 62 bytes
        Ethernet II
        Destination: 00:00:0c:04:b2:33 (Cisco_04:b2:33)
        Source: 00:03:e3:d9:26:c0 (Cisco_d9:26:c0)
        Type: IP (0x0800)
        Internet Protocol, Src Addr: pcp02097455pcs.brmngh01.mi.comcast.net
        (68.41.28.138), Version: 4
        Header length: 20 bytes
        Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
        Total Length: 48
        Identification: 0x9bb8
        Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
        Fragment offset: 0
        Time to live: 106
        Protocol: TCP (0x06)
        Header checksum: 0x529f (correct)
        Source: pcp02097455pcs.brmngh01.mi.comcast.net (68.41.28.138)
        Destination: 170.129.23.60 (170.129.23.60)
        Transmission Control Protocol, Src Port: 4110 (4110), Dst Port: http (80),
        Seq: Source port: 4110 (4110)
        Destination port: http (80)
        Sequence number: 1531912236
        Header length: 0 bytes (bogus, must be at least 20)
```

In this case we see that Ethereal is reporting a bogus TCP header length of 0 bytes.

The TCP Header size should be at least 20 bytes. (Stevens, p.225)
There is definitely something wrong with this packet. This alert is not generated by a specific ruleset for Snort but by one of the preprocessors. Some of the correlations I have seen might point to a defective piece of equipment.
Russell Fulton mentioned seeing a lot of these type fragments coming from 3 Akamai boxes in his DMZ during a recent incident. (Fulton, Russell)
Phil Wood, mentions seeing another of these fragments as well. (Wood, Phil)

That is what is interesting from this trace; we see only two single packets to two different locations from two different locations. Searching that day we see no other packets to or from host 170.129.23.60 or host 170.129.134.11. Many of the other correlations all pointed to numerous packets being received most likely from a faulty system or NIC. So in this case if we took the tack that this was a faulty device we would expect a large number of packets. I examined the logs from the previous day and only saw two entries.

A nslookup report from Sam Spade on the source ports show:

```
        06/21/03 18:06:36 dns 68.41.28.138
        nslookup 68.41.28.138
        Canonical name: pcp02097455pcs.brmngh01.mi.comcast.net
        Addresses:
          68.41.28.138
```

The second source address was not listed as a specific address but a lookup from Sam Spade showed the address from where it might have come from:

```
        06/21/03 18:09:35 IP block 210.243.145
        Address 210.243.145 is 210.243.0.145
        Trying 210.243.0.145 at ARIN
        Trying 210.243.0 at ARIN
```

```
OrgName:    Asia Pacific Network Information Centre
OrgID:      APNIC
Address:    PO Box 2131
City:       Milton
StateProv:  QLD
PostalCode: 4064
Country:    AU

NetRange:   210.0.0.0 - 211.255.255.255
CIDR:       210.0.0.0/7
NetName:    APNIC-CIDR-BLK2
NetHandle:  NET-210-0-0-0-1
Parent:
NetType:    Allocated to APNIC
NameServer: NS1.APNIC.NET
NameServer: NS3.APNIC.NET
NameServer: NS.RIPE.NET
NameServer: RS2.ARIN.NET
NameServer: DNS1.TELSTRA.NET
```

Filtering for the source ports in the trace we find that these two source addresses only spoke to the two destination addresses.

```
G:\windump -n -X -r 2002.10.16  host 68.41.28.138 or host 210.243.145.141
        21:11:54.416507 IP 68.41.28.138.4110 > 170.129.23.60.80: .
        1531912236:1531912264(28) win 28674 (DF)
        0x0000   4500 0030 9bb8 4000 6a06 529f 4429 1c8a        E..0..@.j.R.D)..
        0x0010   aa81 173c 100e 0050 5b4f 202c 0000 0000        ...<...P[O.,....
        0x0020   0000 7002 14f0 14f0 c381 0000 0204 0218       ..p............

        00:36:10.986507 IP 210.243.145.141.3751 > 170.129.134.11.80: R
        704834360:704834368(8) win 0
        0x0000   4500 0028 0000 0000 ec06 39c2 d2f3 918d        E..(......9.....
        0x0010   aa81 860b 0ea7 0050 2a02 eb38 2a02 eb38         .......P*..8*..8
        0x0020   3604 0000 fb65 0000 0000 0000 0000            6....e........
```

Another interesting point is that the second packet is sending a reset back to 170.129.134.11 but there was no initiating traffic from 170.128.134.11.

1.3 Probability the source address was spoofed:

It is difficult to say with the information at hand whether the source address was spoofed. This is either traffic generated by a faulty device or a deliberate attempt to provoke a reaction from a system. Seeing how there was no reply traffic from the destination it is unlikely to be a mapping technique. If so then it is not a particularly effective one. More likely it is an attempt to cause the system to behave erratically. Attacks of that type are generally spoofed to prevent back tracking to the attacker. So the source address is probably spoofed in this case.

1.4 Description of attack:

I was unable to locate a specific attack that utilized invalid TCP size either for mapping or denial of service but that does not mean that there is not an attack that exploits this. Right now there are just two incidents that might be acting in conjunction or might be just two separate events.

There is a possibility that these events are caused by two faulty devices. The fact that the second event was sending uninitiated resets is suspicious and warrants further monitoring. Also going back further into the logs might elicit other attempts to see systems inside with a stealthy scan though the previous log only had 2 entries as well.

There might be some legacy systems or stacks that are susceptible to an invalid TCP size. I was unable to locate any.

1.5 Attack mechanism:

The attack would appear to function by means of causing a system to fail with corrupt input. Since most stacks expect a TCP packet to have at least 20 bytes of data without options, there might be a stack out there that would fail if given less.

1.6 Correlations:

This type of detect was also examined in another net detect by Daniel Clark (Clark, Daniel).
http://cert.uni-stuttgart.de/archive/intrusions/2003/05/msg00183.html

He took the tack that this was just malformed traffic. For the reasons detailed above I am less inclined to jump to that conclusion. Many of the other people experiencing this traffic reported large amounts of the packets received.
(Fulton, Russell)
http://msgs.securepoint.com/cgi-bin/get/snort-0304/352.html

(Snort-users Mailing List)
http://www.somelist.com/mails/312073.html

(Wood, Phil)
http://archives.neohapsis.com/archives/snort/2000-12/0413.html

But in this case we do not see a device a large dump of packets on the wire. We only have two incidents in this log and two in the previous day's log. Granted a defective device might not fail all at once but send a trickle of corrupt packets before finally failing. The other incidents above did not seem to mention that sort of behavior. Faulty devices generated lots of traffic.

1.7 Evidence of active targeting:

It is more likely that this is a directed attack. Neither destination system was listed in the DNS tables for the site yet the attacks were sent to only two systems on port 80. I did not see a shotgun approach or attempt to a series of addresses.

A check of Dshield.org for the 68.41.28.138 address shows that a fightback message was sent to Comcast but that the message was bounced.

Last Fightback Sent: sent to abuse@comcastpc.com on 2002-11-15 22:31:46
message bounced

The time frame is in line with the 2002.10.16 data and probably a result of it. No other listing for this address was found.
There is no mention of the second address at dshield.org 210.243.145.141.
For these specific IP addresses there does not appear to be a pattern of abuse.
If the potential attacker was trying to elicit a response back that was unsuccessful and probably unlikely.

1.8 Severity:

A good equation for determining the risk of exposure to an attack is by examining severity. One equation for determining the severity of an attack is by examining the strength of an attack with the protections in place. The strength of an attack is looked at by the force or lethality it encompasses. A rock thrown at a brick wall will in general bounce off. A rock thrown at a plate glass window will cause much more damage.

The safeguards in place should also be looked at. Now if the plate glass window above has strong wire mesh in front of it we would have good system countermeasures.

Criticality: In this case the criticality is 1. The systems are not mentioned in the DNS tables and see little activity.

Lethality: I would score the lethality as 1 as well. Found little documentation identifying this as an attack including the default rule set for SNORT.

System countermeasures: This is difficult to measure in this case without additional traffic. Right now we only have external source traffic destined internally with an improper header. Do the internal systems have a host based firewall that is screening the traffic and preventing response? Do the systems even exist? There is no other IP activity from those IP addresses either sending or receiving during the trace. I would score this as a 3. Either the systems are there and have some local screening or they do not exist. Not being at the receiving end of a punch is an effective countermeasure.

Network countermeasures: This would be a 1. Bad traffic is being allowed through to the internal system.
Severity = [Criticality (1) + Lethality (1)] – [System Counters (1) + Network Counters (3)]
Severity = 2 – 4 = Negative 2. We do not need to deploy the tiger teams just yet.

1.9 Defensive recommendation:

I would recommend greater screening measures at the perimeter either by adding a stateful firewall or if there is a firewall, improving its rulebase.
If this is a faulty device and the network was being flooded by the TCP offsets we have a couple of options either tuning the network processors to ignore the data or setting up filtering for that particular device.

However, I would continue to monitor at this juncture. We do not see enough data coming from the packets to determine if this is a stealthy attack or just a misconfigured device. Trend analysis over time might show a larger pattern.

1.10 Multiple choice test question:

You are a newly hired intrusion analyst at a multinational corporation. You receive your first Windump logs from your satellite office in London to analyze. London has recently had a rash of SQL slammer attacks. What Windump filters would you use to identify possible SQL slammer traffic activity on your network.

a.. UDP and dst port 1443
b. TCP and dst port 1434
c. ip[9]=17 and udp[2] = 1443
d. UDP and udp[3] = 1434

Answer = D

Explanation: SQL slammer is a UDP based attack to port 1434. The other important thing to remember with WINDUMP filters is that the counting starts at 0 vice 1. In this case Answer D best filters for UDP data to port 1434.

2.0 Network Detect – Buffer overflow attack against the IIS Indexing Service DLL

2.1 Source of Trace.

The source of this trace is the data located on www.incidents.org/raw 2002.10.15. The network appears to be the same as in network detect 1. We see substantial traffic to web servers www.smsc.com and migaard.smsc.com. The IP addresses for those host names are in the range of 170.129.0.0.
 This address range is registered in the ARIN data base to:

```
OrgName:    Standard Microsystems Corporation
OrgID:      SMC-9
Address:    300 Kennedy Drive
City:       Hauppauge Industrial Park
StateProv:  NY
PostalCode:
Country:    US

NetRange:   170.129.0.0 - 170.129.255.255
CIDR:       170.129.0.0/16
NetName:    SMCORP
NetHandle:  NET-170-129-0-0-1
Parent:     NET-170-0-0-0-0
NetType:    Direct Assignment
NameServer: NS.PSI.NET
NameServer: NS2.PSI.NET
Comment:
RegDate:    1994-04-29
Updated:    1994-05-25
```

I reviewed the network log with Ethereal- Network Analyzer version 0.9.6. Examining the network traffic source and destination addresses showed Mac addresses for Cisco Routers. It would seem to indicate that the sensor is on an internal network and not on the DMZ. If we had Mac addresses from a PIX or something other than a Cisco Router that would open up the possibility of the sensor being on a DMZ segment. That does not appear to be the case here.

2.2 Detect was generated by:

This detect was generated by reviewing the 2002.10.15 log with Ethereal. Snort version 2.0 running the standard ruleset missed this traffic but picked up the TCP offset less than 5 issues and bad traffic issues with IGMP.

A simple visual review of the snort data came up with this interesting http sequence that jumped off the page. The www.worm.com in the payload immediately caught the eye as well as the very large HTTP Get.

```
Frame 389 (1482 on wire, 1482 captured)
```

```
    Arrival Time: Nov 15, 2002 01:01:40.376507000
    Time delta from previous packet: 124.260000000 seconds
    Time relative to first packet: 31908.120000000 seconds
    Frame Number: 389
    Packet Length: 1482 bytes
    Capture Length: 1482 bytes
Ethernet II
    Destination: 00:00:0c:04:b2:33 (Cisco_04:b2:33)
    Source: 00:03:e3:d9:26:c0 (Cisco_d9:26:c0)
    Type: IP (0x0800)
Internet Protocol, Src Addr: pc1-glen1-3-cust7.blfs.cable.ntl.com (213.107.102.7), Dst Addr:
170.129.168.122 (170.129.168.122)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 1468
    Identification: 0x574a
    Flags: 0x06
        .1.. = Don't fragment: Set
        ..1. = More fragments: Set
    Fragment offset: 0
    Time to live: 111
    Protocol: TCP (0x06)
    Header checksum: 0x0083 (correct)
    Source: pc1-glen1-3-cust7.blfs.cable.ntl.com (213.107.102.7)
    Destination: 170.129.168.122 (170.129.168.122)
Transmission Control Protocol, Src Port: 4813 (4813), Dst Port: http (80), Seq: 3802143269, Ack:
2909583458, Len: 1416
    Source port: 4813 (4813)
    Destination port: http (80)
    Sequence number: 3802143269
    Next sequence number: 3802144685
    Acknowledgement number: 2909583458
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 64240
    Checksum: 0xf6ea
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 635429, tsecr 3587374810
Hypertext Transfer Protocol
    GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
Content-type: text/xml\n
    Content-type: text/xml\n
    HOST:www.worm.com\n
     Accept: */*\n
```

```
   Content-length: 3569 \r\n
   \r\n
   Data (946 bytes)

0000  55 8b ec 81 ec 18 02 00 00 53 56 57 8d bd e8 fd   U........SVW....
0010  ff ff b9 86 00 00 00 b8 cc cc cc cc f3 ab c7 85   ................
0020  70 fe ff ff 00 00 00 00 e9 0a 0b 00 00 8f 85 68   p..............h
0030  fe ff ff 8d bd f0 fe ff ff 64 a1 00 00 00 00 89   .........d......
0040  47 08 64 89 3d 00 00 00 00 e9 6f 0a 00 00 8f 85   G.d.=.....o.....
0050  60 fe ff ff c7 85 f0 fe ff ff ff ff ff ff 8b 85   `...............
0060  68 fe ff ff 83 e8 07 89 85 f4 fe ff ff c7 85 58   h..............X
0070  fe ff ff 00 00 e0 77 e8 9b 0a 00 00 83 bd 70 fe   ......w.......p.
0080  ff ff 00 0f 85 dd 01 00 00 8b 8d 58 fe ff ff 81   ...........X....
0090  c1 00 00 01 00 89 8d 58 fe ff ff 81 bd 58 fe ff   .......X.....X..
00a0  ff 00 00 00 78 75 0a c7 85 58 fe ff ff 00 00 f0   ....xu...X......
00b0  bf 8b 95 58 fe ff ff 33 c0 66 8b 02 3d 4d 5a 00   ...X...3.f..=MZ.
00c0  00 0f 85 9a 01 00 00 8b 8d 58 fe ff ff 8b 51 3c   .........X....Q<
00d0  8b 85 58 fe ff ff 33 c9 66 8b 0c 10 81 f9 50 45   ..X...3.f.....PE
00e0  00 00 0f 85 79 01 00 00 8b 95 58 fe ff ff 8b 42   ....y.....X....B
00f0  3c 8b 8d 58 fe ff ff 8b 54 01 78 03 95 58 fe ff   <..X....T.x..X..
0100  ff 89 95 54 fe ff ff 8b 85 54 fe ff ff 8b 48 0c   ...T.....T....H.
0110  03 8d 58 fe ff ff 89 8d 4c fe ff ff 8b 95 4c fe   ..X.....L.....L.
0120  ff ff 81 3a 4b 45 52 4e 0f 85 33 01 00 00 8b 85   ...:KERN..3.....
0130  4c fe ff ff 81 78 04 45 4c 33 32 0f 85 20 01 00   L....x.EL32.. ..
0140  00 8b 8d 58 fe ff ff 89 8d 34 fe ff ff 8b 95 54   ...X.....4.....T
0150  fe ff ff 8b 85 58 fe ff ff 03 42 20 89 85 4c fe   .....X....B ..L.
0160  ff ff c7 85 48 fe ff ff 00 00 00 00 eb 1e 8b 8d   ....H...........
0170  48 fe ff ff 83 c1 01 89 8d 48 fe ff ff 8b 95 4c   H........H.....L
0180  fe ff ff 83 c2 04 89 95 4c fe ff ff 8b 85 54 fe   ........L.....T.
0190  ff ff 8b 8d 48 fe ff ff 3b 48 18 0f 8d c0 00 00   ....H...;H......
01a0  00 8b 95 4c fe ff ff 8b 02 8b 8d 58 fe ff ff 81   ...L.......X....
01b0  3c 01 47 65 74 50 0f 85 a0 00 00 00 8b 95 4c fe   <.GetP........L.
01c0  ff ff 8b 02 8b 8d 58 fe ff ff 81 7c 01 04 72 6f   ......X....|..ro
01d0  63 41 0f 85 84 00 00 00 8b 95 48 fe ff ff 03 95   cA........H.....
01e0  48 fe ff ff 03 95 58 fe ff ff 8b 85 54 fe ff ff   H.....X.....T...
01f0  8b 48 24 33 c0 66 8b 04 0a 89 85 4c fe ff ff 8b   .H$3.f.....L....
0200  8d 54 fe ff ff 8b 51 10 8b 85 4c fe ff ff 8d 4c   .T....Q...L....L
0210  10 ff 89 8d 4c fe ff ff 8b 95 4c fe ff ff 03 95   ....L.....L.....
0220  4c fe ff ff 03 95 4c fe ff ff 03 95 4c fe ff ff   L.....L.....L...
0230  03 95 58 fe ff ff 8b 85 54 fe ff ff 8b 48 1c 8b   ..X.....T....H..
0240  14 0a 89 95 4c fe ff ff 8b 85 4c fe ff ff 03 85   ....L.....L.....
0250  58 fe ff ff 89 85 70 fe ff ff eb 05 e9 0d ff ff   X.....p.........
0260  ff e9 16 fe ff ff 8d bd f0 fe ff ff 8b 47 08 64   .............G.d
0270  a3 00 00 00 00 83 bd 70 fe ff ff 00 75 05 e9 38   .......p....u..8
0280  08 00 00 c7 85 4c fe ff ff 01 00 00 00 eb 0f 8b   .....L..........
0290  8d 4c fe ff ff 83 c1 01 89 8d 4c fe ff ff 8b 95   .L........L.....
02a0  68 fe ff ff 0f be 02 85 c0 0f 84 8d 00 00 00 8b   h...............
02b0  8d 68 fe ff ff 0f be 11 83 fa 09 75 21 8b 85 68   .h.........u!..h
02c0  fe ff ff 83 c0 01 8b f4 50 ff 95 90 fe ff ff 3b   ........P......;
02d0  f4 90 43 4b 43 4b 89 85 34 fe ff ff eb 2a 8b f4   ..CKCK..4....*..
02e0  8b 8d 68 fe ff ff 51 8b 95 34 fe ff ff 52 ff 95   ..h...Q..4...R..
02f0  70 fe ff ff 3b f4 90 43 4b 43 4b 8b 8d 4c fe ff   p...;..CKCK..L..
0300  ff 89 84 8d 8c fe ff ff eb 0f 8b 95 68 fe ff ff   ............h...
0310  83 c2 01 89 95 68 fe ff ff 8b 85 68 fe ff ff 0f   .....h.....h....
0320  be 08 85 c9 74 02 eb e2 8b 95 68 fe ff ff 83 c2   ....t.....h.....
0330  01 89 95 68 fe ff ff e9 53 ff ff ff 8b 85 68 fe   ...h....S.....h.
0340  ff ff 83 c0 01 89 85 68 fe ff ff 8b 4d 08 8b 91   .......h....M...
0350  84 00 00 00 89 95 6c fe ff ff c7 85 4c fe ff ff   ......l.....L...
0360  04 00 00 00 c6 85 d0 fe ff ff 68 8b 45 08 89 85   ..........h.E...
0370  d1 fe ff ff c7 85 d5 fe ff ff 5b 53 53 ff c7 85   ..........[SS...
0380  d9 fe ff ff 63 78 90 90 8b 4d 08 8b 51 10 89 95   ....cx...M..Q...
0390  50 fe ff ff 83 bd 50 fe ff ff 00 75 26 8b f4 6a   P.....P....u&..j
```

```
03a0  00 8d 85 4c fe ff ff 50 8b 8d 68 fe ff ff 51 8b   ...L...P..h...Q.
03b0  55 08                                             U.
```

Along with the excessive use of “n” in the Get sequence we see that the do not fragment and fragment fields are both.

This has the earmarks of a buffer overflow attack against a Microsoft web server. The “NNNNN” is normally seen with Code Red.

2.3 Probability the source address was spoofed:

I think that it is more likely in this case that the attacking host was in turn infected by another compromised system and went searching for other systems to infect. That is one of the attack vectors for this piece of code that will be detailed below in description of the attack.

A look at www.dshield.org showed that there were no other complaints against this IP address.

The Google search-engine search of the host name showed that this is a webserver holding information for a band in Ireland at www.theafterglow.com.

Further research with Sam Spade showed that this serve’s ISP is the British broadband company called NTL.

2.4 Description of attack:

This is a well known buffer overflow attack against IIS servers. The attack utilizes a weakness in the idq.dll ISAPI extension.

The Common Vulnerabilities and Exposures List (CVE) gives the following description of the attack:

“CVE-2001-0500
CVE Version: 20030402

This is an entry on the CVE list, which standardizes names for security problems. It was reviewed and accepted by the CVE Editorial Board before it was added to CVE.

Name	CVE-2001-0500
Description	Buffer overflow in ISAPI extension (idq.dll) in Index Server 2.0 and Indexing Service 2000 in IIS 6.0 beta and earlier allows remote attackers to execute arbitrary commands via a long argument to Internet Data Administration (.ida) and Internet Data Query (.idq) files such as default.ida, as commonly exploited by Code Red.

References

- BUGTRAQ:20010618 All versions of Microsoft Internet Information Services, Remote buffer overflow (SYSTEM Level Access)
- MS:MS01-033
- CERT:CA-2001-13
- BID:2880
- XF:iis-isapi-idq-bo(6705)
- CIAC:L-098 “

2.5 Attack mechanism:

As part of its installation process, IIS installs several ISAPI extensions -- .dlls that provide extended functionality. Among these is idq.dll, which is a component of Index Server (known in Windows 2000 as Indexing Service) and provides support for administrative scripts (.ida files) and Internet Data Queries (.idq files).

A security vulnerability results because idq.dll contains an unchecked buffer in a section of code that handles input URLs. An attacker who could establish a web session with a server on which idq.dll is installed could conduct a buffer overrun attack and execute code on the unpatched web server. Idq.dll runs in the System context, so exploiting the vulnerability would give the attacker complete control of the server and allow him to take any desired action on it.

The buffer overrun occurs before any indexing functionality is requested. As a result, even though idq.dll is a component of Index Server/Indexing Service, the service would not need to be running in order for an attacker to exploit the vulnerability. As long as the script mapping for .idq or .ida files were present, and the attacker were able to establish a web session, he could exploit the vulnerability.

Microsoft Security Bulletin MS01-033 6/18/2001
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

2.6 Correlations:

From eEye (eEye) who were the first to report the vulnerability to Microsoft we see the following explanation of the attack:
http://www.eeye.com/html/Research/Advisories/AD20010618.html

> “We investigated the vulnerability further and found that the .ida ISAPI filter was susceptible to a typical buffer overflow attack.
>
> Example:
> GET /NULL.ida?[buffer]=X HTTP/1.1
> Host: werd

> Where [buffer] is aprox. 240 bytes.
>
> The Exploit, as taught by Ryan "Overflow Ninja" Permeh:
>
> This buffer overflows in a wide character transformation operation. It takes the ASCII (1 byte per char) input buffer and turns it into a wide char/unicode string (2 bytes per char) byte string. For instance, a string like AAAA gets transformed into \0A\0A\0A\0A. In this transformation, buffer lengths are not checked and this can be used to cause EIP to be overwritten."

Symantec gave additional information on the worm and remediation procedures. (Symantec: CodeRed Worm)
http://securityresponse.symantec.com/avcenter/security/Content/2001_07_31.html

GIAC network detect submitted by Samuel Adams (Adams, Samuel)
http://cert.uni-stuttgart.de/archive/intrusions/2003/06/msg00000.html

2.7 Evidence of active targeting:

Active Targeting implies taking steps to target a specific series of hosts. The only traffic from the attacking party was to this IP address. So in this case it appears this is an active attack against this host vice a blind shotgun approach.

2.8 Severity:

Severity = (Criticality + Lethality) – (System countermeasures + network countermeasures)

Criticality in the case appears to be 1. The system does not appear to be one of the major web servers for the site. It is not listed in the DNS tables for the site. At this juncture we do not have enough information to say with all certainty that it is not a vital internal system but there is not really anything pointing to that.

Lethality in this case is 1. A check through the trace with Ethereal filtering for the victimized system using the following filter produced no other entries.
ip.addr == 170.129.168.122

If the buffer overflow exploit had worked and the system was infected we would expect it to seek out other systems to exploit. This is not the case here.

System Countermeasures in this case are a 5. The system did not appear to be infected even though the attack got through the network to it. Either the system is properly patched against the exploit or is not at that location. We do not see any other traffic from or to that system so cannot say with any certainty it is there.

Network Countermeasures are a 2. They company is at least logging network traffic but there is no evidence from this trace that they are actually monitoring the traffic. The traffic appears to be getting through their perimeter network into the internal network.

Severity = [(1 +1) – (5+2)] = Negative 5

The attack does not appear to be that severe and does not appear to be compromising any internal systems.

2.9 Defensive recommendation:
I would recommend layered protection against these sorts of attacks.
The primary process for protecting against this type of attack is to ensure that web servers are properly patched. A combination of automated patch management and network scanning can help determine which systems need to be patched.

In addition because of the frequency of new patches and variations to attack it is important to have another line of defense on the host. This can take the form of a host based firewall or intrusion prevention system. Against this form of attack probably the intrusion prevention system is more effective. A web server will need to service HTTP requests so attacks capitalizing on this will have a greater likelihood of reaching the system. An intrusion prevention system can prevent buffer overflows from occurring and keep the system protected until a patch or time to patch is made available.

Moving out into the network, some form of Network Intrusion Detection should be performed. There is an ongoing debate about the effectiveness of network IDS versus network intrusion prevention. Gartner recently published articles in favor of IP over IDS that have only fanned the flames of the debate but the jury is still out on this subject. I think which ever system is used should have some correlation with other devices to protect against these attacks and help coordinate defenses.

Moving outward, the network should utilize some of the newer IOS code sets from Cisco with new commands to identify and block Code Red and Nimda attacks. (Cisco. Using Network-Based Application Recognition and ACLs for Blocking the "Code Red" Worm)
Finally, if a firewall is not in front of the router, which is difficult to tell it is, the firewall rulebase should be configured to protect against these attacks. Many firewalls including Checkpoint Firewall 1, Symantec Gateway Security, and Cisco PIX all have methods of protecting against these types of attacks.

That said it is important to have layers of protection in case the attackers can get by one system. The original code red which this attack resembles used an excess number of “N” to flood the buffer. The next code red variant Code Red II used excess number of “X” to get by those systems utilizing a very rigid signature to identify the attack.

2.10 Multiple choice test question:

Look at the following list of TCP flags. Which is a valid setting of the flags? Choose all that apply.

	U A P R S F
a.	0 1 0 0 0 1
b.	0 1 1 1 1 1
c.	0 1 0 0 1 0
d.	0 0 0 0 1 1

The correct answers are “a” and “c”. SYN/ACKs and ACK/FINs are normal parts of the communication process. Christmas trees with all flags activated or SYN/FIN statements which tell a device to both start and stop a session are invalid (Stevens, p230).

3.0 network detect – Scan Squid Proxy Attempt

3.1 Source of Trace.

The source of this detect was my home network DMZ. The network is a simple 3 layer design with a stateful firewall, network IDS, and host based firewalls with intrusion detection as well. The firewall was configured to send all network traffic to the IDS sensor and there are no other systems providing other services for Internet hosts. The setup is not listed in any DNS tables at the broadband ISP.

3.1 Detect was generated by:

The detect was generated by Eagle X from Engage Security www.engagesecurity.com. Eagle X is a new package for Snort that bundles Snort 2.0, with a MySQL database, Apache Web server, and ACID plugins that can be installed and run on Windows platforms.

The IDS returned the following alert data:

Meta	ID #	Time	Triggered Signature
	3 - 203	2003-06-22 08:59:37	[snort] SCAN Squid Proxy attempt

Sensor	name	interface	filter
	My Net	My net	*none*

Alert Group	*none*

IP

source addr	dest addr	Ver	Hdr Len	TOS	length	ID	flags	offset	TTL	chksum
80.181.181.203	My.Net..109	4	5	0	64	58781	0	0	102	23940

FQDN	Source Name	Dest. Name
	host203-181.pool80181.interbusiness.it	My.Net.org

Options	*none*

TCP

source port	dest port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	seq #	ack	offset	res	window	urp	chksum
1632	3128							X		2645478645	0	11	0	65535	0	59179

Options		code	length	data
	#1	MSS	2	05B4
	#2	NOP	0	
	#3	WS	1	03
	#4	NOP	0	
	#5	NOP	0	
	#6	TS	8	0000000000000000
	#7	NOP	0	
	#8	NOP	0	
	#9	SACKOK	0	

Payload	*none*

I received six of these alerts over a 4 hour period.

Displaying alerts 1-6 of 6 total

□	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
□	#0-(3-203)	[snort] SCAN Squid Proxy attempt	2003-06-22 08:59:37	80.181.181.203:1632	My.net 3128	TCP
□	#1-(3-196)	[snort] SCAN Squid Proxy attempt	2003-06-22 04:46:27	80.181.181.203:2841	My.net:3128	TCP
□	#2-(3-197)	[snort] SCAN Squid Proxy attempt	2003-06-22 04:46:28	80.181.181.203:2841	My.net:3128	TCP
□	#3-(3-198)	[snort] SCAN Squid Proxy attempt	2003-06-22 04:46:29	80.181.181.203:2841	My.net:3128	TCP
□	#4-(3-201)	[snort] SCAN Squid Proxy attempt	2003-06-22 08:59:36	80.181.181.203:1632	My.net:3128	TCP
□	#5-(3-202)	[snort] SCAN Squid Proxy attempt	2003-06-22 08:59:37	80.181.181.203:1632	My.net:3128	TCP

3.3 Probability the source address was spoofed:

It is unlikely that the IP address is spoofed because this is a reconnaissance type of attack and the attacker needs to receive the information back. However, the proxy can be taken over in certain circumstances so that does not necessarily mean the attacker is at the source IP address.

This source IP address 80.181.181.203 belongs to host203-181.pool80181.interbusiness.it

According to the Sam Spade web site, this site belongs to a telecom company in Italy.

```
domain:      interbusiness.it
x400-domain: c=it; admd=0; prmd=interbusiness;
org:         Telecom Italia S.p.A.
descr:       InterBusiness
descr:       Network Service Provider
admin-c:     CD2-ITNIC
tech-c:      FG82-ITNIC
tech-c:      GLM2-ITNIC
postmaster:  FG82-ITNIC
zone-c:      DRS9-ITNIC
nserver:     151.99.125.2 dns.interbusiness.it
nserver:     193.205.245.66 dns3.nic.it
nserver:     151.99.250.2 server-b.cs.interbusiness.it
nserver:     151.99.125.138 dns.opb.interbusiness.it
remarks:     Fully Managed
remarks:     Please report Spam/Abuse only to abuse@interbusiness.it
mnt-by:      INTERBUSINESS-MNT
created:     before 19960129
expire:      20040129
changed:     domain@cgi.interbusiness.it 20020426
source:      IT-NIC

person:      Camillo Di Vincenzo
address:     Telecom Italia S.P.A.
address:     Via Paolo Di Dono, 44
address:     I-00143 Roma
address:     Italy
phone:       +39 06 36871
fax-no:      +39 06 36871
nic-hdl:     CD2-ITNIC
changed:     domain@cgi.interbusiness.it 20001115
changed:     hostmaster@nic.it 20030424
changed:     hostmaster@nic.it 20030428
source:      IT-NIC
```

3.4 Description of attack:

From www.snort.org SID 618 this signature meets the requirements for a reconnaissance scan looking for network information. It is used for information gathering and is not an overt take if of itself.

3.5 Attack mechanism:

The attack stems from the attacker trying to determine if the victim is utilizing ports 21 or 20 and if ports 21 and 20 are being used for FTP. Once that piece of foot printing is accomplished the attacker can break out the FTP specific attacks to compromise a host.
Squid is a proxy-caching server that has a known bug when run in its accelerated mode. Under a certain configuration it will ignore ACL or access lists. Attackers can capitalize on this by using the proxy server to proxy their own attacks.

3.6 Correlations:

Dshield.org did not report any records against this IP address.
Tony Adams ran across a similar detect note in his practical. (Adams, Tony)
www.giac.org/practical/Tony_Adams.doc

Paul Nasrat of SecuriTeam discusses the ACL bug in his whitepaper (Nasrat, Paul)
http://www.securiteam.com/unixfocus/5LP0P0U4UQ.html

Description of the Squid Web Proxy Cache (Squid-Cache)
http://www.squid-cache.org/

GIAC Practical by Eric Galarneau on how to secure a Squid proxy server (Galarneau, Eric)
http://www.sans.org/rr/papers/50/1048.pdf

Stephen Northcutts paper on Ring Zero which helped to narrow down that this was not a Ring Zero attack due to the lack of an exe in the payload. (Northcutt, Stephen, Ring Zero)
http://www.sans.org/resources/idfaq/ring_zero.php

3.7 Evidence of active targeting:

By the nature of this attack it is an active target to determine if further FTP vulnerabilities exits. It will seek out areas that it determines are vulnerable to the attack.

3.8 Severity:

Severity = (Criticality + Lethality) – (System countermeasures + network countermeasures)

Criticality is 4. This is one of the major systems on my network.

Lethality is 1. This is merely a reconnaissance of the network and not a potentially crippling attack on its own.

System countermeasures are 4. The system is well patched, a host firewall is in place, and the system is not running the FTP service.

Network countermeasures are 3. There is a stateful firewall in place and network intrusion detection. However the attack did get through the perimeter into the DMZ.

Severity = [(4 + 1) – (4 + 3)] = Negative 2.

The system admin needs to be aware of attack but the network does not appear to be in any immediate danger from this attack or its antecedents.

3.9 Defensive recommendation:

The perimeter could be tighter on this network. The attack did make it through the firewall. Also the firewall is not configured to report to www.dShield.org . If it was maybe this attacker would be listed in their database and Dshield.org could inform the system admins at the attacking site to correct their system.

Better event correlation between the IDS and the firewall could help. If the firewall rulebase could be changed by the IDS that could help decrease the number of attacks to parse and prevent future exposure.

3.10 Multiple choice test question:

What Windump filter would you use to identify FTP traffic on your network? (Choose two)

a. udp[3]=20
b. udp[3]=21
c. tcp[3]=21
d. tcp[3]=20
e. udp[3]=23

Answer: c and d.

Explanation: FTP traffic is a part of the TCP protocol that uses port 21 for control and port 20 for data.

Section III – Analyze This Scenario

Executive Summary:

In this section we will cover the events of a 5-day span at a University. The dates that I chose were May 8-12, 2003 which covered both weekday and weekend activities.

The results were illuminating. The University was subject to a multitude of attacks and scans. There were several compromised systems internally with substantial Trojan and worm activity.

The University received 1,213,063 events in the Alert log at that time with an average of 202,177 events per day. The data mainly covered May 8-12 but there was some overlap into May13 for the last log.

Most Universities have a very open stance towards network traffic. This appeared to be the case here but there was evidence in the custom rules written that the University was attempting to head off major network outages by looking for Code Red and other worm activity.

Also please note that I use the term “The University” to describe the university providing the logs for this assignment. During the analysis I found that other universities have suspicious traffic destined to this university. Their specific names are used during the analysis.

Files chosen for Analysis

I chose 5 days of scans from the www.incidents.org/logs directory in accordance with the GCIA practical instructions. No prior evaluation was used in choosing these files. They were chosen at random as a 5 day block. However I did choose to include the weekend as well to see how that affected the data grouping.

Scans	Alerts	OOS *
scans.030508	alert.030508	OOS_Report_2003_05_09_1240.txt
scans.030509	alert.030509	OOS_Report_2003_05_10_1240.txt
scans.030510	alert.030510	OOS_Report_2003_05_11_1240.txt
scans.030511	alert.030511	OOS_Report_2003_05_12_1240.txt
scans.030512	alert.030512	OOS_Report_2003_05_13_1240.txt

(* Note that the OOS Report logs are 1 day delayed from the Scan and Alert logs)

Relational Analysis of Systems generating logs

There is substantial overlap between the logs. In the analysis I tend to focus on the Alert logs as those seem to be of the highest priority and utilize the Scan and OOS data to obtain a more granular view.

Analyzing the University data is problematic mainly due to the fact that the analyst is not aware of the security policies for the university in question. Coming up with a quantitative or even qualitative risk analysis will be very difficult without a frame of reference of what “hurts” the University. We do not have an intrinsic or dollar value of systems and network availability. But that does not mean we cannot come up with good information of a qualitative nature. We just need to make some assumptions. Looking at the boundary conditions of the problem we can come up with some good assumptions:

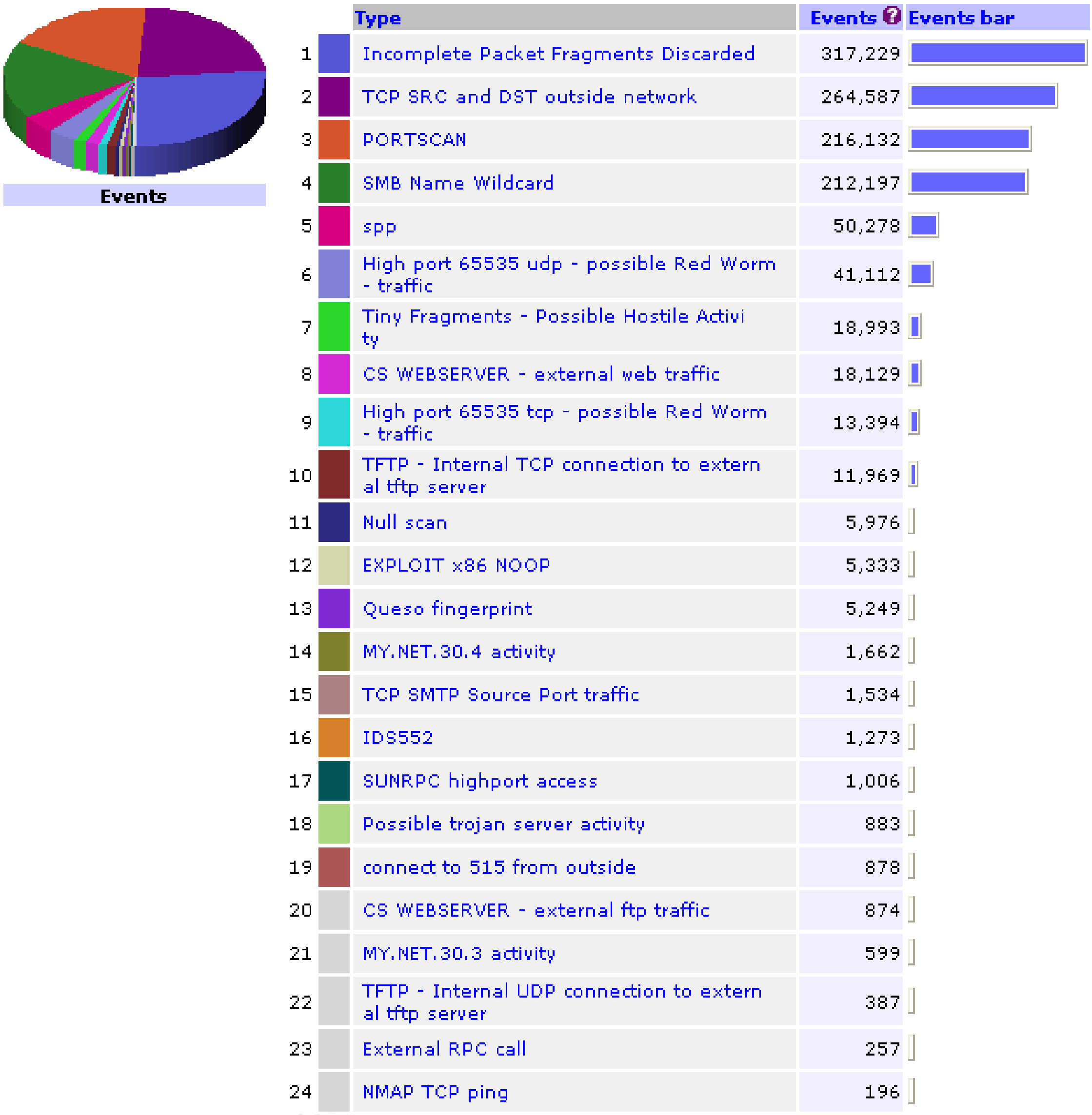
- The network needs to be up and available. Even in the most open door environment, and some universities place no restrictions on traffic, the network has to be there. If we see examples of traffic that could degrade or shut down network communications that will be labeled as bad.

- Internal systems that appear to be compromised are bad. There might not be any valuable data on that system but that system could be used as a springboard to other attacks.
- Crafted packets that are designed to cause systems to break are bad. Most of the portscans like those seen with NMAP and other tools do some sort of packet crafting generally in the TCP flags section to elicit a response or to evade network controls. Those are by nature reconnaissance efforts and less dangerous then a direct attack against a system.
- Stealthy activity scan activity is undesired but unless actual damage to the network or host is evident it is of less concern. We need to be aware of when it is happening and determine if there is a direct relationship between the covert activity and a more damaging attack.

As I said, there is substantial overlap in the logs. The Alert logs had data both of alerts and also of scans. However, the Scan logs provided more detailed data so we will first look at the Alert log data and get an overall picture of alerts. We will then look into the Scan data to get a more granular picture of that area. Similar to the Alert and Scan log, the OOS Reports have more granular data on particular scans.

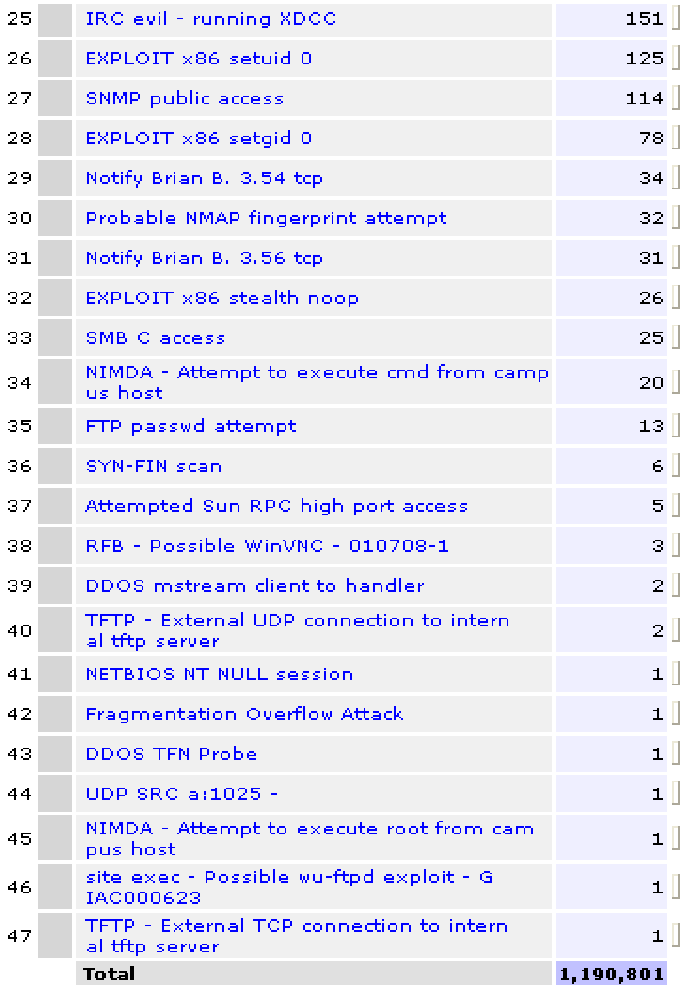
Alert log review:

During the period of May 08 – 12 the University experienced roughly 1,190,801 hits of events that triggered alerts of the Snort IDS sensors. This estimate is rough because there was substantial corruption to the logs and when normalizing the data some of those corrupt hits were removed. However we are definitely in the ball park with that figure.



Events

		Type	Events	Events bar
1	🟦	Incomplete Packet Fragments Discarded	317,229	
2	🟪	TCP SRC and DST outside network	264,587	
3	🟧	PORTSCAN	216,132	
4	🟩	SMB Name Wildcard	212,197	
5	🟥	spp	50,278	
6	🟦	High port 65535 udp - possible Red Worm - traffic	41,112	
7	🟩	Tiny Fragments - Possible Hostile Activity	18,993	
8	🟪	CS WEBSERVER - external web traffic	18,129	
9	🟦	High port 65535 tcp - possible Red Worm - traffic	13,394	
10	🟥	TFTP - Internal TCP connection to external tftp server	11,969	
11	🟦	Null scan	5,976	
12	⬜	EXPLOIT x86 NOOP	5,333	
13	🟪	Queso fingerprint	5,249	
14	🟫	MY.NET.30.4 activity	1,662	
15	🟫	TCP SMTP Source Port traffic	1,534	
16	🟧	IDS552	1,273	
17	🟩	SUNRPC highport access	1,006	
18	🟩	Possible trojan server activity	883	
19	🟥	connect to 515 from outside	878	
20	⬜	CS WEBSERVER - external ftp traffic	874	
21	⬜	MY.NET.30.3 activity	599	
22	⬜	TFTP - Internal UDP connection to external tftp server	387	
23	⬜	External RPC call	257	
24	⬜	NMAP TCP ping	196	



25	IRC evil - running XDCC	151
26	EXPLOIT x86 setuid 0	125
27	SNMP public access	114
28	EXPLOIT x86 setgid 0	78
29	Notify Brian B. 3.54 tcp	34
30	Probable NMAP fingerprint attempt	32
31	Notify Brian B. 3.56 tcp	31
32	EXPLOIT x86 stealth noop	26
33	SMB C access	25
34	NIMDA - Attempt to execute cmd from campus host	20
35	FTP passwd attempt	13
36	SYN-FIN scan	6
37	Attempted Sun RPC high port access	5
38	RFB - Possible WinVNC - 010708-1	3
39	DDOS mstream client to handler	2
40	TFTP - External UDP connection to internal tftp server	2
41	NETBIOS NT NULL session	1
42	Fragmentation Overflow Attack	1
43	DDOS TFN Probe	1
44	UDP SRC a:1025 -	1
45	NIMDA - Attempt to execute root from campus host	1
46	site exec - Possible wu-ftpd exploit - GIAC000623	1
47	TFTP - External TCP connection to internal tftp server	1
	Total	**1,190,801**

The list above shows the alert detections for the 5 day time period. We will initially look at the top alerts by virtue of quantity. Being the most of numerous of alerts these are at least the attacks that causing the most log noise. We would need to correlate this with other network traffic analysis tools to see how much bandwidth is actually being degraded.

Detect: Incomplete Packet Fragments Discarded

This alert is not triggered by an actual rule but by a preprocessor within Snort. We do not know what version of Snort the University is using but there have been times in the past with version 1.8.2 with the defrag processor throwing up a lot of these alerts

References:
The following reference from Marty Roesch details a similar situation (Roesch, Marty):
http://www.mcabee.org/lists/snort-users/Nov-01/msg00820.html

Glen Larratt's practical shows similar information (Larratt, Glen)
http://is.rice.edu/~glratt/practical/Glenn_Larratt_GCIA.html

Dragos Ruiu who wrote the preprocessor gave an explanation of its alerts:
http://archives.neohapsis.com/archives/snort/2001-02/0320.html

> "This message is given by the defragmentation preprocessor when packets bigger than 8k that are more than half empty when the last fragment is received are discarded.
>
> This can be caused by:
> - transmission errors
> - broken stacks
> - and fragmentation attacks "

Recommendation:
Examine the version of preprocessor used and update to the latest fragment preprocessor. This could be a fragmentation attack or could be just be transmission errors. Regardless, these incidents are flooding the logs and need to be tracked down and screened if it is just noise.

Detect: TCP SRC and DST outside network

This alert is generated by the Stream4 preprocessor that does stream reassembly. Dragos Ruiu details the functionality of the Stream 4 preprocessor as:

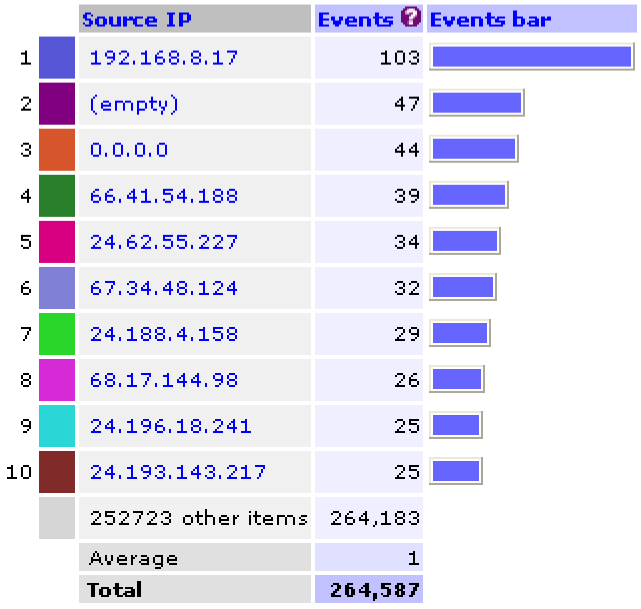
> "Stream4 is an entirely new preprocessor that performs two functions:
>
> 1) Stateful inspection of TCP sessions
> 2) TCP stream reassembly

There is a recent vulnerability to Snort that an attacker could be trying to exploit. SecuriTeam discusses this on their paper regarding vulnerabilities in Snort Preprocessors (Securiteam, Multiple Vulnerabilities in Snort Preprocessors)
http://www.securiteam.com/securitynews/5FP0A2A9QI.html

> **"VU#139129 - Heap overflow in Snort "stream4" preprocessor (CAN-2003-0029)**
> Researchers at CORE Security Technologies have discovered a remotely exploitable heap overflow in the Snort "stream4" preprocessor module. This module allows Snort to reassemble TCP packet fragments for further analysis.
>
> To exploit this vulnerability, an attacker must disrupt the state tracking mechanism of the preprocessor module by sending a series of packets with crafted sequence numbers. This causes the module to bypass a check for buffer overflow attempts and allows the attacker to insert arbitrary code into the heap."

This vulnerability affects Snort versions 1.8.x, 1.9.x, and 2.0 prior to RC1. Snort has published an advisory regarding this vulnerability; it is available at http://www.snort.org/advisories/snort-2003-04-16-1.txt.

Recommendations:
If we look at which are the most prevalent Source addresses that are generating this alert we see and internal address of a non routable address of 192.168.8.17 being the top offender.



		Source IP	Events	Events bar
1	🟦	192.168.8.17	103	
2	🟪	(empty)	47	
3	🟧	0.0.0.0	44	
4	🟩	66.41.54.188	39	
5	🟥	24.62.55.227	34	
6	🟦	67.34.48.124	32	
7	🟩	24.188.4.158	29	
8	🟪	68.17.144.98	26	
9	🟦	24.196.18.241	25	
10	🟫	24.193.143.217	25	
	⬜	252723 other items	264,183	
		Average	1	
		Total	**264,587**	

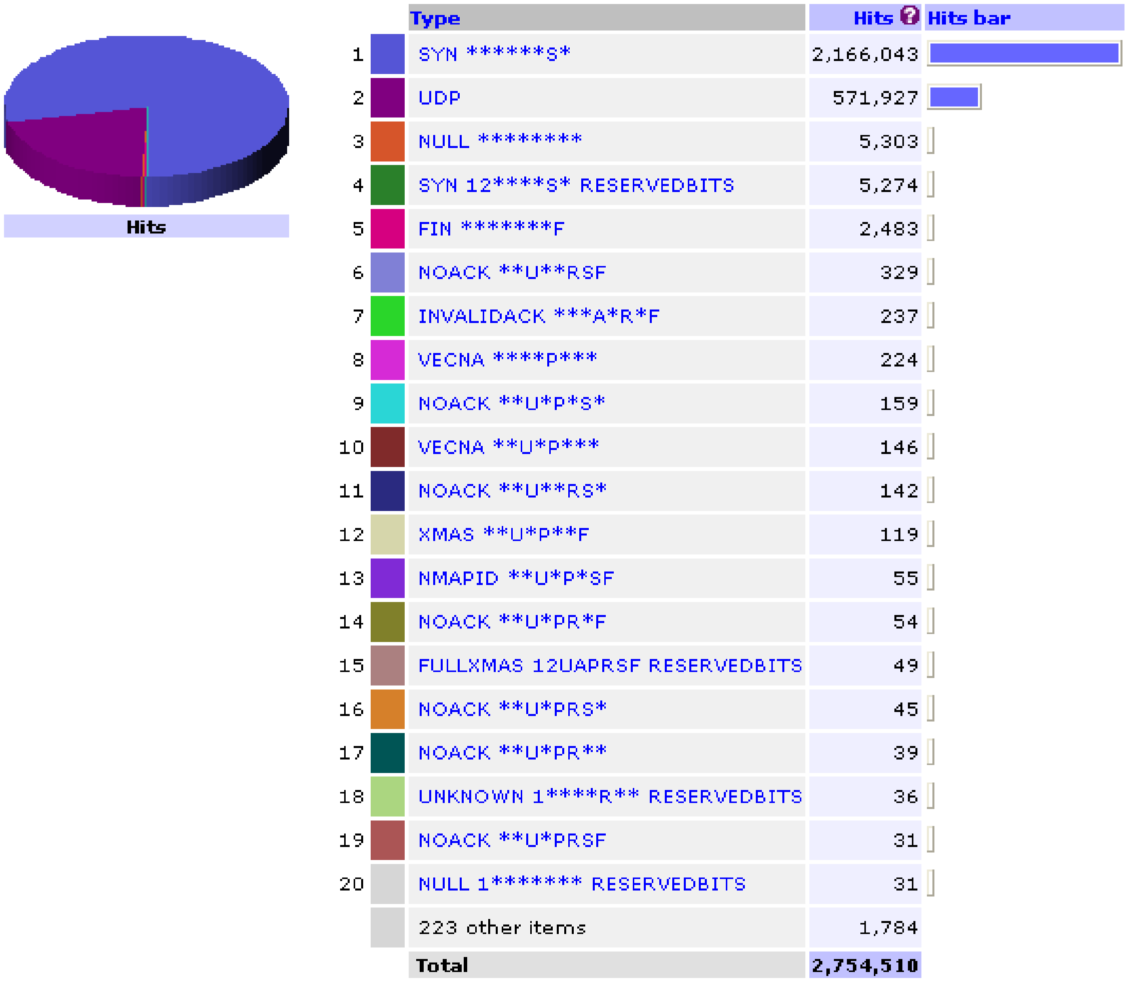
One recommendation to the University analysts is take a hard look at this address. If that range is not a valid internal address then there is a greater likelihood that someone is spoofing the address and attempting the exploit. There is always the possibility, especially in this university setting, for someone to attempt the attack internally as well. Also would recommend that the University upgrade their snort sensors to Snort 2.0 which is not susceptible to the exploit.

References:
Dragos Ruiu gives details on the preprocessor generating this alert. (Ruiu, Dragos)
http://www.snort.org/docs/faq.html#3.14

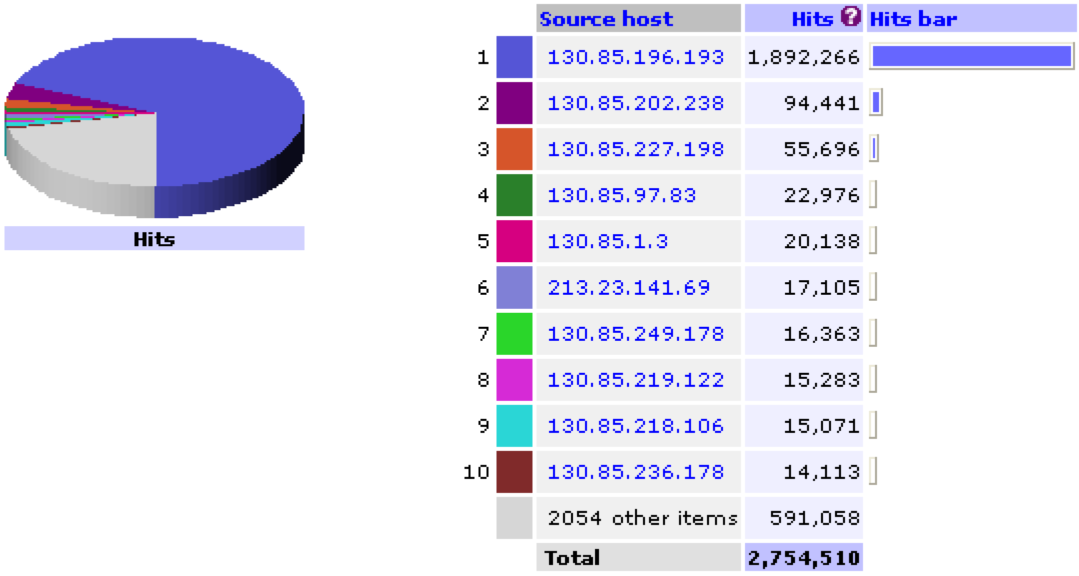
SecuriTeam gives further details of the exploit: (SecuriTeam)
http://www.securiteam.com/securitynews/5FP0A2A9QI.html

Detect: Portscan

There was substantial scanning activity recorded during this time frame. Portscans fall in the realm of reconnaissance and are all too common an event. The problem here is that so many hits for scanning are coming in that they are flooding the event log and perhaps masking more important information by sheer volume. Of special note is the number of SYN scans. In this case it appears that an attacker is trying to perform a denial of service attack known as SYN flooding. By capitalizing on the TCP handshaking process where a system sends a SYN packet to start communications, the remote host expends some resources processing the attempt to communicate. By sending a flood of SYN packets the remote host will soon expend all of its resources attempting to deal with the communication attempts. Below we see the scans and 1 host 130.85.196.193 performing a SYN flood against



		Type	Hits	Hits bar
1	🟦	SYN ******S*	2,166,043	
2	🟪	UDP	571,927	
3	🟧	NULL ********	5,303	
4	🟩	SYN 12****S* RESERVEDBITS	5,274	
5	🟥	FIN *******F	2,483	
6	🟦	NOACK **U**RSF	329	
7	🟩	INVALIDACK ***A*R*F	237	
8	🟪	VECNA ****P***	224	
9	🟦	NOACK **U*P*S*	159	
10	🟫	VECNA **U*P***	146	
11	🟦	NOACK **U**RS*	142	
12	🟨	XMAS **U*P**F	119	
13	🟪	NMAPID **U*P*SF	55	
14	🟫	NOACK **U*PR*F	54	
15	🟫	FULLXMAS 12UAPRSF RESERVEDBITS	49	
16	🟧	NOACK **U*PRS*	45	
17	🟩	NOACK **U*PR**	39	
18	🟩	UNKNOWN 1****R** RESERVEDBITS	36	
19	🟥	NOACK **U*PRSF	31	
20	⬜	NULL 1******* RESERVEDBITS	31	
	⬜	223 other items	1,784	
		Total	**2,754,510**	



		Source host	Hits	Hits bar
1	■	130.85.196.193	1,892,266	▬
2	■	130.85.202.238	94,441	▬
3	■	130.85.227.198	55,696	▬
4	■	130.85.97.83	22,976	▬
5	■	130.85.1.3	20,138	▬
6	■	213.23.141.69	17,105	▬
7	■	130.85.249.178	16,363	▬
8	■	130.85.219.122	15,283	▬
9	■	130.85.218.106	15,071	▬
10	■	130.85.236.178	14,113	▬
	■	2054 other items	591,058	
		Total	**2,754,510**	

A look at Dshield.org shows this address is a well known address at the University of Maryland. A fightback message has been sent to the administrator for that account during the time of this attack. It appears that in this case the University is sending information to Dshield.org.

IP Address: 130.85.196.193

HostName: 130.85.196.193

DShield Profile:

Country:	US
Contact E-mail:	jack@UMBC.EDU
Total Records against IP:	1104
Number of targets:	368
Date Range:	2003-05-13 to 2003-05-13

Summary was recently updated.

Top 10 Ports hit by this source:

Port	Attacks	Start	End

Last Fightback Sent: sent to jack@UMBC.EDU on 2003-05-13 17:44:29

Whois:

OrgName: University of Maryland Baltimore County
OrgID: UMBC
Address: UMBC University Computing
City: Baltimore
StateProv: MD
PostalCode: 21250

```
Country:    US

NetRange:   130.85.0.0 - 130.85.255.255
CIDR:       130.85.0.0/16
NetName:    UMBCNET
NetHandle:  NET-130-85-0-0-1
Parent:     NET-130-0-0-0-0
NetType:    Direct Assignment
NameServer: UMBC5.UMBC.EDU
NameServer: UMBC4.UMBC.EDU
NameServer: UMBC3.UMBC.EDU
Comment:
RegDate:    1988-07-05
Updated:    2000-03-17

TechHandle: JJS41-ARIN
TechName:   Suess, John J.
TechPhone:  +1-410-455-2582
TechEmail:  jack@umbc.edu

# ARIN WHOIS database, last updated 2003-05-21 20:10
```

Recommendation:
A firewall or router with the proper operating system and access list could help cut down the number of scans that reach internally. We do not know the University of Maryland layout. If the students are not assigned static IP addresses it might be difficult to determine who is actually performing the SYN flood. Also many universities have open or unsecured wireless access points on campus that are good targets of opportunities for attackers.
Cooperation between the attacking site and the attacked site is a key factor at this point. The source address in this case remained the same so a change in the access lists of the firewall or router could block the traffic. But if the attacker logs on at a different point and receives another IP or spoofs the IP address it might be a continual chase. Working with the other university could assist tracking down offenders and areas of weak security.

Detect: SMB Name Wildcard

The alert is linked to NetBIOS activity. Prior to 2000 it was mostly attributed to Windows systems looking for NetBIOS resources. After the spring of 2000 there were reports of port 137 scans and possible worm activity.

Recommendations:
One recommendation from the Snort.org FAQ list is that since this could be just internal Windows systems configure the rule so that the source address is not the home network IP address. NetBIOS traffic should not be coming in externally to the network. I would recommend that the University admins block at the firewall NETBIOS traffic from external sources. Often users will build new Windows

systems without taking precautions to lock them down. Cutting off efforts to take advantage from the outside should decrease the chance of compromise.

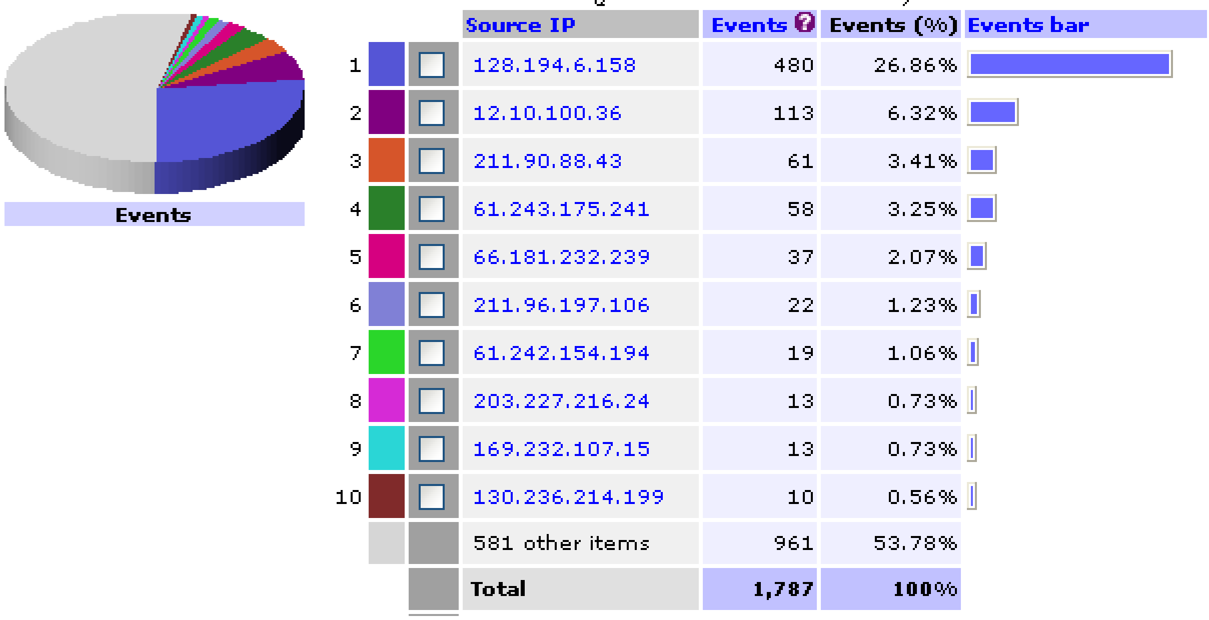
Reference:
Daniel Martin shows how Windows Explorer often generates this alert. (Martin, Daniel. Spoofed SMB Name Wildcard probes.)
http://lists.jammed.com/incidents/2001/05/0034.html

Bryce Alexander writes in a whitepaper at SANs detailing Port 137 scans that have similar behaviour (Alexander, Bryce)
http://www.sans.org/resources/idfaq/port_137.php

Chris Green at www.Snort.org discusses SMB Name Wildcard alerts on a FAQ list. (Green, Chris)
http://www.snort.org/docs/4.15

Detect: SPP

My database lumped anything to do with SPP with one alert.   SPP is the acronym for the Snort Portscan Preprocessor.   The alert can also be tied to the Snort DNS preprocessor.   Firewalls often allow DNS traffic through unregulated and it is possible to use that as a vector for Trojans or other traffic. The plugin will watch the DNS traffic to make sure that it is the correct format and also make sure that only a single response is occurring. We do not have a lot of information on what sort of SPP event is occurring.



			Source IP	Events	Events (%)	Events bar
1	■	□	128.194.6.158	480	26.86%	
2	■	□	12.10.100.36	113	6.32%	
3	■	□	211.90.88.43	61	3.41%	
4	■	□	61.243.175.241	58	3.25%	
5	■	□	66.181.232.239	37	2.07%	
6	■	□	211.96.197.106	22	1.23%	
7	■	□	61.242.154.194	19	1.06%	
8	■	□	203.227.216.24	13	0.73%	
9	■	□	169.232.107.15	13	0.73%	
10	■	□	130.236.214.199	10	0.56%	
	■		581 other items	961	53.78%	
			Total	**1,787**	**100%**	

A look at Dhield.org for the top source host shows that no report was made against this host but that it came from another university, Texas A & M.

IP Address: 128.194.6.158

HostName: 128.194.6.158

DShield Profile:

Country:	US
Contact E-mail:	tech@net.tamu.edu
Total Records against IP:	not processed
Number of targets:	select update below
Date Range:	to

Update Summary

Top 10 Ports hit by this source:

Port	Attacks	Start	End

Last Fightback Sent: not sent

Whois:

```
OrgName:    Texas A&M University
OrgID:      TAMU
Address:    System Data Services Network
Address:    Computing and Information Services
City:       College Station
StateProv:  TX
PostalCode: 77843-3142
Country:    US

NetRange:   128.194.0.0 - 128.194.255.255
CIDR:       128.194.0.0/16
NetName:    TAMU-NET
NetHandle:  NET-128-194-0-0-1
Parent:     NET-128-0-0-0-0
NetType:    Direct Allocation
NameServer: DNS.TAMU.EDU
NameServer: DNS2.TAMU.EDU
NameServer: DNS3.TAMU.EDU
NameServer: AURORA.LATECH.EDU
Comment:
RegDate:    1987-01-16
Updated:    1997-09-30

TechHandle: NG16-ORG-ARIN
TechName:   Texas A&M University
TechPhone:  +1-979-862-2222
TechEmail:  tech@net.tamu.edu

# ARIN WHOIS database, last updated 2003-06-29 21:05
```

Recommendations:
Continue to monitor and attempt to tune the rules in order to have less false positives. If continued activity occurs from this address then the system administrators should work with the other University to ferret out the offenders.

References:
Details on the Snort DNS Preprocessor by Axonpotential (Axonpotential)

Andy Millican's GSEC practical had good details on the SPP plugin. (Millican, Andy, Practical)
http://www.giac.org/practical/GSEC/Andy_Millican_GSEC.pdf

Snort Preprocessor Plugin Source File Template
http://scorpions.net/~fygrave/snort/templates/spp_template.c

Top Alerts by virulence

This is a look at the top Alerts as far as quantity goes. We should also examine the data for especially damaging events with an eye towards compromised systems.

The following 5 detects are listed in order of priority

Notify Brian B 3.54 tcp

Possible Trojan Server Activity

TFTP - Internal TCP connection to External tftp server and TFTP Internal UDP connection to external tftp server

IRC evil - running XDCC

My.Net.30.4 activity

Nimda - Attempt to execute cmd from campus host

Detect Notify Brian B 3.54 tcp and Notify Brian B. 3.56 tcp

These two detects are custom rules to alert an Admin of some sort of dangerous condition or traffic. Since the University admins took the time to write a custom rule alerting for this condition it should be placed at a higher priority. The alerts are similar so I lumped the analysis together.

Samples of traffic that generated the attack are below:

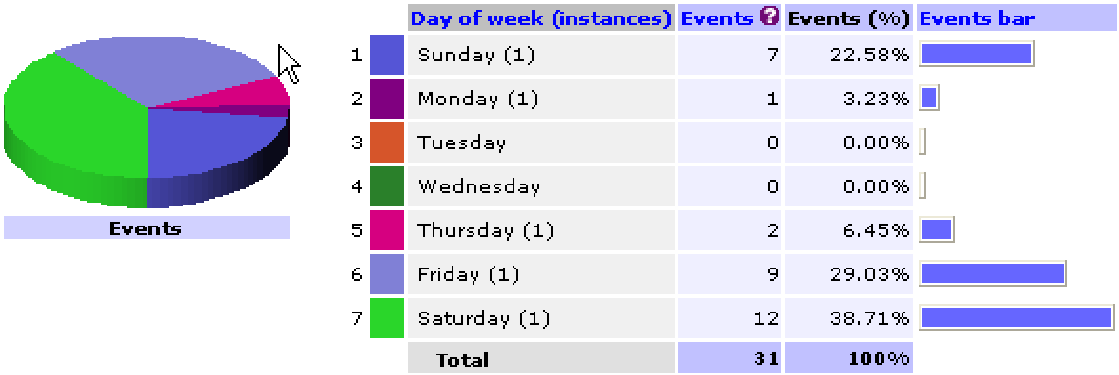
```
05/08-04:53:46.488948  [**] Notify Brian B. 3.54 tcp [**] 80.128.31.66:2166 -> MY.NET.3.54:80
05/08-05:54:23.500154  [**] Notify Brian B. 3.54 tcp [**] 61.53.146.230:1025 -> MY.NET.3.54:139
05/08-07:41:31.721362  [**] Notify Brian B. 3.54 tcp [**] 211.254.169.96:4260 -> MY.NET.3.54:445
05/08-09:36:40.188584  [**] Notify Brian B. 3.54 tcp [**] 211.190.200.134:2786 ->
MY.NET.3.54:445
05/08-13:19:20.413291  [**] Notify Brian B. 3.54 tcp [**] 66.71.1.98:1172 -> MY.NET.3.54:445

05/08-04:00:26.426113  [**] Notify Brian B. 3.56 tcp [**] 218.29.221.184:1025 -> MY.NET.3.56:139
05/09-01:49:33.640050  [**] Notify Brian B. 3.56 tcp [**] 68.154.14.24:2820 -> MY.NET.3.56:445
05/08-07:41:40.824078  [**] Notify Brian B. 3.56 tcp [**] 211.254.169.96:4262 -> MY.NET.3.56:445
```

Port 1025 both tcp and udp are used by Network Blackjack. There are a number of Trojans that bind to that port including NetSpy, Maverick's Matrix and remote storm. However since it is the next port after the last privileged port 1024 there is a possibility that a legitimate application is trying this port. The Microsoft Distributed Transaction Coordinator service, msdtc.exe uses this port as does RFS remote_file_sharing utility. There is a hosting service for SMTP from CommuniLink that will send out traffic via port 1025 to the hosting servers. As the traffic seen here is entering the University I think we can rule that out as an option.

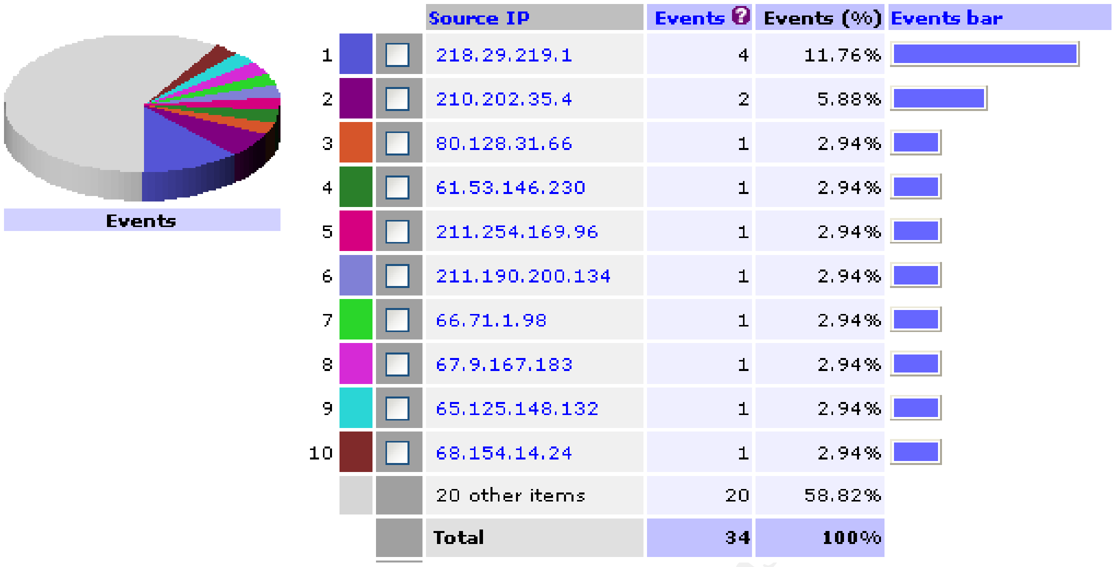
Examining the reports on incidents.org we actually don't see much activity for this attack since the large spike on April 22.   However, I think this rule is left over from that attack possibly because the University was heavily hit.

This attack is interesting because it occurred mostly during the weekend or off hours.



		Day of week (instances)	Events	Events (%)	Events bar
1	🟦	Sunday (1)	7	22.58%	
2	🟪	Monday (1)	1	3.23%	
3	🟧	Tuesday	0	0.00%	
4	🟩	Wednesday	0	0.00%	
5	🟥	Thursday (1)	2	6.45%	
6	🟪	Friday (1)	9	29.03%	
7	🟩	Saturday (1)	12	38.71%	
		Total	31	100%	

Also the following IP addresses were the top source IP addresses that triggered this alert.   A look at DHIELD did not report any submissions on these addresses.



			Source IP	Events	Events (%)	Events bar
1	■	☐	218.29.219.1	4	11.76%	
2	■	☐	210.202.35.4	2	5.88%	
3	■	☐	80.128.31.66	1	2.94%	
4	■	☐	61.53.146.230	1	2.94%	
5	■	☐	211.254.169.96	1	2.94%	
6	■	☐	211.190.200.134	1	2.94%	
7	■	☐	66.71.1.98	1	2.94%	
8	■	☐	67.9.167.183	1	2.94%	
9	■	☐	65.125.148.132	1	2.94%	
10	■	☐	68.154.14.24	1	2.94%	
			20 other items	20	58.82%	
			Total	**34**	**100%**	

Snort.org's easy reference for ports had information on the other ports used.

2166	2166/udp	iwserver	iwserver
2166	2166/tcp	iwserver	iwserver
2786	2786/tcp	aic-oncrpc	aic-oncrpc - Destiny MCD database
2786	2786/udp	aic-oncrpc	aic-oncrpc - Destiny MCD database
2820	2820/udp	univision	UniVision
2820	2820/tcp	univision	UniVision

The iwserver or Image Web Server is a product from Vantive for doing HTML images. I did not see much data stating that this port was being used for malicious purposes. The same could be said for the Database and UniVision products. These are could just be part of the legitimate services for the server and are being lumped in with an over generalized rule.

Recommendations:
The university is already being specifically alerted to this attack. Systems should be patched and configured so they are not susceptible to those trojans that use this vector.

References:
Details of a domain Service that uses port 1025 with SMTP. (CommuniLink, Domain Services)
http://www.communilink.net/en/add_port1025.html

Security focus Archives discuss Network Blackjack on Port 1025 (SecurityFocus, Network Blackjack) http://www.securityfocus.com/archive/105/277275/2002-06-09/2002-06-15/1

Karl Krueger reports that NFS also utilizes this port. (Kruegar, Karl A)
http://cert.uni-stuttgart.de/archive/intrusions/2002/03/msg00014.html

Other services that use this port:
http://www.netsys.com/suse-linux-security/2000-Dec/msg00026.html

With all these other systems utilizing this port it makes sense that an attacker would camouflage their efforts by using a popular channel to perform attacks. Since the ports are popular there is a greater probability of being allowed through the defenses.

Possible Trojan Activity

A sample of some of the alerts from the log shows that this rule is being triggered by traffic to port 27374.

```
05/08-06:56:42.658380  [**] Possible trojan server activity [**] 65.68.62.41:1442 ->
MY.NET.1.209:27374
05/08-06:56:48.712641  [**] Possible trojan server activity [**] 65.68.62.41:1442 ->
MY.NET.1.209:27374
05/08-07:36:32.260152  [**] Possible trojan server activity [**] 24.202.144.208:4219 ->
MY.NET.5.23:27374
05/08-07:36:32.813134  [**] Possible trojan server activity [**] 24.202.144.208:4224 ->
MY.NET.5.28:27374
05/08-07:36:35.781265  [**] Possible trojan server activity [**] 24.202.144.208:4251 ->
MY.NET.5.55:27374
```

Port 27374 is used by the a number of Trojans including Subseven, Bad Blood, and the Saint. It became much more prevalent on the Internet after the Subseven malware, because Subseven causes additional scans from the infected host. A look at Incidents.org shows a spike of target activity during the time frame in question.

References:
Doug Kite’s GCIA practical discusses this alert. (Kite, Doug)

Incidents.org shows trends of this virus activity over time.
http://isc.incidents.org/port_details.html?port=27374

Glock Software shows that other malware such as Bad Blood can talk to Subseven on this port as well probably adding to the prevalence on the Internet. (Glock Software, Bad Blood)

TFTP - External TCP connection to internal tftp server

This appears to be another custom rule for the University. A search at Snort.org showed no rule delivering this specific message. This rule is also associated with a TCP and UDP rule for internal devices attempting to reach external TFTP servers. . One of the problems with TFTP is that it will transfer files in the clear without passwords. In the past it was common to TFTP router configurations to a TFTP server. There is a buffer overflow exploit with Cisco devices running IOS 11.1, 11.2 and 11.3 that can cause a reset of the device.

Recommendations:
It is very bad if someone can reach into your network and reboot a network device. I would recommend to the administrators to block this traffic at the firewall. There is also liability if someone internal to the network causes damage to another network either knowingly or unknowingly. Therefore I concur with the administrators efforts to monitor such traffic and also recommend filtering that traffic at the network egress points.

References:
Windows IT Library gives details on TFTP in the context of Internet server security. (Sheresh, Sheresh, Cowart)
http://www.windowsitlibrary.com/Content/405/13/5.html

Chris Lewis of Network Computing discusses some of the pitfalls associated with TFTP. (Lewis, Chris)
http://www.networkcomputing.com/818/818buyers3.html

John Barkley of NIST explains procedures for testing the Security of TFTP (Barkley, John)
http://csrc.nist.gov/publications/nistpubs/800-7/node141.html

IRC evil - running XDCC

This is another custom rule looking for the IRC or Internet Relay Chat file sharing activity. XDCC works like an automated file server that will list file in a chat room for people to download. The problem is that it can be used as a vehicle to server up other malware that can do password cracking, open up back doors, or install daemons that startup with Windows and bypass the need for a back door.

Examples of the activity:

```
05/08-01:08:11.512328  [**] IRC evil - running XDCC [**] MY.NET.227.246:4073 -> 216.32.207.207:6666
05/08-01:55:55.012929  [**] IRC evil - running XDCC [**] MY.NET.241.246:4605 -> 206.167.75.78:6667
05/08-06:28:00.234451  [**] IRC evil - running XDCC [**] MY.NET.80.209:3008 -> 134.33.33.33:6665
05/08-06:59:11.826590  [**] IRC evil - running XDCC [**] MY.NET.227.246:2816 -> 160.94.151.137:6665
05/08-07:29:20.437603  [**] IRC evil - running XDCC [**] MY.NET.207.78:1525 -> 194.78.213.3:6667
```

The destination ports are within the range for using the IRC servers and the source ports are unremarkable. Port 1525 is used by an Oracle server and if someone was actually using IRC on an Oracle server that would be a large compromise.

Recommendations:
This activity should be monitored but will probably always be in the background at a University. The economics of University life will add temptation to some few. A good security policy and security education could also be used to curtail the use of IRC when the students realize that an attacker could completely compromise their system and the work on them.

References:
Excellent paper on XDCC and how to hack it by TonikGin (TonikGin, XDCC)
http://www.russonline.net/tonikgin/EduHacking.html

Instructions for Connecting to IRCnet Servers and the ports that they use. (Lo, Joseph)
http://www.irchelp.org/irchelp/networks/servers/ircnet.html

Nimda - Attempt to execute cmd from campus host

This detect and the Nimda – Attempt to execute root from campus host are custom rules for the University. Both are significant because there is a good possibility that a campus host is infected with some variant of the Nimda Worm.

Nimda is notable because of the amount and variety of mechanisms it used to infect hosts. It can propagate through email, by infecting HTML files, and by making copies of itself that can infect network shares. If a large enough number of systems become infected then the worm replication efforts could affect network bandwidth.

```
05/10-23:50:45.395242  [**] NIMDA - Attempt to execute cmd from campus host [**]
MY.NET.97.105:1618 -> 130.118.61.14:80
05/11-00:11:50.514390  [**] NIMDA - Attempt to execute cmd from campus host [**]
MY.NET.97.105:1537 -> 130.206.215.21:80
05/11-00:17:04.024490  [**] NIMDA - Attempt to execute cmd from campus host [**]
MY.NET.97.105:2567 -> 130.94.230.29:80

05/10-23:50:47.460336  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.97.105:1469 -> 130.223.20.60:80
```

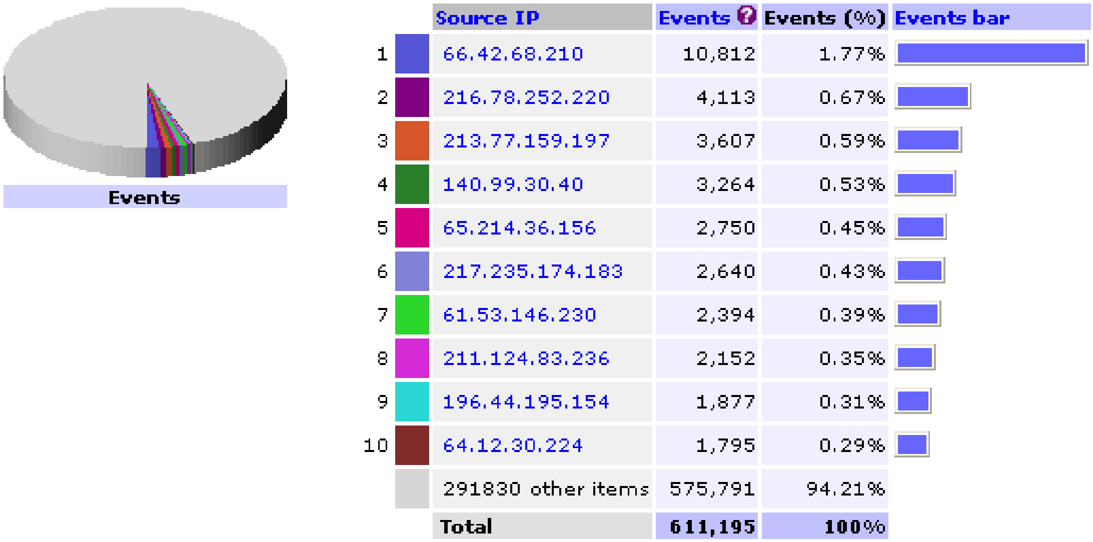
To be thorough, I examined Dshield.org for any references to the addresses listed above. 130.223.0.0 is registered to the University of Lausanne. There were has no records against the two IP addresses above. 130.118.61.14 is owned by the U.S. Geological Survey with not hits against that addresses. They appear to just be victims of the attack.

Recommendations:
Recommend that the system admins seek out host 97.105 and repair that system before it infects other systems or causes more harm. Luckily they are only 21 hits in the database for this alert so that the infestation appears to be small.

References:
Cert Advisory on the Nimda Worm (Cert Advisory, CA-2001-19)
http://www.cert.org/advisories/CA-2001-26.html

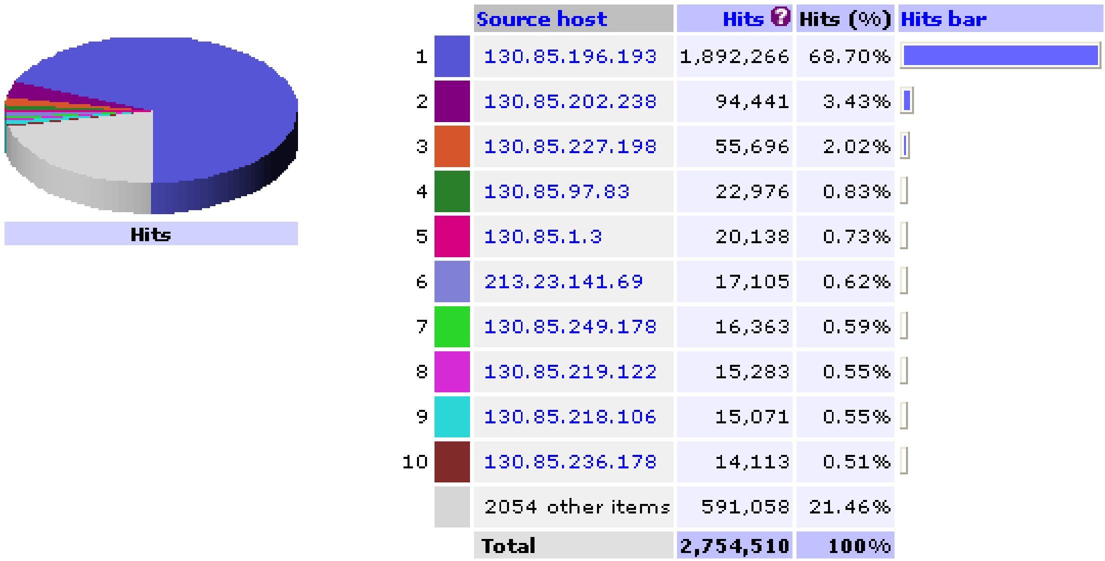
Top Talkers list in terms of Scans, Alerts, and OOS Files

In terms of Alerts the Top Talkers are listed below:



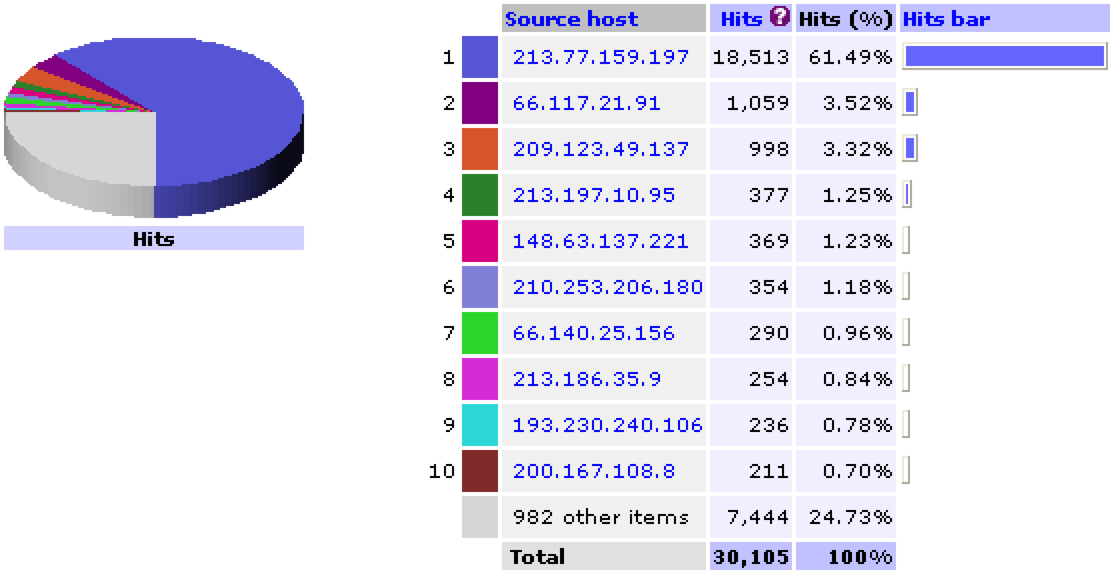
		Source IP	Events	Events (%)	Events bar
1	■	66.42.68.210	10,812	1.77%	▬
2	■	216.78.252.220	4,113	0.67%	▬
3	■	213.77.159.197	3,607	0.59%	▬
4	■	140.99.30.40	3,264	0.53%	▬
5	■	65.214.36.156	2,750	0.45%	▬
6	■	217.235.174.183	2,640	0.43%	▬
7	■	61.53.146.230	2,394	0.39%	▬
8	■	211.124.83.236	2,152	0.35%	▬
9	■	196.44.195.154	1,877	0.31%	▬
10	■	64.12.30.224	1,795	0.29%	▬
	■	291830 other items	575,791	94.21%	
		Total	**611,195**	**100%**	

Top Talkers for Scans:



		Source host	Hits	Hits (%)	Hits bar
1	■	130.85.196.193	1,892,266	68.70%	▬
2	■	130.85.202.238	94,441	3.43%	▬
3	■	130.85.227.198	55,696	2.02%	▬
4	■	130.85.97.83	22,976	0.83%	▬
5	■	130.85.1.3	20,138	0.73%	▬
6	■	213.23.141.69	17,105	0.62%	▬
7	■	130.85.249.178	16,363	0.59%	▬
8	■	130.85.219.122	15,283	0.55%	▬
9	■	130.85.218.106	15,071	0.55%	▬
10	■	130.85.236.178	14,113	0.51%	▬
	■	2054 other items	591,058	21.46%	
		Total	**2,754,510**	**100%**	

Top Talkers for OOS Files



		Source host	Hits	Hits (%)	Hits bar
1	■	213.77.159.197	18,513	61.49%	
2	■	66.117.21.91	1,059	3.52%	
3	■	209.123.49.137	998	3.32%	
4	■	213.197.10.95	377	1.25%	
5	■	148.63.137.221	369	1.23%	
6	■	210.253.206.180	354	1.18%	
7	■	66.140.25.156	290	0.96%	
8	■	213.186.35.9	254	0.84%	
9	■	193.230.240.106	236	0.78%	
10	■	200.167.108.8	211	0.70%	
	■	982 other items	7,444	24.73%	
		Total	30,105	100%	

5 External source addresses and registration information with reasons why chosen.

The Top Talkers list was a good way to decide on which external addresses to do more research on. They were the addresses causing the most impact to the security logs. The
Possible Trojan Servers are also a concern and worth a more in-depth look.

Alert Top Talker 66.42.68.210

This address has no records on DSHIELD.org. The Whois information is not very detailed on this address. Pac-West Telecom is a company that provides telephone and data services through T1 lines.

Whois:

OrgName: Pac-West Telecomm, INC.
OrgID: PWTI
Address: 1776 W. March Lane
Address: Suite 250
City: Stockton
StateProv: CA
PostalCode: 95207
Country: US

NetRange: 66.42.0.0 - 66.42.127.255
CIDR: 66.42.0.0/17
NetName: MDSG-PACWEST-1BLK
NetHandle: NET-66-42-0-0-1
Parent: NET-66-0-0-0-0
NetType: Direct Allocation

```
NameServer: NS1.MDSG-PACWEST.COM
NameServer: NS2.MDSG-PACWEST.COM
NameServer: NS3.MDSG-PACWEST.COM
NameServer: NS4.MDSG-PACWEST.COM
NameServer: NS5.MDSG-PACWEST.COM
NameServer: NS6.MDSG-PACWEST.COM
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2000-11-10
Updated:    2002-11-15

TechHandle: ZP86-ARIN
TechName:   Administrator
TechPhone:  +1-800-722-9378
TechEmail:  admin@mdsg-pacwest.com

OrgTechHandle: ZP86-ARIN
OrgTechName:   Administrator
OrgTechPhone:  +1-800-722-9378
OrgTechEmail:  admin@mdsg-pacwest.com

# ARIN WHOIS database, last updated 2003-04-28 20:10
# Enter ? for additional hints on searching ARIN's WHOIS database.


OrgName:    American Registry for Internet Numbers
OrgID:      ARIN
Address:    3635 Concorde Parkway, Suite 200
City:       Chantilly
StateProv:  VA
PostalCode: 20151
Country:    US

NetRange:   66.0.0.0 - 66.255.255.255
CIDR:       66.0.0.0/8
NetName:    NET66
NetHandle:  NET-66-0-0-0-0
Parent:
NetType:    Allocated to ARIN
NameServer: ARROWROOT.ARIN.NET
NameServer: BUCHU.ARIN.NET
NameServer: CHIA.ARIN.NET
NameServer: DILL.ARIN.NET
NameServer: EPAZOTE.ARIN.NET
NameServer: FIGWORT.ARIN.NET
NameServer: GINSENG.ARIN.NET
NameServer: HENNA.ARIN.NET
NameServer: INDIGO.ARIN.NET
Comment:
RegDate:    2000-07-01
Updated:    2002-08-23

OrgTechHandle: IP-FIX-ARIN
OrgTechName:   ARIN IP Team
OrgTechPhone:  +1-703-227-0660
OrgTechEmail:  hostmaster@arin.net
```

```
OrgNOCHandle: ARINN-ARIN
OrgNOCName:   ARIN NOC
OrgNOCPhone:  +1-703-227-9840
OrgNOCEmail:  noc@arin.net

# ARIN WHOIS database, last updated 2003-04-28 20:10
# Enter ? for additional hints on searching ARIN's WHOIS database.


OrgName:    Pac-West Telecomm, INC.
OrgID:      PWTI
Address:    1776 W. March Lane
Address:    Suite 250
City:       Stockton
StateProv:  CA
PostalCode: 95207
Country:    US
Comment:
RegDate:    1996-04-24
Updated:    2002-11-20

AdminHandle: ZP86-ARIN
AdminName:   Administrator
AdminPhone:  +1-800-722-9378
AdminEmail:  admin@mdsg-pacwest.com

TechHandle: ZP86-ARIN
TechName:   Administrator
TechPhone:  +1-800-722-9378
TechEmail:  admin@mdsg-pacwest.com

# ARIN WHOIS database, last updated 2003-04-28 20:10
# Enter ? for additional hints on searching ARIN's WHOIS database
```

A simple ping –a request returned the following host name “66-42-68-210.stkn.mdsg-pacwest.com [66.42.68.210]”. It appears to be some server in Stockton. STKN seems a likely abbreviation for Stockton. “MDSG” sounds like a messaging server and is inline with the TechEmail address. However it did not answer to port 25. It also did not answer to port 80 or port 443. It is difficult to determine the exact purpose of this server but the naming structure makes it appear to be valid. This inclines the analyst to believe that the IP address was spoofed for the attacks or that the server was compromised. The alerts for this IP address were “High Port 65535 UDP – Possible Red Worm – traffic”. This would indicate that the system was more likely a compromised host sending traffic to the University.

Scan Top Talker 130.85.196.193

For this address I used the lookup feature within Sam Spade.

```
OrgName:    University of Maryland Baltimore County
```

```
OrgID:      UMBC
Address:    UMBC University Computing
City:       Baltimore
StateProv:  MD
PostalCode: 21250
Country:    US

NetRange:   130.85.0.0 - 130.85.255.255
CIDR:       130.85.0.0/16
NetName:    UMBCNET
NetHandle:  NET-130-85-0-0-1
Parent:     NET-130-0-0-0-0
NetType:    Direct Assignment
NameServer: UMBC5.UMBC.EDU
NameServer: UMBC4.UMBC.EDU
NameServer: UMBC3.UMBC.EDU
Comment:
RegDate:    1988-07-05
Updated:    2000-03-17

TechHandle: JJS41-ARIN
TechName:   Suess, John J.
TechPhone:  +1-410-455-2582
TechEmail:  jack@umbc.edu

# ARIN WHOIS database, last updated 2003-06-28 21:05
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

This host predominantly used a simple SYN scan that can be performed from a variety of tools including NMAP. The host did not respond to pings, telnets or web requests. It is difficult to determine what type of system it is or if it is even operational.

DShield.org did not have any information on the IP address. I would recommend screening this address at the firewall to prevent log buildup or bandwidth utilization.

OOS Top Talker 213.77.159.197

Sam Spade returned the following ARIN data on this address:

```
OrgName:    RIPE Network Coordination Centre
OrgID:      RIPE
Address:    Singel 258
Address:    1016 AB
City:       Amsterdam
StateProv:
PostalCode:
Country:    NL
```

```
NetRange:   213.0.0.0 - 213.255.255.255
CIDR:       213.0.0.0/8
NetName:    RIPE-213
NetHandle:  NET-213-0-0-0-1
Parent:
NetType:    Allocated to RIPE NCC
NameServer: NS.RIPE.NET
NameServer: NS3.NIC.FR
NameServer: SUNIC.SUNET.SE
NameServer: AUTH00.NS.UU.NET
NameServer: MUNNARI.OZ.AU
NameServer: SEC1.APNIC.NET
NameServer: SEC3.APNIC.NET
NameServer: TINNIE.ARIN.NET
Comment:    These addresses have been further assigned to users in
Comment:    the RIPE NCC region. Contact information can be found in
Comment:    the RIPE database at http://www.ripe.net/whois
RegDate:
Updated:    2003-04-25

OrgTechHandle: RIPE-NCC-ARIN
OrgTechName:   RIPE NCC Hostmaster
OrgTechPhone:  +31 20 535 4444
OrgTechEmail:  nicdb@ripe.net

# ARIN WHOIS database, last updated 2003-06-28 21:05
```

A ping –a report returned the host name of the address “a197.mielec.sdi.tpnet.pl [213.77.159.197]”. The host did not respond to telnets, telnets to port 25, or web requests. The tpnet.pl is not a known domain to ARIN. There is a good possibility this is a spoofed address or a system that is open to the Internet and has an internal DNS structure unknown to the ARIN database.

Possible Trojan Server – 65.68.62.41

```
OrgName:    SBC Internet Services - Southwest
OrgID:      SBIS
Address:    2701 W 15th St PMB 236
City:       Plano
StateProv:  TX
PostalCode: 75075
Country:    US

NetRange:   65.64.0.0 - 65.71.255.255
```

```
CIDR:       65.64.0.0/13
NetName:    SBIS-5BLK
NetHandle:  NET-65-64-0-0-1
Parent:     NET-65-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.SWBELL.NET
NameServer: NS2.SWBELL.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
Comment:    please send all abuse issue e-mails to abuse@swbell.net
RegDate:    2000-10-03
Updated:    2002-08-08

TechHandle: ZS44-ARIN
TechName:   IPAdmin-SBIS
TechPhone:  +1-888-212-5411
TechEmail:  IPAdmin-SBIS@sbis.sbc.com

OrgAbuseHandle: ABUSE6-ARIN
OrgAbuseName:   Abuse - Southwestern Bell Internet
OrgAbusePhone:  +1-877-722-3755
OrgAbuseEmail:  abuse@swbell.net

OrgNOCHandle: SUPPO-ARIN
OrgNOCName:   Support - Southwestern Bell Internet Services
OrgNOCPhone:  +1-888-212-5411
OrgNOCEmail:  support@swbell.net

OrgTechHandle: IPADM2-ARIN
OrgTechName:   IPAdmin-SBIS
OrgTechPhone:  +1-888-212-5411
OrgTechEmail:  IPAdmin-SBIS@sbis.sbc.com

# ARIN WHOIS database, last updated 2003-06-28 21:05
```

A ping –a returned the following host name adsl-65-68-62-41.dsl.rcsntx.swbell.net [65.68.62.41]. The host did not respond to telnet or web requestes.

Possible Trojan Server – 24.202.144.208

```
Le Groupe Videotron Ltee VL-2BL (NET-24-200-0-0-1)
                                  24.200.0.0 - 24.203.255.255
Le Groupe Videotron Ltee VL-D-MX-18CA9000 (NET-24-202-144-0-1)
                                  24.202.144.0 - 24.202.144.255

# ARIN WHOIS database, last updated 2003-06-28 21:05
```

A ping –a returned the following host name modemcable208.144-202-24.mtl.mc.videotron.ca [24.202.144.208]. The host did not respond to telnet or web requests. Videotron is an ISP for Canada. The host appears to be a cable modem for that ISP however the address could be spoofed.

Correlations from student practicals

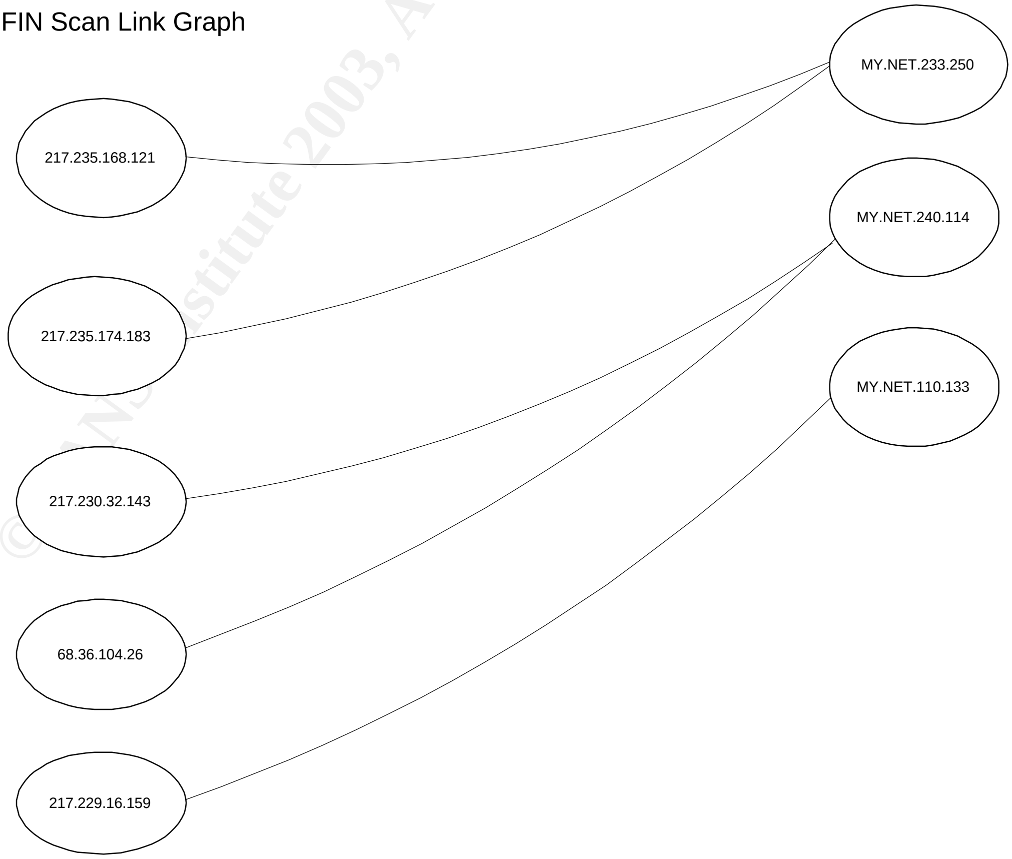
Doug Kite's GCIA Practical (Kite, Doug. Practical)
http://www.giac.org/practical/GCIA/Doug_Kite_GCIA.pdf

Hee So's GCIA practical (So, Hee. Practical)
http://www.giac.org/practical/Hee_So_GCIA.doc

Link graph

A link graph is a good tool to get a good idea of traffic flow patterns. The graph shows that 3 of the internal network servers are being scrutinized by bad parties. The university should review those serves to make sure they are properly patched and evaluate more stringent protection if need be.

SYN-FIN Scan Link Graph



Defensive recommendations

The University could receive some security benefit by implementing a number of steps. First, increase the perimeter security. Have the firewall screen more traffic externally. That will lessen the number of reconnaissance probes, and external attacks that get through to the internal network. Second, the University should leverage the information from Dhield.org. Logs can be setup to automatically upload to the Dshield.org database and alert everyone if an address is behaving maliciously. Third, implement stronger controls against IRC. There is a wealth of opportunities to compromise IRC systems. Last, the University should work closely with the other Universities where attacks are originating from. By leveraging each other resources they have a better chance of combating attacks.

Description of my analysis process

For the analysis of these 5 days it is of course more difficult to determine what is truly vital to the organization involved. Several questions that could not be asked include: do they have trade secrets that need to be protected? Many research schools have proprietary information that needs to be protected. Is this a health research university? In that case it might fall under HIPAA regulations. Is this University in the State of California? With the new piece of legislation Senate Bill 1386, the University could be liable for personal information revealed to an attacker.

With that sort of information it would be easier to determine the relative position of the organization and whether or not they were near dangerous shoal water.

All that said, I took a more visual and holistic approach to the analysis rather than packet by packet analysis. This would be the stance that I would recommend to an analyst at that organization. Get a fix on the overall position of the company by looking at the general metrics of attacks and scans. Then start drilling down on key suspicious data but not so deep you lose track of the flow.

To get a better handle on the total week I concatenated the files by using the copy command into one large file of alerts and another of scans.

This I ran through a database agent to determine the macro data. Sawmill 6 did a great job at pulling out key data points. For those working on Windows systems I would highly recommend this tool. It is far less CPU intensive than other tools I've utilized.

Because of the nature of the tool I needed to look at the alerts and scans separately. Once there was good data on both I could look for correlations.

The Alert data had substantial corruption in it. I utilized NoteTab Pro to help massage the data into a more manageable shape and then later to assist pulling out information.

List of References – Section II and Section III

Section II

1. Wesemann, Daniel. LOGS: GIAC GCIA Version 3.3 Practical De
http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00011.html

2. Cisco. Overview Internet Protocol (IP) Multicast. Copyright © 1992--2002 Cisco Systems, Inc. All rights reserved.
http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ipimt_ov.htm

3. Windump (Copyright (c) 1999-2002, Politecnico di Torino.
http://windump.polito.it

4. Stevens, Richard. TCP/IP Ilustrated, Volume 1. Copyright 1994 by Addison Wesley

5. Fulton, Russell. r.fulton@auckland.ac.nz
http://msgs.securepoint.com/cgi-bin/get/snort-0304/352.html

6. Wood, Phil. cpw@lanl.gov
http://archives.neohapsis.com/archives/snort/2000-12/0413.html

7. Clark, Daniel. LOGS: GIAC GCIA Version 3.3 Practical Detect (D Clark, ver 3)
http://cert.uni-stuttgart.de/archive/intrusions/2003/05/msg00183.html

8. Snort-users Mailing list. lists.sourceforge.net
http://www.somelist.com/mails/312073.html

9. CVE, Common Vulnerabilities and Exposures. The MITRE Corporation
http://www.cve.mitre.org

10. eEye. Copyright (c) 1998-2003 eEye Digital Security
http://www.eeye.com/html/Research/Advisories/AD20010618.html

11. Symantec. Symantec Customer Security Advisory for the CodeRed Worm. Copyright (c) 2001 by Symantec Corp**.**
http://securityresponse.symantec.com/avcenter/security/Content/2001_07_31.html

12. Adams, Samuel C. LOGS GIAC GCIA Version 3.3 Practical Detect

http://cert.uni-stuttgart.de/archive/intrusions/2003/06/msg00000.html

13. Cisco. Using Network-Based Application Recognition and ACLs for Blocking the "Code Red" Worm.Copyright © 1992-2003 Cisco Systems, Inc.
http://www.cisco.com/warp/public/63/nbar_acl_codered.shtml

14. Adams, Tony. Practical. 10 Detects for SANS GIAC Intrusion Analyst Certification. 4/8/2000

www.giac.org/practical/Tony_Adams.doc

15. Nasrat, Paul. Squid HTTPd Acceleration ACL Bug Enables Port Scanning. 7/19/2001

http://www.securiteam.com/unixfocus/5LP0P0U4UQ.html

16. Squid-cache. SQUID Frequently Asked Questions. © 2001 Duane Wessels, wessels@squid-cache.org

http://www.squid-cache.org/Doc/FAQ/FAQ.html

17. Galarneau, Eric. Security considerations with Squid proxy server. April 2, 2003
http://www.sans.org/rr/papers/50/1048.pdf

18. Northcutt, Stephen. What was the Ring Zero Scan? October 11, 1999
http://www.sans.org/resources/idfaq/ring_zero.php

Section III

19. Securiteam, Multiple Vulnerabilities in Snort Preprocessors (RPC, stream4) 4/18/2003

http://www.securiteam.com/securitynews/5FP0A2A9QI.html

20. Ruiu, Dragos. How do I configure stream4?
http://www.snort.org/docs/3.14

21. Roesch, Marty. Incomplete Packet Fragments Discarded. 11/26/2001
http://www.mcabee.org/lists/snort-users/Nov-01/msg00820.html

22. Larratt, Glen. Practical. Intrusion Detection in Depth.
http://is.rice.edu/~glratt/practical/Glenn_Larratt_GCIA.html

23. Securiteam, Multiple Vulnerabilities in Snort Preprocessors (RPC, stream4) 4/18/2003

http://www.securiteam.com/securitynews/5FP0A2A9QI.html

24. Martin, Daniel. Spoofed SMB Name Wildcard probes. 5/4/2001
http://lists.jammed.com/incidents/2001/05/0034.htm

25. Alexander Bryce, Port 137 Scan, 5/10/200
http://www.sans.org/resources/idfaq/port_137.php

26. Green, Chris. What about 'SMB Name Wildcard' alerts?. 3/25/2002
http://www.snort.org/docs/4.15

27. Axonpotential, Snort DNS Preprocessor 1.3
http://www.geocities.com/axonpotential/snort/18/

28. Millican, Andy. Practical Network Reconnaissance -Detection and Prevention. 1/23/2003
http://www.giac.org/practical/GSEC/Andy_Millican_GSEC.pdf

29. CommuniLink, Domain Services.
http://www.communilink.net/en/add_port1025.html

30. SecurityFocus, Network Blackjack 6/14-6/19/2002
http://www.securityfocus.com/archive/105/277275/2002-06-09/2002-06-15/1

31. Krueger, Karl A. Multiple scans to port 1025. 3/21/2002
http://cert.uni-stuttgart.de/archive/intrusions/2002/03/msg00014.html

32. Glock Software, Bad Blood
http://www.glocksoft.com/trojan_list/Bad_Blood.htm

33. Sheresh, Sheresh, Cowart. , Establishing Internet Server Security. Windows IT Library. April 1999
http://www.windowsitlibrary.com/Content/405/13/5.html

34. Lewis, Chris. Keeping Your Network Safe and Sound. Network Computing. 9/24/1997
http://www.networkcomputing.com/818/818buyers.html

35. Barkley, John. Improving the Security of TFTP. 10/17/1994
http://csrc.nist.gov/publications/nistpubs/800-7/node141.html

36. TonikGin, XDCC – An .EDU Admin's Nightmare. 9/11/2002.
http://www.russonline.net/tonikgin/EduHacking.html

37. Lo, Joseph and staff at EFnet, IRChelpIRCnet ServerList. .org Internet Relay Chat (IRC) help archive. 4/23/01

http://www.irchelp.org/irchelp/networks/servers/ircnet.html

38. CERT Advisory CA-2001-19 “Code Red” Worm Exploiting Buffer Overflow in IIS Indexing Service DLL, January 17,2002
http://www.cert.org/advisories/CA-2001-19.html

39. Kite, Doug. Practical. Intrusion Detection in Depth. July 2002
http://www.giac.org/practical/GCIA/Doug_Kite_GCIA.pdf

40. So, Hee. Practical. Intrusion Detection in Depth. 2/16/2002
http://www.giac.org/practical/Hee_So_GCIA.doc

41. Microsoft Security Bulletin MS01-033 6/18/2001
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp