



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Network Monitoring and Threat Detection In-Depth (Security 503)"  
at <http://www.giac.org/registration/gcia>

# **GIAC Certified Intrusion Analyst (GCIA) Practical Assignment Version 3.3**

© SANS Institute 2004, Author retains full rights.

**Mohammed Haron**  
**Submitted April 24<sup>th</sup>, 2003**

## Table of Content:

Topic	Page Number
Assignment 1: Describe the State of Intrusion Detection	<b>2</b>
Summary	4
Introduction	4
Intrusion Detection Systems	4
The Need for Intrusion Analysts	5
Evolution to Intrusion Prevention System	7
Intrusion Prevention System Products	8
Best Practices of IPS Deployment	10
IDS in the Future of IPS World	11
Conclusion	12
References	12
Assignment 2: Detect 1: MISC Source Port 20 to <1024"	<b>16</b>
Source of Trace	16
Detect Was Generated By	16
Probability The Source Address Was Spoofed	17
Description of Attack	18
Attack Mechanism	18
Correlation	19
Evidence of Active Targeting	19
Severity	20
Defensive Recommendation	21
Multiple Choice Test Question	21
References	21
Assignment 2: Detect 2: WEB-MISC cisco /%% DOS attempt	<b>23</b>
Source of Trace	24
Detect Was Generated By	24
Probability The Source Address Was Spoofed	25
Description of Attack	25
Attack Mechanism	25
Correlation	26
Evidence of Active Targeting	26
Severity	26
Defensive Recommendation	27
Multiple Choice Test Question	27
References	27
Assignment 2: Detect 3: MISC Tiny Fragment	<b>29</b>
Source of Trace	30
Detect Was Generated By	30
Probability The Source Address Was Spoofed	31
Description of Attack	32
Attack Mechanism	33
Correlation	34
Evidence of Active Targeting	34
Severity	35
Defensive Recommendation	35
Multiple Choice Test Question	35
References	36
Assignment 3: Analyze	<b>37</b>
Overview of analysis	37
List of files analyzed	38
A List of detects	38

Description of Alerts events	39
Alerts Top Ten Analysis	56
Scans Analysis	57
OOS Analysis	60
Defensive Recommendation	61
A Description of analysis process	62
References	63

© SANS Institute 2004, Author retains full rights.

## **Assignment 1: Describe the State of Intrusion Detection**

### **With Intrusion Prevention, Is There Still A Need For Intrusion Detection?**

#### **Summary**

In this assignment, I will discuss why there is still a need for Intrusion Detection technology to exist and complement the latest Intrusion Prevention technology, which evolved from Intrusion Detection technology and firewall technology. First, I will briefly explain what is Intrusion Detection technology and functions of Intrusion Analysts in IDS world. Then, I will discuss briefly about Intrusion Prevention technology and best practices in “perfect world” for IPS to function. Finally, I will discuss main differences between Intrusion Prevention and Intrusion Detection technologies, and the future of IDS.

#### **Introduction**

Intrusion Prevention is the new buzzword in the Information Security field. It is a new technology that combined the best features of both intrusion-detection and firewall. Intrusion detection is capable of detecting attack in progress and alerting analyst to take action while firewall block attacks from passing through according to access control policies enforced. Static policies enforced on firewalls and alerting analyst on attack detected by intrusion detection system, are no longer ample in handling the magnitude of new automated attacks generated by worms, virus and other malicious tools. Experiences with Code Red worm, Nimda worm and Slammer worm really showed the how fast these attacks were spreading and damages were done in a very short period of time. Slammer worm for example, only requires two infected hosts to bring down a whole subnet in Local Area Network.

In intrusion prevention system, the emphasis is automation in blocking attack detected by the intrusion detection component. John Pescatore, a research director for Internet security at Gartner stated, “If the intrusion-detection function or the antiviral-detection function says this is an attack, and the network is vulnerable to that attack, the firewall blocks it or shunts the packet off to some safe destination, thwarting the attack”.<sup>[23]</sup>

#### **Intrusion Detection System**

Intrusion Detection Systems (IDS) is decision support system for security analysts. It is not about preventing or blocking incoming attacks, but it is about identifying source of the attack, assess the damage caused by the attack, prevent future break-ins and to prevent attack from spreading. IDS looks at the patterns of the traffic going through your networks, examine each packet and try to make intelligent decisions regarding their findings and then alert security analyst for action to be taken.<sup>[34]</sup>

Basically, IDS can be categorized to three different type; Network Intrusion Detection System (NIDS), Host Intrusion Detection System (HIDS) and Hybrid.

1. Network Intrusion Detection System (NIDS)

NIDS is basically deals with information passing on the wire in a network. It intercepts packets traveling along various communication mediums and protocols, such as TCP/IP, and analyzed in a number of ways depending on the capabilities of the IDS. Signature based NID will simply compare the packet to a signature database of known attack and vulnerabilities, while protocol anomaly look for anomalous behavior deviating from the RFCs and normal known traffics.

Even though NIDS were initially incapable to operate in switched networks, encrypted networks and a very high speed network (Gbits), new NIDS products and solutions have resolved these limitations. NIDS can be deployed in switched environment by using TAPS, hubs or spanning port, while encrypted network can still be monitored by NIDS at packet header level which is not encrypted in SSL or VPN connection. However, a product called BIGIP provides integrated SSL encryption and decryption on real-time that decrypted traffics can then be monitored by NIDS.<sup>[55]</sup> Many new NIDS are also capable of handling Gbits network such as Network Sensor NS3000, a product from Sourcefire.<sup>[56]</sup>

2. Host Intrusion Detection System (HIDS)

HIDS runs on host, and basically designed to monitor, detect and respond to user and attack. Some robust HIDS also include audit policy management and centralization, supply data forensics, statistical analysis and evidentiary support, and in certain instances provide some measure of access control. Once classic example of HIDS product is Tripwire.

3. Hybrid Intrusion Detection.

Hybrid intrusion detection on the other hand is basically combination of best features of both network and host-based intrusion detection devices with addition of centralize management. Hybrid solutions provide the logical complement to NID and HID which is a central intrusion detection management. Because of this, hybrid is the best solution for enterprise level deployment of IDS.<sup>[17]</sup>

While IDS is designed only to detect attacks, IDS products such as Snort and RealSecure also equipped with simple prevention component called active-response capabilities by sending TCP Resets packet to stop the attack. Even though this is not really efficient, but this type of reactive defense feature that started the evolution of Intrusion Prevention technology.

### **The Need for Intrusion Analysts**

Intrusion Detection deployment does not stop an attack nor will react to validate alerts on events detected and takes action to stop it. Instead, the critical part of analyzing, validating alerts and takes action to modify or inform people responsible to modify

firewall policies to block attack, can only be done by Intrusion Analysts. It will not work by having IDS deployed and being left alone to do its work. Having a highly skilled Intrusion Analysts is critical in greatly increase the chances of catching malicious activities, virus, worm and other exploits, thus protecting your environment.

Joel Snyder summarized some important points to keep in mind. First, an IDS is only as good as its configuration. IDS need to know everything about your network before something is a false positive or it is a real attack. Information such as port numbers used by HTTP server, need to be configured in IDS so that it will look at the right place. For example, if web server is running port 80 and port 8008, IDS will only recognize port 80 to be valid HTTP traffic, and detected port 8008 traffic as an attack. Thorough audit in of your network should be done for Intrusion Analyst to reconfigure IDS.

Secondly, IDSes are dumb. You have to tell them everything or you will be saturated with false positive alerts. Even if you tell them everything, you will still find IDSes are always one step or two behind the latest attack. IDS products currently available in market do not use artificial intelligence or neural networks, but they look for patterns that match known problems which is signature based IDS. Even for protocol anomaly based IDS, they are matching network traffic with RFCs and normal traffic pattern, thus still producing false positives even though not as many as signature based IDS. As result, highly skilled intrusion analysts are the best asset to fine-tune the IDSes, updating signatures and reducing false positives while at the same time balancing the need of keeping valid attacks and suspicious traffics.

Third, you need to know a lot of details. Each IDS product operates differently, depending if the product is doing stateful matching, context matching, protocol anomaly or pattern searching. All of these have to be considered, in addition to different level of detail perform by IDS products. All of these are based on TCP/IP protocols, thus Intrusion Analysts are expected to know the ins and outs of TCP/IP.

Fourth, you need be prepared to spend a lot of time and money. Freeware IDS does not mean it ends there after the IDS have been installed and configured. However IDS requires time and money to be administered and managed on daily basis. Intrusion Analysts have to continuously analyze alerts produced, take action for valid attacks and reducing false positives.

Fifth is marketing features versus reality of these IDS products. Be sure to evaluate the risks and rewards of newer features that look useful at first, such as active attack evasion. Some of these features seem less than perfect when examined closely by Intrusion Analysts.<sup>[16]</sup>

**FIGURE 5: Security Certification Bonuses**

SECURITY CERTIFICATIONS	MEDIAN % of Base Salary Q1 '02	ANNUAL % Growth Q1 '01-Q1 '02
GIAC Certified Intrusion Analyst (GCIA)	12%	50%
GIAC Certified Incident Handler (GCIH)	10%	N/A
GIAC Certified Firewall Analyst (GCFW)	9%	29%
Certified Information Systems Security Professional (CISSP)	9%	29%
GIAC Systems and Network Auditor (GSNA)	8%	N/A
Certified Information Systems Auditor (CISA)	8%	-20%
GIAC Certified Unix Security Administrator (GCUX)	8%	33%
GIAC Certified Windows Security Administrator (GCWN)	8%	60%
Certified Network Security Professional (CNSP)	7%	17%
GIAC Security Essentials Certification (GSEC)	6%	N/A

Source: Foote Partners LLC

Figure 1: <sup>[10]</sup>

Interestingly enough, David Foote from Infosecuritymag.com in his survey found out that Intrusion Detection field is one of the most demanded skills in Information Security. The Figure 1 above showed that GIAC Certified Intrusion Analyst is the second highest growth of 50 percent in demand between year of 2001 and 2002. <sup>[10]</sup>

### Evolution to Intrusion Prevention System

Interesting new technology has emerged in the information security world called Intrusion Prevention System (IPS). It is not a product, but instead it is a technology, which supposedly combined the best of firewall and Intrusion Detection world. <sup>[1]</sup> While firewall protection is more static according to fixed policies, whether at Link layer or up to Application layer for proxy firewall, firewall lacks of intelligence to detect new attacks and dynamically block the attack. From simple reactive-response capabilities in Intrusion Detection that simply send TCP Reset to attacker to stop UDP flood attack, Intrusion Prevention has evolved to more intelligent and dynamic in detecting and stopping attacks.

The first generation of IPS was not that smart that it would block an attack essentially by adding a firewall rule, blocking all traffics from a hostile IP address. That's fine, until false positives started causing more legitimate traffics being blocked and creating denial of service to itself.

The second-generation of IPS however, operates in a more elegant fashion in which the offending attack is dropped, but any other connection, even from the same host, are



allowed. While this reduces chances of creating its own Denial of Service, false positives are still big issues. <sup>[24]</sup> So, even in IPS space, intrusion analysts are still needed to remove false positives and to verify traffics being blocked are real attacks.

## Intrusion Prevention System Products

Similar to IDS products, IPS products can also divided to two categories, Host Intrusion Prevention Systems (HIPS) and Network Intrusion Prevention Systems (NIPS).

HIPS – One such product is Entercept Intrusion Prevention System ([www.entercept.com](http://www.entercept.com)) that has three separate products for specialized protection that is Entercept Standard Edition, Entercept Web Server Edition and Entercept Database Edition.

Entercept IPS proactively protects servers and applications from attacks that can not be block using firewall such as buffer overflows and worms. Entercept can protect servers from both known and unknown malicious attacks. The method used by Entercept to protect host is by evaluating requests to the operating system before they are processed. Combination of both behavioral rules and signatures are used to detect and prevent both known and unknown attacks.

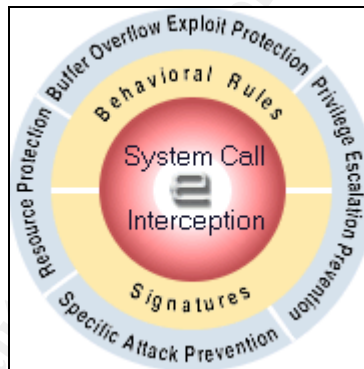


Figure 1 <sup>[60]</sup>

Two main component of Entercept is Entercept Agent, installed on each server and Entercept Console for management, review and reporting. Entercept Agents intercept system calls to the operating systems and if calls determined to be malicious in behavior, will then get blocked. Among other things, Entercept determines the process making the call, the user making the call, the resource being accessed by the call, and the user permissions related to the call. This information is then matched against appropriate behavioral rules and signatures. Calls that attempt malicious behavior or match any specific rules are then blocked. These preventive activities are logged to the Entercept Console for review and reporting.

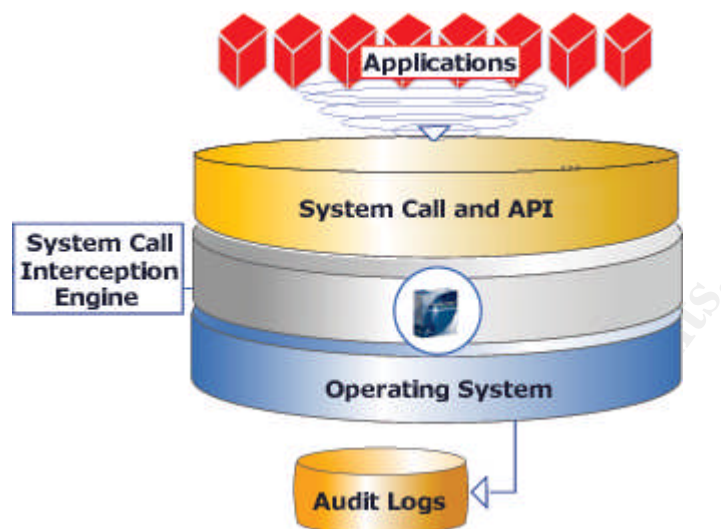


Figure 2: Entercept resides on the server, protecting the operating system and applications <sup>[61]</sup>

NIPS – Commercial products available are like IntruShield 4000 and IntruShield 2600 from Intruvert ([www.intruvert.com](http://www.intruvert.com)) and Attack Mitigator IPS from Toplayer ([www.toplayer.com](http://www.toplayer.com)); while freeware under OpenSource available are Hogwash (<http://hogwash.sourceforge.net/>) and Snort-Inline (<http://www.snort.org/dl/contrib/patches/inline/>) even though Snort-inline is build more towards honeypot concept.

Intruvert for example, has two products available, IntruShield 4000 for enterprise networks and IntruShield 2600 for mid-to-large networks. According to Intruvert, these two sensors deliver real-time network intrusion detection and prevention solution with features listed below:

- Intrusion Intelligence™: Unprecedented capabilities provide detailed, accurate and reliable information related to intrusion identification, relevancy, direction, impact and analysis.
- Virtual IDS: Powerful capability to enforce multiple, highly granular, custom intrusion policies within a single sensor.
- Comprehensive Intrusion Detection: Intelligent detection of known, first-strike, and DoS attacks using a combination of signature, anomaly, and DoS detection techniques.
- Flexible Deployment: Unprecedented flexibility of IDS deployment—including in-line, full-duplex tap, and SPAN modes—to suit any network security architecture.
- Real-time Intrusion Prevention: Proactive capability to stop in-progress attacks coupled with a rich set of automated and user-initiated alerting and response actions.
- Multi-gigabit Performance: Powered by purpose-built hardware that is capable of delivering multi-gigabit performance.

- Automated Real-time Threat Updates: Innovative, automated process delivers real-time, enterprise-wide signature updates without requiring sensor reboots, and provides protection against newly discovered attacks while eliminating manual updates and sensor downtime.
  - Interoperability: Works with leading firewalls, enterprise management applications and Security Information Management (SIM) applications to offer reduced total cost of ownership.<sup>[59]</sup>

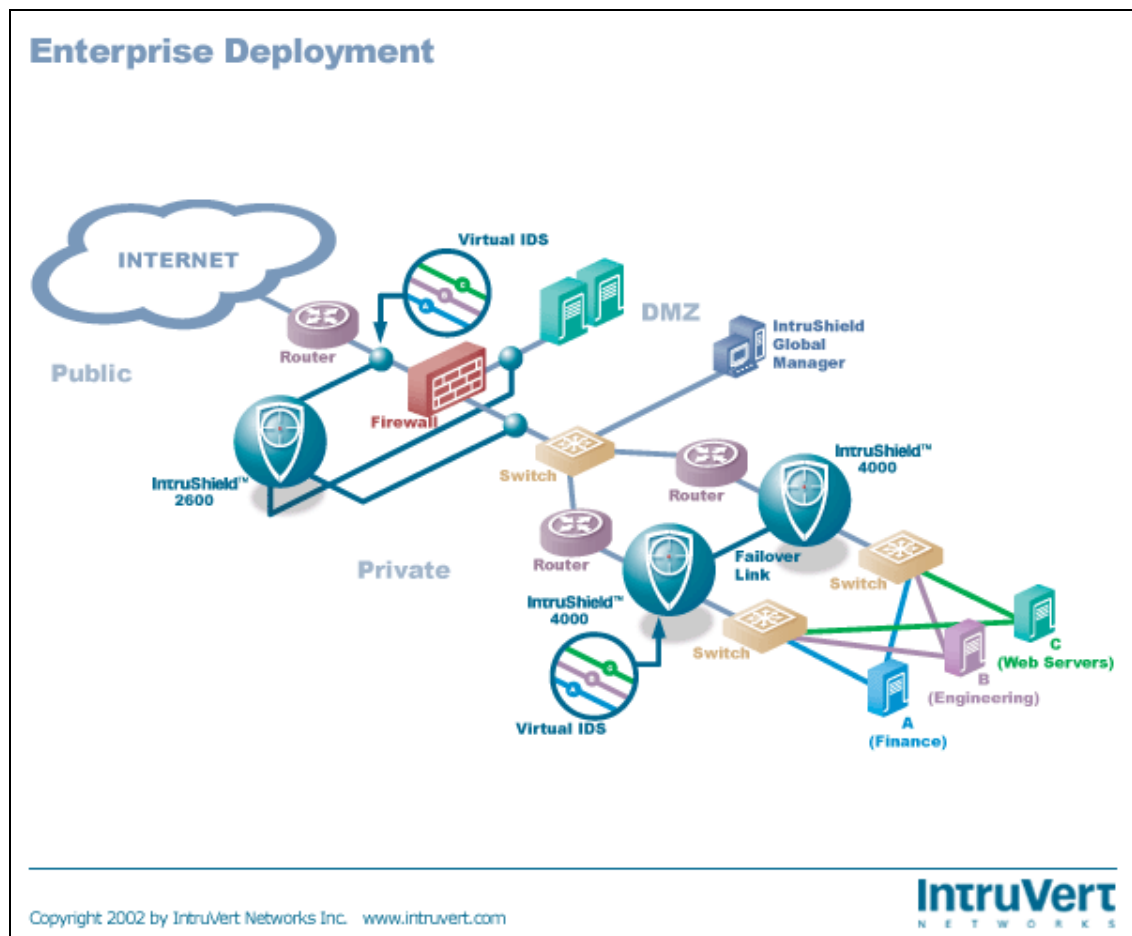


Figure3: Intruvert Enterprise Deployment<sup>[58]</sup>

### Best Practices of IPS Deployment

Any organization that intends to protect itself by using Intrusion Prevention technology, should take a number of factors that address its security requirements into consideration. Best practice should include:

1. **Host-based protection.** Securing at network level is a major challenge. The best place to enforce security is at the desktops and servers, where actual work is done and the potential for damage is greatest.
2. **Real-time prevention decisions.** This is to ensure the highest level of security and minimize the ability to bypass security policy on host. Effective IPS strategy in preventing violation in real time is a major challenge.

3. **Defense in depth – protection from attack from various places.** Attacks usually has multiple phases such as exploiting network and application-level weaknesses, replicating and distributing themselves, and making unauthorized changes. Intrusion Prevention must protect systems from all phases. This is another major challenge for IPS.
4. **Real-time correlation at the agent and enterprise level.** Taken from IDS technology, correlation is vital for effective IPS technology and provides a level of accuracy on prevention decisions that is not covered with signature-based approaches. Even in the IDS worlds, correlation is a challenging task in reducing false positives.
5. **Behavioral approach.** For Intrusion prevention to be proactive, it must enforce appropriate system and application behavior. Dependency on protocol anomaly and signatures is inadequate.
6. **Flexibility to meet unique corporate needs.** Intrusion Prevention solution must be flexible to the unique needs of every corporation in how it configures and manages its systems and applications.
7. **Ease of deployment.** Deployment should minimize personnel overhead in agent deployments, with out-of-the-box functionalities, and allow for new and custom policies to be rolled out as needed.
8. **Centralized event management.** All events generated by agents must roll up into a centralized repository from which alerts and reports may be generated. This is a must for corporate level deployment.
9. **Platform coverage, with support from desktops and servers.** Solution must include coverage for the key operating systems that corporation wishes to protect.
10. **Administration.** Policy management can be accomplished from central point and can be automatically distributed to agents on a configurable interval. Policies must also be exportable for replication and archive purposes. <sup>[25]</sup>

Above best practices can be used as guidelines in choosing Intrusion Prevention products, however, spending the time in testing the product themselves is certainly the best way in establishing a truly deep understanding of the product. <sup>[44]</sup>

### IDS in Future of IPS World

Intrusion Prevention is certainly not Intrusion Detection due IDS's biggest asset that is the Intrusion Analyst who can make sound analysis and judgment on attacks detected. Marty Roesch laid out two scenarios that illustrate why Intrusion Prevention is unlikely to replace IDS:

1. IPS devices only guard the peering points (at best) of network. IN the case of an attack between hosts on the same broadcast network (inside the peering point) you have absolutely no coverage from the IPS. In that case you'll need to have an IDS to tell you what's going on. For example, someone in engineering decides to give him self a raise by hacking into the accounting department and making it so, your IPS has no visibility into this traffic so it's quite worthless. You IDS can see this traffic, however, and collect the relevant information for detection/enforcement of policy and evidence for law enforcement.

2. No IPS is going to be perfect, so attacks are going to slip through them. It can be attacks that they don't know about (new buffer overflows, etc) or even traffic that's legitimate but hostile in your environment, like non-anonymous logins to your anonymous FTP server. If an attack gets by an IDS, how will you know? You better have pretty good IDS to tell you, that's how. <sup>[42]</sup>

One of IPS's biggest problem is that it is lacking the artificial intelligent (AI) or "logic" to determine false positives. <sup>[45]</sup> Toby Kohlenberg brought up similar question that is, "how have you validated that you have a false-positive rate that approaches zero and how would I tune the box to ensure it will never cut off legitimate traffic?" <sup>[49]</sup> This problem can become a great risk of creating its own Denial of Service by actively blocking legitimate traffic in the network.

## Conclusion

The evolution of Intrusion Prevention technology certainly adds another layer of security for better protecting corporate network and hosts. For security professionals, Intrusion Prevention complements current security technology already in place such as firewalls and intrusion detection technology, rather than replaces them. In addition, Marty Roesch stated, "I believe IPS to be more of a threat to (or the future of) firewalls." "Intrusion detection devices have a VERY different role in the network security hierarchy, they provide "awareness" of that's happening on your network, verification of policy compliance and detection of potential threats and anomalies". <sup>[42]</sup>

## References:

1. Snyder, Joel PhD.; "IPS: A Technology, not a product"; Network World; 11/25/02  
<http://www.nwfusion.com/columnists/2002/1125snyder.html>
2. Lindstrom, Pete; "Guide to Intrusion Prevention"; Information Security; October 2002;  
<http://www.infosecurymag.com/2002/oct/sidebar.shtml>
3. Roundtable – "IDS in the trenches"; September 2002;  
<http://www.infosecurymag.com/2002/sep/roundtable.shtml>
4. Roundtable – "IDS at the crossroads"; June 2002;  
<http://www.infosecurymag.com/2002/jun/cover.shtml>
5. Mackey, Richard; "Security Architecture: Layered Insecurity"; June 2002;  
<http://www.infosecurymag.com/2002/jun/insecurity.shtml>
6. Briney, Andy; "What isn't Intrusion Prevention"; April 2002;  
<http://www.infosecurymag.com/2002/apr/note.shtml>
7. Briney, Andy; "Next steps for IDSes"; May 2002;  
<http://www.infosecurymag.com/2002/may/note.shtml>

8. Heiser, Jay; "Misguided Thinking: 5 misconceptions continue to hamper overall security"; November 2002;  
<http://www.infosecurymag.com/2002/nov/curmudgeon.shtml>
9. Bobbitt, Mike; "Inhospitable host: Attacker may try the door, but intrusion prevention tools won't let them in"; October 2002;  
<http://www.infosecurymag.com/2002/oct/cover.shtml>
10. Foote, David; "Security still pays"; August 2002;  
<http://www.infosecurymag.com/2002/aug/securitymarket.shtml>
11. Skoudis, Edward; "Cracker tools and techniques: Faster, Stealthier... More dangerous"; July 2002;  
<http://www.infosecurymag.com/2002/jul/faster.shtml>
12. Ferrell, Robert G.; "Security Freeware: Making sense of firewalls"; June 2002;  
<http://www.infosecurymag.com/2002/jun/makingsense.shtml>
13. Avolio, Fred; "Practical Firewalling"; June 2002;  
<http://www.infosecurymag.com/2002/jun/basics.shtml>
14. Harley, David; "Virus Trends: The future of malicious code"; May 2002;  
<http://www.infosecurymag.com/2002/may/maliciouscode.shtml>
15. Bobbitt, Mike; "Web Security: Bulletproof"; May 2002;  
<http://www.infosecurymag.com/2002/may/bulletproof.shtml>
16. Snyder, Joel; "Everything you need to know about IDSes"; Network World; 04/08/02;  
<http://www.nwfusion.com/columnists/2002/0408snyder.html>
17. Innella, Paul; "The Evolution of Intrusion Detection Systems"; Tetrad Digital Integrity, LLC; November 16 2001;  
<http://online.securityfocus.com/infocus/1514>
18. Newman, David; Snyder, Joel; Thayer, Rodney; "Crying Wolf: False alarm hide attacks"; Network World; 06/24/02;  
<http://www.nwfusion.com/techinsider/2002/0624security1.html>
19. IDS Glossary; Network World; 06/24/02;  
<http://www.nwfusion.com/techinsider/2002/0624security2.html>
20. Snyder, Joel; "Three tips for reducing false alarms"; Network World; 06/24/02  
<http://www.nwfusion.com/techinsider/2002/0624security3.html>
21. "Attack we performed and performance of IDS products tested"; Network World; 10/08/01;  
<http://www.nwfusion.com/reviews/2001/1008revside1.html>
22. Whitepaper; "Beyond IDS: Essentials of Network Intrusion Prevention"; Top Layer;  
[http://www.toplayer.com/bitpipe/IPS\\_Whitepaper\\_112602.pdf](http://www.toplayer.com/bitpipe/IPS_Whitepaper_112602.pdf)
23. Cummings, Joanne; "From Intrusion Detection to Intrusion Prevention"; Network World; 09/23/02;  
<http://www.nwfusion.com/buzz/2002/intruder.html>
24. DeShon, Marcus, PhD; "Intrusion Prevention versus Intrusion Detection";  
<http://www.secureworks.net/techResourceCenter/fullTechArticle.php?article=IpsVsIds>
25. "Technology Best Practices For Intrusion Prevention"; OKENA, Inc.; Jan 1 2002;  
<http://www.okena.com/pdf/IP%20Best%20Practices.pdf>

26. Whitepaper; "Intrusion Detection and Prevention: Protecting Your Network from Attacks Allowed By The Firewall"; OneSecure™ Inc.;  
<http://www.securitytechnet.com/resource/security/ids/idp-whitepaper.pdf>
27. Whitepaper; "Intrusion Detection and Prevention"; OneSecure™ Inc.; 2001;  
[http://www.securitytechnet.com/resource/security/ids/OneSecure\\_IDP\\_Datasheet.pdf](http://www.securitytechnet.com/resource/security/ids/OneSecure_IDP_Datasheet.pdf)
28. Hammond, David; "New Approach To Intrusion Detection: Intrusion Prevention"; November 2001;  
<http://www.scmagazine.com/scmagazine/sc-online/2001/article/049/article.html>
29. Hollander, Yona; "Intrusion Prevention: Why Simple Detection Doesn't Cut It Anymore"; August 2001;  
<http://www.scmagazine.com/scmagazine/sc-online/2001/article/030/article.html>
30. Dunigan, Tom; Hinkel, Greg; "Intrusion Detection and Intrusion Prevention on a Large Network. A Case Study"; Oak Ridge National Laboratory;  
[http://www.usenix.org/publications/library/proceedings/detection99/full\\_papers/dunigan/dunigan\\_html/index.html](http://www.usenix.org/publications/library/proceedings/detection99/full_papers/dunigan/dunigan_html/index.html)
31. Abene, Mark; Kovacich, Greal L.; Lutz, Steven; "Intrusion Detection Provides A Pound of Prevention"; Page 1-6  
<http://www.networkcomputing.com/815/815ws1.html>
32. Graphics; "Network Security Hot Spots";  
<http://img.cmpnet.com/nc/815/graphics/hotspots.pdf>
33. Wang, Feihi; "Vulnerability Analysis, Intrusion Prevention and Detection for Link State Routing Protocols"; 2000  
<http://www.cs.ucdavis.edu/~wu/publications/fwphd.pdf>
34. Taylor, Laura; "**Intrusion detection is not intrusion prevention**"; TechRepublic; 22 August 2002;  
<http://www.zdnet.com.au/itmanager/technology/story/0,2000029587,20267597,00.htm>
35. Conz, James; "The Next Security Software? Introduction of  
[http://www.giac.org/practical/James\\_Conz\\_GCIA.doc](http://www.giac.org/practical/James_Conz_GCIA.doc)
36. Ellis, Joe; "Intrusion Detection Systems: Component architecture"  
[http://www.giac.org/practical/Joe\\_Ellis\\_GCIA.doc](http://www.giac.org/practical/Joe_Ellis_GCIA.doc)
37. Schultise, Jeff;  
[http://www.giac.org/practical/jeff\\_schultise\\_GSEC.doc](http://www.giac.org/practical/jeff_schultise_GSEC.doc)
38. Rudzonis, Brian; "Intrusion Prevention: Does it measure up to the hype?"  
[http://www.giac.org/practical/Brian\\_Rudzonis\\_GSEC.doc](http://www.giac.org/practical/Brian_Rudzonis_GSEC.doc)
39. Terry, Patrick; "Intrusion Detecion is becoming re-created as Intrusion Prevention"  
[http://www.giac.org/practical/Patrick\\_Terry\\_GSEC.doc](http://www.giac.org/practical/Patrick_Terry_GSEC.doc)
40. History of Firewalls; Copyright 2001 Anti-Hack;  
<http://dmsweb.badm.sc.edu/mgsc890/firewalls/fire2.htm>
41. Allen, Julia; Christie, Alan; Fithen, William; McHugh, John; Pickel, Jed; Stoner, Ed; "State of the Practice of Intrusion Detection Technologies"; January 2000; Technical Report; Carnegie Mellon, Software Engineering Institute;  
<http://www.cert.org/archive/pdf/99tr028.pdf>



42. Roesch, Marty; “[Snort-users] New Trend: Intrusion Prevention”; Email [snort-users@list.sourceforge.net](mailto:snort-users@list.sourceforge.net); Friday, Dec 13 2002
43. Gonzalez, Alberto; “[Snort-users] New Trend: Intrusion Prevention”; Email [snort-users@lists.sourceforge.net](mailto:snort-users@lists.sourceforge.net); Friday, Dec 13, 2002
44. Ranum, Marcus J.; “OSEC [WAS: Re: Intrusion Prevention]”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Monday, Dec 30 2002
45. Lo, Roy; “Re: Intrusion Prevention”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Thursday, Oct 31 2002
46. Plato, Andrew; “Re: Intrusion Prevention Systems”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Wednesday, Oct 30 2002
47. Williams, Rick; “Re: Intrusion Prevention”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Wednesday, Dec 25 2002
48. Kohlenberg, Toby; “Re: IDS responses”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Monday, Nov 18 2002
49. Kohlenberg, Toby; “Re: Changes in IDS Companies”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Thursday, Act 31 2002
50. Plato, Andrew; “Intrusion Prevention System”; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Monday, Oct 28 2002
51. Andy; “H/N IPS –what is there?”; Taliskers Network Security Tools; Email [focus-ids@securityfocus.com](mailto:focus-ids@securityfocus.com); Wednesday, Dec 11 2002
52. Bakos, George; “FW: Missing admin sql password in Okena Stormwatch”; Email [intrusions@incidents.org](mailto:intrusions@incidents.org); Wednesday, Dec 18 2002
53. Ranum, Marcus J; Curtin, Matt; Internet Firewalls: Frequent Asked Questions; <http://www.interhack.net/pubs/fwfaq/#SECTION00031000000000000000>
54. Intrusion Detection FAQ; How do you implement IDS (network based) in a heavily switched environment; <http://www.sans.org/resources/idfaq/switched.php>
55. F5 Networks; BIGIP; [http://www.f5.com/solutions/applications/terminal\\_ab/terminal\\_server\\_ab.pdf](http://www.f5.com/solutions/applications/terminal_ab/terminal_server_ab.pdf)
56. Sourcefire; Product: Sourcefire Intrusion Management System; <http://www.sourcefire.com/products/products.htm>
57. Enterscept; Product Overview; <http://www.enterscept.com/products/>
58. Intruvert; Intrusion Prevention System; Enterprise Deployment; <http://www.intruvert.com/products/images/600-enterprise-deployment-with-vids.gif>
59. Intruvert; IntruShield Network IDS Sensors; <http://www.intruvert.com/products/sensors.htm>
60. Enterscept; Enterscept Standard Edition; <http://www.enterscept.com/products/enterscept/index.asp>
61. Enterscept; How Enterscept Works; <http://www.enterscept.com/products/enterscept/prodinfo/overview.asp>



## Assignment #2: Detect 1 : MISC Source Port 20 to <1024"

P	Signature	Classification	Type	Source	Destination	Sensor	Time Stamp »
2	MISC Source Port 20 to <1024	bad-unknown	TCP	216.189.121.3:20	32.245.146.122:80	Sensor01	1:38 PM - 10/16

12:38:00.796507 216.189.121.3.20 > 32.245.146.122.80: S 2489891814:2489891814(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 109, id 5840, bad cksum dcaf!)

### 1. Source of Trace

The source of this trace from the log provided by GIAC at <http://www.incidents.org/logs/Raw/2002.9.1>

### 2. Detect Was Generated By:

Initially, to ease my search for interesting detects, I used Demarc PureSecure [<http://www.demarc.com>] and I ran snort with the command below against the RAW files provided at <http://www.incidents.org/logs/Raw> to upload the alerts into MySQL database. I disabled preprocessor stream4 and preprocessor stream4\_reassemble and got better result than default configuration. Thanks to Daniel Wesemann who brought this up in [intrusions@incidents.org](mailto:intrusions@incidents.org) list.

```
snort -p -l C:\PureSecure\sensor\logs -r C:\PureSecure\sensor\logs_raw\2002.9.1 -c C:\PureSecure\sensor\conf\snort4.conf
```

I used all files listed below:

2002.9.1	2002.9.2	2002.9.3	2002.9.9	2002.9.10
2002.9.11	2002.9.12	2002.9.13	2002.9.14	2002.9.15
2002.9.16	2002.9.17	2002.9.18	2002.9.19	2002.9.20
2002.9.21	2002.9.22	2002.9.23	2002.9.24	2002.9.25
2002.9.26	2002.9.27	2002.9.28	2002.9.29	2002.9.30
2002.9.31	2002.10.1	2002.10.2	2002.10.3	2002.10.4
2002.10.5	2002.10.6	2002.10.7	2002.10.8	2002.10.9
2002.10.10	2002.10.11	2002.10.12	2002.10.13	2002.10.14
2002.10.15	2002.10.16	2002.10.17	2002.10.18	

In this particular detect, this event was detected from file 2002.9.16. Looking at the date on detect, however, showed discrepancy between the file name 2002.9.16 with time stamp of 2002-10-16 13:38:00.

Below is the detect with priority 2 shown on my Demarc PureSecure webpage:

Signature Information			
Signature	Sensor	Event ID	Time Stamp
MISC Source Port 20 to <1024 - Find in Rules	Sensor01 (1)	66424	2002-10-16 13:38:00
Classification Description	Priority	Classification	Time Since Event
Potentially Bad Traffic	2	bad-unknown	93 dy 7 hr 58 min 55 sec Ago

Basic Information							
Src IP	Src Host	Src Port	Src Service	Dst IP	Dst Host	Dst Port	Dst Service
216.189.121.3	-	20	ftp-data	32.245.146.122	-	80	http
Whois :: Trace :: Ping :: DNS				Whois :: Trace :: Ping :: DNS			

IP Information								
Ver	Hlen	TOS	Length	ID	Flags	Offset	Chksum	TTL
4	5	-	48	5840	-	-	56495	109

TCP Information								
Seq	Ack	Urp	Res	Win	Flags	Offset	Chksum	
2489891814	-	-	-	16384	S	7	55331	

Which is triggered by snort rule version 1.90 as below:

```
alert tcp $EXTERNAL_NET 20 -> $HOME_NET :1023 (msg:"MISC Source Port 20 to <1024";
flags:S; reference:arachnids,06; classtype:bad-unknown; sid:503; rev:2;)
```

Then, I used windump to find this particular detect as below:

```
C:\>windump -r 2002.9.16 -vv -X "src 216.189.121.3"
12:38:00.796507 216.189.121.3.20 > 32.245.146.122.80: S 2489891814:2489891814(0) win 16384
<mss 1460,nop,nop,sackOK> (DF) (ttl 109, id 5840, bad cksum dcdf!)
0x0000 4500 0030 16d0 4000 6d06 dcdf d8bd 7903 E..0..@.m.....y.
0x0010 20f5 927a 0014 0050 9468 bbe6 0000 0000 ...z...P.h.....
0x0020 7002 4000 d823 0000 0204 05b4 0101 0402 p.@..#.....
```

### 3. Probability the Source Address Was Spoofed

It is less likely that the source address was spoofed due to the fact that this TCP connection requires completed TCP three-way-handshake i.e. (SYN, SYN-ACK, ACK) for connection to be established. Even though only SYN packet was detected from 216.189.121.3 targeting 32.245.146.122, other SYN-ACK and ACK packets might have been passed undetected due to fact that snort rules are not configured to capture them. The attacker could also play man-in-the-middle to listen for responses of its spoofed source address, but this would be difficult to achieve. Therefore, very high probability that source address is not spoofed.

Bad packet checksum showed was probably resulted from sanitize work done by SANS on the raw files.

Question:

From: Smith, Donald [Donald.Smith@qwest.com]  
Sent: Wednesday, January 29, 2003 8:09 AM  
To: 'Mohammed Haron'; intrusions@incidents.org  
Subject: RE: LOGS: GIAC GCIA Version 3.3 Practical Detect (MHaron)

Can you be sure the source address ISNT spoofed if you see a three way

handshake?

> man-in-the-middle to listen for response to this attack.

Ok that would be one way to spoof and still get results.  
Any others you can think of?

Answer:

Donald is correct. Another possibility is that the three-way-handshake was completed, but “hidden” and not captured because the default snort rule is only capturing the initial packet with “S” flag set. This, in this case, source is not been spoofed.

#### 4. Description of Attack

This attack attempted to establish TCP session to port 80 on a host 32.245.146.122 using active ftp port 20. This could be a scan to see if web service is running on target or a beginning of an attack if target is known to be vulnerable. There is known vulnerability on Cisco 600 series routers running CBOS (Cisco Broadband Operating System) version 2.4.2ap and earlier that vulnerable to denial of service attack caused by a vulnerability in the web-based configuration utility. The web-based utility by default is bind to port 80, even if this service has been disabled. This vulnerability allows remote attacker to send multiple HTTP connection request that cause the router to stop responding or allow any traffic to pass.

#### 5. Attack Mechanism

This attack is targeted to host usually located behind firewall. The attack was attempting to establish active ftp session to port 80 for http service on the target host. In regular passive FTP session, client host will initiate a FTP connection using from ephemeral port (ports above 1024) to port 21 which is default ftp port on a ftp server. Then during this session, authentication occurs between the client and the server host. When the client request a data connection used for file transfer, the client will issue PORT command to the server host with parameters such as IP address and port number to connect to. The server host will then open a connection on port 20 (known as FTP Data Port) to client’s specified IP address and port information to send data to.

In this attack however, utilizing active ftp, the attacker is initiating a connection to target host. Then, attacker can listen for the FTP replies from the target host. The 3-digit numbers replies can give specific meaning such as shown below:

Reply Number	Meaning
125	Data connection already open; transfer starting
200	Command OK
425	Can’t open data connection

These information are valuable for attacker to plan the next attack.

## 6. Correlations

Scott Shinberg <sup>[4]</sup> discussed this active ftp attack in his GCIA practical paper that source port 20 was used to get through firewall. Since TCP connection for active ftp requires that TCP is initiated from the ftp server to the client for sending data, this attempt might pass the firewall. Regular ftp session of the ftp client to send ftp command to ftp server on port 21 in regular FTP session.

Phil Wood <sup>[9]</sup> also suggested that there are possibilities that the rule used in this case, generated a lot of false positive from operating systems with broken IP stack. However, if we would expect false positive from operating systems with broken IP stack, we would expect a lot of similar events to be generated.

A snortsnarf IDS log at <http://openbsd.agero.se/snort/sig/sig20.html> also shown attacks detected by same rule. These events utilizing same attack mechanism, but targeting port 25 for mail service. Only one alert generated for each attempt.

## 7. Evidence of Active Targeting.

It is pretty clear that this attack is targeted to 32.245.146.122 from source IP 216.189.121.3. To know more about the source host, I used a tool called Netcraft at <http://www.netcraft.com> to determine if source host is running any web service. The result is shown below:

” The site [216.189.121.3](http://216.189.121.3) is running **Microsoft-IIS/5.0 on Windows 2000**”

The fact that this source host is running a web service attempting to connect to target host on port 80 is not a good sign. Especially, source host is attempting to initiate TCP connection via active ftp is further proof that this is active targeting. It would be difficult to determine whether this attack was successful or not without additional alerts. This could also be a super slow scan to see a response from port 80 and if the IP is spoofed, man-in-the-middle could have played a role in catching the response.

Question:

From: [Bryce\\_Alexander@Vanguard.com](mailto:Bryce_Alexander@Vanguard.com)  
Sent: Wednesday, January 29, 2003 9:22 AM  
To: Mohammed Haron  
Cc: intrusions@incidents.org  
Subject: Re: LOGS: GIAC GCIA Version 3.3 Practical Detect (MHaron)

You mentioned that you did not see any additional traffic from the source IP and assumed that this indicated an unsuccessful attack. Keep in mind that the logs only contain information that matched some kind of

signature. Could the rest of the traffic simply not match any signature and therefore could have been present, but not logged?  
How many SYN frames do you usually see when the TCP connection is blocked by a firewall or filter (assume it is not crafted)?

Answer:

Bryce is correct. Due to raw files produced with "S" flag set, other packet related to this packet {SYN-ACK, ACK} is then "hidden" and not captured.

Question:

From: Robert Wagner [rwagner@eruces.com]  
Sent: Wednesday, January 29, 2003 5:57 AM  
To: 'Mohammed Haron'; intrusions@incidents.org  
Subject: RE: LOGS: GIAC GCIA Version 3.3 Practical Detect (MHaron)

What is the purpose of the attack? Criticality - Are there any known vulnerabilities with web servers answering requests coming from a privileged port? If not, then is this just a simple scan looking for web servers?

Answer:

This is most probably a scan. However, there is a known vulnerability as listed on CAN-2001-1065 on Cisco router. The web-based configuration utility in Cisco 600 series routers running CBOS 2.0.1 through 2.4.2ap bind itself to port 80 even when web-based configuration services are disabled, which could leave the router open to attack.

## 8. Severity

Severity is calculated using formula:

$$\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures})$$

With each element worth 1 to 5 points where 1 is the least and 5 is the highest.

**Criticality:** Since the attacker is targeting port 80 on a specific IP address, this is probably a web server that the attacker has obtained its IP address. So, I gave 3 points.

**Lethality:** This attack is used to bypass the firewall packet filter. Even though this might open to other attack, this particular attack by itself is not lethal. This could be a scan for future attack. So, I gave 2 points.

System Countermeasures: Since, there was no respond detected from the targeted IP (at least with the undetermined rules used to produce the raw file), I assumed the target IP is quite secure. So, I gave 2 points.

Network Countermeasures: Due to unknown network and the nature of this attack to bypass a firewall, I suspect that the network has some perimeter of defense in place. So, I gave 2 points.

Therefore:

Severity = (3+2)-(2+2) = 1

## 9. Defensive Recommendation

Stateful firewall should be installed that will be able to inspect the content of a packet, as additional defense on top on non-stateful firewall already in place. Routers should be scanned for vulnerabilities and if Cisco 600 series router is use, and running CBOS version 2.4.2ap or earlier, the software will need to be upgraded to version 2.4.2b or higher. Detail information is available at <http://www.cisco.com/warp/public/707/cisco-cbos-webserver-pub.shtml>.

## 10. Multiple Choice Test Question

Please refer to this Snort Rule below and answer the question:

```
misc.rules:alert tcp $EXTERNAL_NET 20 -> $HOME_NET :1023 (msg:"MISC  
Source Port 20 to <1024"; flags:S; reference:arachnids,06; classtype:bad-unknown;  
sid:503; rev:1;)
```

From the Snort rule above, which one below will not generate any event?

- A) 216.189.121.3.20 > 32.245.146.122.1023: **SAck** 2489891814:2489891814(0) win 1638
- B) 216.189.121.3.20 > 32.245.146.122.1023: **S** 2489891814:2489891814(0) win 1638
- C) 216.189.121.3.20 > 32.245.146.122.80: **Ack** 2489891814:2489891814(0) win 1638
- D) 216.189.121.3.20 > 32.245.146.122.80: **SF** 2489891814:2489891814(0) win 1638

Answer: C.

## References:

- 1) Whitehats.com; "IDS6 "SOURCEPORTTRAFFIC-20-TCP"  
[http://www.whitehats.com/cgi/arachNIDS/Show?\\_id=ids6&view=research](http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids6&view=research)

- 2) Maxwell, Mike; "Slow and steady ftp probes";  
<http://www.incidents.org/archives/intrusions/msg03533.html>
- 3) CAN-2001-1065;  
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-1065>
- 4) Shinberg, Scott; GCIA Certification - Practical Assignment;  
[http://www.giac.org/practical/Scott\\_Shinberg\\_GCIA.doc](http://www.giac.org/practical/Scott_Shinberg_GCIA.doc)
- 5) Lam, Jason; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Jason\\_Lam\\_GCIA.doc](http://www.giac.org/practical/Jason_Lam_GCIA.doc)
- 6) Cisco Security Advisory: CBOS Web-based Configuration Utility Vulnerability;  
<http://www.cisco.com/warp/public/707/cisco-cbos-webserver-pub.shtml>
- 7) ISS X-Force Database; cisco-cbos-web-config (7027);  
[http://www.iss.net/security\\_center/static/7027.php](http://www.iss.net/security_center/static/7027.php)
- 8) Netcraft; <http://www.netcraft.com>
- 9) Wood, Phil; Snort-Users mailinglist;  
<http://archives.neohapsis.com/archives/snort/2001-06/0444.html>
- 10) Snortsnarf signature page; MISC Source Port 20 to <1024;  
<http://openbsd.agero.se/snort/sig/sig20.html>
- 11) Stevens, Richard W; TCP/IP Illustrated Volume 1, The Protocols; Nov 2001

## 2.2 Assignment #2: Detect 2: WEB-MISC cisco /%% DOS attempt

08:16:49.626507 207.166.87.157.64785 > 66.54.32.235.80: P 3447537467:3447538164(697) ack 502420461 win 64860 (DF) (ttl 124, id 7952, bad cksum 9cec!)

4500 02e1 1f10 4000 7c06 9cec cfa6 579d E.....@.|.....W.  
4236 20eb fd11 0050 cd7d 3f3b 1df2 53ed B6.....P.|?;..S.  
5018 fd5c ef9b 0000 4745 5420 2f52 6561 P..|....GET./Rea  
6c4d 6564 6961 2f61 6473 2f63 6c69 636b lMedia/ads/click  
5f6c 782e 6367 692f 7777 772e 7573 6174 \_lx.cgi/www.usat  
6f64 6179 2e63 6f6d 2f73 706f 7274 732f oday.com/sports/  
6d69 6c6b 2f6c 6f61 642e 6874 6d2f 2525 milk/load.htm/%%  
5241 4e44 2525 2f53 7065 6369 616c 312f RAND%%/Special1/  
3230 3435 385f 4d69 6c6b 5f53 414d 4d59 20458\_Milk\_SAMMY  
5f32 3030 335f 3239 3835 2f63 6c65 6172 \_2003\_2985/clear  
2e67 6966 2f25 0025 2045 5225 2520 4854 .gif/%.%.ER%.HT  
5450 2f31 2e31 0d0a 4163 6365 7074 3a20 TP/1.1..Accept:.  
2a2f 2a0d 0a41 6363 6570 742d 4c61 6e67 \*/\*..Accept-Lang  
7561 6765 3a20 656e 2d75 730d 0a41 6363 uage:.en-us..Acc  
6570 742d 456e 636f 6469 6e67 3a20 677a ept-Encoding:.gz  
6970 2c20 6465 666c 6174 650d 0a55 7365 ip,.deflate..Use  
722d 4167 656e 743a 204d 6f7a 696c 6c61 r-Agent:.Mozilla  
2f34 2e30 2028 636f 6d70 6174 6962 6c65 /4.0.(compatible  
3b20 4d53 4945 2035 2e35 3b20 5769 6e64 ;.MSIE.5.5;.Wind  
6f77 7320 4e54 2035 2e30 290d 0a48 6f73 ows.NT.5.0)..Hos  
743a 2061 642e 7573 6174 6f64 6179 2e63 t:.ad.usatoday.c  
6f6d 0d0a 436f 6e6e 6563 7469 6f6e 3a20 om..Connection:.  
4b65 6570 2d41 6c69 7665 0d0a 436f 6f6b Keep-Alive..Cook  
6965 3a20 5553 4154 494e 464f 3d55 4944 ie:.USATINFO=UID  
2533 4461 6138 3133 3237 3833 6438 3837 %3Daa8132783d887  
3337 303b 2052 4d49 443d 6161 3831 3332 370;.RMID=aa8132  
3738 3364 3838 3863 6130 3b20 5449 443d 783d888ca0;.TID=  
3173 3335 3064 6130 7570 6d68 7175 3b20 1s350da0upmhqu;.  
4146 4649 4c49 4154 455f 434f 4445 3d75 AFFILIATE\_CODE=u  
7361 3b20 5645 5254 4943 414c 5f43 4f44 sa;.VERTICAL\_COD  
453d 6e61 7469 6f6e 616c 3b20 5549 443d E=national;.UID=  
6161 3831 3332 3738 3364 3838 3733 3730 aa8132783d887370  
3b20 5444 6174 613d 3b20 7631 7374 3d33 ;.TData=;.v1st=3  
4444 3237 3938 4130 4432 3039 3135 333b DD2798A0D209153;  
2042 726f 7773 6572 536e 6966 6665 723d .BrowserSniffer=  
6e61 7669 6761 746f 722e 7479 7065 2533 navigator.type%3  
4432 2533 4225 3041 6e61 7669 6761 746f D2%3B%0Anavigato  
722e 7665 7273 696f 6e25 3344 352e 3525 r.version%3D5.5%  
3342 2530 416e 6176 6967 6174 6f72 2e6f 3B%0Anavigator.o  
7325 3344 2532 3225 3230 5769 6e64 6f77 s%3D%22%20Window  
7325 3230 4e54 2532 3035 2e30 2532 3925 s%20NT%205.0%29%  
3232 2533 4225 3041 6e61 7669 6761 746f 22%3B%0Anavigato  
722e 6a73 5665 7273 696f 6e25 3344 312e r.jsVersion%3D1.  
3325 3342 2530 416e 6176 6967 6174 6f72 3%3B%0Anavigator  
2e76 6253 6372 6970 7445 6e61 626c 6564 .vbScriptEnabled  
2533 4474 7275 6525 3342 2530 410d 0a0d %3Dtrue%3B%0A...  
0a .

08:16:49.746507 207.166.87.157.64785 > 66.54.32.235.80: P 1349850533:1349851229(696) ack

2945117705 win 64860 [tos 0x10] (ttl 240, id 0, bad cksum 0!)

4510 02e0 0000 0000 f006 0000 cfa6 579d E.....W.  
4236 20eb fd11 0050 1df2 54e0 cd7d 41f5 B6.....P..T..}A.  
5018 fd5c 0000 0000 4745 5420 2f52 6561 P..|....GET./Rea  
6c4d 6564 6961 2f61 6473 2f63 6c69 636b lMedia/ads/click  
5f6c 782e 6367 692f 7777 772e 7573 6174 \_lx.cgi/www.usat  
6f64 6179 2e63 6f6d 2f73 706f 7274 732f oday.com/sports/  
6d69 6c6b 2f6c 6f61 642e 6874 6d2f 2525 milk/load.htm/%%



```

5241 4e44 2525 2f53 7065 6369 616c 312f
3230 3435 385f 4d69 6c6b 5f53 414d 4d59
5f32 3030 335f 3239 3835 2f63 6c65 6172
2e67 6966 2f25 0025 2045 5225 2520 4854
5450 2f31 2e31 0d0a 4163 6365 7074 3a20
2a2f 2a0d 0a41 6363 6570 742d 4c61 6e67
7561 6765 3a20 656e 2d75 730d 0a41 6363
6570 742d 456e 636f 6469 6e67 3a20 677a
6970 2c20 6465 666c 6174 650d 0a55 7365
722d 4167 656e 743a 204d 6f7a 696c 6c61
2f34 2e30 2028 636f 6d70 6174 6962 6c65
3b20 4d53 4945 2035 2e35 3b20 5769 6e64
6f77 7320 4e54 2035 2e30 290d 0a48 6f73
743a 2061 642e 7573 6174 6f64 6179 2e63
6f6d 0d0a 436f 6e6e 6563 7469 6f6e 3a20
4b65 6570 2d41 6c69 7665 0d0a 436f 6f6b
6965 3a20 5553 4154 494e 464f 3d55 4944
2533 4461 6138 3133 3237 3833 6438 3837
3337 303b 2052 4d49 443d 6161 3831 3332
3738 3364 3838 3863 6130 3b20 5449 443d
3173 3335 3064 6130 7570 6d68 7175 3b20
4146 4649 4c49 4154 455f 434f 4445 3d75
7361 3b20 5645 5254 4943 414c 5f43 4f44
453d 6e61 7469 6f6e 616c 3b20 5549 443d
6161 3831 3332 3738 3364 3838 3733 3730
3b20 5444 6174 613d 3b20 7631 7374 3d33
4444 3237 3938 4130 4432 3039 3135 333b
2042 726f 7773 6572 536e 6966 6665 723d
6e61 7669 6761 746f 722e 7479 7065 2533
4432 2533 4225 3041 6e61 7669 6761 746f
722e 7665 7273 696f 6e25 3344 352e 3525
3342 2530 416e 6176 6967 6174 6f72 2e6f
7325 3344 2532 3225 3230 5769 6e64 6f77
7325 3230 4e54 2532 3035 2e30 2532 3925
3232 2533 4225 3041 6e61 7669 6761 746f
722e 6a73 5665 7273 696f 6e25 3344 312e
3325 3342 2530 416e 6176 6967 6174 6f72
2e76 6253 6372 6970 7445 6e61 626c 6564
2533 4474 7275 6525 3342 2530 410d 0a0d
RAND%%/Special/
20458_Milk_SAMMY
_2003_2985/clear
.gif/%.%.ER%.HT
TP/1.1..Accept:.
/*..Accept-Lang
uage:.en-us..Acc
ept-Encoding:.gz
ip,.deflate..Use
r-Agent:.Mozilla
/4.0.(compatible
;.MSIE.5.5;.Wind
ows.NT.5.0)..Hos
t:.ad.usatoday.c
om..Connection:.
Keep-Alive..Cook
ie:.USATINFO=UID
%3Daa8132783d887
370;.RMID=aa8132
783d888ca0;.TID=
1s350da0upmhqu;.
AFFILIATE_CODE=u
sa;.VERTICAL_COD
E=national;.UID=
aa8132783d887370
;.TData=;.v1st=3
DD2798A0D209153;
.BrowserSniffer=
navigator.type%3
D2%3B%0Anavigato
r.version%3D5.5%
3B%0Anavigator.o
s%3D%22%20Window
s%20NT%205.0%29%
22%3B%0Anavigato
r.jsVersion%3D1.
3%3B%0Anavigator
.vbScriptEnabled
%3Dtrue%3B%0A...

```

## 1. Source of Trace

The source of this detect is from the logs provided by GIAC at <http://www.incidents.org/logs/Raw/2002.10.13>

## 2. Detect Was Generated By

To ease my search for interesting detects, I initially used Demarc PureSecure <http://www.demarc.com> and I ran snort with the command below against the RAW files provided at <http://www.incidents.org/logs/Raw> to upload the alerts into MySQL database. I disabled preprocessor stream4 and preprocessor stream4\_reassemble to get better result than default configuration, similar to Detect 1 above.

```
snort -p -l C:\PureSecure\sensor\logs -r C:\PureSecure\sensor\logs_raw\2002.10.13 -c
C:\PureSecure\sensor\conf\snort4.conf
```

This detect was generated from raw log file 2002.10.13.

1	WEB-MISC cisco /%% DOS attempt	web- application- attack	TCP	207.166.87.157:64785	66.54.32.235:80	Sensor01	8:16 AM - 11/13
1	WEB-MISC cisco /%% DOS attempt	web- application- attack	TCP	207.166.87.157:64785	66.54.32.235:80	Sensor01	8:16 AM - 11/13

Which is triggered by snort rule version 1.90 as below:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC cisco /%%  
DOS attempt"; flow:to_server,established; uricontent:"/%%"; classtype:web-application-attack; sid:1546;  
rev:4;)
```

### 3. Probability the Source Address Was Spoofed

Source address is very unlikely to be spoofed. These two events occurred after TCP three-way-handshake has been completed and TCP connection has been established. After TCP connection has been established, data was sent from source address to target address on port 80 that contained string /%% and triggered this event.

### 4. Description of Attack

A string of “/%%” was injected in HTTP GET request packet shown in the URL below:

GET./RealMedia/ads/click\_lx.cgi/www.usatoday.com/sports/milk/load.htm/%%RAN  
D%%/Special1/20458\_Milk\_SAMMY\_2003\_2985/clear.gif/%%.ER%%

If the target is Cisco router running a vulnerable IOS version, this packet can cause the router to crash. Some routers will automatically reboot, while others will require power cycling to reboot the routers before start routing packets again.

However, these two events detected above determined to be valid HTTP request on web server [www.usatoday.com](http://www.usatoday.com) , thus they are false positive.

### 5. Attack Mechanism

This denial of service attack on Cisco routers configured to run web services for router configurations and other information, that took advantage of vulnerability that exist on some IOS versions. When attacker sends packet that requesting URL in format shown below:

<http://<Router IP address>/%%>

to the vulnerable routers, these routers can crash, automatically rebooted or requires power cycling to start routing packets again.

However, the events detected above are targeting a web server [www.usatodays.com](http://www.usatodays.com) and not a router since a DNS name was used instead of IP address which requires for it to work. Therefore, the above events are false positives.

Philip Ljunberg in his GIAC GCIA Detect posting to [intrusions@incidents.org](mailto:intrusions@incidents.org) mailing list, detected 4 events and 2 of them are the same alerts as shown and discussed above. While the other 2 alerts a bit different as shown below:

[illegible][illegible]

## 7. Evidence of Active Targeting

## 8. Severity

$$\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network})$$

Countermeasures)

With each elements worth 1 to 5 points where 1 is the least and 5 is the highest.

Criticality: Since the attacker is targeting a web server. So, I gave 2 points.

Lethality: This attack can produced DoS, but this incident is false positive. So, I gave 1 point.

System Countermeasures: Assuming web server not vulnerable to this attack and this is a well known website that hopefully quite secure, so I gave 3 points.

Network Countermeasures: Due to unknown network and the nature of this attack, so I gave 1 point.

Severity = (2+1) – (3+1) = -1

In conclusion, this attack is insignificant.

## 9. Defensive Recommendation

No action should be taken since this is false positives. However, vulnerability scan should be done on all Cisco routers in the environment. If Cisco IOS version running is found to be vulnerable, actions need to be taken to secure the routers. IOS web service can be disabled on the router to eliminate this vulnerability. In addition, ACL can be added to prevent access to this HTTP port except for specific host running web server. Permanent solution to this is to patch the router using patch released by Cisco, available at [www.cisco.com](http://www.cisco.com).

## 10. Multiple Choice Test Question

```
09:16:49.626507 IP (tos 0x0, ttl 124, id 7952, len 737) 207.166.87.157.64785 > 66.54.32.235.80: P [bad tcp cksum ef9b (->fd81)!] 3447537467:3447538164(697) ack 502420461 win 64860 (DF)bad cksum 9cec (->52a2)!
```

```
0x0000      4500 02e1 1f10 4000 7c06 9cec cfa6 579d
0x0010      4236 20eb fd11 0050 cd7d 3f3b 1df2 53ed
0x0020      5018 fd5c ef9b 0000
```

```
09:16:49.746507 IP (tos 0x10, ttl 240, id 0, len 736) 207.166.87.157.64785 > 66.54.32.235.80: P [bad tcp cksum 0 (->3d6)!] 1349850533:1349851229(696) ack 2945117705 win 64860bad cksum 0 (->3da3)!
```

```
0x0000      4510 02e0 0000 0000 f006 0000 cfa6 579d
0x0010      4236 20eb fd11 0050 1df2 54e0 cd7d 41f5
0x0020      5018 fd5c 0000 0000
```

Q: What can you tell from detects and header information (in HEX) above?

A) These are TCP packets to establish HTTP connection

- B) These are TCP packets pushing data to web server
- C) These are UDP packets acknowledging data received from web server
- D) These are IP packets acknowledging data received from web server

Answer: B

References:

- 1- Ljunberg, Phillip;  
<http://cert.uni-stuttgart.de/archive/intrusions/2002/07/msg00211.html>
- 2- CVE-2000-0380;  
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0380>
- 3- Bugtraq; Cisco IOS HTTP %% Vulnerability;  
<http://www.securityfocus.com/bid/1154/info/>
- 4- Snort.org;  
<http://www.snort.org/snort-db/sid.html?sid=1546>

## 2.3 Assignment #2: Detect 3: MISC Tiny Fragment

### Packet 1:

Windump output:

19:02:07.776507 IP (tos 0x0, ttl 235, len 40) 192.9.100.88 > 32.245.235.65: tcp (frag 0:20@60824+)bad cksum 4b6e (->6184)!

```
0x0000 4500 0028 0000 3db3 eb06 4b6e c009 6458    E..(..=...Kn..dX
0x0010 20f5 eb41 0924 0050 0152 f5c0 0152 f5c0    ...A.$P.R...R..
0x0020 7204 0000 4f98 0000 0000 0000 0000    r...O.....
```

Snort output:

[\*\*] MISC Tiny Fragments [\*\*]

10/18-19:02:07.776507 192.9.100.88 -> 32.245.235.65

TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40 MF

Frag Offset: 0x1DB3 Frag Size: 0xFFFFFE261

+++++

### Packet 2:

Windump output:

01:51:58.976507 IP (tos 0x0, ttl 235, len 40) 192.9.100.88 > 32.245.67.198: tcp (frag 0:20@60824+)bad cksum f5e6 (->900)!

```
0x0000 4500 0028 0000 3db3 eb06 f5e6 c009 6458    E..(..=.....dX
0x0010 20f5 43c6 1384 0050 02ca 338c 02ca 338c    ..C...P..3...3.
0x0020 0404 0000 df2a 0000 0000 0000 0000    .....*
```

Snort output:

[\*\*] MISC Tiny Fragments [\*\*]

10/19-01:51:58.976507 192.9.100.88 -> 32.245.67.198

TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40 MF

Frag Offset: 0x1DB3 Frag Size: 0xFFFFFE261

+++++

### Packet 3:

Windump output:

15:27:48.046507 IP (tos 0x0, ttl 235, len 40) 192.9.100.88 > 32.245.218.210: tcp (frag 0:20@60824+)bad cksum 5cdb (->71f3)!

```
0x0000 4500 0028 0000 3db3 eb06 5cdb c009 6458    E..(..=...\...dX
0x0010 20f5 dad2 07c4 0050 0051 8bfa 0051 8bfa    .....P.Q...Q..
0x0020 0504 0000 a4f4 0000 0000 0000 0000    .....
```

Snort output:

[\*\*] MISC Tiny Fragments [\*\*]

10/19-15:27:48.046507 192.9.100.88 -> 32.245.218.210

TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40 MF

Frag Offset: 0x1DB3 Frag Size: 0xFFFFFE261

+++++

### Packet 4:

Windump output:

19:55:49.936507 IP (tos 0x0, ttl 235, len 40) 192.9.100.88 > 32.245.17.123: tcp (frag 0:20@60824+)bad cksum 2832 (->3b4b)!

```
0x0000 4500 0028 0000 3db3 eb06 2832 c009 6458    E..(..=...(2..dX
0x0010 20f5 117b 057c 0050 0146 ef82 0146 ef82    ...{.P.F...F..
0x0020 0504 0000 a998 0000 0000 0000 0000    .....
```

Snort output:

[\*\*] MISC Tiny Fragments [\*\*]

10/19-19:55:49.936507 192.9.100.88 -> 32.245.17.123

[illegible]

```
0x0000 4500 0028 0000 3db3 eb06 ef93 c009 6458      E..(.=.....dX
0x0010 20f5 491b 0a97 0050 020a ac4c 020a ac4c      ..I...P...L...L
0x0020 7004 0000 85c3 0000 0000 0000 0000      p.....
```

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

```
0x0000 4500 0028 0000 3db3 eb06 ecb9 c009 6458      E..(.=.....dX
0x0010 20f5 4cf3 11d3 0050 004b 2774 004b 2774      ..L...P.K't.K't
0x0020 2804 0000 d0dc 0000 0000 0000 0000      (.....
```

=+++++

```
0x0000 4500 0028 0000 3db3 eb06 f568 c009 6458      E..(.=...h..dX
0x0010 20f5 4346 0829 0050 014e 0358 014e 0358      ..CF.)P.N.X.N.X
0x0020 5404 0000 fd67 0000 0000 0000 0000      T...g.....
```

[illegible]

```
0x0000 4500 0028 0000 3db3 eb06 9d6c c009 6458      E...l.dX
0x0010 20f5 9943 082e 0050 02cc 0388 02cc 0388      ...C...P.....
0x0020 7004 0000 860a 0000 0000 0000 0000      p.....
```

Author retains full rights.

10/22-00:50:51.116507 192.9.100.88 -> 32.245.153.67  
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40 MF  
Frag Offset: 0x1DB3 Frag Size: 0xFFFFFE261

+++++

## 1. Source of Trace

The source of these trace are from the logs provided by GIAC at

[www.incidents.org/logs/Raw/2002.9.22](http://www.incidents.org/logs/Raw/2002.9.22)

[www.incidents.org/logs/Raw/2002.9.21](http://www.incidents.org/logs/Raw/2002.9.21)

[www.incidents.org/logs/Raw/2002.9.20](http://www.incidents.org/logs/Raw/2002.9.20)

[www.incidents.org/logs/Raw/2002.9.19](http://www.incidents.org/logs/Raw/2002.9.19)

## 2. Detect Was Generated By:

I used Demarc Puresecure with MySQL <http://www.demarc.com> to get an overall picture of the events from various logs provided by GIAC. Then I ran snort command against the raw files downloaded from <http://www.incident.org/logs/Raw> to upload these events into MySQL database. I disabled preprocessor stream4 and preprocessor stream4\_reassemble to get better result as I did in Detect 1. The command I ran is similar to one shown below:

```
snort -p -l c:\puresecure\sensor\logs -r c:\puresecure\sensor\logs_raw\2002.9.22  
-c c:\puresecure\sensor\conf\snort4.conf
```

Once I found an interesting events on PureSecure console, then I ran windump to capture the event from the raw file as shown below:

```
C:\PureSecure\raw>windump -vvv -Xx -r 2002.9.22 "src 192.9.100.88"
```

The alert generated by snort also captured from the snort logs, and each corresponding alerts detected are shown using windump and snort outputs respectively.

The above events were detected from raw files 2002.9.22, 2002.9.21, 2002.9.20 and 2002.9.19

The Snort rule that triggered these alerts is shown below:

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Tiny  
Fragments";fragbits:M;dsiz< 25;classtype:bad-unknown;sid:522;rev:1;)
```

## 3. Probability the Source Address Was Spoofed

Very unlikely that source address was spoofed. TCP connection could have been established that showed TCP three-way-handshake was completed when these packets were sent but not captured by IDS rules. Another possibility is that this is



a clever scan that sent fragmented TCP packet with Reset flag set to confuse firewall that does not support fragment reassembly, and then waiting for respond from target host or icmp host unreachable respond from router.

Whois query also shows that source address is belongs to Sun Microsystems Corp.

Final results obtained from whois.arin.net.  
Results:

OrgName: Sun Microsystems, Inc  
OrgID: SUN  
Address: 901 San Antonio Road  
City: Palo Alto  
StateProv: CA  
PostalCode: 94303-4900  
Country: US

NetRange: 192.9.10.0 - 192.9.199.255  
CIDR: 192.9.10.0/23, 192.9.12.0/22, 192.9.16.0/20, 192.9.32.0/19, 192.9.64.0/18, 192.9.128.0/18, 192.9.192.0/21  
NetName: SUN3  
NetHandle: NET-192-9-10-0-1  
Parent: NET-192-0-0-0-0  
NetType: Direct Assignment  
NameServer: NS.SUN.COM  
NameServer: NS-BRM.SUN.COM  
NameServer: NS.USEC.SUN.COM  
NameServer: NS-OS.SUN.COM  
Comment:  
RegDate: 1983-10-17  
Updated: 2002-01-16

TechHandle: IS189-ARIN  
TechName: Sun Microsystems, Inc.  
TechPhone: +1-303-272-7000  
TechEmail: Netmaster@sun.com

# ARIN WHOIS database, last updated 2003-04-22 20:10

#### 4. Description of Attack

This attack is using combination of unique fragmentation on TCP packet characteristics and TCP flag Reset being set. The nature of the attack is quite similar to inverse mapping techniques that can compile a list of networks or hosts that are unreachable and then use the converse of that map to determine where things probably are. <sup>[7]</sup>

The target is being probed on port 80, which is the most common port in use. The almost identical characteristics of these fragments were sent to daily to different host on the same first two octet of the IP address. See below:

192.9.100.88 > **32.245.235.65**  
192.9.100.88 > **32.245.67.198**  
192.9.100.88 > **32.245.218.210**

192.9.100.88 > **32.245.17.123**  
192.9.100.88 > **32.245.73.27**  
192.9.100.88 > **32.245.76.243**  
192.9.100.88 > **32.245.67.70**  
192.9.100.88 > **32.245.153.67**

Each of the packet has RST flag set, TTL of 235, fragment ID 0, 20 bytes size, offset of 60824 and More Fragment (MF) fragment flag set.

## 5. Attack Mechanism

Let us discuss the nature of these packet in detail. First, let us examine the fragmentation flag shown in the windump output below which is identical on each fragment occurred daily:

**frag 0:20@60824+**

The “frag 0” shown above specify the value of the identification field in the IP header

”20” shows the size of the data which is 20 bytes

”60824” is the offset of the data in the fragment.

”+” sign shows that More Fragment (MF) flag is set.

The non-zero offset number followed by a “+” shows us that this packet is in the middle of fragmentation. The first packet in beginning of fragmented packet should shows “@0+” for zero offset and more packet to follow. While the last fragmented packet should shows “@60824” for the last offset number without the “+” sign, means no more packet to follow.

Since we know that these packets detected above are all middle fragments, however all these packet also contain TCP header information since only the first fragment can contain the embedded protocol’s header information. This is weird and does not make sense. TCP is also known to avoid fragmentations since if one fragment is lost that came from TCP segment, TCP will time out and resubmit the entire diagram. Due to <sup>[6]</sup>

Since header is added in each of these fragment, let’s examine the TCP flags set in each of these fragments:

Packet	Bolded payload (In HEX)	Header Length (4 bits in binary)	Reserved Bits (6 bits in binary)	TCP Flags (Urg, Ack, Psh, Rst, Syn, Fin)	Meaning
Packet1	7204	0110	001000	000100	RST flag is set. Reserved bit also set
Packet2	0404	0000	010000	000100	RST flag is set. Reserved bit also set
Packet3	0504	0000	010100	000100	RST flag is set. Reserved

					bits also set
Packet4	0504	0000	010100	000100	RST flag is set. Reserved bits also set
Packet5	7004	0111	000000	000100	RST flag is set.
Packet6	2804	0010	100000	000100	RST flag is set. Reserved bits also set
Packet7	5404	0101	010000	000100	RST flag is set. Reserved bits also set
Packet8	7004	0111	000000	000100	RST flag is set.

Many of the above packets has Reserved bit set. However, RFC 3360 stated that since these reserved bits are for future use, the use of these bits is prohibited, except two bits to the left of URG flag bit as shown below.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Header Length				Reserved				C	E	U	A	P	R	S	F
								W	C	R	C	S	S	Y	I
								R	E	G	K	H	T	N	N

These two bits are only used for CWE (Congestion Windows Reduced) and ECE (Explicit Congestion Notification (ECN) Echo) as shown in figure above. <sup>[8]</sup>

All these characteristics are proves that this could be crafted packet for the purpose of probing networks or hosts, that the I have not seen before.

## 6. Correlations

I am also seeing the same source address 192.9.100.88 probing the network that I have access to. See below for the TCP payload information captured:

Packet2: 22 Sep 2002 06:32:32 192.9.100.88 -> MY.NET.208.159

TCP Payload in Hex: 0ED1 **0050** 0095 2F18 0095 2F18 **0504** 0000 D71C 0000

The TCP is targeted to port 80 and the above TCP flag also shows RST flag set and Reserved bits set. This is very similar to the packets we discussed in this detect.

Reto Baumann <sup>[4]</sup> detected similar packet from the GIAC Raw logs. Andrew Rucker Jones commented that even though Fragment ID 0 is rarely used, it is sometimes used even in the middle of conversation. <sup>[5]</sup>

## 7. Evidence of Active Targeting

This is not active targeting since targets are chosen in random. This is probably slow scan attempted to do inverse mapping of networks or hosts available in the

internet.

## 8. Severity

Severity is calculated using formula:

$$\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures})$$

With each elements worth 1 to 5 points where 1 is the least and 5 is the highest.

**Criticality:** In the HEX showed that targeted port 80, which possibly targeting a web server. So, I gave 4 points.

**Lethality:** This packet could be lethal if target is vulnerable to malformed fragment packet. So, I gave 4 point.

**System Countermeasures:** Server might have been patched and no longer vulnerable to malformed fragment packets since these packets keep coming almost daily, so I gave 3 points.

**Network Countermeasures:** There is probably existed some network countermeasures, so I gave 3 point.

$$\text{Severity} = (4+4) - (3+3) = 2$$

## 9. Defensive Recommendation

Install firewall that support fragment reassembly at defense perimeter. Every fragmented packet will first reassemble and analyze to detect whether it is normal fragmented packet or malformed packet before allow to pass through. Normal fragmented packet will then allowed through while malformed fragmented packet will be dropped.

## 10. Multiple Choice Test Question

Q: From a TCP header below, what flag(s) is(are) set?

0ED1 0050 0095 2F18 0095 2F18 0504 0000 D71C 0000

- A) SYN and RST flags are set
- B) SYN flag is set
- C) RST and URG flags are set
- D) RST is set and Reserve bit is also set

Answer is D.

Bytes 0x0504 above is translated to binary value of 0000 000101 000100

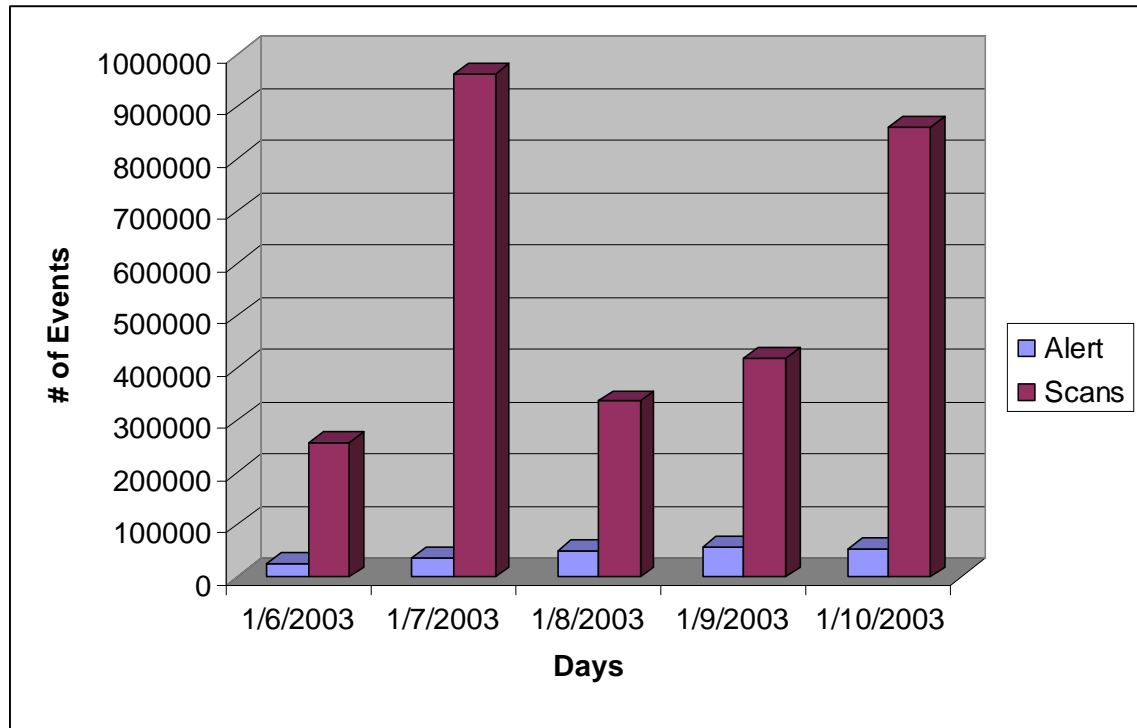
Header Length	Reserve Bit	URG	ACK	PSH	RST	SYN	FIN
0000	000101	0	0	0	1	0	0

#### References:

1. Whois query: <http://www.geektools.com/cgi-bin/proxy.cgi>
2. Ardoin, Cy; Oct 16 1996;  
<http://www.netsys.com/firewalls/firewalls-9610/0570.html>
3. Metati, Prabhaker; Spoofing and Fragmentation;  
<http://www.cs.wright.edu/~pmateti/InternetSecurity/Lectures/IPexploits/>
4. Baumann, Reto; LOGS:GIAC GCIA Version 3.3 Practical Detect(s); 2003 Mar 11;  
<http://cert.uni-stuttgart.de/archive/intrusions/2003/03/msg00136.html>
5. Jones, Andrew Rucker; Re: LOGS: GIAC GCIA Version 3.3 Practical Detect(s); 2003 Mar 11;  
<http://cert.uni-stuttgart.de/archive/intrusions/2003/03/msg00156.html>
6. Stevens, Richard W; TCP/IP Illustrated Volume 1, The Protocols; Nov 2001; pages 148-153
7. Northcutt, Stephen; Novak, Judy; Network Intrusion Detection, An Analyst's Handbook; Second Edition; Sept 2000
8. RFC 3360; Inappropriate TCP Resets Considered Harmful;  
<http://www.faqs.org/rfcs/rfc3360.html>
- 9.

## Assignment #3: Analyze This.

### 3.1 Overview of analysis



Daily graph of number of events generated by Alerts and Scans:

Year 2003	6-Jan	7-Jan	8-Jan	9-Jan	10-Jan
Alert	26416	37731	50608	57620	55258
Scans	258006	964355	336962	417996	860281

Looking at the above bar graph of daily events in the University's security log analyzed, showed huge differences of scans activities and alerts detected in the environment. A big spike in scans occurred on Tuesday and Friday, while alerts graphs are in incline trend.

Even with some security measures already in place, Nimda break loose in the campus network. A lot of internal machines were infected, and/or in process of getting infected. Besides Nimda, many internal machines are also compromised by Trojan, and backdoor controlled by outsiders via IRC XDCC.

Scans activities are very high, either due to Nimda probing activities, or scanning tools such as Queso, nmap and others. Two internal hosts i.e. MY.NET.84.151 and MY.NET.88.193 have been determined to be infected by Code Red worm and caused a massive traffic in internal network. Action need to be taken as soon as possible to take these machine off line and clean them up from worm.

### 3.2 List of files analyzed:

Here are the list of files I choose to analyze from <http://www.incidents.org/logs>

Alert Files	Size (Bytes)
alert.030106.gz	700,538
alert.030107.gz	1,519,896
alert.030108.gz	1,030,895
alert.030109.gz	1,268,750
alert.030110.gz	1,705,287

OOS Files	Size (Bytes)
OOS_Report_2003_01_06_18360.txt	220,163
OOS_Report_2003_01_07_31845.txt	332,803
OOS_Report_2003_01_08_8856.txt	291,843
OOS_Report_2003_01_09_12713.txt	209,923
OOS_Report_2003_01_10_4480.txt	230,403

Scan Files	Size (Bytes)
scans.030106.gz	2,399,652
scans.030107.gz	7,823,839
scans.030108.gz	2,712,394
scans.030109.gz	3,441,161
scans.030110.gz	6,850,256

227633 snort events analyzed in Alert files listed above.

2837600 scans events analyzed in Scans files listed above.

4149 events analyzed in OOS files listed above.

### 3.3 A list of detects

Below is a list of detects prioritized by severity or number of occurrences. Brief description of these events, analysis identifying relationships between sources and targets machines, correlation with other GIAC papers and recommendations are also included. To fulfill the requirement for this paper, I also included a Link Graph and five external source addresses and registration information about these addresses.

There are 227633 alerts detected from Snort log provided by GIAC. Below is the list of Alerts that is more than 1 detect.

Rank	Signature	# of Alerts	# of Sources	# of Dests
#1	High port 65535 tcp - possible Red Worm - traffic	95437	294	276
#2	SMB Name Wildcard	39725	1109	918
#3	Spp_http_decode: IIS Unicode attack detected	31952	420	634

#4	Watchlist 000220 IL-ISDNNET-990517	26083	79	120
#5	TFTP - External UDP connection to internal tftp server	17283	9	5
#6	High port 65535 udp - possible Red Worm - traffic	4391	91	108
#7	Spp_http_decode: CGI Null Byte attack detected	2197	55	80
#8	Watchlist 000222 NET-NCFC	1816	26	26
#9	Possible trojan server activity	1556	18	16
#10	Port 55850 tcp - Possible myserver activity - ref. 010313-1	1335	49	51
#11	Queso fingerprint	1253	86	36
#12	<a href="#">IDS552/web-iis_IIS ISAPI Overflow ida nosize [arachNIDS]</a>	1185	1036	559
#13	Null scan!	601	32	27
#14	EXPLOIT x86 NOOP	558	46	58
#15	Incomplete Packet Fragments Discarded	431	33	19
#16	TFTP - Internal TCP connection to external tftp server	424	2	2
#17	SUNRPC highport access!	368	38	42
#18	IRC evil – running XDCC	197	8	21
#19	SMB C access	152	95	15
#20	TCP SRC and DST outside network	143	14	18
#21	NMAP TCP ping!	132	54	44
#22	EXPLOIT x86 setuid 0	84	63	39
#23	ICMP SRC and DST outside network	74	6	7
#24	TFTP - Internal UDP connection to external tftp server	56	14	11
#25	EXPLOIT x86 setgid 0	42	38	30
#26	Port 55850 udp - Possible myserver activity - ref. 010313-1	15	6	3
#27	Attempted Sun RPC high port access	14	5	7
#28	RFB - Possible WinVNC - 010708-1	13	9	8
#29	Tiny Fragments - Possible Hostile Activity	11	5	4
#30	EXPLOIT x86 stealth noop	11	4	5
#31	TFTP - External TCP connection to internal tftp server	7	1	1
#32	EXPLOIT NTPDX buffer overflow	6	6	6
#33	NIMDA - Attempt to execute cmd from campus host	4	4	2

### 3.4 Below are descriptions of the Alerts events

**Alert#1: High port 65535 tcp - possible Red Worm – traffic**

**Alert#6: High port 65535 udp - possible Red Worm – traffic**

**Alert#3: spp\_http\_decode: IIS Unicode attack detected**

**Alert#12: [IDS552/web-iis\\_IIS ISAPI Overflow ida nosize \[arachNIDS\]](#)**

All the above alerts showed that there are a lot of Code Red worm activities in the environment. Code Red is self-propagating malicious code that exploits IIS-enabled machines vulnerable to exploit exist in CERT advisory [CA-2001-13 Buffer](#)

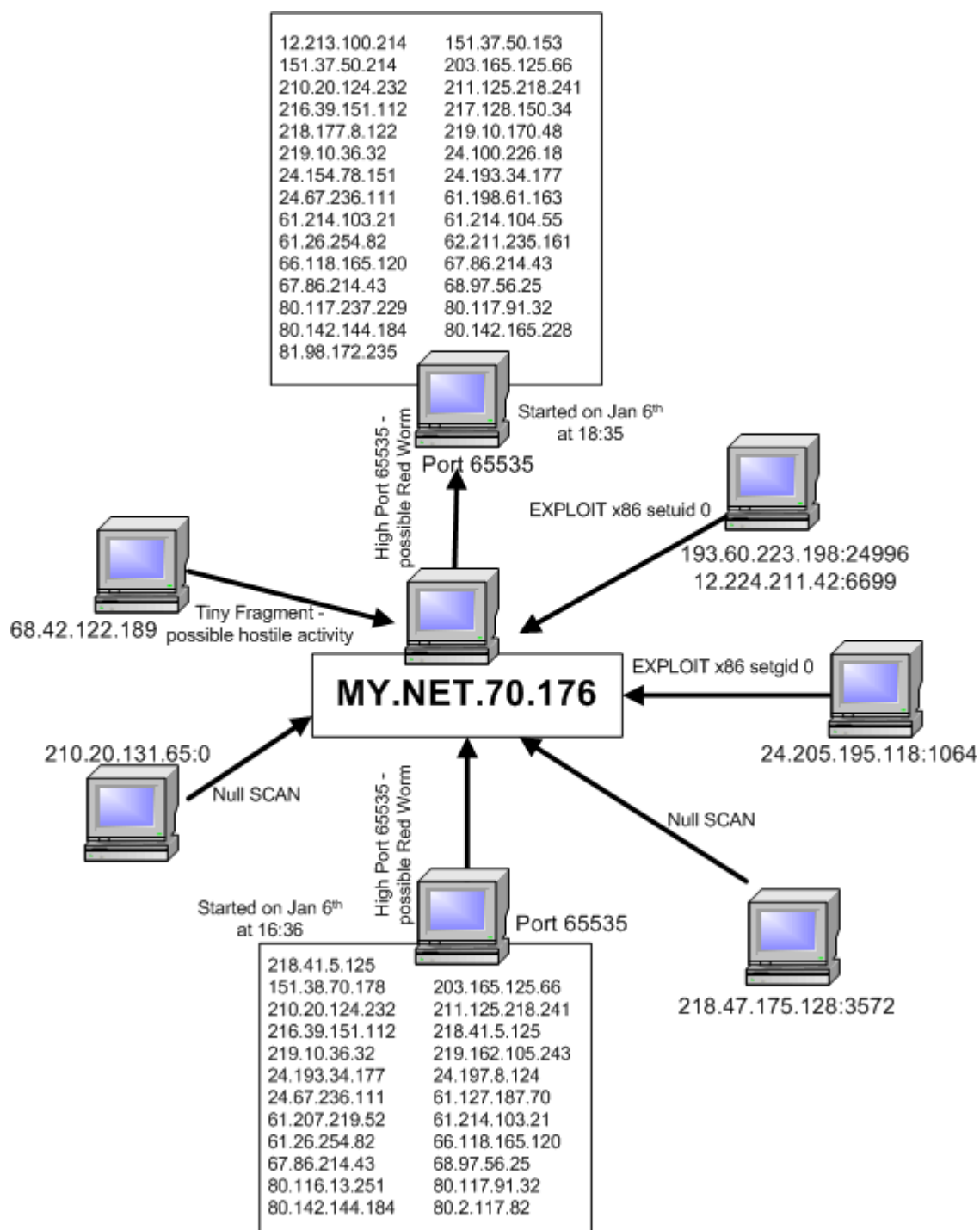


/default.ida?NN  
NN  
NN  
NN  
NNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u780  
1%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u5  
31b%u53ff%u0078%u0000%u00=a

[48]

One of characteristics of Code Red infected machines as shown in Link Graph above, are really noisy in network. Thus, machines with a less alerts generated could be false positives. There is also possibility that the high port scan on port 65535 is not Code Red, but caused buy other worm scan or malicious software. Below is a list of internal machines. The top of the list are surely have been infected by Code Red.

Internal Source	# of Alerts	# of Targets
MY.NET.84.151	21336	129
MY.NET.88.193	17410	135
MY.NET.83.146	2204	85
MY.NET.70.176	623	31
MY.NET.150.213	199	14
MY.NET.91.72	89	10
MY.NET.88.226	63	2
MY.NET.6.40	22	9
MY.NET.113.4	21	8
MY.NET.140.136	16	9
MY.NET.150.83	13	2
MY.NET.198.220	5	1
MY.NET.154.30	4	1
MY.NET.132.50	4	3
MY.NET.150.215	3	1
MY.NET.85.91	2	1
MY.NET.82.114	1	1
MY.NET.182.109	1	1
MY.NET.151.128	1	1



Link Graph: Code Red infected machine

The top 5 external Code Red sources:

IP address	DNS name	# of Alerts
172.186.226.148	?? last hit rt-frrtq08.proxy.aol.com (195.93.89.173)	5102
<p>OrgName: America Online OrgID: AOL</p> <p>NetRange: 172.128.0.0 - 172.191.255.255 CIDR: 172.128.0.0/10 NetName: AOL-172BLK NetHandle: NET-172-128-0-0-1 Parent: NET-172-0-0-0-0 NetType: Direct Allocation NameServer: DAHA-01.NS.AOL.COM NameServer: DAHA-02.NS.AOL.COM Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE RegDate: 2000-03-24 Updated: 2002-08-09</p> <p>TechHandle: AOL-NOC-ARIN TechName: America Online, Inc. TechPhone: +1-703-265-4670 TechEmail: domains@aol.net</p> <p>OrgAbuseHandle: AOL382-ARIN OrgAbuseName: Abuse OrgAbusePhone: +1-703-265-4670 OrgAbuseEmail: abuse@aol.net</p> <p>OrgNOCHandle: AOL236-ARIN OrgNOCName: NOC OrgNOCPhone: +1-703-265-4670 OrgNOCEmail: noc@aol.net</p> <p>OrgTechHandle: AOL-NOC-ARIN OrgTechName: America Online, Inc. OrgTechPhone: +1-703-265-4670 OrgTechEmail: <a href="mailto:domains@aol.net">domains@aol.net</a></p>		
67.69.224.186	Toronto-HSE- ppp3845963.sympatico.ca	2530
<p>Bell Canada BELLNEXXIA-11 (NET-67-68-0-0-1) <a href="http://67.68.0.0">67.68.0.0</a> - <a href="http://67.70.255.255">67.70.255.255</a> Bell Sympatico HSE1129-CA (NET-67-69-192-0-1) <a href="http://67.69.192.0">67.69.192.0</a> - <a href="http://67.69.255.255">67.69.255.255</a></p> <p>OrgName: Bell Canada OrgID: LINX</p> <p>NetRange: <a href="http://67.68.0.0">67.68.0.0</a> - <a href="http://67.70.255.255">67.70.255.255</a> CIDR: <a href="http://67.68.0.0/15">67.68.0.0/15</a>, <a href="http://67.70.0.0/16">67.70.0.0/16</a> NetName: BELLNEXXIA-11 NetHandle: NET-67-68-0-0-1 Parent: NET-67-0-0-0-0 NetType: Direct Allocation NameServer: <a href="http://NS3.BELLGLOBAL.COM">NS3.BELLGLOBAL.COM</a> NameServer: <a href="http://NS4.BELLGLOBAL.COM">NS4.BELLGLOBAL.COM</a> Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE RegDate: 2002-04-18 Updated: 2003-01-16</p> <p>TechHandle: MK1209-ARIN TechName: Khalid, Mohammad TechPhone: +1-800-450-7771 TechEmail: noc@in.bell.ca</p>		

OrgTechHandle: SYSAD1-ARIN  
OrgTechName: Sys Admin  
OrgTechPhone: +1-613-785-0886  
OrgTechEmail: [ip\\_prov@bellglobal.com](mailto:ip_prov@bellglobal.com)

80.200.137.128

??

2462

inetnum: [80.200.0.0](#) - [80.200.255.255](#)  
netname: BE-SKYNET-20011108  
descr: ADSL Customers  
descr: Skynet Belgium  
country: BE  
admin-c: JFS1-RIPE  
tech-c: PDH16-RIPE  
status: ASSIGNED PA  
mnt-by: SKYNETBE-MNT  
changed: ripe@skynet.be 20011212  
source: RIPE

route: [80.200.0.0/15](#)  
descr: SKYNETBE-CUSTOMERS  
origin: AS5432  
notify: noc@skynet.be  
mnt-by: SKYNETBE-MNT  
changed: noc@skynet.be 20011116  
source: RIPE

person: Jean-Francois Stenuit  
address: Belgacom Skynet NV/SA  
address: Rue Carli 2  
address: B-1140 Bruxelles  
address: Belgium  
phone: +32 2 706-1311  
fax-no: +32 2 706-1150  
e-mail: jfs@skynet.be  
nic-hdl: JFS1-RIPE  
remarks: -----  
remarks: Network problems to: noc@skynet.be  
remarks: Peering requests to: peering@skynet.be  
remarks: Abuse notifications to: abuse@skynet.be  
remarks: -----  
mnt-by: SKYNETBE-MNT  
changed: jfs@skynet.be 19970707  
changed: ripe@skynet.be 20021125  
source: RIPE

person: Pieterjan d'Hertog  
address: Belgacom Skynet sa/nv  
address: 2 Rue Carli  
address: B-1140 Brussels  
address: Belgium  
phone: +32 2 706 13 11  
fax-no: +32 2 706 13 12  
e-mail: piet@skynet.be  
nic-hdl: PDH16-RIPE  
remarks: -----  
remarks: Network problems to: noc@skynet.be  
remarks: Peering requests to: peering@skynet.be  
remarks: Abuse notifications to: abuse@skynet.be  
remarks: -----  
mnt-by: SKYNETBE-MNT  
changed: jfs@skynet.be 19990415  
changed: piet@skynet.be 19991210  
changed: piet@skynet.be 20000302  
changed: piet@skynet.be 20020329

source: RIPE		
193.252.60.115	??	2010
<p>inetnum: <a href="#">193.252.60.0</a> - <a href="#">193.252.60.255</a>  netname: IP2000-ADSL-BAS  descr: France Telecom IP2000 ADSL BAS  country: FR  admin-c: WITR1-RIPE  tech-c: WITR1-RIPE  status: ASSIGNED PA  remarks: for hacking, spamming or security problems send mail to  remarks: postmaster@wanadoo.fr AND abuse@wanadoo.fr  remarks: for ANY problem send mail to <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a>  notify: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a>  mnt-by: FT-BRX  changed: <a href="mailto:gestionip.ft@francetelecom.fr">gestionip.ft@francetelecom.fr</a> 20000412  changed: <a href="mailto:gestionip.ft@francetelecom.fr">gestionip.ft@francetelecom.fr</a> 20001024  changed: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a> 20010517  source: RIPE</p> <p>route: <a href="#">193.252.0.0/18</a>  descr: France Telecom  descr: FTI  origin: AS3215  mnt-by: FT-BRX  changed: <a href="mailto:gestionip.ft@francetelecom.fr">gestionip.ft@francetelecom.fr</a> 20001018  source: RIPE</p> <p>role: Wanadoo Interactive Technical Role  address: WANADOO INTERACTIVE  address: 48 rue Camille Desmoulins  address: 92791 ISSY LES MOULINEAUX CEDEX 9  address: FR  phone: +33 1 58 88 50 00  e-mail: abuse@wanadoo.fr  e-mail: postmaster@wanadoo.fr  admin-c: FTI-RIPE  tech-c: TEFS1-RIPE  nic-hdl: WITR1-RIPE  notify: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a>  mnt-by: FT-BRX  changed: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a> 20010504  changed: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a> 20010912  changed: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a> 20011204  source: RIPE</p>		
80.14.209.119	??	1722
<p>inetnum: <a href="#">80.14.209.0</a> - <a href="#">80.14.209.255</a>  netname: IP2000-ADSL-BAS  descr: BSREI105 Reims Bloc2  country: FR  admin-c: WITR1-RIPE  tech-c: WITR1-RIPE  status: ASSIGNED PA  remarks: for hacking, spamming or security problems send mail to  remarks: postmaster@wanadoo.fr AND abuse@wanadoo.fr  remarks: for ANY problem send mail to <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a>  mnt-by: FT-BRX  changed: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a> 20020220  changed: <a href="mailto:gestionip.ft@francetelecom.com">gestionip.ft@francetelecom.com</a> 20020723  source: RIPE</p> <p>route: <a href="#">80.14.0.0/16</a>  descr: France Telecom</p>		

```
descr: Wanadoo Interactive
remarks: -----
remarks: For Hacking, Spamming or Security problems
remarks: send mail to abuse@francetelecom.net
remarks: -----
origin: AS3215
mnt-by: RAIN-TRANSPAC
mnt-by: FT-BRX
changed: karim@rain.fr 20011221
source: RIPE

role: Wanadoo Interactive Technical Role
address: WANADOO INTERACTIVE
address: 48 rue Camille Desmoulins
address: 92791 ISSY LES MOULINEAUX CEDEX 9
address: FR
phone: +33 1 58 88 50 00
e-mail: abuse@wanadoo.fr
e-mail: postmaster@wanadoo.fr
admin-c: FTI-RIPE
tech-c: TEFS1-RIPE
nic-hdl: WITR1-RIPE
notify: gestionip.ft@francetelecom.com
mnt-by: FT-BRX
changed: gestionip.ft@francetelecom.com 20010504
changed: gestionip.ft@francetelecom.com 20010912
changed: gestionip.ft@francetelecom.com 20011204
source: RIPE
```

Three of the above external Code Red sources are XDSL connected machines, one via dialup (PPP) and one undertermined AOL client (assumed to be dialup). All the above ISPs need to be notified about their clients' Code Red infected machines.

Correlation: Joe Ellis <sup>[10]</sup> noted this event as known Code Red alerts. He also agreed the possibilities of false positive on low generating alert sources.

Recommendation: Code Red infected machines need to be taken off network immediately. Then cleaned from the worm and patched to ensure no future infection. To stop any attempts to exploit Unicode vulnerabilities from the Internet, Unicode rules need to be added to egress/ingress filter at outer firewall, to drop packet matching this rule.

### **Alert#33: NIMDA - Attempt to execute cmd from campus host**

Nimda worm was found on September 18<sup>th</sup>, 2001 and it spread quickly around the world. Nimda is a complex virus with a mass mailing worm component which spreads itself in attachments named README.EXE if affected. <sup>[25]</sup>

It is propagating with unprecedented speed across the Internet. The worm appears to have at least four distinct propagation mechanisms, and infect hosts running any version of Windows. The network activity caused by the worm has resulted in an effective denial of service attack at many sites.

The worm scans the Internet for IIS servers and attempts to exploit a number of IIS vulnerabilities to gain control of a victim host including exploitation of the “IIS Directory Transversal Vulnerability” and utilization of backdoors left behind by previous Code Red II worm. Once in control of a victim IIS server, the worm uses TFTP to transfer its code from the attacking machine to the victim.

The worm also takes advantage of a known vulnerability called “Automatic Execution of Embedded MIME Types” discussed in CERT advisory CA-2001-06. Exploiting this vulnerability, Nimda harvests email addresses from the Windows address book and user’s inboxes and sends itself to all addresses as an attachment named “readme.exe”.

If the worm successfully infects a web server, it uses the HTTP service to propagate itself to clients who browse the web server’s pages. Upon infecting a victim server, the worm creates a copy of itself named “readme.eml” and traverses the directory tree, including network shares, searching for web-related files such as those with .html, .htm or .asp extensions, and append a piece of JavaScript to the file. The JavaScript forces a download of readme.eml to any client that views the file via a browser.

The worm is also network aware and propagates via open file shares. It will copy itself to all directories, including those found in network shares, for which user has write permission. This worm copies are named “readme.eml”. Any other host that accesses the share and executes or previews one of these files can become infected. <sup>[26]</sup>

#### **Alert#2) SMB Name Wildcard**

This is NETBIOS SMB Service scan on port 137. There are two possibilities. One is possibilities of scriptkiddies are more aware of NBTSTAT tool capabilities in gathering machine information and secondly, the spread of an internet worm known as network.vbs. Use of standard Netbios “nbtstat” frames, will bring out a node status response from Netbios and SAMBA clients. This response contains a listing of any Netbios names known to that node. <sup>[6]</sup>

Correlation: Brian Credeur noted this event in his GCIA paper as reconnaissance technique. <sup>[27]</sup>

Recommendation: Outer firewall should be configured to block any NetBIOS traffic from going into internal network. That includes all TCP and UDP traffic to ports 135-139.

#### **Alert#4) Watchlist 000220 IL-ISDNNET-990517**

All 76 sources 212.179.x.x of this large number of scans were from cable modem hosts on from Israel. These addresses have been added to this Watchlist, probably

because of previous attack or stream of suspicious packets and activities, originating and/or targeting these networks. The name registration in Whois database is as below:

```
inetnum: 212.179.127.0 - 212.179.127.127
netname: ARAVA-DEVELOPMENT-COMPANY-LTD
descr: ARAVA-DEVELOPMENT-LAN
country: IL
admin-c: ES4966-RIPE
tech-c: NP469-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 20000525
source: RIPE

route: 212.179.64.0/18
descr: ISDN Net Ltd.
origin: AS8551
notify: hostmaster@bezeqint.net
mnt-by: AS8551-MNT
changed: hostmaster@bezeqint.net 20020618
source: RIPE

person: Eran Shchori
address: BEZEQ INTERNATIONAL
address: 40 Hashacham Street
address: Petach-Tikva 49170 Israel
phone: +972 3 9257710
fax-no: +972 3 9257726
e-mail: hostmaster@bezeqint.net
nic-hdl: ES4966-RIPE
changed: registrar@ns.il 20000309
source: RIPE
```

01/08-08:01:43.829910	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1090
01/08-08:01:44.298288	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1091
01/08-08:01:44.328758	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1090
01/08-08:01:44.459532	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1091
01/08-08:01:44.460446	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1091
01/08-08:01:44.460954	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1091
01/08-08:01:44.514752	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1090
01/08-08:01:44.516136	Watchlist 000220 IL-ISDN-990517	212.179.107.226:80 -> MY.NET.177.58:1090

Connection to local machines on Port 1090 as shown in Snort log above showed why these are suspicious. Port 1090 is a well known port used by a Trojan called Xtreme.<sup>[14]</sup>

Correlation: Mark Menke noted this event with suspicious telnet connectivity into the internal network from Watchlist addresses.<sup>[28]</sup>

Recommendation: MY.NET.177.58 machine and other affected hosts need to be cleaned from Trojan and any backdoor installed. Usually, if backdoor already installed, the safest thing to do is to format the harddrive and fully reinstall the OS.



**Alert#5: TFTP - External UDP connection to internal tftp server**  
**Alert#16: TFTP - Internal TCP connection to external tftp server**  
**Alert#24: TFTP - Internal UDP connection to external tftp server**  
**Alert#31: TFTP - External TCP connection to internal tftp server**

#### **External to Internal**

01/08-07:49:35.415308 TFTP - Internal TCP connection to external tftp server 209.126.214.14:69 -> MY.NET.70.225:3979  
01/08-07:49:35.721862 TFTP - Internal TCP connection to external tftp server 209.126.214.14:69 -> MY.NET.70.225:3979  
01/08-07:49:38.591639 TFTP - Internal TCP connection to external tftp server 209.126.214.14:69 -> MY.NET.70.225:3979  
01/08-07:49:38.749840 TFTP - Internal TCP connection to external tftp server 209.126.214.14:69 -> MY.NET.70.225:3979  
01/08-07:49:38.756097 TFTP - Internal TCP connection to external tftp server 209.126.214.14:69 -> MY.NET.70.225:3979

#### **Internal to External**

01/08-07:49:35.328148 TFTP - Internal TCP connection to external tftp server MY.NET.70.225:3979 -> 209.126.214.14:69  
01/08-07:49:35.415558 TFTP - Internal TCP connection to external tftp server MY.NET.70.225:3979 -> 209.126.214.14:69  
01/08-07:49:38.623538 TFTP - Internal TCP connection to external tftp server MY.NET.70.225:3979 -> 209.126.214.14:69  
01/08-07:49:38.668266 TFTP - Internal TCP connection to external tftp server MY.NET.70.225:3979 -> 209.126.214.14:69  
01/08-07:49:38.668383 TFTP - Internal TCP connection to external tftp server MY.NET.70.225:3979 -> 209.126.214.14:69

It is very unusual and suspicious for TFTP connections to and from the Internet. TFTP (Trivial File Transfer Protocol) usually used to upgrade router configuration files that can be automated. The same protocol is also used by Nimda worm to transfer its files. Snort log shown above showed a very active connection between MY.NET.70.225 and external IP address 209.126.214.14 which is suspicious. Without detail information, I can only conclude that MY.NET.70.225 is whether has been compromised or misconfigured.

Correlation: Joe Ellis <sup>[10]</sup> and Michael Wilkinson <sup>[29]</sup> also noted of this events in their paper. There are two possibilities that can happen from this scenario, which is misconfigured router or host already been compromised.

Recommendation: Local host MY.NET.70.225 need to be check for possibilities of misconfiguration or compromised.

#### **Alert#7: spp\_http\_decode: CGI Null Byte attack detected**

This is part of pre-processor that is looking for a string of “%00” in payload of http request. However, it can produce false positive for sites that use cookies encrypted data or SSLencrypted message on port 443. <sup>[11]</sup>

This attack is also known as “Poison NULL byte attack” that “%00” string appended to a URL to confuse a perl script about where the end of an input. <sup>[12] [13]</sup>

Correlation: Pedro Bueno noted in his paper about this event that false positive can occur with sites that included urlencoded binary data. <sup>[34]</sup>

Recommendation: Refine this rule to make more specific in detecting CGI Null Byte attack, and reducing false positive.

#### **Alert#8: Watchlist 000222 NET-NCFC**

This Watchlist is to monitor activities of IP addresses 159.226.x.x originating from The Computer Network Center Chinese Academy of Sciences as shown below from Whois query. Since these IP address has been added to monitored lists, there must be suspicious activities that is being investigated. Whois query on Geektools.com proved that these IP addresses belong to The Computer Network Center Chinese Academy of Sciences as suspected. There were 26 unique sources originating from this domain.

```
01/06-22:17:50.658799 Watchlist 000222 NET-NCFC 159.226.238.63:4024 -> MY.NET.162.91:21
01/06-22:17:50.938510 Watchlist 000222 NET-NCFC 159.226.238.63:4024 -> MY.NET.162.91:21
01/06-22:17:51.222952 Watchlist 000222 NET-NCFC 159.226.238.63:4024 -> MY.NET.162.91:21
01/06-22:17:51.499346 Watchlist 000222 NET-NCFC 159.226.238.63:4024 -> MY.NET.162.91:21
01/06-22:17:52.395668 Watchlist 000222 NET-NCFC 159.226.238.63:4025 -> MY.NET.162.91:4379
01/06-22:17:52.546013 Watchlist 000222 NET-NCFC 159.226.238.63:4024 -> MY.NET.162.91:21
01/06-22:17:52.996810 Watchlist 000222 NET-NCFC 159.226.238.63:4024 -> MY.NET.162.91:21
01/06-22:17:53.293090 Watchlist 000222 NET-NCFC 159.226.238.63:4025 -> MY.NET.162.91:4379
```

Eventhough these activities shown a valid FTP port of 21 connectivity, however, from other activities using non-ephemeral ports i.e. 4379 as shown above looks suspicious and need further investigation. MY.NET.162.91 need to be checked whether this host is really a valid FTP server. If it is, then this host needs to be secured, current patches installed and restrict FTP to only legitimate users. Port 21 can also be used for a number of software including malicious software and hackers toolkits such as Black Construction, Blade Runner, Cattivik FTP Server, CC Invader, Dark FTP, Doly Trojan, Fore, Invisible FTP, Juggernaut 42. Larva, MotIv FTP, Net Administrator, Ramen, Senna Spy FTP server, The Flu, Traitor 21, WebEx and WinCrash. <sup>[14]</sup>

```
OrgName: The Computer Network Center Chinese Academy of Sciences
OrgID: CNCCAS

NetRange: 159.226.0.0 - 159.226.255.255
CIDR: 159.226.0.0/16
NetName: NCFC
NetHandle: NET-159-226-0-0-1
Parent: NET-159-0-0-0-0
NetType: Direct Assignment
NameServer: NS.CNC.AC.CN
NameServer: GINGKO.ICT.AC.CN
Comment: The information for POC handle QH3-ARIN has been reported to
be invalid. ARIN has attempted to obtain updated data, but has
been unsuccessful. To provide current contact information,
please email hostmaster@arin.net.
RegDate: 1992-06-11
Updated: 2002-10-08

TechHandle: QH3-ARIN
```

TechName: Xiqiong, Zhang  
TechPhone: 10 82616000  
TechEmail: [zxq@cstnet.net.cn](mailto:zxq@cstnet.net.cn)

[5]

Correlation: Bruno Marien<sup>[15]</sup> noted this event in his GCIA paper, originating from same sources of 159.226.0.0/16 owned by the Computer Network Center Chinese Academy of sciences and is known to show suspicious activities.

Recommendation: Use SSL for web connectivity that encrypted traffic for web services in MY.NET domain. Restrict upload via FTP and allow download only to outside IP addresses. Use SSH to administer any machines in the network that provide better and stronger authentication, in addition to traffic is encrypted.

#### **Alert#9: Possible trojan server activity**

01/10-02:16:31.373522 Possible trojan server activity MY.NET.91.104:1214 -> 68.18.228.205:27374  
01/10-15:51:55.941624 Possible trojan server activity MY.NET.91.104:1214 -> 217.235.45.31:27374  
01/10-15:51:56.982343 Possible trojan server activity MY.NET.91.104:1214 -> 217.235.45.31:27374  
01/10-15:51:56.982487 Possible trojan server activity MY.NET.91.104:1214 -> 217.235.45.31:27374  
01/10-15:51:56.982603 Possible trojan server activity MY.NET.91.104:1214 -> 217.235.45.31:27374

Port 27374 is known to be used by Trojan such as Bad Blood, Ramen, Seeker, SubSeven, Subseven 2.1.4 DefCon 8, SubSevern Muie and Tftloader. A high probability that host MY.NET.91.104 has been compromised.

Correlation: Jason Lam noted in his paper about traffic on port 27374 which is usually affiliated with the Windows platform Trojan SubSeven.<sup>[16]</sup>

Recommendation: Host MY.NET.91.104 need to be taken offline and cleaned.

#### **Alert#10: Port 55850 tcp - Possible myserver activity - ref. 010313-1**

#### **Alert#26: Port 55850 udp - Possible myserver activity - ref. 010313-1**

These alerts are generated for activities from internal host using port 55850. For host MY.NET.140.9 as shown below, these activities are probably caused by Nimda worm infection.

01/07-20:20:25.947007 High port 65535 udp - possible Red Worm - traffic MY.NET.140.9:65535 -> 128.114.129.62:33479  
01/07-20:20:26.102877 High port 65535 udp - possible Red Worm - traffic MY.NET.140.9:65535 -> 128.114.129.62:33481  
01/07-20:20:26.180657 High port 65535 udp - possible Red Worm - traffic MY.NET.140.9:65535 -> 128.114.129.62:33482  
01/08-11:35:04.113154 Port 55850 udp - Possible myserver activity - ref. 010313-1 MY.NET.140.9:55850 -> 128.143.88.85:33441  
01/08-11:35:04.113719 Port 55850 udp - Possible myserver activity - ref. 010313-1 MY.NET.140.9:55850 -> 128.143.88.85:33442  
01/08-11:35:04.114316 Port 55850 udp - Possible myserver activity - ref. 010313-1 MY.NET.140.9:55850 -> 128.143.88.85:33443

Correlation: Jeff Zahr noted same event in his paper, but he assumed that it is likely due to false positive. <sup>[32]</sup>

Recommendation: Since this is related to Nimda worm infection, machines should be cleaned and patched.

### Alert#11: Queso fingerprint

```
01/06-08:02:37.257517 Queso fingerprint 194.106.96.8:38210 -> MY.NET.70.231:80
01/06-08:03:03.759621 Queso fingerprint 194.106.96.8:38513 -> MY.NET.70.231:80
01/06-08:03:06.964248 Queso fingerprint 194.106.96.8:38544 -> MY.NET.70.231:80
01/06-08:05:04.444609 Queso fingerprint 194.106.96.8:39543 -> MY.NET.70.231:80
01/06-08:06:33.477981 Queso fingerprint 194.106.96.8:40269 -> MY.NET.70.231:80
```

Queso is a utility that query the TCP/IP stack on targeted machines for OS Fingerprinting. Queso also has the capability of sending spoof information. <sup>[3]</sup> This event indicates that a remote user has used the Queso tool to determine the OS of the server. <sup>[4]</sup> Snort log shown above shows clearly that a remote user is trying to determine the OS of a webserver. Whether this is just a scan or a malicious intent for future attack remains to be seen.

Correlation: Akiva Clark noted in his paper about Queso Fingerprinting. He stated that this is an example of older type of fingerprinting scan. <sup>[35]</sup>

Recommendation: Servers that have direct connectivity to the Internet such as Webserver located in DMZ need to be hardened with latest patches installed to reduce vulnerability that can be exploited by an attacker.

### Alert#13: Null scan!

```
01/09-17:15:01.808111 Null scan! 194.109.247.13:0 -> MY.NET.82.248:0
01/09-17:15:06.640268 Probable NMAP fingerprint attempt 194.109.247.13:48156 -> MY.NET.82.248:22779
01/09-17:15:09.668410 Null scan! 194.109.247.13:0 -> MY.NET.82.248:0
01/09-17:15:15.577419 Null scan! 194.109.247.13:6257 -> MY.NET.82.248:6257
01/09-17:15:17.400185 Null scan! 194.109.247.13:0 -> MY.NET.82.248:0
```

Null scan probe indicates that a sequence number of zero has been seen in TCP packet and all control bits has also been set to zero which is abnormal. This should not be seen in a regular TCP packet. Attacker may be scanning target system by sending these specially formatted frames to find out services available or a reconnaissance in finding vulnerabilities to certain attack <sup>[8] [9]</sup> Snort log above showed Null Scan! Probe and determined to be NMAP tool was used.

Correlation: Gustavo Monserrat noted this event as part of fingerprinting probe using nmap tool. <sup>[36]</sup>

Recommendation: Packet matching pattern described above should be block from entering or leaving MY.NET network. Security policy that cover usage of such tool need to be established and users are made aware of such policy.

#### **Alert#14: EXPLOIT x86 NOOP**

The NOOP warning occurs when series of NOP (no operation in assembly language) are found in a stream that could possibly a part of executable files for optimization and alignment reasons; or being part of an exploit code so that the offset doesn't need to be accurate. If it is originating from http port to a very high port, probably someone is downloading an executable. <sup>[7]</sup>

Correlation: Carlin Carpenter <sup>[19]</sup> mentioned this event in her paper, but provides no explanation.

Recommendation: There are many possibilities of false-positive occurring. It is difficult to make recommendation without other information such as payload to make a better analysis. <sup>[37]</sup>

#### **Alert#30: EXPLOIT x86 stealth noop**

This event is triggered by binary data in the packet matched one kind of byte sequence used as filler in buffer overflow attacks. If the attackers suspect that you have a server which is vulnerable to buffer overflow, they will attempt to exploit this vulnerability to gain access. <sup>[37]</sup> However, this binary pattern can also occur in binary data, thus resulting in false positive.

Correlation: Carlin Carpenter <sup>[19]</sup> and Hee So <sup>[24]</sup> mentioned this event in her paper, but provide no explanation.

Recommendation: Further detail analysis that includes payload information is required to determine if this is real attack or false positive.

#### **Alert#22: EXPLOIT x86 setuid 0**

#### **Alert#25: EXPLOIT x86 setgid 0**

This event occurred when shellcode to set the user identity to 0 (root) was detected. If this code is executed successfully, it is possible for the current process to inherit root privileges. However, setuid(2) requires root privileges to be executed in the first place if the current uid is attempting to get a higher privilege level. <sup>[38]</sup> However, many false positive can occur due to binary or text files that matched the content |b017 cd80|. While setuid is for userid 0 (root), setgid attack is targeted for groupid 0 (root).

Correlation: Carlin Carpenter <sup>[19]</sup> and Joe Ellis <sup>[10]</sup> mentioned this event, but provide no explanation.

Recommendation: Further detail analysis that includes payload information is required to determine if this is a real attack or false positive.

#### **Alert#15: Incomplete Packet Fragments Discarded**

Incomplete packet fragment discarded event will triggers when snort received fragment from an 8k or larger packet do not sum more than half the packet when the last fragment is received. <sup>[39]</sup> It is also possible to be caused by bug in snort stream preprocessor. <sup>[40]</sup>

Correlation: Dan Hawrulkiw <sup>[39]</sup> noted this event in his paper that these bad packets are not unique traffic types, and that snort sensors did not received all fragments.

Recommendation: It is important to install stable version of Snort.

#### **Alert#17: SUNRPC highport access!**

#### **Alert#27: Attempted Sun RPC high port access**

01/08-14:17:00.810955	SUNRPC highport access!	64.236.16.137:80 -> MY.NET.55.115:32771
01/08-14:17:00.829727	SUNRPC highport access!	64.236.16.137:80 -> MY.NET.55.115:32771
01/08-14:17:00.829868	SUNRPC highport access!	64.236.16.137:80 -> MY.NET.55.115:32771
01/08-14:17:00.853861	SUNRPC highport access!	64.236.16.137:80 -> MY.NET.55.115:32771
01/08-14:17:00.854108	SUNRPC highport access!	64.236.16.137:80 -> MY.NET.55.115:32771

Correlation: SUNRPC connecting at high port is a concern like shown in the log above. Joe Church mentioned this event in his paper that attacker could have been compromise a target machines with vulnerabilities. In Solaris 2.x Operating Systems, rpcbind listens not only on TCP port 111, and UDP 111, but also on ports greater than 32770. This results in a large number of packet filters, which intend to block access to rpcbind/portmapper, being ineffective. Attacker instead just simply sends packet to a UDP port greater than 32770 on which RPC is listening. <sup>[46]</sup>

#### **Alert#18: IRC evil - running XDCC**

All 8 sources of these alerts are originating from internal machines and the 21 targets are all external.

01/07-13:47:29.057038	IRC evil - running XDCC	MY.NET.88.168:1215 -> 138.121.51.51:6667
01/07-14:47:28.987387	IRC evil - running XDCC	MY.NET.88.168:1215 -> 138.121.51.51:6667
01/07-16:07:29.030032	IRC evil - running XDCC	MY.NET.88.168:4479 -> 132.74.40.10:6667
01/07-16:47:55.699870	IRC evil - running XDCC	MY.NET.88.168:4479 -> 132.74.40.10:6667
01/07-16:48:33.318707	IRC evil - running XDCC	MY.NET.88.168:4479 -> 132.74.40.10:6667
01/07-17:17:28.941959	IRC evil - running XDCC	MY.NET.88.168:4479 -> 132.74.40.10:6667

XDCC is a script that helps automate DCC session during IRC. DCC is a file transfer command in IRC which cannot be used from an anonymous IRC server. However, the other use of DCC is to send private messages. DCC chat is the only way at any time, according to IRC-II Help files that is not logged. XDCC is for the most part was built from toolz, a hacker tool written by hacker known as Yazoo.<sup>[1]</sup>

These traffics showed that 8 internal machines in the University have been compromised.

Correlations: Unfortunately, google search on giac.org could not find any correlation on GIAC papers. However, Chris Cramer fro Duke University explained that IRC channel is the medium using the DCC mechanism. The hacked machines are running a script which automatically logs them into the channel they receive instructions and can up/download files. Users of the IRC channel issue commands to the zombie machines in the form of:

/msg <zombie> xdcc list

/msg <zombie> xdcc send <file number>

The zombies periodically advertise their files for the channel participants.<sup>[23]</sup>

Recommendations: All internal machines such as MY.NET.88.168 and others, need fresh OS reinstall to clean any backdoor already exist in the box. If the University has a policy of restricting IRC usage, then IRC ports i.e. 6665-6669 can be blocked on border firewall.

#### **Alert#19: SMB C access**

These events showed that there were attempts to access the default administrative share C\$. If allowed, the attacker can access to C: filesystem. This event is specific to a vulnerability, but may have been caused by any of several possible exploits.<sup>[2]</sup>

Correlation: Hee So noted this event in his detect. He listed several vulnerabilities in CVE database that might be attempted by attacker such as:

- © CAN-1999-051: A NETBIOS/SMB share password is guessable
- CAN-1999-0519: A NETBIOS/SMB share password is the default, null or missing
- CAN-1999-0520: A system-critical NETBIOS/SMB share has inappropriate access control.
- CVE-2000-0979: Win95/98/ME sending a 1-byte password that matches the first character of the real password, aka the "Share Level Password" vulnerability.<sup>[24]</sup>

Recommendation: All machines in MY.NET domain must be patched against all SMB vulnerabilities. Add additional protection by installing host firewall and host intrusion detection on critical machines.

**Alert#20: TCP SRC and DST outside network**

**Alert#23: ICMP SRC and DST outside network**

01/06-15:00:13.145580	ICMP SRC and DST outside network	192.2.3.11 -> 192.1.3.11
01/06-16:46:11.368419	TCP SRC and DST outside network	192.2.3.11:968 -> 192.1.3.11:514
01/06-16:46:17.369465	TCP SRC and DST outside network	192.2.3.11:968 -> 192.1.3.11:514
01/06-16:46:29.372111	TCP SRC and DST outside network	192.2.3.11:968 -> 192.1.3.11:514
01/09-16:52:47.277302	TCP SRC and DST outside network	192.2.3.11:12865 -> 192.1.3.11:1612
01/09-16:54:02.288463	TCP SRC and DST outside network	192.2.3.11:12865 -> 192.1.3.11:1612
01/09-16:55:17.307077	TCP SRC and DST outside network	192.2.3.11:12865 -> 192.1.3.11:1612

Both alerts above were triggered by sources and destinations that are not in MY.NET network. This could be caused by misconfigured network router or any other equipment. However, looking into the source ports (0, 968, 12865) being used by 192.2.3.11 targeting 192.1.3.11 on ports (0, 514, 1612) respectively, this is probably a scan with spoofed source address.

Correlation: James Hoover noted this in his analysis with similar conclusion. "Because the source addresses are external to the home network and appear to be randomized and the destination address and ports are not randomized, this traffic appears to be generated by a script that is spoofing IP addresses." [33]

Recommendation: To block spoofing, Egress filter can be used to block non-MY.NET addresses from leaving the MY.NET network. If this is caused by routers or any equipments, these misconfigured routers or equipments will need to be fixed.

**Alert#21: NMAP TCP ping!**

Nmap TCP Ping was detected showing that nmap scanning tool has been used the environment to probe hosts. However, since payload information is not considered in the rule that trigger this event, it could possibly false positive.

Nmap TCP Ping works by setting the acknowledge filed to zero and sending a packet with TCP ACK flag set to determine if a network host is active. [45]

**Alert#28: RFB - Possible WinVNC - 010708-1**

VNC is AT&T's remote-control package that allows you to view a machines's desktop running almost any platform you can think of, from almost any other platform you can think of. The server also contains a small HTTP server that can supply a desktop-viewer Java applet to a browser, allowing you to view a desktop remotely from any Java-enabled web client.

The stock VNC uses port 5800 to serve the Java applet, and port 5900 to conduct the RFB (Remote Frame Buffer) dialogue between the client and the server.



These powerful features of VNC and free of charge, makes it popular among Windows System Administrators. However, the same tool can be used by hackers with malicious intents. <sup>[31]</sup>

Below is a some of the detect on this alert:

```
01/06-07:36:37.654342 RFB - Possible WinVNC - 010708-1 MY.NET.113.66:5900 -> 65.185.217.185:55424
01/06-07:37:10.344808 RFB - Possible WinVNC - 010708-1 MY.NET.113.64:5900 -> 65.185.217.185:55427
01/07-07:38:31.569800 RFB - Possible WinVNC - 010708-1 MY.NET.113.63:5900 -> 65.185.217.185:55530
01/07-07:38:37.421466 RFB - Possible WinVNC - 010708-1 MY.NET.113.66:5900 -> 65.185.217.185:55531
01/07-07:38:43.440080 RFB - Possible WinVNC - 010708-1 MY.NET.113.64:5900 -> 65.185.217.185:55533
```

Correlation: Michael McDonnell noted of this event in this paper that internal machines attempted to connect to external IP addresses from port 5900 which is used by VNC. He mentioned the possibilities of students or staffs who are trying to connect to their home PC by using VNC. <sup>[30]</sup>

Recommendation: Block access to VNC on firewall. However, blocking port 5900 will not be effective. Stateful inspection of VNC packet will be more effective.

### **Alert#29: Tiny Fragments - Possible Hostile Activity**

Tiny Fragment alert is triggered when a fragment is smaller than a set threshold value. The minfrag preprocessor threshold configuration checks the size of IP fragments. The concept is that no commercial network equipments known to fragment their traffic to less than 256 bytes, so anything smaller than threshold values are very suspicious. In addition, nmap and fragroute tools fragment to either 8 or 24 bytes fragments. <sup>[17]</sup>

Correlation: Mark Embrich noted this event in his paper, stating that malicious use of fragments would include fragmenting the IP header to get it past a firewall, or a denial of service attacks like Teardrop or Jolt. <sup>[18]</sup>

Recommendation: Hosts need to be upgraded or patched from known vulnerabilities exploitable by Teardrop or Jolt attack. Use of Egress and Ingress filters in firewall to drop packet of these nature, to protect internal network and machines.

### **Alert#32: EXPLOIT NTPDX buffer overflow**

This event indicates that a buffer overflow exploit was attempted against ntpd network time daemon. Some version of ntpd and xntpd are vulnerable to remote root access. Code used in the ntpd network time daemon uses a fixed buffer length 128 in parsing UDP packet. Sending a malformed packet greater length causes the daemon to dump core on a segmentation fault. Because ntpd almost

always runs with root privileges, a carefully constructed exploit can give remote root access. <sup>[21]</sup>

Correlation: Carlin Carpenter <sup>[19]</sup> and Matthew Fiddler <sup>[20]</sup> listed this event in their papers, but there was no discussion. This vulnerability is discussed in CVE-2001-0414.

Recommendation: NTP servers running NTPD or XNTPD need to be checked for this vulnerability. Latest patches that fix this vulnerability need to be installed as soon as possible. If patches are not yet available, NTP service needs to be disabled as a workaround. <sup>[22]</sup>

### 3.5 Alerts Top Ten Analysis:

Alert: Top 10 Source and Target IP addresses

Rank	Total # Source Alerts	Source IP	Total # of Target Alerts	Target IP
1	21336	<a href="#">MY.NET.84.151</a>	28969	<a href="#">MY.NET.84.151</a>
2	17410	<a href="#">MY.NET.88.193</a>	27643	<a href="#">MY.NET.88.193</a>
3	8429	<a href="#">212.179.107.229</a>	17219	<a href="#">192.168.0.253</a>
4	6788	<a href="#">MY.NET.112.204</a>	6786	<a href="#">61.236.39.3</a>
5	5426	<a href="#">212.179.107.228</a>	5970	<a href="#">MY.NET.90.242</a>
6	5102	<a href="#">172.186.226.148</a>	3288	<a href="#">MY.NET.180.39</a>
7	4136	<a href="#">MY.NET.85.74</a>	3078	<a href="#">172.186.226.148</a>
8	3471	<a href="#">MY.NET.111.235</a>	2227	<a href="#">207.200.86.66</a>
9	3465	<a href="#">MY.NET.111.232</a>	2157	<a href="#">207.200.86.97</a>
10	3452	<a href="#">MY.NET.111.230</a>	2134	<a href="#">MY.NET.177.58</a>

The top Nimda infected sources from University's internal network are MY.NET.84.151, MY.NET.88.193, MY.NET.112.204 and MY.NET.85.74.

External IP addresses added in watchlist were also showed up in this top ten sources originating from IP addresses 212.179.107.229 and 212.179.107.228.

### 3.6 SCANS Analysis

Scans: Top 10 Types of Scan Activity

Rank	Type	Hits
1	UDP	2638978
2	SYN *****S*	196401
3	SYN 12*****S* RESERVEDBITS	1197
4	NULL	387
5	INVALIDACK ***A*R*F	150
6	UNKNOWN 1****R** RESERVEDBITS	53

7	UNKNOWN *2*A**S* RESERVEDBITS	41
8	UNKNOWN 1**A*R** RESERVEDBITS	33
9	VECNA ****P***	28
10	NOACK **U**R*F	26
	Total	2803014

Above is a list of scans activities listed by number of occurrences.

#### Scans: Top 10 Source and Target IP addresses

Rank	Total # of Source Alerts	Source IP	Total # of Target Alerts	Target IP
1	943363	<a href="#">130.85.83.146</a>	6790	<a href="#">130.85.70.198</a>
2	721596	<a href="#">130.85.70.176</a>	4206	<a href="#">172.171.155.23</a>
3	157425	<a href="#">130.85.162.90</a>	3517	<a href="#">217.36.24.213</a>
4	154673	<a href="#">130.85.150.213</a>	3477	<a href="#">24.58.246.210</a>
5	113475	<a href="#">130.85.91.252</a>	3427	<a href="#">66.91.16.206</a>
6	95014	<a href="#">130.85.132.20</a>	3192	<a href="#">64.231.88.19</a>
7	60029	<a href="#">130.85.100.20</a>	2928	<a href="#">64.231.90.179</a>
8	57574	<a href="#">130.85.88.238</a>	2482	<a href="#">140.117.181.222</a>
9	42818	<a href="#">130.85.87.50</a>	2299	<a href="#">4.62.59.34</a>
10	39732	<a href="#">130.85.84.178</a>	2268	<a href="#">65.94.247.34</a>

#### Scans: Top 10 Target Ports

Rank	Target Port	Hits	Port Description
1	6257	1663087	Unassigned
2	41170	66384	Unassigned
3	80	48613	HTTP port
4	445	42634	Microsoft-DS
5	137	38413	NETBIOS name service
6	27005	26764	FLEX-LM (1-10)
7	1214	22036	KAZAA Peer to peer
8	443	20817	HTTP over SSL
9	135	12013	EPMAP – DCE endpoint resolution
10	21	10639	FTP port

#### Description of top three scans:

##### Scans#1: UDP Flood Attack

Below are some of the UDP scans analyzed from snort log provided by the GIAC:

```
Jan 6 00:16:20 130.85.83.146:6257 -> 80.117.11.127:6257 UDP
Jan 6 00:16:21 130.85.83.146:6257 -> 67.84.6.98:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 80.11.94.104:6257 UDP
```

```

Jan 6 00:16:20 130.85.83.146:6257 -> 80.117.110.80:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 217.225.46.19:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 24.195.55.152:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 217.225.241.74:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 80.134.60.222:6257 UDP
Jan 6 00:16:21 130.85.83.146:6257 -> 217.127.224.150:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 195.242.71.192:6257 UDP
Jan 6 00:16:20 130.85.83.146:6257 -> 24.196.221.109:6257 UDP
Jan 6 00:16:21 130.85.83.146:6257 -> 80.181.129.210:6257 UDP
Jan 6 00:16:21 130.85.83.146:6257 -> 65.24.235.22:6257 UDP

```

UDP is a connectionless protocol. Therefore no three-way-handshake as with TCP is established to start communication between client and server. If a client sends an UDP packet to a UDP port on a specific system, the system will respond with an ICMP PORT UNREACHABLE reply. Therefore, if no such answer is received, it can be deducted that the UDP port is active. Because of this behavior and many factors that can influence the communication results are usually unreliable. <sup>[41]</sup> Tools such as nmap and udp scan can be used for UDP scans.

However, based on the access to a UDP based server alone the attacker might try some basic attacks which will nevertheless allow him to refine further attacks. NFS server, SNMPXDMI server, RPCBIND/PORTMAPPER server, NIS server and SNMP server are some of the servers that can be exploited. <sup>[42]</sup>

In the case of UDP scans from the log provided, these UDP scans generated 2638978 alerts which is 94% from all scans alerts total.

## Scans#2: SYN \*\*\*\*\*S\* Flood Attack

Below are some of the SYN flood events detected in snort scans logs analyzed:

```

Jan 6 00:10:53 193.253.247.74:4304 -> 130.85.132.30:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4310 -> 130.85.132.31:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4314 -> 130.85.132.32:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4324 -> 130.85.132.37:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4327 -> 130.85.132.39:139 SYN *****S*
Jan 6 00:10:54 193.253.247.74:4268 -> 130.85.132.16:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4339 -> 130.85.132.45:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4349 -> 130.85.132.46:139 SYN *****S*
Jan 6 00:10:53 193.253.247.74:4350 -> 130.85.132.47:139 SYN *****S*
Jan 6 00:10:54 193.253.247.74:4384 -> 130.85.132.59:139 SYN *****S*
Jan 6 00:10:54 193.253.247.74:4385 -> 130.85.132.60:139 SYN *****S*
Jan 6 00:10:54 193.253.247.74:4386 -> 130.85.132.61:139 SYN *****S*
Jan 6 00:10:54 193.253.247.74:4389 -> 130.85.132.62:139 SYN *****S*
Jan 6 00:10:54 193.253.247.74:4329 -> 130.85.132.41:139 SYN *****S*

```

SYN-flood is a simple attack method on computers in a network. This technique makes server so busy that it can't provide service to the legitimate users. Unfortunately SYN-flood is both difficult to detect and avoid, because it is based on the fundamental technique for transfer of data, the TCP protocol. The attack

exploits the way TCP-connections are established between two computers on the network.

This attack utilize the time slot requires for three-way-handshake session to establish TCP connection, leaving the session incomplete by never sending the ACK flag needed. While at the same time the attacker is flooding the victim with a huge number of SYN packets. Most servers has a limited number of simultaneous connections so it is rather easy to lock a server using this technique.<sup>[43]</sup>

### Scans#3: SYN 12\*\*\*\*\*S\* RESERVEDBITS Attack

Below are some of the SYN 12\*\*\*\*\*S\* ReservedBits detected from Scans logs analyzed:

```
Jan 6 00:01:50 65.214.36.150:49339 -> 130.85.99.85:80 SYN 12*****S* RESERVEDBITS
Jan 6 00:08:34 209.47.251.12:42136 -> 130.85.6.40:25 SYN 12*****S* RESERVEDBITS
Jan 6 00:26:32 209.47.251.27:41924 -> 130.85.6.40:25 SYN 12*****S* RESERVEDBITS
Jan 6 00:26:48 65.214.36.150:54591 -> 130.85.165.28:80 SYN 12*****S* RESERVEDBITS
Jan 6 00:29:16 65.214.36.151:48158 -> 130.85.134.11:80 SYN 12*****S* RESERVEDBITS
Jan 6 00:29:19 65.214.36.151:48158 -> 130.85.134.11:80 SYN 12*****S* RESERVEDBITS
Jan 6 00:30:01 65.214.36.151:48158 -> 130.85.134.11:80 SYN 12*****S* RESERVEDBITS
Jan 6 00:38:20 209.47.251.21:52942 -> 130.85.6.40:25 SYN 12*****S* RESERVEDBITS
Jan 6 00:49:36 209.47.251.18:34569 -> 130.85.6.40:25 SYN 12*****S* RESERVEDBITS
Jan 6 00:58:07 209.47.251.27:47219 -> 130.85.6.40:25 SYN 12*****S* RESERVEDBITS
Jan 6 01:16:50 216.174.197.150:50537 -> 130.85.6.40:25 SYN 12*****S* RESERVEDBITS
```

This scans log fits the Queso fingerprint as discussed by Jack Radigan in his GCIA practical paper.<sup>[44]</sup> Another possibility is caused by ECN traffic.

## 3.7 OOS (Out-Of-Specification) Analysis

OOS: Top 10 Source and Target IP addresses

Rank	Total # of Source Alerts	Source IP	Total # of Target Alerts	Target IP
1	762	<a href="#">194.106.96.8</a>	1402	<a href="#">MY.NET.6.40</a>
2	417	<a href="#">MY.NET.70.183</a>	984	<a href="#">MY.NET.1.4</a>
3	318	<a href="#">MY.NET.53.10</a>	767	<a href="#">MY.NET.70.231</a>
4	249	<a href="#">MY.NET.53.84</a>	187	<a href="#">MY.NET.134.11</a>
5	188	<a href="#">65.214.3.151</a>	94	<a href="#">MY.NET.185.48</a>
6	134	<a href="#">209.47.251.30</a>	56	<a href="#">MY.NET.113.4</a>
7	101	<a href="#">66.140.25.156</a>	44	<a href="#">MY.NET.179.78</a>
8	92	<a href="#">81.95.99.139</a>	42	<a href="#">MY.NET.105.42</a>
9	83	<a href="#">65.214.36.150</a>	41	<a href="#">MY.NET.179.77</a>
10	82	<a href="#">209.47.251.18</a>	36	<a href="#">MY.NET.139.230</a>

OOS: Top Source 194.106.96.8 (below)

```
01/06-08:02:22.400600 194.106.96.8:37981 -> MY.NET.70.231:80
01/06-08:03:03.759626 194.106.96.8:38513 -> MY.NET.70.231:80
```

```
01/06-08:03:06.740585 194.106.96.8:38542 -> MY.NET.70.231:80
01/06-08:03:06.964252 194.106.96.8:38544 -> MY.NET.70.231:80
01/06-08:04:42.601229 194.106.96.8:39398 -> MY.NET.70.231:80
01/06-08:04:45.550127 194.106.96.8:39412 -> MY.NET.70.231:80
```

Full Packet Header Information:

```
01/06-10:51:46.550239 194.106.96.8:56068 -> MY.NET.70.231:80
TCP TTL:40 TOS:0x0 ID:3679 IpLen:20 DgmLen:60 DF
12****S* Seq: 0xDC13AF9B Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 693805508 0 NOP WS: 0
```

OOS: Top Target MY.NET.6.40 (below)

```
01/05-00:12:41.118651 209.47.251.20:56018 -> MY.NET.6.40:25
01/05-00:13:40.118167 209.47.251.20:32959 -> MY.NET.6.40:25
01/05-00:22:33.884514 209.47.251.30:53208 -> MY.NET.6.40:25
01/05-00:31:14.349059 209.47.251.20:56924 -> MY.NET.6.40:25
01/05-00:32:15.162532 209.47.251.14:57930 -> MY.NET.6.40:25
01/05-00:33:42.379721 209.47.251.20:41811 -> MY.NET.6.40:25
```

Full Packet Header Information:

```
01/05-00:32:15.162532 209.47.251.14:57930 -> MY.NET.6.40:25
TCP TTL:48 TOS:0x0 ID:10545 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x295592D2 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1380 SackOK TS: 460514015 0 NOP WS: 0
```

To analyze the OOS logs provided by GIAC, I looked at the top source (194.106.96.8) and top target (MY.NET.6.40) as shown above.

The same sources have also triggered Queso fingerprint alert found in Alert logs below:

```
01/06-10:17:05.165474 Queso fingerprint 194.106.96.8:48753 -> MY.NET.70.231:80
01/06-10:18:05.981252 Queso fingerprint 194.106.96.8:50160 -> MY.NET.70.231:80
01/06-10:18:17.592434 Queso fingerprint 194.106.96.8:50497 -> MY.NET.70.231:80
01/06-10:18:37.295917 Queso fingerprint 194.106.96.8:50857 -> MY.NET.70.231:80
01/06-10:18:38.225872 Queso fingerprint 194.106.96.8:50863 -> MY.NET.70.231:80
01/06-10:18:52.274253 Queso fingerprint 194.106.96.8:51187 -> MY.NET.70.231:80
01/06-10:13:55.771003 Queso fingerprint 194.106.96.8:43636 -> MY.NET.70.231:80
01/06-10:13:55.784695 Queso fingerprint 194.106.96.8:43637 -> MY.NET.70.231:80

01/06-00:26:32.561155 Queso fingerprint 209.47.251.27:41924 -> MY.NET.6.40:25
01/06-00:38:20.981326 Queso fingerprint 209.47.251.21:52942 -> MY.NET.6.40:25
01/06-00:49:36.423226 Queso fingerprint 209.47.251.18:34569 -> MY.NET.6.40:25
01/06-00:58:07.658762 Queso fingerprint 209.47.251.27:47219 -> MY.NET.6.40:25
01/06-01:17:55.748159 Queso fingerprint 209.47.251.18:52728 -> MY.NET.6.40:25
01/06-01:11:43.252864 Queso fingerprint 209.47.251.21:35592 -> MY.NET.6.40:25
01/06-02:33:19.849664 Queso fingerprint 209.47.251.21:51905 -> MY.NET.6.40:25
```

It is pretty clear that majority (if not all) of these OOS packets have been the result of Queso fingerprint in MY.NET network.

### 3.8 Defensive recommendations

After thoroughly analyzing and auditing the University's logs, I believed that the University has low security measures in place. Overwhelming number of intrusions that occurred are real attacks and backdoor compromises. A number of machines have been compromised by external machines, either by Nimda worms or backdoors.

Firstly, Nimda worm infections need to be addressed immediately. Especially, four of the top ten sources (MY.NET.84.151, MY.NET.88.193, MY.NET.112.204 and MY.NET.85.74) are originating from Universities' internal machines. Machines infected need to be identified, taken off-line, cleaned from the worm and patched. The Nimda traffic used up a lot of network bandwidth and leaving backdoors that can be used by other exploits.

Secondly, a lot of compromised machines via Trojan, IRC XDCC backdoor, peer to peer connections and huge amount of scanning activities in the environment showed lack of security awareness among community members in the University. This issue can be addressed by having security training, putting security policies in place and enforcement of the policies.

Thirdly, internet protections need to be improved greatly as first line of defense against attacks from Internet. Egress and Ingress filters need to be enabled on firewall to block many malicious activities such as SYN flood attack, UDP scans, backdoor compromises and many others.

### 3.9 A description of analysis process

Hardware used to do this analysis:

- IBM Thinkpad T20 laptop w/512MB RAM running Windows XP
- Dell Inspiron 8200 P4M 1.6GHz w/768MB DDR RAM running Windows XP/RedHat 8.0
- Sun E450 Quad CPU w/2GB RAM 400+ GB RAID running Solaris 8

My initial attempt was to use Demarc PureSecure ([www.demarc.com](http://www.demarc.com)) I had installed in my XP laptop that I used for finding detects in Assignment #2. However, it failed when I tried to run it against Alert, Scan or OOS logs. Later I found out that I can't run snort log through snort again. In my research, I found that **Brandon L. Newport** also had the same issue described in his GCIA paper pg.99 when he tried to run snort against these logs.

Later, I attempted to use SnortSnarf to parse the logs. I used configuration as suggested at <http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/configuration.html#SNORTSNARF-CONFIG> as reference.

First run of SnortSnarf resulted in lot of unknown IP due to "MY.NET" was used

to sanitize the logs provided by GIAC. I found out that **Tod A. Beardsley** had mentioned on the last page of his paper, page 68 that “I learn that SnortSnarf doesn’t like IP addresses like MY.NET.11.8 – it accepts only numeric fields”

Later, I found a hint to address this issue on **Reuben Rubio’s** GCIA practical paper to change MY.NET to something like 10.0 before running snortsnarf. I used SED command to remove MY.NET and change them to 10.0 which is not used in the logs.

What a relief, I was able to run snortsnarf and got the result for Alert logs. However, Scan logs caused out of memory on my laptop. Fortunately, I got access to a Sun E450 box running Sun Solaris 8 with 2GB of physical RAM. Since I combined all Scan logs into one huge file, after more than a day running, memory was exhausted and SnortSnarf failed. Then I ran Scan logs one day at a time. Snortsnarf ran fine but took about a day to finish each one and had to reboot the box every time to free up memory.

Finally, I decided to use Sawmill log parser tool at [www.sawmill.net](http://www.sawmill.net) for analyzing Scans and OOS logs. Thank you to the author for allowing 30 days evaluation to use without purchasing licenses. Then, I was able to run it against 180MB Scans file and finished in only 10 minutes.

#### References:

- 1- Communications of the new order Issue #3.  
<http://216.239.51.100/search?q=cache:Cfi7iegtGPcC:www.attrition.org/~modify/texts/zines/CoTNo/cotno03.txt+IRC+evil+XDCC+6667&hl=en&ie=UTF-8>
- 2- arachnids – IDS339 “NETBIOS-SMB-C\$ACCESS”  
[http://www.whitehats.com/cgi/arachNIDS/Show?\\_id=ids339&view=event](http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids339&view=event)
- 3- Rayford, Joe; GCIA Certification – Practical Assignment; Queso Fingerprint;  
[http://www.giac.org/practical/Joe\\_Rayford\\_GCIA.doc](http://www.giac.org/practical/Joe_Rayford_GCIA.doc)
- 4- arachNIDS; IDS29 “PROBE-QUESO FINGERPRINT ATTEMPT”  
<http://whitehats.com/info/IDS29/>
- 5- Handel, Michael; GCIA Certification – Practical Assignment; Watchlist 000222 NET-NCFC  
[http://www.giac.org/practical/Michael\\_Handel.doc](http://www.giac.org/practical/Michael_Handel.doc)
- 6- Intrusion Detection FAQ: Port 137 Scan;  
[http://www.sans.org/resources/idfaq/port\\_137.php](http://www.sans.org/resources/idfaq/port_137.php)
- 7- Rocha, Luciano; Snort: SHELLCODE X86 NOOP  
<http://www.der-keiler.de/Mailing-Lists/securityfocus/focus-ids/2002-04/0035.html>
- 8- SCAN NULL: <http://www.snort.org/snort-db/sid.html?sid=623>
- 9- IDS4 “PROBE-NUL-SCAN”; <http://www.whitehats.com/info/IDS4>
- 10- Ellis, Joe; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Joe\\_Ellis\\_GCIA.doc](http://www.giac.org/practical/Joe_Ellis_GCIA.doc)



- 11- Stewart, Joe; <http://archives.neohapsis.com/archives/snort/2000-11/0244.html>
- 12- McLain, Vitaly; <http://archives.neohapsis.com/archives/snort/2000-11/0248.html>
- 13- Phrack 55: Perl CGI problem; <http://www.wiretrip.net/rfp/p/doc.asp/i3/d6.htm>
- 14- Ports; [http://www.bekkoame.ne.jp/~s\\_ita/port/port1-99.html](http://www.bekkoame.ne.jp/~s_ita/port/port1-99.html)
- 15- Marien, Bruno; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Bruno\\_Marien\\_GCIA.doc](http://www.giac.org/practical/Bruno_Marien_GCIA.doc)
- 16- Lan, Jason; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Jason\\_Lam\\_GCIA.doc](http://www.giac.org/practical/Jason_Lam_GCIA.doc)
- 17- Roesch, Marty; “[Snort] Tiny Fragment”; email;  
<http://archives.neohapsis.com/archives/snort/2000-05/0103.html>
- 18- Embrich, Mark; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Mark\\_Embrich\\_GCIA.htm](http://www.giac.org/practical/Mark_Embrich_GCIA.htm)
- 19- Carpenter, Carlin; GCIA Certification – Practical Paper;  
[http://www.giac.org/practical/Carlin\\_Carpenter\\_GCIA.doc](http://www.giac.org/practical/Carlin_Carpenter_GCIA.doc)
- 20- Fiddler, Matthew; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Matthew\\_Fiddler\\_GCIA.doc](http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc)
- 21- arachnids; “NTPDX-BUFFER-OVERFLOW”;  
<http://www.whitehats.com/info/IDS492>
- 22- SecurityFocus; “ntpd remote buffer overflow vulnerability”;  
<http://online.securityfocus.com/bid/2540/discussion/>
- 23- Cramer, Chris; “[unisog] hacked university machines”; [unisog@sans.org](mailto:unisog@sans.org);  
<http://staff.washington.edu/dittrich/talks/core02/unisog-xdcc.txt>
- 24- So, Hee; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Hee\\_So\\_GCIA.doc](http://www.giac.org/practical/Hee_So_GCIA.doc)
- 25- F-Secure Virus Descriptions; Nimda;  
<http://www.europe.f-secure.com/v-descs/nimda.shtml>
- 26- Network & Academic Computing Services; “Nimda Worm Information”;  
September 25, 2001; <http://www.nacs.uci.edu/security/nimda-info.html>
- 27- Credeur, Brian; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Brian\\_Credeur\\_GCIA.doc](http://www.giac.org/practical/Brian_Credeur_GCIA.doc)
- 28- Menke, Mark; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Mark\\_Menke\\_GCIA.doc](http://www.giac.org/practical/Mark_Menke_GCIA.doc)
- 29- Wilkinson, Michael; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/michael\\_wilkinson\\_gcia.doc](http://www.giac.org/practical/michael_wilkinson_gcia.doc)
- 30- McDonnell, Michael; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Michael\\_McDonnell\\_GCIA.doc](http://www.giac.org/practical/Michael_McDonnell_GCIA.doc)
- 31- Knapka, Joseph; “VNC server patches for HTTP-via-RFB”;  
<http://home.earthlink.net/~jknappa/vncpatch.html>
- 32- Zahr, Jeff; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Jeff\\_Zahr\\_GCIA.doc](http://www.giac.org/practical/Jeff_Zahr_GCIA.doc)
- 33- Hoover, James; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/James\\_Hoover\\_GCIA.doc](http://www.giac.org/practical/James_Hoover_GCIA.doc)
- 34- Bueno, Pedro; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Pedro\\_Bueno\\_GCIA.doc](http://www.giac.org/practical/Pedro_Bueno_GCIA.doc)
- 35- Clark, Akiva; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Akiva\\_Clark\\_GCIA.doc](http://www.giac.org/practical/Akiva_Clark_GCIA.doc)

- 36- Monserrat; Gustavo; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Gustavo\\_Monserrat\\_GCIA.doc](http://www.giac.org/practical/Gustavo_Monserrat_GCIA.doc)
- 37- “SHELLCODE X86 NOOP”; Snort Signature Database;  
<http://www.snort.org/snort-db/sid.html?sid=648>
- 38- “SHELLCODE x86 setuid 0”; Snort Signature Database;  
<http://www.snort.org/snort-db/sid.html?sid=650>
- 39- Hawrylkiw, Dan; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Dan\\_Hawrylkiw\\_GCIA.doc](http://www.giac.org/practical/Dan_Hawrylkiw_GCIA.doc)
- 40- Roesch, Marty; “[Snort-users] Weird fragmentation plugin error”;  
<http://www.mcabee.org/lists/snort-users/Apr-01/msg00540.html>
- 41- Attack Detail; “UDP Scan”;  
[http://ki.sei.cmu.edu/idar/drill\\_attack.cfm?attack=UDP%20Scan](http://ki.sei.cmu.edu/idar/drill_attack.cfm?attack=UDP%20Scan)
- 42- Vulnerability Detail; “Access to UDP based server”;  
[http://ki.sei.cmu.edu/idar/drill\\_vuln.cfm?vulnerability=Access%20to%20UDP%20based%20server](http://ki.sei.cmu.edu/idar/drill_vuln.cfm?vulnerability=Access%20to%20UDP%20based%20server)
- 43- SYN-Flood; Updated September 19, 1999;  
<http://home.swipnet.se/~w-26153/syn.htm>
- 44- Radigan, Jack; GCIA Certification – Practical Assignment;  
[http://www.giac.org/practical/Jack\\_Radigan\\_GCIA.doc](http://www.giac.org/practical/Jack_Radigan_GCIA.doc)
- 45- Writing Snort Rules; [http://www.defcon.tv/papers/IDS/snort\\_rules.htm](http://www.defcon.tv/papers/IDS/snort_rules.htm)
- 46- Church, Joe; GCIA Certification – Practical Assignment;  
[http://www.network-forensics.net/it%20papers/Joe\\_Church\\_GCIA.pdf](http://www.network-forensics.net/it%20papers/Joe_Church_GCIA.pdf)
- 47- F-Secure; Information on Nimda;  
<http://www.f-secure.com/v-descs/nimda.shtml>
- 48- CERT.org; CERT® Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL;  
<http://www.cert.org/advisories/CA-2001-19.html>