Intrusion Analysis

GIAC Certified Intrusion Analyst (GCIA) Practical Assignment

Version 3.3

Nicholas Cop

July 2003

1 Intrusion Detection: Business & Technical Design 4

Overview 4

1.1.1 Purpose of Document 4
1.1.2 Stakeholders & Benefits 4
1.1.3 Goals 5

1.2 Business Environment 5

1.2.1 Context 5
1.2.2 Assumptions 6
1.2.3 Standards Compliance 6

1.3 Design Principles 7

1.4 High-Level Architecture 7

1.4.1 Functional description 7

1.5 Component Level Architecture 9

1.5.1 Topology: IDS Design 9
1.5.2 Technical description 10
1.5.3 Architectures: System, Network & Platform 10
1.5.4 Intrusion Detection Systems (IDS) 12
1.5.5 IDS Service 12

1.6 Availability & Business Continuity 13

1.6.1 Reliability 13
1.6.2 Maintainability 13
1.6.3 Serviceability 13
1.6.4 Availability 14
1.6.5 Business Continuity Management 14
1.6.6 System Services 14
1.6.7 Network Service & KPIs 15

1.7 Capacity 15

1.7.1 Business Capacity Management 15
1.7.2 Service Capacity Management 15
1.7.3 Resource Capacity Management 16

1.8 Security & Privacy 17

1.8.1 Logical 17
1.8.2 Physical 17
1.8.3 Processes and Procedures 17
1.8.4 Risk Assessment 17
1.8.5 See Appendix A for more details. 18
1.8.6 Privacy 18
1.8.7 Grades of Intrusion Detection 18

1.9 Management 18

1.9.1 IDS Service Authorities & Management 18
1.9.2 Authentication, Authorisation and Accounting (AAA) 19
1.9.3 Network Hardware: 19
1.9.4 Host Hardware: 19

1.10 Training & Skill Sets 19

2 Network Detects 20

2.1 Detect #1 2002.4.26 20

2.1.1 Source of Trace. 20
2.1.2 Detect was generated by: 21
2.1.3 Probability the source address was spoofed: 23
2.1.4 Description of attack: 24
2.1.5 Attack mechanism: 25
2.1.6 Correlations: 25
2.1.7 Evidence of active targeting: 26
2.1.8 Severity: 26
2.1.9 Defensive recommendation: 27
2.1.10 Multiple choice test question: 27

Detect #2: 2002.5.27 33

2.1.11 Source of Trace. 33
2.1.12 Detect was generated by: 33
2.1.13 Probability the source address was spoofed: 34
2.1.14 Description of attack: 35
2.1.15 Attack mechanism: 35
2.1.16 Correlations: 37
2.1.17 Evidence of active targeting: 38
2.1.18 Severity: 38
2.1.19 Defensive recommendation: 39
2.1.20 Multiple choice test question: 39

2.2 Detect #3: 2002.8.14 40

2.2.1 Source of Trace. 40
2.2.2 Detect was generated by: 40
2.2.3 Probability the source address was spoofed: 42
2.2.4 Description of attack: 44
2.2.5 Attack mechanism: 46
2.2.6 Correlations: 46
2.2.7 Evidence of active targeting: 47
2.2.8 Severity: 48
2.2.9 Defensive recommendation: 48
2.2.10 Multiple choice test question: 49

3 Analyse This 50

3.1 Executive Summary 50

3.2 Files Analysed 50

3.3 Top Talkers 50

3.4 Top Ten Talkers (various types) 52

3.4.1 Possible Trojan Alerts 55
3.4.2 Analysis of High port 65535 udp & tcp - possible Red Worm - traffic 56
3.4.3 Possible trojan server activity 57
3.4.4 Standard rules with EXPLOIT 59
3.4.5 Possible RPC Activity 60
3.4.6 Non-descript rules 61
3.4.7 Non-Descript Rules 63
3.4.8 Internet Noise 65

3.5 Registry Information 66

3.6 Link Graph 69

3.7 Analysis Process 70

Appendix A: Risk Analysis 74

1 Intrusion Detection: Business & Technical Design

Overview

This document is a Case Study for the business and technical development of an Intrusion Detection System. It provides a template and an example of how to develop an architecture document that incorporates a business model that is sufficiently detailed to submit to management for the task of developing an framework to build a working, scalable IDS.

Please note that to view Appendix A whilst in MS Word, change to View Normal as the page format is truncated

1.1.1 Purpose of Document

This definition of Network Intrusion Detection System Architecture is a reference document for <COMPANY NAME> systems designers, developers and support staff. It provides the framework and standards for the development and deployment of <COMPANY NAME> Intrusion Detection Systems, associated logging systems and technical infrastructure.

The purpose of the document is to describe and design the architecture of the <COMPANY NAME> Intrusion Detection System. The architecture provides a medium term framework for delivering the required product characteristics.

The architecture provides the reference for:

- Component design;
- Capacity planning;
- Designing particular instances of the service; and
- Future business integration.

1.1.2 Stakeholders & Benefits

Stakeholders	Benefits
Sales Executives	Value added security component Professional Service – Security Roadmap to the future Input to Budgets Input to marketing information
Operations	Context for service delivery Framework for service improvement Synergies across products
Design Architects	Modularity and leverage Context for security management Common language for design
Executive Management	Context for strategic planning Synergies across products Facilitates commitment to longer-term investment strategies
Legal and Contractual Services	Context for supplier management
Partners	Context for integration with <COMPANY NAME>'s technology & infrastructure
Customers	Enhances confidence in <COMPANY NAME>'s security offerings and future direction
All	Allows <COMPANY NAME> to consistent product representation Allows for quicker decision-making and reduces the

	risks associated with change mitigation

1.1.3 Goals

The Intrusion Detection System architecture is designed to:

- Provide an added layer of security to <COMPANY NAME>'s many systems;
- Provide an audit trail to monitor, alert & report on attacks and attackers;
- Provide troubleshooting support for networking and systems problems;
- Assist users and law enforcement in forensic analysis of security incidents;
- Provide a security baseline for <COMPANY NAME> and its users;
- Provide security reports to users;
- Provide an outline of the business environment in which the service is delivered;
- Provide Operational staff training and understanding in IDS issues and technologies;
- Provide a decomposition of the infrastructure into modular components;
- Provide a high-level description of the processes required for capacity and availability management;
- Provide a model for maintaining Confidentiality, Integrity and Availability of any information handled by the IDS service; and
- Gain economies of scale in information systems and technology infrastructure through the standardisation of building blocks and construction methods.

1.2 Business Environment

1.2.1 Context

The <COMPANY NAME> Intrusion Detection System will be created to provide a secure shared server and network infrastructure allowing detection and analysis on multiple Network Security Domains (NSD) for <COMPANY NAME>. The IDS service provides access for multiple LANs to be included for the purpose of a logging and analysis/reporting service.

The IDS service has the ability to provide users dedicated hardware, while benefiting from shared telecommunication services. All of <COMPANY NAME>'s Business Lines will benefit from the IDS Service by the increase in security and added reporting functions. As raw packet data can be captured from various LANs and hosts, this captured data can be outputed into other types of applications to produce a wide variety of information in multiple formats.

The investment benefit in the shared infrastructure components will allow the creation of future hardware redundancy, security and service uptime.

The IDS Service has the ability to improve the security design across all <COMPANY NAME> products while providing economy of scales with the initial investment.

1.2.1.1 IDS Deliverables to User

The <COMPANY NAME> IDS Service will deliver the following to the user:

1. Host and Network Intrusion Detection Systems;
2. Real-time read access to Intrusion Detection Monitoring;
3. Notification of Events of Interest;
4. Professional Service - Security: Analysis of IDS Events of Interest;
5. Security Reports as determined by the SLA or OLA;
6. Audit Trails of IDS Events;
7. Scalable Architecture for IDS expansion; and
8. IDS compliance to AS/NZS 7799.2:2000; ASCI 33 & ISO 9001

1.2.1.2 Strategic Requirements

<COMPANY NAME>'s goal is to provide a security framework that is delivered as a value add service to existing systems. The following points are used for the strategic requirements of the IDS service:

- Seamless integration with corporate systems;
- Flexibility for business requirements such as:
  - using output logs from the IDS sensors as input to provide standard report/system formats;
  - Traffic statistics;
  - Capacity planning;
  - Budget forecasting;
- Standardised logging and security reporting system allows scalability to export data into other systems;
- Capacity for expansion with modular components; and
- Scalability to grow the system as required by <COMPANY NAME>'s business needs.

1.2.1.3 Scope

A <COMPANY NAME> developed IDS solution will utilise open source software built on an infrastructure that can be shared to incorporate other IDS architecture if required.

The architecture will cover the following areas:

- Network Intrusion Detection Software (Snort);
- Host Intrusion Detection Software (Tripwire);
- Database (MySQL);
- System, Network & Platform Architectures;
- <COMPANY NAME> Application & Logging Servers;
- Management & Authorities; and
- Host & Network IDS sensors.

1.2.2 Assumptions

The following has been taken into account in the development of the IDS architecture:

- Security requirements warrant the investment;
- Design adheres to <COMPANY NAME> security policies covering architecture standards and industry best practice;
- Connections between other <COMPANY NAME> Products/Services have appropriate security hardware/software rules applied;
- Suitability of selected products meets the IDS Service deliverables;
- Standard configuration applied to security equipment allows connectivity between various differing Network Security Domains of <COMPANY NAME> products/services without the addition of separate architecture; and
- Design adheres to ITIL, ACSI 33 & AS/NZS 7799.2:2000 principles.

1.2.3 Standards Compliance

- <COMPANY NAME> Security Policy
- AS/NZS 7799.2:2000 - Information Security Management
- ACSI 33 Handbook 13 Defence Signals Directorate
- PSM Commonwealth Protective Security Manual

1.3 Design Principles

Architecture design principles are as follows:

- Modular, layered, component-based architecture (1.6.3.2: Network Design Model);
- Maximum use of open source software;
- Maximum use of <COMPANY NAME>’s in-house skills and knowledge;
- Architecture designed as a layered security model to provide for defence in depth;
- Built to be flexible and extensible for capacity, availability and business requirements;
- Ease of Maintainability and Serviceability;
- Avoidance of complex architectures;
- Conformance to <COMPANY NAME> Corporate security policies and standards;
- Reduce equipment costs by utilising shared devices and environments where possible;
- Provide standardised input for other Corporate systems;
- Unified approach to Incident investigations and analysis; and
- Provide a training environment for Service Operations staff to increase their skill set

1.4 High-Level Architecture

1.4.1 Functional description

Intrusion Detection, both Host and Network, provide an added layer of security to any architecture. The placement of sensors and associated logging architecture determines the effectiveness and security of the design. The LAN traffic volume and the path for the sensor traffic will influence design choices and accessibility of log information.

The <COMPANY NAME> IDS service will collect information from a variety of devices and store the data in a log server. Manipulation of the data to produce reports and generate alerts will be via a separate application server.

Various technologies can be used to gather network and host data with the aim to provide a clear footprint of traffic. Using the same product for both network and host IDS has the benefit of ease of management and log normalisation but the problem of the product IDS missing a particular vulnerability. Some users and services will require a higher level of security for risk mitigation. In such circumstances an additional separate and distinct IDS Service can be incorporated into the model outlined in this document.

1.4.1.1 System Architecture Components

- <COMPANY NAME> IDS Service
- <COMPANY NAME> Infrastructure Management Service
- <COMPANY NAME> AAA Service
- <COMPANY NAME> Billing Service
- <COMPANY NAME> Reporting Service
- <COMPANY NAME> Data Backup Service

1.4.1.2 Network Architecture Components

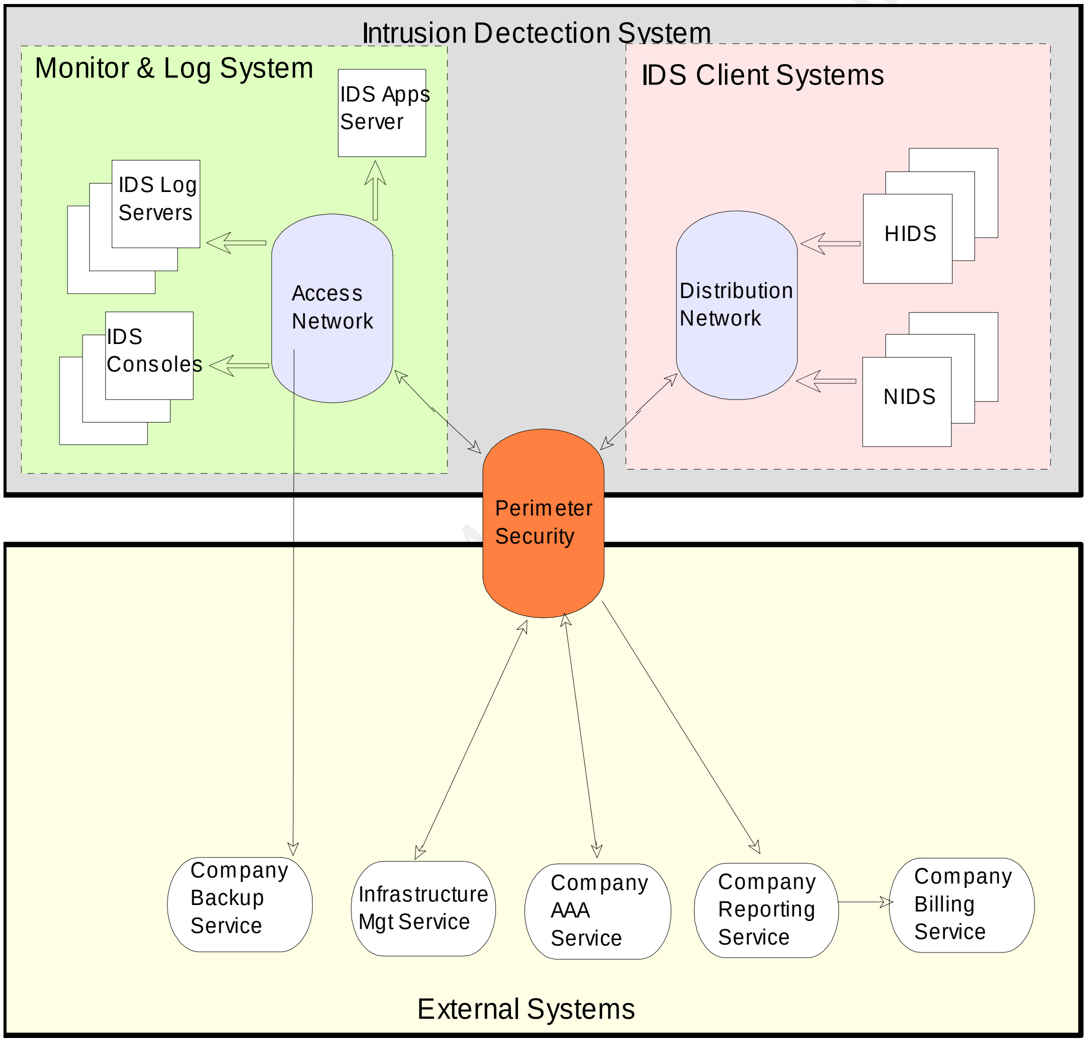
- Router on a Stick Distribution Network Model

- Perimeter Security Network Design
- Independent stand-alone IDS network

1.4.1.3 Platform Architecture Components

- Cisco Switches & Routers
- IDS Security Layer device(s)
- Log & Application Servers
- IDS devices & agents

1.4.1.4 Functional Diagram : IDS Process



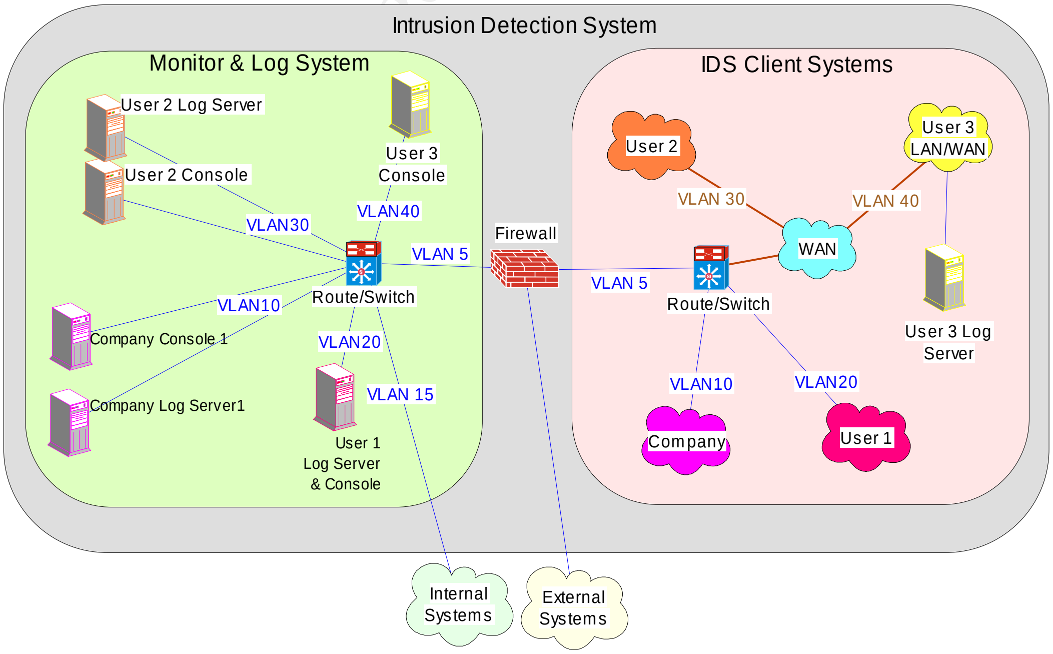
1.4.1.5 Service features

The service will have the following features:

- IDS Log Data Server for receiving network and host logging data from <COMPANY NAME> managed devices to provide audit trails and legal documents;
- IDS Console / Application Server for processing log data, producing reports, viewing IDS information, alerting, paging/escalation and managing security events;
- Intrusion Detection Security Reports will include at minimum:
    - Protocol Analysis;
    - Signature Analysis;
    - Anomaly Report;
    - Alert Report;
    - Events Of Interest (EOI); and
    - Event Report where appropriate.
- Network Sensors will be connected to various Network Security Domains (NSD) via port spanning or network taps as defined below:
    - Public – Port Spanning;
    - X-in-Confidence – Port Spanning;
    - Protected – Port Spanning;
    - Highly Protected – Port Spanning or Network Tap; and
- Very high traffic volume LANs – Network Tap; and
- Assistance to Service Operations staff in trouble-shooting end to end system problems with data log information;

1.5 Component Level Architecture

1.5.1 Topology: IDS Design



1.5.2 Technical description

The technical description of Intrusion Detection will consist of the following areas:

- System, Network & Platform Architectures;
- Intrusion Detection Systems (IDS); and
- IDS Product;

1.5.3 Architectures: System, Network & Platform

1.5.3.1 System Architecture

The IDS System Architecture is designed around an N-tier architecture model and will consist of:

1 Agents that collect data;
2 Log Databases; and
3 Application/Consoles that analyse data and produce some for of output ie reports, alerts etc.

The agents will collect the data and send any logging information directly to the log server and any alerts to the console that will produce real-time alerting. The log server will be used to provide audit trails that can be used in legal proceedings and also input into reports that are required as part of the OLA with the user.

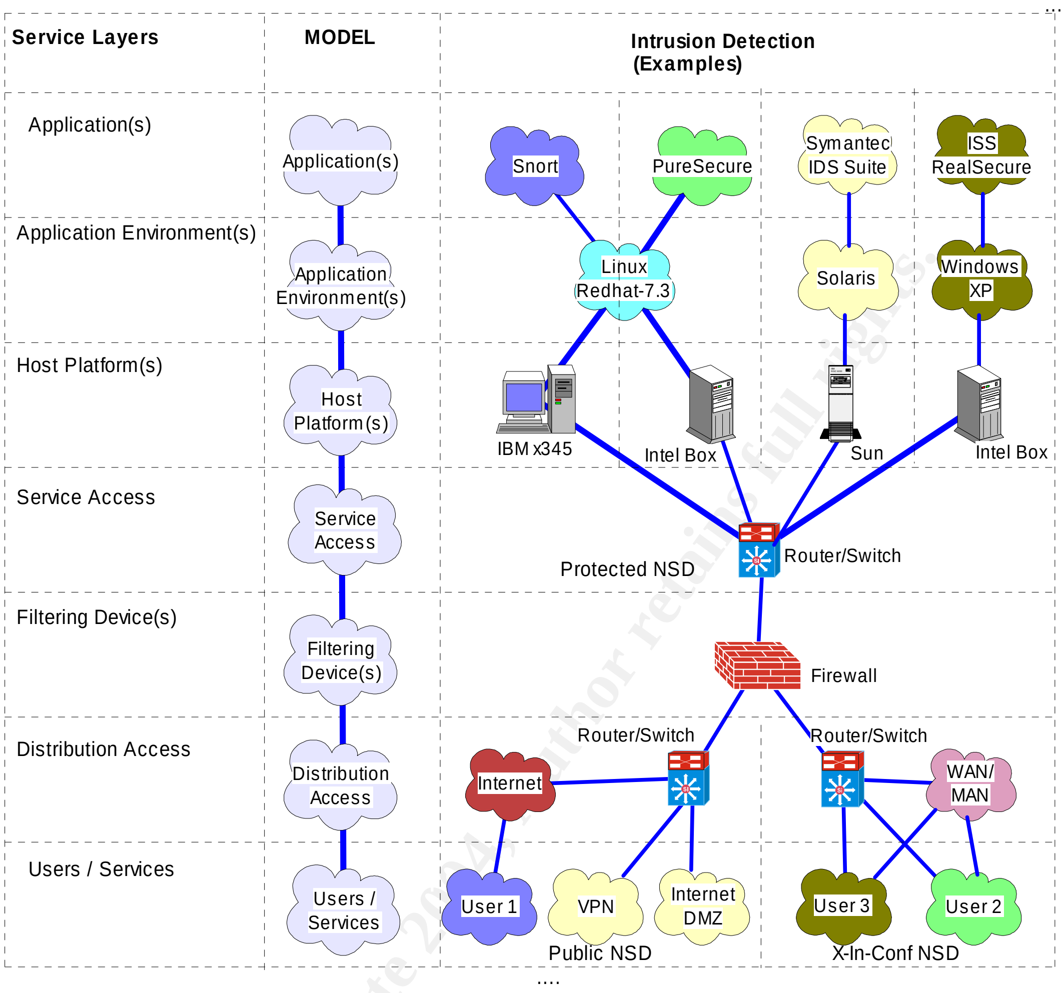
To enhance the business model of an IDS Service past the obvious security benefits, it is proposed that user is given some access to their specific information in the form of on-line security reports. In the past, this has proven successful with the statistics and benefits realised from this user interaction have further enhanced trust and user awareness of <COMPANY NAME>'s capabilities. This gives the user a feeling of being in control of their resources whilst allowing <COMPANY NAME> to manage the user service.

1.5.3.2 Network Architecture

This design will follow the principle of focusing on the service that is to be delivered and moving out to deliver the service to remote connection. In this context, any device that is sending data back to the log server is considered as an information-gathering agent. This design principle will allow any device to be used as an agent and thus provide a modular architecture.

The concept of Network Security Domains (NSD) will be used to provide segregation of users and services. Users can have their own separate NSD or utilise a shared infrastructure. The colour-coded clouds below represent different IDS services, where they are delivered and how separation occurs on a common IDS infrastructure.

Network Design Model



1.5.3.3 Platform Architecture

The specifications outlined below represent a common <COMPANY NAME> IDS system environment. Individual users may choose to utilise different platforms and smaller scale devices as an entry into the IDS service. <COMPANY NAME> will use a shared architecture for leverage and modularity. The functionality of the Application and Log Servers can be initially combined on one platform. As business requirements grow the servers can be separated.

Platform Devices

Model	**Description**
Service Distribution	
CISCO2651XM	High Performance Dual 10/100 Modular Rout with Cisco IOS IP
WS-C2950-24	24 port 10/100 Catalyst Switch Standard Image only
Service Access	
CISCO2651XM	High Performance Dual 10/100 Modular Rout with Cisco IOS IP
WS-C2950-24	24 port 10/100 Catalyst Switch Standard Image only

Log Server	IBM x345, Intel Xeon 2.4GHz/400MHz Ultra320, Rack Server
Application Server	IBM x345, Intel Xeon 2.4GHz/400MHz Ultra320, Rack Server
NIDS	IBM x345, Intel Xeon 2.4GHz/400MHz Ultra320, Rack Server
Security Perimeter	
Existing Firewall	<COMPANY NAME> Firewalls

1.5.4 Intrusion Detection Systems (IDS)

http://www.webopedia.com/TERM/I/intrusion_detection_system.html (08/04/2003)
An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

1.5.5 IDS Service

The open source IDS chosen is called Snort. This IDS software is incorporated with other open source software ie the database system MySql. Having the modularity of integrating multiple open source software allows users and developers to scale and upgrade the IDS components. The IDS is capable of being ported to a number of platforms as outlined below.

1.5.5.1 Snort

http://www.snort.org/about.html (24/02/2003)

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more.

Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilises modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows users using Samba's smbuser.

Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump(1), a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system.

Snort should work any place libpcap does, and is known to have been compiled successfully on the following platforms:

i386	**Sparc**	**M68k/PPC**	**Alpha**	**Other**	
X	X	X	X	X	Linux
X	X	X			OpenBSD
X			X		FreeBSD
X		X			NetBSD
X	X				Solaris
	X				SunOS 4.1.X

				X	HP-UX
				X	AIX
				X	IRIX
			X		Tru64
		X			MacOS X Server
X					Win32 - (Win9x/NT/2000)

Table 2 Snort Compatibility Matrix

1.5.5.2 MySQL

http://searchdatabase.techtarget.com/sDefinition/0,,sid13_gci516819,00.html (15/04/2003)

> *MySQL (pronounced "my ess cue el," not "my sequel") is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. Because it is open source, anyone can download mySQL and tailor it to their needs in accordance with the general public license. MySQL is noted mainly for its speed, reliability, and flexibility. Most agree, however, that it works best when managing content and not executing transactions.*
>
> *It is fully multi-threaded using kernel threads, provides application program interfaces (APIs) for C, C++, Eiffel, Java, Perl, PHP, Python, and Tcl, allows for many column types, and offers full operator and function support in the SELECT and WHERE parts of queries.*
>
> *MySQL currently runs on the Linux, Unix, and Windows platforms. Many Internet startups have been especially interested in mySQL as an alternative to the proprietary database systems from Oracle, IBM, and Informix. Yahoo's news site uses mySQL.*

1.6 Availability & Business Continuity

1.6.1 Reliability

It is proposed that the on-site warranty and maintenance agreements with hardware suppliers with the addition of <COMPANY NAME> spares for the network components, offers sufficient mitigation for the single point of failure scenario. Individual component resiliency is addressed in the same manner as single points of failure and is dependent on budget allocation. The proposed architecture is sufficiently scalable to provide for further reliability enhancements by implementing additional hardware.

1.6.2 Maintainability

The current <COMPANY NAME> Chief Operating Practice is acceptable to provide the IDS maintainability requirements. The network hardware/equipment will be included under the current <COMPANY NAME> maintenance contract.

1.6.3 Serviceability

Mean time to restore targets is less than or equal to 4 Hrs. The 3-year on-site warranties 24x7x4 4-hour response for the IBM xSeries345 will provide for the Log, NIDS and Application/Console Servers. An OLA is required between the IDS Operations work-unit and other <COMPANY NAME> work-units to ensure availability of the infrastructure.

1.6.4 Availability

Measure	Description	Aim
Availability	Timely, reliable access to data and information services for authorised users.	IDS reporting as agreed in OLA.
	24 hours/day, 7 days/week 8.30am to 5.30pm (Monday – Friday; allowing for interstate times; Excluding National Public Holidays)	98%
	5.30pm to 8.30am (Monday – Friday)	95%
	Weekends	95%

1.6.5 Business Continuity Management

The <COMPANY NAME> IDS Service will be part of the overall <COMPANY NAME> BCM plan. IT Service Continuity will be outlined below.

1.6.5.1 Backup & Recovery

Key systems are fully backed up on a daily basis onto tape. This includes for example on-line backups of the databases. The key information which is backed up daily include:

- Database and Directory
- Presentation log files
- Firewall log files
- Application software and log files
- Configuration files
- System Files

Backups are rotated to secure off-site storage. Recoveries are tested periodically. All processes are fully accredited to the ISO 9001 quality standard.

Data will be backed up to provide the required audit trails. Logs and configurations will be backed up locally and remotely. Data comparisons will be run between local and remote data to ensure Integrity.

To ensure availability of the service, there is a need to back up IDS Application Server components, logs & configurations remotely.

To ensure availability of the service, the router and switch configurations will be logged and backed up according to <COMPANY NAME> backup procedures.

When the disk storage of this server rises above 100 - 200 Gbytes, it should be held on <COMPANY NAME>'s SAN (storage area network). The initial storage required seems to be much lower than this threshold. However, it will be attached initially to the SAN so that it can be expanded rapidly as required.

1.6.6 System Services

The IDS Service can provide users with a real-time IDS solution that is accessed by a Web Service. Systems Services are defined by:

1.6.6.1 System Availability:

The cumulative time over the month that the system is available during prime time (08:30 till 17:30) expressed as a percentage of total prime time for the month.
The service is provided on a 24x7 basis during prime time and is delivered based on a 98% availability target calculated on a monthly basis. The monthly Service Availability Report includes details of service downtime and the calculated percentage availability for a given month.

1.6.6.2 System Downtime:

The time the system is unavailable to the User where the cause of failure lies within the <COMPANY NAME> owned server operating system (OS), hardware or software. It includes the time taken to restore the system and includes the time taken by any recovery procedures to restore the system to an acceptable operating status.

1.6.7 Network Service & KPIs

During any monthly reporting cycle the service availability will be equal to, or greater than 98%. This includes hardware, for example routers/switches and firewalls that forms part of the IDS Service Architecture.
Faults are measured when the initial fault investigation commences within:

- 30 minutes of the fault being reported/identified within Business hours; and
- 60 minutes of the fault being reported/identified outside Business hours.

Restoration targets are measured within:

- 100% restoration within 4 hours within business hours; and
- 100% restoration within 8 hours outside Business hours.

KPI	Value	Comment
Availability	98% (during prime time)	Refer to Availability and Business Continuity Section
Time to Recover/ Maximum Downtime	4 hour max prime time 8 hour max non-prime time	Prime time is 8:30am – 5:30pm (Local time) Monday - Friday

1.7 Capacity

The capacity is limited by cost as the components and the network models allow for expansion.

1.7.1 Business Capacity Management

The IDS Service will exist to support other Services. As such, it will be up to these other Services to determine if an IDS is a requirement of their system. The IDS Service will therefore only be required to be able to meet the business capacity of other systems.

1.7.2 Service Capacity Management

The network design will provide adequate capacity for initial deployment of the IDS system. The number of users and distinct IDS service requirements will therefore be dependent on the port capacity of the service distribution and service access 24 port switches. When the number of used ports exceed 70%; strategies such as port rationalisation can be used to group various NSDs onto 100MB hubs. Also, additional switching infrastructure purchase and utilising other router ports will be used.

The proposed IDS Servers have sufficient capacity (memory, CPU and ports) to cater for growth. The determining factor will be the number of networks monitored and volume of traffic that the NIDS can cope with before additional hardware is required. These statistics will be closely monitored to provide a greater understanding in traffic volumes. Business Capacity Management (see above) will be the driving force behind any Service Capacity Management issues. The proposed design is scalable to meet ad-hoc requirements to deliver immediate IDS Services using very little expansion. As stated previously, the NIDS device will be the bottleneck to NIDS Service delivery but this can be mitigated as other (simpler) NIDS devices can be installed to meet immediate business needs.

1.7.2.1 Capacity Monitoring

The following parameters are monitored to facilitate operational capacity planning and traffic engineering:

- **CPU usage –** indication of router and server capability to handle the load.
- **Memory usage –** indication of router and server capability to handle the load.
- **Packet throughput –** indication of WAN links and spanned switch ports capability to handle the offered traffic.

1.7.3 Resource Capacity Management

(from 1.6.3.3 Platform Architecture)

Router: 2651XM	40 Kpps
Switch: 2950-24	3.6-Mpps wire-speed forwarding rate 8-MB packet buffer memory architecture shared by all ports 16-MB DRAM and 8-MB Flash memory Configurable up to 8000 MAC addresses 24 port capacity
Log Server	IBM x345, Intel Xeon 2.4GHz/400MHz, 512MB, 512MB, Open Bay, Ultra320, Rack 2.4GHz/400MHz, 512KB Upgrade with Intel Xeon Processor 2x 1 GB PC2100 CL2.5 ECC DDR SDRAM RDIMM 10/100 Dual port Server Adaptor **2x*** IBM 36.4GB 10K Ultra 160 SCSI Hot-Swap SL HDD IBM 350W H/Swap Redundant Power Suply Upgrade Kit IBM ServeRAID-5I SCSI Controller
Application Server	IBM x345, Intel Xeon 2.4GHz/400MHz, 512MB, 512MB, Open Bay, Ultra320, Rack 2.4GHz/400MHz, 512KB Upgrade with Intel Xeon Processor 2x 1 GB PC2100 CL2.5 ECC DDR SDRAM RDIMM 10/100 Dual port Server Adaptor IBM 36.4GB 10K Ultra 160 SCSI Hot-Swap SL HDD IBM 350W H/Swap Redundant Power Suply Upgrade Kit IBM ServeRAID-5I SCSI Controller
NIDS Server(s)	IBM x345, Intel Xeon 2.4GHz/400MHz, 512MB, 512MB, Open Bay, Ultra320, Rack 2.4GHz/400MHz, 512KB Upgrade with Intel Xeon Processor 2x 1 GB PC2100 CL2.5 ECC DDR SDRAM RDIMM 10/100 Dual port Server Adaptor IBM 36.4GB 10K Ultra 160 SCSI Hot-Swap SL HDD **2x Quad Cards*** IBM 350W H/Swap Redundant Power Suply Upgrade Kit

```
IBM ServeRAID-5I SCSI Controller
```

***Sections in bold underline mark the differentiators in the Server devices**

1.8 Security & Privacy

Security and privacy has been addressed at the following levels:

- Logical;
- Physical;
- Procedures/Processes;
- Risk Assessment;
- Privacy; and
- Grades of Intrusion Detection.

1.8.1 Logical

The IDS adopts the following mechanisms to ensure security and confidentiality of business critical network traffic:

- Traffic VLANing and routing separation between users/services;
- Access controls with a restrictive policy of denying all except those explicitly permitted;
- Device and user authentication;
- Filters and access controls; and
- Network Security Domain segregation.

Where possible, network & server platforms will have their Operating Systems and configurations hardened by:

- Standard secure configurations and templates as per the Chief Operation Practice;
- Timely application of all necessary vendor security patches as per <COMPANY NAME> security policies;
- Disabling (and removal where possible) of all services (including utilities) not specifically needed for this the IDS Service;
- Disabling (and removal where possible) of all default accounts; and
- IDS monitors to check the configurations of the boxes and detect changes in important files in directories.

1.8.2 Physical

The <COMPANY NAME> IDS Service will be located in <COMPANY NAME>'s Computer Centre. Access to the Computer Centre will as per the Chief Operation Practice.

1.8.3 Processes and Procedures

Service and support is performed under the guidelines of <COMPANY NAME>'s ISO 9001 processes. Service modifications are made as per the <COMPANY NAME> Chief Operating Practice.

1.8.4 Risk Assessment

The difficulty in performing a Risk Assessment on an IDS Service is that the Service itself is a mitigation strategy for other Services. This fact makes it difficult to define the loss of Confidentiality, Integrity and Availability of the IDS Service without knowing the specifics of the other monitored Service. To this end, Asset

Classification is used to separate classes of information into Highly Protected, Protected, X-in-Confidence & Public. These classes of information are then used as generic circumstances of types of Services that the IDS Service will monitor. It is recommended that as Services are connected to the IDS, both the IDS Service and the monitored Service are re-evaluated in terms of a Risk Assessment so specific details can be used to define appropriate mitigation strategies.

1.8.5 See Appendix A for more details.

Appendix A: Risk Assessment

1.8.6 Privacy

In the course of its business activities, <COMPANY NAME> collects personal information from individuals for purposes including service delivery, sales and marketing, internal administration, quality management, resource management, account management and technical support.

The records are kept according to the categories set out in the Standard Retention and Disposal Schedule issued by Queensland State Archives. For more information regarding specific timeframes please refer to the Queensland State Archives website http://www.archives.qld.gov.au

1.8.7 Grades of Intrusion Detection

Defence Signals Directorate ACSI 33 Handbook 13 Intrusion Detection Grades 0 to 4 will be used as a guide for determining levels of IDS requirements. http://www.dsd.gov.au/infosec/acsi33/HB13.html (01/05/2003)

1.9 Management

Management of the Intrusion Detection Systems is performed using existing <COMPANY NAME> management tools that form part of the Chief Operating Practice. The following is addressed in terms of management:

- IDS Service Authorities & Management;
- Authentication, Authorisation and Accounting (AAA);
- Network Hardware;
- Host Hardware;
- Application;
- System.

1.9.1 IDS Service Authorities & Management

- The owner of the production IDS Service will be <OWNER>;
- The authority for changes to the IDS Service will be the <COMPANY NAME> Security Manager;
- Security Operations will manage the IDS application the IDS Log Server, firewalls, routing and switching infrastructure, Network IDS devices and Host IDS applications;
- IDS Log monitoring & processes will be defined by the <COMPANY NAME> Security Manager and implemented by Security Operations; and
- The individual components that make up the IDS Service do not deviate from any previously managed device. As such, all IDS devices will be managed by <COMPANY NAME> in the normal manner ie under the Chief Operating Practice.

1.9.2 Authentication, Authorisation and Accounting (AAA)

<COMPANY NAME> will use 2-factor token authentication where possible. The application, logging servers, switches, routers, firewalls, network and host sensors must all have a valid network path to the Authentication NSD. If 2-factor authentication is not available the Chief Operating Practice password policy will be used.

1.9.3 Network Hardware:

Monitoring and support information for the environment utilises existing <COMPANY NAME> network management tools.

1.9.4 Host Hardware:

The following outlines the require performance monitoring from an application and system perspective.

1.9.4.1 Application

- Number of events processed by IDS host devices;
- Date / time of event; and
- Peak processing times.

1.9.4.2 System

Standard monitoring on system utilisation, including:

- CPU Utilisation , captured at 5 minute averages;
- Memory Utilisation, captured at 5 minute averages;
- I/O Utilisation, captured at 5 minute averages; and

1.10 Training & Skill Sets

The following skills are required to develop and support the IDS Service from an application, system and network perspective.

Skills

Skill	Training Level	Owner	Authority	Management
Application :				
Snort	Advanced	Owner	Security Manager	Security Operations
MySQL	Intermediate	Owner	Security Manager	Security Operations
System:				
Linux	Advanced	Owner	Security Manager	Security Operations
Network:				
Firewall	Intermediate	Owner	Security Manager	Security Operations
Router	Advanced	Owner	Security Manager	Security Operations
Switches	Intermediate	Owner	Security Manager	Security Operations

2 Network Detects

2.1 Detect #1 2002.4.26

2.1.1 Source of Trace.

http://www.incidents.org/logs/Raw/2002.4.26 (18 June 2003)

Snort output

```
snort -qXdevr 2002.4.26 host 217.131.173.220
05/27-09:11:04.884488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
217.131.173.220:2839 -> 226.185.188.10:53 UDP TTL:48 TOS:0x0 ID:41677 IpLen:20
DgmLen:58
Len: 30
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 3A A2 CD 00 00 30 11 31 34 D9 83 AD DC E2 B9  .:....0.14......
0x0020: BC 0A 0B 17 00 35 00 26 09 4B 12 34 00 80 00 01  .....5.&.K.4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62  .......version.b
0x0040: 69 6E 64 00 00 10 00 03                          ind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/27-09:44:49.324488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
217.131.173.220:3048 -> 226.185.227.124:53 UDP TTL:48 TOS:0x0 ID:15966 IpLen:20
DgmLen:58
Len: 30
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 3A 3E 5E 00 00 30 11 6F 2F D9 83 AD DC E2 B9  .:>^..0.o/......
0x0020: E3 7C 0B E8 00 35 00 26 E2 05 12 34 00 80 00 01  .|...5.&...4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62  .......version.b
0x0040: 69 6E 64 00 00 10 00 03                          ind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Assumptions

Checksums and timestamps have been changed by GIAC.

2.1.1.1 Network Architecture from 2002.4.26 logs.

From the logs 2002.4.26, the MAC addresses (**00:03:E3**:D9:26:C0 -> **00:00:0C**:04:B2:33) are looked up on http://www.coffer.com/mac_find/ (18 June 2003) and we find that both the MACs are registered to Cisco.

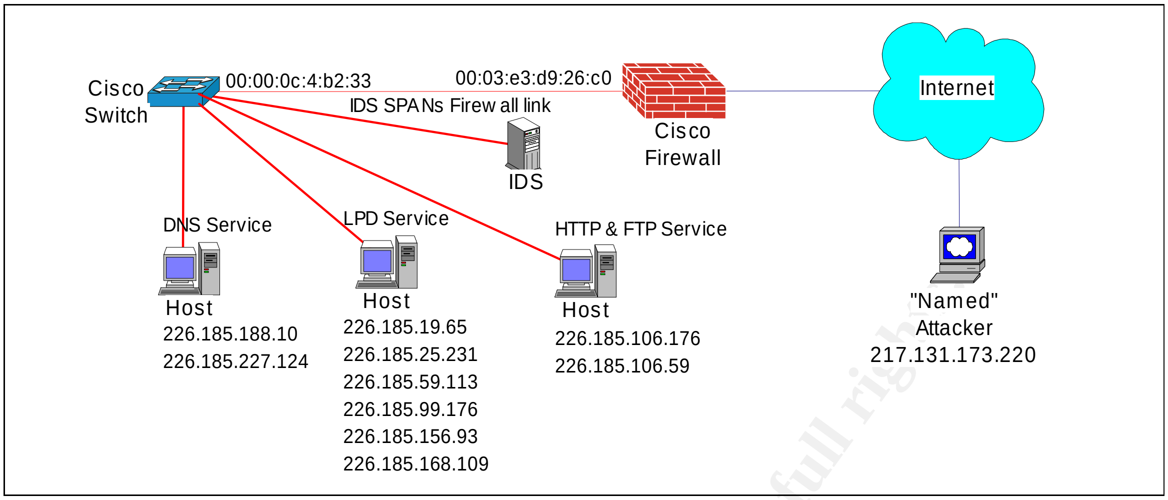
Search results for "00:00:0C"

MAC Address prefix	**Vendor**
00000C	Cisco Systems, Inc.

Search results for "00:03:E3"

MAC Address prefix	**Vendor**
0003E3	Cisco Systems, Inc

Using the above information and the data from the logs on 2002.4.26, we can hypothesise that the architecture may have looked something like Diagram 1. I have used virtual IP addresses on the hosts and limited the diagram to the log date in order to simplify the diagram.



2.1.2 Detect was generated by:

Snort intrusion detection system is used as the IDS event logger.

snort -V

```
Version 1.9.0-ODBC-MySQL-WIN32 (Build 209) By Martin Roesch (roesch@sourcefire.com,
www.snort.org)
1.7-WIN32 Port By Michael Davis (mike@datanerds.net, www.datanerds.net/~mike)
1.8-1.9 WIN32 Port By Chris Reid (chris.reid@codecraftconsultants.com)
```

The required information is extracted out of the log file **2002.4.26.**

# **snort -dvXq -c snort.conf -l log/2002.4.26 -r 2002.4.26**
\log\2002.4.26\217.131.173.220\ UDP_2839-53.ids

```
 [**] DNS named version attempt [**]
05/27-09:11:04.884488 217.131.173.220:2839 -> 226.185.188.10:53
UDP TTL:48 TOS:0x0 ID:41677 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 3A A2 CD 00 00 30 11 31 34 D9 83 AD DC E2 B9  .:....0.14......
0x0020: BC 0A 0B 17 00 35 00 26 09 4B 12 34 00 80 00 01  .....5.&.K.4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62  .......version.b
0x0040: 69 6E 64 00 00 10 00 03                          ind.....
```

Snort Glossary:

An explanation of the parameters used here and above:
-c <rules> Use Rules File <rules>
-d Dump the Application Layer
-e Display the second layer header info
-q Quiet. Don't show banner and status report
-r <tf> Read and process tcpdump file <tf>
-v Be verbose
-V Show version number
-X Dump the raw packet data starting at the link layer

Snort Packet Format

Format (UDP_2839-53.ids)	**Explanation of Fields**
[**] DNS named version attempt [**]	This is what Snort thinks that attack is
05/27-09:11:04.884488	The timestamp

217.131.173.220: 2839	Source IP address of the attack Source port number of attack
226.185.188.10: 53	Target IP address of the attack Target Port number of attack (DNS)
UDP	IP Protocol 17
TTL:48	Time To Live of packet
TOS:0x0	Type of Service = 0 implies normal service
ID:41677	IP Packet ID (Value modified by GIAC)
IpLen:20	The length of the IP portion of packet
DgmLen:58	The total length of the packet (IP+UDP)
Len: 38	The UDP length of the packet
0x0030: 00 00 00 00 00 00 **07** 76 65 72 73 69 6F 6E **04** 62	Forth part of UDP packet represented in hexadecimal (Detect version.bind |**07**| & |**04**|)
.....3....&...E.	First part of UDP packet represented in ASCII
.:>^..0.o/......	Second part of UDP packet represented in ASCII
.|...5.&...4....	Third part of UDP packet represented in ASCII
.......version.b	Forth part of UDP packet represented in ASCII
ind.....	Fifth part of UDP packet represented in ASCII

Snort generated the following alert from the log\2002.4.26\alert.ids

```
[**] [1:1616:4] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/27-09:11:04.884488 217.131.173.220:2839 -> 226.185.188.10:53
UDP TTL:48 TOS:0x0 ID:41677 IpLen:20 DgmLen:58
Len: 38
[Xref => arachnids 278][Xref => nessus 10028]
[**] [1:1616:4] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/27-09:44:49.324488 217.131.173.220:3048 -> 226.185.227.124:53
UDP TTL:48 TOS:0x0 ID:15966 IpLen:20 DgmLen:58
Len: 38
[Xref => arachnids 278][Xref => nessus 10028]
```

Alert.ids Format

Format (Alert.ids)	**Explanation of Fields**
[1:1616:4]	Rule Identifier [Snort ID: 1616 Revision ID:4)
[**] DNS named version attempt [**]	This is what Snort thinks that attack is
[Classification: Attempted Information Leak]	Information about the type of attack.
[Priority: 2]	Intermediate threat Classtype: attempted-recon
05/27-09:11:04.884488	The timestamp
217.131.173.220: 2839	Source IP address of the attack Source port number of attack
226.185.188.10: 53	Target IP address of the attack Target Port number of attack (DNS)
UDP	IP Protocol 17
TTL:48	Time To Live of packet
TOS:0x0	Type of Service
ID:41677	IP Packet ID (Value modified by GIAC)
IpLen:20	The length of the IP portion of packet
DgmLen:58	The total length of the packet (IP+UDP)
Len: 38	The UDP length of the packet
[Xref => arachnids 278]	External reference for attack
[Xref => nessus 10028]	External reference for attack

The Snort rule that triggered the above response was dns.rules

```
# (C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
#  All rights reserved.
# $Id: dns.rules,v 1.29 2003/05/14 18:07:56 cazz Exp $
...
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version
attempt"; content:"|07|version"; nocase; offset:12; content:"|04|bind";
nocase; offset: 12; reference:nessus,10028; reference:arachnids,278;
classtype:attempted-recon; sid:1616; rev:4;)
```

Snort Rule Format

Format (Alert.ids)	**Explanation of Fields**
alert	Generate an alert then log packet
udp	Protocol 17 UDP
$EXTERNAL_NET any ->	Variable defining source external network on any port to (->)
$HOME_NET 53	Variable defining destination home network port 53
msg:"DNS named version attempt";	Informational message stating what the attack is
content:"|**07**|version";	Look in the packet content for the word version and the HEX equivalent (enclosed in |**07**|)
nocase;	Ignore case of previous content
offset:12;	In the content, start looking from the 12th byte
content:"|**04**|bind";	Look in the packet content for the word bind and the HEX equivalent (enclosed in |**04**|)
nocase;	Ignore case of previous content
offset: 12;	In the content, start looking from the 12th byte
reference:nessus,10028;	External reference. Look in nessus site id 10028
reference:arachnids,278;	External reference. Look in arachnids site id 278
classtype:attempted-recon;	Pre-defined priority level of 2 for this attack
sid:1616;	Snort Unique identifier 1616
rev:4;	Revision 4

2.1.3 Probability the source address was spoofed:

Information from http://www.ripe.net/db/whois/whois.html (18 June 2003) revealed the owner of the address. The IP range is registered to "Superonline IP Master" in Istanbul, Turkey. We could contact the organisation to get more information but before we do that (as I don't speak Turkish for a start) we will try and deduce some information.

intnum:	217.131.0.0 - 217.131.255.255
netname:	TR-SUPERONLINE-980319
descr:	Provider Local Registry
country:	TR
admin-c:	SOL1-RIPE
...	

route:	217.131.128.0/17
descr:	SUPERONLINE-AS
origin:	AS6822
notify:	muratoz@superonline.net
...	

It would be difficult (not impossible) to receive data from the DNS reconnaissance unless the source address is active on the Internet.
Traceroute 217.131.173.220 performed at around 14:00 EAST revealed.

```
…
18  212.253.3.44 (212.253.3.44)  830.498 ms  828.748 ms  829.46 ms
19  217.131.130.82 (217.131.130.82)  836.885 ms  839.536 ms  837.49 ms
20  217.131.132.69 (217.131.132.69)  838.023 ms  837.649 ms  837.775 ms
21  217.131.132.65 (217.131.132.65)  845.079 ms  840.845 ms  840.668 ms
…
29  217.131.132.65 (217.131.132.65)  846.187 ms  843.593 ms  844.641 ms
30  217.131.132.69 (217.131.132.69)  847.006 ms  847.047 ms  851.34 ms
```

Traceroute 217.131.173.220 performed at around 20:00 EST revealed.

```
…
20 217.131.132.69 (217.131.132.69) 668.894 ms 674.24 ms 672.728 ms
21 217.131.173.220 (217.131.173.220) 819.287 ms 832.613 ms 825.771 ms
```

From the above 2 tables, we can guess that as the device connects to the ISP it can be accessed from the Internet. When the the device is non-active, the traceroute gets looped until timeout occurs. This would slow down possible attacks but would also create traffic for the ISP. From this information the source is most likely a SOHO and would not have an "always on" connection to the Internet.

Given all of the information presented in this section, I don't believe that this IP address was spoofed.

2.1.4 Description of attack:

This is a reconnaissance of UDP DNS; more specifically BIND. The information gathered can be used to attempt a buffer overflow. Consequences can be a denial of service; DNS cache poisoning or getting root access to the target device.

> *http://www.isc.org/products/BIND/ (18 June 2003)*
> *The BIND DNS Server is used on the vast majority of name serving machines on the Internet, providing a robust and stable architecture on top of which an organization's naming architecture can be built. The resolver library included in the BIND distribution provides the standard APIs for translation between domain names and Internet addresses and is intended to be linked with applications requiring name service.*

Further information and history of BIND can be found at http://www.isc.org/products/BIND/bind-history.html (18 June 2003)

The Snort IDS reported a "[**] DNS named version attempt [**]". "named", is the daemon name of the DNS server. What this rule is detecting is an attempt to get the version of the target hosts DNS server. With this information the attacker would then look up any known vulnerabilities and use them to exploit the DNS named service.

As this is reconnaissance probe and NOT a direct attack, all the CVEs & CANs listed can be used depending on the information were received from the target device. The command used to elicit the BIND version is legitimate as per RFC 1035 (http://www.faqs.org/rfcs/rfc1035.html) and therefore would not be classed as a direct attack and would be the prelude to such.

CVEs & CANs for BIND:

http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=BIND (22 June 2003).
CVE version: 20030402
CVE-2001-0497, CVE-2001-0013, CVE-2001-0012, CVE-2001-0011, CVE-2001-0010, CVE-2000-0888, CVE-2000-0887, CVE-1999-0851, CVE-1999-0849, CVE-1999-0848, CVE-1999-0837, CVE-1999-0835, CVE-1999-0833, CVE-1999-0184, CVE-1999-0024, CVE-1999-0011, CVE-1999-0010, CVE-1999-0009.
CANDIDATES:

CAN-2002-1221, CAN-2002-1220, CAN-2002-1219, CAN-2002-1146, CAN-2002-0684, CAN-2002-0651, CAN-2002-0400, CAN-2002-0029, CAN-1999-1499.

2.1.5 Attack mechanism:

The probe works by requesting the version of BIND named daemon from a DNS server. This is done so an attack can be launched after more information has been gathered. The reason a DNS server is a popular target is that they must be visible to the Internet in order to function. If something is visible then it can be compromised. Once this has been done an attacker gains a "foot in your front door" and will either launch an attack at another Internet visible device (making it more difficult to track the real source of the attack) or more likely, look for another step deeper into your network. Resource records can be altered so that DNS lookups are redirected to bogus sites.

The request for BIND version information is in accordance to RFC specifications and is quite simple to execute. The Detect is a stimulus and no response can be seen in the logs. Below is an example of a dig requesting the BIND version.

# dig @<server name> version.bind txt chaos

dig	*DNS lookup utility. Domain Information Groper is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried.*
server name	*The name of the DNS server that will be queried.*
txt	*Type of Resource Record in text format that is used in master files.*
chaos	*http://www.wikipedia.org/wiki/CHAOSnet (22 June 2002)* *CHAOSnet, developed at MIT in the 1970s, was one of the earliest local area network implementations. CHAOSnet can be regarded as a precursor of both Ethernet and the Internet Protocol, and was supported by early versions of the BIND DNS server.*
Version.bind	*The requested BIND version information from the DNS Server.*

```
# dig @127.0.0.1 version.bind txt chaos
…
;; QUERY SECTION:
;;      version.bind, type = TXT, class = CHAOS

;; ANSWER SECTION:
VERSION.BIND.          0S CHAOS TXT    "8.2.2-P5"
…
```

Using the information we have just captured ie DNS BIND version 8.2.2-P5, we look at the CVE and CAN table above and determine that this version is exploitable according to CVE-2001-0010: (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0010).

2.1.6 Correlations:

This DNS protocol RFC 1035 (http://www.faqs.org/rfcs/rfc1035.html) has been available since November 1987 and

James M. Bloom and Kevin J. Dunlap from the University of California, Berkeley presented a paper "Experiences Implementing BIND, A Distributed Name Server for the DARPA Internet," for the summer 1986 USENIX Conference. This paper was an

implementation for BIND on BSD Unix 4.3. It was at this time when the Detect was first available but came to prominence when the release of the various toolsets such as nslookup gave the probe a much wider audience.
Various BIND exploits derive from using the information gathered, ie **CVEs & CANs for BIND.**

2.1.7 Evidence of active targeting:

The 2002.4.26 log file has been distilled and as can be seen below, the log has eleven different IP addresses and open ports on the 226.185.0.0/16 network. In the case of the named attack, only two hosts were targeted from the one source address of 217.131.173.220.

Externally visible IP addresses and ports

1.	**226.185.188.10:53**	**Target Host 1**
2.	**226.185.227.124.53**	**Target Host 2**

It **looks** as if this attack was specifically directed at the hosts 226.185.188.10 and 226.185.227.124 on UDP port 53 by attacker 217.131.173.220. Due to the nature of a named attack (i.e. DNS) it's a trivial matter to determine DNS services. No other instances of named attacks occurred to any hosts except for UDP hosts 226.185.188.10:53 & 226.185.227.124:53.
A DNS lookup was performed to find the authoritative DNS servers for this particular domain name, which may reveal a targeted attack against a particular company or organisation. For example:

```
# nslookup
set type=ns
companyname.com
```

The results of this will show the DNS servers to a particular domain, and this may be the target of the attack.

2.1.8 Severity:

Criticality = 4

DNS Servers are important to the existence of a majority of organisations on the Internet. The size and number of the class C and class B network addresses in the various logs indicate a large organisation.

Lethality = 1

This is a probe and therefore does not pose an immediate lethal threat. There has been no loss of Confidentiality, Integrity or Availability to the organisation's data. The logs don't show that the BIND version was sent to the probe but we may not have captured that information. Also, someone from a source address that has in-itself very poor security has taken an interest in our network. This must be weighed against the sheer size of probes and attacks coming from the dubious source network. This information should be noted if required for future reference and then attention should be diverted elsewhere.

Countermeasures

No other attacks for these hosts were detected in the logs other than named.

System countermeasure = 4

There would be more evidence of a directed attack against the host.

Network countermeasure = 3
The security policy is unavailable so one point was deducted due to port 515 being open. The logs do not show much in the way of activity so it's either a quiet network or the perimeter is restricting entry.

severity = (criticality + lethality) – (system + network countermeasures)
severity = (4 + 1) – (4 + 4)
severity = -3

2.1.9 Defensive recommendation:

No evidence that the attack succeeded and no other traffic to and from the DNS service were logged. As severity is -3, the current defences are adequate for the time being. It should be noted that patching must still be kept up to date to continue with this severity level. As the criticality level is 4 for DNS service devices, a host sensor should be installed to further provide defence in depth and to give a more detailed picture of traffic patterns. It would be advisable to modify the security policy so that hosts have IDS's installed on all devices where criticality is three and higher and where lethality of attacks are also three and higher.

A useful test to use would be to enter your IP address in the command below to determine if any hosts in your network are leaking BIND information.

```
dig @`nmap -vv -n -sU -T Aggressive -p53 \`nmap -sP -vv -n -T Aggressive -sP <target IP
range> | grep "appears to be up" | cut -f2 -d"(" | cut -f 1 -d")"\` |
grep "Interesting" | cut -f2 -d"(" | cut -f 1 -d")"` txt chaos version.bind > bind.txt
```

2.1.10 Multiple choice test question:

Using the Snort output below:

```
[**] DNS named version attempt [**]
05/27-09:11:04.884488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
217.131.173.220:2839 -> 226.185.188.10:53 UDP TTL:48 TOS:0x0 ID:41677
IpLen:20 DgmLen:58
Len: 38
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 3A A2 CD 00 00 30 11 31 34 D9 83 AD DC E2 B9  .:....0.14......
0x0020: BC 0A 0B 17 00 35 00 26 09 4B 12 34 00 80 00 01  .....5.&.K.4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62  .......version.b
0x0040: 69 6E 64 00 00 10 00 03                          ind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

See next page**. . .**

Which rule was used to extract the information above

```
A) alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version
attempt"; flow:to_server,established; content:"|07|version"; nocase;
offset:12; content:"|04|bind"; nocase; nocase; offset:12;
reference:nessus,10028; reference:arachnids,278; classtype:attempted-recon;
sid:257; rev:6;)

B) alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version
attempt"; content:"|07|version"; nocase; offset:12; content:"|04|bind";
nocase; offset: 12; reference:nessus,10028; reference:arachnids,278;
classtype:attempted-recon; sid:1616; rev:4;)

C) alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:" DNS named version
attempt"; content:"|80 00 07 00 00 00 00 00 01 3F 00 01 02|";
classtype:attempted-admin; sid:314; rev:6; reference:cve,CVE-2001-0010;
reference:bugtraq,2303;)

D) alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS EXPLOIT named
overflow attempt"; flow:to_server,established; content:"|CD80 E8D7 FFFF
FF|/bin/sh"; reference:url,www.cert.org/advisories/CA-1998-05.html;
classtype:attempted-admin; sid:261;  rev:4;)
```

Answer B

Reasons :

A) and D) are TCP traffic

C) is a “DNS EXPLOIT named tsig overflow attempt”

GIAC Posting http://cert.uni-stuttgart.de/archive/intrusions/2003/07/pgp00012.pgp

To: <intrusions@incidents.org> Subject: RE: Are Portscans illegal? - Re: LOGS: GIAC GCIA Version3.3Practical Detect: 1 From: "Nicholas Cop" <Nicholas.Cop@citec.com.au> Date: Thu, 17 Jul 2003 08:41:53 +1000

Hi all,

I thought this would be a little controversial when I decided to add it to my practical but as I did not see any other practicals with this type of information and it has been bugging me about the validity of scanning hosts for the purposes of getting information about events of interest that occur in log files.
Firstly is this legal?
I'm not a lawyer but to my knowledge no one has been prosecuted for this action in my country and this is unlikely to change in the future because of the reasons below. (not even going to try about legal boundaries between countries)
Is it ethical?
To me, it depends on intent. Did I intend to use this information in the pursuit of illegal activities. Of course not. The purpose of the scan was for research and sharing of information to further enhance the knowledge of the information security professional community.
Reasoning
The information I provided in the scan is publicly available. To me this action is like walking down the street and looking at houses, for example house 1 is white, house 2 has bars on the windows, house 3 has no fence etc.
The question comes down to why was I looking at houses. Was it a precursor to an attempted break-in?, was I looking at the houses to better design my own and get ideas? or was I looking at a particular house because of curiosity aroused by the fact the this person looked at my house for some unknown reason.

Michael McDonnell makes a compelling argument when he pointed out the following
"It is not enough that a portscan simply "not damage anything, or chew up their bandwidth." If you portscan someone, and they have no prior knowledge of your activity, they cannot assume it to be benign and must investigate. Dealing with portscans is expensive; even those postscans that are from really nice guys who are certain they are not damaging anything and feel the bandwidth they consume is reasonable. None of us can assume what will and will not cause damage to another's system"
To me this comes back to intent. I believe that an open invitation was sent to look at this host computer because they showed dubious interest [version.bind info] in a network I was asked to analyse i.e. if you don't want people to notice you, don't do anything stupidly noticeable. I performed a ping sweep of a network and to me a ping is an acceptable to test whether a host is up (but not some akin to a ping-of-death), to determine what type of source host this might be [SOHO]. I then targeted the specific host that showed "interest" in the network I was asked to analyse.

This argument will no doubt lead to the question of "what if this was a compromised box and 'owned' by someone else" If I was the legal owner of the compromised host I would like someone to tell me this information, provided they did not have other intentions ie blackmail.

Let's also look at the confidentiality, integrity or availability of the source host.
confidentiality
the information is publicly available as stated above.

integrity
no information was changed.
availability
this is the complicated one but I'll try to give my answer.
As I pointed out above, an invitation was left open to investigate the source because of the type of interest shown in my the network I analysed.
Also, any device connected to the Internet has a very high probability of being scanned, it's the nature of the environment.

Agree/Disagree as I'd like to know what others think when it comes to this sort of security activity and the steps I took.

Nick

```
> -----Original Message-----
> From: Sgt B [mailto:sgt_b2002@yahoo.com]
> Sent: Wednesday, July 16, 2003 8:29 AM
> To: Michael McDonnell; intrusions@incidents.org
> Subject: Re: Are Portscans illegal? - Re: LOGS: GIAC GCIA Version 3.3
> Practical Detect: 1
>
> Portscans are legal. There is no law that says otherwise.
> Saying that, most ISPs will have a clause in their terms of
> service that prohibit this activity. If you want to look at
> it as black and white, then port scanning is legal. If you
> want to talk about ethics, then that's a different story.
> Port scanning is seen as aggressive behaviour, even if you're
> doing it for "good reasons". As an administrator, all I do is
> look at logs. I can't tell the good intentions from the bad.
> Therefore, I'll see your IP address as hostile. As stated in
> another message, I will never port scan from work. If I need
> to do that I'll do it from a seperate account.
> You mention connecting without permission. You do that
> everyday when you visit websites. Do you have written
> permission to visit www.google.com? No, those are public
> services. Is sending a SYN packet to www.google.com illegal?
> No. So by that logic, someone with port 445 open is providing
> a public service (getting grey!). Access is passworded so
> trying to access that makes it "unauthorized access",
> therefore illegal. How's that for a suspect theory! ;-)
> Either way, port scanning is legal, and you cannot be
> prosecuted for it. You can be banned from services on
> external sites, or your ISP can cut your service. So be careful!
>
> Michael McDonnell <michael@winterstorm.ca> wrote:James X wrote:
>
> >Interesting concept.
> >
> >I had not considered portscanning illegal and have often
> performed them
> >when analysing an interesting detect.
> >
> >Provided the portscan was not damaging anything, or chewing up their
> >bandwidth, why would it be iullegal? Or evan immoral?
> >
> It is not enough that a portscan simply "not damage anything,
> or chew up
> their bandwidth." If you portscan someone, and they have no prior
> knowledge of your activity, they cannot assume it to be
> benign and must
> investigate. Dealing with portscans is expensive; even those postscans
> that are from really nice guys who are certain they are not damaging
> anything and feel the the bandwidth they consume is
> reasonable. None of
> us can assume what will and will not cause damage to another's system.
>
> It is always best to get permissions first, or send notification
> simultaneously with your scan attempts.
>
>
> >If so how would you define a portscan? Surely this would be
> a very grey
> >area? If I send a packet a day is that still a scan?
> >
> Yes, it is a "low and slow" scan. These get noticed too (though less
> often) and are the subject of much interest and
> investigation. Often it
> takes more resources to investigate these than a more mundane scan.
>
> --
> Michael McDonnell, GCIA
> Winterstorm Solutions, Inc.
> michael@winterstorm.ca
> http://www.winterstorm.ca/
```

- To: <intrusions@incidents.org>, <kbjo@interpost.no>
- Subject: Re: LOGS: GIAC GCIA Version 3.3 Practical Detect: 1
- From: "Nicholas Cop" <Nicholas.Cop@citec.com.au>
- Date: Thu, 17 Jul 2003 09:10:03 +1000

I completely aggree with the comment about the difficulties in contacting the right person in ISPs when you need something done.
As to what ISP could do, education and awareness is the best option at this stage. If the ISP included in it's contract a service about portscans and information to the end user about securing these services, I think it would go a long way in locking down devices that are open to intrusion. Maybe an ISP could include in its contract with customers a automated monthly free scan service & report which shows what ports are open and how to secure them. Included with this report would be a link to a page where patch downloads are available.
That's just an idea but I think it would help the community in general.

Nick

```
>>> Knut Bjornstad <kbjo@interpost.no> 16/07/2003 16:51:42 >>>
On Tue, Jul 15, 2003 at 03:52:34PM +1000, Nicholas Cop wrote:

A few comments
> 2.1.3 Probability the source address was spoofed:
> Information from http://www.ripe.net/db/whois/whois.html (18 June 2003)
>revealed the owner of the address. The IP range is registered to
>"Superonline IP Master" in Istanbul, Turkey. We could contact the
>organisation to get more information but before we do that (as I don't
>speak Turkish for a start) we will try and deduce some information.
```

I find it hard to beleive they would not understand english. On the other hand its hard to get the attention of ISP's, even without a language problem. Mails to them tends to be routed to nontechnical staff, who then avoid answering in fear of doing anything wrong.

On the other hand, what could they do when one of their online customers have lax security. If they complain to them, they might get lots of trouble with them without getting much understandig of how to secure the boxes.

I find it somewhat remarkable that you portscan the attacker without the consent of the provider. Not that this is illegal in all cases (but it might be in Turkey, have you checked?). But it is generally frowned upon. The policy of my firm is to ban al portscans that have not been allowed in writing by the owners of the adress scanned.

I think it is better to say that it is _probable_ the adress is not spoofed. I think spoofing the adress of a Windows box with lax security might be a good idea for an attacker if he is sitting somewhere in the middle. That the box runs netbios on the internet should be no surprise, there must be millions of them! Furthermore, if you are good, you might do this on purpose...

```
--
--Knut Bjornstad -- ErgoIntegration AS ---Oslo, Norway-------
--kbjo@interpost.no -- t:47 23 14 53 36 -- mob: 901 15 917 --
```

The rest of the conversations had the same opinions, either for or against. Good topic for discussion though ☺

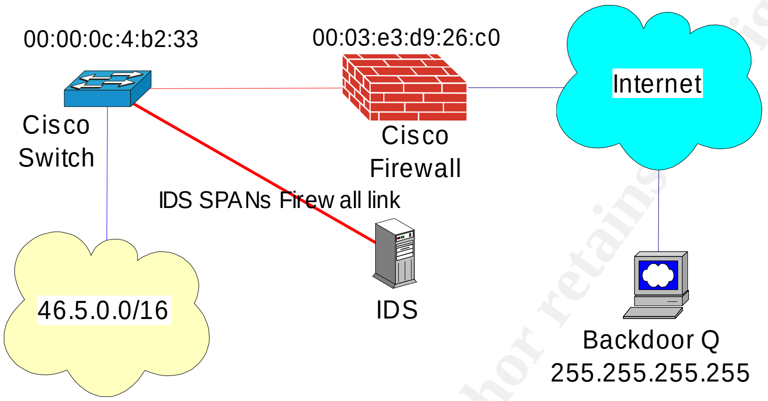
Detect #2: 2002.5.27

2.1.11 Source of Trace.

http://www.incidents.org/logs/Raw/2002.5.27 (18 June 2003)

The same MAC addresses are used as in **Detect #1 2.1.1.1 Network Architecture from 2002.4.26 logs.** There are many more hosting devices but I have summarised the IP address and limited the diagram to the log date to simplify the diagram.

2.1.11.1 Possible Network Architecture from 2002.5.27 logs



2.1.12 Detect was generated by:

Snort intrusion detection system is used as the IDS event logger.

```
snort -V
Version 1.9.0-ODBC-MySQL-WIN32 (Build 209) By Martin Roesch (roesch@sourcefire.com,
www.snort.org)
1.7-WIN32 Port By Michael Davis (mike@datanerds.net, www.datanerds.net/~mike)
1.8-1.9 WIN32 Port By Chris Reid (chris.reid@codecraftconsultants.com)
```

The required information is extracted out of the log file **2002.5.27 (**see 2.1.2 for Snort glossary)

```
snort -deXqr \nick\gcia\2002.5.27 -l \snort\logs\2002.5.27 -c snort.conf
[**] BACKDOOR Q access [**]
06/27-10:51:20.024488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
255.255.255.255:31337 -> 46.5.188.182:515 TCP TTL:14 TOS:0x0 ID:0 IpLen:20
DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 2B 00 00 00 00 0E 06 C8 18 FF FF FF FF 2E 05  .+..............
0x0020: BC B6 7A 69 02 03 00 00 00 00 00 00 00 00 50 14  ..zi..........P.
0x0030: 00 00 7C 40 00 00 63 6B 6F 00 00 00              ..|@..cko...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The above command will produce the following sample packet from the log**2002.5.27**\alert.ids file.

```
[**] [1:184:3] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
06/27-10:51:20.024488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
```

```
255.255.255.255:31337 -> 46.5.188.182:515 TCP TTL:14 TOS:0x0 ID:0 IpLen:20
DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => arachnids 203]
```

Alert.ids Format

Format (Alert.ids)	**Explanation of Fields**
[1:184:3]	Rule Identifier [Snort ID: 184 Revision ID:3)
BACKDOOR Q access	This is what Snort thinks that attack is
Classification: Misc activity	Information about the type of attack.
[Priority: 3]	Intermediate threat Classtype: Misc. Activity
06/27-10:51:20.024488	The timestamp
255.255.255.255: 31337	Source IP address of the attack Source port number of attack
46.5.188.182: 515	Target IP address of the attack Target Port number of attack (printer)
TCP	IP Protocol 06
TTL:14	Time To Live of packet
TOS:0x0	Type of Service= 0 implies normal service
ID:0	IP Packet ID of 0
IpLen:20	The length of the IP portion of packet
DgmLen:48	The total length of the packet (IP+TCP)
***A*R**	ACK & RST flags are set
Seq: 0x0	Sequence number of 0
Ack: 0x0	Acknowledged packet 0
Win: 0x0	Window size of 0
TcpLen: 20	TCP Length of 20
[Xref => arachnids 203]	External site reference for attack

The Snort rule that triggered the above response was backdoor.rules

```
# (C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
#  All rights reserved.
# $Id: dns.rules,v 1.29 2003/05/14 18:07:56 cazz Exp $
...
alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q access";
flags:A+; dsize: >1;  reference:arachnids,203; sid:184;  classtype:misc-
activity; rev:3;)
```

Snort Rule Format

Format (Alert.ids)	**Explanation of Fields**
alert	Generate an alert then log packet
tcp	Protocol 06 TCP
$255.255.255.0/24 any ->	Static class C address of source network on any port to (->)
$HOME_NET any	Variable defining destination home network on any port
msg:"BACKDOOR Q access";	Informational message stating what the attack is
flags:A+;	ACK flag set in addition to any others flags
dsize: >1;	Packet payload size must be bigger than 1 byte
reference:arachnids,203;	External reference. Look in arachnids site id 203
sid:184;	Snort Unique identifier 184
classtype:misc-activity;	Pre-defined class for this attack
rev:3;	Revision 3 of the rule

2.1.13 Probability the source address was spoofed:

The IP source address is most likely spoofed and crafted by an errant application. The attacker did not expect a response back from the target so there is a very high likelihood that this address is spoofed for the following reasons:

- Snort detection engine has defined the salient packets as Backdoor Q;
- Backdoor Q is known to use spoofed IP source addresses;
- 255.255.255.255 is not a valid Internet address;

According to RFC 919 (http://www.faqs.org/rfcs/rfc919.html): Broadcasting Internet Datagrams: October 1984

> The address 255.255.255.255 denotes a broadcast on a local hardware network, which must not be forwarded. This address may be used, for example, by hosts that do not know their network number and are asking some server for it.

2.1.14 Description of attack:

This is trial run to determine if trojans could be activated in the 46.5.0.0/16 network utilising Backdoor Q like activity.

The packet times and target IP addresses are random. This would suggest either that the attacking application targeted networks in a random manner or a sensor was not present for the missing subnet. This attempt was most likely used as a trial run, the results seen in the packet decodes show an "unadvanced" form of the penetration.

The attacker most probably did not expect a response back from the target along the same communication channel as initiated by the first attack. If trojans exist in the 46.5.0.0/16 network, then these packets would be activation commands for the trojans.

There is a generic CVE for this type of activity but is not too detailed CAN-1999-0660 (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660 - under review and will most likely pass as Christey's comment to add the words "back door" to the description field has been done)

2.1.15 Attack mechanism:

The penetration works by sending a stimulus command packet to infected hosts on the targeted network. Snort has defined this penetration as Backdoor Q because it matched the back_door.rule. This penetration attempts to hide in false positives, in other types of attacks such as backdoor attacks or both.

Could it be valid communication?

According to the logs 2002.5.27, there is no matching stimulus for the responsive ACK. Either Snort did not capture this information or it did not happen. Additionally, a RST was sent by the source IP address so that even if there were an open channel of communication to the target that was not captured by the sensor, then the RST would have closed the link. Also assuming that the sensor did not capture all the information, why would a valid application be closing multiple links with a RST instead of a FIN? It wouldn't. With the above information it is probably safe to assume that this was not a valid communication.

Further, an ACK RST is not a graceful way to close a connection and when combined with Seq: 0x0 and Win: 0x0 and would only be valid in situations such as a service not being active on the destination host. This type of packet would be the reply from the destination host such as in a situation where TCPWrappers closes a connection that is not authorised. If you factor in the packet is acknowledging sequence number 0x0, then this is not the case, as the destination host would not be responding to an initial sequence number of 0 all the time.

Is it a Backdoor Q attack?

No

> ***Q-2.4.tgz (*[*http://mixter.void.ru/about.html*](http://mixter.void.ru/about.html) ***25 June 2003) README file:***
> *"The Q client/server suite is designed to provide a maximum of security, especially confidentiality and anonymity."*

Backdoor Q is a stealthy, remote access tool for the TCP, UDP and ICMP protocols and can be complied to use any source socket. To determine if the Snort alert.ids file correctly reported that the traffic is Backdoor Q, characteristics must be examined to determine if they adhere to the description The following was command was executed to determine if all the traffic that Snort alerted upon is the same.

```
1. # snort -qvr 2002.5.27 src 255.255.255.255 | wc
    211     848    9796
2. # snort -qvr 2002.5.27 src 255.255.255.255 and src port 31337 and dst
   net 46.5.0.0/16 and dst port 515 | wc
    211     848    9795
```

The first command shows the number of entries of any source packet of 255.255.255.255 exactly matches the number of entries in the second so it is safe to assume that all the logs in 2002.5.27 have the characteristics of the second command. A section of the output of the second command is shown below.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
06/27-11:12:17.044488 255.255.255.255:31337 -> 46.5.27.52:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
... (Removed for brevity)
```

Is the packet TCP, UDP or ICMP traffic

Yes, TCP

Is the packet stealthy?

No. The source IP address is always 255.255.255.255. This packet should be dropped at the perimeter of the network and/or not be allowed out of the local LAN. The source port is always 31337, which is used for programs such as Back Orifice and similar[1]. The difference being is that it is usually the destination port that is 31337. The choice of this IP source socket does not support a covert action.

Port 515 is listed in /etc/services as

```
# grep 515 /etc/services
printer         515/tcp         spooler         # line printer spooler
printer         515/udp         spooler         # line printer spooler
```

[1] http://isc.incidents.org/port_details.html?port=31337&repax=2&tarax=2&srcax=2&percent=N&days=40&Redraw=Submit+Query (18 June 2003)

The destination port is 515 and in most networks this traffic would be dropped at the firewall. Even if there is a business justification for this service to be web visible, the logs show that multiple IP addresses are targeted in the 46.5.0.0/16 network. This is not stealthy.

Is it a false positive?

No.

Definition: False positive (http://www.itsecurity.com/dictionary/dictionary.htm) 18 June 2003
A false positive is a term applied to a failure in an alerting system - most commonly in an anti-virus product or intrusion detection system. It occurs when a virus or intrusion condition is incorrectly reported; that is, the alerting system reports a virus or intrusion condition that does not exist. Too many false positives can be very intrusive.

The wording of the definition must be examined. The sensor is alerting to an intrusion which does exist, it's just not the one the that sensor thinks it is! As some sort of attack does exist, it should not be classified as a false positive.

As the packet characteristics are a poor choice for penetration they appear to be intentionally crafted in an attempt to hide the packets in backdoor type traffic, making this Detect look like a trial run. The majority of border routers would not allow this type of traffic in, let alone trying to get through firewalls, bastion hosts and IDS's.

2.1.16 Correlations:

Attacks with these packet characteristics (including RST cko in the content) have been reported by various sources. On May 4 2001, Jeff Peterson[2] reported traffic of the same nature as this Detect. In addition, he also reported packets without the cko content. The TTL for cko packets is 12, which differs from our packet TTL of 14.

Jason Storm[3] responded with comments about IRC as the target. Also note that the log that Jason supplied has source addresses other than 255.255.255.255.

Im inclined to think this is some sort of worm.. and its definately doing its thing on IRC (the ip's that drew the packets are all used almost exclusively for irc);

```
Mar 1 21:35:36 asspuma 11: %SEC-6-IPACCESSLOGP: list 101 denied tcp
209.196.44.58(31337) -> 64.149.133.155(515), 1 packet Mar 6 15:43:36
asspuma 445: %SEC-6-IPACCESSLOGP: list 101 denied tcp 209.196.44.58(31337)
-> 64.149.133.58(515), 1 packet Mar 16 06:54:21 asspuma 1696: %SEC-6-
IPACCESSLOGP: list 101 denied tcp 209.112.47.7(31337) ->
64.149.133.33(515), 1 packet
```

Les Gordons's GCIA paper[4] on Backdoor Q provided excellent examples and insights into this traffic behaviour and also in the log analysis of similar traffic. (If you intend on researching Q, then this would be the place to start.)

There are many more examples such as the one above but they all have similar characteristics. The best correlation would be the source of the Q program. A gentleman calling himself Mixter[5] was the author of the program and this attack is

[2] http://lists.jammed.com/incidents/2001/05/0037.html (25 Jun2 2003)
[3] http://lists.jammed.com/incidents/2001/05/0039.html (25 June 2003)
[4] http://www.giac.org/practical/GCIA/Les_Gordon_GCIA.doc (25 June 2003)
[5] http://mixter.void.ru/about.html (25 June 2003)

based on his program. Using Q, the instigators of this Detect are trialing communication methods.

2.1.17 Evidence of active targeting:

The target IP address are not directly targeted, the attack is focusing on the destination port of 515.

From the point of view of the Snort detect, the attack **may** have initially originated from the Internet but being sourced from 46.5.0.0/16 network ie a host was compromised and used to send packets with a broadcast address. The range of targeted addresses is very wide ie 41 various address, in different class C ranges on the 46.5.0.0/16 network were logged by Snort. It seems unlikely that a class B address would not be subnetted by the network administrators so a broadcast like this one would not permeate throughout the various addresses shown in the Detect #2 targets below. Setting the home directory as the attacking IP source address 255.255.255.0/24 identified the targets with the command below.

```
snort -vr 2002.5.27 -l \snort\logs\2002.5.27 -c snort.conf -h 255.255.255.0/24
```

Detect #2 targets

46.5.1.71
46.5.101.204
46.5.137.244
46.5.141.51
46.5.143.40
46.5.147.227
46.5.147.94
46.5.148.219
46.5.172.11
46.5.183.215
46.5.188.182
46.5.189.61
46.5.19.222
46.5.193.238
46.5.198.218
46.5.199.168
46.5.20.22
46.5.202.139
46.5.205.240
46.5.208.223
46.5.21.213
46.5.221.17
46.5.221.52
46.5.221.78
46.5.23.92
46.5.236.171
46.5.236.242
46.5.238.180
46.5.24.65
46.5.240.140
46.5.248.198
46.5.26.83
46.5.27.52
46.5.28.69
46.5.3.72
46.5.55.39
46.5.78.185
46.5.79.52
46.5.85.94
46.5.9.208
46.5.92.96
46.5.94.8

Given the information above it would be most unlikely if the attack was piggybacked for within the 46.5.0.0/16 network. Unfortunately, the closest contiguous IP target addresses are spaced at 20 IPs apart (238 - 218 = 20). The IP addresses are 46.5.193.238 & 46.5.198.218. The time difference in the logs is 27 minutes 03 seconds. This would make an automated attack between them take 1 minute 21 seconds. Following this logic we should be able to calculate other attack time frames if the attack was automated from one source and the attacking program was not randomly sequenced. Trying this on several IP addresses did not validate our findings so the conclusion is reached that the attack times are random.

2.1.18 Severity:

Criticality = 5
This attack targets all hosts.
Lethality = 4
A bit difficult to ascertain because the attack is targeting already infected hosts with a trojan. As this attack is a trigger and also a primitive trial run program I have down graded it.
System countermeasure = 5

The logs show no evidence of return traffic but this doesn't mean that a host responded on a different source/destination socket pair.

Network countermeasure = 1

Broadcast address and port 515 should not have been allowed in.

severity = (criticality + lethality) – (system + network countermeasures)
severity = (5+4) – (5 +1)
severity = 3

2.1.19 Defensive recommendation:

Defences require patching. Unless there is a business requirement for TCP port 515 to be allowed through the firewall, I would recommend closing this port. There would be limited justification to allow direct printing from the Internet to an Internal 46.5.0.0/16 network running the printer service. Also, the IIS5.0 security checklist is available at: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/iis5cl.asp.

Often, if you implement the recommendations in this checklist, you are protected against vulnerabilities, even if you have not patched your server, as the recommendations turn off features which typically cause security problems.

The System and Network Administrators should audit their devices for the purpose of creating a baseline traffic pattern. Traffic should only originate from the Internal network unless there is a business requirement exception (ie use at minimum a stateful firewall). Bastion hosts acting as proxies can be used to "break" traffic connections to Internal devices. Internally initiated outgoing traffic should be confined to the business requirements and compared with normal traffic patterns (easier said then done but you should give it a try). Finally, ensure no broadcasting is allowed of the local LAN ie harden your routers.

```
alert ICMP 255.255.255.0/24 any -> $INTERNAL any (msg: "IDS202/backdoor-Q-
icmp"; itype: 0; dsize: ">1";)
alert TCP 255.255.255.0/24 any -> $INTERNAL any (msg: "IDS203/backdoor-Q-
tcp"; flags: A; dsize: ">1";)
alert UDP 255.255.255.0/24 any -> $INTERNAL any (msg: "IDS201/backdoor-Q-
udp"; dsize: ">1";)
```

The above Snort rules will catch similar activity on other IP protocols so this needs to be added to the IDS.

2.1.20 Multiple choice test question:

What is most unusual about the packet shown below?

```
06/19-09:53:11.952242 $EXTERNAL_NET:21 -> $MY_NET:1880
TCP TTL:255 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
*****R*F Seq: 0x0 Ack: 0xD9E9F0C9  Win: 0x0  TcpLen: 20
```

A) Nothing is unusual
B) TTL:255
C) Seq: 0x0
D) RST, FIN flags are set

Answer: D

Reasons: These two flags should never be seen together. The other parameters are possible in some form or another.

2.2 Detect #3: 2002.8.14

2.2.1 Source of Trace.

http://www.incidents.org/logs/Raw/2002.8.14 (18 June 2003)

The same MAC addresses are used as in **Detect #1 2.1.1.1 Network Architecture from 2002.4.26 logs.** There are many more hosting devices but I have summarised the IP address and limited the diagram to the log date to simplify the diagram.

2.2.2 Detect was generated by:

Snort intrusion detection system is used as the IDS event logger.

snort -V

Version 1.9.0-ODBC-MySQL-WIN32 (Build 209) By Martin Roesch (roesch@sourcefire.com, www.snort.org)
1.7-WIN32 Port By Michael Davis (mike@datanerds.net, www.datanerds.net/~mike)
1.8-1.9 WIN32 Port By Chris Reid (chris.reid@codecraftconsultants.com)

The required information is extracted out of the log file **2002.8.14 (**see 2.1.2 for Snort glossary)

```
snort -dXqr \nick\gcia\2002.8.14 -l \snort\log\2002.8.14 -c snort.conf
```

The above command will produce the following sample packet from file log**2002.8.14**\163.20.13.10\ IP_FRAG.ids.

```
[**] BAD TRAFFIC bad frag bits [**]
09/15-06:50:02.976507 163.20.13.10 -> 115.74.186.97 TCP TTL:109 TOS:0x0
ID:40450 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 05 BC 9E 02 60 00 6D 06 FE 00 A3 14 0D 0A 73 4A  ....`.m.......sJ
0x0020: BA 61 04 75 00 50 77 B5 09 DF 99 C8 61 8D 50 18  .a.u.Pw.....a.P.
0x0030: 44 70 FD 14 00 00 47 45 54 20 2F 64 65 66 61 75  Dp....GET /defau
0x0040: 6C 74 2E 69 64 61 3F 4E 4E 4E 4E 4E 4E 4E 4E 4E  lt.ida?NNNNNNNNN
0x0050: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0060: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0070: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0080: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0090: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00A0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00B0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00C0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00D0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00E0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00F0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0100: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0110: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0120: 4E 4E 4E 4E 4E 4E 4E 25 75 39 30 39 30 25 75 36  NNNNNNN%u9090%u6
0x0130: 38 35 38 25 75 63 62 64 33 25 75 37 38 30 31 25  858%ucbd3%u7801%
0x0140: 75 39 30 39 30 25 75 36 38 35 38 25 75 63 62 64  u9090%u6858%ucbd
0x0150: 33 25 75 37 38 30 31 25 75 39 30 39 30 25 75 36  3%u7801%u9090%u6
0x0160: 38 35 38 25 75 63 62 64 33 25 75 37 38 30 31 25  858%ucbd3%u7801%
0x0170: 75 39 30 39 30 25 75 39 30 39 30 25 75 38 31 39  u9090%u9090%u819
0x0180: 30 25 75 30 30 63 33 25 75 30 30 30 33 25 75 38  0%u00c3%u0003%u8
0x0190: 62 30 30 25 75 35 33 31 62 25 75 35 33 66 66 25  b00%u531b%u53ff%
0x01A0: 75 30 30 37 38 25 75 30 30 30 30 25 75 30 30 3D  u0078%u0000%u00=
0x01B0: 61 20 20 48 54 54 50 2F 31 2E 30 0D 0A 43 6F 6E  a  HTTP/1.0..Con
0x01C0: 74 65 6E 74 2D 74 79 70 65 3A 20 74 65 78 74 2F  tent-type: text/
```

```
0x01D0: 78 6D 6C 0A 48 4F 53 54 3A 77 77 77 2E 77 6F 72  xml.HOST:www.wor
0x01E0: 6D 2E 63 6F 6D 0A 20 41 63 63 65 70 74 3A 20 2A  m.com. Accept: *
0x01F0: 2F 2A 0A 43 6F 6E 74 65 6E 74 2D 6C 65 6E 67 74  /*.Content-lengt
0x0200: 68 3A 20 33 35 36 39 20 0D 0A 0D 0A 55 8B EC 81  h: 3569 ....U...
0x0210: EC 18 02 00 00 53 56 57 8D BD E8 FD FF FF B9 86  .....SVW........
0x0220: 00 00 00 B8 CC CC CC CC F3 AB C7 85 70 FE FF FF  ............p...
0x0230: 00 00 00 00 E9 0A 0B 00 00 8F 85 68 FE FF FF 8D  ...........h....
0x0240: BD F0 FE FF FF 64 A1 00 00 00 00 89 47 08 64 89  .....d......G.d.
0x0250: 3D 00 00 00 00 E9 6F 0A 00 00 8F 85 60 FE FF FF  =.....o.....`...
0x0260: C7 85 F0 FE FF FF FF FF FF FF 8B 85 68 FE FF FF  ............h...
0x0270: 83 E8 07 89 85 F4 FE FF FF C7 85 58 FE FF FF 00  ...........X....
0x0280: 00 E0 77 E8 9B 0A 00 00 83 BD 70 FE FF FF 00 0F  ..w.......p.....
0x0290: 85 DD 01 00 00 8B 8D 58 FE FF FF 81 C1 00 00 01  .......X........
0x02A0: 00 89 8D 58 FE FF FF 81 BD 58 FE FF FF 00 00 00  ...X.....X......
0x02B0: 78 75 0A C7 85 58 FE FF FF 00 00 F0 BF 8B 95 58  xu...X.........X
0x02C0: FE FF FF 33 C0 66 8B 02 3D 4D 5A 00 00 0F 85 9A  ...3.f..=MZ.....
0x02D0: 01 00 00 8B 8D 58 FE FF FF 8B 51 3C 8B 85 58 FE  .....X....Q<..X.
0x02E0: FF FF 33 C9 66 8B 0C 10 81 F9 50 45 00 00 0F 85  ..3.f.....PE....
0x02F0: 79 01 00 00 8B 95 58 FE FF FF 8B 42 3C 8B 8D 58  y.....X....B<..X
0x0300: FE FF FF 8B 54 01 78 03 95 58 FE FF FF 89 95 54  ....T.x..X.....T
0x0310: FE FF FF 8B 85 54 FE FF FF 8B 48 0C 03 8D 58 FE  .....T....H...X.
0x0320: FF FF 89 8D 4C FE FF FF 8B 95 4C FE FF FF 81 3A  ....L.....L....:
0x0330: 4B 45 52 4E 0F 85 33 01 00 00 8B 85 4C FE FF FF  KERN..3.....L...
0x0340: 81 78 04 45 4C 33 32 0F 85 20 01 00 00 8B 8D 58  .x.EL32.. .....X
0x0350: FE FF FF 89 8D 34 FE FF FF 8B 95 54 FE FF FF 8B  .....4.....T....
0x0360: 85 58 FE FF FF 03 42 20 89 85 4C FE FF FF C7 85  .X....B ..L.....
0x0370: 48 FE FF FF 00 00 00 00 EB 1E 8B 8D 48 FE FF FF  H...........H...
0x0380: 83 C1 01 89 8D 48 FE FF FF 8B 95 4C FE FF FF 83  .....H.....L....
0x0390: C2 04 89 95 4C FE FF FF 8B 85 54 FE FF FF 8B 8D  ....L.....T.....
0x03A0: 48 FE FF FF 3B 48 18 0F 8D C0 00 00 00 8B 95 4C  H...;H.........L
0x03B0: FE FF FF 8B 02 8B 8D 58 FE FF FF 81 3C 01 47 65  .......X....<.Ge
0x03C0: 74 50 0F 85 A0 00 00 00 8B 95 4C FE FF FF 8B 02  tP........L.....
0x03D0: 8B 8D 58 FE FF FF 81 7C 01 04 72 6F 63 41 0F 85  ..X....|..rocA..
0x03E0: 84 00 00 00 8B 95 48 FE FF FF 03 95 48 FE FF FF  ......H.....H...
0x03F0: 03 95 58 FE FF FF 8B 85 54 FE FF FF 8B 48 24 33  ..X.....T....H$3
0x0400: C0 66 8B 04 0A 89 85 4C FE FF FF 8B 8D 54 FE FF  .f.....L.....T..
0x0410: FF 8B 51 10 8B 85 4C FE FF FF 8D 4C 10 FF 89 8D  ..Q...L....L....
0x0420: 4C FE FF FF 8B 95 4C FE FF FF 03 95 4C FE FF FF  L.....L.....L...
0x0430: 03 95 4C FE FF FF 03 95 4C FE FF FF 03 95 58 FE  ..L.....L.....X.
0x0440: FF FF 8B 85 54 FE FF FF 8B 48 1C 8B 14 0A 89 95  ....T....H......
0x0450: 4C FE FF FF 8B 85 4C FE FF FF 03 85 58 FE FF FF  L.....L.....X...
0x0460: 89 85 70 FE FF FF EB 05 E9 0D FF FF FF E9 16 FE  ..p.............
0x0470: FF FF 8D BD F0 FE FF FF 8B 47 08 64 A3 00 00 00  .........G.d....
0x0480: 00 83 BD 70 FE FF FF 00 75 05 E9 38 08 00 00 C7  ...p....u..8....
0x0490: 85 4C FE FF FF 01 00 00 00 EB 0F 8B 8D 4C FE FF  .L..........L..
0x04A0: FF 83 C1 01 89 8D 4C FE FF FF 8B 95 68 FE FF FF  ......L.....h...
0x04B0: 0F BE 02 85 C0 0F 84 8D 00 00 00 8B 8D 68 FE FF  .............h..
0x04C0: FF 0F BE 11 83 FA 09 75 21 8B 85 68 FE FF FF 83  .......u!..h....
0x04D0: C0 01 8B F4 50 FF 95 90 FE FF FF 3B F4 90 43 4B  ....P......;..CK
0x04E0: 43 4B 89 85 34 FE FF FF EB 2A 8B F4 8B 8D 68 FE  CK..4....*....h.
0x04F0: FF FF 51 8B 95 34 FE FF FF 52 FF 95 70 FE FF FF  ..Q..4...R..p...
0x0500: 3B F4 90 43 4B 43 4B 8B 8D 4C FE FF FF 89 84 8D  ;..CKCK..L......
0x0510: 8C FE FF FF EB 0F 8B 95 68 FE FF FF 83 C2 01 89  ........h.......
0x0520: 95 68 FE FF FF 8B 85 68 FE FF FF 0F BE 08 85 C9  .h.....h........
0x0530: 83 BD 50 FE FF FF 00 75 26 8B F4 6A 00 8D 85 4C  ..P....u&..j...L
0x0540: FE FF FF 50 8B 8D 68 FE FF FF 51 8B 55 08 8B 42  ...P..h...Q.U..B
0x0550: 08 50 FF 95 6C FE FF FF 3B F4                    .P..l...;.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Snort generated the following alert from the log**2002.8.14**\alert.ids

```
[**] [1:1322:4] BAD TRAFFIC bad frag bits [**]
[Classification: Misc activity] [Priority: 3]
09/15-06:50:02.976507 163.20.13.10 -> 115.74.186.97 TCP TTL:109 TOS:0x0
ID:40450 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
```

Alert.ids Format

Format (Alert.ids)	**Explanation of Fields**
[1:1322:4]	Rule Identifier [Snort ID: 184 Revision ID:3)
BAD TRAFFIC bad frag bits	This is what Snort thinks that attack is
Classification: Misc activity	Information about the type of attack.
[Priority: 3]	Intermediate threat Classtype: Misc. Activity
09/15-06:50:02.976507	The timestamp
163.20.13.10	Source IP address of the attack
115.74.186.97	Target IP address of the attack
TCP	IP Protocol 06
TTL:109	Time To Live of packet is 109 hops
TOS:0x0	Type of Service= 0 implies normal service
ID:40450	IP Packet ID of 40450 (modified by GIAC)
IpLen:20	The length of the IP portion of packet
DgmLen:1468	The total length of the packet (IP+TCP)
DF	Don't fragment
MF	More fragments to follow
Frag Offset: 0x0000	First fragment as the offset is 0
Frag Size: 0x05A8	Size of the fragment

The Snort rule that triggered the above response was bad_traffic.rules

```
# (C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
#  All rights reserved.
# $Id: dns.rules,v 1.29 2003/05/14 18:07:56 cazz Exp $
...
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"BAD TRAFFIC bad frag
bits"; fragbits:MD; sid:1322; classtype:misc-activity; rev:4;)
```

Snort Rule Format

Format (Alert.ids)	**Explanation of Fields**
alert	Generate an alert then log packet
ip	IP Protocol
$ EXTERNAL_NET any ->	Variable defining any IP address to (->)
$HOME_NET any	Variable defining destination home network on any port
msg:"BAD TRAFFIC bad frag bits";	Informational message stating what the attack is
fragbits:MD;	More fragments AND don't fragment are both set
sid:1322;	Snort Unique identifier 1322
classtype:misc-activity;	Pre-defined class for this attack
rev:4;	Revision 4 of the rule

2.2.3 Probability the source address was spoofed:

Some possibilities are:

1. The attack may have originated from a host infected with a working variant.
2. The source IP may have been manually crafted to appear to come from a real IP address after the attacker determined that the source host was already infected with CodeRed.

3. The attacker picked random IP sources.

Looking up the source and destination addresses at the various registries, the following information is found:

http://www.apnic.net/apnic-bin/whois.pl
***inetnum**: 163.20.0.0 - 163.20.255.255 netname: TANET descr: Taiwan Academic Network descr: Ministry of Education computer Center descr: 12F, No 106, Sec. 2, Heping E. Rd., Taipei country: TW*

http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=213.106.221.197&do_search=Search
***inetnum**: 80.7.128.0 - 80.7.151.255 netname: NTL descr: NTL Infrastructure - Oxford country: GB*

http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=213.106.221.197&do_search=Search
***inetnum**: 213.106.216.0 - 213.106.223.255 netname: NTL descr: NTL Internet - Brentford site country: GB*

http://ws.arin.net/cgi-bin/whois.pl
NetRange: 96.0.0.0 - 126.255.255.255
OrgName: Internet Assigned Numbers Authority OrgID: IANA Address: 4676 Admiralty Way, Suite 330 City: Marina del Rey StateProv: CA PostalCode: 90292-6695 Country: US

Using the arguments presented further in this document, point one (above) was discounted as the variant does not appear to make a full TCP connection.

Point two has merit as the attacker has familiarity with CodeRed and the attacker can produce a variant, then they would have no trouble in using the original to infect hosts. The attacker did not get this variant working in this instance and may have just been using trial and error methodology to attempt to determine how perimeter defences would react to bogus CodeRed packets. Playing with code and seeing what it does to systems would hold most interest for me and I would already have a working understanding of the code so I would use already infected hosts IP source addresses. Also the TTL vales from **Frame Similarities** read in conjunction with the real source IP locations are too coincidental to believe that the hops counts are so close.

Point 3 was not chosen as the attacker appears to have familiarity with the CodeRed and it would be trivial for them to use infected host addresses to make it appear like a CodeRed attack.

2.2.4 Description of attack:

This appears to be an attempt at a CodeRed variant, of which the definition below can be found in CVE-2001-0500 at

> *http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0500 (3 July 2003) Buffer overflow in ISAPI extension (idq.dll) in Index Server 2.0 and Indexing Service 2000 in IIS 6.0 beta and earlier allows remote attackers to execute arbitrary commands via a long argument to Internet Data Administration (.ida) and Internet Data Query (.idq) files such as default.ida, as commonly exploited by Code Red.*

The more fragments AND the do not fragment bits are both set. Logically, this will not work because if the do not fragment bit is set, how can more fragments follow. Different devices react differently under this stimulus. (Maybe this is what the attacker was looking for?)

Snort was run again with the bad_traffic.rule disabled and this packet was not detected. See 2.3.5 Defensive Recommendations for a Snort rule.

The rule that was used to alert on this detect did not print the source and destination port numbers. Using tcpdump, the following information was discovered.

tcpdump -nnr /tmp/2002.8.14

```
06:50:02.976507 163.20.13.10.1141 > 115.74.186.97.80: P
2008353247:2008354675(1428) ack 2580046221 win 17520 (frag 40450:1448@0+)
```

tcpdump Glossary:

```
-n     Don't convert host addresses to names.  This can be used to avoid
DNS lookups.

-nn    Don't convert protocol and port numbers etc. to names either.

-vvv   Even more verbose output.  For example, telnet SB ... SE options are
printed in full.  With -X telnet options are printed in hex as well.

-r     Read packets from file (which was created with the -w option).
Standard input is used if file is ``-''.

-X     When  printing  hex,  print  ascii  too.   Thus  if  -x is also set,
the packet is printed in hex/ascii.  This is very handy for analysing new
protocols.  Even if -x is not also set, some parts of some packets may be
printed in hex/ascii.
```

The following packets were extracted from the tcpdump command above.

....

Frame 13. (Detect of Interest)

```
06:50:02.976507 163.20.13.10.1141 > 115.74.186.97.80: P [bad tcp cksum ae51!]
2008353247:2008354675(1428) ack 2580046221 win 17520 (frag 40450:1448@0+)
(ttl 109, len 1468, bad cksum fe00!)
```

Frame 14.

```
07:03:20.616507 80.7.150.154.2841 > 115.74.179.113.80: . [bad tcp cksum 9db3!]
3941285652:3941287080(1428) ack 1074987552 win 17520 (frag 5585:1448@0+)
(ttl 110, len 1468, bad cksum 559f!)
```

. . . .
Frame 23. (Same CodeRed variant with no "continuation" frame ie Frame 14)

```
09:04:20.206507 213.106.221.197.4461 > 115.74.127.253.80: P [bad tcp cksum
b050!] 1285553056:1285554484(1428) ack 255088721 win 17520 (frag
28991:1448@0+) (ttl 112, len 1468, bad cksum 6014!)
```

. . . .

Frame 13 is the focus of the Detect, Frame 14 & 23 were extracted (from the same log) due to their similar characteristics to Frame 13.

Frame Similarities

Frame 13	Frame 14	Frame 23
1. Src IP 163.20.13.10	1. Src IP 80.7.150.154	1.Src IP 213.106.221.197
2. Dst IP 115.74.186.97	2. Dst 115.74.179.113	2. Dst 115.74.127.253
3. Source port 1141	3. Source port 2841	3. Source port 4461
4. Destination port 80	4. Destination port 80	4. Destination port 80
5. Packet size ack 1428	5. Packet size ack 1428	5. Packet size ack 1428
6. PSH & ACK flags set	6. PSH & ACK flags set	6. PSH & ACK flags set
7. Window 17520	7. Window 17520	7. Window 17520
8. First frag & size 1448@0+	8. First frag & size 1448@0+	8. First frag & size 1448@0+
9. Length 1468	9. Length 1468	9. Length 1468
10. TTL 109	10. TTL 110	10. TTL 112
11. Don't Fragment & More Fragments to follow set	11. Don't Fragment & More Fragments to follow set	11. Don't Fragment & More Fragments to follow set

Similarities in the packet exist in that all Frames have the same values for points 4, 5, 6, 7, 8, 9, 11.
What also unusual is that the TTLs are very close indicating that either the number of hops to this destination range from the Ministry of Education computer Center in Taiwan and NTL Infrastructure - Oxford, England are almost identical or that the packet is crafted. Is most likely the latter also considering that the more and don't fragment bits are both set.

Frames 13 & 23 above have the CodeRed payload default.ida string. Frame 14, when examined by Ethereal was documented as a "Continuation" packet. It could not be determined how Ethereal arrived at this conclusion based on visual inspections of the IP ID numbers and Source & Destination socket pairs. If "continuation" was true, the TTL hop count difference between Frames 13 & 14 (ie one) is difficult to believe, given that the target devices are on differing class C networks. Too many inconsistencies give the impression of packet crafting.

It is also worth noting that some devices are susceptible to a side effect of the worm. For example, some Cisco 600 series DSL routers will stop forwarding traffic until power-cycled. The attacker may have known this and tried to find a way to bypass the patches and upgrades using the MF and DF bits set in an attempt to corrupt higher end Cisco devices.

2.2.5 Attack mechanism:

This attack is focusing on port 80 as this port is required to make a presence known on the Internet for web services. Although it's possible to run web services on other ports, port 80 is the "public address port" for the Internet. Running web service on other ports will make it difficult to have the general public view your website. This is sometimes done for the purposes of security through obscurity.

Andy Norman, Matthew Williamson wrote up this description of Code Red
http://www.hpl.hp.com/techreports/2002/HPL-2002-111.pdf
The virus affected versions 4 and 5 of Microsoft's IIS web server, exploiting a buffer overflow vulnerability in the indexing service. The attack consisted of a specially crafted HTTP request that when sent to IIS would cause malicious code to take control of the web server. The primary behaviour of the malicious code was to attempt to propagate as rapidly as possible, by generating IP addresses at random1, and making infective HTTP requests to those addresses. If any of these machines were running vulnerable installations of IIS, they too would become infected. The code attempted to propagate at an incredible rate, with many HTTP requests being sent every second. It has been estimated that the virus could infect on the order of half a million IP addresses a day [2]. The secondary behaviour of Code Red was to deface web sites on the infected host, and prepare the infected machine to participate in a distributed denial of service (DDOS) attack on www.whitehouse.gov, at certain times. Later variants of Code Red (e.g. Code Red II [3]) left Trojan horses and open shares on the compromised machine.

CodeRed requires an established TCP connection to infect other hosts. This variant does not attempt to complete the connection as the IP source address and port has changed. This may explain why Ethereal believes that Frame 14 is a continuation of Frame 13 even though packet characteristics change.

The variant may be pushing data onto a host and may still attempt to place a trojan on the system allowing a backdoor to the infected host. Given this information, the variant is trying to bypass the patches that have been posted for the source worm CodeRed. This would explain the IP Must Fragment & Don't Fragment bits both being set to 1. Perhaps trial and error attempts to see what the results will yield.

Another explanation of the fragmentation is that the first packet was logged by Snort due to the MD & DF bits set and subsequent packets were not detected. The router before the host may have overridden the DF call, fragmented the packet and left the DF bit set before passing the packet to the host.

2.2.6 Correlations:

Security Focus (http://www.securityfocus.com) / ARIS Incident Analysts contacted eEye about this worm. Ryan Permeh and Marc Maiffret from eEye Digital Security performed an analysis of this new worm. Also, Ryan Permeh commented all of the assembly code for better understanding of the worm. See http://www.eeye.com/html/advisories/coderedII.zip for the full details.

Mike Wyman in his GIAC GCIA practical has reported the variations mentioned in this paper. http://cert.uni-stuttgart.de/archive/intrusions/2002/12/msg00204.html.

2.2.7 Evidence of active targeting:

The target IP address are not directly targeted, the attack is focusing on the destination port of 80. The attacker most likely had known which IP addresses to target as this seems to be a test of a CodeRed variant. As can be seen below there were a few anomalous packets directed to port 80. Without going into too much detail for other detects, a pattern of non-normal traffic is present in these logs.

tcpdump -nnr 2002.8.14 port 80

```
05:58:16.486507 115.74.249.65.64308 > 208.33.48.103.80: P [bad tcp cksum
654!] 2924492733:2924493193(460) ack 115835223 win 17520 (DF) (ttl 124, id
25742, len 500, bad cksum baf3!)
05:58:16.576507 115.74.249.65.64308 > 208.33.48.103.80: P
1486310639:1486312559(1920) ack 2808657971 win 33580 [tos 0x10]  (ttl 240,
id 0, len 1960, bad cksum 0!)
05:58:25.106507 115.74.249.65.64310 > 208.33.48.103.80: P [bad tcp cksum
5d65!] 2926802176:2926802659(483) ack 641787330 win 17520 (DF) (ttl 124, id
25834, len 523, bad cksum ba80!)
05:58:25.116507 115.74.249.65.64310 > 208.33.48.103.80: P [bad tcp cksum
9b2f!] 2009952450:2009953910(1460) ack 2285014847 win 33580 [tos 0x10]
(ttl 240, id 0, len 1500, bad cksum 0!)
05:58:25.296507 115.74.249.65.64310 > 208.33.48.103.80: P [bad tcp cksum
cde4!] 2009953304:2009953786(482) ack 2285015331 win 33580 [tos 0x10]  (ttl
240, id 0, len 522, bad cksum 0!)
05:59:23.136507 115.74.249.65.64458 > 208.33.48.103.80: P [bad tcp cksum
5d65!] 2948993232:2948993715(483) ack 3649856443 win 17520 (DF) (ttl 124,
id 27759, len 523, bad cksum b2fb!)
05:59:23.146507 115.74.249.65.64458 > 208.33.48.103.80: P [bad tcp cksum
9f60!] 700863211:700864671(1460) ack 3594104086 win 33580 [tos 0x10]  (ttl
240, id 0, len 1500, bad cksum 0!)
05:59:23.486507 115.74.249.65.64458 > 208.33.48.103.80: P [bad tcp cksum
f02c!] 700864067:700864549(482) ack 3594104570 win 33580 [tos 0x10]  (ttl
240, id 0, len 522, bad cksum 0!)
06:50:02.976507 163.20.13.10.1141 > 115.74.186.97.80: P [bad tcp cksum
ae51!] 2008353247:2008354675(1428) ack 2580046221 win 17520 (frag
40450:1448@0+) (ttl 109, len 1468, bad cksum fe00!)
07:03:20.616507 80.7.150.154.2841 > 115.74.179.113.80: . [bad tcp cksum
9db3!] 3941285652:3941287080(1428) ack 1074987552 win 17520 (frag
5585:1448@0+) (ttl 110, len 1468, bad cksum 559f!)
07:15:32.336507 163.23.216.66.84 > 115.74.127.126.80: . [bad tcp cksum
6e6e!] 181:181(0) ack 0 win 1400 (ttl 48, id 59936, len 40, bad cksum
c41e!)
07:31:28.866507 163.23.216.66.84 > 115.74.41.231.80: . [bad tcp cksum
706d!] 326:326(0) ack 0 win 1400 (ttl 48, id 20771, len 40, bad cksum
b3b1!)
08:25:55.436507 213.107.134.109.3160 > 115.74.42.52.80: . [bad tcp cksum
5f35!] 815033936:815035352(1416) ack 2175679539 win 64240
<nop,nop,timestamp 1127027 271983807> (frag 10878:1448@0+) (ttl 112, len
1468, bad cksum 52f8!)
08:31:58.366507 213.107.134.109.3160 > 115.74.42.52.80: . [bad tcp cksum
5f35!] 0:1416(1416) ack 1 win 64240 <nop,nop,timestamp 1130650 271983807>
(frag 29935:1448@0+) (ttl 112, len 1468, bad cksum 887!)
09:04:20.206507 213.106.221.197.4461 > 115.74.127.253.80: P [bad tcp cksum
b050!] 1285553056:1285554484(1428) ack 255088721 win 17520 (frag
28991:1448@0+) (ttl 112, len 1468, bad cksum 6014!)
09:07:17.496507 24.48.78.100.3080 > 115.74.249.202.80: P [bad tcp cksum
d2d2!] 3344990140:3344990511(371) ack 577707083 win 17520 (DF) (ttl 113, id
25429, len 411, bad cksum 61ef!)
```

```
09:07:17.506507 24.48.78.100.3081 > 115.74.249.202.80: P [bad tcp cksum
b5ef!] 3345051283:3345051656(373) ack 583640118 win 17520 (DF) (ttl 113, id
25431, len 413, bad cksum 61eb!)
09:07:17.526507 24.48.78.100.3082 > 115.74.249.202.80: P [bad tcp cksum
d2d2!] 3345096153:3345096528(375) ack 584130751 win 17520 (DF) (ttl 113, id
25433, len 415, bad cksum 61e7!)
09:07:17.536507 24.48.78.100.3083 > 115.74.249.202.80: P [bad tcp cksum
b5ef!] 3345143785:3345144162(377) ack 584464802 win 17520 (DF) (ttl 113, id
25435, len 417, bad cksum 61e3!)
```

2.2.8 Severity:

Criticality = 4

This will depend on the web application the systems are using. Assuming its IIS and Cisco 600 series DSL routers, then it is very critical to patch. One point off because the attack seems to a trial run and probing attempt.

Lethality = 5

If this attack were to succeed then the result would be very lethal. Owned hosts and causing DoS on routers would devastate any network.

System countermeasure = 3

According to the logs the systems did not appear to respond to the attack. But more checking is needed to assure no compromise occurred.

Network countermeasure = 3

This packet should not be allowed into the network. Better knowledge of the network architecture and placement of sensors is needed to verify depth of penetration.

severity = (criticality + lethality) – (system + network countermeasures)
severity = (4+5)-(3+3)
severity = 3

This penetration needs to be investigated further due the variance of the attack.

2.2.9 Defensive recommendation:

Detection

eEye Digital Security (http://www.eeye.com/) has recently released a free tool which you can use to scan your network for IIS servers which may still be vulnerable to the "Code Red" (and hence "CodeRedII") worm. You can download this tool from the eEye site directly at: http://www.incidents.org/archives/intrusions/msg02292.html

Prevention

All IIS servers should be patched at the latest level. This can be found at: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

A Snort rule submitted by Jim Forster http://www.incidents.org/archives/intrusions/msg00300.html can be used to catch the malicious packet if the MF and DF flags are not both set and hence not picked up the bad-traffic.rule.

```
alert tcp any any <> any 80 (msg: "CodeRed IDA Overflow"; dsize: >1000; flags:
PA+; content:"|2F646566 61756C74 2E696461 3F4E4E4E|"; depth:16;)
```

This rule will catch on a payload size bigger that 1000 bytes, the PSH, ACK and/or other flags set, the string default.ida together with the first 3 repeating N's in HEX. The last three repeating "4E" bytes could be removed and also the string default.ida could be added.

Cisco released this advisory, as some of their devices were susceptible to this worm, either by compromise or by side effect such as DoS.
http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml

2.2.10 Multiple choice test question:

If a new worm called CodeRed NEW is discovered to use the same propagation and trojan characteristics of CodeRed and new injection vector is created to infect hosts with the same result of CodeRed ie root compromise, what would be the likely result.

A) Same problems would occur as with CodeRed
B) No problems would occur as the patches released have fixed the propagation and trojan characteristics.
C) *NIX flavours would also be affected.
D) None of the above

Answer: A
Reasons :
The patch fixes the injection vector but does not stop propagation and trojans from executing, once root compromise occurs.

3 Analyse This

3.1 Executive Summary

The University of Maryland Baltimore County has generated a large amount of differing types of logging information from the Snort Intrusion Detection System. Five days worth of log data will be analysed holistically. Because of the shear volume of data, the Events of Interest (EOIs) will be grouped according to severity. As immediate analysis is impossible, the groups will be determined by the Alert rule documentation, the analyst's knowledge of relevant EOIs and similar traffic patterns.

Each Alert group will be analysed in turn from highest to lowest severity. The Alert analysis will be cross-referenced with the two other log file sets ie Scan files and Out Of Specification files (OOS). This will be done where relevant to show patterns of behaviour, support conclusions and justifications arising from the analysis data.

Tables of Top 10 Talkers are provided to assist in the holistic view. The analysis will refer to these tables where relevant to provide the reader with a summary of traffic patterns and a reference point for further analysis. Further to this, an Analysis Summary and Severity rating will be provided where necessary to further assist the reader with a holistic view and a course of action.

3.2 Files Analysed

http://www.incidents.org/logs

Alert Files	Size	Scan Files	Size	Out-Of-Spec Files	Size
alert.030608.gz	1,629,461	scans.030608.gz	4,798,663	OOS_Report_2003_06_08_22596	1,587,203
alert.030610.gz	714,670	scans.030610.gz	2,565,532	OOS_Report_2003_06_10_6145	21
alert.030611.gz	2,320,998	scans.030611.gz	6,252,031	OOS_Report_2003_06_11_13995	471,043
alert.030612.gz	2,667,783	scans.030612.gz	9,265,581	OOS_Report_2003_06_12_2042	1,448,963
alert.030613.gz	2,606,795	scans.030613.gz	10,141,045	OOS_Report_2003_06_13_16083	1,351,683
alert.all	97,743,677	scans.all	288,176,469	OOS_Report_all	4,893,491

3.3 Top Talkers

Due to scan logs being available I have removed the spp_portscan entries in the alert files that are generated by the Snort portscan preprocessor.

There are so many Events Of Interest (EOIs) that it is difficult to choose which alert to examine first. Because of this reason, I have evaluated and grouped the EOIs (by eye and best guess) to determine the most important analysis first. As I am unfamiliar with the some of the generated messages in the alert file, I will read a literal interpretation of the message. This is where well-documented and well written rules would make the job of an analyst more efficient. (To generate the below EOIs and table, the perl script csv.pl and summarise.pl are used from Tod Beardsley's practical http://www.giac.org/practical/Tod_Beardsley_GCIA.doc)

EOIs by Alert Message Grouped by Priority (highest to lowest)

Possible Trojan activity

```
38613 spp_http_decode: IIS Unicode attack detected
10753 spp_http_decode: CGI Null Byte attack detected
6494  High port 65535 udp - possible Red Worm - traffic
2855  High port 65535 tcp - possible Red Worm - traffic
4920  Possible trojan server activity
1181  IDS552/web-iis_IIS ISAPI Overflow ida nosize
587   IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
239   NIMDA - Attempt to execute cmd from campus host
2     NIMDA - Attempt to execute root from campus host
2     Bugbear@MM virus in SMTP
1     Back Orifice
```

Standard rules with EXPLOIT swapped for SHELLCODE

```
13747 EXPLOIT x86 NOOP
48    EXPLOIT x86 setgid 0
46    EXPLOIT x86 setuid 0
8     EXPLOIT x86 stealth noop
5     EXPLOIT NTPDX buffer overflow ** non-standard
1     EXPLOIT FTP passwd retrieval retr path ** non-standard
```

RPC activity

```
425   [UMBC NIDS IRC Alert] Possible Incoming XDCC Send Reque
32    RFB - Possible WinVNC - 010708-1
26    IRC evil - running XDCC
15    [UMBC NIDS IRC Alert] Possible sdbot floodnet detected
3     [UMBC NIDS IRC Alert] Possible drone command detected.
2     [UMBC NIDS IRC Alert] User joining Warez channel detect
2     [UMBC NIDS IRC Alert] User joining XDCC channel detecte
```

Interesting network patterns of traffic

```
3341  TCP SRC and DST outside network
1978  connect to 515 from outside
241   SNMP public access
20    ICMP SRC and DST outside network
3     TFTP - External UDP connection to internal tftp server
```

Non-descript rules

```
104286      CS WEBSERVER - external web traffic
5099  CS WEBSERVER - external ftp traffic
66593 MY.NET.30.4 activity
10632 MY.NET.30.3 activity
86919 SMB Name Wildcard
277   SMB C access
52    Notify Brian B. 3.56 tcp
48    Notify Brian B. 3.54 tcp
```

Internet Noise

```
7486  Queso fingerprint
5     Probable NMAP fingerprint attempt
1319  Null scan!
266   scan (Externally-based)
493   NMAP TCP ping!
18    NETBIOS NT NULL session
1201  Tiny Fragments - Possible Hostile Activity
2     Fragmentation Overflow Attack
905   Incomplete Packet Fragments Discarded
94    SUNRPC highport access!
93    TFTP - Internal UDP connection to external tftp server
26    TFTP - Internal TCP connection to external tftp server
42    FTP passwd attempt
148   connect to 515 from inside
2     Attempted Sun RPC high port access
955   External RPC call
```

Total Unique Alerts: 54 Total EOIs: 337544

3.4 Top Ten Talkers (various types)

I have summarised the top talkers with comments to get a holistic view of traffic patterns to see if anything "stands out". The notes will be investigated throughout the analysis.

Table 1

EOIs by Src IP Ext.	Number
68.170.69.138	50122
66.207.164.23	35313
128.32.79.218	12009
68.49.35.0	8037
207.151.67.140	7833
216.39.48.2	4427
68.48.110.245	3526
150.214.191.55	3516
211.217.184.210	2718
192.168.2.21 *RFC1918	2330

Table 2

EOIs by Src Port Ext.	Number
1845	49486
6667 *IRC	37665
1025	11317
1026	10277
1027	10102
1028	7302
1029	6082
137	5489
65535 *IRC	5360
1061	3692

Table 3

EOIs by Dst IP Ext.	Number
203.161.233.132	6594
64.235.110.34	2482
211.217.184.210	1907
208.194.163.37	1723
211.76.139.245	1587
216.231.171.27	1479
209.116.81.5	1251
211.239.164.248	1168
61.135.132.210	849
192.151.53.10	833

Table 4

EOIs by Dst Port Ext.	Number
80	36639
6667 *IRC	4956
65535 * IRC	3819
27374	2043
None *port 0	829
443	504
8080	156
515 *policy decision	148
6346	142
25	68

Table 5

EOIs by Dst IP Int.	**Number**
MY.NET.100.165	109518
MY.NET.30.4	66582
MY.NET.190.95	35336
MY.NET.30.3	10632
MY.NET.70.164	2904
MY.NET.114.116	2354
MY.NET.88.223	2237
MY.NET.70.207	2014
MY.NET.86.19	1486
MY.NET.5.55	1455

Table 6

EOIs by Dst Port Int.	**Number**
80	143747
137	86852
51443 *too high	55561
524	9303
21	5165
25	3929
None *port 0	2977
4662 *eDonkey file share	2925
12203	2007
515	1978

Table 7

EOIs by Src IP Int.	**Number**
MY.NET.83.100	4765
MY.NET.97.104	3861
MY.NET.70.164	2021
MY.NET.84.216	1615
MY.NET.70.207	1479
MY.NET.163.76	1251
MY.NET.97.248	1190
MY.NET.97.239	974
MY.NET.97.184	884
MY.NET.75.107	856

Table 8

EOIs by Src Port Int.	**Number**
4662 *eDonkey file share	1910
12203	1479
1249	1329
6257	865
2414	643
None *port 0	387
1587	279
1240	222
1591	192
1483	175

Table 9

EOIs by Relationship Ext.->Int.	
68.170.69.138->MY.NET.30.4	50120
66.207.164.23->MY.NET.190.95	35308
68.49.35.0->MY.NET.30.3	7853
216.39.48.2->MY.NET.100.165	4422
68.48.110.245->MY.NET.30.4	3526
150.214.191.55->MY.NET.100.165	3516
211.217.184.210->MY.NET.70.164	2718
216.231.171.27->MY.NET.70.207	2007
212.106.150.180->MY.NET.88.223	1858
211.104.1.5->MY.NET.100.165	1466

Table 10

EOIs by Relationship Int.->Ext.	
MY.NET.97.104->203.161.233.132	3860
MY.NET.83.100->64.235.110.34	2482
MY.NET.70.164->211.217.184.210	1907
MY.NET.83.100->208.194.163.37	1723
MY.NET.70.207->216.231.171.27	1479
MY.NET.163.76->209.116.81.5	1251
MY.NET.97.248->203.161.233.132	1102
MY.NET.97.239->211.76.139.245	956
MY.NET.97.184->203.161.233.132	884
MY.NET.168.70->61.135.132.210	817

Table 11

EOIs by Relationship Ext.->Ext.	Number
192.5.3.11->66.187.232.101	497
192.168.2.21->24.62.119.33	185
192.168.2.21->212.160.129.140	149
192.168.2.21->68.72.247.214	144
192.168.2.21->195.241.128.137	108
192.168.2.21->24.136.140.102	101
192.168.2.21->80.140.207.177	84
192.168.2.21->66.26.20.108	77
192.168.2.21->80.202.160.134	76
192.168.2.21->80.38.32.13	76

RFC 1918 addresses?

Table 12

OOS Top Talkers		Number
212.106.150.180	MY.NET.88.223	3571
66.117.30.14	MY.NET.233.78	875
66.117.30.14	MY.NET.224.134	860
12.255.198.216	MY.NET.24.44	140
63.100.123.132	MY.NET.24.34	121
80.222.139.136	MY.NET.70.164	112
217.127.44.126	MY.NET.70.225	110
62.212.98.220	MY.NET.84.144	102
209.116.70.75	MY.NET.139.230	99
217.228.191.153	MY.NET.70.225	72

Last address 1st 2 octets matches table 13 top talker

Table 13

Scan-Pair Top Talkers		Number
217.228.154.39	130.85.88.223	481
138.88.107.22	130.85.218.134	360
141.157.19.82	130.85.12.2	136
141.157.19.82	130.85.25.11	123
38.114.128.32	130.85.97.203	80
130.85.217.58	68.54.166.69	60
130.85.217.58	217.84.47.165	59
63.209.10.70	130.85.97.165	55
63.251.52.75	130.85.97.193	44
81.73.221.96	130.85.70.164	43

Top talker 1st two octets match table 12 last talker

3.4.1 Possible Trojan Alerts

```
38613 spp_http_decode: IIS Unicode attack detected
10753 spp_http_decode: CGI Null Byte attack detected
6494  High port 65535 udp - possible Red Worm - traffic
2855  High port 65535 tcp - possible Red Worm - traffic
4920  Possible trojan server activity
1181  IDS552/web-iis_IIS ISAPI Overflow ida nosize
587   IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
239   NIMDA - Attempt to execute cmd from campus host
2     NIMDA - Attempt to execute root from campus host
2     Bugbear@MM virus in SMTP
1     Back Orifice
```

spp_http_decode & IDS552/web-iis

These types of alerts are the precursors to any attempted Trojan insertion. They are performed to see if the host is vulnerable to the specific Trojan attack vector. The IDS552/web-iis traffic is directed to port 80 with no return traffic so it is rated as noise. The spp_http_decode alerts are triggered by the Snort http_decode pre-processor on the unicode-encoded characters ‘.’, ‘\’ or ‘/’ in the http data payload. Getting the top talkers for this alert

```
# grep "spp_http_decode:" alert.all |cut -d" " -f10 | grep -v "MY.NET"|cut -d"." -f1-3|sort|uniq -c | sort -rn
  11980 128.32.79
    297 62.158.82
```

(I only used the first 3 octets to get at least a class C network)

The above is a probe is from the University of California and this traffic and as it is a probe only, it can be moved to the Internet noise category and hence prioritised much lower.

From the table above, the top internal src hosts that caused the above NIMDA alert:

```
# # grep "NIMDA - Attempt to execute " n3 | cut -f 3 -d"]" | cut -f 1 -d
":" | sort | uniq -c | sort -nr
```

142 MY.NET.97.37	76 MY.NET.97.101	18 MY.NET.97.228
1 MY.NET.53.196	1 MY.NET.178.218	1 MY.NET.151.98
1 MY.NET.132.42	1 MY.NET.111.34	

Correlating the internal MY.NET addresses with the scans.all file, we'll look at the highest and lowest

```
# grep "130.85.97.228" scans.all | cut -f 2 -d">" | sort | uniq | wc
  5725   17175  186905
#  grep "130.85.111.34" scans.all | cut -f 2 -d">" | sort | uniq | wc
 14957   31353  397395
```

The number of scans show that these 8 hosts are most likely infected with Nimda.

Severity

severity = (criticality + lethality) – (system + network countermeasures)
severity = (5 + 5) – (1 + 2) = 7
Hosts were infected and the network perimeter did not prevent an external scan.

Correlations

Brian Cahoon has investigated Nimda and its associated alerting characteristics.
http://www.giac.org/practical/GCIA/Brian_Cahoon_GCIA.pdf

Tod Beardsley has done an analysis on these types of internal threats.
http://www.giac.org/practical/Tod_Beardsley_GCIA.doc

Also Snort FAQ is a source of information.
http://www.snort.org/docs/faq.html#4.17

HTML in emails as an injection vector for this type of activity
CAN-2001-0154 (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154)

http://www.giac.org/practical/GCIA/Rick_Larabee_GCIA.doc

spp_http_decode: CGI Null Byte attack detected
spp_http_decode: IIS Unicode attack detected

Defensive Recommendation

Patch the eight infected hosts that were listed above with the patches provided on links on the following web page.
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/virus/Nimda.asp.

Other than patching hosts, preventing external scanning by internal devices may be a way to limit infection rates. An IDS with flexible response can be employed for this purpose.

3.4.2 Analysis of High port 65535 udp & tcp - possible Red Worm - traffic

This is a custom Snort rule. Port 65535 is indicative of Red Worm (aka Adore Worm) activity.

> *William Stearns from Dartmouth's ISTS (http://www.sans.org/y2k/adore.htm).*
> *Adore scans the Internet checking Linux hosts to determine whether they are vulnerable to any of the following well-known exploits: LPRng, rpc-statd, wu-ftpd and BIND.*

There was no evidence of this Worm as the alert and scans files show no sign of the above characteristics.

This alert is yielding unusual results. There are no UDP alerts but 242 lines of UDP scans and 15 lines of TCP scans. All of the scans came from internal addresses except for one (this one scan is Internet noise; hence ignored)

```
# grep ":65535" scans.all | grep -v UDP | wc          Non-UDP traffic
     15     120    1071
# grep ":65535" scans.all | grep UDP | wc             UDP traffic
    242    1694   15897
```

The top talkers with port 65535 show two way communication.

```
grep ":65535" alert.n3 | cut -d \] -f 3 | sort | uniq -c | sort -rn
   2007  216.231.171.27:65535 -> MY.NET.70.207:12203
   1479  MY.NET.70.207:12203 -> 216.231.171.27:65535
   1066  172.191.157.8:65535 -> MY.NET.251.10:5121
    627  MY.NET.97.75:2414 -> 219.116.125.80:65535
    585  219.116.125.80:65535 -> MY.NET.97.75:2414
    138  MY.NET.153.223:6257 -> 217.224.209.200:65535
    118  217.224.209.200:65535 -> MY.NET.153.223:6257
```

Looking at the top talkers, there appears to be multiple internal and external addresses communicating. The ports used and traffic patterns do not indicate the presence of the Worm, instead we see similar traffic to file sharing communication.

What I believe is occurring is some sort of file sharing is obscuring the rationale of these alerts being associated with Red Worm. When the alerts are correlated with the scans it can be seen that internal ports such as 6257 are initiating external connections to 65535 using P2P applications such as WinMX. From the alert file, two way communications are occurring between these hosts using other file sharing programs that have dynamic port configuration.

Other Correlations

http://www.giac.org/practical/Paul_Young_GCIA.pdf
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://www.giac.org/practical/Al_Williams_GCIA.pdf
Alarms were genereated by one internal host involved in Peer to Peer file sharing.

Severity

This rule did not catch Red Worm traffic and it is unknown if P2P file sharing is allowed. A severity rating of 0 would be appropriate and the Security Manager will determine if/when action is required.

Defensive Recommendation

Reconfigure the Snort rule to match the port injection vectors (ie service ports) for Red Worm and only match on IP of devices that are susceptible ie devices running the Linux operating system.

3.4.3 Possible trojan server activity

The custom rule alerts on port activity to 27374. False positives would occur as this port will be used as an ephemeral port by legimate traffic. Scott Shinberg pointed this out in his GCIA practical.

http://www.giac.org/practical/Scott_Shinberg_GCIA.doc
Legitimate traffic would be characterized at that which uses one high port like 27374 and one low port, such as 25 for email. Traffic that uses two high 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

Using this as a basis for searching the alert files, we see below that the top two talkers exhibit 2-way communication and both use high ports. This is also evident in the communication for 24.80.217.0 with MY.NET.88.223 on lines c) and e). Legitimate traffic si shown on lines d) and f).

```
# grep "Possible trojan server activity" alert.all | cut -f 3 -d"]" | sort
| uniq -c | sort -rn
a)   2718  211.217.184.210:27374 -> MY.NET.70.164:4662
b)   1907  MY.NET.70.164:4662 -> 211.217.184.210:27374
c)     24  24.80.217.0:27374 -> MY.NET.88.223:4524
d)     14  MY.NET.12.4:110 -> 68.32.63.62:27374
e)     12  MY.NET.88.223:4524 -> 24.80.217.0:27374
f)     10  MY.NET.12.4:110 -> 68.50.119.57:27374
```

It appears that this is subseven activity but port 4662 is used by eDonkey (file sharing program) so this appears to be a file sharing conversation (if they want to avoid detection they should use a port not know for trojans!)

Severity

This rule did not catch Trojan traffic and it is unknown if P2P file sharing is allowed. A severity rating of 0 would be appropriate and the Security Manager will determine if/when action is required.

Bugbear@MM virus in SMTP & Back Oriface

The bugbear alert was generated because of a custom catchall rule to port 25 that was not picked up by one of the other rules.

```
# grep "Bugbear@MM virus in SMTP" alert.all
06/10-15:00:19.833788  [**] Bugbear@MM virus in SMTP [**] 212.60.67.2:33161
-> MY.NET.6.47:25
06/13-08:48:52.867417  [**] Bugbear@MM virus in SMTP [**]
194.158.96.112:62364 -> MY.NET.24.22:25
```

The one detect of **BackOrifice** in the alert files is a most likely a simple rule to detect on UDP port 31337 and not the Snort preprocessor bo. I'm guessing at this as the the preprocessor would have picked up a BackOrifice packet before the rules. Also, the MY.NET.151.115 host machine does not log any suspicious alerts or perfoms any scanning, I doubt this machine is owned.
For peace of mind, follow the instructions at http://www.nwinternet.com/~pchelp/bo/findingBO.htm and check the host.

Analysis Summary

Nimda is the only likely trojan in the alert files, the rest of the "reported" trojan alerts are file sharing activities triggering the overly sensitive Snort rules. The University should consider tuning it's ruleset by better message naming, grouping alerts to relevant hosts and relocating sensors to get more meaningful information.

3.4.4 Standard rules with EXPLOIT

```
13747 EXPLOIT x86 NOOP
48    EXPLOIT x86 setgid 0
46    EXPLOIT x86 setuid 0
8     EXPLOIT x86 stealth noop
5     EXPLOIT NTPDX buffer overflow ** non-standard
1     EXPLOIT FTP passwd retrieval retr path ** non-standard
```

The EXPLOIT alert appears to be a standard rule with a slight name change. The Snort rules and SIDs for EXPLOIT are akin to SHELLCODE:

EXPLOIT x86 NOOP http://www.snort.org/snort-db/sid.html?sid=648

```
/etc/snort/shellcode.rules:alert ip $EXTERNAL_NET any -> $HOME_NET
$SHELLCODE_PORTS (msg:"SHELLCODE x86 NOOP"; content: "|90 90 90 90 90 90 90
90 90 90 90 90 90 90|"; depth: 128; reference:arachnids,181;
classtype:shellcode-detect; sid:648; rev:5;)
```

EXPLOIT x86 NOOP http://www.snort.org/snort-db/sid.html?sid=1394

```
/etc/snort/shellcode.rules:alert ip $EXTERNAL_NET any -> $HOME_NET
$SHELLCODE_PORTS (msg:"SHELLCODE x86 NOOP"; content:"|61 61 61 61 61 61 61
61 61 61 61 61 61 61 61 61 61 61 61 61|"; classtype:shellcode-detect;
sid:1394; rev:3;)
```

EXPLOIT x86 setuid 0 http://www.snort.org/snort-db/sid.html?id=650

```
/etc/snort/shellcode.rules:alert ip $EXTERNAL_NET any -> $HOME_NET
$SHELLCODE_PORTS (msg:"SHELLCODE x86 setuid 0"; content: "|b017 cd80|";
reference:arachnids,436; classtype:system-call-detect; sid:650; rev:5;)
```

EXPLOIT x86 setgid 0 http://www.snort.org/snort-db/sid.html?sid=649

```
/etc/snort/shellcode.rules:alert ip $EXTERNAL_NET any -> $HOME_NET
$SHELLCODE_PORTS (msg:"SHELLCODE x86 setgid 0"; content: "|b0b5 cd80|";
reference:arachnids,284; classtype:system-call-detect; sid:649; rev:5;)
```

EXPLOIT ntpdx http://www.snort.org/snort-db/sid.html?sid=312

```
alert udp $EXTERNAL_NET any -> $HOME_NET 123 (msg:"EXPLOIT ntpdx
overflow attempt"; dsize: >128; reference:arachnids,492; reference:bugtraq,2540;
classtype:attempted-admin; sid:312; rev:2;)
```

EXPLOIT FTP passwd http://www.snort.org/snort-db/sid.html?sid=356

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP passwd retrieval
attempt"; flow:to_server,established; content:"RETR"; nocase; content:"passwd";
reference:arachnids,213; classtype:suspicious-filename-detect; sid:356; rev:5;)
```

The general consensus of the x86 Snort sids [above] are "Fairly high.[False Positives] Large binary transfers, certain web traffic, and even mail traffic can trigger this rule.". Also http://lists.jammed.com/incidents/2001/10/0023.html, stated that binary downloads/uploads over HTTP (including web enabled email) will cause this rule to fire.

The IP Source 207.151.67.148 (5th entry in Top Talker Table 1) had 7833 alerts all for various MY.NET addresses with a majority for destination port 80. Different source ports were used with each conversation, which looks like data being sent to a web server. There is traffic sent to port 119 (Network News Transfer Protocol) and port 139 (NETBIOS Session Service). Sending binary data to news groups is not unusual but I do question trying to send this type of traffic to port 139.

The NTPDX and FTP EXPLOIT alerts have the same pattern ie attempted compromise and no result.

For this attack to have succeeded there is a high probability that the compromised host is performing some sort of internal scans. No evidence of the EXPLOIT targets were found in the scans files.
The majority of this attack was directed to port 80 but it is noted that a range of reserved and ephemeral ports was used. This is probably due to the sensor placed outside the perimeter.

Severity
severity = (criticality + lethality) – (system + network countermeasures)
severity = (4 + 5) – (4 + 3) = 2
The network countermeasure is difficult to determine due to sensor placement outside perimeter but no return traffic was detected so average rating is given.

Defensive Recommendation
No compromise is evident but the entire packet is currently not available for inspection. Using the SID urls in the above table and following the Corrective Action will be used when complete packet payloads becomes available.
The FTP EXPLOIT packet dump (also if available) should be analysed for the string "passwd".
Permit only required port traffic to associated host IP addresses will reduce and moving the sensor inside the perimeter will reduce number of alerts.

3.4.5 Possible RPC Activity

```
425   [UMBC NIDS IRC Alert] Possible Incoming XDCC Send Reque
15    [UMBC NIDS IRC Alert] Possible sdbot floodnet detected
3     [UMBC NIDS IRC Alert] Possible drone command detected.
2     [UMBC NIDS IRC Alert] User joining Warez channel detect
2     [UMBC NIDS IRC Alert] User joining XDCC channel detecte
32    RFB - Possible WinVNC - 010708-1
26    IRC evil - running XDCC
```

After discovering the 130.85.0.0/16 range is assigned to the University of Maryland Baltimore County, it made it less difficult to determine that any rule with UMBC was most likely a custom rule☺

The UMBC alerts have together generated 42684 entries. All of these alerts will have one of the below ports in common

```
Information is from /usr/share/nmap/nmap-services
irc         6667/tcp #Internet Relay Chat
vnc-http    5900/tcp #Virtual Network Computer HTTP Access,display 0
afs3-fileserver   7000/tcp #file server itself, msdos
afs3-fileserver   7000/udp #file server itself
```

The command below show the top 3 entries fo address that have generated UMBC IRC traffic.

```
# grep "UMBC NIDS IRC Alert" alerts.all | cut -f 4 -d"]" | cut -d":" -f1 |
sort | uniq -c | sort -rn
      35313  66.207.164.23
      4765  MY.NET.83.100
      332  195.159.0.82
```

The top address is the second top talker in Table 1 and the port 6667 appears frequently in the top talkers tables.

Defensive Recommendation

Difficult to stop at Universities but restrictions on bandwidth for each user would limit some usage. Policy and education are the best action in the long term with view to putting in an application firewall and blocking IRC and file sharing.

Analysis Summary

IRC accounts for a large amount of alerts. Unfortunately IRC is an avenue for backdoors, trojans and viruses so the University should define a policy for this type of traffic. If it's allowed, then bandwidth and time restrictions may be applied to individual users. This will involve additional work for the University. Restricting known IRC ports at the perimeter, policing this activity and enforcing deterrents such as account suspension may need to be applied. Alternatively the University can enforce an explicit deny all except for allowed ports. This will not remove the problem altogether but it will significantly reduce the number of alerts and also the University's IRC user population.

3.4.6 Non-descript rules

```
3341  TCP SRC and DST outside network
1978  connect to 515 from outside
241   SNMP public access
20    ICMP SRC and DST outside network
3     TFTP - External UDP connection to internal tftp server
```

TCP & ICMP SRC and DST outside network

This section directly relates to table 11 EOIs by Relationship Ext. -> Ext.
There are too many addresses to spoof and the snort sensor is picking up a lot of various traffic types to be incorrectly configured. A private address, 192.168.2.21, is being accessed by AT&T Broadband Northeast on the eDonlkey file sharing port 4662

The Red Hat address 66.187.232.101 has multiple connections to various sites with using the Gnutella file sharing protocol 6346. This appears as legitimate traffic so this could be and indication that the sensor is placed in another organisation. If this is the case then the source/destination outside the network is not much of an issue. In addition, the University could also be acting as a web relay for some organisations as evidenced by the traffic to/from ports 80 & 443.

The private address 192.168.2.21 is also the last entry in table 1 and accounts for a significant amount of eDonkey file sharing traffic on port 4662 but this is in evidence throughout this analysis. The fact that it is a private address could be an indication that this sensor is behind a proxy firewall and from the paragraph above there is a high probability of such architecture.

Paul Young provided a basis for the above analysis. His analysis was based on UDP but the concept of what type of situations could be give rise to outside source and destination addressing. http://www.giac.org/practical/Paul_Young_GCIA.pdf (page37)

Connect to 515 from outside

Port 515 is usually used by the line printer daemon protocol. Allowing connectivity from an external address is a policy matter for the University to either allow or disallow. There are 2 print servers MY.NET.162.104:515 & MY.NET.24.15:515. The University of Maryland, device 68.155.6.153:32000-33000 range, only accesses the first address. This looks like a UNIX device so nothing unusual here. The same University once again accesses the second address but this time from ports 721 and 722. According to RFC 1179 (Line Printer Daemon Protocol) http://www.lprng.com/LPRng-HOWTO-Multipart/rfc1179ref.htm this is normal behaviour. A Comcast Cable Communications address (68.54.94.58) is also trying access the printer service. This should be investigated.

SNMP public access

This standard Snort rule has probably been modified to trigger on either TCP or UDP with content of "public" (usually it is split into TCP and UDP)
The relevant Snort SIDs are:
UDP http://www.snort.org/snort-db/sid.html?sid=1411
TCP http://www.snort.org/snort-db/sid.html?sid=1412
Also there are CVEs for this alert
cve,CAN-2002-0013 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0013
cve,CAN-2002-0012 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0012
cve,CAN-1999-0517 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0517

As the request is incoming, this traffic is not that important. The relevant MY.NET addresses should be investigated and the System Administrators should ensure that they the passwords are in accordance to the University's password policy.

```
# grep "SNMP public access" n4 | cut -d"]" -f 3 | sort | uniq
 131.118.250.143:1031 -> MY.NET.154.26:161
 131.118.250.143:1031 -> MY.NET.154.31:161
 134.192.86.65:1049 -> MY.NET.190.13:161
 134.192.86.65:1053 -> MY.NET.190.13:161
 134.192.86.65:1058 -> MY.NET.190.13:161
 134.192.86.65:1063 -> MY.NET.190.13:161
 147.46.56.20:1025 -> MY.NET.154.26:161
```

TFTP - External UDP connection to internal tftp server

The alert, scan or OOS logs did not show any response traffic from the three addresses MY.NET.198.247, MY.NET.151.115, MY.NET.198.247 that were involved with this alert. This can be classified as regular Internet noise.

3.4.7 Non-Descript Rules

```
104286      CS WEBSERVER - external web traffic
5099  CS WEBSERVER - external ftp traffic
66593 MY.NET.30.4 activity
10632 MY.NET.30.3 activity
86919 SMB Name Wildcard
277   SMB C access
52    Notify Brian B. 3.56 tcp
48    Notify Brian B. 3.54 tcp
```

CS WEBSERVER - external web & ftp traffic

These events are listed here because the combined web & ftp alerts are the most logged (109,377). This is not a standard snort rule and should be revised to eliminate background noise. This rule logged either any traffic on ports 80 and 21 on MY.NET.100.165 or logged any traffic on IP address MY.NET.100.165. It can not be determined due to other rules superseding this rule by logging the alert the dropping out from the rule list.

According to the logs, someone wanted to find out how much traffic and what source traffic was hitting the address MY.NET.100.165. This information may be used as input to other applications or perhaps some sort of billing data as this host appears to be a web server.

Severity

severity = (criticality + lethality) – (system + network countermeasures)
severity = (1 + 1) – (5 + 5) = -8

Correlations

As this is a unique rule put in to capture traffic as listed in the description above there are no correlations.

Defensive Recommendation

Remove the rule from list or refine it to log only and not alert.

SMB Name Wildcard & SMB C access

From the rule descriptions and traffic in the alert files, these rules match on activity on ports 137 and 139 respectively. The rule that alerts on activity to port 137 (NETBIOS Name Service) and the log shows entry for MY.NET returning 137 traffic to external or internal hosts. 208 different internal class C subnets were targeted by external addresses. The scan files indicated that internal Windows hosts did return traffic (identified as Windows hosts as the source port was also UDP 137). SMB C access showed evidence directed communication with external addresses on ports 139 and 445 with no alerts generated. More information is needed to for analysis.

The sensor that gathered this information was likely placed outside the perimeter, as no logs were found for internal to internal NETBIOS Name Service activity.
If the sensor is outside the perimeter either data is leaking out via NETBIOS or more probably the addresses are outside the perimeter. The command below captures the number internal to external traffic (src & dst ports are both137 UDP) communication pairs in the scan files.

```
# grep -v "\-> 130.85." scans.all | grep -E '.*:137 \-' | cut -d " " -f 4-6
| sort | uniq | wc
```

```
   3154    9462  123266
AND
# grep -v "\-> 130.85." scans.all | grep -E '.*:139 \-' | cut -d " " -f 4-6
| sort | uniq | wc
      0       0       0
```

University hosts outside perimeter along with number of communications per IP:

2365 130.85.97.222:137	2009 130.85.97.13:137	1025 130.85.97.180:137
72 130.85.97.18:137	32 130.85.132.24:137	22 130.85.97.238:137
14 130.85.97.91:137	11 130.85.97.28:137	3 130.85.97.148:137
1 130.85.97.42:137	1 130.85.97.210:137	

Severity

severity = (criticality + lethality) – (system + network countermeasures)
severity = (3 + 3) – (4 + 4) = -2

Correlations

(http://www.giac.org/practical/Tod_Beardsley_GCIA.doc).
Again, except for KaZaA, as Tod pointed out, these packets are used for reconnaissance. This is really just background noise on the network.

http://www.giac.org/practical/Al_Maslowski-Yerges_GCIA.pdf
Windows and Samba clients typically use this type of NetBIOS traffic to find hosts even if they are involved in other communications with the host if they can't resolve the name with DNS

Bryce Alexander
http://www.sans.org/y2k/051300.htm
An interesting side effect of this worm has been a rather strange pattern that periodically shows up in the scans for port 137… It is my speculation that this is caused by systems that are providing proxy services on cable modems in order to share a single IP address on a cable modem. The internal (private) address is leaking out onto the network, most likely due to sharing a single ethernet hub for both internal and external interfaces.

Defensive Recommendation

Remove or refine rule to more specific hosts or network ranges. Do not log this at the perimeter as it creates a lot of noise. Unless there is a specific business requirement, the perimeter should not allow NETBIOS (ports 137-139) and Microsoft –ds (port 445) into or out of the network. If required, specific ports and IP addresses should be allowed but not network ranges!

MY.NET.30.3 & MY.NET.30.4 activity & Notify Brian B. 3.56 & 54

As the names suggest, these rules alert on any activity to these addresses. These 4 devices are web serves using various ports. Port 51443 is used as an alternative to port 443 for Novell's iFolder (file sharing program) http://nscsysop.hypermart.net/ifolder.html. This traffic is one of the top talkers in table 6. Also Novells NCP port 524 is used.

Port 8009 is used for Apache Tomcat. It is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies.
Given this information we can will that these 4 boxes are web servers and the rule was put in to get an idea of traffic patterns.

3.4.8 Internet Noise

```
7486  Queso fingerprint
5     Probable NMAP fingerprint attempt
1319  Null scan!
266   scan (Externally-based)
493   NMAP TCP ping!
18    NETBIOS NT NULL session
1201  Tiny Fragments - Possible Hostile Activity
2     Fragmentation Overflow Attack
905   Incomplete Packet Fragments Discarded
94    SUNRPC highport access!
93    TFTP - Internal UDP connection to external tftp server
26    TFTP - Internal TCP connection to external tftp server
42    FTP passwd attempt
148   connect to 515 from inside
2     Attempted Sun RPC high port access
955   External RPC call
```

Queso fingerprint

Queso sends data to a host and then determines what the host is by the returned fingerprint. There is no evidence in the logs that MY.NET returned any traffic.

```
# grep Queso alert.all | grep -v "\-> MY.NET" | wc
      0       0       0
```

The sensor that captured this information would have been outside the perimeter

Correlations

http://www.giac.org/practical/GCIA/Rick_Larabee_GCIA.doc
All packets that have the 21 ECN Flags set, the Syn flag set, and TOS of 0x00 are registered as Queso scans in the alert files as well.
http://archives.neohapsis.com/archives/snort/2001-01/0200.html

Defensive Recommendation

Drop (silently) all packets that match the above specifications at the perimeter.

Connect to 515 from inside

```
# grep "connect to 515 from inside" alert.all | cut -f 3 -d"]" | sort | uniq
 MY.NET.162.41:721 -> 128.183.110.242:515
# grep 515 /etc/services
printer        515/tcp        spooler        # line printer spooler
printer        515/udp        spooler        # line printer spooler
```

Printing is being directed to an outside address by a proprietary process. Depending on where the network sensor was placed, this address may either be across a private WAN or over the Internet. As the rule appears to have been written to gather information if is likely that no encryption was used and this traffic traversed the Internet.

Severity
severity = (criticality + lethality) – (system + network countermeasures)
severity = (3 + 3) – (3 + 3) = 0
(Severity will depend on the information classification of the data. Even so, it is not difficult to encrypt traffic in some sort of tunnel!!)

Defensive Recommendation
Review the need to print directly and use secure copy (scp) or secure ftp (sftp) to transfer the data or provide an encrypted tunnel for the traffic.

All of the above traffic is rated as Internet noise and is logged for reference. Constant logging of all information will very quickly fill up storage space and the time to review such data will take time away from more important analysis work. It is better to prioritise and miss smaller matters even though it may mean the occasional one gets through rather than get bogged down in trivial matters and miss important event more often.

3.5 Registry Information

The addresses that are important are taken from the Top Talkers Tables

The IP address that contacted MY.NET the most is 68.170.69.138. This was chosen as this address statistically presents the greatest probability of attack.

http://ws.arin.net/cgi-bin/whois.pl

```
Adelphia Cable Communications ADELPHIA-CABLE-4 (NET-68-168-0-0-1)
                              68.168.0.0 - 68.171.255.255
Adelphia 68170640-Z5 (NET-68-170-64-0-1)
                              68.170.64.0 - 68.170.95.255
```

http://www.adelphia.net is a cable service provider.

The IP address that MY.NET visited the most is 203.161.233.132. This gives an understanding of user behaviour.

```
inetnum:      203.161.224.0 - 203.161.255.255
netname:      ILINK
descr:        iLink.net Limited
descr:        Facility Management, Hong Kong
country:      HK
admin-c:      OO4-AP
tech-c:       OO4-AP
mnt-by:       APNIC-HM
mnt-lower:    MAINT-HK-ILINK
changed:      hostmaster@apnic.net 20000112
status:       ALLOCATED PORTABLE
source:       APNIC
person:       operator operator
address:      56/F The Center,
address:      99 Queen's Road Central,
address:      Hongkong
country:      HK
phone:        +852-31231588
fax-no:       +852-22182288
e-mail:       ipadmin@ilink.net
nic-hdl:      OO4-AP
mnt-by:       MAINT-HK-ILINK
changed:      ipadmin@ilink.net 19991230
source:       APNIC
```

http://www.ilink.net is an infrastructure-based ASP

This address was connecting to the printer service (port 515) inside the MY.NET network. It was chosen to determine what entity is allowed inside the perimeter.

```
OrgName:    University of Maryland
OrgID:      UNIVER-270
Address:    System Administration
Address:    3300 Metzerott Road
City:       Adelphi
StateProv:  MD
PostalCode: 20783
Country:    US

NetRange:   131.118.0.0 - 131.118.255.255
CIDR:       131.118.0.0/16
NetName:    MINCNET
NetHandle:  NET-131-118-0-0-1
Parent:     NET-131-0-0-0-0
NetType:    Direct Assignment
NameServer: NS.USMD.EDU
NameServer: UMCPNOC.UMS.EDU
NameServer: NOC.USMD.EDU
NameServer: TRANTOR.UMD.EDU
Comment:
RegDate:    1988-11-15
Updated:    1998-11-24

TechHandle: NM162-ARIN
TechName:   Malmberg, Norwin
TechPhone:  +1-301-445-2758
TechEmail:  malmberg@usmh.usmd.edu
```

This address 66.187.232.101 was involved with external to external IP communication. This information was needed to determine if the address is a possible threat

```
OrgName:    Red Hat, Inc.
OrgID:      REDHAT-1
Address:    2600 Meridian Parkway
City:       Durham
StateProv:  NC
PostalCode: 27713
Country:    US

NetRange:   66.187.224.0 - 66.187.239.255
CIDR:       66.187.224.0/20
NetName:    RED-HAT-BLK
NetHandle:  NET-66-187-224-0-1
Parent:     NET-66-0-0-0-0
NetType:    Direct Assignment
NameServer: NS1.REDHAT.COM
NameServer: NS2.REDHAT.COM
Comment:
RegDate:    2001-11-13
Updated:    2001-11-21

TechHandle: JM3008-ARIN
TechName:   Madison, Jay
TechPhone:  +1-919-547-0012
TechEmail:  noc@redhat.com
```

This address of 192.5.3.11 was involved with external to external IP communication with the previous address.

```
OrgName:    City of Beverly Hills
OrgID:      CBH
Address:    9268 West Third Street
City:       Beverly Hills
StateProv:  CA
PostalCode: 90210
Country:    US

NetRange:   192.5.3.0 - 192.5.3.255
CIDR:       192.5.3.0/24
NetName:    CITYBEVHILLS
NetHandle:  NET-192-5-3-0-1
Parent:     NET-192-0-0-0-0
NetType:    Direct Assignment
Comment:
RegDate:    1994-07-01
Updated:    1994-07-01

TechHandle: EF23-ARIN
TechName:   Fraga, Edward
TechPhone:  +1-310-285-2590
TechEmail:
```

3.6 Link Graph

This is something I came across as I analysing the TCP & ICMP SRC and DST outside network alert. Analysing the logs for these ports, I found traffic with the characteristics of the Slapper Worm.

http://www.honeynet.org/scans/scan25/sol/ralf/sotm25_spenneberg.pdf

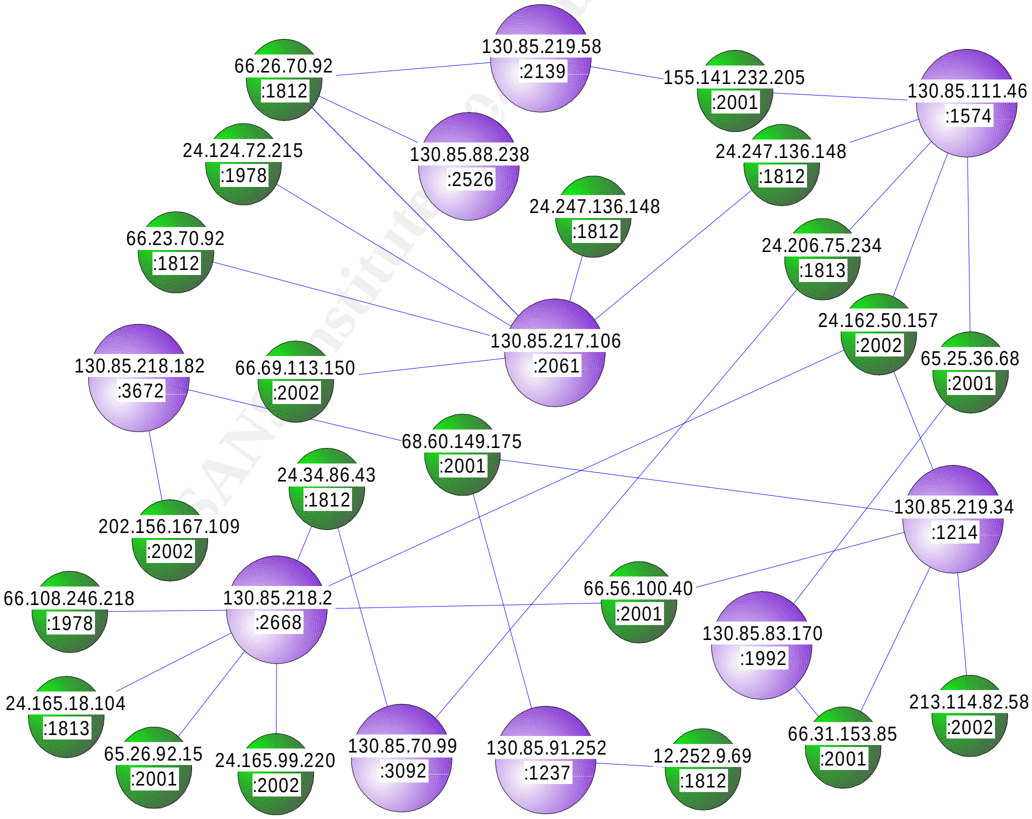
After some investigating, there appears to be range of Slapper worm activity in the log files. The Slapper (and variants) listen on UDP for backdoor commands.

The following shows possible hosts in the scan log that are listening to Slapper ports on UDP 2001, 2002, 1978, 4156, 1812 and TCP 1052.

130.85.104.211	130.85.108.34	130.85.111.34	130.85.111.46	130.85.116.107
130.85.117.133	130.85.1.3	130.85.132.24	130.85.137.7	130.85.153.223
130.85.18.36	130.85.70.164	130.85.70.207	130.85.70.99	130.85.83.170
130.85.84.144	130.85.84.151	130.85.84.244	130.85.88.151	130.85.88.161
130.85.88.205	130.85.88.238	130.85.91.240	130.85.97.101	130.85.97.103
130.85.97.123	130.85.97.134	130.85.97.146	130.85.97.153	130.85.97.163
130.85.97.17	130.85.97.190	130.85.97.21	130.85.97.228	130.85.97.37
130.85.97.41	130.85.97.49	130.85.97.54	130.85.97.56	130.85.97.58
130.85.97.59	130.85.97.68	130.85.97.83	130.85.97.85	130.85.97.89
130.85.97.99	130.85.98.15	130.85.98.37	130.85.98.47	130.85.98.55
130.85.98.74	130.85.98.97	130.85.99.38		

The diagram is a sample of how a worm with multiple variations can infect multiple hosts through both same and differing injection vectors. The connections get very complicated and the threat rises exponentially if devices aren't patched.

Variants and Infections of the Slapper worm



Additional Note
The Slapper worm was not directly detected in the alert files. The spp_http_decodes are an indication but they are not tuned finely enough for detection. I recommend that the table of IP addresses above are investigated immediately to determine the if the Worm exists in the University's network (high probability that it does). The recommedations for the Worm and each of its variants is shown below.

Slapper.A
http://isc.incidents.org/analysis.html?id=167
Slapper.B
http://isc.incidents.org/analysis.html?id=172
Slapper.C (Cinik)
http://isc.incidents.org/analysis.html?id=173
Slapper.C2
http://isc.incidents.org/analysis.html?id=175
SlapperII.A Variant
http://isc.incidents.org/analysis.html?id=176

3.7 Analysis Process

I searched through previous GCIA practicals until I found basic perl scripts that would generate EOIs by various groupings. Some of the scripts I went through were:
Les Gordon
http://www.giac.org/practical/GCIA/Les_Gordon_GCIA.doc
Tod Beardsley
http://www.giac.org/practical/GCIA/Tod_Beardsley_GCIA.doc
Anton Chuvalkin
http://www.giac.org/practical/GCIA/Anton_Chuvakin_GCIA.pdf

Basically I played with the scripts and various greps, cuts, sorts, fgreps etc until I got the information I wanted. Sometimes I had to cut and paste in MS Word or Excel to get the functionality I needed. The UNIX commands were inserted directly into the document with the output where possible so whoever reads the practical can follow what I was trying to achieve.

Before analysing the various alerts, I sanitised the alert.all file by removing some of the Internet noise such as portscanning activity from scan log to get better idea of traffic stastics.

With the scan file I separated into it SYN, UDP, FIN and other to get a better understanding of the types of scanning that was occuring.

The alert files were concatenated into one file called alerts.all and the scan files in a file called scans.all. This was done to get a holistic view of the logs.

References:

Webopedia
http://www.webopedia.com/TERM/I/intrusion_detection_system.html

Snort IDS
http://www.snort.org/about.html July 2003
http://www.snort.org/docs/faq.html#4.17 July 2003
http://www.snort.org/snort-db/sid.html?sid=648 July 2003
http://www.snort.org/snort-db/sid.html?sid=1394 July 2003
http://www.snort.org/snort-db/sid.html?id=650 July 2003
http://www.snort.org/snort-db/sid.html?sid=312 July 2003
UDP http://www.snort.org/snort-db/sid.html?sid=1411 July 2003
TCP http://www.snort.org/snort-db/sid.html?sid=1412 July 2003
http://www.snort.org/snort-db/sid.html?sid=356 July 2003

MYSQL
http://searchdatabase.techtarget.com/sDefinition/0,,sid13_gci516819,00.html April 2003

Queensland State Archives website
http://www.archives.qld.gov.au April 2003

Defence Signals Directorate ACSI 33 Handbook 13
http://www.dsd.gov.au/infosec/acsi33/HB13.html April 2003

Vendor/Ethernet MAC Address Lookup and Search
http://www.coffer.com/mac_find June 2003

RIPE Network Coordination Centre
http://www.ripe.net/db/whois/whois.html July 2003

Request For Comment
http://www.faqs.org/rfcs/rfc1035.html July 2003

Common Vulnerabilities and Exposures July 2003
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0010
cve,CAN-2002-0013 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0013
cve,CAN-2002-0012 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0012
cve,CAN-1999-0517 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0517

Wikipedia
http://www.wikipedia.org/wiki/CHAOSnet July 2003

Request For Comment
http://www.faqs.org/rfcs/rfc1035.html July 2003
http://www.faqs.org/rfcs/rfc919.html July 2003

Mixter
http://mixter.void.ru/about.html June 2003

The ITsecurity.com Dictionary+ of Information Security
http://www.itsecurity.com/dictionary/dictionary.htm July 2003

lists.jammed.com mail list
http://lists.jammed.com/incidents/2001/05/0037.html June 2003
http://lists.jammed.com/incidents/2001/10/0023.html July 2003

Microsoft July 2003
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/iis5cl.asp
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/virus/Nimda.asp

Asia Pacific Network Information Centre
http://www.apnic.net/apnic-bin/whois.pl July 2003

ARIN July 2003
http://ws.arin.net
http://ws.arin.net/cgi-bin/whois.pl

Andy Norman, Matthew Williamson, Hitting back at Code Red, Thursday, 14 March 2002
http://www.hpl.hp.com/techreports/2002/HPL-2002-111.pdf July 2003

The virus affected versions 4 and 5 of Microsoft's IIS web server, exploiting a buffer overflow vulnerability in the indexing service. The attack consisted of a specially crafted HTTP request that when sent to IIS would cause malicious code to take control of the web server. The primary behaviour of the malicious code was to attempt to propagate as rapidly as possible, by generating IP addresses at random1, and making infective HTTP requests to those addresses. If any of these machines were running vulnerable installations of IIS, they too would become infected. The code attempted to propagate at an incredible rate, with many HTTP requests being sent every second. It has been estimated that the virus could infect on the order of half a million IP addresses a day [2]. The secondary behaviour of Code Red was to deface web sites on the infected host, and prepare the infected machine to participate in a distributed denial of service (DDOS) attack on www.whitehouse.gov, at certain times. Later variants of Code Red (e.g. Code Red II [3]) left Trojan horses and open shares on the compromised machine.

Security Focus
http://www.securityfocus.com July 2003

eEye Digital Security
http://www.eeye.com/html/advisories/coderedII.zip July 2003

eEye Digital Security (http://www.eeye.com/) has recently released a free tool which you can use to scan your network for IIS servers which may still be vulnerable to the "Code Red" (and hence "CodeRedII") worm. You can download this tool from the eEye site directly at: http://www.incidents.org/archives/intrusions/msg02292.html

Mail Groups
http://cert.uni-stuttgart.de/archive/intrusions/2002/12/msg00204.html July 2003
http://www.incidents.org/archives/intrusions/msg02292.html July 2003
http://www.incidents.org/archives/intrusions/msg00300.html July 2003
http://cert.uni-stuttgart.de/archive/intrusions/2003/07/pgp00012.pgp July 2003

Cisco
http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml June 2003

Internet Storm Center
http://www.incidents.org/logs July 2003
Slapper.A http://isc.incidents.org/analysis.html?id=167 July 2003
Slapper.B http://isc.incidents.org/analysis.html?id=172 July 2003
Slapper.C (Cinik) http://isc.incidents.org/analysis.html?id=173 July 2003
Slapper.C2 http://isc.incidents.org/analysis.html?id=175 July 2003
SlapperII.A Variant http://isc.incidents.org/analysis.html?id=176 July 2003

Les Gordon GCIA v3.2 November 22, 2002
http://www.giac.org/practical/GCIA/Les_Gordon_GCIA.doc June 2003

Brian Cahoon GCIA v3.3 January 6, 2002
http://www.giac.org/practical/GCIA/Brian_Cahoon_GCIA.pdf June 2003

Tod Beardsley GCIA v3.3 January 5 2003
http://www.giac.org/practical/Tod_Beardsley_GCIA.doc June 2003

Again, except for KaZaA, as Tod pointed out, these packets are used for reconnaissance. This is really just background noise on the network.

Rick Larabee GCIA v3.2
http://www.giac.org/practical/GCIA/Rick_Larabee_GCIA.doc June 2003

All packets that have the 21 ECN Flags set, the Syn flag set, and TOS of 0x00 are registered as Queso scans in the alert files as well.

SANS
http://www.sans.org/y2k/adore.htm

William Stearns from Dartmouth's ISTS (http://www.sans.org/y2k/adore.htm).
Adore scans the Internet checking Linux hosts to determine whether they are vulnerable to any of the following well-known exploits: LPRng, rpc-statd, wu-ftpd and BIND.

http://www.sans.org/y2k/051300.htm

Paul Young GCIA v3.2 2003
http://www.giac.org/practical/Paul_Young_GCIA.pdf June 2003

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.

Al Williams GCIA v3.3 August 2003
http://www.giac.org/practical/Al_Williams_GCIA.pdf June 2003

Scott Shinberg GCIA v2.9 July 2001
http://www.giac.org/practical/Scott_Shinberg_GCIA.doc June 2003

Legitimate traffic would be characterized at that which uses one high port like 27374 and one low port, such as 25 for email. Traffic that uses two high 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

Patrick Powell LPRng Web Page
http://www.lprng.com/LPRng-HOWTO-Multipart/rfc1179ref.htm

Al Maslowski GCIA v3.3 July 5 2003
http://www.giac.org/practical/Al_Maslowski-Yerges_GCIA.pdf June 2003

Windows and Samba clients typically use this type of NetBIOS traffic to find hosts even if they are involved in other communications with the host if they can't resolve the name with DNS

iFolder Tips - June 30, 2003
http://nscsysop.hypermart.net/ifolder.html July 2003

Neohapsis Archives
http://archives.neohapsis.com/archives/snort/2001-01/0200.html July 2003

Adelphia Communications
http://www.adelphia.net July 2003

iLink Holdings Limited
http://www.ilink.net July 2003

Ralf Spenneberg, Scan 25 Analyze a Worm, Novemver 27 2002
http://www.honeynet.org/scans/scan25/sol/ralf/sotm25_spenneberg.pdf July 2003

Anton Chuvakin GCIA v3.1 June 27 2002
http://www.giac.org/practical/GCIA/Anton_Chuvakin_GCIA.pdf July 2003

Appendix A: Risk Analysis

Confidentiality

Risk ID	Risk Description	Consequence	Likelihood	Control	Consequence	Likelihood	Residual Risk
	Confidentiality:			**HP: Highly Protected; Pr: Protected; X: X-in-Confidence; Pu: Public**			
C1	IDS Service loses data confidentiality	Major-HP Major-Pr Moderate-X Moderate-Pu	Likely	2 factor token authentication. Restricted access to IDS devices. Password policy & Disclaimer Data transfer via encrypted tunnels.	Major-HP Major-Pr Minor-X Minor-Pu	Rare	C C D D
C2	Application / Console Server loses data confidentiality	Moderate-HP Moderate-Pr Minor-X Minor-Pu	Likely	Restricted access & SSH only. Password policy & Disclaimer Standard toolkit.	Moderate-HP Moderate-Pr Minor-X Minor-Pu	Rare	C C D D
C3	Log Server loses data confidentiality	Major-HP Major-Pr Moderate-X Moderate-Pu	Likely	Restricted access & SSH only. Password policy. Standard toolkit. Disclaimer	Major-HP Major-Pr Moderate-X Moderate-Pu	Rare	C C D D
C4	Sensors loses data confidentiality	Major-HP Moderate-Pr Minor-X Minor-Pu	Likely	Restricted access & SSH only. Password policy & Disclaimer Standard toolkit.	Moderate-HP Moderate-Pr Minor-X Minor-Pu	Rare	C C D D
C5	Distribution Network loses data confidentiality	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Likely	Restricted access & CONFIG templates 2 Factor token authentication Password policy	Major-HP Major-Pr Minor-X Minor-Pu	Rare	C C D D
C6	Access Network loses data confidentiality	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Likely	Restricted access & CONFIG templates 2 Factor token authentication Password policy &Disclaimer	Major-HP Major-Pr Minor-X Minor-Pu	Rare	C C D D
C7	Security Layer loses data confidentiality	Moderate-HP Moderate-Pr Minor-X Minor-Pu	Likely	Restricted access 2 Factor token authentication. Standard toolkit. Password policy & Disclaimer	Major-HP Major-Pr Minor-X Minor-Pu	Rare	C C D D

Integrity

Risk ID	Risk Description	Consequence	Likelihood	Control	Consequence	Likelihood	Residual Risk
I1	**Integrity:**			**HP: Highly Protected; Pr: Protected; X: X-in-Confidence; Pu: Public**			
I2	IDS Service loses data integrity	Major-HP Major-Pr Minor-X Minor-Pu	Unlikely	IDS on devices	Major-HP Major-Pr Minor-X Minor-Pu	Rare	C C D D
I3	Application / Console Server loses data integrity	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Unlikely	HIDS on Console Server	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Rare	C C D D
I4	Log Server loses data integrity	Major-HP Major-Pr Minor-X Minor-Pu	Unlikely	HIDS on Log Server Integrity checks between local devices and log server	Major-HP Major-Pr Minor-X Minor-Pu	Rare	C C D D
I5	Sensors loses data integrity	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Unlikely	HIDS on Sensors	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Rare	C C D D
I6	Distribution Network loses data integrity	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Unlikely	NIDS on Distribution Network	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Rare	C C D D
I7	Access Network loses data integrity	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Unlikely	NIDS on Access Network	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Rare	C C D D
I8	Security Layer loses data integrity	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Unlikely	HIDS on Firewall Device(s)	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Rare	C C D D

Availability

Risk ID	Risk Description	Consequence	Likelihood	Control	Consequence	Likelihood	Residual Risk
	Availability:			**HP: Highly Protected; Pr: Protected; X: X-in-Confidence; Pu: Public**			
A1	IDS Service is unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D
A2	Application / Console Server is unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D
A3	Log Server is unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D
A4	Sensors are unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D
A5	Distribution Network is unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D
A6	Access Network is unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D
A7	Security Layer is unavailable	Major-HP Major-Pr Minor-X Minor-Pu	Likely	Chief Operating Practice Vendor support & maintenance OLA	Moderate-HP Moderate-Pr Minor -X Minor -Pu	Possible	C C D D