Barbara Morgan

SANS Parliament Square – Intrusion Detection In Depth

GIAC Certified Intrusion Analyst (GCIA) – Practical Assignment Version 3.2

06 October 2002

Table of Contents

Part 1 – Describe the State of Intrusion Detection

Identify the Benefits of Utilising Automated Response within an IDS system and It's Potential Drawbacks

Abstract

The purpose of an Intrusion Detection System (IDS) is to monitor a computing environment for attacks, through collection and analysis of raw data and presentation of pertinent information. An attack could be fairly simple, such as a program that illegally modifies a user name, or complex, involving sequences of events that span multiple systems over an extended time period. Many IDS deployments alert of attacks in real-time, but they leave the determination of a suitable response to a human operator.

A successful attack can compromise your computer within a matter of seconds. Without an automated response your IDS will be reporting the attack, but will not be providing any defence. By the time an operator has been notified of the attack, your system will have already been compromised.

This paper explores response mechanism used within automated responses and tries to answer a number of questions, specifically:

- Can an IDS, with automated response enabled prevent your environment from being compromised?
- Can an attacker gain useful information about your IDS from the automated responses?
- Can an automated response result in unwanted or illegal events?

Background

Intrusion detection is becoming an increasingly important part of any company's security set-up and plays an important role in the prevention, detection and response to security breaches. One of the best ways to defend against computer-based attacks is to build a strong defence mechanism (both perimeter and internal). However in practice preventative measures are likely to be penetrated and a mechanism to detect and respond to any breaches of your defence will greatly assist in damage control.

An IDS will help you to detect intruders in a timely manner, however it will only be able to help prevent the attacker accomplishing his aim if a swift response is initiated. As an attack can be successful in compromising your system(s) within a few seconds, quick action is essential if your IDS is to be more than just a reporting, post-event analysis tool.

Detection Mechanisms

An IDS can be broadly classified into one of two groups, dependant upon its method of detecting intrusions, specifically:

- Anomaly detection
- Pattern matching detection

Anomaly detection is a heuristical IDS approach that looks for deviations from the norm. In the first stage a baseline defining normal network activity is established, from then on an intrusion (suspicious activity) is defined as any unacceptable deviation from expected values as defined by the first stage. The advantage of such an approach is that it does not rely upon known attack patterns and can therefore detect unknown attacks and is not dependent upon the IDS vendor releasing up-to-date signature databases.

The system will be able to "learn" standard behaviour and the set of variables that track behaviour does not require a significant amount of memory storage. However establishing a baseline is often a challenge and not all baseline objects exhibit consistent behaviour, therefore this approach is highly prone to false positives. An attacker who knows that intrusions are being determined based on statistical behaviour might be able to circumvent detection by avoiding activities that are measured and by spreading abusive behaviour among multiple accounts or across an increased time period.

Pattern matching detection compares activities against a collection of known attack signatures to identify intrusions. Known problems are defined in advance and signatures will trigger if the conditions defining the event are matched. The advantage of this approach over anomaly detection is that the system is less prone to false positives. The number and types of events to monitor are constrained to only those data items needed to match a pattern and the number and types of events to monitor for can be adapted according to the environment. For example if you do not have a mail server there is no need to watch for mail server specific attacks[1]. Also a pattern-matching engine tends to be more efficient due to the absence of floating-point computations required for statistical measures. However a pattern matching IDS will only alert on known attacks within its signature database, hence regular updates are vital to ensure the accuracy and usefulness of your IDS. There might be times when new attacks will not be caught by existing patterns and adding your own attack signatures is often a complicated procedure.

Both approaches have their own set of pros and cons and it appears sensible to combine them for a more effective tool. The division between anomaly detection and pattern matching is not completely clean as the overlap with statistical techniques is unavoidable in pattern matchers. For example a brute force login attack will trigger once a certain threshold of failed log-in attempts in a row are reached. This represents a pattern of interest and an anomaly.

IDS Data Sources

There are two main types of information that an IDS examines – network data and system data. Therefore you can often distinguish between a:

- Network based IDS (mostly dedicated systems),
- Host based IDS (an agent present on a host system)

A network-based IDS usually obtains its data by activating a network adapter in promiscuous mode. Most network IDS vendors recommend that you dedicate a system on the network to sniffing and analysing traffic. A network IDS does not introduce a new data source, nor does it alter any existing data source. However with encryption becoming increasingly popular, analysis of network traffic is not

[1] Unless interested in all attack attempts and not just those that pose a defined threat to your particular environment.

always possible as the data portion of the packet, where you would expect to find the majority of exploit code, is not interpreted by the IDS.

A host-based IDS that analyses system data is not presented with encrypted network traffic and can provide vital additional information once a system has been compromised. Furthermore it is often better placed to alert to malicious internal system users. Most host-based sensors do not require significant system resources to function, however they do rely upon a minimal level of auditing. If you currently do not implement this auditing capability within your system, adding this overhead will have to be considered.

Both data sources provide valuable information and some host-based sensors have additional capabilities to also detect network-based attacks (RealSecure Server Sensor, with its Firecell technology for example).

Irrespective of the data being reviewed or the IDS technology implemented, there are several characteristics of a good IDS, specifically:

- Capable of detecting the latest attacks
- Real-time attack detection and alerting
- Provide an audit trail that can be used in court if required
- Scalability
- Enable quick response to help in the prevention of security breaches

Great care is needed with any form of automated response. You do not want an over responsive policy that tries to terminate all of the processes running on behalf of a perpetrator, especially if this affects availability of resources that are crucial to you business. This paper will further address IDS characteristics that are particularly pertinent to automated response.

Automated Response – Theory & Practice

This section deals with the theory and practice of utilising an automated response mechanism. Wherever possible, specific examples taken from ISS RealSecure v6.5 will complement the theoretical discussion. This should not be seen as a recommendation but a reflection of my practical experience.

Introduction

In an ideal scenario your IDS should not only be able to detect an attack, but also be able to assess and control the damage, thus helping in the recovery from the attack.

Currently most IDS implementations are much less ambitious. Typically attack detection triggers a message to be sent to an operator via either a SNMP trap, an email or to a display on a dedicated console. The responsibility then lies with the operator to deal with the situation. This can range from blocking the offending source to requesting additional audit logs and saving these as evidence.

Human investigation and resolution is time consuming and introduces long delays. With the number of attacks increasing and automated attacks now becoming the norm, it becomes more and more apparent that human intervention is too slow to adequately prevent rapid attack propagation. Often the resulting damage such as loss of network availability and data disclosure bear extremely high costs for the business both financially and with regards to their reputation.

Automated response and system diagnostic should therefore be considered. This response can range from system reconfiguration to prevent further propagation of an attack through to restoring the operational status of the system through automatic data restoration.

For the purpose of this document an automated response is defined as follows:

Definition: *An automated response is a response that will stop an attack from propagating once detected by taking actions without human intervention.*

Each response will be analysed according to the following **criteria**:

a) Reaction Method

How does the automated response work in theory and where appropriate, how has it been implemented within RealSecure. Does the response:

- Reduce threat?
    - One-off (e.g. tcp reset)
    - Long term / permanent (e.g. firewall reconfiguration)
- Reduce vulnerability?
    - Shutdown of system / service
- Recover from damage?
    - Script that restores web content

b) Ease of Implementation

How easy is it to implement the proposed response? What is the potential for miss-configuration and is there any dependency on / interaction with other systems (e.g. firewalls)? What are the additional support costs (staff skill level, training costs) and complexity.

c) Reaction Time

Some responses will be effective instantly (e.g. tcp reset), whereas others will take some time before taking effect (e.g. re-configuration of a firewall rulebase that requires verification and reloading).

In some instances a 10 minute delay in the action taking effect is acceptable (e.g. reloading of good website code), at other times the reaction benefit needs to be instantaneously (e.g. a denial of service attack needs to be stopped before reaching its target).

d) Interruption to Service and impact on legal traffic

- No measurable service impact (e.g. tcp reset)

- Service outage due to a system shutdown for example

e) Possible Legal Implications

Most automated responses will not have any legal implications, however some more “ethically disputable” responses such as a return attack may very well have serious legal consequences.

The pros & cons of various options are not black and white. Depending on your business, certain options might not be possible or the benefits cannot justify the disruption caused.

Automated Response Options

The remainder of this paper discusses the various theoretical and practical response options available within an IDS. Each response is discussed in line with the above-mentioned criteria.

1. Network-based Responses

Responses in this category disconnect / reconfigure communications on behalf of (non-IDS) hosts.

a) TCP Reset

TCP resets are a common form of automated response with IDS vendors. A TCP reset is easy to achieve and does not involve much overhead. The circuit containing the perceived attack is forcibly reset. However this does not stop the attacker retrying immediately from the same IP address.

A TCP reset can be classified as a threat reduction response that has instantaneous effect, however it has no long lasting influence. It has no measurable influence on legitimate traffic as long as it is only applied to attack signatures that have a low percentage of false positives. This is generally true for all automated responses and is not specific to TCP resets.

A TCP reset is easy to implement, does not require any additional user training and does not introduce any additional costs.

There are no known legal consequences resulting in you utilising TCP resets within your IDS.

A TCP reset is a good response for most connection based attacks, for example the Unicode exploit against IIS. This attack signature does not have any false positive. Even if your systems are not vulnerable nobody should be allowed to type commands like http://x.x.1.2/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\, as there is no legitimate reason for it.

RealSecure v6.5 Network Sensor offers a TCP reset option called RealSecure Kill.

RSKill: Kills the connection by issuing a tcp-reset command to each party in the session.

The RSKill command within RealSecure can lead to information leakage. If you send out a RealSecure Kill and the source of the attack is monitored by a RealSecure system as well it will alert the owner of the source that one of his packets has been subject of a RealSecure Kill. Within this alert a RealSecure identification number is given. If nothing else the source owner will gain knowledge that this site it monitored by RealSecure. This might prove useful if an attacker wants to try to evade the IDS.

b) Reconfiguration of Perimeter Defences

Your IDS sends instructions to your perimeter device (e.g. a firewall) to prevent certain types of traffic from crossing into your network. This can range from denying a source-type of traffic through to stopping all traffic to or from a certain address.

The reconfiguration of the perimeter defence is a non-instantaneous response option due to a new policy needing to be generated, verified and installed before it can take effect. On some devices applying a new policy can cause a short network interruption where other connections may be lost.

Even so reconfiguration of perimeter devices provides you with an automated means to reduce threat long term, extreme care must be taken when utilising this response as it might result in shunning legitimate sources.

By using spoofed source addresses an attacker can use this automated response to create a Denial of Service (DoS) against your systems, for example:

- The attacker spoofs the source to be an address belonging to one of your main business partners.
- The attacker spoofs a whole range of addresses and stops your customers accessing your site.
- The attacker spoofs an address belonging to yourself (IDS, Firewall manager, etc.) and stops your from further managing your devices.[2]

A further drawback with this solution is that it requires a greater field of expertise from the personnel responsible for technical implementation and support, as they need to understand the firewall / router etc.

The key factor in deciding if this method is available for deployment is based around the compatibility of the necessary components. Not all IDSs are capable of sending reconfiguration instructions, likewise not all firewalls are capable of receiving these instructions. Communications between firewalls and IDSs is not based upon recognised standards, as such great care has to be taken to ensure the product selection is compatible.

There are no known legal implications resulting from this automated response.

Due to the danger of causing a DoS it is not recommended to utilise this response for any attack signature, as once an attacker has confirmed the response they will abuse it.

RealSecure v6.5 Network Sensor gives you the option to reconfigure Checkpoint Firewall 1 rulebases through its OPSEC response option.

OPSEC: Sends a message to a Checkpoint Firewall1 that instructs the firewall to prevent the attacking source address form crossing the firewall boundary for a user-specified period of time.

You can configure your response to either:

[2] Anti-spoofing on your firewall should prevent this scenario.

- Lock destination address
- Lock service
- Lock source address
- Lock source and destination address pair

The message is sent to the firewall using SAMP, Checkpoint's Suspicious Activity Monitoring Protocol, which is part of the Open Platform for Security (OPSEC).

2. Host-based Responses

Responses in this category protect the local host upon which an IDS is installed.

a) Dropping of packets

The IDS adds a "security validation layer" to the TCP stack. It prevents potentially harmful traffic from reaching its target by dropping the packets before they can reach the application on the target machine.

This kind of response reduces the threat immediately and depending on the time limit put on the access propitiation, can have medium to long-term effect. There is no interruption to legitimate activity.

Implementation of such a response is straight forward, has no interdependencies and requires no additional skill set from you personnel. There are no legal implications arising from this automated response.

This automated response is appropriate against any kind of attack that tries to cause damage by corrupting the target machine's TCP/IP stack (e.g. many denial of service attacks).

Within RealSecure v6.5 Server Sensor the response giving you this option is called Block.

Block: Offending packets get dropped before they can reach the TCP/IP stack on the target machine. This response also stops connections for http, ftp, smtp and finger signatures. You can configure your blocking options within the Global Response window and specify how long a particular source should be blocked.

b) Account Suspension

Upon noticing suspicious traffic originating from a user account upon a system your IDS suspends the account for a certain period of time.

This response can be seen as either reducing the threat or reducing vulnerability. It removes a vulnerability by closing down a hacked account, or if the account is used unlawfully by its rightful owner the threat to your environment is reduced by denying this individual further access, thus removing the threat.

In general this kind of response is easy to implement and the effect immediate. However depending on the Operating System disabling an account while the user is still logged in may have a different effect. With some Operating Systems, for example Windows, most access will only be completely removed once the user has logged off. It is therefore best to include a forced log-off in your response.

Unless dealing with a compromised account, this response will not hinder legitimate use of your systems. There are no legal implications associated with this response.

As already mentioned this response is either used upon noticing a compromised account on your system or a user exceeding the access rights for his account.

RealSecure v6.5 Server Sensor has two response options fitting within this category.

Suspend: Prevents a user form logging into a Windows NT machine for a specified period of time.

Disable: Prevents a user from logging into a Windows NT computer until the user's account is manually reinstated.

c) Automated data recovery

Upon identification on the context of a file changing, such as index.html, an automated reaction will be initiated whereby the original file, or indeed the entire website will be restored from a trusted medium such as write-once only CD Rom.

The reaction time will depend on the number of files that need to be copied and can range form almost instantaneous to a matter of a few minutes.

This response addresses the embarrassment of compromise, however, does not address the root cause of the compromise.

This response is easy to implement and does not rely on specialised skills beyond that of a systems administrator. Obviously this response has no legal implications and doesn't interrupt service as a shutdown of the website is not necessary.

RealSecure v6.5 does not specifically offer this response, however you can implement it as part of its "user defined" option discussed later in this document.

d) System Shutdown

For a compromised system a shutdown of the service or of the whole system may often be the only option left to prevent further damage. This will reduce your vulnerability, however it will lead to service outage and thus loss of business.

The reaction time will be swift, however this can depend heavily on your system, as a controlled shutdown may take longer.

This response should only be used as a last resource after careful assessment of the situation. Providing the IDS with enough information to make an educated decision is extremely difficult if not impossible. Your IDS would need to be able to analyse logs from various sources, e.g. firewalls and server logs to reduce the possibility of error. This requires a level of interoperability not easily found within today's commercially available products.

Unless you are a service provider dealing with set Service Level Agreements, a system shutdown has no legal consequences.

If system availability is a critical factor for your business I would not suggest using system shutdown as an automated response to any kind of attack, but instead rely upon the IDS alerting your operator to take appropriate actions.

RealSecure v6.5 does not currently offer this as a standard response, however gives you the option to define this action as part of your user specific responses (see below).

3. Mixed Host and Network based responses

a) Return Attack

In theory an IDS can initiate a return attack, however the practicality and legal implication of such a response are extremely questionable. In order to be able to launch a return attack, the system must perform automated passive information gathering to establish the source of the attack and any vulnerabilities. Attacks are often launched from previously compromised system or the source address may have been spoofed. Attacking a system not owned by the attacker will result in legal consequences on your behalf.

This response does neither reduce your threat nor your vulnerability and only satisfies your desire for revenge.

Due to the severe legal implications associated with this response it is not a valid option for any kind of attack and was mentioned for the sake of completeness only.

b) Responses offered by RealSecure v6.5 to fit various categories

User Specified: Runs a user-defined response. A user-specified response allows you to create a custom response by having RealSecure open an executable file when an event is detected. This executable file can be any .exe or batch file, and can have any command-line options you want.

You can define as many different user-specified responses as you want, but only one can be opened in response to a particular event. To open a series of executables, you can put all the commands into a batch file, which RealSecure can run.

Sensors supported: Server & Network

“User Specified” allows you to select an executable to be performed if an event is triggered. This has the potential of being an extremely powerful tool. Multiple executables can be strung together in a batch file and the response taken by the sensor can be quite complex.

Depending on your chosen options it can either reduce a threat, a vulnerability or indeed both. Successful implementation of this response requires a high level of skill. Reaction time, interruption to service, impact on legitimate traffic and legal implications all depend on the selected executables run.

Other related RealSecure v6.5 features

a) SecureLogic

Runs a user-created SecureLogic script. Secure Logic allows for sophisticated rule definition by correlating information from before, during, and after the attack. This helps to reduce false positives and increases alert quality.

Secure Logic can help to reduce false positives by enabling advanced correlation of events.

This feature is only available for RealSecure v6.5 Server Sensor.

b) Firecell

In addition to the detection of suspicious events (OS and limited network event detection) RealSecure v6.5 Server Sensor offers also a protection / prevention orientated element called “Firecell”.

Firecell technology can block suspicious traffic before it is seen by the server. Users can specify firewall-like rules for filtering out or blocking inbound/outbound traffic.

Can an attacker gain from your automated response?

The two main issues to be concerned about with regards to an attacker gaining from an automated response are:

- Information leakage

- Possibility for the attacker to create a Denial of Service attack

The likelihood of this occurring depends entirely on the kind of automated response and your implementation of it.

TCP resets in general do not have any information leakage issues associated with them, however the implementation in RealSecure 6.5 can lead to information leakage. When a RealSecure Kill is issues in response to an alert and the source address is also monitored by a RealSecure system, the system will alert to the fact that one of its packets has been subject to a RealSecure Kill. The alert contains a RealSecure identification number. This will provide the owner of the source with the knowledge that the destination address is monitored by RealSecure. The information might prove useful in further attempts to evade the IDS.

Perimeter reconfiguration can lead to the some serious Denial of Service scenarios. The attacker can either spoof a whole range of addresses thus stopping customers accessing your site or spoof the address belonging to one of your main business partners. The attacker could also spoof an address belonging to yourself (e.g. your IDS, firewall manager) and stop you from receiving vital information or managing important devices. However the last scenario should be prevented by anti-spoofing on your firewall.

Account suspension could possibly lead to a DoS if an attacker can compromise a large amount of server accounts (or the root account). However if the attacker has this kind of control over your system you will be faced with far more serious problems than those introduced by the automated response.

Due to the service interruption caused by a system shutdown this should only be used as your last resource. If this response is used carelessly it may very well lead to a DoS attack caused by your automated response option.

Centralised vs. Distributed?

Centralised reporting of security incidents has numerous advantages including cost, consistency, and accountability for actions. Conversely, you do not want automated responses, like disconnecting a hacker, to be adversely affected by network delays. The time it takes the attacked node to receive the “disconnected response” from a centralised response database could leave enough of a window for someone to plant a Trojan Horse. A middle-of-the-road approach would be to report security violations at a central console but to let each node in the networks immediately carry out a predefined automated response using its own computing resources, rather than looking up the appropriate response in a centralised database. A configuration option for centralised repost but distribute response would make this possible. [3]

RealSecure uses a middle-of-the-road approach combining centralised and distributed reaction. The reporting of alerts done centrally via an Event Collector collecting alerts and forwarding them to monitoring consoles. However reaction to events is distributed. Each sensor has its own response file stored locally and all actions originate from the sensor itself rather than from the centralised event collector thus ensuring quick minimal delay.

Conclusion

The future of IDS is detection, diagnosis and response to prevent attacks form spreading within your system and provide maximal damage control. Only quick response will be able to achieve this goal.

However care needs to be taken when using automated response to not hinder legitimate traffic. The two main dangers are:

- Responding to false positives
- Stopping of crucial processes that have not yet been corrupted by the attack

Before considering any form of automated response you need a good understanding of your network traffic and systems, your IDS needs to be well-tuned and false alerts kept to a minimum. Automated response should be used selectively and only be applied to attack signatures that are not prone to producing false positives.

If at all possible response should only be taken after sufficient authentication of the attack source. Otherwise a response to an attack from a spoofed source might ban the wrong traffic.

Responses such as system shutdown can stop crucial services and when used carelessly cause more damage in system outage than the attack itself. Shutdown of the whole system or part of a system should only be undertaken after careful damage assessment. Furthermore suitable procedures to ensure

[3] Intrusion Detection p24, Terry Escamilla

the downtime is minimal need to be in place.

Automated response options should be stored locally so that they cannot be adversely affected by network delay.

Ensure that any automated responses deployed do not leak information and thus give an attacker another means of gathering data. This would counteract any positive gain received by this response.

Commercial tools do offer a wide choice of automated responses, however care needs to be taken when using them in order not to leak information or cause unreasonable loss of availability of your systems.

References

Terry Escamilla, “Intrusion Detection – Network Security beyond the Firewall”, John Wiley & sons, Inc. (1998)

http://www.cerias.purdue.edu/coast/intrusion-detection/

Steven Chung, “An Intrusion Tolerance Approach for Protecting Network Infrastructure”, 1999

Internet Security Systems, RealSecure version 6.5

Part 2 – Network Detects

Legend to log formats used

1. Snort log format

```
[**] [1:524:3] BAD TRAFFIC tcp port 0 traffic [**]
```

Signature ID (SID), descriptive name of the event

```
[Classification: Misc activity] [Priority: 3]
```

Classification and priority as given to the event by snort

```
07/14-02:01:20.744488 211.47.255.21:35917 -> 46.5.114.52:0
```

Date & Time, Src Address & Src Port, Traffic Direction, Dst Address & Dst Port

```
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
```

Protocol type, Time to Live, Type of Service, IP ID, IP Length, Datagram Length, Don't Fragment Flag

```
******S* Seq: 0x1C2BF28D  Ack: 0x0  Win: 0x16D0  TcpLen: 32
```

Flags, Sequence Number, Acknowledgement number, Window Size, TCP Length

```
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
```

Options (Maximum Segment Size, Selective Acknowledgment, Window Scale), NOP acts as padding for no option.

2. Windump log format

```
02:01:20.744488 211.47.255.21.35917 > 46.5.114.52.0: S
```

Time, Src Address, Scr Port, Traffic Direction, Dst Address, Dst Port, Flags

```
472642189:472642189(0) win 5840
```

1st Sequence Number : Last Sequence Number (packet size in bytes), Window Size

```
<mss 1460,nop,nop,SackOK,nop,wscale 0> (DF)
```

Options (Maximum Segment Size, Selective Acknowledgment, Window Scale), Don’t Fragment Flag

Detect #1: Bad Traffic tcp port 0

Windump output:

```
02:01:20.744488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:23.734488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:29.734488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:41.734488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:52.744488 211.47.255.21.36282 > 46.5.114.52.0: S 513896898:513896898(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:55.734488 211.47.255.21.36282 > 46.5.114.52.0: S 513896898:513896898(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)

…<truncated for space>

06:11:28.994488 211.47.255.22.40829 > 46.5.148.130.0: S 2446445866:2446445866(0) win
5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
06:11:31.994488 211.47.255.22.40829 > 46.5.148.130.0: S 2446445866:2446445866(0) win
5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
06:11:37.994488 211.47.255.22.40829 > 46.5.148.130.0: S 2446445866:2446445866(0) win
5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
06:11:49.994488 211.47.255.22.40829 > 46.5.148.130.0: S 2446445866:2446445866(0) win
5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
```

Snort Logs:

```
[**] [1:524:3] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
07/14-02:01:20.744488 211.47.255.21:35917 -> 46.5.114.52:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x1C2BF28D  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**] [1:524:3] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
07/14-02:01:23.734488 211.47.255.21:35917 -> 46.5.114.52:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x1C2BF28D  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**] [1:524:3] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
```

```
07/14-02:01:29.734488 211.47.255.21:35917 -> 46.5.114.52:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x1C2BF28D  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**] [1:524:3] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
07/14-02:01:41.734488 211.47.255.21:35917 -> 46.5.114.52:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x1C2BF28D  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

… <truncated for space>

[**] [1:524:3] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
07/14-06:11:49.994488 211.47.255.22:40829 -> 46.5.148.130:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x91D1CD2A  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

1. Source of Trace

The trace was obtained from www.incidents.org/logs/raw/2002.6.14.

Log files 2002.6.15, 2002.6.16, 2002.6.17 and 2002.6.18 where also analysed for correlation data.

The network topology underlying these traces is unknown to me. The following assumption was drawn based on the content of the log files used:

- The monitored network is 46.5.0.0 / 16

2. Detect was generated by

Snort intrusion detection system v1.8.7 with standard ruleset.

The command used to read the raw tcpdump log file was:

```
./snort -dvr <input file> -l <output directory> -h 46.5.0.0/16 -c snort.conf
```

The rule that triggered this alert was SID 524:

```
alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD TRAFFIC tcp port 0 traffic";
sid:524; classtype:misc-activity; rev:3;)
```

This rule triggers for any tcp traffic to or from the external network, where the home network port equals 0. TCP port 0 is a reserved port and there is no legitimate reason for any traffic to be sent to this port. It is therefore classified as 'bad traffic' and should be treated as suspicious.

In this case there was no outbound traffic triggering the alert contained in the log file. In addition to the snort output the following windump command was run on the log file:

```
windump –nr <Input file> "tcp and port 0" > <Output file>
```

For a legend of all log file formats used please refer to "Legend to log formats used " at the beginning of part 2.

3. Probability that the source address was spoofed

The packets are TCP syn packets with destination port 0. No service runs on port 0 and standard traffic would not be assigned to this port. However hping by default uses a destination port of 0 and allows for spoofed syn scans by taking advantage of predictable IP identification numbers, thus enabling an attacker to gather information without giving away his/her identity.

From the logs available there is no indication that a three way handshake has been completed. If these packets have really been generated by a hping scan as I believe, then there is a possibility of the source address being spoofed. However it is also possible that these are hping packets that are not spoofed trying to identify a silent host that can be used as the spoofed source at a later stage.

There is no significant evidence pointing either way and I would say there is a 50% possibility of these addresses being spoofed. Either way these packets belong to an information gathering exercise.

4. Description of attack

TCP Syn packets are sent to various destination hosts on port 0. The DF flag is set, the IP ID is 0 and a set of 4 packets is sent to the same destination. The retry intervals are 3, 6, 12 seconds, which is expected behaviour for retry packets.

Source addresses:

- 211.47.255.20
- 211.47.255.21
- 211.47.255.22
- 211.47.255.23

Destination addresses:

- 46.5.114.52
- 46.5.194.214
- 46.5.193.73
- 46.5.148.130

Each source address sent 16 packets to a single destination IP. They were split into sets of 4 retries (same source port, sequence number). The retry interval is standard (3 seconds, 6 seconds, 12 seconds). This indicated that the attacker is not getting any response from the target hosts.

The source ports were ephemeral (> 1024) ports and the sequence numbers increased with each new packet set as is expected tcp behaviour. However the IP ID for each packet is 0, which is not standard IP behaviour as each IP packet should have a unique IP ID. This is a further indication (in addition to the destination port equalling 0) that the packets were custom crafted.

This is most likely reconnaissance traffic by a tool such has hping with its default destination port (0). The aim of the reconnaissance is most likely to identify if the target system is alive.

5. Attack mechanism

Hping works by sending custom TCP/IP packets to a destination port and reporting the packets it gets back. It is a TCP/UDP/ICMP ping utility with some added functionality. It allows you to control specific options of the packet that may allow it to pass through certain access control devices.

By default hping will send tcp packets to a target host on port 0 with a window size of 64 without any tcp flags set. This is often the best way to perform successful network mapping when a target is behind a perimeter defence device that is configured to drop ICMP. Moreover a tcp null-flag to port 0 has a good probability of not being logged.

Hping allows you to specify the destination port with the –p options. It also gives you the option to fragment packets, which again may help the packets to pass through an access control device. Furthermore the –b option sends packets with a bad UDP/TCP checksum. For a full list of configurable options please refer to http://www.hping.org/manpage.html.

An attacker could have used a hping command similar to "`hping -SN 0 -w 5840 -y a.b.c.d`" to generate these logs.

One of the special features of hping is that it allows you to launch TCP scans from a spoofed source while still providing you with the results of the reconnaissance.

Hping takes advantage of the predictability of the IP ID numbers in many Operating Systems. Every packet is identified by a 16 bit ID number. Many Operating Systems simply increment the session ID by 1 for each packet sent.

The spoofed SYN scan within hping involves three machines:

- A …Attacker
- S …Silent host
  (Must have predictably session IDs, in earlier versions of hping it also needed to be a 'zero traffic' host (i.e. not sending any packets while hping was scanning) so that the attacker can monitor the session IDs. This is not the case in hping2 or later.)
- T …Target

The scan works as detailed below.

1) The attacker stimulates S by sends a tcp SYN packet

2) S responds with a session ID, which A notes down.

3) A now stimulates the target with a spoofed source address of S.

4) T replies to S with either

   a) a SYN/ACK if the port is active

   b) or a RST/ACK if the port is closed.

5) S only replies to T with a RST if it receives a SYN/ACK packet. If this is the case its session ID increments by 1.

The attacker now goes back to step 1 and checks the session ID supplied by S in step 2. If it is incremented by 2 the port is open (or the target alive). If it is incremented by 1 the target is closed[4].

6. Correlations

Checking the log files for 5 consecutive days (14/07 to 18/07) revealed the same sort of scanning activity form these sources on the 14/07, 15/07, 16/07 and 17/07. No packets were captured on the 18/07.

The logs hold no other traffic destined to, or originating from the destinations being analysed.

For further correlation see also:

- GIAC GCIA Version 3.2 Network Detect #3 submitted by Ewen Fung to intrusions@incidents.org on the 02 September 2002
- LOGS: GIAC GCIA Version 3.2 Practical Detect(s) submitted by Ronald Clark to intrusion@incidnets.org on 08 August 2002

7. Evidence of active targeting

Eleven destinations within the 46.5.0.0/16 network were targeted from 5 source addresses (211.47.255.20 - .24). This is either an extremely slow scan or previous reconnaissance traffic singled out the target machines to be of particular interest.

There is no other traffic logged from these source addresses and I believe that this is not active targeting as such but random checks by the source.

8. Severity

[4] It will naturally increase by 1 because of A's packet.

The severity has been calculated with the following formula on a scale from 1 to 5, where 1 is lowest and 5 highest:

severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality = 3: The function of the destination systems is unknown. Therefore a medium value is assigned.

Lethality = 1: This is reconnaissance traffic and even so the information gathered could be used in later attacks, the traffic itself is not damaging the targets.

System countermeasures = 3: The defensive mechanisms on the host are not known. It is assumed we are dealing with reasonably patched systems (no more than 1 patch level behind).

Network countermeasures = 1: The packet-filtering device did not drop packets to destination port 0.

Therefore the overall Severity given to this attack is 0.

9. Defensive recommendation

Ensure the packet-filtering device drops all traffic with destination port 0.

10. Multiple choice test question

```
02:01:20.744488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:23.734488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:29.734488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
02:01:41.734488 211.47.255.21.35917 > 46.5.114.52.0: S 472642189:472642189(0) win 5840
<mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
```

Based on the trace above, what is unusual with this trace:

a) mss 1460

b) destination port = 0

c) the source port is the same for all 4 packets

d) DF is set

Answer: b)

(A maximum segment size of 1460 is standard for ethernet, the source port is the same if the packets are retries – the time interval between the 4 packets is 3, 6, 12 seconds and supports this, there are many valid reasons for the DF flag to be set. However tcp port 0 is a reserved port with no assigned service.)

Incident.org discussion inclusions

```
-----Original Message-----
From: jh [mailto:jh@pentasafe.com]
Sent: 30 September 2002 15:25
To: intrusions@incidents.org
Cc: jh@dok.org
Subject: Re: GIAC GCIA Version 3.2 Practical Detect Analysis


[I had sent this earlier, but apparently no action was ever taken on
this post by the moderator and it was returned - sorry for the
delay].

Anyway, comments below...

On Sun, Sep 22, 2002 at 20:38:07 +0100, "Morgan, Barbara"
<barbara.morgan@cgey.com> wrote:
> 3. Probability the source address was spoofed
> ================================================
> The packets are TCP syn packets with destination port 0. No service
> runs on port 0 and standard traffic would not be assigned to this
> port. However hping by default uses a destination port of 0 and
>.allows for spoofed syn scans by taking advantage of predictable IP
>.session identification numbers, thus enabling an attacker to gather
> information without giving away his/her identity.
>
> From the logs available there is no indication that a three way
> handshake has been completed. If these packets have really been
>.generated by a hping scan as I believe, then there is a possibility
> > of the source address being spoofed.
>
> However it is also possible that these are hping packets that are
> not spoofed trying to identify a silent host that can be used as
the
> spoofed source at a later stage.
```

Another avenue to explore is the possibility it could be a
fingerprint attempt. Many port-zero alerts I've run into have been
fingerprinting related.

> The source ports were epidermal (> 1024) ports and the sequence
> numbers

Perhaps you were going for 'ephemeral?'

> increased with each new packet set as is expected tcp behaviour.
> However the IP ID for each packet is 0, which is not standard IP
>.behaviour as each IP packet should have a unique IP ID. This is a
>.further indication (in addition to the destination port equalling
0) >.that the packets were crafted.

The IPID doesn't always have to contain a number other than zero.
Linux will, for efficiency purposes, send the packet with the DF flag
set and a zero IPID if the packet is small enough to be delivered
without fragmenting. There are a couple posts concerning this at:

http://marc.theaimsgroup.com/?l=bugtraq&m=98992536801625&w=2
http://marc.theaimsgroup.com/?l=bugtraq&m=101709117512191&w=2

> By default hping will send tcp packets to a target host on port 0
> with a window size of 64 without any tcp flags set. This is often
the >.best way perform successful network mapping when a target is
behind a
> perimeter defence device configured to drop ICMP. Moreover a tcp
>.null-flag to port 0 has a good probability of not being logged.

How about a brief explanation of why it has a good probability of not
being logged?

> 9. Defensive recommendation
> ===========================
> Ensure the packet-filtering device drops all traffic with
> destination port 0.

Source port 0 is a good one too, or even oddball TCP packets (ie:
with
all or no flags set, etc).

--
I don't know if it's what you want, but it's what you get. :-)
          -- Larry Wall in <10502@jpl-devvax.JPL.NASA.GOV>

Answer:

jh,

1) I have come to the conclusion that this is not a fingerprinting
attempt for the following reasons. In a fingerprinting scan I would
expect to see many different deformed packets sent to a target to
prompt responses from the target system identifying the OS. The

traces I analysed did not contain any malformed syntax as such - no illegal flag combinations, no variation on options etc. The source simply sent 4 packets of the same nature to each target. So in conclusion, whilst these types of packets may form a part of a fingerprint scan, there is no evidence of other packets that would allow a successful fingerprint of the system.

2) I was not aware the issues you have raised regarding Linux and IPID equal to 0.

3) With regards to your point of the logging of these scans on the firewall, to be honest the point I made I have read about from numerous sources, however I have never seen any supporting evidence for this behaviour. The following paragraph is my attempt to rationalise this statement and should not be considered facts.

The majority of firewalls are designed to allow or disallow connections base on a variety of parameters, however many assume the correct setting-up, maintenance off, and subsequent closure of TCP virtual circuits. The use of null and other obscure illegitimate flag combinations, being completely non expected can bypass the firewalls internal processes. To reduce an overburdened firewall log where every packet in a virtual circuit is logged it is common to only log the establishment of the circuit, or connections that disobey the rulebase based upon address or port. Those failed connections that are logged are typically attempts to establish an illegal connection or the pretence of being part of an established connection. Null scans are not attempting to be either of the two previous examples, therefore the firewall code may not record the event.

4) I agree with your suggestions about also dropping source port 0 and other odd packets, however I do not believe that these extra measures would help in preventing these particular packets entering your network. They would however prevent a lot of other mischief.

Thanks for all your comments, some made me think quite a bit :-)
Barbara

-----Original Message-----
From: Segall, Iain [mailto:iain.segall@cgey.com]
Sent: 26 September 2002 09:38
To: 'Morgan, Barbara'; 'intrusions@incidents.org'
Subject: RE: GIAC GCIA Version 3.2 Practical Detect Analysis

Barbara,

You say that you think that this is not active targeting. Have you thought about the possibility that as the attacker has narrowed his targets down that they believe they have found some vulnerable boxes?

Regards

Iain Segall

Answer:

Iain,

In my analysis I did mention that the target machines might have been singled out by previous reconnaissance traffic. However looking through previous logs there was no indication of any such attempt.

I therefore concluded that no such reconnaissance did take place and this was the first time that these machines were scanned.

You might argue that this is active targeting, however I believe that these machines were part of a much wider scan and not the specific aim of the attacker.

Thanks for your comments
Barbara

Detect #2: Web-CGI formmail access

Snort Log:

```
[**] WEB-CGI formmail access [**]
07/16-08:27:52.504488 24.93.246.20:4228 -> 46.5.180.133:80
TCP TTL:113 TOS:0x0 ID:20762 IpLen:20 DgmLen:388 DF
***AP*** Seq: 0x6940BC2E  Ack: 0x506BDE53  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72  GET /cgi-bin/for
6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65  mmail.pl?recipie
6E 74 3D 3C 66 6F 72 6D 6D 61 69 6C 65 64 40 79  nt=<formmailed@y
61 68 6F 6F 2E 63 6F 6D 3E 66 6F 72 6D 6D 61 69  ahoo.com>formmai
6C 65 64 40 79 61 68 6F 6F 2E 63 6F 6D 26 73 75  led@yahoo.com&su
62 6A 65 63 74 3D 77 77 77 2E 58 58 58 58 2E 63  bject=www.XXXX.c
6F 6D 2F 63 67 69 2D 62 69 6E 2F 66 6F 72 6D 6D  om/cgi-bin/formm
61 69 6C 2E 70 6C 20 31 2E 39 26 65 6D 61 69 6C  ail.pl 1.9&email
3D 53 6B 61 6E 6E 65 64 40 61 6F 6C 2E 63 6F 6D  =Skanned@aol.com
26 62 6F 64 79 3D 6D 69 53 6C 65 64 54 4D 20 61  &body=miSledTM a
6F 6C 25 32 45 63 6F 6D 26 62 6F 64 79 3D 6D 69  ol%2Ecom&body=mi
53 6C 65 64 54 4D 20 48 54 54 50 2F 31 2E 31 43  SledTM HTTP/1.1C
6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70  ontent-Type: app
6C 69 63 61 74 69 6F 6E 2F 78 2D 77 77 77 2D 66  lication/x-www-f
6F 72 6D 2D 75 72 6C 65 6E 63 6F 64 65 64 0D 0A  orm-urlencoded..
55 73 65 72 2D 41 67 65 6E 74 3A 20 47 6F 7A 69  User-Agent: Gozi
6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69  lla/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 35 2E 35 3B 20 77  ble; MSIE 5.5; w
69 6E 64 6F 77 73 20 32 30 30 30 29 0D 0A 48 6F  indows 2000)..Ho
73 74 3A 20 77 77 77 2E 58 58 58 58 2E 63 6F 6D  st: www.XXXX.com
0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65  ..Connection: Ke
65 70 2D 41 6C 69 76 65 0D 0A 0D 0A              ep-Alive....

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

1. Source of Trace

The trace was obtained from www.incidents.org/logs/raw/2002.6.16

The network topology underlying these traces is unknown to me. The following assumption was drawn based on the content of the log files used:

- The monitored network is 46.5.0.0 / 16

2. Detect was generated by

The detect was identified by Snort intrusion detection system v1.8.7 with a standard ruleset.

The command used to read the raw tcpdump log file was:

```
./snort -dvr <input file> -l <output directory> -h 46.5.0.0/16 -c snort.conf
```

The rule that triggered this alert was SID 884

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI formmail
access";flags:A+; uricontent:"/formmail"; nocase; reference:bugtraq,1187; reference:cve,CVE-
1999-0172; reference:arachnids,226; classtype:web-application-activity; sid:884;  rev:6;)
```

This signature triggers for any tcp packet originating externally and destined for a web server with destination port equal to HTTP_PORTS complying with the following rule options:

- Flags: A+: at least the ACK flag is set
- Uricontent: “/formmail”: within the content the text string “formmail” is found
- Nocase: deactivates the case sensitivity of content rules

For a legend of all log file formats used please refer to “Legend to log formats used “ at the beginning of part 2.

3. Probability the source address was spoofed

The probability that the source address is spoofed is extremely low. For the packet involved in this event, to be accepted by the target it has to be part of an established connection.

Sequence numbers are 32-bit integers, with a possible range of 0 through to 4,294,967,295. To successfully spoof the necessary acknowledgment, the attacker would have to guess the correct value for the target SYN. As most modern operating systems implement random initial sequence numbers, brute-forcing this value is impossible.

With modern techniques and tools, it is possible to successfully spoof an address on your own local network, however performing this will gain the attacker very little.

4. Description of attack

Source address: 24.93.246.20

Destination address: 46.5.180.133

Destination port: 80

Within the packet payload the cgi script */cgi-bin/formmail.pl* is requested and a number of variables are defined.

Matt Wright FormMail is a universal web-form-to-email gateway that allows for the creation of arbitrary form submission web pages without writing a dedicated CGI program for each one.

Please refer to Matt's Script Archive, Inc. on http://www.scriptarchive.com for more information regarding FormMail.

A number of vulnerabilities have been identified with this program since its release in July 1995:

- FormMail allows remote execution of arbitrary commands (CVE-1999-0172)

  User supplied data (from the "recipient" hidden field) is passed to a Perl OPEN function without correct input verification. This allows the use of the command separation shell meta-character (;) to execute arbitrary commands on the remote host.

  Consequences of this could be the destruction of data, web site defacement or the elevation of privileges through locally exploitable vulnerabilities. (see also BID 2079)

  This vulnerability is present in version 1.0 of the FormMail program.

- Remote Resource Usage (CVE-1999-173)

  The FormMail CGI program allows remote sites to make use of your Web server's resources by using your FormMail program for their own sites. This is present in versions of FormMail prior to v1.3, where the security hole was plugged by adding the @referrers variable.

- FormMail discloses environmental variable information (CVE-2000-0411)

This vulnerability allows an unauthorised remote user to obtain CGI environmental variable information from a web server running FormMail by requesting a specially formed URL containing variables such as PATH, DOCUMENT_ROOT, SERVER_PORT. Such a crafted URL will email the gathered information to the address given.

Exploit (as detailed in http://www.securityfocus.com/bid/1187):

http:/target/cgibin/formmail.cgi?env_report=PATH&recipient=<email address>&required=&firstname=&lastname=&email=&message=&Submit=<message>

The information obtained can be used to assist in any further attacks.

This vulnerability is present in version 1.6, 1.7 and 1.8

- FormMail anonymous email / spamming vulnerability (CAN-2001-0347)

FormMail.pl relies on information transmitted by the web client to the web server. Authentication of incoming requests is limited to a rudimentary validation check on the HTTP_REFERER script, which can easily be bypassed by simply declining to provide any referrer.

This makes forwarding of email messages which have been effectively anonymised trivial and the script became very popular with bulk e-mail spammers.

This vulnerability exists in versions 1.0 to 1.9.

To identify which FormMail vulnerability the attacker tried to exploit in this particular trace a closer look at the packet payload is required:

```
47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
```

Formmail.pl is requested

```
6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
6E 74 3D 3C 66 6F 72 6D 6D 61 69 6C 65 64 40 79   nt=<formmailed@y
61 68 6F 6F 2E 63 6F 6D 3E 66 6F 72 6D 6D 61 69   ahoo.com>formmai
6C 65 64 40 79 61 68 6F 6F 2E 63 6F 6D 26 73 75   led@yahoo.com&su
```

The email recipient is specified as formmailed@yahoo.com

```
6C 65 64 40 79 61 68 6F 6F 2E 63 6F 6D 26 73 75   led@yahoo.com&su
```

```
62 6A 65 63 74 3D 77 77 77 2E 58 58 58 58 2E 63   bject=www.XXXX.c
6F 6D 2F 63 67 69 2D 62 69 6E 2F 66 6F 72 6D 6D   om/cgi-bin/formm
61 69 6C 2E 70 6C 20 31 2E 39 26 65 6D 61 69 6C   ail.pl 1.9&email
```

The subject of the email is www.XXXX.com/cgi-bin.formmail.pl 1.9

```
61 69 6C 2E 70 6C 20 31 2E 39 26 65 6D 61 69 6C   ail.pl 1.9&email
3D 53 6B 61 6E 6E 65 64 40 61 6F 6C 2E 63 6F 6D   =Skanned@aol.com
```

The return email address is specified as skanned@aol.com

```
26 62 6F 64 79 3D 6D 69 53 6C 65 64 54 4D 20 61   &body=miSledTM.a
6F 6C 25 32 45 63 6F 6D 26 62 6F 64 79 3D 6D 69   ol%2Ecom&body=mi
```

The email body contains the message “miSled™.aol.com”

```
53 6C 65 64 54 4D 20 48 54 54 50 2F 31 2E 31 43   SledTM HTTP/1.1C
6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70   ontent-Type: app
6C 69 63 61 74 69 6F 6E 2F 78 2D 77 77 77 2D 66   lication/x-www-f
6F 72 6D 2D 75 72 6C 65 6E 63 6F 64 65 64 0D 0A   orm-urlencoded..
55 73 65 72 2D 41 67 65 6E 74 3A 20 47 6F 7A 69   User-Agent: Gozi
6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69   lla/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 35 2E 35 3B 20 77   ble; MSIE 5.5; w
69 6E 64 6F 77 73 20 32 30 30 30 29 0D 0A 48 6F   indows 2000)..Ho
73 74 3A 20 77 77 77 2E 58 58 58 58 2E 63 6F 6D   st: www.XXXX.com
0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65   ..Connection: Ke
65 70 2D 41 6C 69 76 65 0D 0A 0D 0A               ep-Alive....
```

The web browser used by the attacker shows up as Gozilla/4.0. However this field does not necessarily reflect the real browser as the field could have been modified or simply hard-coded into the exploit used.

This trace is a stimulus and relates to the vulnerability classified in CAN-2001-0357. The attacker is probing the system to see if it is vulnerable to the formail.pl spam exploit.

If an email is sent to formmailed@yahoo.com then the attacker knows that the system is vulnerable and may use the site as a spam relay at a later date.

This vulnerability has been fixed in FormMail 1.91. Even so FormMail 1.9 employs measures to attempt to validate the legitimacy of the recipient address it is still susceptible to spam exploits as the measures can be circumvented via cleverly crafted input.

See also

- http://www.securityfocus.com/bid/2079
- http://www.whitehats.com/info/IDS226

- http://xforce.iss.net/static/299.php
- http://xforce.iss.net/static/300.php
- http://www.securityfocus.com/bid/1187
- http://www.securiteam.com/exploits/5NP0I0U1GG.html
- http://www.securityfocus.com/bid/2469
- http://www.securiteam.com/securitynews/5KP0C2K3PM.html
- http://www.securiteam.com/unixfocus/5UP060U6BU.html
- http://www.iss.net/security.center/static/6442

5. Attack mechanism

FormMail is designed to accept variables from any form and mail them to a specified recipient email address and obtains desired destination e-mail addresses form the HTTP client. The HTTP client in turn expects it from the pre-defined value of a hidden HTML form filed called recipient within the form (or forms) associated with the specific web site or server where a given instance of FormMail is installed

Authentication of incoming requests is limited to the following rudimentary check on the HTTP_REFERER environment variable passed to the script from the local web server. (The web server, in its turn, obtains this value from the Referer: HTTP header supplied by the HTTP client.)

```
if ($ENV{'HTTP_REFERER'}) {
   foreach $referer (@referers) {
      if ($ENV{'HTTP_REFERER'} =~ m|https?://([^/]*)$referer|i) {
         $check_referer = 1;
         last;
      }
   }
}
else {
   $check_referer = 1;
}
```

The “security check” is designed to verify that information submitted via the form came via a proper or designated domain. However there exist several ways to easily bypass and circumvent the HTTP_REFERER checking code shown above.

The most obvious way is simply to decline to provide any Referer: header at all as part of the HTTP request. In such cases, the attacker will effectively ``pass'' the HTTP_REFERER validation test, and the remainder of the FormMail script will then be executed.

Once passed the HTTP_REFERER validation check the attacker has mostly free hand to manipulate the form input variables in order to direct an email message of his choice to any selected destination(s).

A URL such as

http://www.example.com/cgi-bin/FormMail.pl? recipient=email@address-to-spam.com&message=Proof%20that%20FormMail.pl%20can%20be%20used%20to%20send%20anonymous%20spam.

will send an anonymous email message to the recipient if the FormMail version used is vulnerable.

For more details regarding this attack mechanism please refer to http://marc.theaimsgroup.com/?l=bugtraq&m=98433523520344&w=2.

6. Correlations

No further attack signatures triggered by 24.93.246.20 were recorded in the log files for 2002.6.14 to 2002.6.18.

In the same period many web based attacks (including 28 WEB-CGI formmail access attacks from 9 sources) have been aimed at 46.5.180.133.

However this is a well-known vulnerability that has been exploited to send large quantities of unsolicited `spam' e-mail to large numbers of recipients.

Internet users became aware of this misuse between May 1997 and March 2001 and the original advisory can be found at: http://www.securityfocus.com/archive/1/168177 .

A report indicated that spammers were actively searching the web for exploitable FormMail scripts can be found at: http://www.extremetech.com/article/0,3396,s%253D25124%2526a%253D18236,00.asp#story4.

At present, spammers are continuing and increasing their abuse of earlier versions of FormMail, especially the widely deployed 1.6 version, at various sites where these earlier versions are, unfortunately, still installed.

7. Evidence of active targeting

The attack was directed at specific host which seems to be the web server and is searching for a well documented vulnerability in FormMail.

My conclusion is that this is active targeting.

8. Severity

The severity has been calculated with the following formula on a scale from 1 to 5, where 1 is lowest and 5 highest:

severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality = 3: The critically of your web server depends on your business and as this is unknown a medium value is assigned. Beside the web server any systems receiving spam email will be victims if the attack is successful. Again the systems are not known and a medium value is appropriate.

Lethality = 3: Information gathering for further attacks or assisting in anonymous spamming. Consumes your bandwidth, depending on the business this is of varying importance, again a medium value is assigned.

System countermeasures = 5: The defensive mechanisms on the host are not known. As this is an old vulnerability the assumption is made that a software patch has been applied.

Network countermeasures = 1: As the destination is a web server the packet-filtering device allows port 80 traffic.

Therefore the overall Severity given to this attack is 0.

9. Defensive recommendation

As FormMail is a program with a long history of security issues look for alternatives wherever possible.

If FormMail is used within the environment upgrade any versions of FormMail prior to v 1.91.

Even so all spam-related security holes in FormMail should have been fixed in version 1.91, it is strongly recommended that FormMail never be deployed in conjunction with any kind of e-mail auto-responder. Otherwise there is a possibility of your web server to be abused as the middle-man in a stealth mail bombing attack.

If at all possible hard code the recipient's email address in the formmail.pl program. Do not rely on the address submitted by the user.

10. Multiple choice test question

```
[**] WEB-CGI formmail access [**]
07/16-08:27:52.504488 24.93.246.20:4228 -> 46.5.180.133:80
TCP TTL:113 TOS:0x0 ID:20762 IpLen:20 DgmLen:388 DF
***AP*** Seq: 0x6940BC2E  Ack: 0x506BDE53  Win: 0x4470  TcpLen: 20
```

True or False:

It is easy to spoof the source address in the above packet trace.

Answer: False

(For the packet to be accepted by the target it has to be part of an established connection. To successfully spoof the source the correct acknowledgement number needs to be provided, i.e the attacker would have to guess the correct value of the target SYN. Most modern operating systems implement random sequence numbers and the chance of guessing the correct one is 1 in 4,294,967,295.)

Detect #3: Web-Misc long basic authentication string

Snort Log:

```
[**] WEB-MISC long basic authorization string [**]
07/18-00:57:28.934488 32.101.80.176:1165 -> 46.5.180.133:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:1044
***AP*** Seq: 0x24AD5B1F  Ack: 0x10D3F517  Win: 0x7F80  TcpLen: 20
47 45 54 20 2F 6D 61 69 6E 2F 76 65 6E 64 6F 72  GET /main/vendor
2F 74 61 6E 64 63 2E 68 74 6D 6C 20 48 54 54 50  /tandc.html HTTP
2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D  /1.1..Accept: im
61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78  age/gif, image/x
2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F  -xbitmap, image/
6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65  jpeg, image/pjpe
67 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76  g, application/v
6E 64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74  nd.ms-powerpoint
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E  , application/vn
64 2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C  d.ms-excel, appl
69 63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20  ication/msword,
2A 2F 2A 0D 0A 52 65 66 65 72 65 72 3A 20 68 74  */*..Referer: ht
74 70 3A 2F 2F 77 77 77 2E 58 58 58 58 2E 63 6F  tp://www.XXXX.co
6D 2F 73 75 70 2F 73 75 70 2E 68 74 6D 6C 0D 0A  m/sup/sup.html..
41 63 63 65 70 74 2D 4C 61 6E 67 75 61 67 65 3A  Accept-Language:
20 65 6E 2D 75 73 0D 0A 41 63 63 65 70 74 2D 45   en-us..Accept-E
6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 64  ncoding: gzip, d
65 66 6C 61 74 65 0D 0A 55 73 65 72 2D 41 67 65  eflate..User-Age
6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20  nt: Mozilla/4.0
28 63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53 49  (compatible; MSI
45 20 35 2E 35 3B 20 57 69 6E 64 6F 77 73 20 4E  E 5.5; Windows N
54 20 35 2E 30 3B 20 54 33 31 32 34 36 31 29 0D  T 5.0; T312461).
0A 48 6F 73 74 3A 20 77 77 77 2E 58 58 58 58 2E  .Host: www.XXXX.
63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A  com..Connection:
20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 41 75 74   Keep-Alive..Aut
68 6F 72 69 7A 61 74 69 6F 6E 3A 20 42 61 73 69  horization: Basi
63 20 63 32 52 72 5A 6E 4E 6B 61 6D 5A 73 61 7A  c c2RrZnNkamZsaz
70 7A 62 47 52 6D 61 32 70 7A 62 47 52 6D 61 32  pzbGRma2pzbGRma2
70 73 63 32 52 72 61 6D 5A 73 63 32 74 6B 61 6D  psc2RramZsc2tkam
5A 73 5A 47 74 71 63 32 52 72 5A 6D 6F 3D 0D 0A  ZsZGtqc2RrZmo=..
0D 0A 47 45 54 20 2F 6D 61 69 6E 2F 76 65 6E 64  ..GET /main/vend
6F 72 2F 74 61 6E 64 63 2E 68 74 6D 6C 20 48 54  or/tandc.html HT
54 50 2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20  TP/1.1..Accept:
69 6D 61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65  image/gif, image
2F 78 2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67  /x-xbitmap, imag
65 2F 6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A  e/jpeg, image/pj
70 65 67 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E  peg, application
2F 76 6E 64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69  /vnd.ms-powerpoi
6E 74 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F  nt, application/
76 6E 64 2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70  vnd.ms-excel, ap
70 6C 69 63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64  plication/msword
```

```
2C 20 2A 2F 2A 0D 0A 52 65 66 65 72 65 72 3A 20  , */*..Referer:
68 74 74 70 3A 2F 2F 77 77 77 2E 58 58 58 58 2E  http://www.XXXX.
63 6F 6D 2F 73 75 70 2F 73 75 70 2E 68 74 6D 6C  com/sup/sup.html
0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75 61 67  ..Accept-Languag
65 3A 20 65 6E 2D 75 73 0D 0A 41 63 63 65 70 74  e: en-us..Accept
2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C  -Encoding: gzip,
20 64 65 66 6C 61 74 65 0D 0A 55 73 65 72 2D 41   deflate..User-A
67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E  gent: Mozilla/4.
30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D  0 (compatible; M
53 49 45 20 35 2E 35 3B 20 57 69 6E 64 6F 77 73  SIE 5.5; Windows
20 4E 54 20 35 2E 30 3B 20 54 33 31 32 34 36 31   NT 5.0; T312461
29 0D 0A 48 6F 73 74 3A 20 77 77 77 2E 58 58 58  )..Host: www.XXX
58 2E 63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F  X.com..Connectio
6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 41  n: Keep-Alive..A
75 74 68 6F 72 69 7A 61 74 69 6F 6E 3A 20 42 61  uthorization: Ba
73 69 63 20 63 32 52 72 5A 6E 4E 6B 61 6D 5A 73  sic c2RrZnNkamZs
61 7A 70 73 61 32 70 73 61 32 59 34 4E 79 67 71  azpsa2psa2Y4Nygq
4A 69 67 71 4A 69 67 71 4A 69 67 71 4E 79 6F 6D  JigqJigqJigqNyom
4B 43 6F 6D 4B 43 6F 6D 4B 43 6F 6D 4B 43 6F 6D  KComKComKComKCom
4B 43 6F 6D 4B 43 6F 6D 4B 43 6F 6D 4B 43 6F 6D  KComKComKComKCom
4B 43 6F 6D 4B 43 6F 6D 4B 43 6F 6D 4B 43 6F 6D  KComKComKComKCom
4B 43 6F 6D 4F 51 3D 3D 0D 0A 0D 0A              KComOQ==....

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

[**] WEB-MISC long basic authorization string [**]
07/18-00:58:04.634488 32.101.80.176:1168 -> 46.5.180.133:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:1049
***AP*** Seq: 0x29A24FB7  Ack: 0x11E32194  Win: 0x7F80  TcpLen: 20
47 45 54 20 2F 6D 61 69 6E 2F 76 65 6E 64 6F 72  GET /main/vendor
2F 74 61 6E 64 63 2E 68 74 6D 6C 20 48 54 54 50  /tandc.html HTTP
2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D  /1.1..Accept: im
61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78  age/gif, image/x
2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F  -xbitmap, image/
6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65  jpeg, image/pjpe
67 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76  g, application/v
6E 64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74  nd.ms-powerpoint
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E  , application/vn
64 2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C  d.ms-excel, appl
69 63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20  ication/msword,
2A 2F 2A 0D 0A 52 65 66 65 72 65 72 3A 20 68 74  */*..Referer: ht
74 70 3A 2F 2F 77 77 77 2E 58 58 58 58 2E 63 6F  tp://www.XXXX.co
6D 2F 73 75 70 2F 73 75 70 2E 68 74 6D 6C 0D 0A  m/sup/sup.html..
41 63 63 65 70 74 2D 4C 61 6E 67 75 61 67 65 3A  Accept-Language:
20 65 6E 2D 75 73 0D 0A 41 63 63 65 70 74 2D 45   en-us..Accept-E
6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 64  ncoding: gzip, d
65 66 6C 61 74 65 0D 0A 55 73 65 72 2D 41 67 65  eflate..User-Age
6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20  nt: Mozilla/4.0
28 63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53 49  (compatible; MSI
45 20 35 2E 35 3B 20 57 69 6E 64 6F 77 73 20 4E  E 5.5; Windows N
54 20 35 2E 30 3B 20 54 33 31 32 34 36 31 29 0D  T 5.0; T312461).
```

```
0A 48 6F 73 74 3A 20 77 77 77 2E 58 58 58 58 2E  .Host: www.XXXX.
63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A  com..Connection:
20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A 47   Keep-Alive....G
45 54 20 2F 6D 61 69 6E 2F 76 65 6E 64 6F 72 2F  ET /main/vendor/
74 61 6E 64 63 2E 68 74 6D 6C 20 48 54 54 50 2F  tandc.html HTTP/
31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D 61  1.1..Accept: ima
67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 2D  ge/gif, image/x-
78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F 6A  xbitmap, image/j
70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 67  peg, image/pjpeg
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E  , application/vn
64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74 2C  d.ms-powerpoint,
20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E 64   application/vnd
2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C 69  .ms-excel, appli
63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20 2A  cation/msword, *
2F 2A 0D 0A 52 65 66 65 72 65 72 3A 20 68 74 74  /*..Referer: htt
70 3A 2F 2F 77 77 77 2E 58 58 58 58 2E 63 6F 6D  p://www.XXXX.com
2F 73 75 70 2F 73 75 70 2E 68 74 6D 6C 0D 0A 41  /sup/sup.html..A
63 63 65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20  ccept-Language:
65 6E 2D 75 73 0D 0A 41 63 63 65 70 74 2D 45 6E  en-us..Accept-En
63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 65  coding: gzip, de
66 6C 61 74 65 0D 0A 55 73 65 72 2D 41 67 65 6E  flate..User-Agen
74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20 28  t: Mozilla/4.0 (
63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53 49 45  compatible; MSIE
20 35 2E 35 3B 20 57 69 6E 64 6F 77 73 20 4E 54   5.5; Windows NT
20 35 2E 30 3B 20 54 33 31 32 34 36 31 29 0D 0A   5.0; T312461)..
48 6F 73 74 3A 20 77 77 77 2E 58 58 58 58 2E 63  Host: www.XXXX.c
6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20  om..Connection:
4B 65 65 70 2D 41 6C 69 76 65 0D 0A 41 75 74 68  Keep-Alive..Auth
6F 72 69 7A 61 74 69 6F 6E 3A 20 42 61 73 69 63  orization: Basic
20 4B 53 67 71 4B 53 67 71 4B 53 67 71 4B 69 59   KSgqKSgqKSgqKiY
71 4A 6C 35 65 4A 69 55 6C 4A 46 34 6C 4A 43 51  qJl5eJiUlJF4lJCQ
6C 58 6C 34 6C 4A 46 34 6C 4A 46 34 6C 4A 46 34  lXl4lJF4lJF4lJF4
6C 4A 46 34 6C 4A 46 34 6C 4A 46 34 6C 4A 46 34  lJF4lJF4lJF4lJF4
6C 4A 46 34 6C 4A 46 34 6C 4A 46 34 6C 4A 46 34  lJF4lJF4lJF4lJF4
6C 4A 46 34 6C 4A 46 34 6C 4A 46 34 6C 4A 46 34  lJF4lJF4lJF4lJF4
6C 4A 44 70 65 4A 53 52 65 4A 53 51 6D 58 69 70  lJDpeJSReJSQmXip
65 4A 69 6F 6D 58 69 67 71 4B 53 67 71 4B 53 67  eJiomXigqKSgqKSg
71 4B 69 59 71 4A 6C 34 71 4A 6C 34 71 4A 6C 34  qKiYqJl4qJl4qJl4
71 58 69 5A 65 4A 53 52 53 58 69 55 6B 4A 53 51  qXiZeJSRSXiUkJSQ
6A 4A 53 51 6A 4A 56 35 65 4A 6C 34 34 4E 7A 67  jJSQjJV5eJl44Nzg
35 4E 7A 59 71 4E 7A 59 34 4E 7A 59 34 0D 0A 0D  5NzYqNzY4NzY4...
0A                                               .

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

1. Source of Trace

The trace was obtained from www.incidents.org/logs/raw/2002.6.17

The network topology underlying these traces is unknown to me. The following assumption was drawn based on the content of the log files used:

- The monitored network is 46.5.0.0 /16

2. Detect was generated by

The detect was identified by Snort intrusion detection system v1.8.7 with a standard ruleset.

The command used to read the raw tcpdump log file was:

```
./snort -dvr <input file> -l <output directory> -h 46.5.0.0/16 -c snort.conf
```

The rule that triggered this alert was SID 1260

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC long
basic authorization string"; flags:A+; content:"Authorization\: Basic "; nocase; dsize:>1000;
classtype:attempted-dos; reference:bugtraq,3230; sid:1260;  rev:5;)
```

This signature triggers if a packet complies to all of the following:

- IP protocol = tcp
- Source IP is external
- Destination IP is part of the HTTP_SERVERS group
- Destination port is part of the ports defined in HTTP_PORTS
- Flags:A+ - at least the ACK flag is set
- Within the content the text string “Authorization\ : Basic” is found (string not case sensitive)
- Dsize: > 1000

For a legend of all log file formats used please refer to “Legend to log formats used “ at the beginning of part 2.

3. Probability the source address was spoofed

The probability of the source address being spoofed is extremely low. For the packet involved in this event to be accepted by the target is has to be part of an established connection – or at least make the target believe it belongs to one (ACK flag is set).

To successfully spoof the acknowledgement the attacker would have to guess the correct value for the target SYN. Sequence numbers are 32-bit integers with a possible range of 0 through to 4,294,967,295 [1]. Most modern Operating Systems implement random initial sequence numbers; this makes guessing the correct value nearly impossible.

With modern tools and techniques it is possible to successfully spoof an address on you own local network, however performing this will gain the attacker very little.

4. Description of attack

Source address: 32.101.80.176

Destination address: 46.5.180.133

Destination port: 80

Examining the packet payload found the offending string “Authentication: Basic”. However the password following is not over 2048 bytes. This is simply a long HTTP GET request and therefore the second criteria of dsize > 1000 has also been triggered.

5. Attack mechanism

In AOL Server version 3.0 and 3.2 a buffer overflow vulnerability exists that can allow a remote user to crash the server. The vulnerability is exploited by sending an overly large (2048 bytes) password string.

An attacker could create custom crafted packets which can overwrite stack variables resulting in the execution of arbitrary code.

CVE-2001-1067

Please refer to http://online.securityfocus.com/bid/3230 for more details.

6. Correlations

There are only two instances of this alert. Many web based attacks have been recorded against 46.5.180.133 within the log files 2002.6.14 to 2002.6.17, however none came form source 32.101.80.176.

7. Evidence of active targeting

Most traffic to 46.5.180.133 is to port 80, which indicates that the destination of the attack is a web server.

However in this case the snort IDS was triggered by a false positive. For the AOL buffer overflow exploit to work the password string must be 2048 byte. The two packets causing this alert had a total Datagram Length of 1044 and 1049 bytes respectively.

These two packets are long HTTP GET requests that triggered this signature due to their Datagram Length being over 1000 bytes.

8. Severity

The severity has been calculated with the following formula on a scale from 1 to 5, where 1 is lowest and 5 highest:

severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality = 3: Most likely a web server. The critically of your web server depends on your business and as this is unknown a medium value is assigned.

Lethality = 5: If successfully allows an attacker to execute arbitrary commands on your system and can lead to DoS.

System countermeasures = 5: The defensive mechanisms on the host are not known. It is assumed that patches are applied every month. As the vulnerability was first reported in August 2001 it is assumed that the server has been patched if indeed it was an AOL server.

Network countermeasures = 1: As the destination is a web server the packet-filtering device allows port 80 traffic.

Therefore the overall Severity given to this attack is 2.

9. Defensive recommendation

For any AOL server version 3.0 and 3.2 upgrade to a patched version of the package.

http://aolserver.com/archive/server/aolserver-3.4.tar.gz

To reduce the number of false positives triggered by innocent web traffic modify your snort signature by increasing the Datagram Length threshold to 2048.

10. Multiple choice test question

```
[**] WEB-MISC long basic authorization string [**]
07/18-00:57:28.934488 32.101.80.176:1165 -> 46.5.180.133:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:1044
***AP*** Seq: 0x24AD5B1F  Ack: 0x10D3F517  Win: 0x7F80  TcpLen: 20
```

How can you be sure the above packet trace is a false positive?

a) DgmLen: 1044

b) TCPLen: 20

c) Both the Ack and Push flag are set and the packet is part of an existing connection.

Answer: a)

(The AOLServer Long Authentication Sting Buffer Overflow Vulnerability is exploited by sending a password of 2048 bytes.)

Part 3 – Analyse This

Executive Summary

SANS University have approached Morgan Limited to aid them in a review of their IDS logs. Concern was raised within the university that undesirable activity may have taken place, be it unauthorised external access attempts or abuse of legitimate user privileges. Morgan Limited are pleased to present the findings of their review within this report.

The review took place over several days within the month of August 2002. The logs files containing the potentially suspicious activity covered the dates 1st August through to 5th August 2002.

The goal of the review was widespread and far reaching, however the key aims of the review were:

- To identify suspicious external systems that were attempting to penetrate the university's external network.
- To identify internal systems within the university campus that were participating in illegitimate network activity in contravention of the university's computer acceptable usage policy.
- To identify any systems within the university campus that may have been compromised by an external user or have been affected by a virus, worm or similar automated program.

Upon completion of the review, Morgan Limited identified a number of areas of concern. There was strong evidence of an internal system having been compromised with the Nimda worm. There was also circumstantial evidence of a further nine systems having been infected. A number of internal systems originated several differing exploits against Microsoft IIS systems, specifically the Unicode and ISAPI overflow exploits. No evidence was discovered that may concretely identify these systems as having been compromised, nor of use by mischievous internal users. These specific issues are detailed fully within section 5 of this report. Morgan Limited highly recommend that these systems be reviewed as a matter of urgency and the appropriate measures taken as detailed in section 5.

Beyond the specific issues previously detailed, SANS University, like many other educational establishments, suffers from the dual pronged attack of external users expecting an easy compromise and internal users attempting less than legitimate activities. These findings are based upon the large number of hosts involved in the suspicious activity and the variety of exploits. Morgan Limited do not believe at this moment that the university is subject to a concentrated attack from an individual or group of individuals as the recorded activity does not identify a sufficient intensity of vulnerability mapping from any specific host.

SANS University should seek to rectify the issues identified within this report. Issues identified strongly indicate several systems that have either been compromised by worms or are being used in an illegitimate manner by internal users. Additionally traditional mechanisms of security best practice should be enforced such as a suitable computer security policy and defence mechanism strength in depth both at the firewall and behind.

1. *Introduction*

Morgan Limited was approached by SANS University to conduct a security audit based upon the analysis of five consecutive days of log data gathered by the university's Snort IDS engine.

The log data comprises of three types:

- Alert Logs – Logs containing signature matches as identified by the IDS

- Scan Logs – Logs containing network reconnaissance scans as identified by the IDS

- OOS Logs – Logs containing "Out of Spec" packets, i.e. TCP packets with strange or illegal combination of flags set. All entries in this log file are related to events which generated alerts in the first two sets of logs, and thus, provide corroborative data for those events.

The log data was made available on www.incidents.org/logs and the following log files that have been selected for analysis:

Alert Logs	Portscan Logs	Out of Spec Logs
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	Scans.020804.gz	oos_Aug.4.2002.gz
alert.020805	Scans.020805.gz	oos_Aug.5.2002.gz

The date range of 1 August to 5 August 2002 has been selected because it represents the most recent set of 5 consecutive days of log file where all three file types are available.

2. Methodology

The aim of this analysis is to answer the following questions:

- Who is attacking the network and how?
    - Are the attacks internal or external?
    - How can defence mechanisms be improved?
- Who is scanning the network and how?
    - Are the attacks internal or external?
    - How can defence mechanisms be improved?
- Have any hosts been compromised?

To facilitate the identification of "interesting" traffic in the vast amount of log data provided first data is presented statistically.

From these statistics the most interesting signatures/targets are selected for a closer analysis.

3. *Alert Summary*

Over 2.2 million alerts were recorded within the 5 days of log data provided. That is an average of 446134.4 alerts per day or around 5 alerts per second.

However the alerts are not evenly distributed and the majority of all alerts (68%) were recorded on the 5th August with a slow build-up over the weekend.

Date	Number of Alerts
1/08/02 (Thursday)	66232
2/08/02 (Friday)	44493
3/08/02 (Saturday)	72626
4/08/02 (Sunday)	528375
5/08/02 (Monday)	1518946
Alert Total:	2230672

The table below shows a list of all the different alert signatures triggered over the 5 day period.

# of Alerts	Event Name	Alert Summary	# of distinct Source IPs	# of distinct Destination IPs
874497	NIMDA - Attempt to execute cmd from campus host	Nimda is a worm affecting Microsoft systems, modifying web documents and certain executables	10	105426
492624	spp_http_decode: IIS Unicode attack detected	Unicode attacks use Unicode characters in place of regular of ascii characters in a uri to bypass local access control	572	86341
481322	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize	An attempt to exploit a buffer overflow vulnerability within a ISAPI extension which can give rise to web page defacement originating from and internal source	MY.NET.84.234	480587
122875	NIMDA - Attempt to execute root from campus host	Nimda is a worm affecting Microsoft systems, modifying web documents and certain executables	MY.NET.100.208	69454

106849	UDP SRC and DST outside network	Both the source and destination of the packet are external	148	14
53561	spp_http_decode: CGI Null Byte attack detected	Triggers if the http decoding routine finds a %00 in an http request, an attacker may try to confuse a Perl script about where the end of it’s input is. Sites that use cookies with URL encoded binary data or SSL encrypted traffic may trigger false positives.	37	54
30074	SMB Name Wildcard	An attempt to identify Netbios resources on a Windows network. Once resources are identified specific exploits can be launched	8957 (all external)	2737 (all internal)
24214	TFTP - External UDP connection to internal tftp server	63/udp connection to an internal tftp server from an external source.	7	4
14577	External RPC call	An external source attempts to call a RPC port	6	6250
11917	Watchlist 000220 IL-ISDNNET-990517	Packets received with a source specified in Watchlist 000220	98	41 (all internal)
4113	Possible trojan server activity	Triggered by any acitivity with source or destination port 27374 (used by SubSeven and Ramen)	339	451
2543	SUNRPC highport access!	A high SUNRPC port has been accessed	14 (all external)	13 (all intenal)
2053	IRC evil - running XDCC	Alert on IRC communication	8 (all internal)	28 (all external)
1305	Watchlist 000222 NET-NCFC	Packets received with a source specified in Watchlist 000222	35	25 (all internal)

1293	EXPLOIT x86 NOOP	A number of contiguous bytes are detected that could be no-operation machine language codes. NOOPs are often used to pad out buffer overflow attacks. This alert is indicating that it may have found an attempt to run attack code via a buffer overflow exploit.	34 (all external)	38 (all internal)
1120	Queso fingerprint	Queso is a tool used to identify operating systems. OS determination is a first step in compromising a system.	71 (all external)	30 (all internal)
927	SNMP public access	This alert is triggered when an SNMP request, port 161, is made with a password of "public" which is the default community string.	11 (all external)	4 (all internal)
788	connect to 515 from outside	Indicates an attempt to access a server using the lpr print spooler service	4	604 (all internal)
730	Attempted Sun RPC high port access	Records an attempt to access a high SUN RPC port	138 (all external)	6 (all internal)
679	Samba client access	Triggers for destination port equal 139 for an external source	2 (all external)	422 (all internal)
628	High port 65535 udp - possible Red Worm - traffic	Identifies possible udp Code Red traffic	70	61
314	IDS552/web-iis_IIS ISAPI Overflow ida nosize	An attempt to exploit a buffer overflow vulnerability within a ISAPI extension which can give rise to web page defacement originating from and external source	300 (all external)	241 (all internal)
260	ICMP SRC and DST outside network	Both the source and destination of the packet are external	7	8
236	SMB C access	An attempt to utilise Netbios to connect to the C drive	6 (all external)	139 (all internal)

173	TFTP - Internal UDP connection to external tftp server	63/udp connection to an external tftp server from an internal source.	6	10
166	beetle.ucs	Beetle.ucs is a host that houses a CD burner allowing university students to burn disks. This seems to be a user defined snort signature and without the exact signature content I can only guess that this alert indicates remote access to this host.	36	37
136	Incomplete Packet Fragments Discarded	Fragments can be used to bypass firewalls and perform denial of service attacks. This check highlight possible dangerous packets.	17 (all external)	11 (all internal)
88	NMAP TCP ping!	Stealthy scan trying to identify which computers on a network are alive.	27 (all external)	25 (all internal)
58	EXPLOIT x86 setuid 0	Shellcode with an attempt to change user permission to root	29 (all external)	25 (all internal)
57	Null scan!	A likely source of the null scan is NMAP. The null scan might be used to identify a computer's operating system.	16 (all external)	9 (all internal)
53	Tiny Fragments - Possible Hostile Activity	Malicious fragmentation can be used to launch denial of service attacks or can be a method of network mapping that goes undetected by some firewalls.	11 (all external)	6 (all internal)
48	EXPLOIT x86 stealth noop	NOOPs are often used to pad out buffer overflow attacks. This alert is indicating that it may have found an attempt to run attack code via a buffer overflow exploit.	2 (all external)	2 (all internal)
44	High port 65535 tcp - possible Red Worm - traffic	Identifies possible tcp Code Red traffic	13	14
42	STATDX UDP attack	Attempt to exploit Statd demon	3 (all external)	36 (all internal)

38	EXPLOIT x86 setgid 0	Shellcode attempting to change group permissions to root	24 (all external)	20 (all internal)
13	SMB CD...	This event indicates an attempt to circumvent directory access control by trying to change to the "..." directory.	3 (all external)	6 (all internal)
13	TCP SRC and DST outside network	Both the source and destination of the packet are external	7	8
11	MY.NET.30.4 activity	Traffic involving MY.NET.30.4	3 (all external)	MY.NET.30.4
11	HelpDesk MY.NET.70.50 to External FTP	Outbound ftp traffic form MY.NET.70.50	MY.NET.30.50	2
11	External FTP to HelpDesk MY.NET.70.50	Inbound ftp traffic to MY.NET.70.50	5	MY.NET.70.50
9	HelpDesk MY.NET.70.49 to External FTP	Outbound ftp traffic form MY.NET.70.49	MY.NET.30.49	2
8	External FTP to HelpDesk MY.NET.70.49	Inbound ftp traffic to MY.NET.70.49	5	MY.NET.70.49
6	TFTP - External TCP connection to internal tftp server	63/tcp connection to an internal tftp server from an external source.	2	2
5	EXPLOIT NTPDX buffer overflow	An attempt to exploit a news server	3 (all external)	3 (all internal)
4	HelpDesk MY.NET.83.197 to External FTP	Outbound ftp traffic form MY.NET.85.83	MY.NET.83.197	3 (all external)
3	Back Orifice	Triggers on source or destination port 31337	2 (all external)	2 (all internal)
3	RFB - Possible WinVNC - 010708-1	Detection of VNC traffic	3 (all external)	3 (all internal)
3	DDOS shaft client to handler	Communication between shaft client and handler	209.73.180.8	MY.NET.70.161
2	Traffic from port 53 to port 123	Records traffic between port 53 & 123	2	2

2	SYN-FIN scan!	The SYN-FIN scan tries to identify open ports. The attacker uses the invalid SF flag combination to try to elude detection by intrusion detection systems, or possibly to fingerprint the operating system.	2 (all external)	2 (all internal)
1	MY.NET.30.3 activity	Traffic involving MY.NET.30.3	129.171.149.28	MY.NET.30.3

A total of 53 different signatures were detected.

4. Top Talkers Summary

Top 10 Sources (External and Internal)

Source Address	# of Alerts	# of distinct Exploits	# of distinct Destination IPs
MY.NET.100.208	1433754	4	107664 (all external)
MY.NET.84.234	481327	2	480590 (all but 1 external)
3.0.0.99	51359	1 (UDP Scr Dst Outside N/W)	1 (10.0.0.1)
63.250.213.12	32117	1 (UDP Scr Dst Outside N/W)	1 (233.28.65.148)
MY.NET.81.37	27085	2 (http_decodes)	2 (all external)
194.98.189.139	8375	2 (STATDX and External RPC)	5456 (all internal)
MY.NET.85.74	6990	2	12 (all external)
80.137.90.34	6899	2	47 (all internal)
MY.NET.111.230	6090	1 (TFTP - External UDP connection to internal tftp server)	1 (192.168.0.216)
MY.NET.111.231	6059	1(TFTP - External UDP connection to internal tftp server)	1 (192.168.0.216)
Total # of Alerts:	**2060055**		

The Top 10 Source Addresses are responsible for 92% of all alerts generated. The majority of the sources are internal.

Top 10 Destinations (External and Internal)

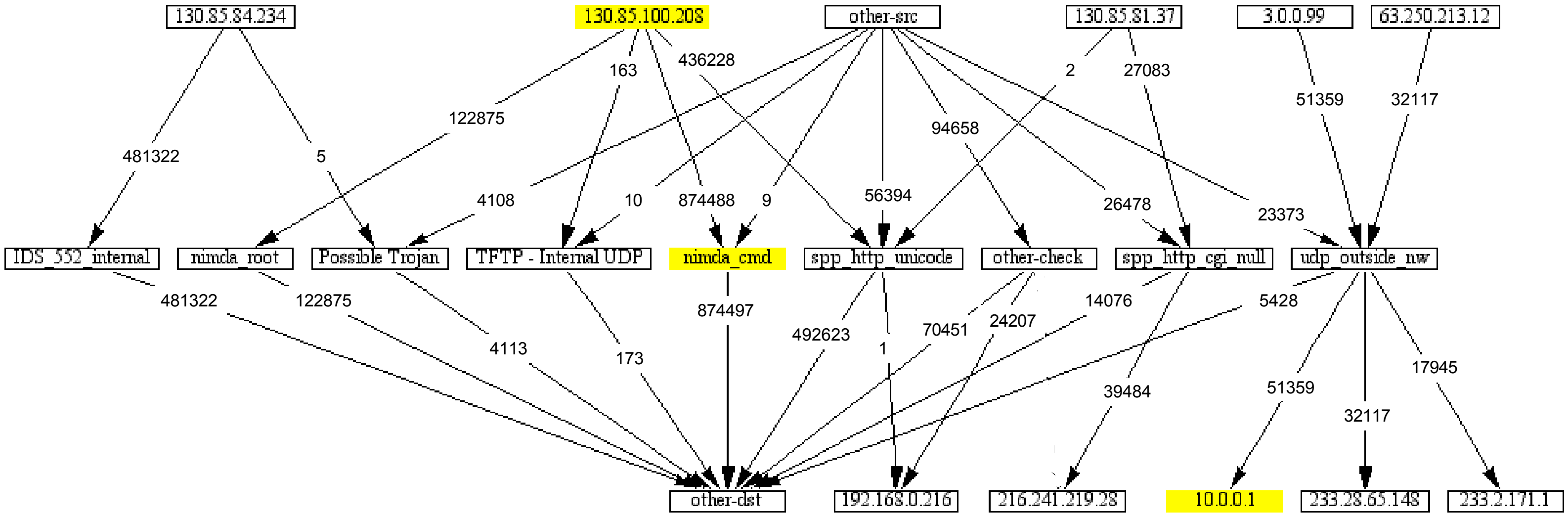
Destination Address	# of Alerts	# of distinct Exploits	# of distinct Source IPs
10.0.0.1	51359	1 (UDP SRC and DST outside network)	1 (3.0.0.99)
216.241.219.28	39484	1 (spp_http_decode: CGI Null Byte attack detected)	5
233.28.65.148	32115	1 (UDP SRC and DST outside network)	1 (63.250.213.12)
192.168.0.216	24208	2	5
233.2.171.1	17945	1 (UDP SRC and DST outside network)	100
152.163.210.84	6457	2	2
233.28.65.173	4975	1 (UDP SRC and DST outside network)	3
207.200.86.97	4758	1 (spp_http_decode: IIS Unicode attack detected)	8
MY.NET.104.204	4489	9	1552
209.10.239.135	3631	1 (spp_http_decode: CGI Null Byte attack detected)	5
Total # of Alerts:	**189421**		

Only 8% of all alerts generated are addressed to the Top 10 Destination Addresses. There are a total of 596644 different destinations (an average of less than 4 alerts per target).

The majority of destinations are external. This indicates that most alerts were triggered by suspicious internal traffic.

Top 5 Relationship Link Graph

The link graph below shows the relationship between the top 5 sources, the top 5 destinations and the top five alerts plus the additional alerts triggered by the top 5 sources. Those highlighted in yellow are the highest ranked within their respective category based upon volume.



The depicted sources are responsible for 90.8% of all alerts, however, the depicted destinations only received 4.2 % of all alerts. Only 4.2% of all alerts do not belong to one of the 8 specified signatures.

There is a very strong correlation between the top sources and signatures. However the alerts are distributed over an extremely large set of destinations and very little correlation exists between the top sources and top destinations.

5. Top 5 alert analysis

1. Nimda – Attempt to execute cmd from campus host

Nimda is a worm that affects systems running Microsoft Windows 95, 98, ME, NT and 2000. It spreads by multiple mechanisms:

- from client to client via email
- from client to client via open network shares
- from web server to client via browsing of compromised web sites
- from client to web server via active scanning for and exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities (VU#111677 and CA-2001-12)
- from client to web server via scanning for the back doors left behind by the "Code Red II" (IN-2001-09), and "sadmind/IIS" (CA-2001-11) worms

Nimda modifies web documents and certain executable files found on the systems it infects and creates numerous copies of itself under various file names.

Once infected the client machines begin scanning for vulnerable IIS servers. Nimda looks for backdoors left by previous IIS worms: Code Red II [IN-2001-09] and sadmind/IIS worm [CA-2001-11]. It also attempts to exploit various IIS Directory Traversal vulnerabilities (VU#111677 and CA-2001-12).

The infected client machine attempts to transfer a copy of the Nimda code via tftp (69/UDP) to any IIS server that it scans and finds to be vulnerable.

Nimda can lead to denial of service as a result of network scanning and email propagation.

Please see "CERT Advisory CA-2001-26 Nimda Worm" for more detail.

Alert Statistics:

This alert accounts for 39.2% of all alerts with a total of 874497 occurrences from 10 internal sources going to 105426 external destinations.

Source IP	# of Alerts
MY.NET.100.208	874488
MY.NET.82.87	1

MY.NET.130.20	1
MY.NET.70.169	1
MY.NET.70.144	1
MY.NET.111.30	1
MY.NET.70.16	1
MY.NET.83.176	1
MY.NET.105.10	1
MY.NET.165.19	1

The most common source / destination pairing is MY.NET.100.208 → 130.116.101.102 with 33 events.

Correlation:

MY.NET.100.208 is responsible for nearly all Nimda_cmd alerts. In addition this source also triggers all of the Nimda_root alerts and 89% of all IIS Unicode alerts detected by the IDS.

Furthermore MY.NET.100.208 also triggered 163 TFTP alerts (internal UDP connection to external tftp server).

The 163 TFTP alerts are suspicious as once a client is infected it attempts to transfer a copy of the Nimda code via tftp (69/UDP) to any IIS server that it scans and finds to be vulnerable.

MY.NET.100.208 generated 1433591 Nimda / Unicode alerts and is overall responsible for 64% of all alerts recorded over the five day period.

It is highly likely that this machine has been compromised.

Recommendations:

A system administrator should to check MY.NET.100.208 and the other 9 sourced for any signs of Nimda or any other compromise.

For a check guide please refer to “CERT Advisory CA-2001-26 Nimda Worm”

Consider applying both ingress and egress content filtering. The University needs to stop accepting and generating Nimda traffic immediately.

2. spp_http_decode: IIS Unicode attack

Unicode attacks cover a broad class of exploits against Microsoft IIS web servers. If successful his exploit allows an attacker to execute command by issuing crafted http queries. This exploit is a variant form an older exploit called dot dot, which allows for directory traversal.

The Unicode vulnerability is exploited by substituting standard ascii characters in a URI with their Unicode representations, thus possible bypassing access control restrictions.

Code Red (I and II), Nimda and sadmin all rely in some part on Unicode translation tricks to escape from a normal IIS directory and climb up and around a file system via relative directory commands.

For more detail regarding the Unicode vulnerability see:

- http://xforce.iss.net/alerts/advise68.php
- http://www.securityfocus.com/archive/96/183184
- http://rr.sans.org/threats/unicode.php.

Alert Statistics:

This alert accounts for 22.1% of all alerts with a total of 492624 occurrences from 572 unique sources to 86341 different destinations.

Top 5 sources:

Source IP	# of Alerts
MY.NET.100.208	436228
MY.NET.85.74	6982
MY.NET.152.19	2885
MY.NET.153.145	2827
MY.NET.153.168	2003

Top 5 external sources:

Source IP	# of Alerts
80.137.90.34	6889

151.203.178.36	2475
202.98.223.86	503
61.32.238.10	443
218.0.239.64	424

There are a total of 229 unique internal source addresses responsible for a total 478630 events and a total of 343 unique external source addresses responsible for a total 13994 events.

Top 5 internal destinations:

Destination IP	# of Alerts
MY.NET.105.10	302
MY.NET.91.154	301
MY.NET.70.58	295
MY.NET.91.8	294
MY.NET.5.96	291

Correlation:

In conjunction with the Nimda alerts from MY.NET.100.208 there is a strong likelihood of the University having been susceptible to a successful attack.

Both 80.137.90.34 and 151.203.178.26 have triggered more than a 1000 Unicode alerts.

80.137.90.34 ranks with a total of 6990 alerts in place 8 in the overall top talkers list. All alerts for this source were recorded on the 05/08/02. In addition to the IIS Unicode attacks the "beetle.ucs" attack was triggered 10 times by this source. The address belongs to "Deutsche Telekom AG" and checking www.dshield.org there were no attacks registered for this IP.

151.203.178.26 belongs to "Verizon Internet Services, US" and no attackes were registered on www.dshield.org. Beside the Unicode alerts this IP also triggered 7 SMB Wildcard alerts. All alerts from this source were triggered on the 05/08/02.

However without having access to the responses from the target system it is not possible to determine if these are successful Unicode attacks.

Recommendation:

Ensure all machines are patched with the latest updates.

Traffic from top offending external machines should be put on a watch list and if more Unicode events are recorded for them the respective owners should be contacted.

3. IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize

The Microsoft Internet Services Application Programming Interface (ISAPI) allows extensions to functionality on IIS and ISA servers. This vulnerability exploits an issue with the coding of one of these extensions, which can lead to a buffer overflow, giving rise to defacement of web pages.

For more detail please see:

- http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise79
- http://whitehats.com/info/IDS552
- http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

Alert Statistics:

This alert makes up 21.6% of all alerts with a total of 481322 events from a single source (MY.NET.84.234) to 480587 destinations.

Source IP	# of Alerts
MY.NET.84.234	481322

All but one event are targeted at external destinations; no destination was targeted more than 3 times.

Only 481116 of alerts had a well know http port (80) as the destination port, the rest (206) were sent to spurious destination ports.

Correlation:

MY.NET.84.234 is ranked at second place in the top ten talkers list. Apart from the ISAPI Overflow attacks this source also triggered 5 “Possible Trojan server activity” alerts.

The source was mainly active on the 04/08/02.

Recommendation:

Investigate source MY.NET.84.234 closer by examining the server logs. Establish if this activity is due to the machine being compromised, or a breaching of the University’s Security Policy.

If there does not already exist a Security Policy defining user behaviour Morgan Limited strongly advices to produce one.

Malicious traffic originating from SANS University need to be stopped. This is best accomplished through the reconfiguration of the perimeter filtering device.

4. Nimda – Attempt to execute root from campus host

See “Nimda – Attempt to execute cmd from campus host” for details.

Alert Statistics:

This signature accounted for 5.5% of all alerts with a total of 122875 events all originating from one source (MY.NET.100.208) to 69454 destinations. All destinations are external.

Source IP	# of Alerts
MY.NET.100.208	122875

No destination was targeted more than 6 times.

Correlation:

See “Nimda – Attempt to execute cmd from campus host” for details.

Recommendation:

See “Nimda – Attempt to execute cmd from campus host” for details.

5. UDP SRC and DST outside network

This alert is triggering when UDP packets are identified upon the internal network and both the source and destination addresses are external. This is most likely attempted spoof attacks, as unlike TCP UDP does not operate a controlled circuit and therefore there are no sequence numbers to guess, making such an attack much easier.

Alert Statistics:

UDP Source and Destination outside network events make up 4.8% of all alerts. There is a total of 106849 occurrences from 148 sources to 14 different destinations.

Top 5 sources:

Source IP	# of Alerts
3.0.0.99	51359
63.250.213.12	32117
63.250.213.73	4975
128.2.121.148	358
129.69.9.126	356

List of destinations:

Destination IP	# of Alerts
10.0.0.1	51359
233.28.65.148	32117
233.2.171.1	17945
233.28.65.173	4975
229.55.150.208	185
233.40.70.50	172
68.34.76.5	35
68.34.76.6	26
239.255.255.250	25
239.255.255.253	5
129.6.15.28	2
239.255.255.251	1
192.168.1.12	1

207.46.226.34	1

Correlation:

There were 3 source / destination pairs with over 1000 events:

- 3.0.0.99 → 10.0.0.1 (51359 events)
- 63.250.213.12 → 233.28.65.148 (32117 events)
- 63.250.213.73 → 233.28.65.173 (4975 events)

Recommendations:

Set up anti-spoofing on your perimeter defence. This will stop these packets from entering or leaving your network.

6. Port Scan Analysis

The following port scan types were recorded during 1 August to 5 August 2002:

Scan Type	# of Scans
UDP	3054482
SYN	983321
VECNA	71
INVALIDACK	60
NULL	59
NOACK	52
UNKNOWN	18
FIN	4
FULLXMAS	4
SYNFIN	4
XMAS	4
NMAPID	1

The next table shows the top 10 source addresses, all are internal and likely to be the result of malicious users as opposed to compromised hosts.

Source	# of Scans
MY.NET.70.200	2436010
MY.NET.84.234	478409
MY.NET.100.208	169326
MY.NET.70.207	134718
MY.NET.82.2	125957
MY.NET.165.24	104405
MY.NET.83.150	89918
MY.NET.81.27	31866
MY.NET.137.7	28433
MY.NET.70.133	23441

The next table shows the top 10 destination addresses

Destination	# of Scans
216.254.108.19	11179

67.68.113.139	9158
62.229.74.253	8697
140.192.175.183	8288
66.130.178.166	7474
210.187.110.110	6806
216.254.108.22	6628
12.245.28.142	6590
66.245.32.193	6472
66.233.62.244	5678

7. OOS Log Analysis

The OOS logs contain packets that have flag combinations that differ from the standards defined for TCP/IP. Such packets can be caused by innocent packet corruption, but can also be crafted deliberately in an attempt to fingerprint your hosts. There is no standard defining the response for such packets and different operating systems respond differently to invalid flag sets.

Potential attackers use applications such as nmap or queso to generating stimulus traffic towards a host, usually as a form of fingerprinting.

All possible flag combinations found within the OSS logs:

Flag Set	# of packets	Flag Set	# of packets
21S*****	1604	2*SF**A*	1
21S*R***	2	2*SFRPA*	1
21*FR***	2	*1SFR***	1
2*SF****	2	**SFRPAU	1
21S***A*	2	21SFRPAU	1
21*FRPAU	2	21S*R*AU	1
2*SFR**U	2	21S***AU	1
**SFR*A*	1	**SF***U	1
21**RPAU	1	2*SFR*A*	1
21SFR*AU	1	*1SF**AU	1
*1SF*P*U	1	*1SF**A*	1
21*F****	1	21S**P**	1

21SFR*A*	1	21SFR**U	1
21S**PAU	1	21SF*P*U	1

Top 10 sources:

Source	# of packets
68.32.126.64	652
62.76.241.129	345
209.116.70.75	214
212.35.180.17	83
65.210.154.210	48
213.250.44.19	29
61.132.74.239	18
202.155.91.142	18
209.132.232.101	18
211.154.85.159	17

Top 10 destinations:

Destination	# of packets
MY.NET.6.7	660
MY.NET.97.217	241
MY.NET.97.238	104
MY.NET.100.217	95
MY.NET.253.20	85
MY.NET.111.198	54
MY.NET.100.165	43
MY.NET.253.125	41
MY.NET.253.114	37
MY.NET.6.40	34

Correlation with Alert and Port Scan Logs

68.32.126.64 and 62.76.241.129

For the top 2 sources 68.32.126.64 and 62.76.241.129 no entries could be found in either the attack or port scan logs.

209.116.70.75

Source 209.116.70.75 is also the originator of 642 port scans and shows up in the alert logs as the originator of 645 Queso Scans.

209.116.70.75 belongs to Red Hat, Inc.

This source is suspicious and should be watched for further attacks.

8. Suspicious host summary

Top 3 internal sources

MY.NET.100.208

This is a possible compromised host. Please see “Nimda – Attempt to execute cmd from campus host” for detail.

This host also triggered 169326 port scan entries (#3 in the overall port scan table).

This host needs to be looked at by a system administrator

MY.NET.84.234

The majority of all alerts are IIS ISAPI Overflow events. See “IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize” for more details.

In addition there are 5 “Possible Trojan server activity” events with a source port of 27374 associated with SubSeven and Ramen.

478409 port scans recorded form this source (#2 in the overall port scan table)

This host seems to be either compromised or is being misused for dubious hacking activity and needs to be heavily examined.

MY.NET.81.37

All alerts (27085) originating from this source belong to either one of the following two signatures:

a) CGI Null Byte attack detected

b) IIS Unicode attack detected

All destinations are external.

This host seems to be either compromised or is being misused for dubious hacking activity and needs to be heavily examined.

Top 5 external sources

For the top 5 external hosts a whois lookup was performed and www.dshield.org was queried for any reported attacks from the respective source.

3.0.0.99

51359 UDP Source and Destination outside network alerts to destination 10.0.0.1.

OrgName: General Electric Company
Address: GE IDS Princeton, NJ 08540
Country: US
TechPhone: +1-518-612-6672
TechEmail: genictech@ge.com

# ARIN Whois database, last updated 2002-09-27 22:05

No attacks have been reported to www.dshield.org for this source.

63.250.213.12

32117 UDP Source and Destination outside network alerts to destination 233.28.65.148

OrgName: Yahoo! Broadcast Services, Inc.
Address: 701 First Avenue Sunnyvale, CA 94089
Country: US
TechPhone: +1-408-349-7183
TechEmail: netblockadmin@yahoo-inc.com

# ARIN Whois database, last updated 2002-09-27 22:05

No attacks have been reported to www.dshield.org for this source.

194.98.189.139

This source caused 8375 alerts split between two signatures.

a) Statdx UDP attack

b) "External RPC call" alerts 5456 internal source

```
inetnum:     194.98.189.128 - 194.98.189.143
address:     UUNET FRANCE
address:     215, Avenue Georges Clemenceau
address:     F-92024 NANTERRE Cedex
phone:       +33 1 56 38 22 00
fax-no:      +33 1 56 38 22 01
e-mail:      net-adm@mciworldcom.fr
remarks:     **************************************
remarks:     For all spamming or hacking problems
remarks:     please send your requests directly to
remarks:     abuse@fr.uu.net
remarks:     **************************************
```

Searching for other attacks reported for this source on http://www.dshield.org revealed the following results

DShield Profile: Country: FR

Contact E-mail: abuse_AT_fr.uu.net (bounced)

Total Records against IP: 392

Number of targets: 380

Date Range: 2002-08-05 to 2002-08-05

63.250.213.73

4975 UDP Source and Destination outside network alerts to destination 233.28.65.173

```
OrgName:    Yahoo! Broadcast Services, Inc.
Address: 701 First Avenue Sunnyvale, CA 94089
Country: US
TechPhone:  +1-408-349-7183
TechEmail:  netblockadmin@yahoo-inc.com

# ARIN Whois database, last updated 2002-09-27 22:05
```

No attacks have been reported to www.dshield.org for this source.

80.137.90.34

6889 IIS Unicode alerts and 10 beetle.ucs alerts

```
descr:      Deutsche Telekom AG, Internet service provider
abuse:       abuse@t-ipnet.de
person:      Security Team
address:     Deutsche Telekom AG
address:     Technikniederlassung Schwaebisch Hall
address:     D-89070 Ulm
address:     Germany
phone:       +49 731 100 84055
fax-no:     +49 731 100 84150
e-mail:     abuse@t-ipnet.de
```

No attacks have been reported to www.dshield.org for this source.

Recommendation

A list if the top external offenders should be kept and the owner of the IP address contacted.

References

http://www.hping.org

GIAC GCIA Version 3.2 Network Detect #3 submitted by Ewen Fung to intrusions@incidents.org on the 02 September 2002

LOGS: GIAC GCIA Version 3.2 Practical Detect(s) submitted by Ronald Clark to intrusion@incidnets.org on 08 August 2002

O'Reillys, Internet Core Protocols

http://www.scriptarcive.com

http://www.securityfocus.com.bid/1187

http://www.securityfocus.com/bid/2079

http://www.whitehats.com/info/IDS226

http://xforce.iss.net/static/299.php

http://xforce.iss.net/static/300.php

http://www.securiteam.com/exploits/5NP0I0U1GG.html

http://www.securityfocus.com/bid/2469

http://www.securiteam.com/securitynews/5KP0C2K3PM.html

http://www.securiteam.com/unixfocus/5UP060U6BU.html

http://www.iss.net/security.center/static/6442

http://marc.theaimsgroup.com/?l=bugtraq&m=98433523520344&w=2.

http://www.securityfocus.com/archive/1/168177 http://www.extremetech.com/article/0,3396,s%253D25124%2526a%253D18236,00.asp#story4

http://online.securityfocus.com/bid/3230

http://aolserver.com

CERT Advisory CA-2001-26 Nimda Worm

http://xforce.iss.net/alerts/advise68.php

http://www.securityfocus.com/archive/96/183184

http://rr.sans.org/threats/unicode.php

www.dshield.org

http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise79

http://whitehats.com/info/IDS552

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

Angela Orebaugh, “SANS GIAC Certification Practical Assignment”

Brian K Sheffler, “SANS GIAC Certification Practical Assignment”

Colby DeRodeff, “SANS GIAC Certification Practical Assignment”

Gary Smith, “SANS GIAC Certification Practical Assignment”

Brian K Sheffler, “SANS GIAC Certification Practical Assignment”

Jeff Holland, “SANS GIAC Certification Practical Assignment”

Tod A Beardsley, “SANS GIAC Certification Practical Assignment”

Steven L Drew, “SANS GIAC Certification Practical Assignment”

Appendix A: Approach for "Analyse This"

1) Alert, Port scan and OOS logs were combined in three files for easier analysis

2) The alerts and port scan files were converted into comma-separate values and in the alert file all spp_portscan entries were removed as these events are covered in the port scan file.

   The tools used for this was "csv.pl" by Tod A. Beardsley, GIAC GCIA Practical (version 3.1)

3) Events were grouped in various ways into "Events of Interests"

   The tools used for this was "summarize.pl" by Tod A. Beardsley, GIAC GCIA Practical (version 3.1)

4) The port scan logs were analysed with the help of two perl scripts devised by Chris Kuethe, GIAC GCIA Practical. The scripts used were "scanalyze" and "scancount"

5) The OOS logs were analysed with the help of grep commands to prepare the data and Excel.