



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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

## Interested in learning more?

Check out the list of upcoming events offering  
"Intrusion Detection In-Depth (Security 503)"  
at <http://www.giac.org/registration/gcia>



**SANS Intrusion Detection in Depth  
GCIA Practical Assignment  
Version 3.4**

**Stephen Breault  
Sans Atlanta - GA**

|   |           |
|---|-----------|
| <b>Part 1 – Describe the state of intrusion detection .....</b> | <b>3</b>  |
| Introduction.....   | 3         |
| Conclusion.....   | 11        |
| References.....   | 10        |
| <br>  |           |
| <b>Part 2 – Network Detects.....</b>                            | <b>12</b> |
| Detect #1 – Trin00.....   | 12        |
| Detect #2 – Socks proxy.....                                    | 20        |
| Detect #3 – Null Scan.....                                      | 28        |
| Posted Detect .....   | 34        |
| <br>  |           |
| <b>Part 3 – Analyze This.....</b>                               | <b>36</b> |
| Executive summary.....  | 36        |
| Methodology.....  | 37        |
| Logs.....   | 37        |
| Top Ten Alerts.....   | 38        |
| Top Ten Scan.....   | 46        |
| Top Ten Source IP from scan.....                                | 51        |
| External Address of Interest.....                               | 57        |
| OOS Top Ten.....  | 69        |
| Insights about internal machine.....                            | 70        |
| Link Graph Analysis.....  | 71        |
| Overall defensive recommendations.....                          | 72        |
| References.....   | 72        |

© SANS Institute. All rights reserved.

## **Wireless network a chink in the armor**

### **Introduction**

It is no great secret that Wireless local area networks (WLAN) have recently undergone an explosion in popularity. I was recently in Atlanta during the SANS conference, and I was amazed at the popularity of this relatively new technology. It seems that everybody at the conference had this capability at hand. The locals boasted about Atlanta's numerous access points, which seemed to cover almost the entire city.

The popularity of wlan is easy to understand. At a glance we can see that the low cost, and moderately easy install along with the mobile freedom afforded by this technology are key factors in its increasing popularity. (I proclaim low cost only when we compare this to the typical local area network where it's necessary to pull cable through walls and sometimes run miles of it through cramped spaces).

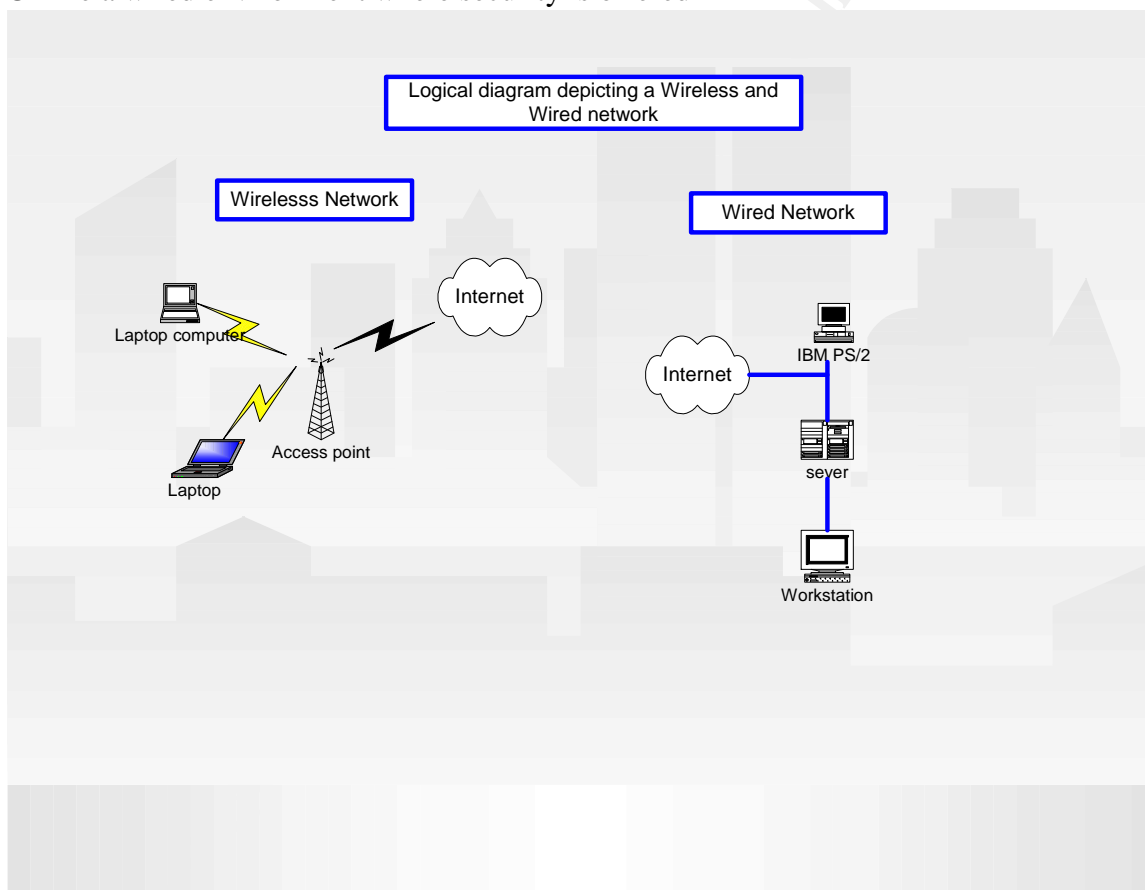
As we grow more dependant on wireless technology for business, education and other utilities we can surely say that lurking in the shadow is the threat of individuals who are waiting to exploit some of it's weaknesses. In this paper my intention is to bring to light what makes wireless networks vulnerable. Also the ways, which someone can find the potentially vulnerable access points along with the vulnerabilities, I will also discuss how to secure the wireless network. Covered also are the necessary steps required to protect your wlan against malicious intentions, and the challenges that we face in doing so.

That being said how does securing a wireless network relate to the state of intrusion detection today you ask? There is precious little out there today detailing what constitutes a wireless lan, and its terminology. Many of us have heard of SSID and other terms but don't understand the meaning of them. I will clearly define what these terms mean in easy to understand terms. Only through knowledge can true security come from.

Another topic, which has garnered much attention is the still maturing technology of wireless intrusion detection systems. Included in this paper will be a listing of a couple of commercial solutions out there today. Even with this technology securing a wireless lan can be a difficult task. Why you ask? Simply because there is not a lot of information out there detailing how to proactively do so. It is this papers intention to help bridge that divide.

## **Background information**

Wlan broadcasts users identification and traffic over the airwave much like a pebble dropped in a pond that creates a circular effect causing ripples. It propagates over an area that is not really controlled by any physical boundaries other than the weakening of the signal over a distance greater than the transmitting capability. In essence non-intended users such as a person on the floor below or in parking lot across the street are almost guaranteed the success of a potentially valuable intercept. This is a military term for which is used in Electronic Warfare. This is where exploiting intelligence through intercept is done and is the bread and butter of intelligence agencies all over the world. Unlike a wired environment where security is offered



through the use of physical means such as locked doors, security passes and where transmission is limited to a cable, wireless has the potential to offer information freely without any boundaries. Above is an illustrated example.

## **Security and integrity of WLAN**

This is where WEP comes in to play, the IEEE 802.11 standard which details wireless is equipped with **Wireless Equivalency Protocol** (which was dispelled as a viable security mechanism). It for its most basics function provides some encrypted security through the use of secret keys and algorithm. It allows for users, and hosts a protected way of recognizing each other over the airwave. A significant flaw with WEP was discovered in the encryption and authentication. University research teams exposed those flaws as a result the IEEE along with Wi-Fi Alliance developed **Wi-Fi Protected Access** or WPA. This provides strong data encryption and added user identification that was earlier recognized as one of the flaws with the equivalency protocol. Another weakness with wireless equivalency protocol (WEP) is the system administrator has the ability to activate or de-activate at will security features. Some access points come with WEP disabled therefore leaving the onus on the system administrator to activate the feature. This leaves a lot of room for human error and neglect. Its usually the first thing that a person looking for an easy exploit will surely find. Much like some features in the Windows operating system that are often left in a default mode only to be discovered after a significant event occurred that might of been preventable.

**SSID**(service set identifier) also referred to as a network name is broadcasted by most access points, and as such a hacker can easily identify these by using a wireless sniffer. Cisco for example has in the past used the word “tsunami” as an SSID leaving very little work for intruders in their attempt to ascertain the SSID name. The use of tools such as Airmagnet, and Netstumbler, also help their quest into further probing or infiltrating networks. The broadcasting of SSID is a feature that can be disabled therefore reducing the risk of advertising identifying properties of a network. This is by no means a stop all security feature. A hacker intent on finding an SSID only has to wait until someone accesses the network, and with the tools mentioned above he can then sniff the SSID from the frame that the wireless station uses to connect to the access point

**DHCP** (dynamic host configuration protocol) is a feature, which assigns IP addresses automatically to users who have the proper SSID (can you see where I am going with this?). The resulting factors are since IP addresses are not static in this context you have just configured your network to offer a possible attacker a valid IP address. This can allow them to further bury themselves into your networks. Evidently this should not be used and as such it is wise to assign static IP address to specific users where it is permissible to do so.

**MAC** (media access control) filtering is a method by which a network interfaces hard coded address is stored on the access points list for acceptable users. As such any MAC address not on this list will not be allowed on the network. This again is not a bulletproof method of securing a network as MAC addresses can also be spoofed. The software tool [SMAC](#) allows you to spoof MAC addresses rather trivially. Even with that in mind as you can see it offers another layer of security.

**Radius** (Remote Authentication Dial In User service) this works much like the tcp handshake, first the user connects to the network access client, then [radius](#) sends an access request to an authentication server with information like passwords and message

authenticator, after the request is made radius sends a challenge (set of key's) to the user after authentication of key's the request is then accepted or rejected access.

**EAP** (extensible authentication protocol) is the protocol used to encapsulate information destined for the authentication server. This is required as information is sent over the airwave; normally this is a radius server as described above. Radius servers can support multiple types of [EAP](#).

### **How porous can WLAN be?**

It is said that if you have 10,000 users you have 10,000 holes that could potentially expose vulnerabilities in your network. The following is from [Intel's](#) article on wireless lan security which illustrates this point very well;

“Two elements of WLAN security, access to the network and data protection, are known respectively as authentication and encryption. Security breaches commonly come from rogue access points (AP), which are set up by employees without the knowledge of the network administrator and installed with the security features turned off (which is the default setting). An individual PC can also be a security risk if it is connecting to a network in ad hoc mode or operating in peer-to-peer fashion.” <http://www.intel.com/business/bss/infrastructure/wireless/security/index.htm>

These access points as mentioned above are potentially the most damaging to corporate secrets, and security. They are usually set up by employees without the consent, or knowledge of the companies security staff. Furthermore they are often the subject of poor security practices such as settings left in default mode. Often the reasons why an employee would want to set up an access point are reasonable. They might for example want to do their work remotely, for testing purposes, or even just simply to have access to the network. Unfortunately there is no way of physically preventing these rogue points from propping up. Only through the steadfast adherence to a companies established security policy can well informed employees prevent these access points from appearing without the approval of the IT staff.




### **The Wireless Battleground**

#### **a) War driving**

Some of the resulting attack or scanning vectors created by these rogue access points are becoming better known as they are glorified by various hacking ezines. War driving doesn't exclusively look for rogue access points with no security in place. The path of least resistance though is normally traveled in an attempt to gain access to a wireless network. War drivers are not just looking to exploit the latest vulnerabilities in operating systems by using these unsecured access points. Some of these cyber criminals also are only interested in stealing some online time via someone else's bandwidth. A war driver with the help of the following tools; laptop, wireless card, antennae and a GPS (optional) can drive around and look for available hot spots. After these spots are found they can be marked down on a map or logged into the GPS for future use. There exists a variety of software to help the war driver stumble onto "up for grabs" wireless networks. Tools such as [Netstumbler](#), which is most likely the program of choice for Window users. [Airsnot](#) which is capable of breaking( with the intercept of a certain amount of packets) WEP, and a sniffer called [Wellenreiter](#). This is but a sampling of the tools available today to the aspiring wireless hacker. As you can see there are many resources available for the determine war driver.

### **b) War chalking**

Wlan users in Europe started war chalking as a way to expose hot spots for users. The idea was to identify area's where an open access point could be reached. Upon the discovery of a node the War chalking person would then in turn write one of the below noted symbols on the ground with chalk. These symbols could indicate an open node, a closed node or wep node, amongst other criteria. The diagram below from mobile commerce indicates the markings that a war chalker could employ.

| let's warchalk..!              |  |
|--------------------------------|--|
| KEY                            | SYMBOL   |
| OPEN NODE                      | ssid<br><br>bandwidth                   |
| CLOSED NODE                    | ssid<br>                                |
| WEP NODE                       | ssid    access contact<br><br>bandwidth |
| blackbeltjones.com/warchalking |  |

[www.mobile.commerce.net/ story.php](http://www.mobile.commerce.net/story.php)

Wireless administrators probably feel that these types of markings expose their networks to a myriad of potential vulnerabilities. This hopefully reinforces the thought of properly configuring and securing the networks they administer against intruders. Implementing the proper and responsible steps required to secure their wireless assets will result in a lack of cryptic chalk marks by their office buildings.



## Hardening the perimeter: Defensive measures

Security and defensive measures are certainly abundant in both scope and type, but they all start with a strong security policy. This template will dictate what is acceptable use and what is not along with the steps required to keep the network secure. One of the most important factors though is end-user education.

End-users must know and be reminded of the risk associated with rogue access points. Topics such as default settings, war drivers, and the use of personal firewalls on notebooks for increased security. This is information that should be common knowledge. Users must also be aware of the risk of not using the proper level of security as it is important for them to understand the risks.

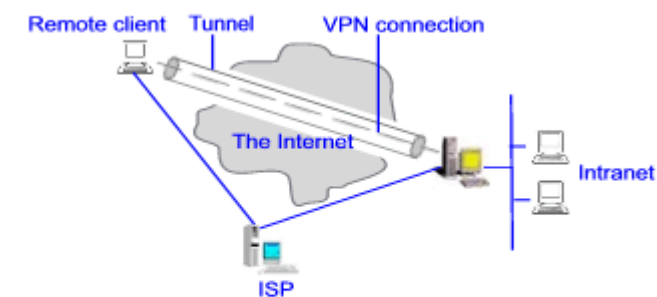
The following is an excerpt from the “wireless best security practices” guide from intel’s web site. It suggests that;

“Making virtual private networks (VPN) a requirement to access the network wirelessly is a scaleable and proven way to help protect your network. A strong protection mechanism isolates the WLAN user from the wired network by using the proven combination of a network firewall and Internet protocol security (IPsec) based VPN”

[http://www.intel.com/business/bss/infrastructure/wireless/security/best\\_practices.htm](http://www.intel.com/business/bss/infrastructure/wireless/security/best_practices.htm)

At the moment this is regarded as one of the best security strategies. In effect VPN offers an encrypted tunnel across a shared network such as the internet. The user would connect to the Wireless access Point (WAP) then to the VPN server, which would in turn provides a secure tunneled path to the private network such as an intranet. Below is an example of a VPN tunnel from [www.gifcomp.com](http://www.gifcomp.com).

<http://www.gifcomp.com/images/VPN.gif>

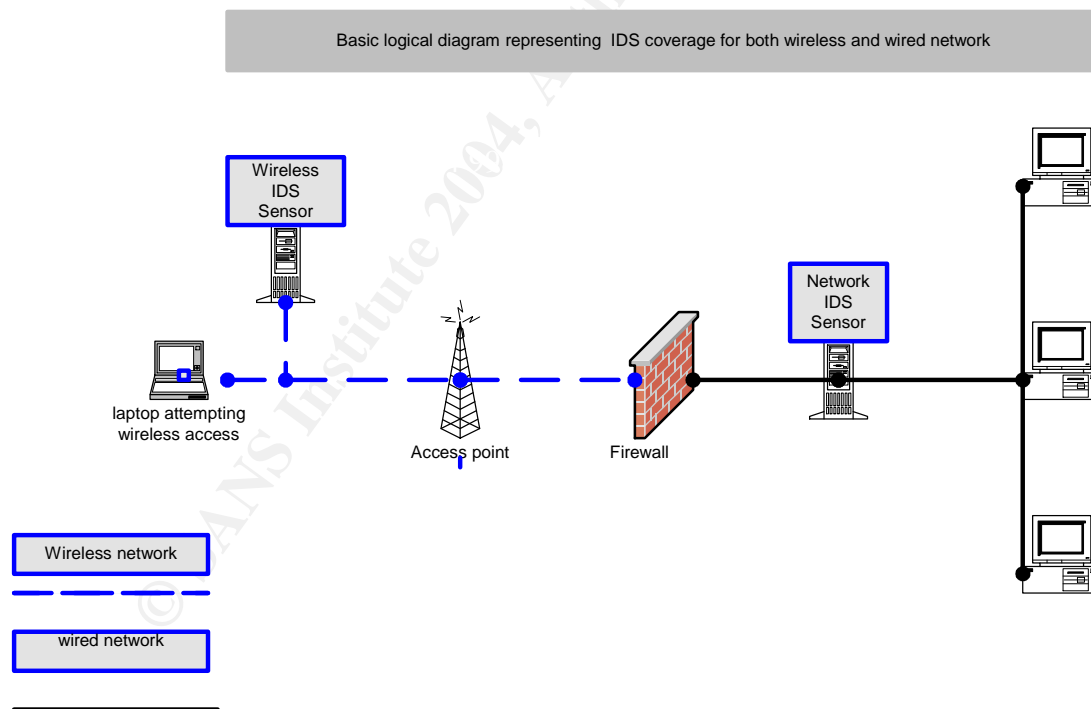


Using a VPN connection to connect a remote client to the intranet

The use of a wireless firewall can also be an option, there are now many types of these. They can be configured to practically any type of network supporting all kinds of users. It

will allow administrators to control individual access in turn allowing specific devices to internal networks. With the addition of configurable firewalls we can then limit various IP address access to the internal wired network to only those who are authorized.

We have seen so far that a hacker can break WEP, intercept DHCP addresses, capture a SSID, and on top of this he can also spoof MAC addresses. Having read all of this it may seem as if it is impossible to secure a wireless network. Let me assure you that it is not mission impossible. A critical layer of your defensive perimeter is the IDS. Unfortunately lan based intrusion detection systems do not have the ability to detect wireless based attacks. There are however wireless intrusion detection systems such as [AirOS 2.0](#), [airmagnet](#) and [airdefense](#) just to name a few. These will provide intrusion detection capabilities for your wireless resources. Some of these IDS's provide partial solutions to help discover rogue access points on your networks. They also offer a monitoring capability that lan based IDS cannot provide. Intrusion detection in this case is a manner in which both wired and wireless IDS can complement each other. They do not have to be separate entities. Analysts using both types of IDS would then have the power of correlating events. It would help them gain a more in depth understanding of the deluge of packets flowing across their networks.



Take a look at the above diagram, it illustrates a logical example that could be used for IDS sensor placement for both the wireless and wired lan. We can see that packets will be picked up by the wireless ids and then proceed to our access point. Same theory applies for the wired network, packets are picked up by our wired IDS sensors. This type of

layout would give the IDS analyst a good vision of the activity on the network regardless of its point of ingress. Also limiting the potential of any attack being launched without the analysts knowledge.

In order to ensure that a wireless network is free of rogue access points, and other possible dangers network administrators should in fact physically walk with detection software. In effect they would be trying to gain entry into their own systems using the aggressors own. They would, much like war drivers do, walk around their facilities utilizing many of the same tools, and look for access points.

Wireless access points signal strength should only be as strong as what is required for use. It is not necessary to have the signal broadcast beyond the required work space. If at all possible steps should be taken to reduce transmitter power and directionality. These measures should be verified and confirmed on a regular basis as well.

### **Case Study**

To put the threat posed to wireless networks today into context I will lay out a brief scenario for you, which will attempt to give you the reader further insight into this pernicious problem.

Coffee in hand you just exit the elevator and wave hello to the person sitting in the hallway as you enter your office. You can't help but notice the sexy Acer laptop in gleaming red that the person has.

This person cheerfully waves hello back, and returns to his perusal of your companies wireless access point. He has been running Kismet and Aircnort for the past hour now collecting a rich harvest of completely unfiltered packets. Within the space of that one hour this seemingly innocent individual has completely mapped your network, and snarfed over a dozen passwords.

The hacker closes their laptop now knowing exactly what versions of web server, and other key information. With this in hand the hacker will be back tomorrow with exploits geared towards your servers to attempt to further implant himself into your network.

Do you find this scenario far fetched? Let me assure you it most definitely is not, and is probably happening as you are reading this in some part of the world.

### **CONCLUSION**

We explored the basics of wlan broadcasting and it's capabilities along with its relative ease of intercept for intelligence gathering purposes. We also covered the Wireless Equivalancy Protocol, and its security weakness when pitted against the use of software such as aircnort to break the security features of encryption and authentication.

Also discovered was that the greatest threat to our network is the probability of a rogue access point existing and that could possibly translate into a wide open door for an intruder.

Discussed was the purpose of war driving and war chalking and understanding the potential vulnerabilities our networks are broadcasting over the air. Not to mention the markings left on the ground should a hacker find an open WAP.

These attack vectors could quite easily result in a full blown network breach. We must take into account though the security options detailed earlier in this paper as well to help mitigate these types of penetration attempts. Wireless IDS's and proper tuning of the wireless technology can go a long way towards securing your wireless assets.

Good defensive measures are never infallible but they do offer a protective security barrier against hackers who would surely be discouraged at the site of such a well-protected fortress. They surely would rather invest their time and effort into probing other networks that are weaker.

### **References:**

IEEE 802.1x Remote Authentication DIAL in User Service (radius) Usage Guidelines  
<ftp://ftp.isi.edu/in-notes/rfc3580.txt>

RADIUS Support for Extensible Authentication Protocol (EAP)  
<ftp://ftp.isi.edu/in-notes/rfc3579.txt>

<http://www.intel.com/business/bss/infrastructure/wireless/deployment/hotspot.pdf#vlan>

[http://www.atstake.com/research/reports/acrobat/atstake\\_wlan\\_hotspots.pdf](http://www.atstake.com/research/reports/acrobat/atstake_wlan_hotspots.pdf)

Wireless security: The gaps and how to fill them  
<http://www.eweek.com/article2/0,4149,1507241,00.asp>

Wireless wall frequently asked questions  
<http://www.cranite.com/solutions/wirelesswall/faqs.php>

Identifying Rogue Access Points by Jim Geier  
<http://www.wi-fiplanet.com/tutorials/article.php/1564431>

Wireless intrusion protection  
<http://www.networkassociates.com/us/nailabs/about/wip.asp>

Powerful Wireless Security Tools for Free by Vincent Ryan  
<http://wireless.newsfactor.com/perl/story/22124.html>

Wardriving HOWTO by FRED

<http://www.wardriving.com/doc/Wardriving-HOWTO.txt>

Wireless LAN Security

<http://www.intel.com/business/bss/infrastructure/wireless/security/index.htm>

<http://www.airsnort.shmoo.com/>

Aruba Wireless Networks Breaks New Ground with Wireless IDS and Secure Traffic Engineering Capabilities

<http://www.80211news.com/publications/page207-623161.asp>

Build your own hot spot by Jason Luther and Allen Fear

[http://reviews.cnet.com/4520-6603\\_7-5023845-1.html](http://reviews.cnet.com/4520-6603_7-5023845-1.html)

<http://www.gifcomp.com/images/VPN.gif>

Minimizing WLAN Security Threats; Jim Geier

<http://www.wi-fiplanet.com/tutorials/article.php/1457211>

So WEP fails to block everyone out, right? Does that mean all hope is lost

<http://www.wireless-network-guide.com/wireless-network-mac-filtering.php>

## **Part 2: Network detects**

### **Detect #1 – Trin00**

As I was examining raw log files, using snort, that were posted on incident.org/logs/raw website I came across a trace that immediately caught my attention. I then decided to further investigate and analyze this trace to use as my first of three network detects for assignment #2.

### **Source of trace**

The data in this trace was obtained from zip file 2003.12.15 and the actual raw file was named 2003.12.15.12

This is an assumption of a simplified network diagram taking in consideration the source and destination IP's and MAC addresses. It should be noted that although the ttl's through out the whole raw file are changing randomly both IP's and MAC mostly remain the same a possible indication of a tool at work. After looking up the mac address it became apparent that this could

possibly be Vmware, a tool that can simulate a lab environment onto a pc. It offers the ability to have what would look like different OS loaded on a single pc that would behave like separate physical machines. The attackers mac address in this case belongs to Intel corp ip 10.10.10.165 it is seen as trying all sort of attacks against ip 172.20.201.1 and the IDS is likely located in between the two. I should note that the previous statement cannot be absolutely without a doubt confirmed and that of course mac address can also be spoofed.



```

14:17:52.078334 0:3:47:8c:89:c2 0:50:56:40:0:6d 0800 60:
IP (tos 0x0, ttl 128, id 23805, len 39)
10.10.10.165.31335 > 172.20.201.1.27444:
[udp sum ok] udp 11
0x0000  4500 0027 5cfd 0000 8011 5404 0a0a 0aa5  E..\....T....
0x0010  ac14 c901 7a67 6b34 0013 47cf 706e 6720  ....zgk4..G.png.
0x0020  6c34 3461 6473 6c00 0000 0000 0000  144adsl.....
  
```

```

14:17:52.079597 0:50:56:40:0:6d 0:3:47:8c:89:c2 0800 81:
IP (tos 0xc0, ttl 63, id 53851, len 67) 172.20.201.1 > 10.10.10.165: icmp 47:
172.20.201.1 udp port 27444 unreachable
0x0000  45c0 0043 d25b 0000 3f01 1eda ac14 c901  E..C.[..?.....
0x0010  0a0a 0aa5 0303 86e6 0000 0000 4500 0027  .....E..'
0x0020  5cfd 0000 7f11 5504 0a0a 0aa5 ac14 c901  \....U.....
0x0030  7a67 6b34 0013 47cf 706e 6720 6c34 3461  zgk4..G.png.l44a
0x0040  6473 6c                                     dsl
  
```

**Detect Generated By:**

This detect was generated by snort Win 32 ids version 1.9.1 with the current rule set. The following command was entered to run snort thus enabling me to search through the alert files that were created.

```
Snort -c /path/snort.conf -r /path/2003.12.15.12 -l /path/snort.log
```

Below is the actual alert chosen to investigate further;

```

[**] [1:237:1] DDOS Trin00:Master to Daemon(default pass detected!)
[**][Classification: Attempted Denial of Service] [Priority: 2]
11/18-14:17:52.078334 0:3:47:8C:89:C2 -> 0:50:56:40:0:6D type:0x800
len:0x3C
  
```

```
10.10.10.165:31335 -> 172.20.201.1:27444 UDP TTL:128 TOS:0x0 ID:23805
IpLen:20 DgmLen:39
Len: 11
[Xref => http://www.whitehats.com/info/IDS197]
```

The following is the rule defined by snort, the rule describes that UDP packets from external net directed to our home net destination port 27444, that contains the string l44adsl generate alarm DDOS Trin00 master to Daemon (default pass detected!)

```
alert udp $EXTERNAL_NET any -> $HOME_NET 27444 (msg:"DDOS
Trin00\ :Master to Daemon(default pass detected!)" ; content:"l44adsl" ;
reference:arachnids,197 ; classtype:attempted-dos ; sid:237 ; rev:1 ;)
```

To generate the trace I used windump with the following command on the raw log file 2003.12.15.12

```
windump -r 2003.12.15.12 -nvXes 1500 ip and host 10.10.10.165 and host 172.20.201.1 | more
```

Below is what snort snarf reported when processed, we can determine that there was only one instance of this alert.



**SnortSnarf signature page**  
**DDOS Trin00 Master to Daemon default password attempt**  
SnortSnarf v021111.1

[Signature section \(519\)](#)
[Top 20 source IPs](#)
[Top 20 dest IPs](#)

1 alerts with this signature using input module SnortFileInput, with sources:

- c:\snort\etc\log>alert.ids

Earliest such alert at **15:17:52.078334** on 11/18/2003

Latest such alert at **15:17:52.078334** on 11/18/2003

|   |   |                                |
|---|---|--------------------------------|
| DDOS Trin00 Master to Daemon default password attempt     | <a href="#">1 sources</a>                   | <a href="#">1 destinations</a> |
| Priority: 2   | Classification: Attempted Denial of Service |                                |
| <a href="#">[sid:237]</a> <a href="#">[arachNIDS:197]</a> |   |                                |

**Sources triggering this attack signature**

| Source                       | # Alerts (sig) | # Alerts (total) | # Dsts (sig) | # Dsts (total) |
|------------------------------|----------------|------------------|--------------|----------------|
| <a href="#">10.10.10.165</a> | 1              | 283              | 1            | 9              |

### **Destinations receiving this attack signature**

| Destinations                                   | # Alerts (sig) | # Alerts (total) | # Srcs (sig) | # Srcs (total) |
|--|----------------|------------------|--------------|----------------|
| <a href="http://172.20.201.1">172.20.201.1</a> | 1              | 23               | 1            | 2              |

---

[SnortSnarf](#) brought to you courtesy of [Silicon Defense](#)

Authors: [Jim Hoagland](#) and [Stuart Staniford](#)

See also the [Snort Page](#) by Marty Roesch

Page generated at Thu Apr 22 11:28:33 2004

### **Probability the Source Address was Spoofed**

The probability of the source address being spoofed in this trace is low. The likelihood of the IP address being spoofed is very low, as master needs to communicate orders to daemons on specific ports along with a password and daemons need to respond to their master on a specific port again with a password, hence relying on their address to communicate between each other as demonstrated in the trace. According to RFC 1918 and Internet Assigned Numbers Authority (IANA) 10.x.x.x and 172.x.x.x is reserved for private internets although these address were the result sanitization.

### **Description of the Attack**

Trin00 is distributed denial of service tool (DDOS) used to consume the resources of a host with massive amount of UDP packets essentially flooding the host to the point where it can no longer respond to any other requests. The flood is accomplished by sending UDP packets containing 4 bytes (zeros) to a host which he in turn would reply "ICMP port unreachable message" until there is no or very little bandwidth left. This is accomplished with the co-ordination and implication of many different source hosts focusing on desired target. A Trin00 network is made up of a limited amount of masters and many clients or daemons. The master would be activated by an intruder indicating which hosts to flood and for how long, then the master would activate the many clients to start flooding the specific host IP, that was indicated by the intruder, for a certain duration. CVE has assigned the number CAN-2000-0138 (under review) and their description of DDOS is as follows;

" A system has a distributed denial of service (DDOS) attack master, agent, or zombie installed, such as (1) Trin00 (2) Tribe Flood Network (TFN), Tribe flood Network 2000(TFN2K), (4) stacheldraht, (5) mstream, or (6) shaft."

### **Attack Mechanism**

In this detect the evidence of a possible attack proved to be unsuccessful. The Master, which is typically contacted by the daemons through default port 31335 tried to ping the daemon 172.20.201.1, it should also be noted that the master always communicates with the daemons on default port 27444. We can determine that the master was trying to communicate with the daemon because the communication between the two always contains the password "l44adsl" the password used in this trace is only directed towards a



unix based system. The windows default password is “[.].. Ks” When the master issues commands or the daemon replies to instructions this password is also present. Typically these ports are used in this manner as default, but surely these could possibly change.

```
14:17:52.078334 0:3:47:8c:89:c2 0:50:56:40:0:6d 0800 60:
IP (tos 0x0, ttl 128, id 23805, len 39)
10.10.10.165.31335 > 172.20.201.1.27444:
[udp sum ok] udp 11
0x0000 4500 0027 5cfd 0000 8011 5404 0a0a 0aa5 E..\....T....
0x0010 ac14 c901 7a67 6b34 0013 47cf 706e 6720 ....zgk4..G.png.
0x0020 6c34 3461 6473 6c00 0000 0000 0000 144adsl.....

14:17:52.079597 0:50:56:40:0:6d 0:3:47:8c:89:c2 0800 81:
IP (tos 0xc0, ttl 63, id 53851, len 67) 172.20.201.1 > 10.10.10.165: icmp 47:
172.20.201.1 udp port 27444 unreachable
0x0000 45c0 0043 d25b 0000 3f01 1eda ac14 c901 E..C.[..?.....
0x0010 0a0a 0aa5 0303 86e6 0000 0000 4500 0027 .....E..'
0x0020 5cfd 0000 7f11 5504 0a0a 0aa5 ac14 c901 \....U.....
0x0030 7a67 6b34 0013 47cf 706e 6720 6c34 3461 zgk4..G.png.144a
0x0040 6473 6c dsl
```

In the trace above we can clearly see that the Master attempted to “ping” the daemon by sending the command `png` and password `144adsl`, to the destination port `27444`. The reply in this case indicates that the port `27444` is unreachable indicating that the host is not part of the `trin00` network. Had the host `172.20.201.1` been listening for instructions on the proper port and was part of the `trin00` network the expected reply would have been `pong` along with the password `144adsl` sent to UDP port `31335`. The reason why the above master is trying to ping the daemon is most likely to elicit a response from all it’s daemons to see if they are still alive. It should be noted that when masters establish the relationship with the daemons they store the daemons IP address into a script, which is then used to be part of the `tri00` network. This will be necessary information to launch an attack when the master is instructed by the attacker to activate the attack by instructing the daemons to start flooding a host with udp packets.

## Correlations

Below I included correlating data and links to further complement my analysis.

<http://staff.washington.edu/dittrich/misc/trinoo.analysis>

This is an excerpt from the analysis of David Dittrich which has done a tremendous job at explaining the trinoo network;

"If the trinoo master sends a "png" command to a daemon on port 27444/udp, the daemon will reply to the server that just sent the "png" command by sending the string "PONG" on port 31335/udp:

```
UDP Packet ID (from_IP.port-to_IP.port): 10.0.0.1.1024-192.168.0.1.27444
 45 E 00 . 00 . 27 ' 1A . AE . 00 . 00 . 40 @ 11 . 47 G D4 . 0A . 00 .
00 . 01 .
 C0 . A8 . 00 . 01 . 04 . 00 . 6B k 34 4 00 . 13 . 2F / B7 . 70 p 6E n
67 g 20
 6C l 34 4 34 4 61 a 64 d 73 s 6C l"
```

This demonstrate the command `png` along with the password `ladsl` to the daemon on port 27444

This is the CVE entry that relates information dealing with distributed denial of service attack such as Trinoo.

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2000-0138>

|             |  |
|-------------|--|
| Name        | CAN-2000-0138 (under review)   |
| Description | A system has a distributed denial of service (DDOS) attack master, agent, or zombie installed, such as (1) Trinoo, (2) Tribe Flood Network (TFN), (3) Tribe Flood Network 2000 (TFN2K), (4) stacheldraht, (5) mstream, or (6) shaft.   |
| References  | <ul style="list-style-type: none"><li>• CERT:CA-2000-01</li><li>• CERT:IN-99-04</li><li>• SUN:00193</li><li>• ISS:20000209 Denial of Service Attack using the TFN2K and Stacheldraht programs</li><li>• ISS:20000502 "mstream" Distributed Denial of Service Tool</li><li>• URL:<a href="http://xforce.iss.net/alerts/advise48.php3">http://xforce.iss.net/alerts/advise48.php3</a></li><li>• BUGTRAQ:19991206 Analysis of trinoo</li><li>• BUGTRAQ:19991206 Analysis of Tribe Flood Network</li><li>• BUGTRAQ:19991229 Analysis of "stacheldraht"</li><li>• BUGTRAQ:20000211 DDOS Attack Mitigation</li><li>• BUGTRAQ:20000211 TFN2K - An Analysis</li><li>• BUGTRAQ:20000211 A DDOS proposal.</li><li>• BUGTRAQ:20000429 Re: Source code to mstream, a DDoS tool</li><li>• URL:<a href="http://marc.theaimsgroup.com/?l=bugtraq&amp;m=95715370208598&amp;w=2">http://marc.theaimsgroup.com/?l=bugtraq&amp;m=95715370208598&amp;w=2</a></li><li>• URL:<a href="http://marc.theaimsgroup.com/?l=bugtraq&amp;m=95722093124322&amp;w=2">http://marc.theaimsgroup.com/?l=bugtraq&amp;m=95722093124322&amp;w=2</a></li></ul> |

Internet Security System has a brief description of the master vs slave relationship in the trinoo network.

<http://xforce.iss.net/xforce/xfdb/3570>

The link below describes some of the alert rules that can be used to detect a trin00 network.

<http://www.security-express.com/archives/bugtraq/1999-q4/0254.html>

As always CERT CC can always be relied upon to have a very conclusive description of the attack.

[http://www.cert.org/incident\\_notes/IN-99-07.html](http://www.cert.org/incident_notes/IN-99-07.html)

### **Evidence of Active Targetting**

This scan is not the result of active targeting, we can determine that the host in the raw file is being scanned by the same attacker attempting numerous intrusion attempts, we can determine that this is most likely not the result of active targeting as the masters and daemons already know who they are and which is identified as either being a daemon or master. In order for this attack to work it is required to know IPs as commands and responses are issued to one another, although this seems as an attempt to re-contact the zombie the whole raw file indicates that this is just part of a bigger scan seemingly attempting different vulnerability attempts.

### **Severity**

Severity is calculated in following manner;

$severity = (criticality + lethality) - (system\ countermeasures + network\ countermeasures)$

### **Criticality**

The command PNG appears to be sent to a host ( most likely and end user) that does not have the proper port active as a result I will assign this a 1.

### **Lethality**

This has the potential to be part of a network responsible for disabling or affecting the resources a potential critical element of some bodies network as a result I will assign this a 4.

### **System Countermeasures**

It is difficult to speculate but in this case trin00 is directed towards a Unix machine, you can determine by the password l44adsl and the high ports that are being used along with the random IP ID numbers. Trin00 is an older attack and relies on the same port to conduct it's attack therefore it is likely that the host in question has been appropriately patched, as a result I will assign a 3.

## Network Countermeasures

Snort was able to generate an alarm as a result of running this raw file through it, I can only assume that this network would have an IDS between the host and the outside world. Although it is likely impossible to be accurate in that statement it is easy to conclude that the signatures identifying this as DDOS are very specific ie; ports, passwords and commands which leads me to speculate that there most likely would be sufficient security measure in a network as a result I will assign a 3.

(1+4) – (3+3) = -1

## Defensive Recommendations

Using an IDS with signatures against trin00 would prove to be very effective at identifying traces of any suspected malicious activity, as the methods used to create and communicate within a trin00 network are constant, such as the use of specific ports along with specific passwords and commands would surely trigger an IDS alarm. Ensuring that ports 31335, 27665, 27444 be blocked and that the proper security patches of operating systems remain current are also part of good defensive measures.

## Multiple choice test question

```
14:17:52.078334 0:3:47:8c:89:c2 0:50:56:40:0:6d 0800 60:
IP (tos 0x0, ttl 128, id 23805, len 39)
10.10.10.165.31335 > 172.20.201.1.27444:
[udp sum ok] udp 11
0x0000 4500 0027 5cfd 0000 8011 5404 0a0a 0aa5      E..\....T....
0x0010 ac14 c901 7a67 6b34 0013 47cf 706e 6720      ....zgk4..G.png.
0x0020 6c34 3461 6473 6c00 0000 0000 0000      144adsl.....
```

How can we assume or effectively conclude that this is directed towards a unix type operating system

- A) UDP Ports 31335 and 27444
- B) PNG
- C) 144adsl
- D) []. Ks

Ans-> is C) 144adsl this password is seen associated with unix operating system.

## **References:**

This is possibly the most descriptive work done and covers every aspect of the attack.

<http://staff.washington.edu/dittrich/misc/trinoo.analysis>

Symantec articles offers some guidance on technical details and recommendations

<http://www.symantec.com/avcenter/venc/data/w32.dos.trinoo.html>

Cert has a very comprehensive description of trin00

[http://www.cert.org/incident\\_notes/IN-99-07.html](http://www.cert.org/incident_notes/IN-99-07.html)

The link below describes strategies to defend against UDP denial of service attacks.

<http://cio.cisco.com/warp/public/707/3.html>

Network associates has a good analysis of the attack.

[http://vil.nai.com/vil/content/v\\_98488.htm](http://vil.nai.com/vil/content/v_98488.htm)

Here is an indepth look at describing every steps and technical details of the attack.

<http://www.ece.cmu.edu/~adrian/630/readings/trinoo.analysis.txt.pdf>

## **DETECT #2**

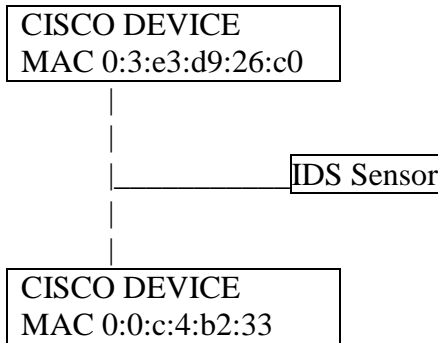
As I was examining raw log files, using snort, that were posted on incident.org/logs/raw website this trace caught my attention. I then decided to further investigate and analyze.

### **Source of trace**

This detect can be found incident.org/logs/raw file 2002.10.18. The following command was used;

```
windump -r 2002.10.18 -nvXes 1500 ip and host 202.108.254.204 and net 170.129 and dst port 1080 | more
```

Although the network cannot for absolute certainty be determine, I have included below a suspected network diagram, I should also note that although the mac address are included these are just as easily spoofed.



```

19:43:59.236507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
IP (tos 0x0, ttl 46, id 52921, len 40) 202.108.254.204.53469 > 170.129.149.62.1080: S
[tcsum ok] 1844151687:1844151687(0) win 1024
0x0000 4500 0028 ceb9 0000 2e06 b51d ca6c fecc E..(.....l..
0x0010 aa81 953e d0dd 0438 6deb 8587 6deb 8587 ...>...8m...m...
0x0020 5002 0400 e6ed 0000 0000 0000 0000 P.....
  
```

```

20:36:23.816507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
IP (tos 0x0, ttl 46, id 29679, len 40) 202.108.254.204.2897 > 170.129.215.53.1080: S
[tcsum ok] 1196016012:1196016012(0) win 1024
0x0000 4500 0028 73ef 0000 2e06 cdf0 ca6c fecc E..(s.....l..
0x0010 aa81 d735 0b51 0438 4749 c18c 4749 c18c ...5.Q.8GL..GL..
0x0020 5002 0400 3fbd 0000 0000 0000 0000 P...?.....
  
```

```

21:28:48.676507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
IP (tos 0x0, ttl 46, id 25248, len 40) 202.108.254.204.14924 > 170.129.252.40.1080: S
[tcsum ok] 661927106:661927106(0) win 1024
0x0000 4500 0028 62a0 0000 2e06 ba4c ca6c fecc E..(b.....L.l..
0x0010 aa81 fc28 3a4c 0438 2774 34c2 2774 34c2 ...(:L.8't4.'t4.
0x0020 5002 0400 450e 0000 0000 0000 0000 P...E.....
  
```

```

22:21:13.116507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60
: IP (tos 0x0, ttl 45, id 52742, len 40) 202.108.254.204.53269 > 170.129.212.139.1080: S
[tcsum ok] 1612303946:1612303946(0) win 1024
0x0000 4500 0028 ce06 0000 2d06 7783 ca6c fecc E..(....-..w..l..
0x0010 aa81 d48b d015 0438 6019 ce4a 6019 ce4a .....8`..J`..J
0x0020 5002 0400 3286 0000 0000 0000 0000 P...2.....
  
```

```

23:13:37.586507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
  
```

```
IP (tos 0x0, ttl 45, id 50551, len 40) 202.108.254.204.55170 > 170.129.190.247.1080: S
[tcp sum ok] 848927323:848927323(0) win 1024
0x0000 4500 0028 c577 0000 2d06 95a6 ca6c fecc      E..(w..-....l..
0x0010 aa81 bef7 d782 0438 3299 9a5b 3299 9a5b      .....82..[2..[
0x0020 5002 0400 038c 0000 0000 0000 0000      P.....
```

```
00:06:02.316507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
IP (tos 0x0, ttl 46, id 64618, len 40) 202.108.254.204.29105 > 170.129.212.89.1080: S
[tcp sum ok] 1145863455:1145863455(0) win 1024
0x0000 4500 0028 fc6a 0000 2e06 4851 ca6c fecc      E..(j....HQ.l..
0x0010 aa81 d459 71b1 0438 444c 7d1f 444c 7d1f      ...Yq..8DL}.DL}.
0x0020 5002 0400 6b0d 0000 0000 0000 0000      P...k.....
<snip
```

**Detect was generated by**

This detect was generated by snort Win 32 ids version 1.9.1.  
The following command was entered to run snort thus enabling me to search through the alert files that were created.

```
Snort -c /path/snort.conf -r /path/2002.10.18 -l /path/snort.log
```

Below is the actual alert that was generated by snort;

```
[**] [1:615:4] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/17-19:43:59.236507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
202.108.254.204:53469 -> 170.129.149.62:1080 TCP TTL:46 TOS:0x0
ID:52921 IpLen:20 DgmLen:40
*****S* Seq: 0x6DEB8587 Ack: 0x6DEB8587 Win: 0x400 TcpLen: 20
[Xref => http://help.undernet.org/proxyscan/]
```

After looking into the folder assigned to the IP that generated the alarms the following was discovered.

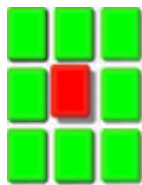
```
[**] SCAN SOCKS Proxy attempt [**]
11/17-19:43:59.236507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
202.108.254.204:53469 -> 170.129.149.62:1080 TCP TTL:46 TOS:0x0
ID:52921 IpLen:20 DgmLen:40
*****S* Seq: 0x6DEB8587 Ack: 0x6DEB8587 Win: 0x400 TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00
.....3.....&...E.
0x0010: 00 28 CE B9 00 00 2E 06 B5 1D CA 6C FE CC AA 81
.(.....l....
0x0020: 95 3E D0 DD 04 38 6D EB 85 87 6D EB 85 87 50 02
.>...8m...m...P.
0x0030: 04 00 E6 ED 00 00 00 00 00 00 00 00 00 00 00 .....
```

Below is the rule that detected the activity. It defines that any external net looking for port 1080 by sending a Syn packet to alert the following message of “SCAN SOCKS Proxy attempt” it is classified as attempted recon and has a sid 615, this is the fourth revision.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN SOCKS Proxy attempt"; flags:S,12; reference:url,help.undernet.org/proxyscan/; classtype:attempted-recon; sid:615; rev:4;)
```

**Probability the source address was spoofed**

It is unlikely that the source address is spoofed. In this case there is an attempt at discovering possible open proxy’s, this is done by sending a Syn packet to a target, which if configured to offer the desired service, will then respond with a Syn Ack indicating that it is ready to initiate the rest of the tcp handshake sequence. As such the attacker that is attempting to discover open proxy’s will need to get a response to effectively determine the state of the desired target. Dshield.org has recognized the IP as belonging to a net in China, of course there is a very slim possibility that the Ip would be spoofed but as mentioned above it is very unlikely if the intention is to get a response back.



**DShield.org**  
Distributed Intrusion  
Detection System


**Protect Yourself ! Protect Others!**  
**Participate in DShield.org**

**IP Info**

---

Check another IP Address:

---

**IP Address:** 202.108.254.204

**HostName:** 202.108.254.204

**DShield Profile:**

|                           |  |
|---------------------------|--|
| Country:                  |  CN |
| Contact E-mail:           | chunguangcanlanxiaobajie@sina.com  |
| AS Number:                | 4808   |
| Total Records against IP: | not processed  |
| Number of                 | select update below  |



|             |    |
|-------------|----|
| targets:    |    |
| Date Range: | to |

### [Update Summary](#)

**Whois:** [Querying whois.apnic.net]  
[whois.apnic.net]  
% [whois.apnic.net node-2]  
% Whois data copyright terms  
<http://www.apnic.net/db/dbcopyright.html>

(cached)

---

[ [Home](#) | [Login](#) | [What's New](#) | [Intro](#) | [Submit](#) | [Clients](#) | [Web Submission](#) | [All Reports](#) |  
[Mail Lists](#) | [Links](#) | [About](#) | [Privacy](#) ]

### **Description of the attack**

The offending IP in this case is 202.108.254.204, which is scanning for the socks proxy server port 1080 on the subnet 170.129.x.x. This scan is done in a manner, which attempts to allow the attacker a degree of stealth. They are attempting to evade possible detection measures in place. The method used is that a single Syn packet every other hour directed at the subnet mentioned above.

It does not seem that this scan may be directed at other networks beyond this one, but that cannot be proven definitively. The scanner seems to be looking for IP's on subnet 170.129 at random, there is no noticeable pattern such as incrementing IP's, and even the computer generated time stamp is random every other hour. Of note is the ending of the time stamp, which consistently ends with 6507. It is strange behavior indeed and a little too odd to be purely coincidental in this analysts opinion. Though I was unable to find any correlation for this odd time behavior.

Furthermore we can speculate that the offending machine is possibly a linux operating system, as the ttl's are 46 throughout, again this also seems a little odd that this remains consistent throughout the trace. Taking this same ttl plus the oddball matching computer timestamps would lead me to believe that this is a automated tool. We should see some small variance given that the scan is done at different times of day.

The windows size is set to 1024, which is not consistent with any of the popular operating systems. We can also determine according to the trace that the maximum segment size is not present and this should definitely be present in the initial syn packet. This could presumably indicate that it is not a random tool at work directed at some random net block but possibly active targeting and a possible indication that the packets

were crafted. There is a [CVE-1999-0291](#) which offers a standardized name and a brief explanation of this type of scan.

### **Attack mechanism**

The attack mechanism is based on the stimulus/response behavior in order to gain some insight on the availability of the socks proxy services. A tcp packet with the Syn flag set is sent in the hopes of receiving a response such as a Syn ack which would indicate that the socks proxy is possibly available for use. The reason why someone would be looking for a response on port 1080 is to first do some reconnaissance work enabling the would be attacker to identify active hosts. Once the attacker has identified which host has port 1080 available, he can then make use of that machines IP to perform malicious activity. Should the proxy server be misconfigured or have weak or missing password he could then direct his attack towards other networks by using the proxy server of that machine camouflaging himself as that IP or simply surf the internet anonymously.

Further to the above mentioned there are some people who dedicate some of their scanning results on websites posting who has these improperly configured proxy, I've included a link below to which serves as an example;

<http://www.rrdb.org/prodb.php?l=en>

There are some tools out there available for use that will automatically hunt for proxies, all you have to do is choose which net block is of interest and the script will do the rest for you, below is an example of one tool.

<http://prdownloads.sourceforge.net/yaph/yaph-0.91.tar.gz?download>

### **Correlations**

Bruce Auburn LOGS: GIAC GCIA Version 3.3 Practical Detect(s) has a very thorough analysis of a sock proxy scan

<http://cert.uni-stuttgart.de/archive/intrusions/2003/06/msg00360.html>

CVE-1999-0291

Description = The wingate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-1999-0291>

This link provides some insight on some of the possible OS using typical settings that you would see in packet traces.

<http://project.honeynet.org/papers/finger/traces.txt>

Snort has a short but very good description that includes some of the most important details when dealing with improperly configured socks server.

<http://www.snort.org/snort-db/sid.html?sid=615>

This link provides an example as a tool that can be used to hunt for open proxies

<http://yaph.sourceforge.net/>

### **Evidence of active targeting**

The targeted machine is scanned every hour, the scanner seems to want to remain undetected. We can determine that there is a syn packet sent approximately every 52min at random machines. This leads me to believe that the scanner is attempting a low and slow scan of our network possibly targeting the network although it is possible that the scan includes other networks outside of our sensor coverage

### **Severity**

Severity is calculated in following manner;

$severity = (criticality + lethality) - (system\ countermeasures + network\ countermeasures)$

### **Criticality**

The attacker is seen scanning random IP's as such we can assume that no reconnaissance work has been done previously. Very little is known about our target hosts and the service they offer, I can only assume that they are normal user machines as such I will assign 2.

### **Lethality**

Should a socks proxy server be available to the scanner, he would have the ability to stage attacks utilizing our IP's as a front and he could possibly have further access to internal networks as a result of having said IP, I will assign a 4.

### **System countermeasures**

Little is known about the hosts network, which leads me to give a less than average mark for system countermeasures, I will assign a 2

### **Network counter measures**

I can only assume that the perimeter device is dropping all syn-ack outbound as there is no evidence of the target host replying, although normal users machines with port 1080 closed would respond with reset-ack, I will assign a 3

### **Defensive recommendations**

If it is required to use a socks proxy server, ensure that only the necessary services are offered such as http. Ensure that only the internal or recognized IP's have access to the proxy server. When reviewing logs verify that only authorized traffic and authorized users are seen using the socks server and of course a strong password is required.

### **Multiple choice question**

Why would a socks proxy server be the target of malicious users.

- A) Attackers can masquerade their IP as being the target host gain access to the network or simply surf the net freely.
- B) Attackers can set up some type of P2P.
- C) Mostly used for gaming purposes
- D) all of the above.

Answer is A) attackers can masquerade themselves, gain access to the network and just simply surf the internet freely.

### **References:**

Microsoft has identified some flaws in the proxy server it describes that some of the winsock servers may incorrectly handle request from remote host resulting in a denial of service. They have made a patch available to rectify this you can find the URL below;  
<http://www.microsoft.com/downloads/details.aspx?familyid=c81688b7-20fb-45eb-bafd-031aod2923e6&displaylang=en>

This is an article that reviews many of type of scans in it's most basic forms including.  
[http://www.auditmypc.com/freescan/readingroom/port\\_scanning.asp](http://www.auditmypc.com/freescan/readingroom/port_scanning.asp)

Example of list of available proxies.  
<http://www.rrdb.org/prodb.php?l=en>

Snort.org description  
<http://www.snort.org/snort-db/sid.html?id=615>

Below the link serve as an example of websites offering open proxies  
<http://www.rrdb.org/prodb.php?l=en>

### **DETECT #3**

#### **Source of Trace**

The data in this trace was obtained from raw file 2003.12.15.10 from incident.org. Below is the network diagram taking in consideration that I have no knowledge of the actual layout I have to base my assumption on the IP and MAC of course this is just a supposition as these can be spoofed.

|                    |            |                 |
|--------------------|------------|-----------------|
| 10.10.10.113       |            | 192.168.17.135  |
| 0:a:95:7c:24:0     | IDS Sensor | 0:50:56:40:0:6d |
| apple computer inc |            | VMWARE inc      |

**Detect was generated by**

This detect was generated by snort Win 32 ids version 1.9.1. The following command was entered to run snort thus enabling me to search through the alert files that were created.

```
Snort -c /path/snort.conf -r /path/2003.12.15.12 -l /path/snort.log
```

Below is the actual alerts chosen to investigate further;

```
[**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection
[**]
11/18-14:14:20.894717 0:A:95:7C:24:0 -> 0:50:56:40:0:6D type:0x800
len:0x3C
10.10.10.113:59194 -> 192.168.17.135:886 TCP TTL:52 TOS:0x0 ID:44944
IpLen:20 DgmLen:40
***** Seq: 0x0 Ack: 0x0 Win: 0x400 TcpLen: 20
```

The following is the rule defined by snort, the rule describes that packets sent from external IP with seq and ack 0 along with all the tcp flags set to 0 display the message SCAN NULL.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN NULL"; flags:0;
seq:0; ack:0; reference:arachnids,4; classtype:attempted-recon; sid:623; rev:1;)
```

After looking into the folder assigned to the IP that generated the alarms the following was discovered.

```
[**] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**]
11/18-14:14:44.533812 0:A:95:7C:24:0 -> 0:50:56:40:0:6D type:0x800
len:0x3C
10.10.10.113:59195 -> 192.168.17.135:9992 TCP TTL:44 TOS:0x0 ID:33402
IpLen:20 DgmLen:40
***** Seq: 0x0 Ack: 0x0 Win: 0x400 TcpLen: 20
0x0000: 00 50 56 40 00 6D 00 0A 95 7C 24 00 08 00 45 00
.PV@m...|$...E.
0x0010: 00 28 82 7A 00 00 2C 06 25 AC 0A 0A 0A 71 C0 A8
.(z...%...q..
0x0020: 11 87 E7 3B 27 08 00 00 00 00 00 00 00 00 50 00
...i'.....P.
0x0030: 04 00 B6 F6 00 00 55 55 55 55 55 55 .....
```

To generate the trace I used windump with the following command on the raw log file 2003.12.15.10

```
C:\snort\snort_handson\section1>windump -r 2003.12.15.10 -nevXs 0 ip and host
10.10.10.113 and host 192.168.17.135
```

```
15:14:20.061563 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: IP (tos 0x0, ttl 41, id 10280,
len 40) 10.10.10.113.59194 > 192.168.17.135.1486: .
```

```
[tcp sum ok] win 2048
```

```
0x0000 4500 0028 2828 0000 2906 82fe 0a0a 0a71 E..(((..).....q
0x0010 c0a8 1187 e73a 05ce 0000 0000 0000 0000 .....:.....
0x0020 5000 0800 d431 0000 5555 5555 5555 P....1..UUUUUU
```

```
15:14:20.140541 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: IP (tos 0x0, ttl 45, id 33332,
len 40) 10.10.10.113.59194 > 192.168.17.135.12000: . [tcp sum ok] win 2048
```

```
0x0000 4500 0028 8234 0000 2d06 24f2 0a0a 0a71 E..(.4.-.$....q
0x0010 c0a8 1187 e73a 2ee0 0000 0000 0000 0000 .....:.....
0x0020 5000 0800 ab1f 0000 5555 5555 5555 P.....UUUUUU
```

```
15:14:20.140630 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: IP (tos 0x0, ttl 49, id 62585,
len 40) 10.10.10.113.59194 > 192.168.17.135.3086: . [tcp sum ok] win 2048
```

```
0x0000 4500 0028 f479 0000 3106 aeac 0a0a 0a71 E..(.y..1.....q
0x0010 c0a8 1187 e73a 0c0e 0000 0000 0000 0000 .....:.....
0x0020 5000 0800 cdf1 0000 5555 5555 5555 P.....UUUUUU
```

```
15:14:20.140712 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: IP (tos 0x0, ttl 58, id 49136,
len 40) 10.10.10.113.59194 > 192.168.17.135.108: . [tcp sum ok] win 3072
```

```
0x0000 4500 0028 bff0 0000 3a06 da35 0a0a 0a71 E..(.....:5...q
0x0010 c0a8 1187 e73a 006c 0000 0000 0000 0000 .....:1.....
0x0020 5000 0c00 d593 0000 5555 5555 5555 P.....UUUUUU
```

```
15:14:20.140767 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: IP (tos 0x0, ttl 40, id 11951,
len 40) 10.10.10.113.59194 > 192.168.17.135.987: . [tcp sum ok] win 1024
```

```
0x0000 4500 0028 2eaf 0000 2806 7d77 0a0a 0a71 E..(....(.}w...q
0x0010 c0a8 1187 e73a 03db 0000 0000 0000 0000 .....:.....
0x0020 5000 0400 da24 0000 5555 5555 5555 P....$.UUUUUU
```

```
>snip
```

I then decided to run the file through snort snarf v021111.1 this would give me a better picture of the whole file enabling me to browse at the html file created in an organised fashion.

As can be witnessed below the null scan generated 1986 alerts, which I then decided to explore further.



**SnortSnarf signature page**

**SCAN NULL**

**SnortSnarf v021111.1**

1986 alerts with this signature using input module SnortFileInput, with sources:

- c:\snort\etc\log>alert.ids2

Earliest such alert at **15:14:20.061563** on 11/18/2003

Latest such alert at **15:15:39.530319** on 11/18/2003

|   |  |                                |
|---|--|--------------------------------|
| SCAN NULL   | <a href="#">1 sources</a>                  | <a href="#">1 destinations</a> |
| Priority: 2   | Classification: Attempted Information Leak |                                |
| <a href="#">[sid:623]</a> <a href="#">[arachNIDS:4]</a> |  |                                |

---

**Sources triggering this attack signature**

| Source                       | # Alerts (sig) | # Alerts (total) | # Dsts (sig) | # Dsts (total) |
|------------------------------|----------------|------------------|--------------|----------------|
| <a href="#">10.10.10.113</a> | 1986           | 1990             | 1            | 1              |

---

**Destinations receiving this attack signature**

| Destinations                   | # Alerts (sig) | # Alerts (total) | # Srcs (sig) | # Srcs (total) |
|--------------------------------|----------------|------------------|--------------|----------------|
| <a href="#">192.168.17.135</a> | 1986           | 1990             | 1            | 1              |

---

[SnortSnarf](#) brought to you courtesy of [Silicon Defense](#)

Authors: [Jim Hoagland](#) and [Stuart Staniford](#)

See also the [Snort Page](#) by Marty Roesch

As can be seen we can determine that there is only one source and one destination that triggered 60 alerts, below are a snippet of the alarms.

```
[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-15:17:47.459375 10.10.10.113:59194 -> 192.168.17.135:663
TCP TTL:46 TOS:0x0 ID:26762 IpLen:20 DgmLen:40
***** Seq: 0x0 Ack: 0x0 Win: 0xC00 TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]
```

```
[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-15:17:47.459450 10.10.10.113:59194 -> 192.168.17.135:1663
TCP TTL:49 TOS:0x0 ID:45284 IpLen:20 DgmLen:40
***** Seq: 0x0 Ack: 0x0 Win: 0x800 TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]
```

### **Probability the source address was spoofed**

It is unlikely that the source address is spoofed, the basics behind this type of attack is to discover what ports are open on a host, therefore part of the intelligence gathering the attacker is expecting a response from it's probe in this case a non response is a possible indication that the port is open. That been said it is however possible that the IP might be spoofed although this is unlikely and would obviously serve no cause to an attacker to spoof an IP to conduct this type of probe. It should be noted that it is however possible that the attacking machine has been compromised and has been taken over by an attacker and he in-turn is conducting his reconnaissance work with said machine, of course this is only speculation but this scan is very noisy and there is no attempt at camouflaging this activity.

### **Description of the attack**

This is clearly a scan to which its purpose is to enumerate open ports of host 192.168.17.135. This type of scan has all of its tcp flag set to 0( urg, ack, push, rst, syn fin). The sequence number and the ack are both set to zero as well, the attacker is scanning the target host very quickly and is quite noisy, we can determine that the attacker is scanning about 20 ports per second. RFC 793 describes that closed ports have to reply with a RST and open ports will however not respond at all. This type of recognition is sometimes performed as a prelude to an attack to further determine which service are available on a host and in turn the attacker can target a specific exploit associated to a service that is running on the target host. There is no CVE number assigned for this type of scan

### **Attack mechanism**

We can determine that the packets sent by 10.10.10.113 are crafted, as packets should never have all control bits set to 0 furthermore the seq numbers and ack numbers are also set to 0. The speed at which the packets are sent along with the UUUUUU in the ascii context throughout the trace along with the illegal packets indicates that a tool is used to scan. The trace indicates that there are only two ephemeral source ports they 59195 and 59194 this is further indication of a tool at work since these should increment. The IP ID numbers can be seen as incrementing randomly this would lead me to assume that the attacker is using a unix type of operating systems as windows OS would increment by 1. The most popular tool used by Unix type of machines to perform scans that has the above particularities would indicate that this scanning tool is possibly nmap.

### **Correlations**



The link below is a practical by Gregory Lalla LOGS: GIAC GCIA Version 3.4 Practical Detect in his paper goes a great details and provides some test results of NMAP

<http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00007.html>

The snort signature database has a good description of what is the main purpose of a null scan, it includes impact and detailed information.

<http://www.snort.org/snort-db/sid.html?id=623>

Iss has a short description of the properties of a null scan.

[http://www.iss.net/security\\_center/advice/Intrusions/2000309/default.htm](http://www.iss.net/security_center/advice/Intrusions/2000309/default.htm)

This site by Security forums offers a tutorial of nmap and it's many different type of scans.

<http://www.security-forums.com/forum/viewtopic.php?t=7872&postdays=0&postorder=asc&start=0>

### **Evidence of active targeting**

We can assume that the target host is actively being targeted during this scanning session. It seems as though our offender is specifically looking for a non-response from our networks enabling him to determine the open port. The scan being performed is very noisy and is actively looking for well-known ports as well as ephemeral port.

### **Severity**

Severity is calculated in following manner;

$severity = (criticality + lethality) - (system\ countermeasures + network\ countermeasures)$

### **Criticality**

Not knowing the network personally I will have to assume that the this is an end user workstation although the scan includes ephemeral ports and well known ports and possibly not a critical asset as such I will take a middle of the road approach and assign a 3 as criticality.

### **Lethality**

We have determine that there was only one reset sent by the target host. Taking in consideration the large amount of packets sent to the hosts and the one reply we have to assume that the attacker did not get much valuable information from our host, indicating that the devices are doing a good job at filtering these illegal packets, as such I will assign a 2.

### **System countermeasures**

The technique being used here is an older one, we can assume that there is a perimeter device blocking all attempts and that most up to date operating systems are protected against this type of recognition, as a result I will assign a 4.

### **Network counter measures**

It is safe to assume that the attack got as far as the IDS, it is unlikely that it would have made it through the firewall and less likely that the target network would suffer negative consequences as a result of this scan, we can assign a 4.

### **Defensive recommendations**

Monitoring the firewall and ids logs and ensuring that all systems are current with the latest updates will ensure some defensive measures. Illegal tcp flag combination should be used in penetration test to determine that routers, firewalls, ids and other defensive measure are in place and that they do in fact all detect or drop this type of activity.

This type of test will ensure that nothing gets missed.

**(2+4) - (4+4) = -2**

### **Multiple choice question**

What would be the purpose of sending an illegal tcp packet with all combination flags set to zero.

- a) Random error
- b) Enumerating active services
- c) Distributed denial of service attack
- d) All the above

The correct answer is b), should an illegal packet with all the combination flag set to zero be sent to a target host port that is closed and that has not been patched for such a thing it would then respond with a rst.

### **References:**

Snort website offers a good description of the null scan.

<http://www.snort.org/snort-db/sid.html?id=623>

Below is a link to nmap the possible tool at use, it describes the many different type of scanning methods including the null scan.

[http://www.insecure.org/nmap/nmap\\_doc.html#fin](http://www.insecure.org/nmap/nmap_doc.html#fin)

An article about nmap and how it's many uses.

<http://computercops.biz/modules.php?name=nmap>

Arachnids intrusion database describes the null scan and it's features can be found in the link that I've included below

<http://www.whitehats.com/info/IDS4>

An article describing port scans

<http://security-protocols.com/modules.php?name=News&file=article&sid=630>

### **Posted detect**

Top three question for peer review on incident.orgs mailing list.

06 Jun 04

Donald.Smith@qwest.com GCIA

<http://pgp.mit.edu:11371/pks/lookup?op=get&search=0xAF00EDCC>

I reserve the right to be wrong but don't exercise it too often.

Can you spoof a mac address outside your LAN?

What would be the value of spoofing the mac outside your lan?

->> absolutely, tools like smac can help do this. The reason why you would want to spoof mac can be to access wireless network that it's authentication and authorization is based on mac or simply for legitimate reason test your network. <http://www.klcconsulting.net/smac/>

I have cut this to the 1st packet and 4th.

Look at the source port and the ID. Notice anything unusual?

The ID's and ports nearly match. I suspect this was done by a tool that had a flawed psuedo random generator filling the ID and source port number.. Crafted!?!

->>>>I agree that the packets might be crafted, although there is nothing that can actually pinpoint the fact that this might be a flawed pseudo random generator filling the ID and source port numbers... as there is no mathematical evidence to go by, although it is not entirely impossible, I have to concentrate on hard facts and would not want to make a wrong assumption based on speculation.

Now if you look at the times of the packets they are mostly 52mins apart. So a scheduled scan (you mention this below in the proper spot but the time corralate better then you imply.

->>>> good call on the time, I should have been more attentive to the small details.

```
> 19:43:59.236507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
> IP (tos 0x0, ttl 46, id 52921, len 40) 202.108.254.204.53469 >
> 170.129.149.62.1080: S [tcp sum ok] 1844151687:1844151687(0) win 1024
> 0x0000 4500 0028 ceb9 0000 2e06 b51d ca6c fecc
> E.(.....1..
> 0x0010 aa81 953e d0dd 0438 6deb 8587 6deb 8587
> ...>...8m...m...
> 0x0020 5002 0400 e6ed 0000 0000 0000 0000
P.....
```

```
>
<SNIP>
>
> 22:21:13.116507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60
> : IP (tos 0x0, ttl 45, id 52742, len 40) 202.108.254.204.53269 >
> 170.129.212.139.1080: S [tcp sum ok] 1612303946:1612303946(0) win
1024
> 0x0000 4500 0028 ce06 0000 2d06 7783 ca6c fecc
> E..(....-.w..l..
> 0x0010 aa81 d48b d015 0438 6019 ce4a 6019 ce4a
> .....8`...J`...J
> 0x0020 5002 0400 3286 0000 0000 0000 0000
P...2.....
<SNIP>
```

Network counter measures

```
> I can only assume that the perimeter device is dropping all
> syn-ack outbond as there is no evidence of the target host
> replying, I will assign a 3
```

Not the assumption I would make. Your assuming systems are responding on 1080 but the responses are being dropped? Based on your assumption above these are normal user machines do those typically have 1080 open?

->>>>>You're Wright should these port be closed on normal user machines you would expect a reset as a response,

```
> Multiple choice question
>
> Why would a socks proxy server be the target of malicious users.
>
> A) Attackers can masquerade their IP as being the
> target host
> IP.
> B) Attackers can gain further access to the target hosts
> network.
> C) Attackers can surf the web freely.
> D) All of the above
>
> Answer is D) all of the above.
```

The notes for this assignment make it clear that there should be a single answer (not all the above). Your likely to loose a few points if you don't modify this.

- > Multiple choice question
- >
- > Why would a socks proxy server be the target of malicious users.
- >
- > A) Attackers can masquerade their IP as being the target host gain access to the network or simply surf the net freely.
- > B) Attackers can set up some type of P2P.
- > C) Mostly used for gaming purposes
- > D) all of the above.
- >
- Answer is A) attackers can masquerade themselves, gain access to the network and just simply surf the internet freely.

### **Assignment #3 Analyze This**

#### **Executive Summary**

You will find below the summary of my findings with reference to the log files that were analysed. These included 5 days worth of Out of spec, Scan, and alarms. The files are composed of what type of activity is seen on the universities network. As a result we can then determine what are the threat's to our network and what is acceptable use according to the universities policies.

It has been determined that some of the universities network computers are infected with malware in this case win32.agobot. Also noted was the use of p2p software, which in light of recent legislation in the United States could result in legal action against the University. This is a serious issue which needs to be addressed immediately.

Of note are scans originating from outside the universities network. These are seen actively probing in an attempt to fingerprint some of the resources on the university network. Though scans are largely background internet noise there were some scans of concern such as the following. There was a probable attempt to infect PC's with the RPC DCOM exploit. It is of some concern due to the exploit resulting in high level access. IP 213.180.193.68 should be blocked at the perimeter device as it is seen as being a very noisy scanner that is directly responsible for an extraordinary amount of scans to port 61872. This IP according to Dshield has over 241,000 complaints launched against it.

Also troubling is IP 192.168.150.226 (this is an internal address) seen scanning for some of the more better known trouble ports on the internal network. This has caused some 1900 alarms over the last few days. This IP is possibly experimenting with discovery

tools, or this could possibly be related to IT staff hard at work proactively probing the internal network for weaknesses.

Of particular note is the amount of traffic that was generated as a result of possible DNS activity from IP 192.168.1.3 and 192.168.1.4. (assumed DNS servers based on the IP address). These two IP's are responsible for over 36,000 scans alerts. While likely that this is normal DNS activity this should be confirmed by local staff.

Lastly it would be beneficial for the University to do a complete audit of all services being offered. Once this is done only the required ports could be left open on the border router. Secondly there should be an attempt made at tightening up the IDS signatures in an effort to reduce false positives. Overall the security of the university is in decent shape and with some tweaking can be made far more resilient.

### **Methodology**

All three file types (alert, scan, oos) were concatenated into one file using the win32 port of cat.

```
ie: cat.exe alert1 alert2 alert3 alert4 alert5 > alert_file
```

The win32 port of sed was used to remove instances of MY.NET and 130.85 in the above noted file types. They were replaced with 192.168 so that snortsnarf could parse the file.

```
ie: sed.exe s/MY.NET/192.168/g alert.file > alert_fileout
```

[Ricky Smith's](#) GIAC practical included a perl script called parse-oos.pl which was then used to massage the OOS file so that snortsnarf could parse the data.

Files for the above noted 3 types (alert, scan, oos) were quite large with the exception of the OOS file type. Snortsnarf was having issues processing this data even on a 1u dual cpu with 2Gb of ram.

Bearing this in mind the below noted was done to pare down the large amount of data represented in these files;

Scan file (original file was 26MB)

```
grep -v 192.168.75.84 as it was src port 4899 (win32.agobot activity)
```

```
grep -v 213.180.193.68 as it was src port 61872 and likely p2p activity
```

this left 3.x MB's in scan\_file3 file (which is composed of that file types 5 days worth of traffic)

Scan files have now been successfully parsed in snortsnarf.

Alert File (original file size was 99.2MB)

Noted a massive amount of spp\_portscan acty in the alert\_file. Seen as this is portscan activity I elected to grep -v this traffic out of the file. This resulted in an original file size of 99.2MB being whittled down to 12.7MB. Now with a more realistic file size snortsnarf has parsed the file successfully.

OOS File

No pruning required as these files were small to begin with

| OOS                   | Scans        | Alerts       |
|-----------------------|--------------|--------------|
| OOS_Report_2004_04_25 | Scans.040425 | Alert.040420 |
| OOS_Report_2004_04_26 | Scans.040426 | Alert.040421 |
| OOS_Report_2004_04_27 | Scans.040427 | Alert.040422 |
| OOS_Report_2004_04_28 | Scans.040428 | Alert.040423 |
| OOS_Report_2004_04_29 | Scans.040429 | Alert.040426 |

Top Ten talkers (in terms of overall volume)

The top ten alerts are based on the frequency of alerts, below you will find a list of the said top alerts.

| Priority | Signature (click for sig info)     | # Alerts | # Sources | # Dests | Detail link             |
|----------|------------------------------------|----------|-----------|---------|-------------------------|
| N/A      | EXPLOIT x86 NOOP                   | 38903    | 2084      | 916     | <a href="#">Summary</a> |
| N/A      | 192.168.30.4 activity              | 35289    | 313       | 1       | <a href="#">Summary</a> |
| N/A      | High port 65535 tcp - possible Red | 19471    | 119       | 154     | <a href="#">Summary</a> |

|     |  |       |     |     |                         |
|-----|--|-------|-----|-----|-------------------------|
|     | Worm - traffic                             |       |     |     |                         |
| N/A | 192.168.30.3 activity                      | 15900 | 197 | 1   | <a href="#">Summary</a> |
| N/A | SMB Name Wildcard                          | 8627  | 61  | 639 | <a href="#">Summary</a> |
| N/A | Tiny Fragments - Possible Hostile Activity | 4412  | 5   | 18  | <a href="#">Summary</a> |
| N/A | RFB - Possible WinVNC - 010708-1           | 2355  | 15  | 17  | <a href="#">Summary</a> |
| N/A | Null scan!                                 | 1936  | 89  | 54  | <a href="#">Summary</a> |
| N/A | NMAP TCP ping!                             | 869   | 218 | 67  | <a href="#">Summary</a> |
| N/A | Possible trojan server activity            | 419   | 48  | 52  | <a href="#">Summary</a> |

### Number 1 Top Talker

| Signature (click for sig info) | # Alerts |
|--------------------------------|----------|
| EXPLOIT x86 NOOP               | 38903    |
|                                |          |

[NOOP](#) is one of a series of idempotent instructions used to fill an address space. These are used as “fillers” when the exact location of certain exploit ie: return address cannot or has not been ascertained. As such these NOOP instructions are used to provide some type of “slide” with which the attackers exploit code can slide into ie: have a bigger attack window if you will. The aim is to exploit insecure coding practices.

False positive can be high as these signatures can trigger on binary files being transferred or just daily web traffic.

### Defensive recommendations

If budget permits the purchase of an application layer firewall would greatly mitigate this type of attack. Barring this it is imperative that the IDS analyst become familiar with legitimate exploit code. By that I mean actively go out and download freely available exploit code, which they will then compile and test in a lab environment. This will allow them to see the exploit itself at the packet level thereby seeing the 0x90 in the hex portion of the packet.

<http://www.snort.org/snort-db/sid.html?id=648>



04/20-14:42:47.165784 [\*\*] [EXPLOIT x86 NOOP](#) [\*\*] [220.197.192.39:55638](#) -> [192.168.29.138:80](#)

04/20-14:42:47.470474 [\*\*] [EXPLOIT x86 NOOP](#) [\*\*] [220.197.192.39:55638](#) -> [192.168.29.138:80](#)

04/20-14:42:47.471606 [\*\*] [EXPLOIT x86 NOOP](#) [\*\*] [220.197.192.39:55638](#) -> [192.168.29.138:80](#)

04/20-14:42:47.473529 [\*\*] [EXPLOIT x86 NOOP](#) [\*\*] [220.197.192.39:55638](#) -> [192.168.29.138:80](#)

### Number 2 Top Talker

| Signature (click for sig info)        | # Alerts |
|---------------------------------------|----------|
| <a href="#">192.168.30.4 activity</a> | 35289    |

The below noted alert was almost excluded from our “Top Ten Talkers” list because it is not an alert as evidenced by the packets seen below. Unlike the other alert above we have no definitive alert to attach to this, but rather simply the IP address to go with. After viewing the “alert” it was noted that 313 sources were going to one internal IP address.

It is probable that syn packets are being sent to the one host, however this cannot be verified, as we do not have access to the original binary log. By looking at the alerts down below we can see that this is directed to port 80 and 51443 which are used by [Novell](#) Secure iFolder applications.

[Erik Montcalm GCIA](#) practical describes the same event (probes to port 80 and 51443) [Tom King GCIA](#) practical indicates that by doing a google search for “[University of Maryland ifolder](#)” further credence to the theory (of port 80/51443 being iFolder acty) surfaced. This alarm is probably due to outside IP’s trying to access Novell Network iFolder. This service is designed to let users have access from anywhere to the management software. All of the above information was derived by simple google searches, which revealed the noted correlation from the GCIA analysts that I hyperlinked to.

Defensive recommendations include an access list at the perimeter device to include only the desired ports you wish to have open. Secondly you can also filter or restrict access to certain applications by filtering IP addresses at the router. At a minimum it is imperative to ensure that the application is fully patched and kept up to date.

04/22-20:06:03.112345 [\*\*] [192.168.30.4 activity](#) [\*\*] [68.33.49.146:1041](#) -> [192.168.30.4:80](#)

04/22-20:06:03.182152 [\*\*] [192.168.30.4 activity](#) [\*\*] [68.33.49.146:1041](#) ->

[192.168.30.4:80](#)

04/22-20:06:06.332832 [\*\*] [192.168.30.4 activity](#) [\*\*] [68.33.49.146:1050](#) -> [192.168.30.4:51443](#)

04/22-20:06:06.426585 [\*\*] [192.168.30.4 activity](#) [\*\*] [68.33.49.146:1050](#) -> [192.168.30.4:51443](#)

### Number 3 Top Talker

| Signature (click for sig info)                    | # Alerts |
|---|----------|
| High port 65535 tcp - possible Red Worm - traffic | 19471    |

The Red Worm aka [Adore Worm](#) is composed of a series of vulnerabilities affecting services such as LPRng, BIND, and rpc-statd. If the worm gains access to a machine via one of these vulnerabilities it will replace programs such as ps with a trojanized version. Also a program called “icmp” will be inserted. This “icmp” program will listen for a specific ICMP packet and once it is received it will open a backdoor on TCP port 65535.

Defensive recommendations for this attack would be to ensure that you have your linux distribution up to date and fully patched. Though this attack has been around for several years many new people to linux sometimes start with an older linux distribution, which is vulnerable to this attack. Hence the importance of fully updating, and patching your linux distro prior to putting it online.

04/22-01:37:03.210282 [\*\*] [High port 65535 tcp - possible Red Worm - traffic](#) [\*\*] [192.168.43.8:3883](#) -> [64.12.24.35:65535](#)

04/22-01:37:03.538440 [\*\*] [High port 65535 tcp - possible Red Worm - traffic](#) [\*\*] [192.168.43.8:3883](#) -> [64.12.24.35:65535](#)

04/22-01:37:05.943031 [\*\*] [High port 65535 tcp - possible Red Worm - traffic](#) [\*\*] [192.168.43.8:3883](#) -> [64.12.24.35:65535](#)

04/22-01:37:09.147722 [\*\*] [High port 65535 tcp - possible Red Worm - traffic](#) [\*\*] [192.168.43.8:3883](#) -> [64.12.24.35:65535](#)

### Number 4 Top Talker

| Signature (click for sig info) | # Alerts |
|--------------------------------|----------|
| 192.168.30.3 activity          | 15900    |

The 15900 alarms were all as a result of alerts with a destination port of 524. This port is used for all communications between Netware 5 clients and servers, and can also use both tcp and udp for connections.

Defensive recommendation would include having the network administrator review the rules that are triggering these alarms. Further investigation of IP 192.168.30.3 would be a worthwhile effort as this could possibly have been compromised. It is difficult to speculate what the purpose of the said machine is in this instance as only limited amount of information has been made available to me.

Port 524, 1033, 3019

|  |
|--|
| 04/21-08:17:59.641574 [ ** ] <a href="#">192.168.30.3 activity</a> [ ** ] <a href="#">131.92.177.18:1033</a> -> <a href="#">192.168.30.3:524</a> |
| 04/21-08:17:59.651378 [ ** ] <a href="#">192.168.30.3 activity</a> [ ** ] <a href="#">131.92.177.18:1033</a> -> <a href="#">192.168.30.3:524</a> |
| 04/21-08:17:59.771753 [ ** ] <a href="#">192.168.30.3 activity</a> [ ** ] <a href="#">131.92.177.18:1033</a> -> <a href="#">192.168.30.3:524</a> |
| 04/21-08:19:02.362353 [ ** ] <a href="#">192.168.30.3 activity</a> [ ** ] <a href="#">131.92.177.18:1033</a> -> <a href="#">192.168.30.3:524</a> |

### Number 5 Top Talker

| Signature (click for sig info) | # Alerts |
|--------------------------------|----------|
| SMB Name Wildcard              | 8627     |

The SMB Name Wildcard alert was recorded 8627 times over five days. All these alerts came from the universities host 192.168.11.4 from port 137 and were directed towards other external IP's addresses port 137. Though why we are seeing port 137 from the university network going out to an external network is unknown and requires further investigation. This is normally only ever used for internal networks.

|   |
|---|
| 04/20-14:27:15.956298 [ ** ] <a href="#">SMB Name Wildcard</a> [ ** ] <a href="#">192.168.11.4:137</a> -> <a href="#">210.120.128.117:137</a> |
| 04/20-14:27:18.944981 [ ** ] <a href="#">SMB Name Wildcard</a> [ ** ] <a href="#">192.168.11.4:137</a> ->                                     |

[210.120.128.117:137](#)

04/21-10:21:11.759759 [\*\*] [SMB Name Wildcard](#) [\*\*] [192.168.11.4:137](#) -> [210.120.128.117:137](#)

04/21-10:21:39.463651 [\*\*] [SMB Name Wildcard](#) [\*\*] [192.168.11.4:137](#) -> [210.120.128.117:137](#)

Port 137 is the NetBIOS Name Service, and is used to query a computers name, which is composed of 15 characters. If the name is not 15 characters in length the remaining spaces will be “white spaced”. A 16<sup>th</sup> character is also assigned to the computer by the operating system itself to denote its function or role ie: PDC, WINS, workstation.

As mentioned on <http://www.whitehats.com/info/ids177> “this can be considered background noise”. We can consider this background noise as this is normal traffic from port 137 to port 137 emanating from the universities host, further investigation would surely be required if indication of outside hosts were to asking for port 137 information.

Defensive recommendations would be to ensure that unless absolutely necessary that all external traffic destined for internal port 137 be blocked at the border router. This port is only used for internal traffic realistically, and should not be open to the world as it were. As mentioned above why this address is going outbound on port 137 to an exterior address on port 137 remains unknown and requires further scrutiny.

Correlation information was found using google and [Brian Sheffler GCIA # 531](#)  
[www.giac.org/practical/gcia/al\\_williams.gcia.pdf](http://www.giac.org/practical/gcia/al_williams.gcia.pdf)  
<http://www.whitehats.com/info/ids177>  
[http://www.google.ca/search?q=cache:hpcVyqUEwd4J:www.giac.org/practical/Joe\\_Ellis\\_GCIA.doc+SMB+Name+Wildcard+8627+&hl=en](http://www.google.ca/search?q=cache:hpcVyqUEwd4J:www.giac.org/practical/Joe_Ellis_GCIA.doc+SMB+Name+Wildcard+8627+&hl=en)

### Number 6 Top Talker

| Signature (click for sig info)                             | # Alerts |
|--|----------|
| <a href="#">Tiny Fragments - Possible Hostile Activity</a> | 4412     |

Tiny fragments the alarm is an indication of a fragmentation attack. This is indicative of someone trying to send an exploit through over various packet fragments. This alert should be looked at closely.

Defensive recommendation include blocking all fragmented packets at the border router or perimeter devices.

04/20-14:21:48.455880 [\*\*] [Tiny Fragments - Possible Hostile Activity](#) [\*\*]  
[209.164.32.205](#) -> [192.168.69.254](#)

04/20-14:21:54.477794 [ \*\* ] [Tiny Fragments - Possible Hostile Activity](#) [ \*\* ]  
[209.164.32.205](#) -> [192.168.69.254](#)

04/20-14:22:30.456618 [ \*\* ] [Tiny Fragments - Possible Hostile Activity](#) [ \*\* ]  
[209.164.32.205](#) -> [192.168.69.254](#)

04/20-14:23:18.469813 [ \*\* ] [Tiny Fragments - Possible Hostile Activity](#) [ \*\* ]  
[209.164.32.205](#) -> [192.168.69.254](#)

### Number 7 Top Talker

|                                  |      |
|----------------------------------|------|
| RFB - Possible WinVNC - 010708-1 | 2355 |
|----------------------------------|------|

WinVNC is a well known hacking tool that is planted on a victims machines so that the hacker can remotely manage that machine. This can be done via the command line interface as well (DOS prompt or Xterm), which makes it doubly dangerous. One does not have to have a user execute a picture say with a double extension on it to have this program installed. While (WinVNC) is not used to gain entry onto a machine this program is however used to further ones control over a machine, which has been breached via other means.

04/20-16:44:49.693534 [ \*\* ] [RFB - Possible WinVNC - 010708-1](#) [ \*\* ] [24.43.50.166:1404](#)  
-> [192.168.70.156:5900](#)

04/20-16:44:52.899889 [ \*\* ] [RFB - Possible WinVNC - 010708-1](#) [ \*\* ] [24.43.50.166:1412](#)  
-> [192.168.70.156:5900](#)

04/20-16:44:56.492706 [ \*\* ] [RFB - Possible WinVNC - 010708-1](#) [ \*\* ] [24.43.50.166:1416](#)  
-> [192.168.70.156:5900](#)

04/20-16:44:59.673254 [ \*\* ] [RFB - Possible WinVNC - 010708-1](#) [ \*\* ] [24.43.50.166:1464](#)  
-> [192.168.70.210:5900](#)

### Number 8 Top Talker

|            |      |
|------------|------|
| Null scan! | 1936 |
|------------|------|

This type of activity is effectively used to probe for open ports or to gather information in order to provide some insight on the type of operating system which is more commonly known as os fingerprinting. The idea is to send a crafted packet with all it's top flag(urg,ack, push,rst,syn,fin) set to zero with the ack and sequence number set to zero as well. This according to RFC 793 will elicit the closed ports to reply with a RST and open ports will not respond to such probes

A null scan is primarily used in the hopes of evading an IDS such as Snort. This is no longer true. Snort, as we can see easily picked up this scan type. Why as well the scanner

is using a src and dst port of 0 for the most part is unknown. There is little to be gained by doing so outside of the fact that you will verify if the machine is indeed turned on or not. Noted below in the snortsnarf snippet is the random packets with oddball ports in them such as the below noted port 14 to port 58552. Outside of a possible peer to peer connection attempt (does not make sense in this case p2p connection) I could not find a scenario for which these packets would fit.

Defensive recommendation for this type of scan is having an IDS in place to detect it. In reality this type of scan or other scans for that matter are of little danger in and of themselves. It is pretty much normal background internet noise really. The only thing is that it might indicate a possible upcoming exploit attempt.

|   |
|---|
| 04/22-18:43:28.144094 [**] <a href="#">Null scan!</a> [**] <a href="#">61.48.8.56:0</a> -> <a href="#">192.168.112.209:0</a>      |
| 04/22-18:43:28.148583 [**] <a href="#">Null scan!</a> [**] <a href="#">61.48.8.56:0</a> -> <a href="#">192.168.112.209:0</a>      |
| 04/22-18:43:28.150688 [**] <a href="#">Null scan!</a> [**] <a href="#">61.48.8.56:14</a> -> <a href="#">192.168.112.209:58552</a> |
| 04/22-18:43:28.152945 [**] <a href="#">Null scan!</a> [**] <a href="#">61.48.8.56:0</a> -> <a href="#">192.168.112.209:0</a>      |

### Number 9 Top Talker

|                |     |
|----------------|-----|
| NMAP TCP ping! | 869 |
|----------------|-----|

Nmap will use a packet with the ack flag set for certain types of scans such as one to perform OS discovery. Why though this probable Nmap scan is coming from a source port of 80 is unknown to this analyst. Also as noted below in the snortsnarf alert snippet is the fact that it is src port 80 to dst port 53 followed by src port 53 to dst port 53. After extensive searches online no correlation besides from other GCIA analysts was found on this subject. The correlation from these same GCIA analysts themselves could not find a reason as to why this odd src port to dst port activity was being seen. This is evidenced by [Glenn Larratt's](#) GCIA paper. While it is not definite that this is Nmap in action it is quite possible that it is. Furthermore this may be someone using this tool who does not really know how to use it or the precepts of TCP/IP itself.

Defensive recommendations would be to possibly change the banner on all services being offered. This will help slow down an attacker as they may very well use exploit code which is not useful due to the fact that the banner has given them the wrong BIND version for example. As well if the company can afford it the purchase of an application layer firewall would be most beneficial to stop application layer attacks.

|   |
|---|
| 04/20-12:54:09.427494 [**] <a href="#">NMAP TCP ping!</a> [**] <a href="#">63.211.17.228:80</a> -> <a href="#">192.168.1.3:53</a> |
|---|

04/20-12:54:09.427505 [\*\*] [NMAP TCP ping!](#) [\*\*] [63.211.17.228:53](#) -> [192.168.1.3:53](#)

04/20-13:50:37.152995 [\*\*] [NMAP TCP ping!](#) [\*\*] [63.211.17.228:80](#) -> [192.168.1.4:53](#)

04/20-13:50:37.153006 [\*\*] [NMAP TCP ping!](#) [\*\*] [63.211.17.228:53](#) -> [192.168.1.4:53](#)

## Number 10 Top Talker

|                                 |     |
|---------------------------------|-----|
| Possible Trojan server activity | 419 |
|---------------------------------|-----|

This alert can be seen as being triggered by the destination or source port of 27374. The reason it is triggering in this case is as noted below we have an internal web server talking to an external IP on that machines 27374 port. Subseven Trojan is known to be on that port by default if a machine is infected. Subseven is a well known Trojan which will allow one full control over a victims machine.

Defensive recommendations would be to ensure that all computers have an up to date virus program and that port 27374 be blocked at the router as well. Also periodic sweeps of the network using a bpf filter to look for traffic on this port would be beneficial as well.

04/20-16:51:18.422019 [\*\*] [Possible trojan server activity](#) [\*\*] [192.168.24.34:80](#) -> [146.145.55.124:27374](#)

04/20-17:22:32.761992 [\*\*] [Possible trojan server activity](#) [\*\*] [192.168.24.34:80](#) -> [202.163.198.206:27374](#)

04/21-00:42:58.967081 [\*\*] [Possible trojan server activity](#) [\*\*] [192.168.24.34:80](#) -> [24.35.13.157:27374](#)

04/21-00:42:58.967209 [\*\*] [Possible trojan server activity](#) [\*\*] [192.168.24.34:80](#) -> [24.35.13.157:27374](#)

## Top 10 scan

The top 10 scan file are analyzed by their frequency and by the most active IP's that have created these alerts.

| Priority | Signature (click for sig info)  | # Alerts | # Sources | # Dests | Detail link             |
|----------|---------------------------------|----------|-----------|---------|-------------------------|
| N/A      | spp_portscan: UDP scan          | 38034    | 8         | 8046    | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP *****S* scan  | 16037    | 28        | 7067    | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP 12****S* scan | 64       | 18        | 5       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP ***A*R*F scan | 9        | 6         | 2       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP *****F scan   | 5        | 5         | 3       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP ***** scan    | 2        | 1         | 1       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP **U*P*S* scan | 2        | 2         | 2       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP *2*A**S* scan | 2        | 1         | 1       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP *2***R** scan | 1        | 1         | 1       | <a href="#">Summary</a> |
| N/A      | spp_portscan: TCP 1****R** scan | 1        | 1         | 1       | <a href="#">Summary</a> |

## Scan 1

|                        |       |
|------------------------|-------|
| spp_portscan: UDP scan | 38034 |
|------------------------|-------|

It can be determined that there was 38034 UDP scans which equates to about 70% of all scans. The majority of these are originating from the universities net. The below noted information shows what appears to be normal DNS activity ie: the university DNS server is querying another upstream DNS server to resolve a query. The one thing of note here is that the source port of 192.168.1.3 and 192.168.1.4 remain the same when doing these queries to various external DNS servers. I am unsure if this is normal BIND activity. Extensive googling was done to find out if BIND works this way with nil results. Bearing this in mind I would suggest that a binary log sample be obtained to verify if this is indeed normal BIND behaviour.

```
Apr 25 00:10:41 192.168.1.3:62479 -> 202.104.32.253:53 UDP
```

```
Apr 25 00:10:41 192.168.1.3:62479 -> 209.173.53.162:53 UDP
```



Apr 25 00:10:41 [192.168.1.3:62479](#) -> [66.230.133.40:53](#) UDP

Apr 25 00:10:41 [192.168.1.3:62479](#) -> [65.125.228.67:53](#) UDP

|                             |       |       |
|-----------------------------|-------|-------|
| <a href="#">192.168.1.4</a> | 10127 | 10136 |
|-----------------------------|-------|-------|

Apr 25 00:10:41 [192.168.1.4:32788](#) -> [64.158.219.3:53](#) UDP

Apr 25 00:10:41 [192.168.1.4:32788](#) -> [192.48.79.30:53](#) UDP

Apr 25 00:10:41 [192.168.1.4:32788](#) -> [216.52.17.51:53](#) UDP

Apr 25 00:10:42 [192.168.1.4:32788](#) -> [216.66.69.69:53](#) UDP

## Scan 2

|                                |       |
|--------------------------------|-------|
| spp_portscan: TCP *****S* scan | 16037 |
|--------------------------------|-------|

It can be determine that 16037 alerts are a result of Syn scans, which represent about 29% of all the scan alerts.

The reason syn scans are seen a great deal is due to either one of two things primarily. The first is legitimate activity ie: a new connection is being set up and is following the normal TCP/IP handshake of syn, syn/ack, followed by ack. The other is that someone may be sending syn packets to a specific machine on a specific port to see if there is a service listening. If there were a service listening then the machine would respond with a syn/ack. There is not much that can be done about these types of scans beyond simply monitoring them for possible trends as a prelude to an attack.

Apr 25 00:10:40 [192.168.97.77:2052](#) -> [92.174.124.68:80](#) SYN \*\*\*\*\*S\*

Apr 25 00:10:40 [192.168.97.77:2053](#) -> [113.211.81.123:80](#) SYN \*\*\*\*\*S\*

Apr 25 00:10:40 [192.168.97.77:2008](#) -> [84.252.105.195:80](#) SYN \*\*\*\*\*S\*

Apr 25 00:10:40 [192.168.97.77:2054](#) -> [159.160.22.153:80](#) SYN \*\*\*\*\*S\*

## Scan 3

|     |                                  |    |    |   |                         |
|-----|----------------------------------|----|----|---|-------------------------|
| N/A | spp_portscan: TCP 12*****S* scan | 64 | 18 | 5 | <a href="#">Summary</a> |
|-----|----------------------------------|----|----|---|-------------------------|

This type of scan, which has the reserved bits set (reserved bits now assigned to ECN and CWR) and the syn flag set is probably as a result of Queso. Queso is a tool which attempts to find the operating system being used by a user through the use of such packets as noted below.

|   |
|---|
| Apr 25 00:11:08 <a href="#">68.54.84.49:43815</a> -> <a href="#">192.168.6.7:110</a> SYN 12****S*<br>RESERVEDBITS |
| Apr 25 00:12:11 <a href="#">68.54.84.49:43816</a> -> <a href="#">192.168.6.7:110</a> SYN 12****S*<br>RESERVEDBITS |
| Apr 25 00:14:20 <a href="#">68.54.84.49:43818</a> -> <a href="#">192.168.6.7:110</a> SYN 12****S*<br>RESERVEDBITS |
| Apr 25 00:21:48 <a href="#">68.54.84.49:43825</a> -> <a href="#">192.168.6.7:110</a> SYN 12****S*<br>RESERVEDBITS |

#### Scan 4

|     |                                 |   |   |   |                         |
|-----|---------------------------------|---|---|---|-------------------------|
| N/A | spp_portscan: TCP ***A*R*F scan | 9 | 6 | 2 | <a href="#">Summary</a> |
|-----|---------------------------------|---|---|---|-------------------------|

This appears to be an illegal flag combination. While the fin and ack is a legal combination the reset on its own is illegal. It also works the same for the other way around a rst and ack is legal, but having a fin on its own is not. Someone may be using this flag combination to do OS enumeration. As such this scan poses no threat other

|   |
|---|
| Apr 25 00:34:06 <a href="#">200.181.139.158:2234</a> -> <a href="#">192.168.97.84:3419</a> INVALIDACK<br>***A*R*F |
| Apr 25 00:34:09 <a href="#">200.181.139.158:2234</a> -> <a href="#">192.168.97.84:3419</a> INVALIDACK<br>***A*R*F |
| Apr 25 00:34:15 <a href="#">200.181.139.158:2234</a> -> <a href="#">192.168.97.84:3419</a> INVALIDACK<br>***A*R*F |

#### Scan 5

|     |                               |   |   |   |                         |
|-----|-------------------------------|---|---|---|-------------------------|
| N/A | spp_portscan: TCP *****F scan | 5 | 5 | 3 | <a href="#">Summary</a> |
|-----|-------------------------------|---|---|---|-------------------------|

The below noted scan is using an illegal flag combination. The fin flag should only be seen with a ack accompanying it. This scan used to be effective several years ago when firewalls were still being developed.

Apr 25 00:39:11 [203.113.195.208:4029](#) -> [192.168.98.71:6346](#) FIN \*\*\*\*\*F

## Scan 6

|     |                              |   |   |   |                         |
|-----|------------------------------|---|---|---|-------------------------|
| N/A | spp_portscan: TCP ***** scan | 2 | 1 | 1 | <a href="#">Summary</a> |
|-----|------------------------------|---|---|---|-------------------------|

Null packets are where a packet with no flags being set are seen. This type of scan is used still to this day even though most every firewall will drop it out of hand. It was effective back when firewalls were a maturing technology, but is no longer very effective.

```
Apr 25 00:12:52 67.119.232.234:45578 -> 192.168.12.4:110 NULL *****
Apr 25 00:34:46 67.119.232.234:45834 -> 192.168.12.4:110 NULL *****
```

## Scan 7

|     |                                 |   |   |   |                         |
|-----|---------------------------------|---|---|---|-------------------------|
| N/A | spp_portscan: TCP **U*P*S* scan | 2 | 2 | 2 | <a href="#">Summary</a> |
|-----|---------------------------------|---|---|---|-------------------------|

This is another illegal flag combination scan. These flags should have an ack with them were it only say the syn or the psh. Having the urgent, push, and syn on their own however is clearly an illegal combination. Probably the person doing this scan is again seeing if he can traverse a firewall with this combination. Most modern implementations of firewalls would drop this packet outright.

```
Apr 25 00:34:13 217.88.218.210:3136 -> 192.168.97.84:4899 NOACK **U*P*S*
```

## Scan 8

|     |                                 |   |   |   |                         |
|-----|---------------------------------|---|---|---|-------------------------|
| N/A | spp_portscan: TCP *2*A**S* scan | 2 | 1 | 1 | <a href="#">Summary</a> |
|-----|---------------------------------|---|---|---|-------------------------|

This scan is with the syn and ack flags set as well as the ECN flag set. The ECN (Explicit Congestion Notification) bit is used for congestion notification. Newer operating systems such as Windows 2000 have this capability. If a computer is ECN aware as it were it will advertise this in its syn packet by setting the ECN bit, and if the machine being syn'd is also ECN aware it will respond with a syn/ack while also setting its ECN bit. This as such is normal activity.

```
Apr 25 00:41:30 167.127.101.50:443 -> 192.168.97.176:3797 UNKNOWN *2*A**S*
RESERVEDBITS
```

Apr 25 00:45:53 [167.127.101.50:443](#) -> [192.168.97.176:3802](#) UNKNOWN \*2\*A\*\*S\*  
RESERVEDBITS

### Scan 9

N/A spp\_portscan: TCP \*2\*\*\*R\*\* scan 1 1 1 [Summary](#)

This scan as such is another illegal flag combination. The reset flag is supposed to be accompanied by the ack flag as well. Not to mention that the ECN flag should only be seen during the syn and syn/ack portion of the handshake. It is possible that this is the result of a broken TCP/IP stack, but it is rather unlikely. Probably more likely that someone is engaging in some packet crafting.

Apr 25 00:11:36 [192.168.12.4:143](#) -> [24.35.55.152:64600](#) UNKNOWN \*2\*\*\*R\*\*  
RESERVEDBITS

Apr 25 00:48:32 [192.168.12.4:993](#) -> [151.196.116.95:51625](#) UNKNOWN 1\*\*\*\*R\*\*  
RESERVEDBITS

### Scan 10

spp\_portscan: TCP 1\*\*\*\*R\*\* scan 1 1 1 [Summary](#)

Once again this is an illegal flag combination. The reset flag should never be seen by itself. It should always be accompanied by the ack flag as well. Also odd that the CWR flag is set here. The CWR flag should only be seen during the normal exchange of data to signal congestion.

Apr 25 00:48:32 [192.168.12.4:993](#) -> [151.196.116.95:51625](#) UNKNOWN 1\*\*\*\*R\*\*  
RESERVEDBITS

### Noted below are the Top Ten Source IP's in the Scan File with description

| Rank    | Total # Alerts | Source IP          | # Signatures triggered | Destinations involved  |
|---------|----------------|--------------------|------------------------|------------------------|
| rank #1 | 26580 alerts   | <b>192.168.1.3</b> | 2 signatures           | (6430 destination IPs) |
| rank #2 | 10136 alerts   | <b>192.168.1.4</b> | 2 signatures           | (2912 destination IPs) |

|          |             |                        |              |                        |
|----------|-------------|------------------------|--------------|------------------------|
| rank #3  | 5769 alerts | <b>192.168.97.77</b>   | 2 signatures | (2792 destination IPs) |
| rank #4  | 4610 alerts | <b>61.37.174.136</b>   | 1 signatures | (2978 destination IPs) |
| rank #5  | 1929 alerts | <b>192.168.150.226</b> | 1 signatures | (254 destination IPs)  |
| rank #6  | 1716 alerts | <b>192.168.34.14</b>   | 1 signatures | (156 destination IPs)  |
| rank #7  | 1061 alerts | <b>192.168.110.72</b>  | 1 signatures | (156 destination IPs)  |
| rank #8  | 574 alerts  | <b>192.168.97.84</b>   | 1 signatures | (398 destination IPs)  |
| rank #9  | 385 alerts  | <b>192.168.97.128</b>  | 1 signatures | (124 destination IPs)  |
| rank #10 | 339 alerts  | <b>192.168.25.66</b>   | 1 signatures | (92 destination IPs)   |

### Scan 1

|                        |       |
|------------------------|-------|
| spp_portscan: UDP scan | 38034 |
|------------------------|-------|

It can be determined that there was 38034 UDP scans which equates to about 70% of all scans. The majority of these are originating from the universities net. What is of note in the below noted snippet is the fact that the source port remains the same ie: 62479. In all likelihood this is a DNS server bearing in mind that it is going to port 53 outbound and that the IP is a low level in the numbering convention ie: 192.168. **1.3** As mentioned below also I am unsure if the anchored source port is a normal behaviour of BIND. Normal behaviour is a new ephemeral port for every packet sent out. Keeping this anomaly in mind it would be prudent to further investigate this IP address. This particular alert alone is responsible for 26580 alarms. It is unlikely though that this is a scan but rather is normal DNS activity. In all likelihood this is a false positive.

|                 |                                   |    |                                   |     |
|-----------------|-----------------------------------|----|-----------------------------------|-----|
| Apr 25 00:10:41 | <a href="#">192.168.1.3:62479</a> | -> | <a href="#">202.104.32.253:53</a> | UDP |
| Apr 25 00:10:41 | <a href="#">192.168.1.3:62479</a> | -> | <a href="#">209.173.53.162:53</a> | UDP |
| Apr 25 00:10:41 | <a href="#">192.168.1.3:62479</a> | -> | <a href="#">66.230.133.40:53</a>  | UDP |
| Apr 25 00:10:41 | <a href="#">192.168.1.3:62479</a> | -> | <a href="#">65.125.228.67:53</a>  | UDP |

We see once again with the below noted snortsarf snippet that 192.168.1.4 has a source port anchored at 32788. In all likelihood this is a DNS server. Also lending weight to this theory is the low IP address of it 192.168. **1.4** What I was unable to find out though is if this is normal BIND behaviour if this is indeed a DNS server. Also were this a DNS server it would certainly account for all the outbound requests to port 53. This particular

alert is responsible for 10136 events. Once again though this is probably a false positive and is simply normal DNS behaviour of the internal DNS servers going out to resolve requests.

|                             |       |       |
|-----------------------------|-------|-------|
| <a href="#">192.168.1.4</a> | 10127 | 10136 |
|-----------------------------|-------|-------|

|                 |                                   |    |                                 |     |
|-----------------|-----------------------------------|----|---------------------------------|-----|
| Apr 25 00:10:41 | <a href="#">192.168.1.4:32788</a> | -> | <a href="#">64.158.219.3:53</a> | UDP |
| Apr 25 00:10:41 | <a href="#">192.168.1.4:32788</a> | -> | <a href="#">192.48.79.30:53</a> | UDP |
| Apr 25 00:10:41 | <a href="#">192.168.1.4:32788</a> | -> | <a href="#">216.52.17.51:53</a> | UDP |
| Apr 25 00:10:42 | <a href="#">192.168.1.4:32788</a> | -> | <a href="#">216.66.69.69:53</a> | UDP |

## Scan 2

|                                |       |
|--------------------------------|-------|
| spp_portscan: TCP *****S* scan | 16037 |
|--------------------------------|-------|

It can be determine that 16037 alerts are as a result of Syn scan which represent about 29% of all the scan alerts.

The basic idea behind a syn scan is to first identify a potential target host. By doing this it can then be determined who is alive, and what type of service that is being targeted. Such as http, netbios, mail server etc... services can easily be determine by sending a syn packet ,which of course is the first of the three way handshake, after which the target host will reply with either rst if the service is not available or syn-ack if the services are offered.

This way an attacker can direct it's attack according to the services that are being offered by the scanned host.

## Scan 3

Looking at the below noted snortsnarf output 192.168.97.77 appears to be some kind of web proxy server. My reasoning for this is that there are 4 varied destination addresses with a port of 80 all coming from the same one source IP during the same second. This lends credence to my thought that this is a web proxy server. Once again this would not be a scan but rather normal activity. I cannot say for certain whether or not that this is say a Squid server for I have never observed the packets outputted by one, but it seems rather likely that it is.

|                 |                                    |    |                                   |             |
|-----------------|------------------------------------|----|-----------------------------------|-------------|
| Apr 25 00:10:40 | <a href="#">192.168.97.77:2052</a> | -> | <a href="#">92.174.124.68:80</a>  | SYN *****S* |
| Apr 25 00:10:40 | <a href="#">192.168.97.77:2053</a> | -> | <a href="#">113.211.81.123:80</a> | SYN *****S* |
| Apr 25 00:10:40 | <a href="#">192.168.97.77:2008</a> | -> | <a href="#">84.252.105.195:80</a> | SYN *****S* |

```
Apr 25 00:10:40 192.168.97.77:2054 -> 159.160.22.153:80 SYN *****S*
```

#### **Scan 4**

This is possibly associated with the software called shockwave2 which is used to view files that are designed with macromedia director it is a plug in used for web browser, there has been 4610 alarms as a result. No other correlation was noted for port [1257](#) after extensive googling.

```
Apr 25 00:10:50 61.37.174.136:4394 -> 192.168.130.2:1257 SYN *****S*
```

```
Apr 25 00:10:50 61.37.174.136:4395 -> 192.168.130.3:1257 SYN *****S*
```

```
Apr 25 00:10:50 61.37.174.136:4396 -> 192.168.130.4:1257 SYN *****S*
```

```
Apr 25 00:10:50 61.37.174.136:4399 -> 192.168.130.7:1257 SYN *****S*
```

#### **Scan 5**

From the appearance of the below noted snortsнарf snippet 192.168.150.226 is using a scanner like nmap to scan machines for known trouble ports like the ones noted below. Curious as well is the fact that this person has included port 3410 which is known as the default port for [Optix Pro](#), which is a rather nasty piece of malware that has the capability to kill all firewall and antivirus protection on the pc it has been installed on. This has caused 1929 alarms over the last five days. Looking over all of the scans this person did it seems as if they are learning a tool such as nmap or other type scanner which has the same type of functionality. This IP address does not exhibit worm type behaviour as it were ie: it does not actively scan for long periods or always use the same ports to scan for.

```
Apr 25 00:23:17 192.168.150.226:4565 -> 136.165.137.25:135 SYN *****S*
```

```
Apr 25 00:23:17 192.168.150.226:4547 -> 136.165.137.25:1025 SYN  
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:2754 -> 136.165.137.25:445 SYN *****S*
```

```
Apr 25 00:23:17 192.168.150.226:4693 -> 136.165.137.25:6129 SYN  
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:4975 -> 136.165.137.25:139 SYN *****S*
```

```
Apr 25 00:23:17 192.168.150.226:3099 -> 136.165.137.25:3410 SYN  
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:2985 -> 136.165.137.25:1433 SYN  
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:3553 -> 136.165.136.16:135 SYN *****S*
```

```
Apr 25 00:23:17 192.168.150.226:3288 -> 136.165.136.16:1025 SYN
```

```
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:4451 -> 136.165.136.16:3127 SYN
```

```
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:3542 -> 136.165.136.16:6129 SYN
```

```
*****S*
```

```
Apr 25 00:23:17 192.168.150.226:1727 -> 136.165.136.16:80 SYN *****S*
```

## Scan 6

Scanning for mail servers on outside networks could be legitimate and could be associated with sending smtp messages to outside clients. This may well be legitimate traffic, although it would surely be prudent to have a more detailed look at the below source IP as this could possibly be related to a Trojan, worm or virus.

```
Apr 25 00:12:06 192.168.34.14:33639 -> 65.255.32.250:25 SYN *****S*
```

```
Apr 25 00:12:06 192.168.34.14:33656 -> 216.188.76.37:25 SYN *****S*
```

```
Apr 25 00:12:06 192.168.34.14:33652 -> 69.59.167.204:25 SYN *****S*
```

```
Apr 25 00:12:06 192.168.34.14:33653 -> 24.238.181.32:25 SYN *****S*
```

## Scan 7

Port 2234 is associated with P2P file sharing called [soul seeker](#) or direct play which is a tool designed for multi player games to be played over the internet. This type of activity can consume a lot of resources; by this I mean bandwidth and if at all possible this type of activity should be limited if resources are seen as being somewhat “over taxed” although this is dictated by the policies that are set forth by the universities.

```
Apr 25 00:14:41 192.168.97.84:4324 -> 12.144.2.150:2234 SYN *****S*
```

```
Apr 25 00:14:41 192.168.97.84:4327 -> 209.197.56.62:2234 SYN *****S*
```

```
Apr 25 00:14:41 192.168.97.84:4330 -> 68.161.89.157:2234 SYN *****S*
```

```
Apr 25 00:14:41 192.168.97.84:4333 -> 141.209.211.62:2234 SYN *****S*
```

## Scan 8

As seen below in the snortsnarf output 172.143.178.233 is scanning the network we are auditing for port 135 followed by syn packets again to port 4444. Bearing both ports in mind we can safely say this person is trying to see if port 135 is open so they can execute the [RPC DCOM](#) exploit followed by them checking to see if certain IP addresses have a bind shell listening for connections on the default bind() port of tcp 4444.



There is really no reason to have port 135 open on the border router. This should be closed at all costs unless an effective business case can be built to have it open. The risks associated with this service are well known, and can be severe.

|  |
|--|
| Apr 25 00:10:41 <a href="#">172.143.178.233:1728</a> -> <a href="#">192.168.190.104:135</a> SYN<br>*****S* |
| Apr 25 00:10:41 <a href="#">172.143.178.233:1729</a> -> <a href="#">192.168.190.105:135</a> SYN<br>*****S* |
| Apr 25 00:10:41 <a href="#">172.143.178.233:1730</a> -> <a href="#">192.168.190.106:135</a> SYN<br>*****S* |
| Apr 25 00:10:42 <a href="#">172.143.178.233:1732</a> -> <a href="#">192.168.190.93:4444</a> SYN<br>*****S* |
| Apr 25 00:10:43 <a href="#">172.143.178.233:1733</a> -> <a href="#">192.168.190.95:4444</a> SYN<br>*****S* |
| Apr 25 00:10:45 <a href="#">172.143.178.233:1733</a> -> <a href="#">192.168.190.95:4444</a> SYN<br>*****S* |

### Scan 9

This type of scan is the result of universities net looking for possible file sharing P2P network. In this case the specific filesharing program that is sought is [eDonkey](#) Which enables users to exchange all type of media. It is however as mentioned above can be resources intensive and should be actioned according to the universities policies. Not to mention the fact that file sharing probably is being done with copyrighted material which could result in legal action against the univeristy.

|   |
|---|
| Apr 25 00:45:04 <a href="#">192.168.97.118:2383</a> -> <a href="#">221.147.131.122:4662</a> SYN *****S* |
| Apr 25 00:45:04 <a href="#">192.168.97.118:2387</a> -> <a href="#">164.77.91.111:4662</a> SYN *****S*   |
| Apr 25 00:45:04 <a href="#">192.168.97.118:2385</a> -> <a href="#">211.222.246.158:4662</a> SYN *****S* |
| Apr 25 00:45:04 <a href="#">192.168.97.118:2389</a> -> <a href="#">212.180.127.155:4662</a> SYN *****S  |

### Scan 10

This scanning activity is possibly the result of estamp or evidentiary timestamp as the source port and seems to be directed at random ephemeral ports. It is rather likely though that this is p2p activity.

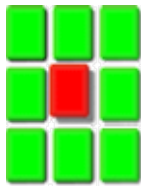
|   |
|---|
| Apr 28 13:30:01 <a href="#">192.168.153.33:1982</a> -> <a href="#">67.167.41.152:1772</a> UDP   |
| Apr 28 13:30:01 <a href="#">192.168.153.33:1982</a> -> <a href="#">195.240.177.242:3629</a> UDP |
| Apr 28 13:30:01 <a href="#">192.168.153.33:1982</a> -> <a href="#">24.217.85.179:1532</a> UDP   |

Apr 28 13:30:02 192.168.153.33:1982 -> 213.112.23.192:3476 UDP

### Five IP's of interest to be further investigated

IP 213.180.193.68 was chosen as a result of being a very noisy and very large part of the scan file. As a result of pruning I was able to reduce the size of the file. File size was the result of a massive scan from port 61872 to another ephemeral port.

As it is indicated in Dshield we can clearly see that this IP is a repeat offender and is the culprit of over 241000 complaints. This is the reason why this was chosen as the first of my external sources and registration address. University staff should have this said IP blocked at their perimeter device as there is strong evidence that this is not part of a normal users intentions.



**DShield.org**  
Distributed Intrusion  
Detection System


#### IP Info

Check another IP Address:

**IP Address:** 213.180.193.68

**HostName:** proxychecker.yandex.net

**DShield  
Profile:**

|                           |   |
|---------------------------|---|
| Country:                  |  RU |
| Contact E-mail:           | kostik@comptek.ru   |
| AS Number:                | 0   |
| Total Records against IP: | 241769  |
| Number of targets:        | 64  |
| Date Range:               | 2004-06-08 to 2004-06-23  |

[request contact update](#)

Summary was recently updated.

**Last**

**Fightback** sent to kostik@comptek.ru on 2003-12-29 13:27:36

**Sent:**

**Whois:** % This is the RIPE Whois server.  
% The objects are in RPSL format.  
%  
% Rights restricted by copyright.  
% See <http://www.ripe.net/ripenncc/pub-services/db/copyright.html>

inetnum: 213.180.192.0 - 213.180.193.255  
netname: COMPTEK-NET1  
descr: CompTek International  
descr: 3, Gubkina str., Moscow, 117809  
country: RU  
admin-c: YNDX1-RIPE  
tech-c: YNDX1-RIPE  
status: ASSIGNED PA  
notify: noc@yandex.net  
mnt-by: COMPTEK-MNT-RIPE  
changed: wawa@comptek.ru 20020607  
source: RIPE

route: 213.180.192.0/20  
descr: CompTek network / special  
origin: AS13238  
notify: noc@comptek.ru  
mnt-by: COMPTEK-MNT-RIPE  
changed: wawa@comptek.ru 20010123  
source: RIPE

role: Yandex LLC Network Operations  
address: Yandex LLC  
address: 40A Vavilova st.  
address: 117333, Moscow, Russia  
phone: +7 095 9743555  
fax-no: +7 095 9743565  
e-mail: noc@yandex.net  
trouble: -----

trouble: Points of contact for Yandex  
LLC Network Operations  
trouble: -----

trouble: Routing and peering issues:  
noc@yandex.net  
trouble: SPAM issues:  
abuse@yandex.ru  
trouble: Network security issues:  
abuse@yandex.ru  
trouble: Mail issues:  
postmaster@yandex.ru  
trouble: General information:  
info@yandex.ru  
trouble: -----

admin-c: VL11-RIPE  
tech-c: KBG2-RIPE  
notify: noc@yandex.net

```


nic-hdl:      YNDX1-RIPE
mnt-by:      COMPTEK-MNT-RIPE
changed:     wawa@comptek.ru 20020607
source:     RIPE

```

#2 This address was chosen because we have identified earlier that it was seen as scanning port 110 POP3 by sending abnormal packets. Due to this it bears closer scrutiny as it may be indicative of possible exploit code being sent next.

**IP Address:** 68.54.84.49

**HostName:** pcp01741335pcs.howard01.md.comcast.net

|                         |                           |  |
|-------------------------|---------------------------|--|
| <b>DShield Profile:</b> | Country:                  |  US |
|                         | Contact E-mail:           | abuse@comcastpc.com  |
|                         | AS Number:                | 0  |
|                         | Total Records against IP: | not processed  |
|                         | Number of targets:        | select update below  |
|                         | Date Range:               | to   |

[request contact update](#)  
[Update Summary](#)

**Whois:**

```

CustName: Comcast Cable Communications, Inc.
Address: 3 Executive Campus
Address: 5th Floor
City: Cherry Hill
StateProv: NJ
PostalCode: 08002
Country: US
RegDate: 2003-03-19
Updated: 2003-03-19

NetRange: 68.54.80.0 - 68.54.95.255
CIDR: 68.54.80.0/20
NetName: BALTIMORE-A-4
NetHandle: NET-68-54-80-0-1
Parent: NET-68-32-0-0-1
NetType: Reassigned
Comment: NONE
RegDate: 2003-03-19
Updated: 2003-03-19

TechHandle: IC161-ARIN
TechName: Comcast Cable Communications Inc
TechPhone: +1-856-317-7200
TechEmail: cips_ip-registration@cable.comcast.com

OrgAbuseHandle: NAPO-ARIN

```

OrgAbuseName: Network Abuse and Policy Observance  
OrgAbusePhone: +1-856-317-7272  
OrgAbuseEmail: abuse@comcast.net

OrgTechHandle: IC161-ARIN  
OrgTechName: Comcast Cable Communications Inc  
OrgTechPhone: +1-856-317-7200  
OrgTechEmail: cips\_ip-registration@cable.comcast.com

# ARIN WHOIS database, last updated 2004-06-22 19:10  
# Enter ? for additional hints on searching ARIN's WHOIS  
database.

OrgName: Comcast Cable Communications, Inc.  
OrgID: CMCS  
Address: 1800 Bishops Gate Blvd  
City: Mt Laurel  
StateProv: NJ  
PostalCode: 08054  
Country: US

NetRange: 68.32.0.0 - 68.63.255.255  
CIDR: 68.32.0.0/11  
NetName: JUMPSTART-1  
NetHandle: NET-68-32-0-0-1  
Parent: NET-68-0-0-0-0  
NetType: Direct Allocation  
NameServer: DNS01.JDC01.PA.COMCAST.NET  
NameServer: DNS02.JDC01.PA.COMCAST.NET  
Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE  
RegDate: 2001-11-29  
Updated: 2003-11-05

TechHandle: IC161-ARIN  
TechName: Comcast Cable Communications Inc  
TechPhone: +1-856-317-7200  
TechEmail: cips\_ip-registration@cable.comcast.com

OrgAbuseHandle: NAPO-ARIN  
OrgAbuseName: Network Abuse and Policy Observance  
OrgAbusePhone: +1-856-317-7272  
OrgAbuseEmail: abuse@comcast.net

OrgTechHandle: IC161-ARIN  
OrgTechName: Comcast Cable Communications Inc  
OrgTechPhone: +1-856-317-7200  
OrgTechEmail: cips\_ip-registration@cable.comcast.com

# ARIN WHOIS database, last updated 2004-06-22 19:10  
# Enter ? for additional hints on searching ARIN's WHOIS  
database.

OrgName: Comcast Cable Communications, Inc.  
OrgID: CMCS

Address: 1800 Bishops Gate Blvd  
 City: Mt Laurel  
 StateProv: NJ  
 PostalCode: 08054  
 Country: US  
 Comment:  
 RegDate: 2001-11-29  
 Updated: 2004-02-05

AbuseHandle: NAPO-ARIN  
 AbuseName: Network Abuse and Policy Observance  
 AbusePhone: +1-856-317-7272  
 AbuseEmail: abuse@comcast.net

AdminHandle: IC161-ARIN  
 AdminName: Comcast Cable Communications Inc  
 AdminPhone: +1-856-317-7200  
 AdminEmail: cips\_ip-registration@cable.comcast.com

TechHandle: IC161-ARIN  
 TechName: Comcast Cable Communications Inc  
 TechPhone: +1-856-317-7200  
 TechEmail: cips\_ip-registration@cable.comcast.com


# ARIN WHOIS database, last updated 2004-06-22 19:10  
 # Enter ? for additional hints on searching ARIN's WHOIS  
 databas

#3 This external IP was chosen because it resulted in 687 alarms, it can be determined by looking at Dshield again that this is a repeat offender. Would be a good idea to look at exactly what this IP address is doing.

**IP Address:** 192.26.92.30

**HostName:** c.gtld-servers.net

**DShield Profile:**

|                           |  |
|---------------------------|--|
| Country:                  |  US |
| Contact E-mail:           | nstld@verisign-grs.com   |
| AS Number:                | 0  |
| Total Records against IP: | 17895  |
| Number of targets:        | 30   |
| Date Range:               | 2004-06-08 to 2004-06-23   |

[request contact update](#)

new data is currently being imported.

**Last Fightback Sent:** not sent

**Whois:**

OrgName: VeriSign Global Registry Services  
 OrgID: VGRS

Address: 21345 Ridgetop Circle  
City: Dulles  
StateProv: VA  
PostalCode: 20166  
Country: US

NetRange: 192.26.92.0 - 192.26.92.255  
CIDR: 192.26.92.0/24  
NetName: VGRSGTLD-3  
NetHandle: NET-192-26-92-0-1  
Parent: NET-192-0-0-0-0  
NetType: Direct Assignment  
NameServer: L2.NSTLD.COM  
NameServer: D2.NSTLD.COM  
NameServer: E2.NSTLD.COM  
NameServer: C2.NSTLD.COM  
Comment:  
RegDate: 2000-11-30  
Updated: 2001-03-20

TechHandle: ZV22-ARIN  
TechName: VeriSign Global Registry Services  
TechPhone: +1-703-318-6444  
TechEmail: nstld@verisign-grs.com

OrgTechHandle: AH678-ARIN  
OrgTechName: Herrmann, Andrew  
OrgTechPhone: +1-703-948-3333  
OrgTechEmail: aherrmann@verisign.com

# ARIN WHOIS database, last updated 2004-06-22 19:10  
# Enter ? for additional hints on searching ARIN's  
WHOIS database.

OrgName: Various Registries (Maintained by ARIN)  
OrgID: VR-ARIN  
Address: 3635 Concord Parkway, Suite 200  
City: Chantilly  
StateProv: VA  
PostalCode: 20151  
Country: US

NetRange: 192.0.0.0 - 192.255.255.255  
CIDR: 192.0.0.0/8  
NetName: NET192  
NetHandle: NET-192-0-0-0-0  
Parent:  
NetType: Early Registrations, Maintained by ARIN  
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: 1993-05-01  
Updated: 2003-10-01

# ARIN WHOIS database, last updated 2004-06-22 19:10  
# Enter ? for additional hints on searching ARIN's  
WHOIS database.

OrgName: VeriSign Global Registry Services  
OrgID: VGRS  
Address: 21345 Ridgetop Circle  
City: Dulles  
StateProv: VA  
PostalCode: 20166  
Country: US  
Comment:  
RegDate: 2000-11-30  
Updated: 2004-01-13

AdminHandle: KS804-ARIN  
AdminName: Silva, Ken  
AdminPhone: +1-703-948-3432  
AdminEmail: ksilva@verisign.com

TechHandle: AH678-ARIN  
TechName: Herrmann, Andrew  
TechPhone: +1-703-948-3333  
TechEmail: aherrmann@verisign.com

# ARIN WHOIS database, last updated 2004-06-22 19:10  
# Enter ? for additional hints on searching ARIN's  
WHOIS database.

#4 This IP was chosen as result of the nmap scan, which was quite big in size. Further to this Dshield has 3724 records against this particular external IP. It should be blocked at the external router for the mean time.


---

Check another IP Address:

---

**IP Address:** 63.211.17.228  
**HostName:** proximitycheck1.allmusic.com

**DShield Profile:**

|                 |  |
|-----------------|--|
| Country:        |  US |
| Contact E-mail: | abuse@level3.com   |



|                           |                          |
|---------------------------|--------------------------|
| AS Number:                | 0                        |
| Total Records against IP: | 3724                     |
| Number of targets:        | 114                      |
| Date Range:               | 2004-06-20 to 2004-06-23 |

[request contact update](#)

[Update Summary](#)

**Last Fightback Sent:** not sent

**Whois:**

OrgName: Level 3 Communications, Inc.  
 OrgID: LVLT  
 Address: 1025 Eldorado Blvd.  
 City: Broomfield  
 StateProv: CO  
 PostalCode: 80021  
 Country: US

NetRange: 63.208.0.0 - 63.215.255.255  
 CIDR: 63.208.0.0/13  
 NetName: LEVEL4-CIDR  
 NetHandle: NET-63-208-0-0-1  
 Parent: NET-63-0-0-0-0  
 NetType: Direct Allocation  
 NameServer: NS1.LEVEL3.NET  
 NameServer: NS2.LEVEL3.NET  
 Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE  
 RegDate: 1999-05-28  
 Updated: 2001-05-30

TechHandle: LC-ORG-ARIN  
 TechName: level Communications  
 TechPhone: +1-877-453-8353  
 TechEmail: ipaddressing@level3.com

OrgAbuseHandle: APL8-ARIN  
 OrgAbuseName: Abuse POC LVLT  
 OrgAbusePhone: +1-877-453-8353  
 OrgAbuseEmail: abuse@level3.com

OrgTechHandle: TPL1-ARIN  
 OrgTechName: Tech POC LVLT  
 OrgTechPhone: +1-877-453-8353  
 OrgTechEmail: ipaddressing@level3.com

OrgTechHandle: ARINC4-ARIN  
 OrgTechName: ARIN Contact  
 OrgTechPhone: +1-800-436-8489  
 OrgTechEmail: arin-contact@genuity.com

# ARIN WHOIS database, last updated 2004-06-22 19:10  
 # 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: 63.0.0.0 - 63.255.255.255  
CIDR: 63.0.0.0/8  
NetName: NET63  
NetHandle: NET-63-0-0-0-0  
Parent:  
NetType: Allocated to ARIN  
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: 1997-04-25  
Updated: 2003-10-01

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

OrgTechHandle: ARIN-HOSTMASTER  
OrgTechName: Registration Services Department  
OrgTechPhone: +1-703-227-0660  
OrgTechEmail: hostmaster@arin.net

# ARIN WHOIS database, last updated 2004-06-22 19:10  
# Enter ? for additional hints on searching ARIN's  
WHOIS database.

OrgName: Level 3 Communications, Inc.  
OrgID: LVL3  
Address: 1025 Eldorado Blvd.  
City: Broomfield  
StateProv: CO  
PostalCode: 80021  
Country: US  
Comment:  
RegDate: 1998-05-22  
Updated: 2003-11-06

AbuseHandle: APL8-ARIN  
AbuseName: Abuse POC LVL3

```

AbusePhone: +1-877-453-8353
AbuseEmail: abuse@level3.com

AdminHandle: APL7-ARIN
AdminName: ADMIN POC LVLT
AdminPhone: +1-877-453-8353
AdminEmail: ipaddressing@level3.com

TechHandle: TPL1-ARIN
TechName: Tech POC LVLT
TechPhone: +1-877-453-8353
TechEmail: ipaddressing@level3.com

TechHandle: ARINC4-ARIN
TechName: ARIN Contact
TechPhone: +1-800-436-8489
TechEmail: arin-contact@genuity.com

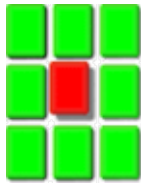
```

```

# ARIN WHOIS database, last updated 2004-06-22 19:10
# Enter ? for additional hints on searching ARIN's
WHOIS database.

```

#5 This external source address was chosen because it has created 4678 alarms directed at 13 different hosts. These are different alarm types for which purpose is to enumerate available services. They are Nmap, null scan, syn-fin scan, probable Nmap fingerprint attempt. I would recommend IT staff keep a very close eye on said IP even if there are no available records against it just yet and should block said IP at the perimeter device.



## DSHield.org

### Distributed Intrusion Detection System


#### IP Info

---

Check another IP Address:

---

**IP Address:** 209.164.32.205

**HostName:** 209.164.32.205.ptr.us.xo.net

|                         |                           |               |
|-------------------------|---------------------------|---------------|
| <b>DSHield Profile:</b> | Country:                  | US            |
|                         | Contact E-mail:           | abuse@xo.com  |
|                         | AS Number:                | 0             |
|                         | Total Records against IP: | not processed |

|                    |                     |
|--------------------|---------------------|
| Number of targets: | select update below |
| Date Range:        | to                  |

[request contact update](#)  
[Update Summary](#)

**Whois:**

OrgName: XO Communications  
 OrgID: XOXO  
 Address: Corporate Headquarters  
 Address: 11111 Sunset Hills Road  
 City: Reston  
 StateProv: VA  
 PostalCode: 20190-5339  
 Country: US

ReferralServer:  
 rwhois://rwhois.eng.xo.com:4321/

NetRange: 209.164.0.0 - 209.164.63.255  
 CIDR: 209.164.0.0/18  
 NetName: XOXO-BLK-18  
 NetHandle: NET-209-164-0-0-1  
 Parent: NET-209-0-0-0-0  
 NetType: Direct Allocation  
 NameServer: NAMESERVER.CONCENTRIC.NET  
 NameServer: NAMESERVER1.CONCENTRIC.NET  
 NameServer: NAMESERVER2.CONCENTRIC.NET  
 NameServer: NAMESERVER3.CONCENTRIC.NET  
 Comment: For best results, please send all  
 spam and worm reports only to abuse@xo.com.  
 RegDate: 1997-11-14  
 Updated: 2003-08-08

OrgAbuseHandle: XCNV-ARIN  
 OrgAbuseName: XO Communications, Network  
 Violations  
 OrgAbusePhone: +1-866-285-6208  
 OrgAbuseEmail: abuse@xo.com

OrgTechHandle: XCIA-ARIN  
 OrgTechName: XO Communications, IP  
 Administrator  
 OrgTechPhone: +1-703-547-2000  
 OrgTechEmail: ipadmin@eng.xo.com

# ARIN WHOIS database, last updated 2004-06-22  
 19:10  
 # 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: 209.0.0.0 - 209.255.255.255  
CIDR: 209.0.0.0/8  
NetName: NET209  
NetHandle: NET-209-0-0-0-0  
Parent:  
NetType: Allocated to ARIN  
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: Formerly delegated to the InterNIC  
RegDate: 1996-06-01  
Updated: 2003-10-01

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

OrgTechHandle: ARIN-HOSTMASTER  
OrgTechName: Registration Services Department  
OrgTechPhone: +1-703-227-0660  
OrgTechEmail: hostmaster@arin.net

# ARIN WHOIS database, last updated 2004-06-22  
19:10  
# Enter ? for additional hints on searching  
ARIN's WHOIS database.

OrgName: XO Communications  
OrgID: XOXO  
Address: Corporate Headquarters  
Address: 11111 Sunset Hills Road  
City: Reston  
StateProv: VA  
PostalCode: 20190-5339  
Country: US  
Comment:  
RegDate:  
Updated: 2003-12-16

ReferralServer:  
rwhois://rwhois.eng.xo.com:4321/

AbuseHandle: XCNV-ARIN  
AbuseName: XO Communications, Network  
Violations

AbusePhone: +1-866-285-6208  
AbuseEmail: abuse@xo.com

AdminHandle: XCIA-ARIN  
AdminName: XO Communications, IP  
Administrator  
AdminPhone: +1-703-547-2000  
AdminEmail: ipadmin@eng.xo.com

TechHandle: XCIA-ARIN  
TechName: XO Communications, IP Administrator  
TechPhone: +1-703-547-2000  
TechEmail: ipadmin@eng.xo.com

# ARIN WHOIS database, last updated 2004-06-22  
19:10  
# Enter ? for additional hints on searching  
ARIN's WHOIS database.

(cached)

### OOS top Ten talker by volume

The OOS top ten talker criteria was chosen as a result of the number of total alerts.

| Rank    | Total # Alerts | Source IP              | # Signatures triggered | Destinations involved          |
|---------|----------------|------------------------|------------------------|--------------------------------|
| rank #1 | 1590 alerts    | <b>68.54.84.49</b>     | 1 signatures           | 192.168.6.7                    |
| rank #2 | 169 alerts     | <b>66.225.198.20</b>   | 1 signatures           | 192.168.12.6                   |
| rank #3 | 136 alerts     | <b>204.92.130.36</b>   | 1 signatures           | 192.168.60.17,<br>192.168.12.6 |
| rank #4 | 93 alerts      | <b>68.55.57.217</b>    | 1 signatures           | (3 destination IPs)            |
| rank #5 | 79 alerts      | <b>67.119.232.234</b>  | 1 signatures           | 192.168.12.4                   |
| rank #6 | 67 alerts      | <b>193.251.135.126</b> | 1 signatures           | 192.168.34.11,<br>192.168.6.7  |

|          |           |                      |              |                                   |
|----------|-----------|----------------------|--------------|-----------------------------------|
| rank #7  | 51 alerts | <b>66.249.110.72</b> | 1 signatures | 192.168.60.17,<br>192.168.12.6    |
|          |           | <b>217.173.160.6</b> | 1 signatures | 192.168.102.55,<br>192.168.152.17 |
| rank #9  | 49 alerts | <b>66.249.110.68</b> | 1 signatures | 192.168.60.17,<br>192.168.12.6    |
| rank #10 | 44 alerts | <b>66.249.110.70</b> | 1 signatures | 192.168.60.17,<br>192.168.12.6    |

12\*\*\*\*S\*

The above noted OOS packet type was noted in the OOS files. This is the only type of packet also that was noted. This type of packet as noted above in the Top Ten Scan types is generally used by Queso, which is an OS fingerprinting tool.

### **Insights about internal machines**

IP 192.168.1.3 and 192.168.1.4 need to be examined very closely. They are very active in what seems to be legitimate DNS traffic. Although after close examination it can be determined that it does not necessarily follow the rules of normal TCP/IP stack behaviour. By this I mean that the random ephemeral port being used by DNS port 53 of universities host don't always change ie: they don't go up by one for every packet sent out. They remain constant.

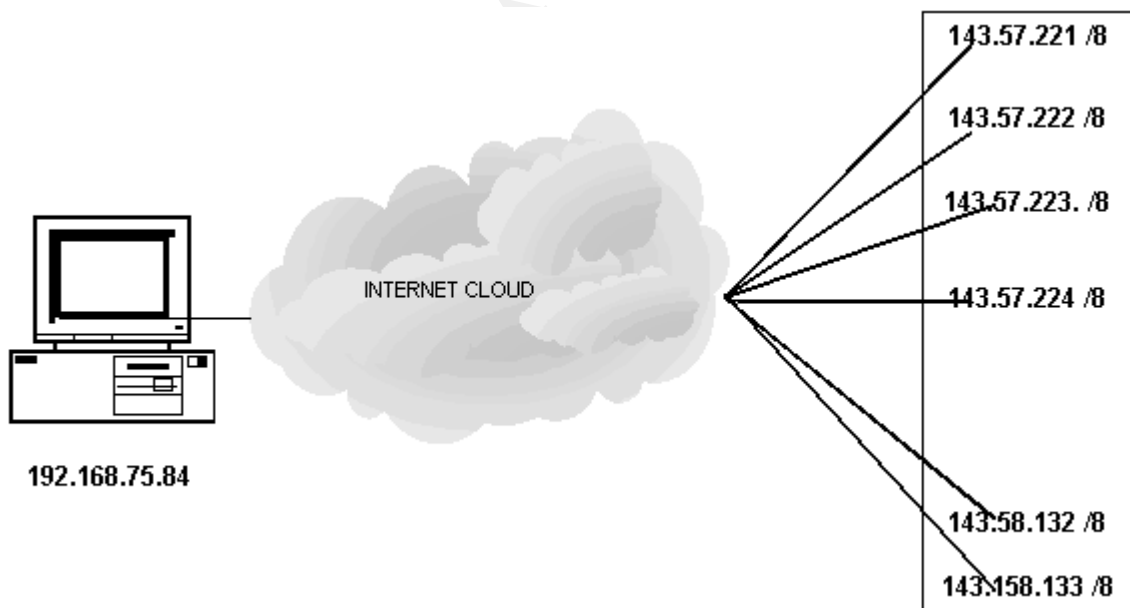
Further to this there exists a worm that behaves much like P2P and has a backdoor on DNS port 53 this worm is called [sinit](#). Although I am not advocating that this is the case and that the both machines would be infected with some type of malware, I am however strongly suggesting that these two IP be scrutinized .

IP 192.168.34.14 is seen as scanning outside networks for their mail service port 25 (smtp) this could possibly be normal traffic although I cannot confirm this. IT staff should have a close look at this machine as it could possibly be infected.

192.168.75.84 is clearly infected with the win32.agobot worm. This machine needs to be cleaned, and verified closely, but realistically will probably have to be rebuilt.

IP 192.168.150.226 is seen as scanning for some of the more well known trouble ports on its own network which has caused some 1900 alarms over the last few

### Link Graph Analysis



The above noted link graph illustrates only one host which has been infected with the win32.agobot worm. You can clearly see the enormous amount of bandwidth that this worm is using in order to propagate itself. Note that the last octet is not being completely scanned however a good portion of each one is. Lastly also note that this is a partial listing of the IP address ranges being scanned. This type of graph clearly illustrates the perils of not having an antivirus solution or worse having one that is not up to date.



Bearing this in mind it is imperative to have an effective antivirus solution in place as well as perhaps an effective SMTP security solution such as [ESafe](#). This was removed from the scan file as it was simply too big for snortsnarf to handle. Due to the size and nature of this worm it made for an excellent link graph analysis.

### **Correlation from 3 various file types**

| OOS                   | Scans        | Alerts       |
|-----------------------|--------------|--------------|
| OOS_Report_2004_04_25 | Scans.040425 | Alert.040420 |
| OOS_Report_2004_04_26 | Scans.040426 | Alert.040421 |
| OOS_Report_2004_04_27 | Scans.040427 | Alert.040422 |
| OOS_Report_2004_04_28 | Scans.040428 | Alert.040423 |
| OOS_Report_2004_04_29 | Scans.040429 | Alert.040426 |

As noted in the above date disparity doing correlation was going to be exceedingly difficult. The file selection seen above has also been approved by Jamie French. Upon close scrutiny of the above files especially the 26<sup>th</sup> for the Alert and Scans and OOS there was no correlation to be found. The top 20 for alert and scan were simply not found in each of the file types to try and build some correlation. Unfortunately due to the broken files date wise correlation is simply not available.

### **Overall Defensive Recommendations**

It is advisable that the university tighten up its routers in terms of what ports are left open. To that end it is also important that egress filtering also be used and implemented on the border routers. This will help manage and contain any malware outbreaks such as the one witnessed earlier ie: [win32.agobot worm](#).

Furthermore it is also advisable to firewall off subnets as well if at all possible as this will also help contain any malware infestations. Lastly regular internal scans should be conducted to look for known trouble ports such as port 4444 (RPC DCOM exploit) and port 3410 (Optix Pro Trojan)

University IT staff may also want to install more network sensors within the internal network. This would help detect any untoward activity both outbound and directed towards other internal machines. It is also advisable to perhaps have a software program like ESafe on the SMTP server to strip off known trouble attachments like .pif .scr and .bat

### **References**

Ricky Smith practical  
[http://www.giac.org/GCIA\\_600.php](http://www.giac.org/GCIA_600.php)

NOOP description from snort.org  
<http://www.snort.org/snort-db/sid.html?id=648>

Novell article from tech tips

<http://www.tek-tips.com/gfaqs.cfm/lev2/3/lev3/19/pid/871/fid/3352>

Erik Montcalm practical

[http://www.google.ca/search?q=cache:ubCzzrofVMwJ:www.giac.org/practical/GCIA/Erik\\_Montcalm\\_GCIA.pdf+erik+montcalm+gcia+port+51443&hl=en](http://www.google.ca/search?q=cache:ubCzzrofVMwJ:www.giac.org/practical/GCIA/Erik_Montcalm_GCIA.pdf+erik+montcalm+gcia+port+51443&hl=en)

Tom King practical

[http://www.giac.org/practical/GCIA/Tom\\_King\\_GCIA.pdf](http://www.giac.org/practical/GCIA/Tom_King_GCIA.pdf)

Novell article detailing universities use of novell products

<http://www.novell.com/news/leadstories/2001/oct16/>

Adore worm from sans

<http://www.sans.org/y2k/adore.htm>

Adore worm article from security response

<http://securityresponse.symantec.com/avcenter/venc/data/linux.adore.worm.html>

SMB Wildcard reference link

<http://www.whitehats.com/info/ids177>

Al Williams practical

[www.giac.org/practical/gcia/al\\_williams.gcia.pdf](http://www.giac.org/practical/gcia/al_williams.gcia.pdf)

SMB Ref link

<http://www.whitehats.com/info/ids177>

[http://www.google.ca/search?q=cache:hpcVyqUEwd4J:www.giac.org/practical/Joe\\_Ellis\\_GCIA.doc+SMB+Name+Wildcard+8627+&hl=en](http://www.google.ca/search?q=cache:hpcVyqUEwd4J:www.giac.org/practical/Joe_Ellis_GCIA.doc+SMB+Name+Wildcard+8627+&hl=en)

Glen Laratt practical

<http://is.rice.edu/~glratt/practical/nmapTping>

Subseven

<http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html>

Optixpro

<http://www.diamondcs.com.au/index.php?page=archive&id=analysis-optixpro>

Soul seek faq

<http://www.slsknet.org/faq.html#ports>

Cursory analysis of the RPC DCOM exploit rewritten by hdm  
<http://www.security-forums.com/forum/viewtopic.php?p=61723>

Idonk link faqs  
<http://www.idonk.com/faq.html#port>

sinit  
<http://www.lurhq.com/sinit.html>

esafe  
<http://www.whitehatinc.com/aladdin/mail/>

Agobot  
<http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=37776>

Brian Sheffler gcia paper  
<http://www.google.ca/search?hl=en&ie=UTF-8&q=brian+sheffler+gcia&meta=>

© SANS Institute 2004, Author retains full rights.

# Upcoming Training

Click Here to  
**{Get CERTIFIED!}**



|  |                        |                             |                |
|--|------------------------|-----------------------------|----------------|
| SANS San Antonio 2017                                      | San Antonio, TX        | Aug 06, 2017 - Aug 11, 2017 | Live Event     |
| SANS Boston 2017   | Boston, MA             | Aug 07, 2017 - Aug 12, 2017 | Live Event     |
| SANS Adelaide 2017   | Adelaide, Australia    | Aug 21, 2017 - Aug 26, 2017 | Live Event     |
| SANS Virginia Beach 2017                                   | Virginia Beach, VA     | Aug 21, 2017 - Sep 01, 2017 | Live Event     |
| SANS Network Security 2017                                 | Las Vegas, NV          | Sep 10, 2017 - Sep 17, 2017 | Live Event     |
| SANS vLive - SEC503: Intrusion Detection In-Depth          | SEC503 - 201709,       | Sep 11, 2017 - Oct 18, 2017 | vLive          |
| Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth | Baltimore, MD          | Sep 25, 2017 - Sep 30, 2017 | vLive          |
| SANS London September 2017                                 | London, United Kingdom | Sep 25, 2017 - Sep 30, 2017 | Live Event     |
| SANS Baltimore Fall 2017                                   | Baltimore, MD          | Sep 25, 2017 - Sep 30, 2017 | Live Event     |
| Community SANS Scottsdale SEC503                           | Scottsdale, AZ         | Oct 02, 2017 - Oct 07, 2017 | Community SANS |
| SANS October Singapore 2017                                | Singapore, Singapore   | Oct 09, 2017 - Oct 28, 2017 | Live Event     |
| Community SANS Ottawa SEC503                               | Ottawa, ON             | Oct 16, 2017 - Oct 21, 2017 | Community SANS |
| SANS Berlin 2017   | Berlin, Germany        | Oct 23, 2017 - Oct 28, 2017 | Live Event     |
| SANS Seattle 2017  | Seattle, WA            | Oct 30, 2017 - Nov 04, 2017 | Live Event     |
| San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth | San Diego, CA          | Oct 30, 2017 - Nov 04, 2017 | vLive          |
| SANS San Diego 2017  | San Diego, CA          | Oct 30, 2017 - Nov 04, 2017 | Live Event     |
| SANS Paris November 2017                                   | Paris, France          | Nov 13, 2017 - Nov 18, 2017 | Live Event     |
| Community SANS Pensacola SEC503                            | Pensacola, FL          | Nov 27, 2017 - Dec 02, 2017 | Community SANS |
| SIEM & Tactical Analytics Summit & Training                | Scottsdale, AZ         | Nov 28, 2017 - Dec 05, 2017 | Live Event     |
| SANS Cyber Defense Initiative 2017                         | Washington, DC         | Dec 12, 2017 - Dec 19, 2017 | Live Event     |
| SANS Security East 2018                                    | New Orleans, LA        | Jan 08, 2018 - Jan 13, 2018 | Live Event     |
| SANS Las Vegas 2018  | Las Vegas, NV          | Jan 28, 2018 - Feb 02, 2018 | Live Event     |
| SANS Dallas 2018   | Dallas, TX             | Feb 19, 2018 - Feb 24, 2018 | Live Event     |
| SANS OnDemand  | Online                 | Anytime                     | Self Paced     |
| SANS SelfStudy   | Books & MP3s Only      | Anytime                     | Self Paced     |