Global Information Assurance Certification Paper

Intrusion Detection In Depth GCIA Practical Assignment

Version 3.0

Scott Baird
May 24, 2002

Table of Contents

(NOTE: This document has been formatted for two-sided printing)

Assignment 1: Describe the State of Intrusion Detection

Purpose: A Behind the Scenes Look at RealSecure

This paper takes a behind the scenes look at the RealSecure Intrusion Detection System to see how it is put together. The knowledge provided here will allow you to get at the data directly, and with some database and web authoring knowledge, allow you to develop your own web application to report on and monitor your RealSecure IDS.

Introduction

RealSecure is an Intrusion Detection System created by Internet Security Systems (ISS). They have provided good summary information about RealSecure and its components in the introduction to their Workgroup Manager User Guide. It states the following:

> RealSecure assets are daemons and daemon components that you have defined on your network. Daemon components include sensors and event collectors. RealSecure sensors include network, OS, and server sensors. Sensors detect attacks to the network or system. An event collector collects logged real-time events from the sensors and stores these events in the enterprise database. The shipped enterprise database is MSDE, but can be upgraded to MS SQL 7 or greater.

The following table describes the RealSecure components:

Component	Description
Workgroup Manager	The Graphical User Interface (GUI) and the collected database from the sensors. This includes the RealSecure console, event collector, and databases.
console	The central controlling point for the sensors. Manages the RealSecure assets deployed across your intranet. Also runs reports from the enterprise database.
event collector	The event collector maintains the enterprise database that stores events detected by sensors. You can run and view reports from the enterprise database.
network sensor	The network sensor runs on a network segment, analyzing the traffic flow and looking for intrusions and signs of network abuse.
OS sensor	The OS sensor runs on a crucial computer system, monitoring user and administrator activity and watching for signs of improper system use.

Component	Description
Server sensor	The server sensor is similar to the network sensor, except it monitors network traffic to and from only one computer. The OS sensor is integrated with the server sensor, so the server sensor is also able to monitor system information.
Asset database	Stores information about your network assets, such as computers and RealSecure components on your network, and allows multiple consoles to use the same set of assets. Contains the names and IP addresses of network and RealSecure assets.
Console database	Stores events from version 5.x sensors.
Enterprise database	Stores events detected by version 6.x sensors, which communicate directly with the event collector. Reports can be run against this database.

(Internet Security Solutions, p.2)

The basic configuration consists of the asset database (RSAsset60), the enterprise database (ISSED), one or more event collectors, and one or more sensors (some combination of network, OS and server). Each of the sensors has a local "database" file that stores events as the sensor detects them. The event collector(s) connect to the sensor(s) over an encrypted channel and retrieves any new events. This allows downtime for the databases or event collectors without losing events.

Rather than address strengths and weaknesses, or the best way to deploy a RealSecure IDS, this paper will take a look at some of the core tables within the databases and how to harness the data within them. See the list of references if you wish to have more data regarding deployment, key features, or strengths and weaknesses of RealSecure or IDS's in general. (Nexus; Wassom; Marshall)

The configuration being considered here is made up of only serve sensors. Although most, if not all, of the information provided will probably apply to OS and network sensors as well. But as the available environment consists of only server sensors, that is all that is addressed here.

Tables

The naming convention of <database_name>..<table_name> will be used for clarity as to which database the table belongs. However, most of the tables that are used are found in the Enterprise database, which is named ISSED. There are a couple of useful tables in the Asset database, which is named RSAsset60.

RSAsset60..RSAsset

This table contains all the sensors that are currently configured in the system. It is the primary table in the Asset database and has the following structure:

Column Name	Type
ID	int
Name	varchar(255)
AssetType	int
IsMember	bit
ProviderID	int
ProviderAssetType	int
IsFirewall	bit
ControlDestPort	int
ControlSrcPort	int

The columns of interest are: Name, and ProviderAssetType. The Name column contains the name of the sensor. The ProviderAssetType column contains an integer which specifies the type of asset. This could be used to join to the RSProviderAssetType to get the name of the type or can simply be used to identify the type of asset wanted.

RSAsset60..RSProviderAssetType

This table contains a list of the asset types and has the following structure and data:

Column Name	Type
ProviderAssetType	int
ProviderID	int
DisplayName	varchar(255)

ProviderAssetType	ProviderID	DisplayName
1	1	Group
2	1	Host
3	1	Network
4	1	issDaemon
5	1	issDaemon Component
6	1	Category1
7	1	EventCollector

The ProviderAssetType that is of interest is the issDaemon Component and as can be seen in the table above, this is type 5. So when selecting records from the RSAsset60..RSAsset table, the only records wanted have a ProverAssetType=5.

In a simple RealSecure configuration where one server acts as the console, event collector, and database server, these are the only two tables from the RSAsset60 database that are interesting.

ISSED..Events

This table is the heart of the system. There are a few others that have additional data, but the ISSED..Events table is where all "logged" events get logged. It has the following structure:

Column Name	Type
SecChkID	int
ProtocolID	int
DayID	int
TimeID	smallint
EventID	int
ActionID	int
ProdID	int
EventDate	datetime
SrcPort	int
SrcIPAddress	varchar(60)
SrcPortName	varchar(60)
DestPort	int
DestIPAddress	varchar(60)
DestPortName	varchar(60)
SrcEthernetAddr	varchar(60)
SrcEthernetVendor	varchar(60)
DestEthernetAddr	varchar(60)
DestEthernetVendor	varchar(60)
TCPFlags	varchar(50)
ICMPType	varchar(50)
ICMPCode	varchar(50)
EventPriority	int
MonitorIPAddress	IPADDRESS_TYPE
RemoteEventID	int
AlertID	varchar(26)
AlertTimeSeqID	int
SensorAddress	varchar(60)
AlertType	int
SensorName	varchar(60)
LocalTimezoneOffset	int
AlertTimePrecision	int
AlertNameType	int
AttackSuccessful	tinyint
AttackFragmented	tinyint
DisplaySrcIPAddress	varchar(60)
DisplayDestIPAddress	varchar(60)
DisplaySensorAddress	varchar(60)
OrigEventName	varchar(60)
AttackOrigin	varchar(60)
ResourceID	int
ResourceSubID	varchar(60)
Application	varchar(60)

Column Name	Type
UserName	varchar(60)
State	tinyint
AlertFlags	int

The key columns from this table are:

Column Name	Description
ProtocolID	The number of the network protocol. This can be used to join to the ISSED..Protocol table to get the ProtocolName (name) and ProtocolDesc (description)
EventID	A unique identifier for an event. It is used to join to other tables to retrieve more information about the event. The most common table used would be ISSED..EventParams, which has other information that is event dependent.
EventDate	The date and time, stored in GMT, that the event was recorded by the sensor. This may not correspond to the date and time it was logged, as there may have been a communication problem between the sensor and the console.
SrcPort	The source port number. It can be used to join to the ISSED..Services table on the ServRFCPort column to obtain the ServiceName (name), ServiceProtocol (protocol), and ServBriefDesc (description). However, the SrcPortName (name) is already stored in this table, so there's no need to go to the ISSED..Services table if that is all that is wanted.
SrcIPAddress	The source IP address, but it is in a full xxx.xxx.xxx.xxx format with leading 0's when necessary. Instead use the DisplaySrcIPAddress column listed below.
SrcPortName	The name of the service expected to be running on the source port.
DestPort	The destination port number. See SrcPort above for related information.
DestIPAddress	The destination IP address with the same caveats as the previous SrcIPAddress.
DestPortName	The name of the service expected to be running on the destination port.
SrcEthernetAddr	The MAC address of the source computer.
SrcEthernetVendor	The vendor of the network card of the source computer.
DestEthernetAddr	The MAC address of the destination computer.
DestEthernetVendor	The vendor of the network card of the destination computer.

Column Name	Description
TCPFlags	TCP flag: URG, ACK, PSH, RST, SYN, FIN. Unfortunately, this is not currently used.
ICMPType	ICMP Type: 0=Echo Reply, 3=Destination Unreachable, 4=Source Quench, etc.
ICMPCode	ICMP Code: Varies with ICMP Type
EventPriority	The priority of the event: 1=High, 2=Medium, 3=Low
SensorAddress	The sensor's IP address, but it is in a full xxx.xxx.xxx.xxx format with leading 0's when necessary. Instead use the DisplaySrcIPAddress column listed below.
SensorName	The name of the sensor reporting the event.
LocalTimezoneOffset	The number of seconds difference between the sensor's local time and GMT.
AttackSuccessful	A number representing the success of an attack: 0=Success, 1=Failure, 2=Unknown, NULL=Not Applicable
DisplaySrcIPAddress	The source IP address. This should be used rather than SrcIPAddress.
DisplayDestIPAddress	The destination IP address. This should be used rather than DestIPAddress.
DisplaySensorAddress	The sensor's IP address. This should be used rather than SensorAddress.
OrigEventName	The name of the event detected.
AttackOrigin	The name of the source of the attack.
Application	The name of the application for which the event was generated, for example the application MSSQL could report an MSSQL_Shutdown event.
UserName	The name of the user that performed the action that generated the event. This is only useful for "Server" related events such as Logon_with_admin_privileges or User_added_to_local_admin_group.

ISSED..EventParams

This table stores variable information based on the type of event logged.

Column Name	Type
EventParamID	int
ParamName	varchar(50)
EventID	int
ParamValue	varchar(255)
ParamOrder	int
ParamDataType	varchar(30)

Column Name	Type
ParamBlob	RAWDATA_TYPE

The key columns from this table are:

Column Name	Description
ParamName	The name of the parameter.
EventID	The Event ID that links back to the ISSED..Events table.
ParamValue	The value for the parameter.

Although the following list is not exhaustive, the ParamName commonly takes on one or more of the following values:

```
AttackOrigin
AlertFormatVersion
ResponseList
DestinationAddress
SourceAddress
SystemAgent
UserName
User
User's Domain
Privileges
Object Handle
Object Server
Process ID
Workstation
Program
AttackSuccessful
Code
EventType
Message
Server
Service
Purpose
Group
IANAProtocolId
DestinationPort
DestinationPortName
SourcePort
SourcePortName
Real Group
Real User
```

In turn, each of these would then have its value stored in ParamValue. Here are all the Params for one event.

ParamName	EventID	ParamValue
AlertFormatVersion	1127217	85
ResponseList	1127217	LOGDB=LogWithoutRaw:0
SystemAgent	1127217	Server001
IssueID	1127217	2001011
IssueName	1127217	SMTP relay attempt
err	1127217	Relaying_is_prohibited
ret	1127217	550
SourcePort	1127217	3772
SourcePortName	1127217	Port# 3772
DestinationPort	1127217	25
DestinationPortName	1127217	smtp
IANAProtocolId	1127217	6
SourceAddress	1127217	X.X.168.10
SourceEthernetAddress	1127217	00:00:00:00:11:11
DestinationAddress	1127217	X.X.68.41
AttackSuccessful	1127217	0

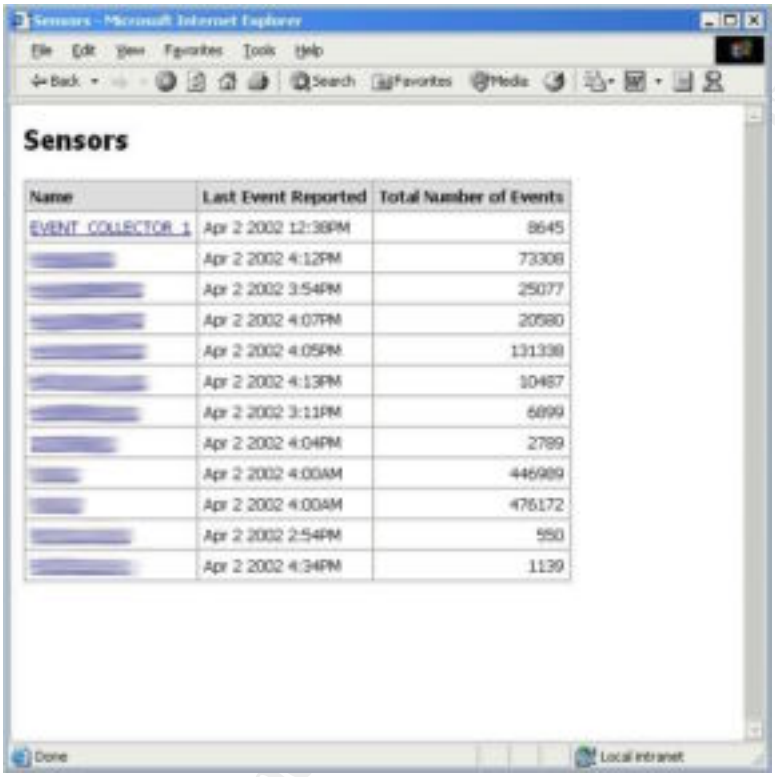
Sometimes this information is redundant, while other times there is a value added.

View of Sample Application

Now that the database has been deciphered, with a little bit of SQL and ASP code, you can create a website to monitor the IDS as well as get custom reports.

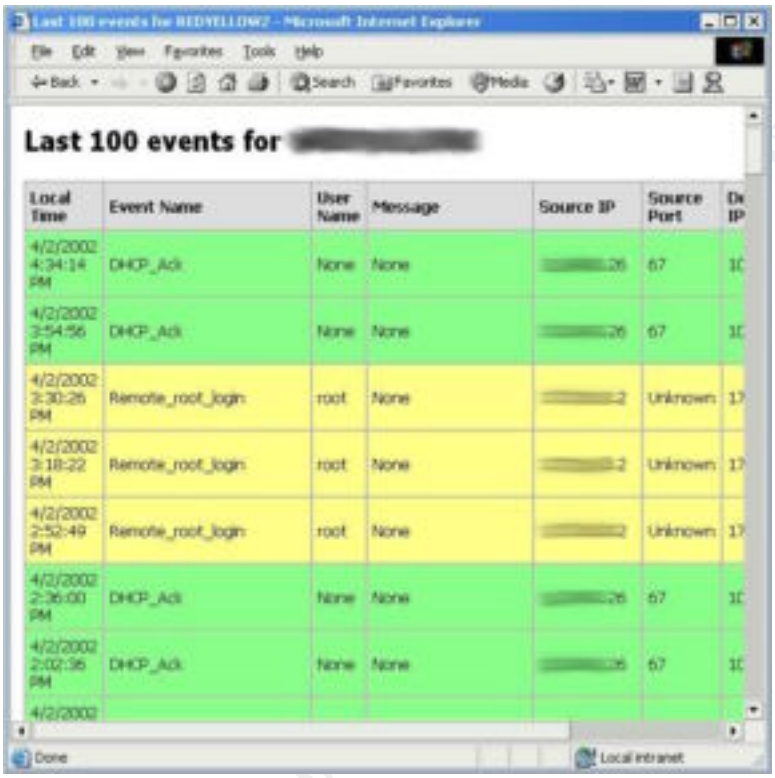
The home page allows you to choose from viewing sensors, events, or ports.

When choosing any of these three, a summary page shows the chosen item along with the number of events logged for that item as well as the last time an event was logged. In this case, all the configured sensors are listed.



When choosing events, a similar list is produced but by event name. Choosing ports gives a list of the top 10 destination ports, along with the last event reported and total number of events for that port. This is similar to the "Top 10 Target Ports" at DShield.org. (Euclidean Consulting)

Then selecting one of the links will show you more detailed information for particular item.



Of course, this is just the beginning. Since all the data is stored in a relational database, many different views of the data can be generated. In addition, filters can be added to weed out false positives.

Covering how to install and configure SQL Server and IIS is out of scope for this paper. Writing the SQL and ASP code itself is out of scope, as well. For more information regarding these topics, see the list of references. (Microsoft, Microsoft Press)

Code Behind the Sample Application

The stored procedures were created in a separate database named Monitor located on the same database server as the RSAsset60 and ISSED databases. The stored procedures were kept separate to ensure they would not interfere with the RealSecure database structure.

Stored Procedure - pSensorInfo

This returns a list of all the configured sensors along with the number of events it has logged and the last time an event was logged. The name of the issDaemon Component has been set as <machine_name>@<ip_address>. On the other hand, the events logged contain the <machine_name> as the sensor name.

Therefore, some string manipulation is done to link the RSAsset60..RSAsset table to the ISSED..Events table.

```
USE Monitor
GO

PRINT 'Creating pSensorInfo'

IF EXISTS (SELECT * FROM sysobjects where name =
'pSensorInfo' and type = 'P')
    DROP PROCEDURE pSensorInfo
GO

CREATE PROCEDURE pSensorInfo AS
SELECT
    'SensorName' = UPPER( E.SensorName )
    ,'LastEvent' = ISNULL( CONVERT( VARCHAR,
MAX(DATEADD(SECOND, -E.LocalTimezoneOffset, E.EventDate))),
'None' )
    ,'EventCount' = ISNULL( COUNT(E.SensorName), 0 )
FROM
    RSAsset60..RSAsset A (NOLOCK)
LEFT OUTER JOIN
    ISSED..Events E (NOLOCK)
ON
    E.SensorName = SUBSTRING( A.Name, 1, CHARINDEX( '@',
A.Name ) - 1 )
WHERE
    ProviderAssetType = 5
GROUP BY
    E.SensorName
    ,A.Name
ORDER BY
    E.SensorName
GO

GRANT EXECUTE ON pSensorInfo TO public
GO

PRINT 'Done'
```

Stored Procedure - pSensorLast100Events

For a given sensor, this returns the most recent events logged, up to 100 total, and sorts them by date in descending order.

```
USE Monitor
GO

PRINT 'Creating pSensorLast100Events'

IF EXISTS (SELECT * FROM sysobjects where name =
'pSensorLast100Events' and type = 'P')
    DROP PROCEDURE pSensorLast100Events
GO

CREATE PROCEDURE pSensorLast100Events
    @SensorName varchar(32)
AS
SELECT TOP 100
    'LocalTime' = DATEADD( SECOND, -E.LocalTimezoneOffset,
E.EventDate )
    ,'OrigEventName' = ISNULL( E.OrigEventName, 'None' )
    ,'UserName' = ISNULL( E.UserName, 'None' )
    ,'Message' = ISNULL( EP.ParamValue, 'None' )
    ,'DisplaySrcIPAddress' = ISNULL( E.DisplaySrcIPAddress,
'Unknown' )
    ,'SrcPort' = ISNULL( CONVERT( varchar, E.SrcPort ),
'Unknown' )
    ,'DisplayDestIPAddress' = ISNULL(
E.DisplayDestIPAddress, 'Unknown' )
    ,'DestPort' = ISNULL( CONVERT( varchar, E.DestPort ),
'Unknown' )
    ,E.EventPriority
FROM
    ISSED..Events E (NOLOCK)
LEFT OUTER JOIN
    ISSED..EventParams EP (NOLOCK)
ON
    E.EventID = EP.EventID
AND EP.ParamName = 'Message'
WHERE
    SensorName = @SensorName
ORDER BY
    EventDate DESC
GO

GRANT EXECUTE ON pSensorLast100Events TO public
GO
```

```
PRINT 'Done'
```

Stored Procedure - pEventInfo

This returns a list of each type of event logged along with the number of times it has been logged and the last time that it was logged.

```
USE Monitor
GO

PRINT 'Creating pEventInfo'

IF EXISTS (SELECT * FROM sysobjects where name =
'pEventInfo' and type = 'P')
    DROP PROCEDURE pEventInfo
GO

CREATE PROCEDURE pEventInfo AS
SELECT
    'OrigEventName' = E.OrigEventName
    ,'LastEvent'    = ISNULL( CONVERT( VARCHAR,
MAX(DATEADD(SECOND, -E.LocalTimezoneOffset, E.EventDate))),
'None' )
    ,'EventCount'   = ISNULL( COUNT(E.OrigEventName), 0 )
    ,E.EventPriority
FROM
    ISSED..Events E (NOLOCK)
GROUP BY
    E.OrigEventName
    ,E.EventPriority
ORDER BY
    E.OrigEventName
GO

GRANT EXECUTE ON pEventInfo TO public
GO

PRINT 'Done'
```

Stored Procedure - pEventLast100Details

For a given event, this returns the most recently logged occurrences of this event, up to 100 total, and sorts them by date in descending order.

```
USE Monitor
```

```
GO

PRINT 'Creating pEventLast100Details'

IF EXISTS (SELECT * FROM sysobjects where name =
'pEventLast100Details' and type = 'P')
    DROP PROCEDURE pEventLast100Details
GO

CREATE PROCEDURE pEventLast100Details
    @OrigEventName varchar(64)
AS
SELECT TOP 100
    'LocalTime' = DATEADD( SECOND, -E.LocalTimezoneOffset,
E.EventDate )
    ,'SensorName' = ISNULL( E.SensorName, 'None' )
    ,'UserName' = ISNULL( E.UserName, 'None' )
    ,'Message' = ISNULL( EP.ParamValue, 'None' )
    ,'DisplaySrcIPAddress' = ISNULL( E.DisplaySrcIPAddress,
'Unknown' )
    ,'SrcPort' = ISNULL( CONVERT( varchar, E.SrcPort ),
'Unknown' )
    ,'DisplayDestIPAddress' = ISNULL(
E.DisplayDestIPAddress, 'Unknown' )
    ,'DestPort' = ISNULL( CONVERT( varchar, E.DestPort ),
'Unknown' )
    ,E.EventPriority
FROM
    ISSED..Events E (NOLOCK)
LEFT OUTER JOIN
    ISSED..EventParams EP (NOLOCK)
ON
    E.EventID = EP.EventID
AND EP.ParamName = 'Message'
WHERE
    OrigEventName = @OrigEventName
ORDER BY
    EventDate DESC
GO

GRANT EXECUTE ON pEventLast100Details TO public
GO

PRINT 'Done'
```

Stored Procedure - pPortInfo

This returns a list of the top 10 destination ports from the logged events along with the number of events and the last time an event was logged that had it as the destination port.

```
USE Monitor
GO

PRINT 'Creating pPortInfo'

IF EXISTS (SELECT * FROM sysobjects where name =
'pPortInfo' and type = 'P')
    DROP PROCEDURE pPortInfo
GO

CREATE PROCEDURE pPortInfo AS
SELECT TOP 10
    'DestPort' = E.DestPort
    ,'DestPortName' = E.DestPortName
    ,'LastEvent' = ISNULL( CONVERT( VARCHAR,
MAX(DATEADD(SECOND, -E.LocalTimezoneOffset, E.EventDate))),
'None' )
    ,'EventCount' = ISNULL( COUNT(E.DestPort), 0 )
FROM
    ISSED..Events E (NOLOCK)
GROUP BY
    E.DestPort
    ,E.DestPortName
ORDER BY
    ISNULL( COUNT(E.DestPort), 0 ) DESC
GO

GRANT EXECUTE ON pPortInfo TO public
GO

PRINT 'Done'
```

Stored Procedure - pPortLast100Events

For a given port, this returns the most recently logged events with the port as the destination, up to 100 total, and sorts them by date in descending order.

```
USE Monitor
```

```
GO

PRINT 'Creating pPortLast100Events'

IF EXISTS (SELECT * FROM sysobjects where name =
'pPortLast100Events' and type = 'P')
    DROP PROCEDURE pPortLast100Events
GO

CREATE PROCEDURE pPortLast100Events
    @PortNumber varchar(32)
AS
SELECT TOP 100
    'LocalTime' = DATEADD( SECOND, -E.LocalTimezoneOffset,
E.EventDate )
    ,'SensorName' = UPPER( E.SensorName )
    ,'OrigEventName' = ISNULL( E.OrigEventName, 'None' )
    ,'UserName' = ISNULL( E.UserName, 'None' )
    ,'Message' = ISNULL( EP.ParamValue, 'None' )
    ,'DisplaySrcIPAddress' = ISNULL( E.DisplaySrcIPAddress,
'Unknown' )
    ,'SrcPort' = ISNULL( CONVERT( varchar, E.SrcPort ),
'Unknown' )
    ,'DisplayDestIPAddress' = ISNULL(
E.DisplayDestIPAddress, 'Unknown' )
    ,'DestPort' = ISNULL( CONVERT( varchar, E.DestPort ),
'Unknown' )
    ,E.EventPriority
FROM
    ISSED..Events E (NOLOCK)
LEFT OUTER JOIN
    ISSED..EventParams EP (NOLOCK)
ON
    E.EventID = EP.EventID
AND EP.ParamName = 'Message'
WHERE
    DestPort = @PortNumber
ORDER BY
    EventDate DESC
GO

GRANT EXECUTE ON pPortLast100Events TO public
GO
```

```
PRINT 'Done'
```

Web Page - Default.htm

This displays a very simple default home page with three choices.

```
<html>
<head>
    <title>IDS Home</title>
</head>

<body>
<h1>IDS Home</h1>
<a href="Sensors.asp">List Sensors</a><br>
<a href="Events.asp">List Events</a><br>
<a href="Ports.asp">List Top 10 Ports</a><br>

</body>
</html>
```

Web Page - Sensors.asp

This displays all the currently configured sensors in the RealSecure system and for each sensor, the last time an event was logged and the total number of events.

```
<%
' Open the Database Connection
Set Con =Server.CreateObject("ADODB.Connection")
Con.Open "DRIVER={SQL
Server};SERVER=DBSERVER;UID=USER;PWD=PASSWORD;DATABASE=Moni
tor"

sqlString = "EXEC pSensorInfo"

SET RS = Con.Execute( sqlString )
%>
<html>
<head><title>Sensors</title></head>
<body>
<h1>Sensors</h1>
<%
IF RS.EOF THEN
%>
<b>There are no sensors currently configured.</b>
```

```
<%
ELSE
%>
<table cellpadding=4 cellspacing=0 border=1>
<tr>
  <td bgcolor=DDDDDD>
    <b>Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Last Event Reported</b>
  </td>
  <td bgcolor=DDDDDD align=right>
    <b>Total Number of Events</b>
  </td>
</tr>
<%
WHILE NOT RS.EOF
%>
<tr>
  <td>
    <a href="SensorEvents.asp?ids_sensor=<%=RS(
"SensorName" )%>">
    <%=RS( "SensorName" )%></a>
  </td>
  <td>
    <%=RS( "LastEvent" )%>
  </td>
  <td align=right>
   <%=RS( "EventCount" )%>
  </td>
</tr>
<%
RS.MoveNext
WEND
%>
<tr>
</table>
<%
END IF
%>

</body>
</html>
```

Web Page - SensorEvents.asp

This displays the most recent events, up to 100, for the server selected. The events are color coded by priority. Low is green. Medium is yellow. High is red.

```
<%
ids_sensor = TRIM( Request( "ids_sensor" ) )
%>

<%
' Open the Database Connection
Set Con =Server.CreateObject("ADODB.Connection")
Con.Open "DRIVER={SQL
Server};SERVER=DBSERVER;UID=USER;PWD=PASSWORD;DATABASE=Moni
tor"

sqlString = "EXEC pSensorLast100Events '" & ids_sensor &
"'"

SET RS = Con.Execute( sqlString )
%>
<html>
<head><title>Last 100 events for <%= ids_sensor
%></title></head>
<body>
<h1>Last 100 events for <%= ids_sensor %></h1>
<%
IF RS.EOF THEN
%>
<b>There are no events for <%= ids_sensor %></b>
<%
ELSE
%>
<table cellpadding=4 cellspacing=0 border=1>
<tr>
  <td bgcolor=DDDDDD>
    <b>Local Time</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Event Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>User Name</b>
  </td>
  <td bgcolor=DDDDDD>
```

```
    <b>Message</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Source IP</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Source Port</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Destination IP</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Destination Port</b>
  </td>
</tr>
<%
WHILE NOT RS.EOF
IF RS( "EventPriority" ) = 1 THEN
    BG = "FF8888"
ELSEIF RS( "EventPriority" ) = 2 THEN
    BG = "FFFF88"
ELSEIF RS( "EventPriority" ) = 3 THEN
    BG = "88FF88"
ELSE
    BG = "FFFFFF"
END IF
%>
<tr>
  <td bgcolor=<%= BG %>>
  <%=RS( "LocalTime" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "OrigEventName" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "UserName" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "Message" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DisplaySrcIPAddress" )%>
  </td>
  <td bgcolor=<%= BG %>>
```

```
  <%=RS( "SrcPort" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DisplayDestIPAddress" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DestPort" )%>
  </td>
</tr>
<%
RS.MoveNext
WEND
%>
</table>
<%
END IF
%>

</body>
</html>
```

Web Page - Events.asp

This displays all the events that have been logged in the RealSecure system and for each event, the last time it was logged as well as the total number of times.

```
<%
' Open the Database Connection
Set Con =Server.CreateObject("ADODB.Connection")
Con.Open "DRIVER={SQL
Server};SERVER=DBSERVER;UID=USER;PWD=PASSWORD;DATABASE=Moni
tor"

sqlString = "EXEC pEventInfo"

SET RS = Con.Execute( sqlString )
%>
<html>
<head><title>Events</title></head>
<body>
<h1>Events</h1>
<%
IF RS.EOF THEN
%>
<b>There are no events in the database.</b>
```

```
<%
ELSE
%>
<table cellpadding=4 cellspacing=0 border=1>
<tr>
  <td bgcolor=DDDDDD>
    <b>Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Last Event Reported</b>
  </td>
  <td bgcolor=DDDDDD align=right>
    <b>Total Number of Events</b>
  </td>
</tr>
<%
WHILE NOT RS.EOF
IF RS( "EventPriority" ) = 1 THEN
    BG = "FF8888"
ELSEIF RS( "EventPriority" ) = 2 THEN
    BG = "FFFF88"
ELSEIF RS( "EventPriority" ) = 3 THEN
    BG = "88FF88"
ELSE
    BG = "FFFFFF"
END IF
%>
<tr>
  <td bgcolor=<%= BG %>>
    <a href="EventDetails.asp?ids_event=<%=RS(
"OrigEventName" )%>">
    <%=RS( "OrigEventName" )%></a>
  </td>
  <td bgcolor=<%= BG %>>
    <%=RS( "LastEvent" )%>
  </td>
  <td bgcolor=<%= BG %> align=right>
   <%=RS( "EventCount" )%>
  </td>
</tr>
<%
RS.MoveNext
WEND
%>
```

```
<tr>
</table>
<%
END IF
%>

</body>
</html>
```

Web Page - EventDetails.asp

This displays the most recent events, up to 100, for the event selected. The events are color coded by priority. Low is green. Medium is yellow. High is red.

```
<%
ids_event = TRIM( Request( "ids_event" ) )
%>

<%
' Open the Database Connection
Set Con =Server.CreateObject("ADODB.Connection")
Con.Open "DRIVER={SQL
Server};SERVER=DBSERVER;UID=USER;PWD=PASSWORD;DATABASE=Moni
tor"

sqlString = "EXEC pEventLast100Details '" & ids_event & "'"

SET RS = Con.Execute( sqlString )
%>
<html>
<head><title>Last 100 events named <%= ids_event
%></title></head>
<body>
<h1>Last 100 events named <%= ids_event %></h1>
<%
IF RS.EOF THEN
%>
<b>There are no events named <%= ids_event %></b>
<%
ELSE
%>
<table cellpadding=4 cellspacing=0 border=1>
<tr>
  <td bgcolor=DDDDDD>
    <b>Local Time</b>
```

```
  </td>
  <td bgcolor=DDDDDD>
    <b>Sensor Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>User Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Message</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Source IP</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Source Port</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Destination IP</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Destination Port</b>
  </td>
</tr>
<%
WHILE NOT RS.EOF
IF RS( "EventPriority" ) = 1 THEN
    BG = "FF8888"
ELSEIF RS( "EventPriority" ) = 2 THEN
    BG = "FFFF88"
ELSEIF RS( "EventPriority" ) = 3 THEN
    BG = "88FF88"
ELSE
    BG = "FFFFFF"
END IF
%>
<tr>
  <td bgcolor=<%= BG %>>
  <%=RS( "LocalTime" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "SensorName" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "UserName" )%>
```

```
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "Message" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DisplaySrcIPAddress" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "SrcPort" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DisplayDestIPAddress" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DestPort" )%>
  </td>
</tr>
<%
RS.MoveNext
WEND
%>
</table>
<%
END IF
%>

</body>
</html>
```

Web Page - Ports.asp

This displays the top 10 destination ports that have been logged in the RealSecure system and for each port, the last time an event was logged and the total number of events.

```
<%
' Open the Database Connection
Set Con =Server.CreateObject("ADODB.Connection")
Con.Open "DRIVER={SQL
Server};SERVER=DBSERVER;UID=USER;PWD=PASSWORD;DATABASE=Moni
tor"

sqlString = "EXEC pPortInfo"

SET RS = Con.Execute( sqlString )
```

```
%>
<html>
<head><title>Top 10 Ports</title></head>
<body>
<h1>Top 10 Ports</h1>
<%
IF RS.EOF THEN
%>
<b>There are no events for any port.</b>
<%
ELSE
%>
<table cellpadding=4 cellspacing=0 border=1>
<tr>
  <td bgcolor=DDDDDD>
    <b>Number</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Last Event Reported</b>
  </td>
  <td bgcolor=DDDDDD align=right>
    <b>Total Number of Events</b>
  </td>
</tr>
<%
WHILE NOT RS.EOF
%>
<tr>
  <td align=right>
    <a href="PortEvents.asp?port_number=<%=RS( "DestPort"
)%>">
    <%=RS( "DestPort" )%></a>
  </td>
  <td>
    <%=RS( "DestPortName" )%>
  </td>
  <td>
    <%=RS( "LastEvent" )%>
  </td>
  <td align=right>
   <%=RS( "EventCount" )%>
```

```
  </td>
</tr>
<%
RS.MoveNext
WEND
%>
<tr>
</table>
<%
END IF
%>

</body>
</html>
```

Web Page - PortEvents.asp

This displays the most recent events, up to 100, for the port selected. The events are color coded by priority. Low is green. Medium is yellow. High is red.

```
<%
port_number = TRIM( Request( "port_number" ) )
%>

<%
' Open the Database Connection
Set Con =Server.CreateObject("ADODB.Connection")
Con.Open "DRIVER={SQL
Server};SERVER=DBSERVER;UID=USER;PWD=PASSWORD;DATABASE=Moni
tor"

sqlString = "EXEC pPortLast100Events '" & port_number & "'"

SET RS = Con.Execute( sqlString )
%>
<html>
<head><title>Last 100 events for Port <%= port_number
%></title></head>
<body>
<h1>Last 100 events for Port <%= port_number %></h1>
<%
IF RS.EOF THEN
%>
<b>There are no events for <%= port_number %></b>
<%
```

```
ELSE
%>
<table cellpadding=4 cellspacing=0 border=1>
<tr>
  <td bgcolor=DDDDDD>
    <b>Local Time</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Sensor Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Event Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>User Name</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Message</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Source IP</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Source Port</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Destination IP</b>
  </td>
  <td bgcolor=DDDDDD>
    <b>Destination Port</b>
  </td>
</tr>
<%
WHILE NOT RS.EOF
IF RS( "EventPriority" ) = 1 THEN
    BG = "FF8888"
ELSEIF RS( "EventPriority" ) = 2 THEN
    BG = "FFFF88"
ELSEIF RS( "EventPriority" ) = 3 THEN
    BG = "88FF88"
ELSE
    BG = "FFFFFF"
END IF
%>
```

```
<tr>
  <td bgcolor=<%= BG %>>
  <%=RS( "LocalTime" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "SensorName" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "OrigEventName" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "UserName" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "Message" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DisplaySrcIPAddress" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "SrcPort" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DisplayDestIPAddress" )%>
  </td>
  <td bgcolor=<%= BG %>>
  <%=RS( "DestPort" )%>
  </td>
</tr>
<%
RS.MoveNext
WEND
%>
</table>
<%
END IF
%>

</body>
</html>
```

Conclusion

While RealSecure is robust when it comes to configuration, expandability, and customization of rules, signatures, and actions, it lacks a flexible reporting mechanism. It does have reporting capabilities, but they are fairly limited and inflexible. Knowing how the data is stored in the database allows the analysts to access the data directly, thus enabling them to quickly analyze and filter the data according to their needs.

References

Internet Security Solutions. "RealSecure Workgroup Manager User Guide." Version 6.5. URL: http://documents.iss.net/literature/RealSecure/RS_WGM_UG_6.5.pdf (2 Apr. 2002).

Nexus Consortium Inc. "Internet Security Systems." URL: http://www.nexusnet.com/ids.htm (2 Apr. 2002)

Wassom, Darrin. "Intrusion Detection Systems: An Overview of RealSecure." 27 Sep. 2001. URL: http://rr.sans.org/intrusion/IDS2.php (2 Apr. 2002)

Marshall, Geoff. "RealSecure: Version: 6.0." Feb. 2002. URL: http://www.scmagazine.com/scmagazine/sc-online/2002/review/07/product.html (2 Apr. 2002)

Euclidean Consulting. "Top 10 Target Ports." URL: http://www.dshield.org/topports.html (2 Apr. 2002)

Microsoft. "Internet Info Services 5.0 Support Center." URL: http://support.microsoft.com/default.aspx?xmlid=fh%3BEN-US%3Biis50 (2 Apr. 2002)

Microsoft. "SQL Server Support Center." URL: http://support.microsoft.com/default.aspx?xmlid=fh%3BEN-US%3Bsql (2 Apr. 2002)

Buyens, Jim. Web Database Development Step by Step. Microsoft Press, 3 May 2000.

Assignment 2: Network Detects

About the Logs

The format for the logs used in this section is as follows:

```
date time [name_of_attack (ids_signature_number)] protocol
     source_IP:source_port -> dest_IP:dest_port
     details if they exist
```

I created the format to loosely resemble a snort log. The data originally comes from Cisco Secure Policy Manager (Version 2.3.3i) network sensors. The alerts were exported into a comma separated value format from the CSPM database using Cisco's cvtnrlog.exe utility. Signature name information was taken from the sig.data file and combined with the resulting .csv file which was then filtered with grep, cut, and awk to produce the final output.

Detect #1: Half-open SYN attack

```
2002/03/04 17:35:51 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:54678 -> MY.NET.19.178:25

2002/03/04 23:37:34 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:45462 -> MY.NET.19.178:25

2002/03/05 00:51:00 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:52440 -> MY.NET.19.178:25

2002/03/05 01:11:00 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:54807 -> MY.NET.19.178:25

2002/03/05 01:40:55 [Half-open SYN attack (3050)] TCP/IP
     204.193.93.30:1365 -> MY.NET.19.178:25

2002/03/05 18:33:48 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:25303 -> MY.NET.19.178:25

2002/03/05 19:10:20 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:32023 -> MY.NET.19.178:25

2002/03/05 19:20:29 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:34136 -> MY.NET.19.178:25

2002/03/05 20:05:36 [Half-open SYN attack (3050)] TCP/IP
```

```
     206.126.4.130:41111 -> MY.NET.19.178:25

2002/03/05 23:30:21 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:4760 -> MY.NET.19.178:25

2002/03/06 02:41:31 [Half-open SYN attack (3050)] TCP/IP
     206.126.4.130:18071 -> MY.NET.19.178:25
```

1. Source of Trace
This was taken from my employer's network.

2. Detect was generated by
Cisco Secure Policy Manager (Version 2.3.3i) generated these alerts. The network node that reported them monitors an external segment of the network. See the "About the Logs" section at the beginning of this assignment for more information about the logs.

3. Probability the source address was spoofed
The probability is high. CERT Advisory CA-1996-21 states, "Creating half-open connections is easily accomplished with IP spoofing." The initial SYN is the only packet that needs to be sent, so any IP can be crafted for the packet since the client is not expecting any reply. If it was used as part of a Denial of Service (DoS) attack, then the attack was poor since the alerts are spaced out quite a bit.

4. Description of attack
Attack against any of the well-known service ports such as TCP port 21 FTP, TCP port 23 Telnet, TCP port 25 SMTP, and TCP port 80 WWW. Its CVE number is CVE-1999-0116 (CVE Version: 20020309).

5. Attack mechanism
The client starts the 3-way handshake by sending an initial SYN, but after the server replies with a SYN-ACK, the client does not reply with an ACK. This is commonly used for DoS attacks because almost every company has a web presence and thus allows TCP port 80 traffic into at least one machine. In addition, they most certainly communicate via e-mail which leaves TCP port 25 open.

6. Correlations
This traffic occurred very sporadically over and therefore difficult to trace.

7. Evidence of active targeting
This was actively targeted at the company's mail server—specifically at just the SMTP service.

8. Severity
Criticality: Since the mail server happens to also be the DNS server, it gets a 5.

Lethality: Although a DoS or total lockout could occur, that doesn't appear to be the case this time, so it gets a 3.

System Countermeasures: Although the system is older, it is reasonably current with patches and lockdowns, so it gets a 4.

Network Countermeasures: Unfortunately the server also acts as the perimeter firewall, so it gets a 4 since the firewall duties aren't on a separate server.

(Critical + Lethal) – (System + Net Countermeasures) = Severity

(5 + 3) – (4 + 4) = 0

9. Defensive recommendation
Move the e-mail and DNS services off the firewall and on to a server in a DMZ. Having the firewall perform all these functions leaves the network too open for

10. Multiple choice test question
Which of the following characteristics is exhibited by a Half-open SYN attack?

a) The attack is directed at the well-known UDP ports.
b) The attack is directed at the well-known TCP ports.
c) The attack is directed at both the well-known UDP and TCP ports.
d) The attack is directed at the ephemeral TCP ports.

Answer: b

11. References
Carnegie Mellon Software Engineering Institute CERT Coordination Center. "CERT® Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks." URL: http://www.cert.org/advisories/CA-1996-21.html (2 Apr. 2002)

Cisco Systems Inc. "Network Security Database, Exploit Signature, Half-open SYN Attack." URL: http://<local_cspm_server>:8080/nsdb/html/expsig_3050.html (only locally accessible)

The Mitre Corporation. "CVE-1999-0116." URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0116 (2 Apr. 2002)

Detect #2: Windows RedButton

```
2002/03/02 01:57:04 [Windows RedButton (3307)] TCP/IP
     PRV.NET.24.216:1154 -> PRV.NET.56.75:139

2002/03/01 23:10:26 [Windows RedButton (3307)] TCP/IP
```

```
     PRV.NET.24.216:1154 -> PRV.NET.56.75:139

2002/03/02 03:46:29 [Windows RedButton (3307)] TCP/IP
     PRV.NET.24.183:2132 -> PRV.NET.56.75:139

2002/03/02 03:54:42 [Windows RedButton (3307)] TCP/IP
     PRV.NET.24.183:1374 -> PRV.NET.56.73:139

2002/03/02 20:08:42 [Windows RedButton (3307)] TCP/IP
     PRV.NET.32.191:2066 -> PRV.NET.56.75:139

2002/03/02 17:56:50 [Windows RedButton (3307)] TCP/IP
     PRV.NET.6.65:1163 -> PRV.NET.56.124:139

2002/03/02 20:43:26 [Windows RedButton (3307)] TCP/IP
     PRV.NET.6.65:1163 -> PRV.NET.56.124:139

2002/03/02 22:31:05 [Windows RedButton (3307)] TCP/IP
     PRV.NET.36.68:2698 -> PRV.NET.56.75:139

2002/03/02 19:44:29 [Windows RedButton (3307)] TCP/IP
     PRV.NET.36.68:2698 -> PRV.NET.56.75:139

2002/03/03 18:17:22 [Windows RedButton (3307)] TCP/IP
     PRV.NET.24.57:1578 -> PRV.NET.56.75:139

2002/03/03 18:57:22 [Windows RedButton (3307)] TCP/IP
     PRV.NET.24.78:2095 -> PRV.NET.56.75:139
```

1. Source of Trace

This was taken from my employer's network.

2. Detect was generated by

Cisco Secure Policy Manager (Version 2.3.3i) generated these alerts. The network node that reported them monitors an internal segment of the network. See the "About the Logs" section at the beginning of this assignment for more information about the logs.

3. Probability the source address was spoofed

The probability is very low since this detect occurred within the corporate network and the source IP's were identified to belong to system administrators.

4. Description of attack
Attack against the registry of a Windows NT machine that could potentially compromise the password file.

5. Attack mechanism
The RedButton tool takes advantage of a flaw in Windows NT 3.51 and 4.0 (pre-Service Pack 3) which allows remote access of the system registry. This would allow someone to create a Windows Share without having the appropriate rights. If the registry was configured to share the correct folder(s), the next time the machine was rebooted, those folders would allow access by Everyone. This would leave the password file open for download.

However the CSPM Network Security Database asserts, "There is a theoretical possibility that a combination of one or more network, system, or connection management tools could make a series of network accesses that would trigger this signature."

6. Correlations
The attack originated from several different machines within the LAN. These machines were traced back to system administrators.

7. Evidence of active targeting
Only a few machines were targeted and they all perform similar roles. However, since the source was identified to be internal system administrators, it is unlikely that any real targeting was occurring.

8. Severity
Criticality: Since the servers are Domain Controllers, it gets a 5.

Lethality: If the attack was effectively used and if password requirements were weak, root access is possible, so it gets a 5.

System Countermeasures: The systems are well patched for this attack, so it gets a 5.

Network Countermeasures: Because all the traffic was within the core local area network, there was no firewall protection, so it gets a 1.

(Critical + Lethal) – (System + Net Countermeasures) = Severity

(5 + 5) – (5 + 1) = 4

9. Defensive recommendation
Adding a firewall is an option, but I would rather turn off all file sharing on the domain controllers or whatever machines are targeted.

10. Multiple choice test question
Which of the following is the best defense against a Windows RedButton attack?

a) Turn off Windows File and Printer sharing on the target machines.
b) Isolate the target machines behind a firewall.
c) Ensure the target machines are well monitored by an Intrusion Detection System.
d) Put all the target machines on the same network switch.

Answer: a

11. References

Cisco Systems Inc. "Network Security Database, Exploit Signature, Windows Redbutton Attack." URL: http://<local_cspm_server>:8080/nsdb/html/expsig_3307.html (only locally accessible)

Detect #3: WWW iPlanet shtml Buffer Overflow

```
2002/03/03 00:11:03 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.63:2220 -> MY.NET.168.10:80

2002/03/03 02:05:48 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.42:1983 -> MY.NET.168.10:80

2002/03/03 02:06:47 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.42:2047 -> MY.NET.168.10:80

2002/03/03 17:09:55 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.42:1110 -> MY.NET.168.10:80

2002/03/03 18:26:05 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.36.97:1104 -> MY.NET.168.10:80

2002/03/03 18:33:17 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.36.97:1225 -> MY.NET.168.10:80

2002/03/03 18:39:17 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.36.97:1344 -> MY.NET.168.10:80
```

```
2002/03/03 18:49:12 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.36.97:1399 -> MY.NET.168.10:80

2002/03/03 19:44:57 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.63:1142 -> MY.NET.168.10:80

2002/03/03 21:46:40 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.63:1634 -> MY.NET.168.10:80

2002/03/03 22:31:05 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.42:1308 -> MY.NET.168.10:80

2002/03/04 01:11:11 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.63:2653 -> MY.NET.168.10:80

2002/03/04 02:11:50 [WWW iPlanet shtml Buffer Overflow
(5121)] TCP/IP
     PRV.NET.26.63:3200 -> MY.NET.168.10:80
```

1. Source of Trace

This was taken from my employer's network.

2. Detect was generated by

Cisco Secure Policy Manager (Version 2.3.3i) generated these alerts. The network node that reported them monitors an internal segment of the network. See the "About the Logs" section at the beginning of this assignment for more information about the logs.

3. Probability the source address was spoofed

The probability is low. This alert was generated by traffic between internal workstations and the corporate proxy server. Looking at the proxy logs shows web sites being accessed with very long web addresses. These are advertising sites that contain extra "referring site" information for billing purposes.

4. Description of attack

Cisco states, "This signature triggers if a request with more than 180 characters between slashes (/ or) is received with a .shtml suffix"

5. Attack mechanism
This takes advantage of a buffer overflow problem with the .shtml logging, which could allow arbitrary code to be run on the target machine.

6. Correlations
Comparing the alerts with the proxy server logs, it can be determined that although the web addresses do match the criteria for triggering the alert, they are not malicious.

7. Evidence of active targeting
There is no evidence of active targeting since all web traffic passes through the proxy server.

8. Severity
Criticality: The business won't come to a halt if there are problems with the proxy server, but since it would affect the entire company it gets a 3.

Lethality: If it was not a false positive some potential damage could be done, so it gets a 4.

System Countermeasures: The server is well maintained and up to date with patches but not necessarily fully hardened, so it gets a 4.

Network Countermeasures: It is not part of any domain and windows file and printer sharing is turned off, so it gets a 4.

(Critical + Lethal) – (System + Net Countermeasures) = Severity

(3 + 4) – (4 + 4) = –1

9. Defensive recommendation
As these were false positives, and as sufficient countermeasures are in place, the only countermeasures would be to facilitate correlation between the network sensors and the proxy log.

10. Multiple choice test question
Which of the following resources could be used to correlate the iPlanet .shtml Buffer Overflow alerts reported by the network sensor?

a) Firewall logs.
b) Web proxy logs from the target server.
c) tcpdump logs from the target server.
d) All of the above.

Answer: d

11. References

Cisco Systems Inc. "Network Security Database, Exploit Signature, iPlanet .shtml Buffer Overflow." URL: http://<local_cspm_server>:8080/nsdb/html/expsig_5121.html (only locally accessible)

The Mitre Corporation. "CVE-2000-1077." URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-1077 (2 Apr. 2002)

Detect #4: Qmail Length Crash

```
2002/03/02 06:16:25 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:49312 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 06:16:34 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:49312 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 06:16:37 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:49312 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 06:16:41 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:49312 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 06:16:51 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:49312 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 21:40:24 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:36862 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 21:40:27 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:36862 -> MY.NET.19.178:25
     Qmail Long Command Crash

2002/03/02 21:40:29 [Qmail Length Crash (3109)] TCP/IP
     155.251.246.228:36862 -> MY.NET.19.178:25
     Qmail Long Command Crash
```

1. Source of Trace
This was taken from my employer's network.

2. Detect was generated by
Cisco Secure Policy Manager (Version 2.3.3i) generated these alerts. The network node that reported them monitors an external segment of the network. See the "About the Logs" section at the beginning of this assignment for more information about the logs.

3. Probability the source address was spoofed
The probability is very low. Since mail services require a 3-way handshake before exchanging any data, the source is who it claims to be.

4. Description of attack
This is a buffer overflow attack against a mail server.

5. Attack mechanism
By sending a long command or list of recipients to a qmail server, all of the server's memory will be utilized causing the qmail server to crash.

6. Correlations

7. Evidence of active targeting
It is active targeting since the traffic is directed at the mail server.

8. Severity
Criticality: Since the mail server happens to also be the DNS server, it gets a 5.

Lethality: If in fact we were running qmail on the mail server it would be a 4 since this attack shuts down the server, but as we are running sendmail, the server won't be affected, so it gets a 1.

System Countermeasures: Even though the system is older but relatively up to date with patches, since it's not running qmail and is impervious to this attack, it gets a 5.

Network Countermeasures: Unfortunately the server also acts as the perimeter firewall, so it gets a 4 since the firewall duties aren't on a separate server.

(Critical + Lethal) – (System + Net Countermeasures) = Severity

(5 + 1) – (5 + 4) = –3

9. Defensive recommendation
If it were running qmail, then making sure it was running the latest version of qmail would be the best recommendation.

10. Multiple choice test question
A Qmail length crash attempts to harm your mail server by?

a) Sending a buffer overflow that would execute malicious code.
b) Sending a long command or list of recipients which causes the Qmail process to utilize all the memory.
c) Set all the TCP flags on the packets containing the e-mail.
d) Send the data to port different from the SMTP port (25) causing the system to crash.

Answer:

11. References
Cisco Systems Inc. "Network Security Database, Vulnerability, Qmail Command Length Crash." URL: http://<local_cspm_server>:8080/nsdb/html/vul_1421.html (only locally accessible)

Cisco Systems Inc. "Network Security Database, Exploit Signature, Q-Mail Length Crash." URL: http://<local_cspm_server>:8080/nsdb/html/expsig_3109.html (only locally accessible)

The Mitre Corporation. "CAN-1999-0250 (under review)." URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0250 (2 Apr. 2002)

Detect #5: Suspicious Mail Attachment

```
2002/03/05 08:17:53 [Suspicious Mail Attachment (3110)]
TCP/IP
     MY.NET.19.178:35298 -> 65.218.171.4:25
     Suspicious Mail Attch: virgin.scr"

2002/03/06 02:23:03 [Suspicious Mail Attachment (3110)]
TCP/IP
     24.25.227.33:3070 -> MY.NET.19.178:25
     Suspicious Mail Attch: filename="using.pif"

2002/03/06 05:31:57 [Suspicious Mail Attachment (3110)]
TCP/IP
     24.25.227.35:4436 -> MY.NET.19.178:25
     Suspicious Mail Attch: Emulator.scr"
```

1. Source of Trace
This was taken from my employer's network.

2. Detect was generated by
Cisco Secure Policy Manager (Version 2.3.3i) generated these alerts. The network node that reported them monitors an external segment of the network. See the "About the Logs" section at the beginning of this assignment for more information about the logs.

3. Probability the source address was spoofed
The probability is low.

4. Description of attack
A file with a questionable file extension is attached to the mail message—it may be an executable or script.

5. Attack mechanism
Malicious code is attached to an e-mail. When people open the attachment, it could either be an executable that runs or a script that the operation system will interpret and run. This is the primary method for worm propagation throughout the Internet. The code typically includes a routine that will mail itself out to everyone listed in the recipients address book.

This can cause harm by simply resending itself. This will clog the mail server causing delays in mail delivery and it can consume a lot of the Internet bandwidth and thus slow down all traffic.

Of course code can be included to modify or delete files on the user's local machine as well, which poses an additional threat.

6. Correlations
There are a number of different worms still roaming the Internet today, with Nimda and Code Red as some of the more infamous ones. McAfee reports that W32/Hybris.gen@MM was discovered on 10/16/2000 and may contain an attachment called sexy virgin.scr.

7. Evidence of active targeting
Internet worms vary in how targeted they are, but for the most part, they are not targeted. They spread to as many e-mail addresses as they can when activated.

8. Severity
Criticality: Since the mail server happens to also be the DNS server, it gets a 5.

Lethality: Since it could potentially shut down the mail server, it gets a 5.

System Countermeasures: Although the system is older, is reasonably current with patches and lockdowns, so it gets a 4.

Network Countermeasures: Unfortunately the server also acts as the perimeter firewall, but since any suspicious attachment gets quarantined at the mail server, any worms will not spread, so it gets a 5

(Critical + Lethal) – (System + Net Countermeasures) = Severity

(5 + 5) – (4 + 5) = 1

9. Defensive recommendation

Since instituting quarantine for suspicious attachments there have been no problems with worms. The following is the list of file types that are considered suspicious.

File Extension	File Type
.ade	Microsoft Access project extension
.adp	Microsoft Access project
.asx	Windows Media Audio or Video shortcut
.bas	Visual Basic class module
.bat	Batch file
.chm	Compiled HTML Help file
.cmd	Windows NT Command script
.com	MS-DOS program
.cpl	Control Panel extension
.crt	Security certificate
.exe	Program
.hlp	Help file
.hta	HTML program
.inf	Setup Information
.ins	Internet Naming Service
.isp	Internet Communication settings
.js	JScript Script file
.jse	Jscript Encoded Script file
.mda	Microsoft Access add-in program
.mde	Microsoft Access MDE database
.mdz	Microsoft Access wizard program
.msc	Microsoft Common Console document
.msi	Windows Installer package
.msp	Windows Installer patch
.mst	Visual Test source files
.pcd	Photo CD image
.pif	Shortcut to MS-DOS program
.prf	Microsoft Outlook profile settings
.reg	Registration entries
.scf	Windows Explorer command

File Extension	File Type
.scr	Screen saver
.sct	Windows Script Component
.shb	Shell Scrap Object <http://pc-help.org/security/scrap.htm>
.shs	Shell Scrap Object <http://pc-help.org/security/scrap.htm>
.vb	VBScript file
.vbe	VBScript encoded script file
.vbs	Visual Basic Script file
.wsc	Windows Script Component
.wsf	Windows Script file
.wsh	Windows Script Host Settings file

10. Multiple choice test question
What is a reasonable and effective way to handle mail worms and viruses, including new variations?

a) Keep your mail server in a DMZ.
b) Make sure your virus software is up to date.
c) Configure your mail server to quarantine all e-mail with attachments that are potentially dangerous.
d) Do not allow your mail server to act as a relay server.

Answer: c

11. References
Cisco Systems Inc. "Network Security Database, Vulnerability, SMTP VBScript Mail Attachments." URL: http://<local_cspm_server>:8080/nsdb/html/vul_14125.html (only locally accessible)

Cisco Systems Inc. "Network Security Database, Exploit Signature, Suspicious Mail Attachment." URL: http://<local_cspm_server>:8080/nsdb/html/expsig_3110.html (only locally accessible)

McAfee Security. "Avert". URL: http://vil.nai.com/vil/content/v_98873.htm (2 Apr. 2002)

Assignment 3: "Analyze This" Scenario

Executive Summary

I have been asked to analyze 5 days' worth of snort IDS data collected by the University. The dates that have been chosen are December 22, 2001 through December 26, 2001. The files that were retrieved originally had a .gz extension in place of the .clean extension, or were simply added on to the out of spec filenames. The actual log files that were analyzed are:

Alert Files	Scan Files	Out of Spec Files
alert.011222.clean	scans.011222.clean	oos_Dec.22.2001
alert.011223.clean	scans.011223.clean	oos_Dec.23.2001
alert.011224.clean	scans.011224.clean	oos_Dec.24.2001
alert.011225.clean	scans.011225.clean	oos_Dec.25.2001
alert.011226.clean	scans.011226.clean	oos_Dec.26.2001

The alert files cover the entire 5 days and contain events that the IDS captured and successfully identified as possible attacks. This is determined from the "signature" of each packet. Most packets are normal traffic, don't match any signature, and are therefore ignored. The packets that match a signature are logged to an alert file. For example, packets that have the ACK flag set along with the URI portion of a request that contains "get //", would trigger the "WEB-MISC prefix-get //" signature.

Over this five-day period, the IDS identified 127 different signatures. The hour that registered the most alerts was 26 Dec 2001 between 06:00 and 07:00. However, there was more interesting traffic on 25 Dec 2001 between 21:00 and 23:00. For most of the hour-long blocks, the number of scans was greater than the number of alerts, but during these two hours the number of alerts was greater than the scans. Also, the number of out of spec events was several orders of magnitude greater than at all other times. These two hours will be covered in detail later.

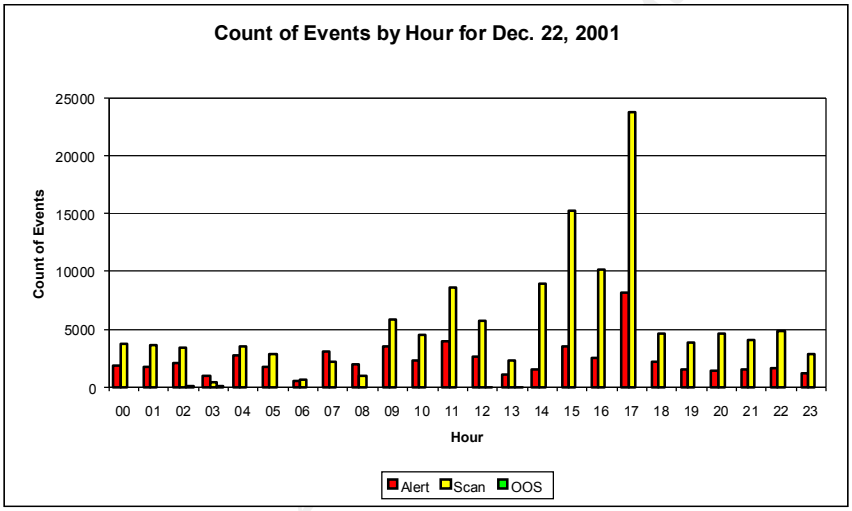
The scan files cover the entire 5 days and contain logs recording anomalous activity from one-to-many and many-to-many machines and/or between two machines but over a wide range of ports. These logs also show that the most activity originated from within the home network. This can be explained by either an internal person gathering information about external machines that have set off alerts, or it could be explained by a compromised internal machine that is being used to attack external machines.

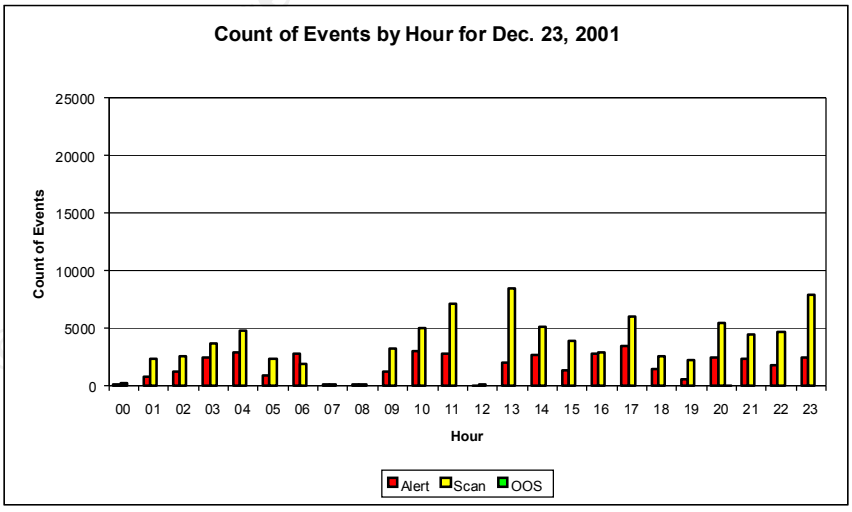
The out of spec files only cover the first 4 days. It's possible that because of the extremely high number of out of spec events at the end of the fourth day, something got triggered to stop the logging on the fifth day. Other than the

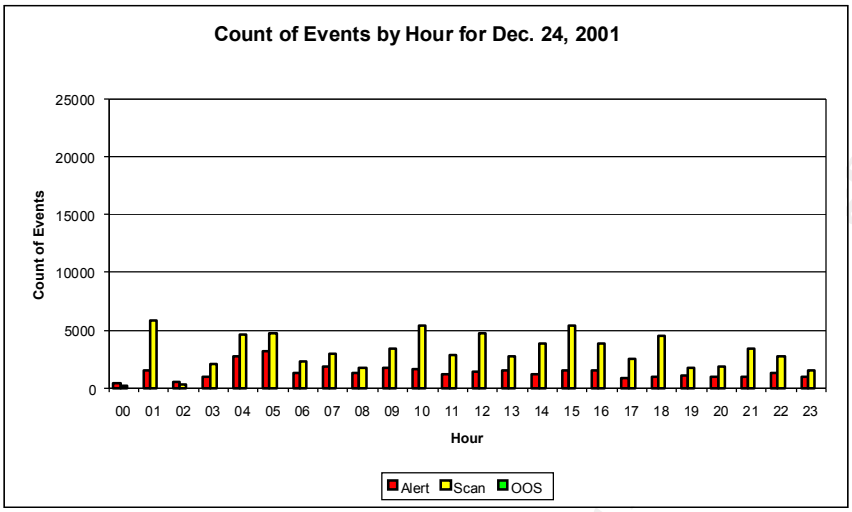
interesting two hours, the bulk of the out of spec events were either slow port scans in an attempt to have OS fingerprinting go unnoticed, some type of fragmentation attack, or attacks against http and smtp ports.

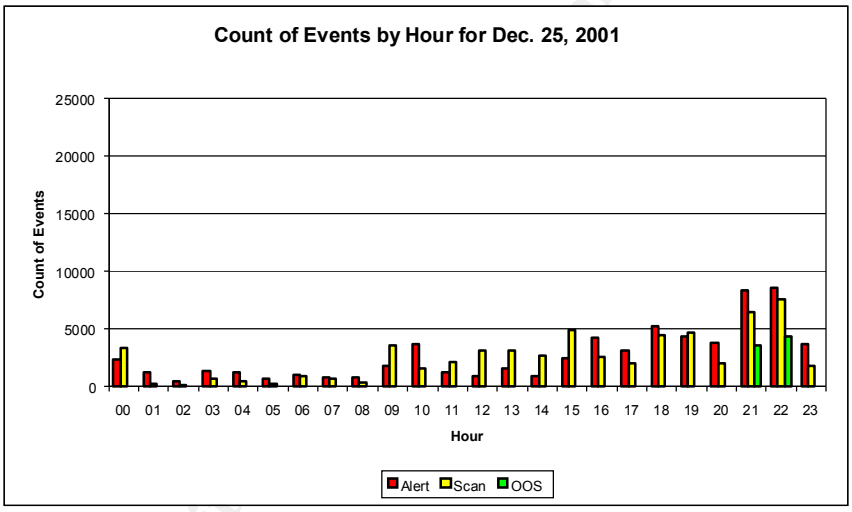
The amount of activity is significant. If the network were properly secured with firewalls and turning off unnecessary services or daemons, the numbers of alerts could be easily reduced.

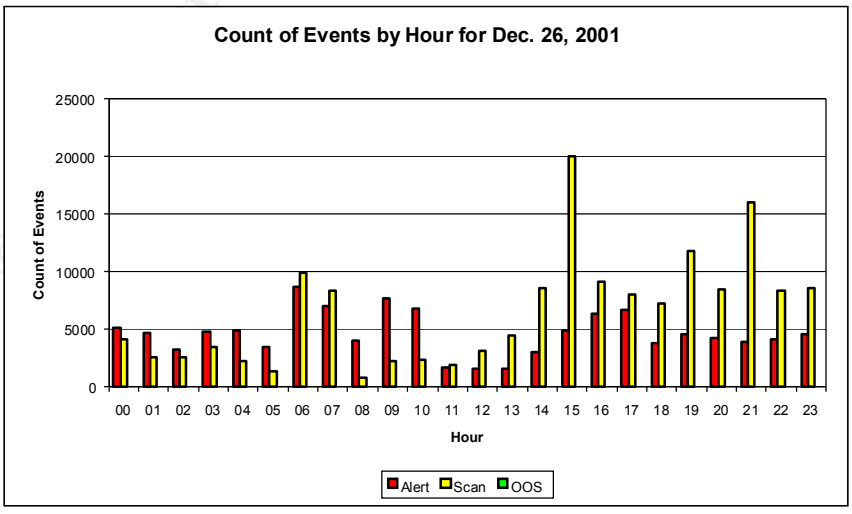
The following charts show the number of each type of event (alert, scan, oos) on an hourly basis. There is one chart for each day analyzed. They have been normalized such that each day's chart has the same maximum value so you can readily see how the amount of traffic compares from one day to the next.











List of Top Most Numerous Detects

Signature	# Alerts	# Sources	# Dests	Description
Watchlist 000220 IL-ISDNNET-990517	62318	22	13	This is most likely either a peer-to-peer file-sharing program such as Kazaa, Morpheus, etc. or a variant of Code Red that uses 1214 as the dest port. There were many different source and destination machines that logged this alert, but the primary source of this alert (61295 alerts) came from source machine 212.179.35.118 on port 60339 with the destination machine MY.NET.70.70 on port 1214. Correlations: http://www.giac.org/practical/Rick_Yuen_GCIA.doc http://www.giac.org/practical/REUBEN_RUBIO_GCIA.doc http://www.giac.org/practical/Ben_Thomas_GCIA.doc http://seifried.org/security/ports/1000/1214.html

Signature	# Alerts	# Sources	# Dests	Description
MISC traceroute	32793	67	7	This indicates that a traceroute from a UNIX based host or tracert from a Windows based host was run against the network. This is used to probe the network infrastructure to determine how deep into the network the destination machine resides. This was fairly evenly distributed among most of the 67 source machines. However, the primary destination (32492 alerts) was MY.NET.140.9. There were 6 other destination machines that logged this alert. Correlations: http://www.giac.org/practical/Stan_Hoffman_GCIA.doc http://www.giac.org/practical/Jeff_Zahr_GCIA.doc http://www.giac.org/practical/Mike_Poor_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
CS WEBSERVER - external web traffic	18080	3438	1	This reports web traffic from an external machine coming into the network. It originated from many source machines but was only reported by one destination machine. That would either indicate that the rule was defined only on the destination machine (MY.NET.100.165) or that the destination machine was the only one exposed to external web traffic. Since 39 internal machines logged web traffic from external machines, this rule must be specific to MY.NET.100.165. Correlations: http://www.giac.org/practical/Stan_Hoffman_GCIA.doc http://www.giac.org/practical/Mike_Poor_GCIA.doc http://www.giac.org/practical/Edward_Peck_GCIA.doc http://www.iana.org/assignments/port-numbers

Signature	# Alerts	# Sources	# Dests	Description
MISC source port 53 to <1024	16955	4019	8	As the rule indicates, the source port is 53 (DNS request). Of the 8 destination machines, 3 of them combined received 16140 of the alerts and the distribution among them was fairly even with MY.NET.1.3 slightly ahead of MY.NET.1.4 and MY.NET.1.5. These 3 machines are like the DNS servers with MY.NET.1.3 being the primary one. Therefore, the bulk of this activity is normal DNS requests. Correlations: http://www.giac.org/practical/Stan_Hoffman_GCIA.doc http://www.giac.org/practical/Jeff_Zahr_GCIA.doc http://www.giac.org/practical/Mike_Poor_GCIA.doc http://www.iana.org/assignments/port-numbers

Signature	# Alerts	# Sources	# Dests	Description
ICMP Echo Request BSDtype	11550	19	9	The Whitehats site (http://www.whitehats.com/IDS/152) gives this fine description, "This event indicates that a ping request was sent to your network. Ping requests are usually used to determine whether a host is responsive, but can be misused to map your network. This particular ping was probably generated by BSD/OS, FreeBSD, NetBSD, OpenBSD 2.5, Linux, or Solaris 2.5-2.7. This event is specific to a particular exploit and is detected based on a particular string of characters found in the packet payload. Signatures for this event are very specific. " The majority of these alerts (6257 total) came from 128.223.4.21, 141.213.11.120, 147.46.59.144 with MY.NET.70.148 as the destination. However, there were 1757 alerts logged as originating from MY.NET.60.39 and destined to 24.180.204.24. Correlations: http://www.giac.org/practical/Mike_Poor_GCIA.doc http://www.giac.org/practical/Edward_Peck_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
INFO MSN IM Chat data	10305	145	195	This alert is based on a rule that looks for specific data in the payload of the packet that would identify it as MSN Instant Messenger Chat traffic. It would appear this implementation of the rule does not look for a specific destination port since there are 318 different destination ports. However the majority of the alerts logged (6408) do have port 1863 as the destination port and an external destination machine (all of them in the Microsoft domain), which is to be expected. The rest of the alerts are the return traffic from Microsoft to MY.NET.x.x. Although this traffic is normal, it could be used transfer confidential information outside of the internal network. Correlations: http://www.giac.org/practical/Mike_Poor_GCIA.doc http://www.giac.org/practical/Edward_Peck_GCIA.doc http://www.giac.org/practical/Stan_Hoffman_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
WEB-MISC prefix-get //	9644	571	3	This is from web traffic to port 80 that tries to gather information about the destination machine. There were only 3 destination machines. The first, MY.NET.253.114, with 9311 alerts and the second, MY.NET.253.115, with 332 alerts are very likely web servers. They also are in the same Class C address space. The third, MY.NET.140.2 only had one alert and is in a different Class C address space. This is likely suspicious traffic Correlations: http://www.giac.org/practical/Edward_Peck_GCIA.doc http://www.giac.org/practical/Stan_Hoffman_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
MISC Large UDP Packet	7748	27	4	The bulk of this traffic comes from 216.106.172.149 and destined to MY.NET.153.210 on destination ports 1434 (Microsoft SQL Monitor), 3888 (unknown), and 3872 (unknown). The Whitehats site (http://www.whitehats.com/IDS/247) says this about large (> 4K payload) UDP packets, "This event indicates that an abnormally large UDP packet was sent to your server. This may indicate a denial of service attack or the use of a covert channel. Since this event was caused by a UDP packet, the source IP address could be easily forged. Also, it has been noted that the due to the nature of this event the attacker does not normally require response traffic. In most cases this means that the event should be analyzed along with other supporting data before acting on the event." Because of the odd destination ports, this is very suspect traffic that should be investigated further. Correlations: http://sunsite.securitycentralhq.com/mirrors/security/snort/Files/Current/misc.rules http://www.giac.org/practical/Edward_Peck_GCIA.doc http://www.giac.org/practical/Stan_Hoffman_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
SCAN Proxy attempt	5753	61	4669	There were of course only two destination ports that triggered this alert, 1080 and 8080. Most proxy servers listen on 1080, so if that port is open and unsecured, an attacker can use the proxy server to hide the original source. All the source machines were external, but the scan of the internal network was very methodical as only a few machines were scanned several times. Correlations: http://www.snort.org/snort-db/sid.html?id=615 http://www.giac.org/practical/Edward_Peck_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
Queso fingerprint	5132	34	26	The Whitehats site (http://www.whitehats.com/IDS/29) reports, "This event indicates that a remote user has used the Queso tool to determine the OS fingerprint of the server." Basically, an attacker is trying to determine what operating system and version is being run a given machine. Once they have this information, they have a direction of possible exploits to use to gain control of the machine. There were a wide variety of source and destination ports and a fair number of source and destination machines. However, the main players in this alert were 206.65.191.129 for the source and MY.NET.98.177 as the destination with a total of 4510 alerts between them. Looking closer at these data, over 1400 destination ports were scanned from the same source port. Correlations: http://www.giac.org/practical/Edward_Peck_GCIA.doc http://www.giac.org/practical/Ben_Thomas_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
ICMP Source Quench	5111	25	93	ICMP Source Quench is sent from the receiving machine to the sending machine when it is having difficulty keeping up with the traffic that is being sent to it. (SANS Institute 3.1 TCP/IP for Intrusion Detection p. 4-6) The primary source of this alert is MY.NET.5.13. It is being sent out to many destinations distributed fairly evenly over most of the 5-day period. It starts on 22 Dec 2001 at 00:00 and there are alerts during most hours until 26 Dec 2001 at 22:00. Correlations: http://www.giac.org/practical/Edward_Peck_GCIA.doc
SYN-FIN scan!	5026	1	5026	This is a commonly used for port scanning and OS fingerprinting. The TCP packet that is sent has both the SYN and FIN flags set and would be a crafted packet. In this case, all the alerts were generated by the same source, 24.0.28.234. It was also picked up as a scan as well as out of spec. A closer look at the packets shows that they all have the same source and destination ports, 22 (ssh – secure shell). They also have the same IP ID and blocks of similar TCP Acknowledgement numbers. However, they do differ in that each destination is different. Correlations: http://www.giac.org/practical/Edward_Peck_GCIA.doc

Due to limited space for the analysis, only the most numerous alerts have been analyzed. The analysis stops with the top 5 external talkers' alerts. The other alerts will not be analyzed at this time.

Top Talkers

Rank	Alerts	Scans	OOS
1	212.179.35.118	MY.NET.87.50	24.0.28.234
2	216.106.172.149	MY.NET.98.203	210.125.178.52
3	24.0.28.234	211.248.231.10	199.183.24.194
4	MY.NET.5.13	65.165.14.43	64.172.24.155
5	206.65.191.129	210.77.145.30	24.36.185.188
6	65.165.14.43	210.58.102.86	141.157.92.22
7	MY.NET.60.11	204.152.184.75	211.39.150.91
8	65.207.94.30	24.44.21.206	65.165.238.50
9	128.223.4.21	24.0.28.234	202.168.254.178
10	141.213.11.120	MY.NET.84.185	213.84.157.192

(NOTE: 206.65.191.129 was number 11 for Scans)

Select External Sources

These five external IP addresses were chosen because they were the top five generators of alerts. They also were the source of the top 13 alerts. Unlike the first two, the third, fourth and fifth IP addresses all generated traffic that showed up as either a scan event, an out of spec event, or both.

To find the information I first looked up the IP address at ARIN's Whois server (http://ws.arin.net/cgi-bin/whois.pl). If there was no registrar information provided by ARIN, I went to the InterNIC's site (http://www.internic.net/whois.html) with the domain name that ARIN provided to find the registrar. Once I had the registrar information, either from ARIN or InterNic, I went to the registrar's Whois server to get the rest of the information.

The times of activity reported have been rounded to the hour.

1. *212.179.35.118*
   This machine was responsible for nearly all of the "Watchlist 000220 IL-ISDNNET-990517" alerts which started in full swing on 25 Dec 2001 at 15:00 and continued sustained alerts between 1000 and 3000 per hour until 26 Dec 2001 at 10:00.

   From ARIN Whois Server (http://ws.arin.net/cgi-bin/whois.pl)

European Regional Internet Registry/RIPE NCC (NET-RIPE-NCC-)
These addresses have been further assigned to European users.
Contact info can be found in the RIPE database, via the
WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/perl/whois/
NL

Netname: RIPE-NCC-212
Netblock: 212.0.0.0 - 212.255.255.255
Maintainer: RIPE

Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN) nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

NS.RIPE.NET	193.0.0.193
NS.EU.NET	192.16.202.11
AUTH03.NS.UU.NET	198.6.1.83
NS2.NIC.FR	192.93.0.4
SUNIC.SUNET.SE	192.36.125.2
MUNNARI.OZ.AU	128.250.1.21
NS.APNIC.NET	203.37.255.97

To search on arbitrary strings, see the Database page on
the RIPE NCC website at http://www.ripe.net/perl/whois/

Record last updated on 16-Oct-1998.
Database last updated on 23-May-2002 19:59:23 EDT.

From the Ripe Whois Database (http://www.ripe.net/perl/whois/)

% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 212.179.35.96 - 212.179.35.127
netname: EPLICATION-LTD
mnt-by: INET-MGR
descr: EPLICATION-LTD-HOSTING
country: IL
admin-c: ZV140-RIPE
tech-c: MZ4647-RIPE
status: ASSIGNED PA

notify: hostmaster@isdn.net.il
changed: hostmaster@isdn.net.il 20020312
source: RIPE

route: 212.179.0.0/17
descr: ISDN Net Ltd.
origin: AS8551
notify: hostmaster@isdn.net.il
mnt-by: AS8551-MNT
changed: hostmaster@isdn.net.il 19990610
source: RIPE

person: Zehavit Vigder
address: bezeq-international
address: 40 hashacham
address: petach tikva 49170 Israel
phone: +972 52 770145
fax-no: +972 9 8940763
e-mail: hostmaster@bezeqint.net
nic-hdl: ZV140-RIPE
changed: zehavitv@bezeqint.net 20000528
source: RIPE

person: Meron Ziv
address: Bezeq International
address: hashacham 40
address: petach tiqua
address: Israel
phone: +972-3-9257710
e-mail: hostmaster@bezeqint.net
nic-hdl: MZ4647-RIPE
changed: hostmaster@bezeqint.net 20010107
source: RIPE

2. *216.106.172.149*
This machine was responsible for the majority of the "MISC Large UDP Packet" alerts. These occurred on 22 Dec 2001 from 17:00 to 18:00 and again on 23 Dec 2001 from 16:00 to 18:00.

From ARIN Whois Server (http://ws.arin.net/cgi-bin/whois.pl)

iBEAM Broadcasting Corporation (NETBLK-IBEAM)
645 Almanor Ave., suite 100
Sunnyvale, CA 94085
US

Netname: IBEAM
Netblock: 216.106.160.0 - 216.106.175.255
Maintainer: BEAM

Coordinator:
Le, Stewart (SL895-ARIN) stle@ibeam.com
408-830-3572

Domain System inverse mapping provided by:

NS1.IBEAM.COM	204.233.70.15
NS2.IBEAM.COM	204.247.99.125

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 22-Jan-2002.
Database last updated on 23-May-2002 19:59:23 EDT.

From Network Solutions Whois server (http://www.netsol.com/cgi-bin/whois/whois)

Registrant:
iBEAM Broadcasting Corporation (IBEAM3-DOM)
645 Almanor Ave. Suite 100
Sunnyvale, CA 94086
US

Domain Name: IBEAM.COM

Administrative Contact:
Lopez, Ed (ELR41) elopez@IBEAM.COM
iBeam Broadcasting
645 Almanor Ave.
Suite 100
Sunnyvale , CA 94086
(408) 523-1633 (FAX) (408) 730-0262

Technical Contact:
iBeam Hostmaster (IH598-ORG) hostmaster@IBEAM.COM
iBeam Broadcasting
645 Almanor Ave.
Suite 100
Sunnyvale , CA 94086
US
(408) 523-1700
Fax- (408) 730-0262

Record expires on 10-Jul-2003.
Record created on 10-Jul-1998.
Database last updated on 24-May-2002 01:04:18 EDT.

Domain servers in listed order:

NS1.IBEAM.COM	204.233.70.15
NS2.IBEAM.COM	204.247.99.125
NS3.IBEAM.COM	216.106.164.9
NS4.IBEAM.COM	212.187.254.159
NS5.IBEAM.COM	216.106.167.125

3. *24.0.28.234*
This machine was responsible for the "SYN-FIN scan!" that scanned 5026 machines in MY.NET.x.x. This scan occurred on 25 Dec 2001 from 21:00 to 23:00 and accounts for the large amount of out of spec data during that time.

From ARIN Whois Server (http://ws.arin.net/cgi-bin/whois.pl)

@Home Network (NETBLK-HOME-CORP-1)
425 Broadway
Redwood City, CA 94063
US

Netname: HOME-CORP-1
Netblock: 24.0.16.0 - 24.0.31.255

Coordinator:
Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

Record last updated on 09-Apr-1998.
Database last updated on 23-May-2002 19:59:23 EDT.

From Network Solutions Whois server (http://www.netsol.com/cgi-bin/whois/whois)

Registrant:
Home Network (HOME5-DOM)
425 Broadway St.
Redwood City, CA 94063
US

Domain Name: HOME.NET

Administrative Contact:
Excite@Home Estate (QUSSRZDUDO) nic-contact@EXCITEHOME.NET

At Home Corporation
450 Broadway Street
Redwood City , CA 94063
US
650-556-5000
Fax- 650-556-5511

Technical Contact:
DNS admin (DA1596-ORG) dnsadmin@EXCITECORP.COM
Excite, Inc.
450 Broadway
Redwood City, CA 94063
US
(650) 556-5000
Fax- - - - - (650) 568-6030

Record expires on 19-May-2006.
Record created on 18-May-1995.
Database last updated on 24-May-2002 01:30:52 EDT.

Domain servers in listed order:

UDNS1.ULTRADNS.NET	204.69.234.1
UDNS2.ULTRADNS.NET	204.74.101.1

4. *206.65.191.129*
This machine was responsible for the "Queso fingerprint" as previously mentioned. This took place on 26 Dec 2001 from 02:00 to 03:00 and then later that day from 16:00 to 18:00.

From ARIN Whois Server (http://ws.arin.net/cgi-bin/whois.pl)

UUNET Technologies, Inc. (NETBLK-NETBLK-UUNETCBLK64-67)
3060 Williams Drive, Suite 601
Fairfax, Virginia 22031
US

Netname: NETBLK-UUNETCBLK64-67
Netblock: 206.64.0.0 - 206.67.255.255
Maintainer: UU

Coordinator:
UUNET Postmaster (UUPM-ARIN) postmaster@uunet.uu.net
703-206-5440

Domain System inverse mapping provided by:

AUTH00.NS.UU.NET	198.6.1.65
AUTH01.NS.UU.NET	198.6.1.81

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 26-Sep-2001.
Database last updated on 23-May-2002 19:59:23 EDT.

From Network Solutions Whois server (http://www.netsol.com/cgi-bin/whois/whois)

Registrant:
UUNET Technologies, Inc. (UU-DOM)
3060 Williams Drive Ste 601
Fairfax, VA 22031
USA

Domain Name: UU.NET

Administrative Contact, Technical Contact:
UUNET, AlterNet - Technical Support (OA12) help@UU.NET
3060 Williams Drive
Fairfax, VA 22031
+1 (800) 900-0241

Record expires on 21-May-2004.
Record created on 20-May-1987.
Database last updated on 24-May-2002 02:09:28 EDT.

Domain servers in listed order:

AUTH00.NS.UU.NET 198.6.1.65
AUTH60.NS.UU.NET 198.6.1.181
AUTH200.NS.UU.NET 195.129.12.82
AUTH210.NS.UU.NET 195.129.12.74

5. *65.165.14.43*
This machine was responsible for the majority of the "SCAN Proxy attempt" and occurred on 26 Dec 2001 from 06:00 to 08:00.

From ARIN Whois Server (http://ws.arin.net/cgi-bin/whois.pl)

Sprint (NETBLK-SPRINTLINK-2-BLKS) SPRINTLINK-2-BLKS 65.160.0.0 - 65.174.255.255
SYSTEMS SOLUTIONS INC (NETBLK-FON-110133555275610) FON-110133555275610
65.165.12.0 - 65.165.15.255

SYSTEMS SOLUTIONS INC (NETBLK-FON-110133555275610)
2108 E THOMAS RD

PHOENIX, AZ 85016
US

Netname: FON-110133555275610
Netblock: 65.165.12.0 - 65.165.15.255

Coordinator:
Troxel, Dan (DT73-ARIN) dant@SYSPAC.COM
602-955-5566 (FAX) 6029550085

Record last updated on 05-Apr-2001.
Database last updated on 23-May-2002 19:59:23 EDT.

From Network Solutions Whois server (http://www.netsol.com/cgi-bin/whois/whois)

Registrant:
Systems Solutions, Inc. (SYSPAC-DOM)
2108 E. Thomas Road
Phoenix, AZ 85016

Domain Name: SYSPAC.COM

Administrative Contact, Technical Contact:
DNS Administrator (DA24755-OR) dns
admin@ALEVELHIGHER.COM
Systems Solutions, Inc.
2108 E. Thomas Rd.
Phoenix , AZ 85016
US
602-955-0900
Fax- 602-955-7795

Record expires on 30-Aug-2002.
Record created on 29-Aug-1994.
Database last updated on 24-May-2002 02:35:04 EDT.

Domain servers in listed order:

NS1.ALEVELHIGHER.COM 65.165.37.5
NS2.ALEVELHIGHER.COM 65.165.12.165

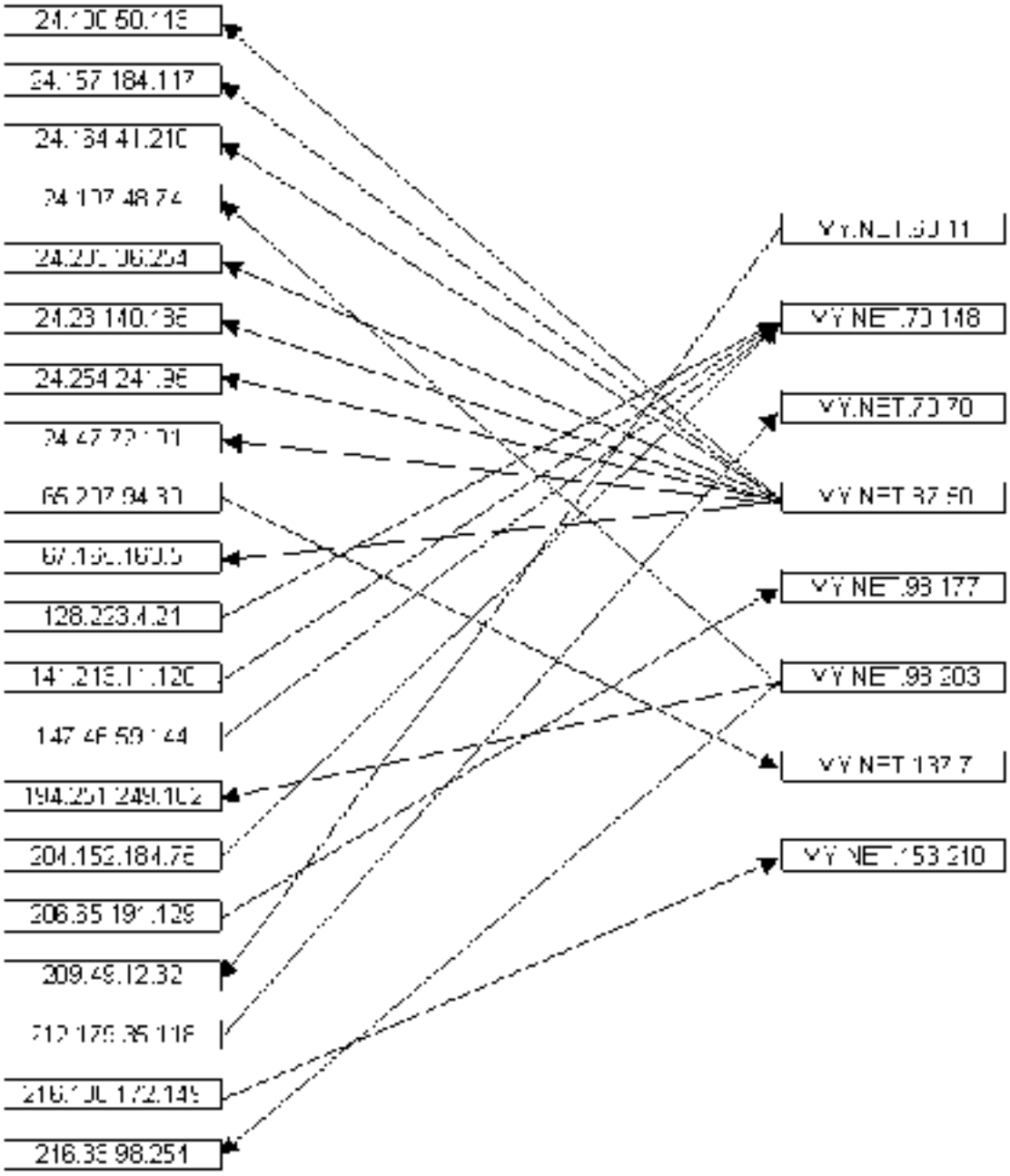
Link Graph

I combined all the alert, scan, and oos data into one. From this superset of data, I have pulled out the top 20 links between machines according to the number of logs created by that pair.

Source	Destination	# Logs
212.179.35.118	MY.NET.70.70	61295
MY.NET.87.50	24.164.41.210	20604
MY.NET.98.203	216.33.98.254	11066
206.65.191.129	MY.NET.98.177	7899
MY.NET.98.203	194.251.249.182	7144
204.152.184.75	MY.NET.70.148	6162
MY.NET.87.50	24.157.184.117	5660
216.106.172.149	MY.NET.153.210	5648
MY.NET.87.50	24.23.140.185	5575
MY.NET.98.203	24.197.48.74	4428
MY.NET.87.50	24.254.241.95	4292
65.207.94.30	MY.NET.137.7	3661
128.223.4.21	MY.NET.70.148	3610
MY.NET.60.11	209.49.12.32	3586
MY.NET.87.50	24.47.72.191	3520
141.213.11.120	MY.NET.70.148	3460
MY.NET.87.50	24.100.50.113	3284
147.46.59.144	MY.NET.70.148	3017
MY.NET.87.50	67.165.163.5	2791
MY.NET.87.50	24.203.36.254	2640

The graph is on the following page.

Link Graph



Insights into Internal Machines

MY.NET.87.50 is likely compromised as it has generated a significant number of scans.

MY.NET.60.11 is also compromised because it was the source of over 3000 “BACKDOOR NetMetro File List” alerts.

You can include MY.NET.98.203 in that list too as it has been doing a large amount of UDP scans.

Other machines that serve in particular roles (DNS, Web, SQL) have been noted in the analysis of the alerts.

MY.NET.153.210 was the target of a "MISC Large UDP Packet" and should be analyzed to see if it has been compromised.

MY.NET.137.7 is also under attack yielding many ICMP Destination Unreachable alerts.

MY.NET.98.177 has not only been hit hard by the Queso fingerprint, but GNUTella alerts as well.

MY.NET.70.70 got hit hardest by "Watchlist 000220 IL-ISDNNET-990517" and should definitely be analyzed.

MY.NET.70.148

Defensive Recommendations

Things appear to be fairly wide open. Due to the amount of malicious activity that is occurring the first thing to do, if possible, would be to pull the compromised machines off the network and either disinfect them or reinstall a hardened system. Then, there should be at least one firewall protecting the network. The lack of one is evident due to the number of machines that were scanned (over 5000). Either that or the rule set on the firewall needs to be tightened down. Also, any unnecessary services or daemons should be shut down on machines that don't need them. There are some services that get configured and made active with the default installation of some operating systems.

Analysis Process

The alert logs were processed with SnortSnarf initially to give me a picture of what had gone on during the 5 days. In addition, I wrote Windows batch files to filter all the logs with versions of gawk (or awk) and grep (or fgrep) that run natively in a Windows 2000 command shell (cmd). These were obtained at ftp://ftp.uni-koeln.de/pc/win32/misc/unxutils.zip. The filters reformatted the log data into either comma or tab delimited files that I could load into a Microsoft SQL 7.0 database. Once I had the data in the database, I could query against source/destination IP address, source/destination port, and alert name in any combination.

I started with the most numerous alerts and worked my way down from there until the top 5 external talkers had had their alerts analyzed.

References

All references have been noted earlier, either in-line with the text or at the end of sections.

Last Updated: October 19th, 2017

Upcoming Training

SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
Las Vegas 2018 - SEC503: Intrusion Detection In-Depth	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS London February 2018	London, United Kingdom	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced