Global Information Assurance Certification Paper

GIAC Intrusion Detection in Depth

"Evaluation Explanation"
SANS 2002 San Antonio
GCIA Practical Assignment 3.3

BY Kerry Long
October 3,2002

Table of Contents

Practical Abstract:

This paper is in response to the GIAC requirements in effect under Practical Assignment 3.3. As such it deals with a research topic; an analysis of three-network log files; and the thorough analysis of 5 days worth of IDS logs from a local University. The research topic focuses on the lack of credible intrusion detection evaluation techniques that provide reliable and repeatable methods to judge network-based intrusion technologies. The network log analyses focus on packets apparently captured on A SANS provided cable network. Of interest is a new detect that might be a new passive listening Trojan horse. The thorough analysis of a week's worth of University Snort log files leads to the discovery that this University's network is less than secure. It also attests to the stunning popularity of the game "Medal of Honor" among students.

The State of Intrusion Detection: Evaluating the Detector - Part 1

Abstract

This paper examines the current options available to effectively evaluate network-based intrusion detection systems. First, a brief discussion is conducted that describes the need for a formalized, testing methodology for adequately assessing new IDS techniques and technologies. A background on the foundations for this effort is followed by a detailed discussion on the current attempts by academia, government, and industry to address this need.

Introduction

We are for all intents and purposes still witnessing the beginning stages of intrusion detection technology. Other networking technologies such as routers, switches and even firewalls seem far more advanced than are the current offerings of network-based intrusion detection systems (NIDS). All of these products accomplish their respective goals both effectively and efficiently. The tasks performed by each of these devices are almost completely automated and as a result incredibly fast and predictable.

Current NIDS technologies, on the other hand, require a great deal of human intervention and analysis to deliver on their goal of detecting network intrusions. The current technology is both manpower intensive and tedious. There is also a

great deal of variability in the quality of the results obtained. Undoubtedly this is due in large part to the complexity of the problem that NIDS' attempt to address. Unfortunately, the problem is only going to get worse.

As router, switch and firewall technologies advance so too does the speed and amount of network traffic that will enter an enterprise. Existing NIDS devices are soon to be overwhelmed by an ever-increasing onslaught of information that will be demanded by network users. This has created a potential crisis in the field that some euphemistically call the "NIDS challenge".

From my own vantage point at a major government research laboratory, I am already witnessing many diverse solutions to this NIDS challenge. All solutions seem promising when pitched by the project's proponent. Many of the methods employ vastly different techniques and/or technologies. It is currently very difficult to assess the relative merits of these various techniques and technologies. Unlike some more established networking products, NIDS products have not had a reliable, repeatable methodology for testing. Even NIDS products that have been "tested" by established NIDS vendors through customer focus groups or self-initiated tests have little to offer besides anecdotal evidence to their effectiveness.

The lack of an accurate testing methodology actually retards advances in NIDS technology. Promising methods and advances are often indistinguishable from less promising methods. The NIDS industry has had no generally accepted, objective way to determine where to devote resources to improve performance. This has been the unfortunate state of NIDS technology. All this may be changing, however, with the introduction of several testing initiatives.

In the Beginning

The need for an effective methodology for testing NIDS technology has been recognized for quite some time. Back in October 1993, a group of researchers from the University of California, Davis wrote an influential paper entitled " A Methodology for Testing Intrusion Detection Systems (Puketza et all, 1993)." In this work the authors set out to define a detailed methodology that would accurately assess three performance objectives. The first performance objective, an NIDS' detection range, focused on the NIDS' ability to detect known intrusions throughout streams of benign traffic. The second performance objective, economy in resource utilization, dealt with the percentage of system resources that a host must devote to the NIDS function. The third and final performance objective, resilience to stress, measured an NIDS' ability to handle increasing amounts of network activity.

In addition to defining a testing methodology, the UC researchers developed test scripts that would simulate attacks known at the time. Though good as a first

step, current NIDS performance issues such as false positive rates and detection of anomalous behaviors were not really focused on in this work.

For several years, IBM Zurich Labs has also been a pioneering member in the effort to assess NIDS technology. In 2000 they built on previous work and developed a methodology for assessing NIDS' based on the amount of false positives and false negatives that an NIDS would register (Alessandri, 2000). The methodology is based on a fairly complex set of defined activities. These activities are classified as either malicious or non-malicious and are further subcategorized by the vulnerabilities they attempt to exploit, the interface they use, etc. Efforts to implement this model appear to be still ongoing.

These works and other similar academic endeavors created the foundation on which many of the current well-known initiatives are based.

The Current Players

DefCon

A Hacker group recently has attempted to set up the "ultimate NIDS test bed"(Middleton, 2001). At DefCon 2001, a group of hackers calling themselves the Shmoo group captured all the network traffic initiated during the well-known 72-hour hack fest "Capture the Flag. " This information has been made available to the public to be used as an NIDS test suite. Unfortunately, just throwing a series of captured exploits at an NIDS does little to test its real world capabilities. Without applying a rigorous detailed testing methodology, there is little information that can be gained about the NIDS' ability to distinguish false positives in normal traffic for example. At best, this traffic could be used to determine the rate of false negatives that a given NIDS technology might register.

NFR Security Inc. obtained the DefCon data and tested its NFR NIDS product with it. What they found was that the traffic is "highly unusual" and contains traffic that "would virtually never be seen on a production network. (Ranum, 2001)" Ranum complains in particular about ARP spoofing attacks that most organizations separated from an attacker by a router would never have to face. His assertion is that A NIDS probably has better things to do than to detect an attack than can never reach most NIDS in the first place.

DARPA, MIT, and The Lincoln Labs

In 1998 DARPA contracted MIT's Lincoln Labs to monitor network sensors at the Air Force Research Laboratory and collect network attack data that was directed towards the Air Force network. Lincoln Labs documented and categorized various attacks that were found and created test sets of network traffic that contained both these network attacks and normal network traffic. These data

sets were then provided to intrusion detection researchers to test intrusion detection concepts and technologies.

Many developers have used attack data available through MIT's Lincoln Labs to test their product's detection capabilities. The data is more realistic than the overly hostile data captured at DefCon. It is also very helpful that actual normal traffic samples are provided. Realistic network traffic allows at least a rudimentary determination of an NIDS product's false positive rate.

Though this effort was a good first step in evaluating the performance of intrusion detection technology, it did not provide a detailed methodology for testing intrusion detection systems. Indeed it is not clear exactly how MIT used this data to test various NIDS systems. John McHugh of Carnegie Mellon's SEI states this in his 2000 critique of this methodology when he says, "The appropriateness of the evaluation techniques used needs further investigation"(McHugh 2000). This data is now fairly dated and is often used by the designers of the new NIDS system throughout the development process. Thus results from these tests often are not indicative of a product's performance on an actual network.

What about NIAP?

The National Security Agency (NSA) and The National Institute of Science and Technology (NIST) created the National Information Assurance Partnership (NIAP) in the late 1990's to assess the technical security of information technology products including NIDS'. NIAP employs testing and processes that

are in compliance with the International Standard ISO/IEC 15408:1999, Common Criteria for Information Technology Security Evaluations.

The NIAP methodology, however, has not been widely embraced by either the producers or consumers of NIDS technologies. There are probably several reasons for this.

Traditionally, the NIAP process has focused on assessing the security in information technology products, not on assessing the functionality of information security products. This focus has made the NIAP process less than ideal for assessing the effectiveness of competing NIDS methods and technologies

The NIAP testing process can be somewhat confusing to people initially confronted with it. This is especially the case for companies approaching NIAP testing for the first time. NIAP does not conduct any evaluations itself, but instead certifies commercial laboratories to conduct tests on its behalf. A NIAP laboratory does not necessarily have any set criteria to judge a product against. A NIAP laboratory's job is to assess whether a product complies with either a vendor's written claims (security target) or a customer's written requirements (protection profile). It is quite possible for a vendor to submit a product for

evaluation based upon a formal document of product claims (security target) and be found to be within compliance with respect to this document.

Because of this, it can be very difficult for a consumer to assess the merits of this product without first taking the time to read and understand the claims a vendor is making within its security target. If this sounds onerous and time-consuming for a potential consumer, it is. Even more problematic is the difficulty in comparing this product to another similar product, which has likely been submitted with a completely different security target.

In an attempt to make the NIAP process more useful in comparing similar IT technologies, the federal government has written several protection profiles or formal requirements for various IT products that it routinely purchases. These protection profiles are fast becoming the default criteria that all vendors are using to test their products against. There currently is a standard protection profile version 1.1, dated December 10, 2001 that details the federal government's requirement for an NIDS system. In order for a federal government agency to be able to purchase an NIDS system, the product must be in compliance with this protection profile. Unfortunately, this protection profile currently focuses more on the security and auditing associated with the actual NIDS device, rather than with the ability of the device to perform intrusion detection. This is understandable considering NIAP's traditional focus. Besides, the security architecture of the actual device is much easier to assess than the ability to perform a mission as complex as intrusion detection.

The NIAP process was designed to be extensible. Vendors submit suggested revisions to the federal government's protection profiles through the Information Assurance Technical Framework (IATF); so there is the possibility that the Intrusion detection protection profile will become more useful in the future.

Intrusion Detection Systems (IDS) Group Test (Edition 3)[1]

Developed by The U.K. based NSS group, this test suite provides a standard methodology that can be used by the NSS group to assess the performance of network based intrusion detection systems. This test has recently been applied to five modern NIDS products.[2] These include Cisco Secure IDS 2.5 Model 4230, Entercept 2.5, Internet Security Systems RealSecure 7.0, NFR HID 2.0, Okena StormWatch 2.1, and Snort 1.8.6. NSS states in its literature "The NSS Group IDS Report is considered the

[1] Intrusion Detection Systems Group Test (Edition 3) is copyrighted by The NSS group 1991-2002

[2] One Host-based IDS product was also tested. Previous versions of the methodology were used to test several additional IDS's. Results are available for purchase at http://www.nss.co.uk/shop/index.htm

definitive guide to IDS "(NSS web page 2002). Whether this is true or not, the testing methodology appears to be relatively thorough.

NSS starts by setting up what it considers to be a standard network environment and places the NIDS product to be evaluated on it. Its first test procedure launches a series of captured and custom-written attacks against the monitored network and determines which attacks are detected by the NIDS. This is similar to what occurs in the Lincoln Labs test procedures. Unlike Lincoln Labs, however, the NSS attack data set is not made available to the public for further evaluation. The NSS data sets also do not appear to include normal network traffic to test each NIDS product on its false positive rate. This testing methodology should include this test, because a product's false positive rate is quite significant in determining a NIDS' detection effectiveness.

The next step in NSS' test methodology is to test each NID product under increasing network loads. This is a very useful test, because it is important to know if a NID can maintain its detection capabilities as network traffic increases. Ironically, the test methodology goes to great lengths to generate realistic network traffic for this test. It would not appear to be too difficult to use this same traffic to assess the product's false positive rate. NSS in a further test applies this concept of network load even further by generating large network sessions that stateful NIDS must inspect. NSS then measures the effectiveness of the NID in detecting the same attacks, while tracking numerous large network sessions.

Lastly the NSS test methodology uses a series of IDS evasion tools to determine a NIDS ability to continue to detect network attacks even when they are being cleverly disguised through fragmentation and other techniques. NSS details the various tools used, but does provide any insight into the exact procedures that it uses with these tools to test this aspect of a NIDS' capability.

In summary, the NSS testing methodology seems to be the most useful and thorough yet for measuring a NIDS effectiveness in detecting network attacks. Unfortunately, NSS does not provide enough detail about its methodology or its data sets for that matter, for other organizations to independently verify their results. It also makes it very difficult to compare this methodology to other testing methodologies. As such, the NSS test suite is only of limited use to organizations assessing newer IDS techniques and technologies that are not ready to pay NSS to perform an assessment.

Open Security Evaluation Criteria (OSEC)[3]

[3] Open Security Evaluation Criteria (OSEC) is a registered trademark of Neohapsis Inc.

Neohapsis Inc of Chicago, Illinois has recently developed a methodology that promises to address the lack of repeatable, relevant, and objective testing in the NIDS product space. Neohapsis is a consulting company that has a great deal of experience with assessing and installing NIDS systems. Over time they developed a set of criteria for assessing NIDS systems that is designed to be used by the entire intrusion detection industry. To accomplish this, Neohapsis published their entire testing procedures on the Internet and sought comments from industry and other interested parties. The result is an open-source like testing methodology that all parties are free to use. Of course to be become an OSEC certified product, a vendor must submit his product to Neohapsis for testing. They have to make money some how. Still this is preferable to the relatively closed methodology that NSS currently employed.

The testing procedures employed by Neohapsis are also more comprehensive and rigorous than anything I have currently witnessed. To detail all the tests here would be nearly impossible. The reader is encouraged to go to http://osec.neohapsis.com to fully comprehend the detail of the test suite. However, a brief overview of OSEC's eight major testing categories will be helpful in understanding its depth. The categories follow:

1) Device Integrity Checking (sensor) tests
These tests verify that the sensor itself is not easily subject to compromise or Denial of Service (DoS) attacks. These tests are mandatory.

2) Signature Baseline Tests
These test verify that the sensor picks up the basic attacks used throughout the testing procedure under minimal background traffic conditions. Since all the OSEC testing uses real attacks against victim hosts, this indicates a signature failure.

3) State Tests
These tests verify that the sensor has a stable state table sized for the traffic levels it is designed to monitor. A network-based IDS (NIDS) should be able to track a number of sessions. In addition, it should be able to validate that state is correctly monitored and preserved both when monitoring existing traffic, and when under stress.

4) Discard Tests
These tests verify that the sensor does not use significant resources while handling traffic that does not match any monitoring rule in its sniffing rules. To achieve high speeds, most NIDS sensors discard traffic that falls outside their signature set. These tests assess the ability of the sniffing portion of the IDS to hand off possibly significant traffic to the rule-processing portion of the tool while under various traffic loads.

5) Engine Flex Tests
This set of tests access the sensor's ability to recognize attacks when under maximum traffic conditions. These tests are designed for stressing specific inspection modules with background traffic that is valid on regularly monitored ports while valid attacks are injected.

6) Evasion Tests
These tests assess the sensor's ability to recognize attacks that are sent through various evasion mechanisms. These tests are designed to verify the sensor's ability to deal with currently published means to evade network-based IDS sensors.

7) Inline/Tap Engine Tests
This test suite assesses the sensor's ability to recognize attacks when reading traffic in-line or from a tap. These tests verify the engine's ability to re-integrate directional streams and to cope with traffic that exceeds half-duplex fiber speeds.

What is truly unique about the OSEC process is the willingness of Neohapsis to disregard certain tests in their methodology if a product vendor chooses not to undergo them. This fact is of course reflected in the evaluation report. This allows vendors to customize the test to fit the capabilities that they are advertising. This kind of test achieves some of the customization of criteria adopted by the NIAP process without abandoning a comparable baseline for performance all together.

Neohapsis envisions making the OSEC process a true open standard that will be completely open to vendor and user critique. This is already occurring as Neohapsis begins the work of crafting version 2.0 of the OSEC NIDS testing methodology. Neohapsis envisions making each successive version of the criteria tougher and with vendor/user comments it is likely to get even better.

Two NIDS products have currently finished OSEC testing. They are Intruvert's Intrushield and ISS' real Secure 7.0. Both these NID products have fared quite well in the tests. This is not surprising since the test criteria are available for all vendors to see ahead of time. Neohapsis hopes that vendors will improve their products based on the OSEC criteria even before submitting products for testing. If these two products are any indication, Neohapsis' plan may be working. Both Intrusion, Inc and SourceFire have products that are in the process of testing.

Other

For the sake of completeness I include a reference to the "OPEN SOURCE SECURITY TESTING METHODOLOGY MANUAL" that is attempting to do what Neohapsis has done with OSEC using the open source community. For details please see http://www.ideahamster.org/osstmm-description.htm. This effort is ambitious in scope and is trying to define a methodology for all aspects of security testing. As such, the IDS test section by necessity is much more general in nature than OSEC's detailed processes. Still this could be a good first step if anyone actually pays attention to this initiative.

Don't Confuse playing a Violin with Music

There are several IDS testing tools on the market that purport to test an IDS' performance. These tools for the most part consist of packet generators, IDS evasion tools, and in some cases canned attack scenarios. These tools include products such as NIDSbench, http://packetstorm.widexs.nl/UNIX/IDS/nidsbench/nidsbench.html, Whisker, Tcpreplay etc. A more detailed list of IDS testing tools can be found at http://www.ideahamster.org/tools/ids.shtml. Unfortunately none of these tools currently provide a true test of a NIDs product. These tools have little intelligence built in and rely on the user to determine how they are to be used. Indeed a user who has a well-thought out testing methodology such as that provided by OSEC could use these tools to aid him in conducting meaningful tests. Indeed this is hopefully what will develop in the future as the OSEC standard gains more exposure and acceptance. A user without such a robust testing methodology, however, can expect little more than confusing noise from these tools.

In Conclusion

The problem of evaluating network intrusion detection technology has been around as long as the problem of how to detect network intrusions. For most of intrusion detection's brief history, there really has not been any measurable way to assess the relative or even individual effectiveness of a NIDS product or technology. This has led to confusion among consumers and a lack of direction among researchers and vendors. What is needed is an expansive, open testing methodology that can be altered and improved by all concerned parties in the intrusion detection community. This methodology needs to have buy in from both vendors and consumers. Seeking their inputs and making changes to the methodology that reflect current changes in the NIDS product space will go a long way in achieving this. Only OSEC seems to have grasped this concept. As such, OSEC may be the best chance for solving the "NIDS challenge".

References

Alessandri, Dominique. “Using Rule-based Activity Descriptions to Evaluate Intrusion-Detection Systems.” IBM Research, October 2000.

The Idea Hamster Organization. “About The Open Source Security Testing Methodology Manual.” 2001. URL: http://www.ideahamster.org/osstmm-description.htm.

Lippmann, Richard. Haines, Joshua. Fried, David. Korba, Jonathan. Das, Kumar. “The 1999 DARPA Off-Line Intrusion detection Evaluation.” Lincoln Laboratory, MIT, September 1999.

McHugh J. “Testing Intrusion Detection Systems: a Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory.” ACM Press, New York, 2000.

NIAP. September 2002. URL: http://niap.nist.gov/

NSS Inc. “IDS Performance Testing.” September 2002. URL: http://www.nss.co.uk/

Puketza, Nicholas. Zhang , Kui . Chung, Mandy. Mukherjee, Biswanath. Olsonn, Ronald. “A Methodology for Testing Intrusion Detection Systems.” University of California, Davis. September 1996.
URL: http://citeseer.nj.nec.com/puketza96methodology.html

Ranum, Marcus. “Experiences Benchmarking Intrusion Detection systems.” NFR Security, December 2001. URL: http://www.nfr.com/publications/request-form.html

End of Assignment #1

Network Detects –Part 2

Trace #1 Packets from the great beyond targeting lpd port

1. Source of Trace:

www.incidents.org/logs/Raw/2002.6.14

2. Detect was generated by:

Snort using an unknown rule set in tcpdump binary format. These files have been read into Ethereal and the output of one such mysterious packet follows.

Frame 23 (60 on wire, 60 captured)
   Arrival Time: Jul 13, 2002 21:06:49.364488000
   Time delta from previous packet: 211.620000000 seconds
   Time relative to first packet: 3431.950000000 seconds
   Frame Number: 23
   Packet Length: 60 bytes
   Capture Length: 60 bytes
Ethernet II
   Destination: 00:00:0c:04:b2:33 (00:00:0c:04:b2:33)
   Source: 00:03:e3:d9:26:c0 (00:03:e3:d9:26:c0)
   Type: IP (0x0800)
   Trailer: 000000
Internet Protocol, Src Addr: 255.255.255.255 (255.255.255.255), Dst Addr: 46.5.42.167 (46.5.42.167)
   Version: 4
   Header length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..0. = ECN-Capable Transport (ECT): 0
      ....0 = ECN-CE: 0
   Total Length: 43
   Identification: 0x0000
   Flags: 0x00
      .0.. = Don't fragment: Not set
      ..0. = More fragments: Not set
   Fragment offset: 0
   Time to live: 14
   Protocol: TCP (0x06)
   Header checksum: 0x5c27 (incorrect, should be 0x5422)
   Source: 255.255.255.255 (255.255.255.255)
   Destination: 46.5.42.167 (46.5.42.167)

Transmission Control Protocol, Src Port: 31337 (31337), Dst Port: 515 (515), Seq: 0, Ack: 0, Len: 3
    Source port: 31337 (31337)
    Destination port: 515 (515)
    Sequence number: 0
    Next sequence number: 3
    Acknowledgement number: 0
    Header length: 20 bytes
    Flags: 0x0014 (RST, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .1.. = Reset: Set
        .... ..0. = Syn: Not set
        ....0 = Fin: Not set
    Window size: 0

3. Likelihood Source address was spoofed:

The source address is 255.255.255.255. I'd put the probability near 100% that it was spoofed. I guess the source is hoping that I respond back to the whole world? The broadcast address is not a valid source address for a packet unless this packet was a response to some stimulus. Actually, even if it were in response to a stimulus the source address would never be 255.255.255.255. It would always be the host's IP address even if it were one of the hosts receiving a packet addressed to 255.255.255.255. One possible reason for using this address might be to ensure that there is no way of tracing where the packet came from.

4. Description of the attack:

The attack is targeting port 515, presumably the LPD printer daemon. It resembles a scan in that it seems to be targeting random source addresses in this subnet at various times. The packet appears to be crafted. It uses a TCP flag combination of Rst/Ack, which should never be seen in normal TCP communications. It uses the elite source port of 31337 and its source address is 255.255.255.255 and the packet contains no data.

With a source address of 255.255.255.255, the sender is not going to get any response back. This assumes that the sender is outside my local area subnet because properly configured routers will block packets sent out to the universal broadcast address.

What if the sender is on my local segment? If this were true, responses to 255.255.255.255 would be visible to him. However he is likely not on my local segment as the Ethernet addresses clearly show.
The Source Ethernet address of this captured packet, 00:03:e3:d9: 26:c0 indicates that the packet came from a Cisco device, probably a router (http://standards.ieee.org/regauth/oui/oui.txt). This means there are three possibilities. The router was responding with this really odd packet addressed from the elite port to the printer service port with packets containing no data; the packet is being routed by the router from the outside; or some one was on this local segment and was spoofing the MAC address of the router.

Why couldn't this be a host masquerading as a Cisco router? The destination MAC of the packet is not the destination host specified by the IP address but another intermediate Cisco device, 00:00:0c:04:b2: 33. This is likely another router or a Cisco cable modem concentrator (This was Bryce Alexander's theory at least). Therefore it seems that we are capturing the packet on a network segment that connects two Cisco devices. This is unlikely to have hosts on it - the link between two Cisco devices is usually point-to-point. Thus it seems that either the router forwarded this packet to our host from outside our local network or the router itself is using source port 31337 to query our printing service. I can think of no reason a router would do this, so it is most likely that this packet came from outside our network.

If this packet came from outside our network it probably was not in response to a packet we sent. At least not due to any legitimate traffic we sent. If our host were attempting to communicate with a Trojanized box somewhere, this packet might be a reasonable response from that Trojan. However, it is likely that the Snort rules would also flag traffic we sent from our port 515 to their port 31337 soliciting this response. Since I did not see this in the logs it seems less likely. Of course if the response were timed to wait for a couple of days this still might be possible. Either way we have problems.

I do not believe that this is a scan. There can be no reasonable expectation that results are going to be returned to an external host. The alternative supposition is that there is a new Trojan out there or a variant of the common port 515 Trojans, Kork or Ramen that is listening for this trigger packet to spring into action. I have found no literature concerning this possibility, so at this point I will have to leave it to further study.

5. Attack mechanism:

There really does not appear to be any CVE associated with this traffic. There are plenty of CVE's associated with lpd daemon exploits such as CVE-2001-0353, which details a buffer overflow in Solaris 8 systems. None of these CVE's seems applicable, however. This really seems to be a fairly ineffective attack. NO TCP session is established that we can detect. If this is a trigger packet for a

Trojan residing on our network it might be quite clever. It is so obviously crafted and appears so unlikely to work that analysts may very well dismiss it as an errant scanning attempt or a poorly constructed annoyance. Of course a good stateful firewall should block this traffic, but not everyone has one of those.

6. Correlations:

Others have seen such traffic. In http://online.securityfocus.com/archive/19/187958, a user attributes the traffic to a script kiddie that doesn't know what they are doing. Users in a discussion at http://lists.jammed.com/incidents/2001/07/0017.html have come much the same conclusion. The packets they describe and puzzle over are identical to the packet detailed above.

7. Active Targeting:

This "scan" seems to targeting random hosts in this subnet with little hope of receiving a response from a listening host. There appears to be no active targeting.

8. Severity:

Severity= Criticality + Lethality-System Counter Measures – Network Counter Measures.

Criticality = +1 (various systems of unknown function)
Lethality: = +1 (attack is very unlikely to succeed even if a new Trojan was in the wild that listened for such packets.)
Countermeasures
System: = -1 (lets assume there aren't any)
Network: = -1 (lets assume there aren't any… there obviously does not appear to be any stateful firewalls in the path)

Severity = 0

9. Defense Recommendation:

Port 515 should be blocked at the boundary router or firewall. If remote print jobs need to be sent from outside the boundary, ensure all relevant vendor patches are applied to the lpd daemon and the underlying operating system that the daemon is running on. Port 31337 is being flagged by Snort, but this does little to actually protect against a new Trojan that might be residing on our network. A stateful firewall put in place at the boundary of this enclave would make this attack very unlikely to succeed. It would block all these unsolicited packets sent to port 31337 while allowing packets that are part of legitimate TCP sessions that just happen to be using the ephemeral port 31337 to pass through. The firewall

could also be configures to block all traffic from source address 255.255.255.255 to make absolutely sure this traffic does not pass through.

10. Multiple choice question:

The lpd W0rm or Kork worm, as it also is known, was originally scripted by a 19 year old Australian to do the following

a.) Exploit Windows NT 4.0 vulnerability CVE –2000-0876
b.) Exploit Linux Red Hat version 7.0
c.) Create two new privileged accounts Kork And Kork2 on the host
d.) Disable the Solaris 8 in.lpd daemon
e.) Both b and c
f.) Both c and d

Answer = e

11. Responses to questions posted at incidents.org

Date of post: 4 September 2002.

I did not receive numerous responses to my post but I think I can cobble together three questions to answer.

The first question comes from Bryce Alexander our discussion follows:

Mr. Alexander

Another thing to consider is that even though the destination port is 515, another device on the same network could be listening promiscuously and will execute certain commands depending on the payload of this packet. In this case the packet contains the letters cko, this could be a "phone home" command. Has anyone seen anything other than "cko" in the data portion of these packets?

I am not so sure I would dismiss this out of hand. Remote control commands have been hidden quite well in the past.

Me

I am by nature paranoid also. All these packets seem to contain 6 extra bytes. As you noted this packet contained "cko" the others I looked at contained other seemingly random characters. Thinking about it further, if
I were programming a hidden packet to trigger a listening process, the last thing I would do is use the port 31337 or the source address

255.255.255.255. Even the most ill configured IDS and the most junior analyst would pick up on this. However, maybe the packet's author is flaunting his/her skill and the IDS community. By crafting what seems to be such an obviously crafted packet, maybe he is counting on the fact that we look at it and dismiss it as a poor packet-crafting attempt. Maybe he is hiding it best by exposing it openly. Another possibility I hadn't considered before is the fact that the author could be aware of a separate TCP/IP stack on a listening machine that has been altered from that used by the operating system. Basically it would intercept the packets from the
NIC driver before they proceeded up the operating system's TCP/IP stack. In this way he could carry on illicit conversations flouting TCP/IP's rule structure. The boundary router would get in the way of responding back to 255.255.255.255, but there is nothing stopping an altered host from responding back to a pre-arranged valid address.

There is also the possibility this detect occurred on a network attached by a cable modem and/or DSL line. I have heard that DSL and cable providers are not always good at configuring their border routers to stop private addresses /broadcast addresses from being forwarded. (I am sure other ISP's are similarly guilty). If a person on a DSL network sent this packet to others known to be on the company's DSL network, it might be a very effective Denial of Service attack against other members using that company's DSL/cable service. This would be contained upstream of the DSL/cable provider by a properly configured router, but there are probably a fair number of people that could be affected.

The second question also comes from Bryce Alexander:

Mr. Alexander:

If it were a denial of service you would expect to see a large number of packets in a short time, did your data support that?

Me:

There certainly were several packets directed at various hosts in this subnet, however, not enough to be a denial of service attack directly. Suppose that each of these hosts on my subnet responded back to the 255.255.255.255 with their very own reset packet. By rights a host should not respond back to a reset packet directed to them from 255.255.255.255, but host stacks have done dumber things. If each of these hosts responded to the reset packet, they would respond to every host on the local subnet. In my logs there certainly were enough queried hosts that a response by each of them to all hosts on the subnet would generate a fair amount of traffic on the local segment. If the local ISP did not do a good job configuring their internal router, the propagation could extend further then this local subnet and cause even more disruption. I found another victim of this traffic that certainly seems to think it is a DDOS attack that is attempting to use this multiplicative effect.

(http://lists.jammed.com/incidents/2001/04/0092.html) Of course their traffic has the Syn flag set, instead of the RST flag that I witnessed. This would make it a more effective Ddos attack. I wish my packets had the Syn flag set instead.

The third question comes from Bob Fitton

Bob Fitton wrote:

I'm coming in late, and missed some of the thread, but... IF these packets were successful in generating a reply, that reply would go to IP address 255.255.255.255 at tcp port 31337. Thus it would/could/might be an attempt to wake up any and all trojaned local systems. Plausible?

Me:

Wow this would be really devious. If the above stated Ddos theory is possible, this should be possible too. The RST packets that would be sent out to every host on the subnet could not contain any " hidden" instructions since they are presumably coming from uninfected, normal hosts. Therefore the Trojans would have to be preprogrammed to initiate independent action or to establish communication once it received a RST packet directed to its "elite" port. I guess this would make it very difficult to trace the person who triggered the Trojan. The guilty party would appear to be one of the innocent hosts responding to the initial bogus broadcast address. Since the person went to the trouble of using 255.255.255.255 as the source address and all the rest of this, it is doubtful the Trojan would establish communication with this person. This would give him away and negate all this deception. Instead the Trojan would be preprogrammed to undertake an independent action.

Trace #2 Packets targeting web server

1. Source of Trace:

www.incidents.org/logs/Raw/2002.6.14

2. Detect was generated by:

Snort using an unknown rule set in tcpdump binary format. These files have been read into Ethereal and the output follows:

Packet #1

```
Internet Protocol, Src Addr: 194.78.59.253 (194.78.59.253), Dst Addr: 46.5.34.28 (46.5.34.28)
   Version: 4
   Header length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
```

.... ..0. = ECN-Capable Transport (ECT): 0
....0 = ECN-CE: 0
Total Length: 40
Identification: 0xcc09
Flags: 0x00
.0.. = Don't fragment: Not set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 47
Protocol: TCP (0x06)
Header checksum: 0x7861 (incorrect, should be 0x715a)
Source: 194.78.59.253 (194.78.59.253)
Destination: 46.5.34.28 (46.5.34.28)
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 80 (80), Seq: 372, Ack: 0, Len: 0
Source port: 80 (80)
Destination port: 80 (80)
Sequence number: 372
Acknowledgement number: 0
Header length: 20 bytes
Flags: 0x0010 (ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 1400
Checksum: 0x60e3 (incorrect, should be 0x59dc)

Arrival Time: Jun 8, 2002 20:55:30.754488000

Packet # 2

```
Internet Protocol, Src Addr: 202.29.28.1 (202.29.28.1), Dst Addr:
46.5.52.5 (46.5.52.5)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 40
    Identification: 0x012b
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0x4a84 (incorrect, should be 0x437d)
    Source: 202.29.28.1 (202.29.28.1)
    Destination: 46.5.52.5 (46.5.52.5)
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 80 (80),
Seq: 1017, Ack: 0, Len: 0
```

```
Source port: 80 (80)
Destination port: 80 (80)
Sequence number: 1017
Acknowledgement number: 0
Header length: 20 bytes
Flags: 0x0010 (ACK)
    0... .... = Congestion Window Reduced (CWR): Not set
    .0.. .... = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .0.. = Reset: Not set
    .... ..0. = Syn: Not set
    .... ...0 = Fin: Not set
Window size: 1400
Checksum: 0x64a2 (incorrect, should be 0x5d9b)
```

3. Likelihood Source address was spoofed:

The source address is 194.78.59.253/ 202.29.28.1. According to DShield.org these hosts have a nasty reputation and have been reported several hundred times. Port 80 is one of the ports they are known to target. These packets and their brethren appear to scanning for active port 80/ web servers throughout the subnet. In order for the scan to be successful, a valid source address must be provided for responses to be returned to. These addresses could certainly be spoofed, but then the attacker would not receive any of the packets that this scan is generating because a TCP session would not have been established. Based on these facts, I do not believe that this address is spoofed.

4. Description of the attack:

The attack is targeting port 80, presumably the HTTP daemon. It resembles a scan in that it seems to be targeting different source addresses in this subnet at various times. The scans come from a few different IP addresses - the above are only an example. The pattern over the period of several hours seems to be that each address scans a particular host 2 or 3 times within about 5 seconds. The packets are practically identical with the exception of the ip address information. Every packet has the Ack flag set and contains no HTTP data.

The other offending IP's also have negative records according to Dshield.org that are very similar in nature. This might be a coordinated scan from different hosts all under the control of a single entity. Such a scan would be stealthier than one that originated from the same IP address. IDS rules often have thresholds that alert based on number of connections that occur between hosts in a certain time period. This might be a method of avoiding IDS detection. By having a source port of 80, these packets are trying to pass through many packet filtering firewalls

or ACL's that do not filter out web connections –especially web connections that appear to be the result of a client query (i.e. The Ack flag). If you are providing a public web service you often will allow connections with destination port 80 in even if you are using a stateful firewall, however the ACK flag should allow a properly configured stateful firewall to screen out the traffic since a previous SYN packet in the firewall's state table will not match it.

Likely the perpetrator has decided to target publicly accessible web servers in an organization's DMZ, where firewall protection is likely to be minimal for performance reasons. In this way an attacker can conduct reconnaissance on our network and determine which hosts are running web daemons and which are not protected by sophisticated firewalls

5. Attack mechanism:

The attack seems to work by attempting to establish a TCP connection on port 80 with several hosts on our network. 2 network connects are attempted in a very short period of time by 2 different hosts to the same box on our network. In this way 4 connections are attempted in a period of seconds to the same host. I assume the hope is that at least one of these packets will generate a response that will indicate if a web service is present.

I considered the possibility that these packets might be actual acknowledgements to SYN packets that my hosts were sending. Perhaps my hosts were the ones performing the scanning. However, as the sample packet shows, the ACK number is 0. Any legitimate ACK packet response would have a value other than 0. Therefore I believe this packet is a crafted packet sent unsolicited to these hosts and not a response to a malicious packet sent from my hosts.

Still these packets do not appear to be a scan searching for web daemons. These scans are very directed. It is true that they are designed to evade simple firewalls and filtering devices, yet they do not appear to be searching ranges of addresses looking for web servers. Instead, they appear to be directed to several specific IP addresses. Based on research I have conducted, I have concluded that these packets are actually being sent by load balancing devices that are attempting to establish the fastest paths from external networks to web servers on my network. These packets are crafted and appear just like a legitimate scan to the IDS. The negative reports attributed to these addresses at Dshield.Org are likely due to people misinterpreting this behavior as hostile. See Correlations for further details. I don't believe his traffic is trying to be malicious and probe for web servers in protected enclaves. They have set the Ack flag to 0 instead of something that might be mistaken for a legitimate value. I believe this is done on purpose to allow more sophisticated firewalls to screen out these packets before they reach web servers in protected enclaves. The purpose of these scans is to

determine optimum paths to available web servers. They are not really concerned in determining optimum paths to servers that are not going to be available for public web browsing.

6. Correlations:

I found this correspondence on an old Incident.Org thread. Here it is in its entirety.

Date: Fri, 26 Apr 2002 08:50:03 -0400
From: Chris Brenton
Subject: Re: Anyone else seeing TCP ACKs on port 80?

Barry Dorrans wrote:
>
> I've getting a rash of these over various servers recently. It seems to
> cycle, and doesn't come from a set IP.
>
> For example, on the 24th I had ACKS from
>
> 12.150.55.x

Hard to say without seeing the full IP address, but my guess is you are looking at an outbound load balancing device working over multiple ISPs. Here is what I've seen from one of the IP ranges you mention:

```
Mar 26 11:59:31 gw2 kernel:  DROP_FORWARD  IN=eth0 OUT=eth1
SRC=12.150.55.120 DST=12.33.246.130 LEN=40 TOS=0x00 PREC=0x00 TTL=55
ID=23194 PROTO=TCP SPT=80 DPT=53 WINDOW=1400 RES=0x00 ACK URGP=0

Mar 26 11:59:31 gw2 kernel:  DROP_FORWARD  IN=eth0 OUT=eth3
SRC=12.150.55.120 DST=12.33.247.6 LEN=40 TOS=0x00 PREC=0x00 TTL=55
ID=23202 PROTO=TCP SPT=80 DPT=80 WINDOW=1400 RES=0x00 ACK URGP=0

Mar 26 11:59:36 gw2 kernel:  DROP_FORWARD  IN=eth0 OUT=eth1
SRC=12.150.55.120 DST=12.33.246.130 LEN=40 TOS=0x00 PREC=0x00 TTL=55
ID=23388 PROTO=TCP SPT=80 DPT=53 WINDOW=1400 RES=0x00 ACK URGP=0

Mar 26 11:59:37 gw2 kernel:  DROP_FORWARD  IN=eth0 OUT=eth3
SRC=12.150.55.120 DST=12.33.247.6 LEN=40 TOS=0x00 PREC=0x00 TTL=55
ID=23396 PROTO=TCP SPT=80 DPT=80 WINDOW=1400 RES=0x00 ACK URGP=0
```

Note the alternating pattern between the public DNS server (.130) and the Web server (.6). Also note the fixed source port of 80 (note this is almost identical to an nmap fingerprint packet except no options are set). Finally, they are an ACK with no session.

The source in question is looking to time the round trip delay between sending this packet from multiple source IPs, and receiving an ACK/RST. The fastest reply becomes the preferred route.

HTH,

If port 80 is truly running on my hosts, there is no reason to believe they would not send the ACK /RST packet described in this conversation.

7. Active Targeting:

This definitely appears to be targeted at web servers on my network. The coordination of several packets sent to the same host from different sources seems to indicate that the sender expects a response from these hosts on port 80. Previous more random scans or some sort of webbot has probably identified these hosts as publicly available web servers on my network. It is likely these hosts are targeting many hosts on the Internet in a methodical fashion. Therefore it is debatable whether my hosts are being any more actively targeted than any other web server on the Internet.

8. Severity:
Severity= Criticality + Lethality-System Counter Measures – Network Counter Measures.

Criticality = +3 (enterprise web servers)
Lethality: = +1 ("attack" is likely to succeed but is only going to lead at worse to a confirmation that this box is running HTTP.)
Countermeasures
System: = -1 (lets assume there aren't any)
Network: = -1 (lets assume there aren't any. we determined from the last example that stateful firewalls were probably not present)

Severity = 2

9. Defense Recommendation:

Continue to monitor this traffic to ensure it does not begin to lead to exploitation attempts. Scan the targets of this scan with NMAP to determine if they really are running port 80 services and if they are, check to see if they are really authorized by the organization to be running publicly accessible web services. Consider placing publicly accessible web servers in a DMZ and blocking port 80 from entering into the protected enclave. Ensure web servers in the DMZ are hardened with the latest patches.

10. Multiple choice question:
Employing this type of device can go a long way in preventing the above type of scan attempt (Ack scan) of web services.

a.) Boundary router

b.) Network based IDS
c.) Host based IDS
d.) Stateful firewall
e.) Demilitarized Zone (DMZ)
Answer = d

Trace # 3 My Network has Worms

1. Source of Trace:

www.incidents.org/logs/Raw/2002.6.14

2. Detect was generated by:

Snort using an unknown rule set in tcpdump binary format. These files have been read into Ethereal and the output of one example of the packet follows.

```
Frame 145 (113 on wire, 113 captured)
   Arrival Time: Jul  3, 2002 06:21:56.264488000
   Time delta from previous packet: 2.870000000 seconds
   Time relative to first packet: 37141.240000000 seconds
   Frame Number: 145
   Packet Length: 113 bytes
   Capture Length: 113 bytes
Ethernet II
   Destination: 00:00:0c:04:b2:33 (00:00:0c:04:b2:33)
   Source: 00:03:e3:d9:26:c0 (00:03:e3:d9:26:c0)
   Type: IP (0x0800)
Internet Protocol, Src Addr: 62.2.78.216 (62.2.78.216), Dst Addr: 46.5.180.145 (46.5.180.145)
   Version: 4
   Header length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..0. = ECN-Capable Transport (ECT): 0
      .... ...0 = ECN-CE: 0
   Total Length: 99
   Identification: 0xa3c6
   Flags: 0x04
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
   Fragment offset: 0
   Time to live: 110
   Protocol: TCP (0x06)
   Header checksum: 0xff63 (incorrect, should be 0xf95d)
   Source: 62.2.78.216 (62.2.78.216)
   Destination: 46.5.180.145 (46.5.180.145)
Transmission Control Protocol, Src Port: 2175 (2175), Dst Port: 80 (80), Seq: 3985497033, Ack:
2331480033, Len: 59
   Source port: 2175 (2175)
```

Destination port: 80 (80)
Sequence number: 3985497033
Next sequence number: 3985497092
Acknowledgement number: 2331480033
Header length: 20 bytes
Flags: 0x0018 (PSH, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 1... = Push: Set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 8760
Checksum: 0xda16 (incorrect, should be 0xbf01)
Hypertext Transfer Protocol
GET /scripts/..%5c%5c../winnt/system32/cmd.exe?/c+dir\r\n
Data (4 bytes)

3. Likelihood Source address was spoofed:

The source address is 62.2.78.216. According to DShield.org this host is a cable modem subscriber, but has no documented cases of ill behavior. Of course it is likely this cable modem user gets a new DHCP address every so often, so it is hard to know what he has been up to in the past. This user is trying to scan multiple addresses for vulnerable web servers, so it is very unlikely that this address is spoofed. With a spoofed address there is no way for TCP to establish its three-way handshake and thus a viable session with the host. The attacker is hoping to get responses back from his scans so he will most likely not spoof his address. This indicates that he doesn't care if he is detected which may indicate that he is conducting the scans from somebody else's box that he currently owns. After all, A smart hacker would not be so easily traced.

4. Description of the attack:

The attack is targeting port 80, presumably the HTTP daemon. Over a period of about 10 seconds the same host targeted seven different IP addresses on this monitored subnet and then sent no more detectable traffic for the rest of the day. This may indicate that the process running this scan is interleaving it among several different subnets. This is a common method for such scans to try and avoid detection, by reducing the amount of noise generated on any one network. The hosts were not scanned in sequential order, but the destination addresses appeared to be occurring in some sort of pattern. The scan of the destination host's last octet was 135, 134,145,151,153, 158, 250. I believe the attacker was attempting to ascertain if any of the IP addresses are hosting web servers that are vulnerable to a very specific exploit. It is issuing the same get request every time.

GET /scripts/..%5c%5c../winnt/system32/cmd.exe?/c+dir\r\n. The Cert.org web site identified this as an exploit for the "Directory Transversal Vulnerability."

5. Attack mechanism:

The attack seems to work by attempting to establish a TCP connection on port 80 with several hosts on our network. According to Cert.Org, the Get request is attempting to take advantage of the "Directory Traversal vulnerability." The Cert web site documents this vulnerability as affecting Windows IIS servers. This could be due to the Nimda worm, which is attempting to execute a command shell with the privileges of the Web server process. If Nimda can do this, it can then infect the system by copying a file/files with the following names to the web directory.

- root.exe
- admin.dll
- Getadmin.dll
- Getadmin.exe
- * *.eml

The signature for the Nimda virus however does not match exactly with the packets that I have collected. Nimda issues a get request with the signature GET **/scripts/..%5c../winnt/system32/cmd.exe?/c+dir ,** which slightly differs by one character from the signature that I obtained.

It is very possible that this packet is the result of another tool that is attempting to do something Nimda like. Regardless of the tool that generated it, the intent of the attack is the same as an initial Nimda infection. Specifically it is attempting to exploit unpatched Windows NT IIS servers that are vulnerable to what Microsoft calls a canonicalization (the %5 causes this in this example) error affecting CGI scripts. If an URL requesting a CGI script located in a folder were malformed in a particular way, the wrong permissions would be applied. Rather than applying the permissions for the folder that contains the requested file, those of a folder further up the tree would be applied. When certain types of files are requested via a malformed URL, the canonicalization yields an incorrect result. It locates the correct file, but concludes that the file is located in a different folder than it actually is. As a result, it applies the permissions from the wrong folder. If the scripts directory is on the C drive, the attacker can then execute any file on the C drive with the permission of IUSR_Machine. In our case he is executing the DOS command prompt. From the command prompt he can manipulate any files on the C drive he desires that IUSR_Machine has access to. He can also launch an ftp session to some box of his choosing to download malicious software that can be hidden on the compromised box. A backdoor on this system is almost assured to be installed if the attacker gains this level of access.

6. Correlations:

I found a similar match on the Get query I submitted on Cert.Org's web site http://www.cert.org/advisories/CA-2001-26.html. It describes the packets sent from a Nimda infected host to web servers on a given subnet. http://www.eeye.com/html/Research/Advisories/AD20001003.html suggests that this query might be from an attacker or attack script that is attempting to execute the IIS Unicode exploit. The vulnerability this is trying to exploit is identified on the CVE web site under number **CVE-2001-0333**
. Microsoft details the vulnerability and a patch to correct it at http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-057.asp.

7. Active Targeting:

This is targeting hosts on my network, but the attacker seems to be scanning IP addresses sequentially on my subnet and may be jumping off to scan other networks periodically; so there is no true awareness of the web servers residing on my network.

8. Severity:

Severity= Criticality + Lethality-System Counter Measures – Network Counter Measures.

Criticality = +3 (enterprise web servers)
Lethality: = +2 (attack could succeed if it finds a box on my network that is vulnerable. A compromised box would have to be taken off the net and rebuilt. The attack has been around for a while so the lethality probably has subsided)
Countermeasures
System: = -1 (lets assume there aren't any)
Network: = -1 (lets assume there aren't any as before)

Severity = 3

9. Defense Recommendation:

- Immediately verify the installation of the Microsoft IIS patch for Windows-based web servers, available at http://www.microsoft.com/Downloads/Release.asp?ReleaseID=32061(for Windows NT 4.0)
- Administrators should scan their network with nmap to determine where web servers reside. These servers should then be checked to see if they

are Windows boxes…they probably will have ports 135 and 139 open as well as 80/443. Apply Microsoft's URLscan/lockdown software to each of the identified IIS boxes to ensure they are not vulnerable to this attack. The tool is available at http://www.microsoft.com/windows2000/downloads/recommended/urlscan/default.asp.

10. Multiple choice question:

Nimda is not too proud to take advantage of vulnerabilities created by other worms. Specifically it attempts to exploit this/these other worms' handiwork

a.) sadmind/IIS worm
b.) Ramen worm
c.) Code Red II worm
d.) Worm.ExploreZip
e.) Both a & b
f.) Both a & c
Answer = f

References:

Works cited have been included in the text where appropriate. A brief synopsis follows.

Cert.org. September 2002. http:// www.cert.org/

Dshield.org. September 2002. http://www.dsield.org/

Incidents.org. September 2002. http://www.incidents.org/

Jammed.com. September 2002. http://lists.jammed.com/incidents/

Red Alert Inc. September 2002. http://www.atomictangerine.com

Security Focus. September 2002. http://online.securityfocus.com/archive

Microsoft.com. September 2002. http://www.microsoft.com/windows2000/downloads/

END of Assignment #

Analyze This – Part 3

Executive Summary

This report is the culmination of a five-day effort to monitor the security of the University publicly accessible network. IDS sensors were placed in such a way to capture traffic leaving and entering the university domain. Five days certainly does not constitute a rigorous analysis of the network's security posture. It does however, provide a digestible and enlightening insight into the type of behaviors that are currently jeopardizing the security of the university's network. For the purposes of this analysis, the authors have assumed that the security goals of the University's network focus on the confidentiality of sensitive university information, the availability of University computing assets, and the integrity of university information and reputation as it relates to its computer assets.

Between July 22 and July 27, there were a number of troubling activities that occurred on the university network backbone that deserve immediate attention. We have taken the time to carefully examine several of these behaviors to illustrate the potential threat that these behaviors may cause.

After reading this report, university managers will want to pay particular attention to correcting the security issues associated with following hosts:

U.Net.140.9 –Myserver Ddos agent
U.Net.178.199, U.Net.83.9, U.Net.117.20 – potential Subseven slaves
U.Net.150.240, U.Net.150.120, and U.Net.153.45 - likely Adore worm infection
U.Net.157.241, 246, 247 – Nimda worm infection
U.Net.117.27 – Nimda worm infection
U.Net.162.226 –potential back orifice slave

This is not the first such analysis conducted on the University's network. Several predecessors of ours have conducted similar analyses, which the University currently possesses. We will attempt to build on the efforts of these studies to provide additional insight into the University's information security posture. Where we borrow from previous studies, we will annotate this in the body of our report

Examining the Network logs

This analysis focused on 15 log files generated by the University's own intrusion detection system (Snort). The files are categorized as either alerts, scans, or out of specification (oos)[*]. Total number of alerts for the five-day period numbered 1,615,870 entries. The total number of documented scans for the same period was 6,486,866 entries. Oos totals were mercifully smaller at a mere 1647 entries.

alert.020722	scans.020722	oos_Aug.1.2002
alert.020723	scans.020723	oos_Aug.2.2002
alert.020724	scans.020724	oos_Aug.3.2002
alert.020726	scans.020726	oos_Aug.4.2002
alert.020727	scans.020727	oos_Aug.5.2002

Alerts and their Consequences

The size of the log files produced over the course of these five days is daunting. There is little doubt that true security vulnerabilities are identified within the logs. Also likely is that the vast majority of the traffic in these logs is the result of Snort rules triggering on permitted or at the very least relatively benign traffic. The challenge at hand is to separate out the majority of permissible traffic from that which directly compromises the security of this network.

A fairly good way to reduce the problem set quickly is to identify relatively benign services that are running on the University network that may or may not be sanctioned. Numerous analysts in past studies have identified the presence of file sharing programs such as Napster and Kazaa on the University network. It was not surprising to us that these services continue to be used and that they might be responsible for a proportion of the alerts that are in the log files. Many analysts in previous studies have identified these services as security problems that need to be addressed by University officials. It is our assertion that file sharing services have continued unabated since these analyses were first conducted and that University officials have at least tacitly accepted these services as permitted ones. Therefore we can eliminate these "permitted" services from further consideration in our security analysis.

Legitimate services such as email and web services have also been responsible for numerous alerts in past analysts studies - mainly due to TCP sessions that use legitimate ephemeral ports that match suspicious ports or that exceed some Snort defined threshold for number of connections. By using lists of known University servers that have already been identified by previous analysts, we can quickly identify traffic that is expected from hosts and what traffic is not. Lists of

[*] *Files from the 25th of July may not have been available due to complications arising from the riots that ensued on campus when the Chess team lost a semifinal match. Presumably oos files were not available until the following week for the same reason.*

these expected services can be found in reports submitted by Hee So, February 16, 2002 and K. Haugsness December 2, 2001. It is useful to point out that the University uses private network addressing within its network. Both 192.168. X.X and 10.X.X.X are used throughout the University. It is not clear whether the university employs network address translation to allow these hosts to communicate with the Internet as a whole or if these hosts are limited to internal University communications.

In light of these factors, we are now ready to consider the alert files generated by the University's IDS. Intrusion Alerts mean little without understanding the significance of the alert and without understanding what actions to take as a result of the alert. There are several alerts that require immediate attention. We will detail them here. Alert numbers correspond to each alert's respective position on the alert summarization chart that follows this discussion.

Alert #1 Nimda Alerts

The alerts that Snort generated are indicative of an actual compromise. This alert accounted for nearly half of the total log file compiled over the five-day monitoring period. The traffic all appears to be similar to this:

```
07/22-03:09:08.744707  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2288 -> 193.25.152.127:80
07/22-03:09:08.745289  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2289 -> 193.25.152.128:80
07/22-03:09:08.745328  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2290 -> 193.25.152.129:80
07/22-03:09:08.745589  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2291 -> 193.25.152.130:80
07/22-03:09:08.796201  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2293 -> 193.25.152.132:80
07/22-03:09:08.796210  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2294 -> 193.25.152.133:80
07/22-03:09:08.796750  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2295 -> 193.25.152.134:80
07/22-03:09:08.816048  [**] NIMDA - Attempt to execute cmd from campus
host [**] U.Net.157.246:2297 -> 193.25.152.136:80
```

Notice that a University host is systematically sending traffic to successive hosts on a selected subnet. This behavior is consistent with that exhibited by a Nimda infected host. The vast amount of this traffic and its obvious scripting indicate that is not likely to be the result of normal University communications. The Snort signature is triggering on get requests such as "GET /c/winnt/system32/cmd.exe?/c+dir" which would be hard to explain as anything except malicious behavior. See http://www.cert.org/advisories/CA-2001-26.html for an explanation of the type of vulnerability Nimda is attempting to exploit. Nimda is searching for other IIS boxes that have not been patched to infect. This is likely making the University less than popular in the Internet-wide community. The hosts that appear to be infected are U.Net.157.246, U.Net.117.27,

U.Net.157.241, and U.Net.157.247. We recommend disconnecting them from the University Network, scanning and cleaning them by an up to date virus scanner, removing any new accounts like “guest” that have been added, patching the server and monitoring each of them closely afterwards to see if they continue to exhibit any unusual behavior. A scan of all University IIS servers should be made with some tool such as Nessus to determine if any other servers are vulnerable to the Nimda exploit. If additional servers are vulnerable there is a good chance they have been infected also. An up to date virus scanner should be run against all IIS servers as well as Internet Explorer clients and Exchange email clients. Chances are several web client and email client machines are also infected if they run the Microsoft operating system. Similar actions as those stated above should be undertaken for each machine that has been infected. You can run an automated script that does much of the work for you at http://www.wileyc.edu/computer%20support%20services/software/download/antivirus/Symantec/Tools/Nimda/Nimda_tool.html Further details can be obtained at (http://www.thesitewizard.com/news/Nimdaworm.shtml). Of course you could just run Linux based clients… Linux is not vulnerable to Nimda.

Alert #5 Myserver Alerts

The alerts that Snort generated are indicative of an actual compromise. There is a percentage of the traffic that appears to be originating from a Gnutella server port. These are probably the result of legitimate file-sharing going on between a Gnutella server and a client using the ephemeral port 55850. The remainder of the traffic, however, is similar to this:

```
07/23-19:27:50.074197  [**] Port 55850 udp - Possible Myserver activity
- ref. 010313-1 [**] U.Net.140.9:55850 -> 137.99.92.20:33436
07/23-19:27:50.074695  [**] Port 55850 udp - Possible Myserver activity
- ref. 010313-1 [**] U.Net.140.9:55850 -> 137.99.92.20:33437
07/23-19:27:50.075167  [**] Port 55850 udp - Possible Myserver activity
- ref. 010313-1 [**] U.Net.140.9:55850 -> 137.99.92.20:33438
07/23-19:27:50.075852  [**] Port 55850 udp - Possible Myserver activity
- ref. 010313-1 [**] U.Net.140.9:55850 -> 137.99.92.20:33439
07/23-19:27:50.076316  [**] Port 55850 udp - Possible Myserver activity
- ref. 010313-1 [**] U.Net.140.9:55850 -> 137.99.92.20:33440
07/23-19:27:50.077040  [**] Port 55850 udp - Possible Myserver activity
- ref. 010313-1 [**] U.Net.140.9:55850 -> 137.99.92.20:33441
```

U.Net.140.9 is scanning methodically from port 55850 looking systematically at hosts on various subnets for what appears to be open RPC services. I cannot think of a legitimate reason for such large amounts of such carefully crafted traffic. http://archives.neohapsis.com/archives/incidents/2000-10/0136.html details how Myserver is a distributed denial of service agent that attempts to attack 1024 by means of several controlled “zombie” machines. It attacks variants of Unix and trojanizes the ls and ps commands. It also places a root kit in the /lib directory. Examining these files on the host are one of ways we can tell if this is truly a Myserver infection. Another way to determine if this box is infected with Myserver is to query ps for the 55850 service. A Myserver host’s

version of ps is trojanized and won't actually show the 55850 service running on the box. Netstat will however. If this is a Microsoft box, this mysterious process is something other than Myserver.

The method of how this DDOS Trojan spreads is not well documented. Apparently it is rare and has not been well studied. Whatever is running on this host appears to be scanning for vulnerable RPC services, so this may be one of the ways that Myserver spreads. Regardless of what the process is, we suggest taking this box off-line, removing the operating system and reloading from trusted media, applying relevant software patches.

Alert #6 Spp_Http ISS Unicode /CGI null byte attack

(Attacks Grouped together since they are generated by the same snort preprocessor function and result form the same basic Windows Directory Transversal vulnerability.)

The fear is that people are exploiting buffer overflows in web servers and hiding them from firewalls and IDS's through the use of Unicode characters. { http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise68 and Angela D. Orebaugh March 11, 2002

Specifically it is attempting to exploit unpatched Windows NT IIS servers that are vulnerable to a broad base of attacks called Directory transversal attacks. The IIS Unicode variety appends a unicode slash or backslash after a directory with execute permissions and is then able to run additional commands.

The CGI null byte attack takes advantage of what Microsoft calls a canonicalization (the %5 causes this in this example) error affecting CGI scripts. If an URL requesting a CGI script located in a folder were malformed in a particular way, the wrong permissions would be applied. Rather than applying the permissions for the folder that contains the requested file, those of a folder further up the tree would be applied. When certain types of files are requested via a malformed URL, the canonicalization yields an incorrect result. It locates the correct file, but concludes that the file is located in a different folder than it actually is. As a result, it applies the permissions from the wrong folder. If the scripts directory is on the C drive, the attacker can then execute any file on the C drive with the permission of IUSR_Machine.

Much of this traffic could be just part of a normal web session. It has been my personal experience that this alert can generate a large number of false positives. This makes it very hard to distinguish between normal and hostile web sessions. However, the host below has a record of attacking hosts with Dshield.Org so we should be suspicious.

```
07/22-03:27:22.804200  [**] spp_http_decode: IIS Unicode attack
detected [**] 130.60.242.52:2005 -> U.Net.100.158:80

07/22-03:27:23.061436  [**] spp_http_decode: IIS Unicode attack
detected [**] 130.60.242.52:2013 -> U.Net.100.158:80

07/22-03:27:23.298994  [**] IDS452/web-iis_http-iis-unicode-binary [**]
130.60.242.52:2022 -> U.Net.100.158:80

07/22-03:27:23.298994  [**] spp_http_decode: IIS Unicode attack
detected [**] 130.60.242.52:2022 -> U.Net.100.158:80
```

This alert is often the result of a false positive usually because it is returned when browsing Chinese or SSL encrypted sites. http://archives.neohapsis.com/archives/snort/2001-08/0075.html This could be the case for several of the alerts in our study (This university has a fair number of Chinese students). Still the University should examine the hosts identified as destinations by these alerts and ensure that they are not vulnerable to these exploits. A thorough Nessus or similar scan should be conducted on our hosts to ensure that they are not vulnerable. Examine U.Net.100.158 to determine if it's running a vulnerable version of IIS. The relevant patch has been available from Microsoft since August 2000 and is available at http://www.microsoft.com/windows2000/downloads/critical/q269862.
If this box is vulnerable there is a good chance that a back door has been installed. This probably means deleting the operating system and reloading from trusted media is the only way to obtain a reliable server.

Alert # 8 / 16 External RPC Calls Alerts/ Sun RPC High Port Access

These alerts are essentially synonymous. Both deal with external hosts scanning our network for the open portmapper service - Port 111 and 32771 in the Sun RPC case. This traffic is indicative of a deliberate attempt to determine what RPC services are running on the University network. It is difficult from these logs to determine what information was returned by these queries. Several RPC services are vulnerable to buffer overflow attacks. This can yield to a situation where an attacker can gain elevated access to a University host merely by taking advantage of buffer overflow vulnerability in an RPC service. Traffic such as this is troubling:

```
07/27-21:31:55.860142  [**] External RPC call [**] 24.95.192.71:1676 ->
U.Net.80.149:111
07/27-21:31:56.370351  [**] External RPC call [**] 24.95.192.71:1676 ->
U.Net.80.149:111
07/27-21:34:09.516394  [**] External RPC call [**] 24.95.192.71:1779 ->
U.Net.80.149:111
07/27-21:34:10.031920  [**] External RPC call [**] 24.95.192.71:1779 ->
U.Net.80.149:111
07/27-21:34:10.532449  [**] External RPC call [**] 24.95.192.71:1779 ->
U.Net.80.149:111
```

Since the pattern of the scans in our logs do not usually involve four packets being sent to the same address, this seems to indicate that the port mapper service of U.Net.149.111 may have responded back to the scanning host. Similarly, U.Net.99.179, U.Net.154.27, and U.Net.137.36 seem to have responded back to port scans of the Sun RPC service. At this point these hosts should be scanned by Nessus or other vulnerability scanner to determine what information they may be providing an outsider. They should also be scanned to see what RPC services they are running. All RPC services should be checked to ensure that they are either not vulnerable to exploit or have been patched. None of these RPC services should be allowed to be accessed from arbitrary hosts on the Internet.

Alert # 9 Possible Red Worm –Adore

These Snort generated alerts do not appear to be the result of false positives. We have three hosts that are sending back suspicious traffic from the port normally associated with the Adore worm, 65535. U.Net.150.240, U.Net.150.120, and U.Net.153.45 all are exchanging numerous packets with various hosts repeatedly. The numerous occurrences of the port 65535 in communications between these hosts and other external boxes make it very unlikely that this was the result of a random ephemeral port assignment.

```
High port 65535 udp - possible Red Worm - traffic [**]
63.250.205.8:9911 -> U.Net.150.120:65535
65535 udp - possible Red Worm - traffic [**] 63.250.205.47:57215 ->
U.Net.150.120:65535
65535 udp - possible Red Worm - traffic [**] 63.250.205.47:57215 ->
U.Net.150.120:65535
High port 65535 udp - possible Red Worm - traffic [**]
66.250.64.10:65535 -> U.Net.71.243:65535
```

Adore attacks known vulnerabilities on Linux such as RPC.statd and others documented in http://www.sans.org/y2k/adore.html. What we may be seeing in these logs however is not the infection phase, but rather the backdoor that Adore leaves after it has exploited its victims. The fact that numerous different boxes are communicating with hosts on our network using this port may mean that these backdoors have been leaked to the hacker community. Adore does leave a Trojanized ps file behind so it should be possible to confirm if these boxes have been compromised. Dartmouth has a utility that can automate this for you at http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm. An up to date virus scanner that can detect Adore should be run on these hosts and on all hosts to determine if there are Adore infections. The Dartmouth utility could also be run to remove this worm. Once a backdoor has been discovered, however, the truly safe solution is to reformat the affected systems hard drive and reinstall the operating system from a trusted media and then apply the necessary patches to each of the vulnerabilities listed in the SANS document above.

Alert # 11 Possible Trojan Server Alerts

The alerts generated by Snort definitely appear to be legitimate indications of hostile activity. Much of the traffic that generated these alerts is the result of very organized scanning for the port 27374, which is normally associated with the Subseven Trojan that affects Windows hosts. Three hosts in particular, U.Net.178.199, U.Net.83.9, U.Net.117.20 have initiated prolonged communications on this port and are suspicious. For example:

```
22-05:29:11.744644  [**] Possible Trojan server activity [**]
216.110.36.14:1829 -> U.Net.178.199:27374
07/22-05:29:11.745863  [**] Possible Trojan server activity [**]
U.Net.178.199:27374 -> 216.110.36.14:1829
07/23-05:50:19.569331  [**] Possible Trojan server activity [**]
216.110.36.14:2214 -> U.Net.178.199:27374
07/23-05:50:19.570432  [**] Possible Trojan server activity [**]
U.Net.178.199:27374 -> 216.110.36.14:2214
```

Such traffic is disturbing for it is indicated that hosts on our network have established sessions with external hosts using the 27374 port.

Most of the alerts from Snort were generated due to the scanning activity of external hosts directing packets at the 27374 port over a range of our hosts. This merely means that external attackers are probably searching for an existing Trojan on our network. The fact that they are searching means that they are probably not aware of a specific Trojan residing on our network. This is at least a little comforting. U.Net.178.199, U.Net.83.9, U.Net.117.20, however, seem to respond to the 27374 directed packets. What is worse is that the external host we seem to be communicating with in the example above is a known attacker according to Dshield.org and is one of the members on our Top Suspects list (which we detail further on in this study).

The site http://www.hackfix.org/subseven/ details all the problems caused by the Subseven Trojan. If hosts are indeed infected by this Trojan on the University network, it means they are completely controlled by outside attackers. These hosts will then allow an attacker to mount reconnaissance and attacks from inside the protected boundaries of the network. Such attacks are likely to evade notice by current IDS technologies that focus on the network boundaries. This should be corrected as soon as possible. The Trojan could be enabling all sorts of dangerous behavior that will likely not be detected. Updated virus scanning software should detect the Trojan and should be used on all University hosts to ensure that boxes are not infected. Infected boxes will probably need to be scrubbed and reloaded from trusted media, due to the possibility of additional hostile code being loaded on them.

Alert # 12 Connections to 515 from the Outside

This Snort alert provides some very necessary insight into the attempted attacks that are occurring on the University network. Two external addresses 64.30.217.125 and 24.123.46.10 seem to be conducting the majority of the scanning activity. This pattern is indicative of the scans:

```
07/22-12:15:27.845467  [**] connect to 515 from outside [**]
64.30.217.125:4361 -> U.Net.10.251:515
07/22-12:15:27.860028  [**] connect to 515 from outside [**]
64.30.217.125:4362 -> U.Net.10.252:515
07/22-12:15:27.884098  [**] connect to 515 from outside [**]
64.30.217.125:4364 -> U.Net.10.254:515
```

It would probably be advisable to block these two IP addresses from entering our network by placing ACL's to block them at the boundary device.

64.30.217.125 has a record with Dsield.org as an attacker and can be seen in our Top Suspects list further in this report to be engaged in extensive RPC scanning as well. This host also apparently tried to send an exploit against this service running on another host. See below (Explanation of the X86 exploit is found later in the alert summarization).

```
EXPLOIT x86 NOOP [**] 64.30.217.125:4634 -> U.Net.136.3:515
```

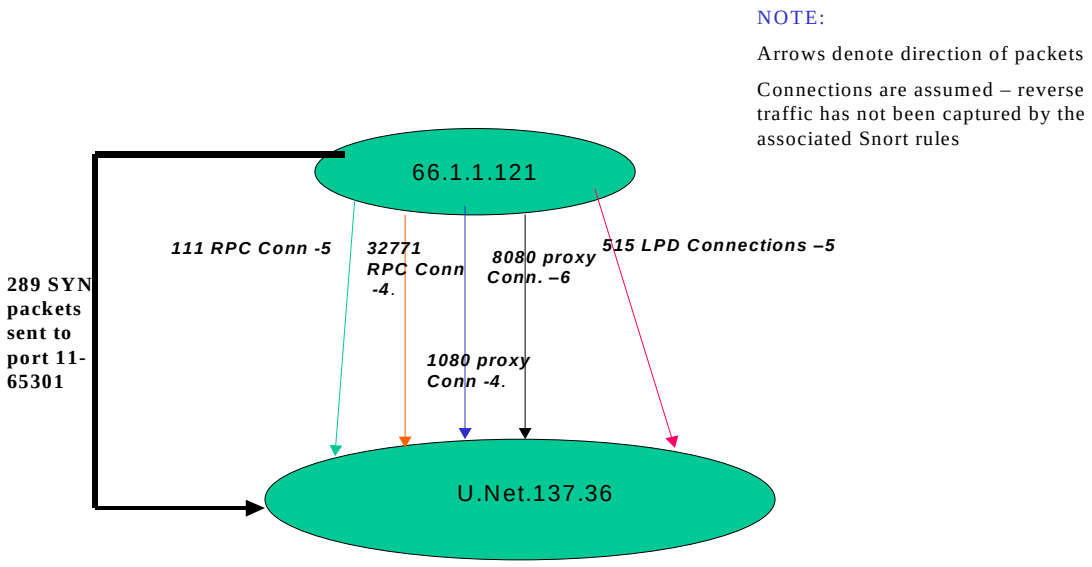
LPD services have a number of vulnerabilities associated with them. Since this is a Unix service, the hosts most likely to be susceptible will be running Unix variants. Several of these vulnerabilities allow an intruder to use a buffer overflow to execute arbitrary commands on the system with super user privileges. More information on these vulnerabilities can be found at http://www.cert.org/advisories/CA-2001-30.html .

One Host on our network seems to be accepting quite a few packets on its 515 port. It triggers this alert 5 times. This is different than the random scans we have seen up to this point. This indicates that a possible TCP connection has been established between U.Net.137.36 and 66.1.1.121.

```
66.1.1.121:4297 -> U.Net.137.36:515
66.1.1.121:4297 -> U.Net.137.36:515
66.1.1.121:4375 -> U.Net.137.36:515
66.1.1.121:4375 -> U.Net.137.36:515
66.1.1.121:4375 -> U.Net.137.36:515
```

There is the possibility that this is a legitimate LPD session set up between these two hosts. It is not wise to allow external hosts to connect to a local printer daemon, but this does not necessarily mean there is something suspicious going on. In fact after analyzing the nature of all the traffic alerts that 66.1.1.121 has sent to our network over this 5 day analysis period, it can be seen that it is all directed to U.Net.137.36 and could certainly seem to be legitimate sessions for

various services that 137.36 is offering. However after correlating with the separate Snort scan alerts, we detected a SYN scan that 66.1.1.121 initiated before establishing these "legitimate' sessions. This makes this traffic very suspect. See graph below.



Apparent connections between 66.1.1.121 and U.Net.137.36

Though we have no evidence that it has been exploited, U.Net.137.36 is potentially vulnerable to many different exploits that take advantage of externally available LPD, Proxy, and RPC services. The University should immediately scan this box to determine if it is vulnerable to either an RPC or LPD exploit. If the box is vulnerable, it probably has been compromised. A fresh rebuild from trusted media and application of all relevant software patches would be the only way to recover if this is the case. Even if the box is not compromised, the University should consider whether it is wise to allow services such as Proxy, RPC, and LPD to be accessed by external clients. Blocking these services at the boundary firewall would go along way in preventing several potential abuses from occurring.

Alert # 29 Back Orifice

There were only 2 of these Snort alerts in the entire 5-day monitoring session. What makes them suspicious is that they are not the result of an apparent random ephemeral port assignment. If they were, the port they were directed to should be a recognizable service port. "39849" is not registered with Snort.org

as a well-known service port so this traffic becomes suspect. The alert is pictured below.

07/22-09:07:24.199539 [**] Back Orifice [**] 66.129.222.70:39849 -> U.Net.162.226: 31337

It is certainly possible that this traffic is due to some innocent behavior that is occurring as the result of client responses from our host to some unknown service that is being run by the host 66.129.222.70. There are also other Trojans that employ the UDP port 31337. What makes this traffic truly suspicious is correlation with our top suspects list. 66.129.222.70 is a very bad person according to DShield.org. About the same time these UDP packets were sent to port 31337, a flurry of UDP packets is exchanged between the hosts with unrecognizable ports. These packets came from correlating the separate scan logs with both the alert files and the top suspect list.

```
Jul 22 09:07:00 66.129.222.70:37305 -> U.Net.162.226:19273 UDP
Jul 22 09:07:00 66.129.222.70:19032 -> U.Net.162.226:31676 UDP
Jul 22 09:07:00 66.129.222.70:60389 -> U.Net.162.226:34146 UDP
Jul 22 09:07:03 66.129.222.70:1922 -> U.Net.162.226:1967 UDP
Jul 22 09:07:04 66.129.222.70:0 -> U.Net.162.226:0 UDP
Jul 22 09:07:02 66.129.222.70:24657 -> U.Net.162.226:14018 UDP
Jul 22 09:07:02 66.129.222.70:2721 -> U.Net.162.226:42329 UDP
Jul 22 09:07:03 66.129.222.70:20218 -> U.Net.162.226:50509 UDP
Jul 22 09:07:03 66.129.222.70:2124 -> U.Net.162.226:53270 UDP
Jul 22 09:07:03 66.129.222.70:47404 -> U.Net.162.226:13578 UDP
Jul 22 09:07:04 66.129.222.70:52171 -> U.Net.162.226:34291 UDP
Jul 22 09:07:04 66.129.222.70:321 -> U.Net.162.226:11922 UDP
Jul 22 09:07:04 66.129.222.70:525 -> U.Net.162.226:10120 UDP
Jul 22 09:07:04 66.129.222.70:51284 -> U.Net.162.226:61797 UDP
Jul 22 09:07:08 66.129.222.70:1922 -> U.Net.162.226:1967 UDP
Jul 22 09:07:08 66.129.222.70:0 -> U.Net.162.226:0 UDP
Jul 22 09:07:05 66.129.222.70:40209 -> U.Net.162.226:51761 UDP
Jul 22 09:07:06 66.129.222.70:59421 -> U.Net.162.226:33006 UDP
Jul 22 09:07:07 66.129.222.70:50486 -> U.Net.162.226:19133 UDP
Jul 22 09:07:08 66.129.222.70:21895 -> U.Net.162.226:15161 UDP
Jul 22 09:07:08 66.129.222.70:58040 -> U.Net.162.226:3173 UDP
Jul 22 09:07:08 66.129.222.70:2813 -> U.Net.162.226:3125 UDP
Jul 22 09:07:08 66.129.222.70:4455 -> U.Net.162.226:4552 UDP
Jul 22 09:07:09 66.129.222.70:4455 -> U.Net.162.226:4552 UDP
Jul 22 09:07:12 66.129.222.70:0 -> U.Net.162.226:0 UDP
Jul 22 09:07:09 66.129.222.70:65365 -> U.Net.162.226:25731 UDP
Jul 22 09:07:12 66.129.222.70:58040 -> U.Net.162.226:3173 UDP
Jul 22 09:07:09 66.129.222.70:23286 -> U.Net.162.226:46245 UDP
Jul 22 09:07:12 66.129.222.70:1922 -> U.Net.162.226:1967 UDP
Jul 22 09:07:10 66.129.222.70:1144 -> U.Net.162.226:8801 UDP
Jul 22 09:07:11 66.129.222.70:27820 -> U.Net.162.226:43798 UDP
Jul 22 09:07:11 66.129.222.70:13778 -> U.Net.162.226:25397 UDP
Jul 22 09:07:12 66.129.222.70:32863 -> U.Net.162.226:32863 UDP
Jul 22 09:07:12 66.129.222.70:36661 -> U.Net.162.226:23043 UDP
Jul 22 09:07:12 66.129.222.70:19272 -> U.Net.162.226:9644 UDP
Jul 22 09:07:16 66.129.222.70:0 -> U.Net.162.226:0 UDP
Jul 22 09:07:28 66.129.222.70:0 -> U.Net.162.226:0 UDP
```

```
Jul 22 09:07:28 66.129.222.70:58040 -> U.Net.162.226:3173 UDP
Jul 22 09:07:25 66.129.222.70:24504 -> U.Net.162.226:61366 UDP
Jul 22 09:07:26 66.129.222.70:44666 -> U.Net.162.226:22364 UDP
Jul 22 09:07:26 66.129.222.70:28099 -> U.Net.162.226:17125 UDP
Jul 22 09:07:27 66.129.222.70:4455 -> U.Net.162.226:4552 UDP
Jul 22 09:07:27 66.129.222.70:22665 -> U.Net.162.226:54617 UDP
Jul 22 09:07:28 66.129.222.70:7107 -> U.Net.162.226:23845 UDP
Jul 22 09:07:28 66.129.222.70:27383 -> U.Net.162.226:24869 UDP
Jul 22 09:07:28 66.129.222.70:14172 -> U.Net.162.226:32059 UDP
Jul 22 09:07:29 66.129.222.70:58040 -> U.Net.162.226:3173 UDP
Jul 22 09:07:29 66.129.222.70:0 -> U.Net.162.226:0 UDP
Jul 22 09:07:29 66.129.222.70:4048 -> U.Net.162.226:59285 UDP
```

Only scanning this box with a current virus scanner will determine for sure if this is benign traffic or not.
http://www.symantec.com/avcenter/warn/backorifice.html details specifics of the behavior of the Back Orifice Trojan. It essentially allows complete anonymous control of a Windows box by a master box that communicates with it through TCP port 31337. Such a host could greatly endanger the security of the University network for it is inside the protected boundary and could launch further attacks that were not detectable by our Network based IDS. What makes this even more troubling is that Back Orifice does not have a readily explainable method for infection. It is not like a worm. Therefore if a box is infected with it, it could indicate some very disturbing possibilities such as deliberate infection by a trusted insider. Any box infected should be removed from the network, have its hard drive scrubbed and the operating system reloaded from trusted media. There are just to many things that an attacker can do to the machine to allow it to function after such an infection.

Summary

Our analysts have taken the liberty of providing a detailed summarization and analysis of all the alerts detected by the University's IDS.
An explanation of each of the alert logs follow:

Alert Type	Number of Occurrences	Why We Care	What to DO (defensive actions)
1. Nimda Alert	655,358	"Among the numerous things done by the Nimda worm are, in no particular order, the addition of JavaScript code to the web pages served by infected servers to automatically cause your visitors to download the worm to their computers as an attachment to Microsoft Outlook; opening your	We need to analyze this traffic more closely. Systems that are infected need to be identified and cleaned. Hosts that appear to be infected are U.Net.157.246, U.Net.117.27,

		system to outside access; modifies the boot sequence of your computer to include the Nimda worm; adds a guest account, with administrative rights, on your system; and so on." (http://www.thesitewizard.com/news/Nimdaworm.shtml)	U.Net.157.241, and U.Net.157.247. Current virus software needs to be used to isolate and contain this worm.
2. Spp_portscan	645,647	Not a very informative alert. Better to correlate with scan logs. Does give some indication of scans directed at our hosts.	We can ignore this for now, but look at again in context of scan logs and other detected exploits
3. UDP src and dst outside net	121,480	Jason Lam states in his 14 Oct 2001 study quite correctly that this could indicate crafted packets being formed by a compromised host in our network. He also concedes it could be due to a misconfigured Snort rule. It is likely due to traffic being directed to our private networks and to our multicast servers that have not been included in snort's configuration file as internal to our network.	Many of the alerts are triggered by private addresses that can't be routed outside the University communicating with other University hosts or trying to communicate with General Electric??? (UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137). Rest of the traffic is directed to a multicast address 229.55.150.208, which is serving the university some type of video content. It is probably safe to ignore these alerts for now.
4. Watchlist 000222	64,703	This China-based network has been known to launch attacks against other networks according to {www.dshield.org} this is likely why it is on our Watchlist. `07/22-03:27:22.687604 [**]`	Appears to be legitimate web browsing traffic between Chinese clients and our web server. This is a legitimate Computer

		`Watchlist 000222 NET-NCFC [**] 159.226.39.135:63320 -> U.Net.99.174:80`	Science dept web server according to our previous lists of known network services. On examining the web server content several of the participants identified on the page are Chinese perhaps explaining the communication with China.
5. Possible My server traffic	28283	Many analysts have seen this traffic. Identifies it as a Myserver demon that is a Ddos agent that could be residing on our network. It binds to port 55850 and listens for a command to start launching a distributed denial service attack against other hosts. It trojanizes the ls and ps Unix services, so it is hard to spot just by looking at the processes running. http://archives.neohapsis.com/archives/incidents/2000-10/0136.html	Port 55850 resides on another network communicating with us on a gnutella file sharing port. Probably Can dismiss all this. **U.Net.140.9 is scanning methodically from port 55850** so this host needs attention. K. Haugsness in his GCIA practical saw this host being continuously tracrt'd back in Feb 2002.
6. Spp_Http ISS Unicode /CGI null byte attack detected. (Grouped together since they are generated by the same snort preprocessor function and are a result of the same vulnerability)	14205	The fear is that people are exploiting buffer overflows in web servers and hiding them from firewalls and IDS's through the use of Unicode characters. { http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise68 and Angela D. Orebaugh March 11, 2002 Much of this traffic could be just part of a normal web session. It has been my personal experience that this alert can generate a large number of false positives. This makes it very hard to distinguish between normal and hostile web sessions. However, the host below has a record of attacking hosts	This causes multiple false positive usually because it is returned when browsing Chinese or SSL encrypted sites. http://archives.neohapsis.com/archives/snort/2001-08/0075.html This could be the case for several of the alerts in our study (This university has a fair number of Chinese students). Still will want to examine the hosts identified as destinations by these

		with Dshield.Org so we should be suspicious.
		
		`07/22-03:27:22.804200 [**] spp_http_decode: IIS Unicode attack detected [**] 130.60.242.52:2005 -> U.Net.100.158:80`
		
		`07/22-03:27:23.061436 [**] spp_http_decode: IIS Unicode attack detected [**] 130.60.242.52:2013 -> U.Net.100.158:80`
		
		`07/22-03:27:23.298994 [**] IDS452/web-iis_http-iis-unicode-binary [**] 130.60.242.52:2022 -> U.Net.100.158:80`
		
		`07/22-03:27:23.298994 [**] spp_http_decode: IIS Unicode attack detected [**] 130.60.242.52:2022 -> U.Net.100.158:80`
			alerts and ensure that they are not vulnerable to these exploits. A thorough Nessus or similar scan should be conducted on our hosts to ensure that they are not vulnerable. Examine U.Net.100.158 to determine if it's running a vulnerable version of IIS. The relevant patch has been available from Microsoft since August 2000. If it is vulnerable there is a good chance that a back door has been installed. This probably means a reliable server can only be obtained by deleting the operating system and reloading from trusted media.

7. External access of TFTP server	12541	Outside hosts are accessing files on University host without having to provide any authentication. This allows attackers to place unwanted files on your machine for distribution to others. The TFTP service also has known holes that can be exploited to gain root access.	Add hosts U.net.109.105, U.net.111.231, U.Net.111.230, and U.net.111.219 to list of known University TFTP servers. They are all communicating with internal private network. For example: U.Net.111.219:69 -> 192.168.0.216:8018 Disregard this traffic.
8. External RPC calls	8638	External hosts may be probing for RPC vulnerabilities or exploiting an RPC vulnerability. Several RPC services are vulnerable to buffer overflow attacks which could yield to elevated access for an attacker on these boxes.	RPC scans from several hosts looking for RPC port mapper. No way from logs to determine what information was gathered. It appears

			very organized as if scripted. Examine further. Scan all internal hosts to see if service is running and disable the service or protect from outside access by means of access filtering on the RPC ports you are hosting.
9. Possible Red Worm - Adore	1501	Attacks known vulnerabilities in Linux hosts. Infection can result in a backdoor installed on port 65535 which is why this alert triggered.	Check U.Net.150.240, U.Net.150.120, and U.Net.153.45. All have suspicious traffic going to port 65535 over several days from the same subnets 63.250.219.x and 63.250.205 .X. See detailed discussion above for specific defensive actions.

10. IRC evil XDCC	1162	We are running multiple sessions of an IRC protocol (shocker…) This is used for file sharing. People could be downloading copyrighted material, viruses, or other unwanted material. One of our top suspects 216.110.36.14 is exchanging quite a lot of packets using this service with a previously identified potential Subseven victim on our network. This is worrisome. See below.
		
		`07/24-19:42:00.065739 [**] IRC evil - running XDCC [**] U.Net.178.199:1955 -> 216.110.36.14:6667`
		`07/24-20:06:04.995416 [**] IRC evil - running XDCC [**] U.Net.178.199:1955 -> 216.110.36.14:6667`
		`07/24-20:21:32.980972 [**] IRC evil - running XDCC [**] U.Net.178.199:1955 ->`
			We've seen this in previous analysts reports. We have several users going to XDCC servers to exchange files. Given the open nature of a University we probably don't care if students are exchanging files. We need to examine the behavior of this service interacting with U.Net.178.199. It does not seem benign. We recommend blocking outbound access to the port 6667 at the boundary destined for host 216.110.36.14.

		216.110.36.14:6667 07/24-20:32:08.959465 [**] IRC evil - running XDCC [**] U.Net.178.199:1955 -> 216.110.36.14:6667 07/24-20:55:01.072873 [**] IRC evil - running XDCC [**] U.Net.178.199:1955 -> 216.110.36.14:6667	
11. Possible Trojan server activity	1152	External hosts looking for or controlling university compromised boxes. Subseven Trojans allow an attacker to completely control a host with the Trojan running. This could create an in road to our network that will allow the enemy to attack other hosts on our network from inside our protected boundary.	Very organized scanning of University assets for Subseven Trojan (port 27374). Appears that one of our boxes responded and may now be under the control of external user. Need to follow advice in the more detailed description above.
12. Connect to 515 from outside	1151	515 is the LPD service, which could be vulnerable to exploit. Cert .org lists several LPD vulnerabilities at http://www.cert.org/advisories/CA-2001-30.html. Several of these vulnerabilities allow an intruder to use a buffer overflow to execute arbitrary commands on the system with super user privileges.	Looks like most traffic is due to a scan for this port by 64.30.217.125 and 24.123.46.10 More worrisome is this exchange of packets. 66.1.1.121:4297 -> U.Net.137.36:515 66.1.1.121:4297 -> U.Net.137.36:515 66.1.1.121:4375 -> U.Net.137.36:515 66.1.1.121:4375 -> U.Net.137.36:515 66.1.1.121:4375 -> U.Net.137.36:515 Check to see if U.Net.137.36 needs to be accessible to the Internet as a printer. If it does ensure it is hardened. If not block the 515 service at the border router. Block

			the two scanning addresses from accessing University's network. They are up to no good.
13. Queso Scans	920	Scanning for exploitable services on University network. It sends non-standard packets in an attempt to map the OS of the host{www.whitehats.com} and Jason Lamb, 14 Oct 2001 This would give potential attackers a great advantage when selecting what attacks to launch against our hosts.	Looks like legitimate traffic to a previously identified SMTP server. May be alerting due to ECN bits that are set in the packets. May want to investigate further.
14. Incomplete packets/ Null scan	792/396	Hostile host could be injecting bogus packets into the network. Similar packets causing both alerts. Both Null scans and incomplete packets can be used by an adversary to scan for available hosts/services. A Null scan uses packets with no TCP flags set; an incomplete packet could be any type of malformed IP packet. Both packets could and should generate a reset packet from the receiving host that can tell	All traffic occurs during a very brief period of time relative to the duration of the logs. It appears to be something misconfigured on the network. Still, check that hosts that receive this traffic are not sending strange traffic of their own. This is the only way to

		a hacker that a host is listening on a particular port.	guarantee this is not some unknown illicit communication. Of course a stateful firewall could screen out such traffic.
15. IIS_http_Unicode binary	759	Similar in scope to other Unicode attacks mentioned previously. It uses Microsoft IIS vulnerability to elevate access and run arbitrary code. IIS 4 & 5 with unicode support can be vulnerable to the encodings of traditional characters such as "/" in its unicode representation. Usage of this may allow a remote attacker to execute arbitrary commands on the server.	Most captures appear to be legitimate web traffic. However, we do have somebody who is scanning us for vulnerabilities. 213.93.159.116:4239 -> U.Net.87.209:80 213.93.159.116:4254 -> U.Net.87.215:80 213.93.159.116:4374 -> U.Net.88.49:80 213.93.159.116:3351 -> U.Net.89.136:80 213.93.159.116:3890 -> U.Net.90.157:80 213.93.159.116:4375 -> U.Net.91.148:80 213.93.159.116:4381 -> U.Net.91.154:80 213.93.159.116:4673 -> U.Net.92.3:80 Look out for this guy in the future. Maybe place a filtering rule on the University gateway router for this IP address.

16. Sun RPC high port access	669	Scanning for ghost portmapper 32771. This is bad because unlike port 111 and 135, which people know to block to prevent external RPC services, not everyone knows about this "ghost" RPC-mapper for SUN. The attacker is querying this service to find out what other RPC services are running. Armed with this knowledge he can employ one of several buffer overflow attacks against the RPC services to try to elevate his privileges on them. One such attack is the RPC.cmsd attack that attempts a buffer overflow on the calendar service. http://www.iss.net/security_center/advice/Intrusions/2001717/default.htm	There are both scans and accesses to the port mapper from the outside. Further the hosts scanning and attaching to this portmapper are using ports like 53 and 80 and so are probably not legitimate users. Need to investigate the hosts U.Net.99.179,U.Net.154.27, and U.Net.137.36 to see what information they are providing our adversary. We can accomplish this by scanning these hosts with Nessus or the like. We need to block this port at our boundary device in the future
17. Scan for proxies	525	People are searching our network for proxy servers to make themselves anonymous in their web browsing.	This is more of an annoyance. Still a scan of the university network should be done to ensure that such services are not provided to the outside. This is an instance where the integrity of the university could be damaged if its assets are unknowingly used as a jumping off point for illicit behavior.
18. Exploit X86	294	Possibility of an exploit attempt where setgid (0) or NOOPs are sent in order to gain access to a receiving host. It is common in several possible exploits. Affects Linux hosts that are running services that handle plain text	False positives due to transfer of binary data likely. Also a lot of attempts made by 64.30.217.125 to attack RPC ports: EXPLOIT x86 NOPS

		ASCII. Usually results in elevation of privileges on the exploited box. http://www.shmoo.com/mail/ids/jun00/msg00035.shtml and http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids284&view=event	[**] 64.30.217.125:977 -> U.Net.163.131:33001 looks like it may have succeeded. Examine this host and block 64.30.217.125 from accessing your network again. An IDS scanning services like FTP, DNS, etc for this signature followed prompt investigation may be the only way you can be truly sure this is not exploiting you. Applying current patches for these services will also mitigate the change of compromise.
19. SNMP public access	178	SNMP access from outside the network could indicate illicit control or monitoring of network assets or hosts. There is also the famous ASN.1 exploits that could deny service to assets running SNMP services by sending it malformed ASN.1 formatted messages. This is detailed in the at http://www.vnunet.com/News/1129277	Much of the traffic is coming from a different branch of the University of Maryland. The rest is coming from Vanderbilt University. This appears to be legitimate SNMP traffic, but ensure these hosts need to access these SNMP assets. It would be best to block all external SNMP access at the firewall or gateway router.
20. Beetle.ucs	101	CD-R sharing utility (see Edward Peck, August 4, 2001) Users are taking information off the Internet or their network and making CDs.	In a secure environment such easy information sharing as this might be troubling. In this environment it is hard to imagine this traffic being any more dangerous than every

			thing else that is going on.
21. Tiny Fragments	91	Crafted packets? The concern with almost any packet that is abnormal is that they might be used to map hosts on our network through RST responses or they might be some sort of illicit communication. Truly effective illicit communication will appear to be like normal traffic, so this is probably not too likely a possibility.	Appears to be due to misconfiguration. All traffic occurs between two hosts within a 20-minute timeframe.
22. SMTP relay denied	44	External hosts are attempting to use our resources to forward their email. This would waste our resources and allow illicit users to disguise the true source of emails. These emails could be traced back to us and damage our reputation.	Does not appear that external hosts are succeeding. Monitor the situation. Run a scan from outside the firewall to determine if these services are available on our network. Use the tool http://online.securityfocus.com/tools/2557 to accomplish this. Disable this relay service.
23. NMAP TCP ring	44	Somebody is scanning our network for open services with the NMAP scanning tool. Since it is coming from the outside our network this may mean somebody is searching for vulnerable ports to exploit on our system. They are sending Ack packets most likely to solicit a RST response.	Monitor this user…may want to block this IP address at the boundary router
24. Statdx	31	Very specific RPC attack –only works on Linux hosts. Issues a buffer overflow by inserting a bunch of ASCII 90's in the TCP packets. The attacker hopes to gain root privileges when the RPC.statd service processes the packet and faults. http://www.whitehats.com/info/IDS442 and CVE-2000-666	Scanning to see if this exploit works on hosts. Seems to know which ports and which hosts to target. This should be investigated further. Block this IP address at the gateway router. Scan your internal hosts to ensure that they are not running

			the RPC.statd service or if you are patch it.
25. NetMetro Alert	9	Could be a host with a Trojan program residing on it. Works on Windows boxes. Allows complete remote control. http://www.digitaltrust.it/arachnids/IDS79/event.html	Looks to be a result of a normal client traffic using ephemeral port 5031. There is nothing unusual in the packet contents to indicate otherwise.
26. DOS FTP globbing	4	An exploit that takes advantage of an FTP buffer overflow. In past analyses the amount of these packets was relatively high. (see T. Chapman, October 2001) Apparently certain Unix operating systems like BSD and Linux can be vulnerable because they have a faultily coded glob() function that is called to deal with shorthand notations for files. This can yield elevated privileges on the exploited box. http://www.cert.org/advisories/CA-2001-07.html	Appears to be a likely false positive due to the limited nature of the session. This server is an identified FTP server on the network. Ensure that it is not vulnerable to this exploit by scanning with Nessus or related tool.
27. Possible WinVNC service running	4	A potentially vulnerable service that allows control over a remote windows box. The authentication for this service is pretty weak so it is troubling if it is exposed to the Internet.{http://www.uk.research.att.com/vnc/winvnc.html}	Ensure that this service is disabled or authorized and adequately protected from unauthorized access through the use of an appropriate ACL on the router.
28. MyParty	4	Could be infected with this virus. This virus is spread by email but has the tendency to drop a back door on an infected box that will then contact a host probably somewhere in Russia. The back door allows remote control of the box. This has been seen on Windows servers and clients. http://www.f-secure.com/v-descs/myparty.shtml	Appears to be the result of using an ephemeral port that matches virus signature. Still a scan for this service on the network with NMAP might be prudent. Running an up to date virus scanner on all hosts should also

			mitigate possibility of this infection.
29. Back Orifice	2	Could have a Trojan residing on a host. Back Orifice allows complete control of a Windows machine by the attacker through the use of the Back Orifice server port. It appears that we are being sent messages by some process on this port. This is very worrisome.	Appears to be a real infection. Immediately block offending ip address at border router - take box off the network and analyze. If we are running Back Orifice we should delete the operating system and reload from trusted media. Back Orifice [**] 66.129.222.70:39849 -> U.Net.162.226:31337 07/22-09:07:24.199539 [**] Back Orifice [**] 66.129.222.70:39849 -> U.Net.162.226:31337 Symantec lists eradication details at http://www.symantec.com/avcenter/warn/backorifice.html

Scans

Results from the scan logs can shed light on reconnaissance attempts being made on the University network or to confirm virus activity. In affect they can be used as an early warning system to determine where the next threats will come from. As in the case of the University alert logs, a large number of scans alerts can be explained by permissible behavior that has exceeded some predefined traffic threshold in the University IDS. A brief synopsis of the top scans detected by the University IDS follows:

Scan Type	Number of Occurrences	What is it	What to do (defensive actions)
Medal of	2,348,218	Traffic destined for ports 12203 and 12300. This traffic	If this traffic uses up too much bandwidth,

Honor		is the result of a computer multiplayer game. I have it on good authority it is “awesome.” Apparently it is very popular.	block 12203 both incoming and outgoing at the border router. Otherwise assign more homework. These students have too much free time.
Kazaa	1,463,923	Traffic destined for port 1214. Used by Kazaa app to share files on the net.	See action above
Nimda	933,321	4 University hosts are scanning the Internet for further Nimda victims. This matches well with the Alert file, which identified these four hosts as infected. A sample of one such scan follows: Jul 22 03:08:51 U.Net.157.246:2087 -> 193.25.151.183:80 SYN ******S* Jul 22 03:08:51 U.Net.157.246:2092 -> 193.25.151.188:80 SYN ******S* Jul 22 03:08:51 U.Net.157.246:2093 -> 193.25.151.189:80 SYN ******S* Jul 22 03:08:51 U.Net.157.246:2094 -> 193.25.151.190:80 SYN ******S* Jul 22 03:08:51 U.Net.157.246:2095 -> 193.25.151.191:80 SYN ******S* Jul 22 03:08:51 U.Net.157.246:2096 -> 193.25.151.192:80 SYN ******S* Jul 22 03:08:51 U.Net.157.246:2097 -> 193.25.151.193:80 SYN ******S* Jul 22 03:18:25 U.Net.157.246:4864 -> 193.25.177.219:80 SYN ******S* Jul 22 03:18:25 U.Net.157.246:4865 -> 193.25.177.220:80 SYN ******S* Jul 22 03:18:25 U.Net.157.246:4866 -> 193.25.177.221:80 SYN ******S* Jul 22 03:18:25 U.Net.157.246:4867 -> 193.25.177.222:80 SYN ******S*	Isolate the machines U.Net.157.246, U.Net.117.27, U.Net.157.241, and U.Net.157.247 and rid them of their infection.
SMTP Traffic	66,730	Involves host U.Net.6.40, which has already been identified as a known University email server by K.	Traffic appears to be normal. No action required

		Haugsness December 2, 2001.	
AFS	33,849	A file-sharing program that has been noted in several analysts' studies. According to Phrack it is quite popular among universities. {www.phrack.com}	This is permissible traffic. No defensive action needs to be taken.
RPC scans	15,199	Hackers looking for port 111 to run RPC exploits	This is likely an early warning that there are RPC attacks the hackers want to try on University assets. Block port 111 at the border router from outside access.

Top Talkers

It can be informative to look at the top scan generators by host address. This often provides additional insights into the analyses that we have already conducted on the scan alert files.

Address	# of alerts generated	Possible reason	What to do (defensive action)
U.Net.70.207	1,376,275	Medal of Honor traffic	Draft student into Army
U.Net.157.247	503,454	Nimda scanning the world	Eradicate Nimda infection
U.Net.157.241	262,816	Nimda scanning the world	Eradicate Nimda infection
U.Net.157.246	167,008	Nimda scanning the world	Eradicate Nimda infection
U.Net.6.40	66,730	University SMTP server exchanging information	Normal traffic
U.Net.70.133	27,338	University AFS server	Normal traffic
U.Net.86.19	11,103	University SMTP server exchanging information	Normal traffic
208.186.13.245	7,234	Scanning for available RPC portmapper port	Potentially hostile-block this ip address from accessing network
140.179.152.245	6511	External AFS server University communicates with	Normal traffic
209.45.97.133	5082	Scanning Univ. network for open web servers	May want to block this IP at the very least monitor this IP.

Suspects list

There are a number of external hosts that have attempted to send packets repeatedly to university assets that appear to be hostile or potentially hostile. We have listed the hosts that we feel are most suspicious/ominous complete with the reason for our suspicions and additional identification information (compliments of {http://www.dsield.org/). We recommend blocking these ip addresses at the border router until such activity subsides.

<table>
<tr><th>Host/ Reason for suspicion</th><th>More info</th></tr>
<tr><td>61.172.255.20

RPC scans and attempts at RPC Statdx attack</td><td>inetnum: 61.172.244.0 - 61.172.255.255
netname: SHTELE-XINCHAN-IDC
descr: Shanghai Information Industrial Co.
country: CN
admin-c: ZY108-AP
tech-c: JZ5-AP
mnt-by: MAINT-CHINANET-SH
mnt-lower: MAINT-CN-SHTELE-XINCHAN
changed: ip-admin@mail.online.sh.cn 20020619
Source: APNIC</td></tr>
<tr><td>211.23.189.197

RPC scans and attempts at RPC Statdx attack</td><td>HostName:211-23-189-197.HINET-IP.hinet.net
DShield Profile:<table><tr><td>Country:</td><td>TW</td></tr><tr><td>Contact E-mail:</td><td>Network-center_AT_hinet.net (bounced)</td></tr><tr><td>Total Records against IP:</td><td>738</td></tr><tr><td>Number of targets:</td><td>426</td></tr><tr><td>Date Range:</td><td>2002-07-21 to 2002-07-22</td></tr></table>Ports Attacked (up to 10):

Fightback: sent to network-center@hinet.net on 2002-05-25 18:20:35
no reply received

Source: Dshield.org</td></tr>
</table>

216.110.36.14

Scanning for Subseven, may own a university box. (See alert section)
Print server scanning, RPC scanning, attempting to run exploits on print servers and RPC services.

DShield Profile:

Country:	US
Contact E-mail:	domain@FIBR.NET
Total Records against IP:	54
Number of targets:	4
Date Range:	2002-07-25 to 2002-07-25

Ports Attacked (up to 10):

Whois:
Fibrcom (NETBLK-FIBRNET5)
70 NE Loop 310, Suite 900
San Antonio, TX., 78216
US

Netname: FIBRNET5
Netblock: 216.110.0.0 - 216.110.95.255
Maintainer: FIBR

Source: Dshield.org

64.30.217.125

Scanning RPC, print services and Attacking printer and RPC services

EXPLOIT x86 NOOP [**]
64.30.217.125:4634 ->
U.Net.136.3:515
07/22-12:22:06.995845 [**]
External RPC call [**]
64.30.217.125:992 ->
U.Net.136.3:111

EXPLOIT x86 NOPS [**]
64.30.217.125:977 ->
U.Net.163.131:33001

HostName:ont-cvx1-125.linkline.com
DShield Profile:

Country:	US
Contact E-mail:	mbenz@linkline.com
Total Records against IP:	1
Number of targets:	1
Date Range:	2002-07-22 to 2002-07-22

Source: Dshield.org

<table>
<tr>
<td>208.186.13.245

Largest RPC scan of our network</td>
<td>IP Address: 208.186.13.245
Hostname: dillweed.cache.net
DShield Profile:
<table>
<tr><td>Country:</td><td>US</td></tr>
<tr><td>Contact E-mail:</td><td>abuse@ELI.NET</td></tr>
<tr><td>Total Records against IP:</td><td>290</td></tr>
<tr><td>Number of targets:</td><td>231</td></tr>
<tr><td>Date Range:</td><td>2002-07-31 to 2002-07-31</td></tr>
</table>
Source: Dshield.org</td>
</tr>
<tr>
<td>66.129.222.70

Sending multiple UDP packets to U.Net.162.226 with strange ports after contacting via tcp on Back Orifice port.</td>
<td>HostName: atlas.newdig.com
DShield Profile:
<table>
<tr><td>Country:</td><td>US</td></tr>
<tr><td>Contact E-mail:</td><td>rommel@viclink.com</td></tr>
<tr><td>Total Records against IP:</td><td>16</td></tr>
<tr><td>Number of targets:</td><td>13</td></tr>
<tr><td>Date Range:</td><td>2002-08-20 to 2002-08-20</td></tr>
</table>
Whois:
Valley Internet Company (NETBLK-VICLINK)
PO Box 1286
McMinnville, OR 97128
US
Netname: VICLINK
Netblock: 66.129.192.0 - 66.129.223.255
Maintainer: VIC

Source: Dshield.org</td>
</tr>
</table>

63.250.219.X

May be attaching to back door; definitely scanning

HostName:UNKNOWN-63-250-219-.yahoo.com
DShield Profile:

Country:	US
Contact E-mail:	netops@broadcast.com
Total Records against IP:	82
Number of targets:	31
Date Range:	2002-08-26 to 2002-08-26

Ports Attacked (up to 10):

Port	Attacks
11793	1
14900	10

Source: Dshield.org

63.250.205.X

May be attaching to back door, definitely scanning. See graph detailing the various UDP connections that are being made. From the graph it can be seen that multiple hosts from this subnet are attempting to send UDP packets to several university hosts spanning the gamut of the University subnet range.

HostName: wmcontent05.bcst.yahoo.com

Country:	US
Contact E-mail:	netops@broadcast.com
Total Records against IP:	108
Number of targets:	58
Date Range:	2002-08-25 to 2002-08-25

Ports Attacked (up to 10):

Port	Attacks
1755	1

Source: Dshield.org

203.37.255.97

Seems to know about ghostmapper ports

University host is having a bizarre conversation using UDP. Notice the difference in times, yet our source port stays the same, meaning it is not ephemeral. We are communicating with the DNS service on a box that previously scanned us. This doesn't look normal at all.

Jul 27 01:05:42 U.Net.137.7:1121 -> 203.37.255.97:53 UDP
Jul 27 07:34:46 U.Net.137.7:1121 -> 203.37.255.97:53 UDP
Jul 27 07:52:12 U.Net.137.7:1121 -> 203.37.255.97:53 UDP
Jul 27 07:52:16 U.Net.137.7:1121 -> 203.37.255.97:53 UDP
Jul 27 07:52:17 U.Net.137.7:1121 -> 203.37.255.97:53 UDP

HostName: ns.apnic.net
DShield Profile:

Country:	AU
Contact E-mail:	abuse_AT_telstra.net. (bounced)
Total Records against IP:	6676
Number of targets:	136
Date Range:	2002-08-26 to 2002-08-26

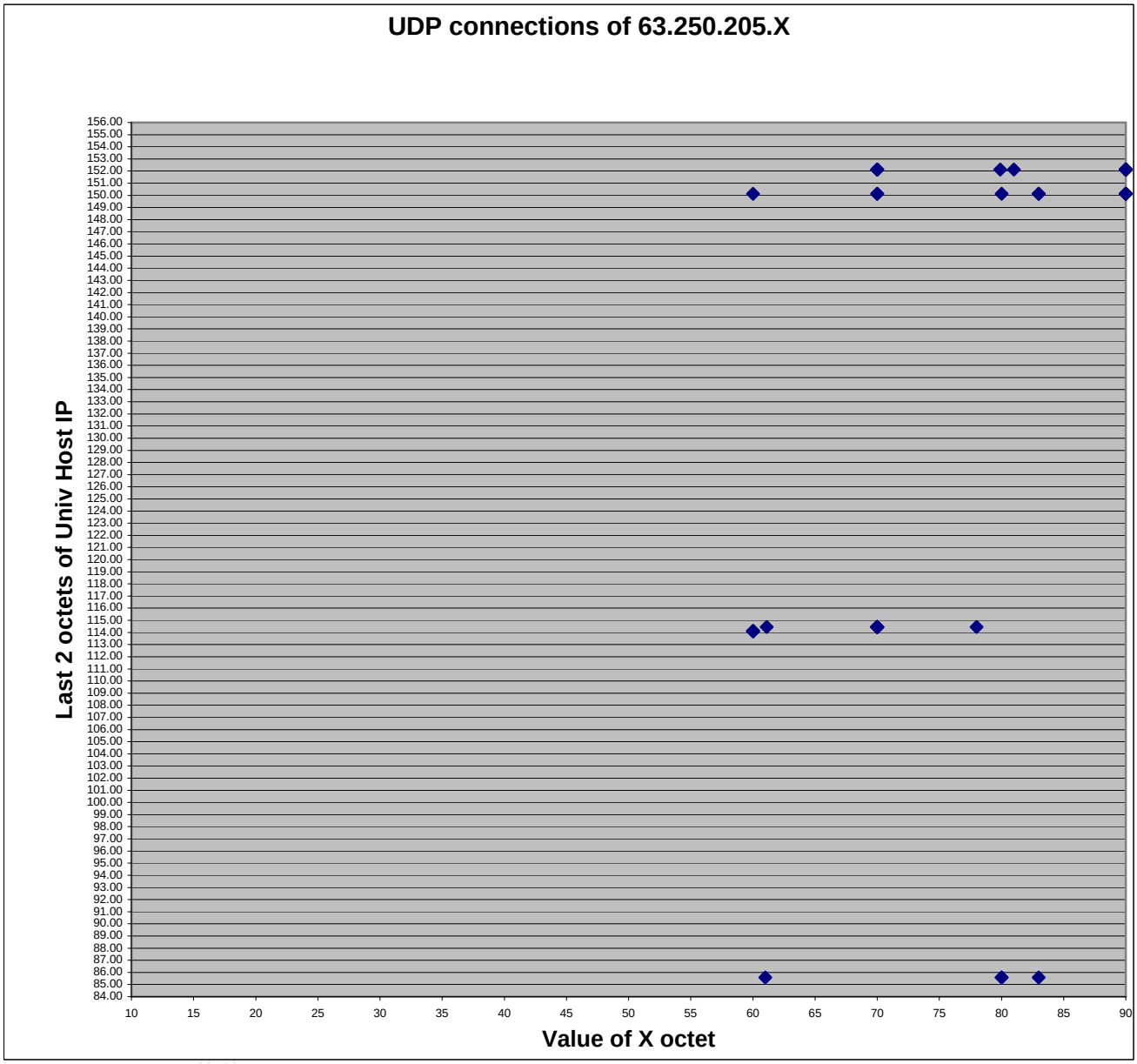
Ports Attacked (up to 10):

Port	Attacks
53	3

Source: Dshield.org

Graphical representation of a Scan

This graph provides a more discernible way of assessing the nature of distributed scanning activity on the University network. This graph was created from a thorough analysis of the Snort scan alerts. This particular graph shows where the intersection of scanning initiators and scanning victims occur. In log format the true nature of such a scan is often lost. Each of these points was recorded as separate UDP scans in the Snort logs. Looking at this graph we can see that these scans are likely related and are likely directed by the same entity. They all come from the same subnet after all and do not appear to overlap. This graph shows 3 definite concentrations of scanning activity. This gives an indication of the coordination and pattern that the attacking process is undertaking in scanning our network. Such a pattern may be a good way to fingerprint specific scanning tools in the future.



Graph detailing UDP packets sent from 63.250.205.X
Network to hosts on our network

Description of the Analysis Process

Between the Scan logs and the Alert logs there were nearly 8 million records to be considered. In both the case of the alert and the scan files we began our

analysis by concatenating all 5 alert files into a single composite file. We then used the command

sort –T D:\ alertall |uniq >newfilename

to reduce redundant entries in each of the files. This only reduced the size of these files nominally. The composite files were still far too sizeable to deal with in any effective manner.

We decided to break the problem down into more manageable subcomponents through the use of consideration and elimination. Simply put, we separated out a small subset of the composite file, stored and catalogued it separately, studied it carefully, and then eliminated its members from the composite file and as a result, further consideration.

After looking at the alert and scan files, we noticed that each was comprised of only a relatively small number of unique alerts compared to the number of overall entries. Through the use of carefully constructed grep statements, we were able (with a great deal of practice) to break up the alert and scan files into component files that contained only one type of alert. These files were also quite sizeable but they allowed us to concentrate on the alert at hand and to focus on detecting trends and anomalies among similar alerts. This was quite time consuming, but it was the most obvious way that we could truly get an in depth understanding of the traffic that caused these alerts over a five day period. After each file was looked at to consider time relationships, the time stamps were removed from each of the entries in the alert and scan files. Each of these files were then imported into tables inside Excel and or Word which tended to ensure that the formatting of each entry was much more standardized than it had been in the original alert files. With time stamps removed and formatting enforced, the UNIX uniq command worked much better at reducing duplicate alerts. In this way each of the component files were significantly paired down into files that could then be analyzed very closely for aberrant ports, addresses and the like. This was mainly accomplished by doing ad-hoc queries using grep, awk, sort, etc.

Once suspicious pattern were discovered using this method, it was then possible to search in the larger composite files for the times when incidents occurred, and the number of times incidents occurred.

The oos composite file really did not provide more information into the nature of network intrusions than was gained through the thorough analysis of the alert and scan files. Using similar techniques to those performed above, we discovered that the vast majority of the traffic captured in these files was due to the ECN bits being set. Apparently the Snort rule still regarded these reserve bits being set as an aberration. There were 22 packets that had combinations of TCP flags that were illegal. This could be due to an attempt at scanning using

these malformed packets. Given the amount of scanning performed on the university network over the five days, it is not a surprising find

Epilogue

There is little doubt that there exists substantial threat to the availability, integrity and confidentiality of university network assets. Even in the open environment that is embraced and advocated among universities, such a situation as detailed in these logs is a grave cause for concern. The defensive recommendations specified in this study should be considered a minimum set of guidelines to protect the resources that make the university mission possible. To provide further protections to these network assets it is apparent that an information security policy needs to be either created, revised, or enforced. Incoming and outgoing traffic should be screened by a firewall that is consistent with the policies set forth in the information security policy. Vulnerability scanning should be conducted periodically and problems quickly addressed. Virus scanners should be run on all University assets and the definitions constantly updated. The University should adopt a computer inventory system that not only tracks the serial numbers of assets it owns, but versions of operating systems these assets run and the services these assets provide to the outside world. Services not permitted by University policy should be blocked from leaving the boundary of the University network. These simple suggestions will go along way to mitigate many of the problems that have been witnessed in these logs. The University must begin taking action soon. Otherwise, this University will risk the very assets it so clearly needs to perform its mission.

References

Works cited have been acknowledged throughout this text where appropriate. A brief summary follows.

Haugsness, K., GIAC Practical, December 2, 2001.

Lam, Jason, GIAC Practical, Oct 14, 2001.

Northcutt, Stephen. Novak, Judy, “Network Intrusion Detection-An Analyst’s Handbook”. New Riders, Indianapolis, In 2001.

Orebaugh, D. Angela, GIAC Practical, March 11, 2002.

Peck, Edward, GIAC Practical, August 4, 2001.

So, Hee, GIAC Practical, February 16, 2002.

Stevens, W. Richard. “TCP/IP Illustrated, Volume 1”. Addison Wesley Longman, Reading 1994.

The various web sites:

Dartmouth University. September 2002. http://www.ists.dartmouth.edu/

Dshield.org. September 2002. http://www.dsield.org/

Digitaltrust.com.September 2002. http://www.digitaltrust.it/

F-secure.September 2002. http://www.f-secure.com/

http://www.hackfix.org/subseven/

Incidents.org. September 2002. http://www.incidents.org/

ISS.com: September 2002. http://www.iss.net/security_center/

The Open Web Application Security Project. September 2002. http://www.owasp.org/

Phrack.com. September 2002. {http://www.phrack.com}/

Samspade.org. September 2002. http://www.samspade.org/

Simovits Consulting. September 2002. http://www.simovits.com/

TheSiteWizard.com. September 2002. http://www.thesitewizard.com/

Symantec. September 2002. http://www.symantec.com/

United Kingdom Research Division of AT&T. September 2002. {http://www.uk.research.att.com}

Vnunet.con. September 2002. http://www.vnunet.com/News

Whitehats.com. September 2002. http://www.whitehats.com/

END of Assignment 3

Last Updated: September 23rd, 2017

Upcoming Training

SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS London February 2018	London, United Kingdom	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	Tysons, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced