Global Information Assurance Certification Paper

Intrusion Detection In-Depth
GCIA Practical Assignment
Version 3.2
Julien Radoff
October 4, 2002

Table of Contents

Section I - State of IDS

Section II - Network Detects

Section III - Analyze This

SECTION ONE - THE STATE OF IDS

Introduction: Choose your weapons

Sept 11 thinking has heightened the awareness of network security in the Government sector and even the private sector, giving the security technician a boost in attention, resources, funding, employment and yes, pressure. Despite his new status, this job has never been more difficult. He is increasingly overworked and under-funded. [1]

We face the proliferation of fast-ethernet and higher bandwidth, increase in workload, major increase in hacking attempts[1], and a better-armed enemy. We also face an overwhelming number of tools to choose from during an economy of cutbacks. Many tools are free but many are more sophisticated enterprise applications costing in the tens of thousands of dollars. Despite these changes we must implement security within a budget and produce results to management.

We have already created thousands of specialized tools, which do very specific things. We do not have the ability to implement comprehensive rule-based testing and give all of the data meaningful analysis. The reason? Our CPUs and data storage can't handle the load and we can't handle the load from all of the false positives and even real threats.

The more tools we create, the more complicated our lives get. Another response has been to implement multi-tiered database-driven analysis with rule-based automated alerts and updates to our think tanks like the CVE lists. This also has its prices, one of which being narrowing the window of EOI interesting traffic that we can analyze. We have to resort to more of a 'catch some' rather than 'catch all' solution.

For a typical medium sized enterprise company, which is the smallest one will find with any kind of realistic IDS budget, the amount of traffic generated is too much to deal with without some new thinking. We can't do everything, and so we have to pick our battles. What if there was a tool that *could* do everything, though? It might be very expensive but it would make things simpler. It would probably need a lot of hardware but we've already committed ourselves to that anyway and we're still overwhelmed.

There are tools that are very comprehensive suites, like the NetIQ suite, which cover a lot of ground, and maybe adequate for some people, but they offer only a partial solution if you require a large spectrum of solutions. What we need is a

[1] Jackson

super-tool that can do everything.

There are tools now that can do closer to 'everything' than ever before, and possibly someday may evolve into a singular weapon against intruders.

These are referred to as NFAT tools (Network Forensics Analysis Tools). The initial cost is decidedly high, but in the long run, your security system will be simpler and cheaper. They can't do everything, but they can do much more than the name 'forensic tool' implies. They have features in one place that replace several conventional tools. Furthermore, if used creatively, in my opinion, they can eliminate some of the heavy reliance on certain kinds of approaches to IDS. With an NFAT tool and a couple of other basic network security measures (firewall, some simple filtering, and a few key host-based NIDS on the network), I believe most, if not all of the bases can be covered.

There are drawbacks in the approach as well which I will address. These solutions are not perfect either, but all things being equal, wouldn't we rather employ a portable, more centralized approach? As these solutions improve with each version, a single-application IDS system may be created in the not-so-distant future.

**

The proliferation of software

IDS tools have splintered from the 4 major CIDF EADR categories into several major categories as a result and many tools do not fit into one category neatly any more. There is a lot of gray area and cause for confusion. These are just some examples:

1. **Sniffer tools** - TCPdump
2. **Traffic analysis tools** - Snort
3. **Hardware sensors NIDS** - Dragon Sensor
4. **Databases**- TCP quad format tables
5. **Traffic statistical tools** - Perlscript scripts -->Excel, MyNetWatchman
6. **Tools for log analysis** - Dshield, Perl, SnortSnarf
7. **Host-based IDS** - Dragon Squire
8. **Tools for specific detects (low sensors)** - Cybersensor spies on API win32 calls across the network.
9. **Host-based protection** - Filemon by sysinternals monitors all file usage on a server.
10. **Alerting tools** - Dragon Server
11. **Honeypots -** Labrea
12. **Anti-Stegonographic tools -** new feature in NetIQ's suite of IDS and

analysis tools
13. **Analysis tools** - Cybercop
14. **Tools for creating link charts -** Visio

This does not take into account the anti-virus, firewall, hardening, Internet usage and other related security measures, which need to be acquired and run from the same budget. This also does not take into account the task of providing an infrastructure that is secure to run these various tools, which becomes a very real full-time job in and of itself. Also we need people who can manage them.

NFAT tools generate remarkable graphical representations, which are very helpful in plotting graphs and reports as well as network topology diagrams. NFAT tools use their own sensors to display live traffic or record it for later analysis, or they can build these representations from reading log files from several different systems and putting the information together. These diagrams show simulated traffic zipping around the network. Their GUI interfaces drill down on events for custom reporting.

Until the ultimate security tool is created, I suggest that it may be possible to rely less on payload analysis and traffic analysis based on thousands of rules, and other difficult and hardware-intensive work network traffic detects, and focus on a particular aspect of IDS - **unusual traffic patterns**. The NFAT tools all have the power to do standard IDS and capture all data on the network in a fairly efficient manner, but the ability to analyze patterns can reveal the same types of attacks.

In fact, traffic patterns can tell a very clear story in several instances, particularly if represented visually. A picture is worth a thousand words *and takes ever so much less time to read*. NFAT gives us lots of pictures instantly.

Many products advertise that monitor 'unusual traffic patterns', but what I mean by unusual traffic patterns is the particular visual diagram of computers talking to each other (in other words, it creates **link charts**). It does this automatically and can represent traffic in time-lapse animation. These are four categories of unusual traffic patterns:

1. Connections or data transfers between two machines that don't ordinarily speak to each other
2. Speaking to each other during unusual times, or
3. Unusually high amount of connections per given period of time
4. Anything else that defies benchmarks and baselines

NFAT solutions such as Silentrunner, Intellitactics NSM, NetDetector and Netintercept do this. These tools offer the following:

- Sensors
- Rule-based NIDS
- Consolidation of log files,
- Creation of graphs for benchmarks of tallies and profiles
- Charting of traffic with 2-D and sometimes 3-D modeling of links (SilentRunner)
- Creation of custom alerts
- Forensics
- Cataloguing of attacks according to CVE risk level (Intellitactics)

I would argue that if used creatively, these tools could eliminate some of the need for

-Complex payload-based rule sets on NIDS
-Low and slow scan detects (a difficult type of hack to trace)
-Covert channel detect rules
-Password theft payload NIDS rules

This second idea will be explained in detail in the next section.

***

What are we usually worried about?

If we examine the hacker we find two categories of behavior:

Script Kiddies: A person who sends a stock nontargeted script, trojan, virus or worm (such as MS Outlook exploits) without much knowledge of the network or the exploit.

OR this timeline (as presented in the IDS course) :

1 Initial contact or Reconnaissance
2 Focused threat
3 Compromised non-core system
4 Compromised core system

If we elaborate on this timeline we often find familiar patterns :

1. Initial Contact - *scans high volume and low and slow, zone transfer attempts, ICMP probes, random attacks*
2. focused threat (initial break -in) - *spoof or session hijack through tcp flag and ip id and sequence number manipulation resulting in DOS of client machine, buffer overflow, service or daemon hijack to permit some unauthorized power*.
3. compromised system - *after break in, newly gained power or recon information is used to take over a machine to plant a trojan, connect to a server with a spoofed address, run or copy a script from result of buffer overflow, or set up zombie machines for DDOS*
4. *compromised core system - from compromised system, attacker does a DOS on core system or entire network, steals information, steals password or accounts, or plants dangerous script or otherwise damages the system.*

Stage 1 The first stage can't be prevented but it is easily detected. Most scans are pretty similar and a simple IDS tool can detect them, but they can't detect Low and Slow scans. Slow and Low (or Low and Slow) scans only send out a few packets over a long period of time and most scanning IDS rules depend on the rapid, high volume of traffic a scan produces to create an alert. A traffic pattern analysis tool like one of the ones discussed can detect both kinds of scans using baseline graphs. If a scan is allowed, the source can be blocked. Slow and low scans are critical because they indicate a sneakier and smarter attacker, or someone who has already found information through a previous scan, but 'low and slow' scans are very hard to detect. NFAT covers another base here because they are good at these kinds of detects as well as detecting standard scans and IDS filtering.[3] If we can filter the scans in the first place, we've really accomplished a lot.

Script kiddies are less sophisticated hackers who will blindly aim an attack at addresses. They use commonly known attacks and therefore easier to keep track of. Furthermore, unfocused attacks don't find an acceptable target unless they get lucky.

For a script kiddie, this is a one-shot deal. They will stop after the initial contact. They can be easily thwarted with basic IDS tools. They may send a worm, which proceeds automatically through the other 3 stages, however.

---

[3] Anelmo
* How can SilentRunner detect a stolen password file? It checks for all passwords which travel across the network in clear text. How many hackers who use exploited password will bother to encrypt the attack?!
** See appendix II charts

For the more common, high volume stage one threats, even the PIX router for instance, has its own rule base, and can employ rules requiring payload analysis because it is capable of stateful viewing of connections.

Because of CPU overhead, as with all payload and connection analysis, you must pick and choose which to enable, but the PIX router can do a lot.

If this is combined with basic rule-based IDS, many threats are avoided.

…HOWEVER, the sophisticated and dangerous attackers will use this initial contact stage for scanning and recon and succeed by flying under the radar. If they succeed in recon, they move to stage two and become much harder to detect.

Stage 2 ... Stage 2 becomes important, and is key to my thesis. Once recon has been done, the attacker knows what he wants to do and has hundreds of tools to use that are very hard to keep track of without serious CPU power. If he finds out what you're running, he can pick and choose his tools. However, ***at the end of the day, these hundreds of exploits often follow the same few patterns when viewed on a network diagram **.*** These patterns are fairly simple and there aren't too many of them.

For example, a hacker will often try to spoof, hijack or create some unusual channel setting up worms, DDOS attacks, hijacking, and covert channels, or communication on unusual ports. These types of attacks always create strangely routed traffic. At this point we have a stealthier stream of traffic, but strange activity from a topological standpoint. Therefore, the more efficient route is to employ traffic analysis over packet analysis. Enter NFAT.

Examples:

- Worms always display some kind of cascading or multiplying effect.
- DDOS always sets up communication links from an outsider to several inside machines, which then focus on a target machine. Sometimes a master zombie machine is used to trigger the DDOS drones.
- Covert channels always send traffic to the outside through some unusual portal or gateway.
- Spoofing and Hijacking have their own unusual patterns (see Appendix II).

NFAT tools can display uncommon traffic patterns and catch all of these things, displaying them in visual format, without depending on analyzing the packet internally.

SilentRunner can also map ip to Mac addresses, which is a great boost in detecting spoofing because simply spoofing an ip address will not hide the hacker's tracks. Silentrunner can play back events in time-lapse fashion, which makes attacks even more easily diagnosed.

At this point, the odds are that the attacker has still done minimal damage because he has probably only commandeered a workstation or other entity to server as a platform to attack his real target, but this is where he needs to be stopped. He is probably a more sophisticated attacker, he has gathered intelligence, and we may not be aware of where he is attacking us from or even his existence. At this point, an alert and a visual representation of the LAN traffic could be very helpful if it could be produced quickly. NFAT tools, in theory, can do this before he does anymore damage. If the attacker gets passed this stage undetected, there's not a whole lot of hope he can be caught in time.

Stage 3 The attacker needs to break in an establish himself. He's going to compromise the system. Odds are, it will be a workstation. Of course, the hacker may stop here. He may just shut down the workstation and call it a day, but if not, he will move on to use it as a command post and give himself root access, a place to look around, access to services on the machine, create a covert channel to an important source of information, or a prompt to trigger the DDOS or DOS.

A good start for him is to 'browse' for victims on the network from his home base on the workstation if he can. He can run **showmount** or **Net View** and find important resources. He can try to get root passwords.

Host based detections on workstations are not a reasonable solution. It would be too costly. Common workstations should not be so critical to the infrastructure anyway. He may succeed in commandeering the workstation, however, finding a server, or bringing down the network may be his real goal and what we should really be worried about. If this is the case, we proceed to stage four.

Suppose the attacker sets up a covert channel because his goal is to steal information. These can be very hard to detect. SilentRunner can detect coded covert channels and even decode the scrambled data sent on them. In a demonstration, SilentRunner decrypted scrambled photographs sent over covert channels - a new type of hacker covert channel called 'stegging' [4], and encrypted messages embedded in .jpg. It even sorted the photos with pattern

---

[4] NetIQ advertisement

matching into different categories with a very high rate of success, recognizing photographs of different cars (in one test) and different celebrities (in another test) and sorting them. This was useful in one case where a firm insider was selling classified data and shipping it outside the firm through the company network. I saw this in a demonstration myself. This is employed when employees are trying to smuggle out information to spies.

Suppose the attacker is trying to find /etc/passwd files. NFAT tools could baseline these files to alert of any access to them from machines other than those by admins.

Stage 4 Stage four is the stage when internal servers, information, devices or networks get clobbered. These are big problems. For all servers (except externally visible or DMZ servers such as Web, DNS, etc), host based solutions are a last gasp to prevent a compromised network from being brought down when dealing with outside-the-firewall hackers. We have to monitor our DMZ devices very carefully and give them special treatment, allocating much of our IDS resources to them and watch them closely. It would be nice if we could do this with the whole network but we can't on a good-sized network. Whatever detects or logs we may get after the fact are not going to be useful until disaster has already struck.

NFAT tools can record literally all of the traffic on a network. This is the stage where a server is often sent a specific program or bug exploiting the current popular weakness, which hopefully we've already patched against. But this is very late in the game. Hopefully we've caught the hacker in stage one or two. NFAT tools bear the load in the first two or three stages and endeavor to minimize our reliance on defense at stage four. This should really be our goal. If our network team lives in stages three and four, it is probably not sleeping well. Information about the network has been heavily compromised. Host-based solutions are really the best way to go if an attacker gets to stage four, but NFAT tools combined with firewalls reduce our reliance on them. By detecting unusual traffic patterns, NFAT tools can free up the firewalls to do other things and provide a more comprehensive blanket.

Stage 5 (picking up the pieces) Of course, if a network is hacked successfully, this is an unfortunate thing, but it is not a complete failure. It happens sometimes. A complete failure is a case where we can’t correct our mistakes. The important mission here is to present to your boss why it happened, how it happened and find a solution immediately. As the name implies, Network Forensics tools excel at this. They can reconstruct the attack quickly, visually and give you a course of action.

****

Specific Products

SilentRunner only runs on Windows, so you can forget about a customized, line speed, efficiently running, hardware-based system. Also this brings all of the extra involvement of securely maintaining a Windows server, an additional hardship for shops not currently using this kind of system. SilentRunner's recommendations for hardware are also very steep and illustrate how their whopper of a suite is really going to tax i386 hardware.

Netdetector and Netintercept have their own custom hardware, which runs a version of freeBSD in multiprocessor format.

Intellitactics NSM can run on Windows or Solaris and has less intimidating requirements, although it relies on database and Web Server software, which naturally leads to a server farm setup (not necessarily a bad thing). In one part of its white paper it also says it can be supported on LINUX." Reportable fields include source/target, ip/mac/hostname, native event code, category code, business grouping, priority, operating system, system version, CVE, risk levels, and more."[5] NSM also can encrypt its data.[6] It is Java based and very efficient as its white paper shows.

The four systems mentioned here have different alert/alarm capabilities. Being able to customize alerts is essential so pick a tool, which you are sure you can use exactly the way you require. NetDetector and NSM seem to have better alert/alarm capabilities.

Silentrunner has had its own security problems in the past and there seems to be some confusion over just what versions were affected [*]. From some chatter on the internet it seems that many doubt SilentRunner's performance

---

[5] Intellitactics

[6] NSM

[*] See conflicting reports from SecurityFocus quoting CVE and Online Security Systems

capabilities. One should thoroughly test the suite before investing. I have not tested the suite but the good folks at O'Reilly have[7].

If you are going to record all or most of your traffic be sure to plan for a robust storage system. If you have never explored this area of network administration you have some learning curve ahead of you.

*****

Conclusion

In conclusion, a firewall, simple IDS system and host-based sensors combined with a comprehensive tool such as an NFAT tool, might be able to eliminate most hacking activity that is realistically stoppable. NFAT tools can collect events at a very high rate, and although 'forensics tool' implies after-the-fact analysis, many of these tools can relate this information in real time and graphically.

These tools take the burden off of many other kinds of rule-based IDS systems because of their own IDS capabilities and because their graphical capabilities lend themselves well to other pre-emptive types of analysis. These tools can combine log files and analyze them, eliminating hours of perl-scripting, mashing numbers into Excel graphs and patching little buggy freeware tools, maintaining databases of updated attacks and allowing people running the show to focus, possibly eliminating some of the workload.

Added bonuses in traffic analysis and other custom tools from these suites is the ability to detect inside attacks, links, tallies, profiles, base-lining are handled quickly and nicely, various kinds of encoded traffic can be decoded, and alerts are available. Various types of auditing can be done as well. For instance, all unprotected passwords can be neatly captured as they fly across the network and displayed with SilentRunner so we can tighten up the servers that allow them.

SilentRunner's big selling point is that it stops inside threats. SilentRunner's PR department wants you to think of their tool that way. After all, they say, inside threats are more damaging and occur more often. This is true [8][9]. We must keep in mind also that If a threat comes from the outside and proceeds to stage three

---

[7] Simpson

[8] Simpson

[9] Zwicky p 27

(see above), the hacker is now and 'insider' for all practical purposes. He has eluded the IDS and the firewall. Whatever one calls them NFAT tools can be used creatively to cover many security needs.

There are downsides of course. These tools are expensive. However, after we consider the basic investments already made in staffing and firewalls for today's medium sized companies, the cost isn't so overwhelming. These tools are relatively new, and in time maybe will be improved to have enough features to be the primary security tool on an enterprise.

In the meantime, because of their versatility, the sales of these tools are growing. Today they may fall a little short of the super tools of the future which do everything, but as for now, they are simplifying security for many networks by placing many jobs under one umbrella application and this trend is likely to continue.

Sources

- Jackson, Joab. Washington Post Jun 20 2002
- Anelmo, Joseph C. Washington Techway . April 25, 2002
- NetIQ advertisement
- Intellitactics Web page URL:http://www.intellitactics.com/html/nsm_feature_sheet.html#
- NSM technical WhitePaper
- Garfinkel, Simpson. Network Forensics: Tapping the Internet. O'reilly Publishing.
- Zwicky, Elizabeth, et al. Building Internet Firewalls 2nd edition. January, 2000. Oreilly publishing.

APPENDIX I

" Multiple buffer overflow vulnerabilities exist in the collector (cle.exe) component of SilentRunner. The routines that parse passwords for many common protocols such as POP, HTTP, FTP, etc., do not perform necessary bounds checking on user-supplied passwords. It is possible for any user on any network monitored by a SilentRunner collector to craft long strings that will crash the collector and possibly execute arbitrary code on any system running the SilentRunner collector."

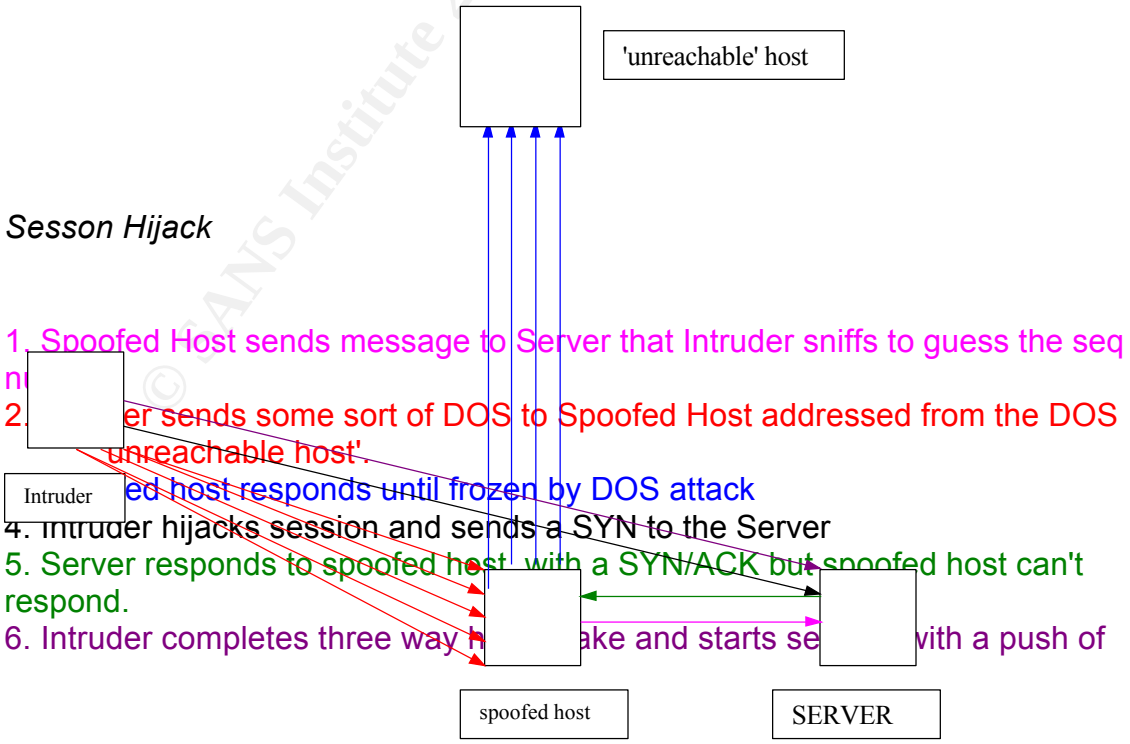
" Affected Versions:
Raytheon SilentRunner 2.0
Raytheon SilentRunner ***2.0.1"***

--Internet Security Systems Web Site
http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise91

Online SecurityFocus Web Site lists CAN-2001-0636 as being vulnerable to SilentRunner 1.6.1, and 2.0, and specifically lists 2.0.1 as NOT vulnerable.

--http://online.securityfocus.com/bid/3150

APPENDIX II



Sesson Hijack

1. Spoofed Host sends message to Server that Intruder sniffs to guess the seq n
2. er sends some sort of DOS to Spoofed Host addressed from the DOS unreachable host'.
ed host responds until frozen by DOS attack
4. Intruder hijacks session and sends a SYN to the Server
5. Server responds to spoofed host with a SYN/ACK but spoofed host can't respond.
6. Intruder completes three way h ake and starts se with a push of

data
uploading its DOS/flood/trojan/destructive script to server

Once a recon has been done and not detected, there are many different ways to create the initial DOS, many addresses to choose from, many different DOS/floods/trojans/destructive scripts, which could be sent to bring down the host, but *ALL OF THEM CREATE THIS VERY STRANGE PATTERN. IF WE JUST LOOK FOR THE PATTERN* and use the ARP capabilities of SilentRunner, it becomes obvious what is going on and hopefully we can be alerted right away. Furthermore a SYN flood is hard to detect by simple packet analysis[11]. A diagram like this is much more enlightening.

This is true for many other types of attacks as well.

[11] SANS intrusion detection course, section four, module 8, page 10

SECTION TWO - THREE DETECTS

DETECT ONE

[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.834488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4527 -> 46.5.180.133:80 TCP TTL:107 TOS:0x0 ID:29762
IpLen:20 DgmLen:99 DF
AP Seq: 0xF5D98495 Ack: 0xB824F82 Win: 0x4470 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69 32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A r..ir..

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

[**] WEB-IIS cmd.exe access [**]
06/10-17:30:31.824488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x70
65.96.221.104:4527 -> 46.5.180.133:80 TCP TTL:240 TOS:0x10 ID:0 IpLen:20
DgmLen:98
AP Seq: 0xF5D984D0 Ack: 0xB825AEA Win: 0x0 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69 32/cmd.exe?/c+di
72 0D 64 69 72 0D r.dir.

=+
=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.834488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4528 -> 46.5.180.134:80 TCP TTL:107 TOS:0x0 ID:29764
IpLen:20 DgmLen:99 DF
AP Seq: 0xF5DA2F8E Ack: 0xB59375C Win: 0x4470 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69 32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A   r..ir..

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

```
[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.834488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4529 -> 46.5.180.135:80 TCP TTL:107 TOS:0x0 ID:29766
IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xF5DAB93B Ack: 0xBE32F8B Win: 0x4470 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69 32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A r..ir..

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

[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.844488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4540 -> 46.5.180.145:80 TCP TTL:107 TOS:0x0 ID:29774
IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xF5E24CD3  Ack: 0x16EA616D  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.844488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4546 -> 46.5.180.151:80 TCP TTL:107 TOS:0x0 ID:29776
IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xF5E62FDA  Ack: 0xB0D08CE  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

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

[**] WEB-IIS cmd.exe access [**]
06/10-17:30:32.324488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x70
65.96.221.104:4546 -> 46.5.180.151:80 TCP TTL:240 TOS:0x10 ID:0 IpLen:20
DgmLen:98
```

```
***AP*** Seq: 0xF5E63015  Ack: 0xF5E63015  Win: 0x0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 64 69 72 0D                                r.dir.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.834488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4548 -> 46.5.180.153:80 TCP TTL:107 TOS:0x0 ID:29768
IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xF5E75A6A  Ack: 0xB7E5DEA  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

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

[**] WEB-IIS cmd.exe access [**]
06/10-17:30:34.864488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x70
65.96.221.104:4548 -> 46.5.180.153:80 TCP TTL:240 TOS:0x10 ID:0 IpLen:20
DgmLen:98
***AP*** Seq: 0xF5E75AA5  Ack: 0xF5E75AA5  Win: 0x0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 64 69 72 0D                                r.dir.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/10-17:30:30.834488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
65.96.221.104:4553 -> 46.5.180.158:80 TCP TTL:107 TOS:0x0 ID:29770
IpLen:20 DgmLen:99 DF
***AP*** Seq: 0xF5EB6122  Ack: 0x8C6752B5  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..
```

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

1.1 Source of Trace:

Raw files from:

http://www.incidents.org/logs/Raw/2002.5.9

1.2 Detect Generated By:

SNORT Win32 1.8 triggered this rule:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access"; flags: A+; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002; rev:2;)

This was the command used: **snort -c snort.conf -d -e -l log -r 2002.05.9**

It is looking for Ack Push flags and CME.EXE attempts on port 80, which would give the attacker a prompt.

I used this command to generate a TCP dump analysis: **windump –r 2002.5.9 -vvv > file.txt**

This is a portion of the Windump output.

17:30:30.834488 h000476b9bf5d.ne.client2.attbi.com.4527 > 46.5.180.133.80: P [bad tcp cksum eae4!] 4124673173:4124673232(59) ack 193089410 win 17520 (DF) (ttl 107, id 29762, len 99, bad cksum a005!)

17:30:31.824488 h000476b9bf5d.ne.client2.attbi.com.4527 > 46.5.180.133.80: P [bad tcp cksum 41f9!] 59:117(58) ack 2921 win 0 [tos 0x10] (ttl 240, id 0, len 98, bad cksum 0!)

17:30:30.834488 h000476b9bf5d.ne.client2.attbi.com.4529 > 46.5.180.135.80: P [bad tcp cksum eae4!] 4124752187:4124752246(59) ack 199438219 win 17520 (DF) (ttl 107, id 29766, len 99, bad cksum 9fff!)

1.3 Probability the Source Address Was Spoofed:

Unlikely. A worm wouldn't work that way. Even if this was a targeted attack, it wouldn't require any kind of spoofing to gain access.

1.4 Description of Attack:

Despite the alert, it is in fact a worm called sadmind/IIS. This attack is basically like Code Red or NIMDA. It's a variation. Although snort saw this as more of a specific attack on weak IIS permissions, snort reports some worms this way.

Question:

"... what do you mean, by 'rapidity with which the attack repeats?' Repeats against the same host? Against different hosts? " - Gary Morris.

Answer:

It attacked machines 133 once and 135 twice within one a five seconds of each other. The timestamp is 17:30 for all of them.

At first glance this looks like a script kiddie trying to get access to a Windows NT or 2000 server, but due to the rapidity with which the attack repeats we can deduce that this attack is behaving like a worm. It tries one packet and then either goes on to the next host, or tries a second packet first. The two different packets can be distinguished by the TTL values and the last command. It is either **c+dir.dir.** or **c+dir..ir..**Several machines were attacked within five seconds. Also if you notice all of the first kind of packet hit all the machines within the first second, while the second kind of packet hit 1 to 4 seconds later.

This is how IIS logs see the attack:

2001-05-06 12:20:19 10.10.10.10 - 10.20.20.20 80 GET /scripts/../../winnt/system32/cmd.exe /c+dir 200 -

What this script tries to do is first access the Windows machine to test for any weak file permissions enabling it to run commands. NIMDA is similar, but has a bigger payload and tries more tests. Eventually, if successful this attack defaces web pages also.

1.5 Attack Mechanism:

It tries to use the IIS scripts directory as a springboard to pop out to the winnt/system32 directory where a lot of powerful tools and sensitive files are

located. From this directory it tries to get a command prompt and execute a directory listing. On the Solaris platform, it creates a similar attack but also opens up a root shell on TCP port 600, among other things.

1.6 Correlations:

http://www.cert.org/advisories/CA-2001-11.html
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp

The following is an excerpt of a description of the attack from the Symantec Press Center:

> Sadmind/IIS is the latest worm designed to attack un-patched versions of Microsoft Internet Information Server (IIS) versions 4.0 and 5.0 Web servers and un-patched versions of Solaris 7 or lower. The Sadmind/IIS worm exploits a buffer overflow vulnerability in the Sadmind program used to remotely control system administration on Solaris operating systems. Once the Solaris system is compromised, the worm searches for Microsoft systems running IIS Web server v. 4.0 or v. 5.0, where it defaces the targeted Web page. The worm further scans to identify other Solaris systems to compromise.
>
> Exploiting server vulnerabilities can result in hackers gaining remote administrator access. This level of access can enable any level of hacker to wreak havoc on systems such as Solaris and IIS, which are commonly used as the internal backbone for an organization's e-mail and Web servers.*

A similar exploit using a buffer overflow is referenced at CVE-2000-0331.

1.7 Evidence of Active Targeting:

Seven different machines were hit with this worm. This is probably not active targeting. With a worm, there is no need for active targeting.

---

* --http://www.symantec.com/press/2001/n010514a.html

1.8 Severity:

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Criticality = 5
Since this is a worm the severity is calculated based on the fact that it is likely the Web Servers will be hit eventually. Also, many servers, which are improperly configured, have IIS running when it is not necessary, leaving them open for an attack.

Lethality = 5
If the attacker gains this access, the intelligence he can gather can be pretty substantial and he will have gained at least limited exe power on the server which he can earmark for future attacks. However, as this is a worm, there is not necessarily an attacker paying attention to the results. Defacing the web pages is extremely serious though. Because of all of these factors I give it a maximum rating.

System Countermeasures = 3
Without access to the detailed configuration of the server, (if Windows or Solaris is even the platform), it is impossible to tell what level of protection exists. Most machines could use some work, so I'll guess a 3.

Network Countermeasures = 4
The good news about SNORT mis-identifying this as cmd exe access attack is that it shows the system is concerned with this kind of access, and any similar attacks are probably being blocked. In other words, it's going to catch many similar worms because it is working at a lower level - unauthorized execute permissions through the web. Worms commonly have been slightly altered and renamed, which enable them to sneak by signatures designed for them, but the mechanics of the attack remain similar. Although the system really should be patched and checked for permissions, because it is hard to distinguish legitimate IIS script access from hacking, for this specific attack, the countermeasure is good. System hardening is strongly recommended for all IIS Webservers. Script and EXE permissions should be avoid wherever possible, as well as directory traversal permissions.

SEVERITY = 3

1.9 Defensive Recommendations:

Be sure that IIS is patched and permissions are gone over with a fine-toothed comb.

Make sure Solaris is patched. Prevent access to root through port 600 or at least keep an eye on it if possible. Keep an eye out for the following directories on Solaris as they are evidence of a hack:

/dev/cub contains logs of compromised machines
/dev/cuc contains tools that the worm uses to operate and propagate

1.10 Multiple Choice Question:

What is a hint that this is an attack that we can deduce just by looking at the header?

A.TTL values alternate between 107 and 240
B. Invalid checksum.
C. The TCP options are set to **AP**
D. B and C only
E. All of the above.

Answer A

DETECT TWO

```
[**] DNS named version attempt [**]
06/10-05:33:34.284488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
203.107.136.133:1169 -> 46.5.29.29:53 UDP TTL:45 TOS:0x0 ID:45068
IpLen:20 DgmLen:58
Len: 38
12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72  .4...........ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03        sion.bind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+=+
[**] DNS named version attempt [**]
06/10-05:53:04.084488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
203.107.136.133:2252 -> 46.5.246.131:53 UDP TTL:45 TOS:0x0 ID:27249
IpLen:20 DgmLen:58
Len: 38
```

12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72 .4...........ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03 sion.bind.....

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

[**] DNS named version attempt [**]
06/10-06:00:29.374488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
203.107.136.133:3463 -> 46.5.215.240:53 UDP TTL:45 TOS:0x0 ID:63142
IpLen:20 DgmLen:58
Len: 38
12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72 .4...........ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03 sion.bind.....

=+
=+=+=+=+=+=+=+
[**] DNS named version attempt [**]
06/10-05:05:22.174488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
203.107.136.133:3561 -> 46.5.22.96:53 UDP TTL:45 TOS:0x0 ID:20309
IpLen:20 DgmLen:58
Len: 38
12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72 .4...........ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03 sion.bind.....

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

[**] DNS named version attempt [**]
06/10-04:23:09.434488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
203.107.136.133:4302 -> 46.5.139.47:53 UDP TTL:45 TOS:0x0 ID:25974
IpLen:20 DgmLen:58
Len: 38
12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72 .4...........ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03 sion.bind.....

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

[**] DNS named version attempt [**]
06/10-05:52:14.404488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x48
203.107.136.133:4315 -> 46.5.216.88:53 UDP TTL:45 TOS:0x0 ID:26333
IpLen:20 DgmLen:58
Len: 38

12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72 .4...........ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03   sion.bind.....

=+

2.1 Source of Trace:

raw files from

http://www.incidents.org/logs/Raw/2002.5.9

2.2 Detect Generated By:

Win32 Snort 1.8 with snort rule:

alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version attempt"; content:"|07|version"; offset:12; content:"|04|bind"; nocase; offset: 12; reference:arachnids,278; classtype:attempted-recon; sid:257; rev:1;)

This was the command used with snort: **snort -c snort.conf -d -e -l log -r 2002.05.9**

Sample from **Windump –r 2002.5.9 -vvv > file.txt**

05:53:04.084488 J9T9O7.2252 > 46.5.246.131.53: [bad udp cksum f9f9!] 4660 [b2&3=0x80] TXT CHAOS)? version.bind. [|domain] (ttl 45, id 27249, len 58, bad cksum b0ce!)

2.3 Probability the Source Address Was Spoofed:

Unlikely. This attack is done for reconnaissance. The attacker needs to see the response to determine course of action.

2.4 Description of the attack:

CVE-1999-0009

UDP packets were fired at a class B address in what looks like very randomized fashion in the early daylight hours with a crafted payload querying for the DNS BIND version at port 53. A response would tell the attacker two things: Whether un-trusted IP addresses are allowed to query DNS servers through UDP, and what the version of BIND is so he can devise an exploit for that version. Earlier versions of BIND have severe vulnerabilities.

Question:

" Assuming that there are several phases in an attack scenario, finding a live machine, finding a live machine with a specific exposure, and lastly, exploiting the exposure. Where in the attack process is the attacker? Would the attack be next in this case? "

--Oliver Viitamaki, Intrusions mailing list.

I like this question because it is in keeping with my theme of the Essay.

This is what I referred to in the State of Intrusion Detection section as step one of an attack process engaged in by the serious hacker. This could be a class B scan of completely randomized addresses over an hour and a half time, only a few of which were in the network. This would explain the randomness and lack of detects, but let's take this opportunity to be paranoid. We *could* explain it this way: Let's say that these are our eight DNS servers and the hacker is targeting them on purpose. He's already done enough recon to locate them all, which means he knows too much about our network already. He's taking an hour and a half to do this because he's analyzing responses from each DNS server one at a time. Fortunately our IDS system is blocking the responses for these eight servers.

If a response slipped by he could wait to see what version of BIND it is running and see also how it reacts to un-trusted UDP packets. If he finds an old version of BIND he could attack them with any number of exploits. This would be stage 2.

If he finds loose DNS permissions, he could perform a zone transfer in order to poison the DNS database, or he could simply view it in order to do further recon on the network. If it is a Windows network, he can discover all of the devices on the network and their names and IP addresses, and look for a specific target such as a database server, which would normally be disguised from the outside. He could use the compromised DNS machine from stage 2 to set up some sort of attack on his ultimate goal, gaining root access to the database server.

The NFAT tool might detect transfer of traffic between the DNS server and database server based on it being an unusual traffic pattern.

2.5 Attack Mechanism:

The Query is the mechanism. It is for reconnaissance purposes.

2.6 Correlations:

http://www.whitehats.com/info/IDS278

CVE-1999-0009

2.7 Evidence of Active Targeting:

This could be active targeting. If the entire class B address was being scanned, some kind of pattern and more addresses would be expected in the scan. This is either a slow and low scan which if heavily disguised, or active targeting.

2.8 Severity:

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

Criticality = 5
DNS machines are among the most important machines on a network.

Lethality = 3
After successful recon, the attacker still has a few hurdles to go through before doing any damage.

System Countermeasures = 3
Most of the newer implementations of BIND have less vulnerabilities, but many machines are still vulnerable to flooding and Windows machines are commonly mis-configured to trust too many machines for zone transfers. A DNS machine should be as guarded as possible, allowing out the minimum amount of information about itself. The DNS machine is unknown in this case so I'm using 3 as an average.

Network Countermeasures = 3

Regardless of what version of BIND is running or if the DNS servers will respond to un-trusted UDP packets, we still want our DNS servers to be shielded from any kind of DOS attempt. Firewall IDS rules are the best way to prevent this and this measure seems to be a good first step to preventing initial recon, but we don't know what will happen if DNS is attacked.

SEVERITY = 2

2.9 Defensive Recommendations:

Make sure DNS servers, especially older UNIX DNS and WinNT 4 machines, are patched if needed and only authenticate to trusted servers over secure lines. There are many versions of BIND, which have particular vulnerabilities. Keep up to date with the latest versions, patches and vulnerabilities.

Query:

"Please describe what you mean by "secure lines" in this context." -- Oliver Viitamaki, Intrusions mailing list.

DNS servers do not have to transfer unsecured data. The traffic can be encrypted or tunneled between connections for zone transfers and updates. In Windows 2000, there is a setting to allow transfers only to authenticated machines, which should be checked.

When DNS servers are queries for zone transfers, they must transmit the information via TCP because UDP won't carry the amount of information a typical zone transfer contains. This enables us to let TCP source checking to kick in. Also it enables us to filter outgoing TCP packets from port 53 from our DNS machine, unless they are headed to our own downstream DNS servers. TCP is sort of a 'secure channel' if you compare it to UDP.

Many firewalls also restrict DNS queries by type. You can block these. You can also create false records. My HINFO record is telling everyone that I'm running MacOSX.

10. Multiple Choice question.
What does TXT.CHAOS refer to?

A. 10 00 03 in the packet
B. Query is to be replied to with CHAOS encryption
C. TXT.CHAOS is a brand of BIND
D. Version of BIND before 4.0

answer: A

DETECT THREE

```
[**] WEB-IIS _vti_inf access [**]
06/10-03:30:55.864488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x13F
194.78.64.131:17987 -> 46.5.180.133:80 TCP TTL:47 TOS:0x0 ID:33381
IpLen:20 DgmLen:305 DF
***AP*** Seq: 0x1FFED07B  Ack: 0xA8D2BD56  Win: 0x1C84  TcpLen: 20
47 45 54 20 2F 5F 76 74 69 5F 69 6E 66 2E 68 74  GET /_vti_inf.ht
6D 6C 20 48 54 54 50 2F 31 2E 31 0D 0A 44 61 74  ml HTTP/1.1..Dat
65 3A 20 4D 6F 6E 2C 20 31 30 20 4A 75 6E 20 32  e: Mon, 10 Jun 2
30 30 32 20 30 38 3A 32 36 3A 35 36 20 47 4D 54  002 08:26:56 GMT
0D 0A 4D 49 4D 45 2D 56 65 72 73 69 6F 6E 3A 20  ..MIME-Version:
31 2E 30 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A  1.0..Accept: */*
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F  ..User-Agent: Mo
7A 69 6C 6C 61 2F 32 2E 30 20 28 63 6F 6D 70 61  zilla/2.0 (compa
74 69 62 6C 65 3B 20 4D 53 20 46 72 6F 6E 74 50  tible; MS FrontP
61 67 65 20 34 2E 30 29 0D 0A 48 6F 73 74 3A 20  age 4.0)..Host:
77 77 77 2E 58 58 58 58 2E 63 6F 6D 0D 0A 41 63  www.XXXX.com..Ac
63 65 70 74 3A 20 61 75 74 68 2F 73 69 63 69 6C  cept: auth/sicil
79 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74  y..Content-Lengt
68 3A 20 30 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E  h: 0..Connection
3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 43 61  : Keep-Alive..Ca
63 68 65 2D 43 6F 6E 74 72 6F 6C 3A 20 6E 6F 2D  che-Control: no-
63 61 63 68 65 0D 0A 0D 0A                       cache....

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

3.1 Source of Trace:

http://www.incidents.org/logs/Raw/2002.5.9

3.2 Detect Generated By :

Win32 Snort 1.8 activated this rule:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"WEB-IIS _vti_inf access";flow:to_server,established;

uricontent:"_vti_inf.html"; nocase; classtype:web-application-activity; sid:990; rev:5;)

Snort analyzed the date with this command:

snort -c snort.conf -d -e -l log -r 2002.05.9

Windump –r 2002.5.9 -vvv > file.txt created the following output:

03:30:55.864488 194.78.64.131.17987 > 46.5.180.133.80: P [bad tcp cksum 1a30!] 536793211:536793476(265) ack 2832383318 win 7300 (DF) (ttl 47, id 33381, len 305, bad cksum e90b!)

3.3 Probability the Source Address was Spoofed:

There is no reason for it to be. This service (FPSE) has been set up specifically for remote access, so, it is not designed to trust or not trust specific machines, provided that it is being served in the same virtual directory as the main web site, which is almost certainly the case. Generally, recon is easier if not done by spoofed addresses, so that the attacker can easily listen for responses.

3.4 Description of the Attack:

The hacker is trying to find out if this machine has FrontPage Extensions installed, so that he can implement any number of possible attacks. There are several worms and buffer overflow exploits which FPSE is vulnerable to.

3.5 Attack Mechanism:

The attacker looked for active web servers and sent a request via port 80 to confirm that the_vti_inf.html file was installed.

This is the content of the sample vti inf.htm file in c:\inetpub\wwwroot\

```
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title> FrontPage Configuration Information </title>
</head>

<body>
<!-- _vti_inf.html version 0.100>
<!--
```

```
        This file contains important information used by the FrontPage client
        (the FrontPage Explorer and FrontPage Editor) to communicate with the
        FrontPage server extensions installed on this web server.

        The values below are automatically set by FrontPage at installation.  Normally, you do
not need to modify these values, but in case
        you do, the parameters are as follows:

        'FPShtmlScriptUrl', 'FPAuthorScriptUrl', and 'FPAdminScriptUrl' specify
        the relative urls for the scripts that FrontPage uses for remote
        authoring.  These values should not be changed.

        'FPVersion' identifies the version of the FrontPage Server Extensions
        installed, and should not be changed.
--><!-- FrontPage Configuration Information
   FPVersion="4.0.0.0"
   FPShtmlScriptUrl="_vti_bin/shtml.exe"
   FPAuthorScriptUrl="_vti_bin/_vti_aut/author.exe"
   FPAdminScriptUrl="_vti_bin/_vti_adm/admin.exe"
-->
<p><!--webbot bot="PurpleText"
preview="This page is placed into the root directory of your FrontPage web when FrontPage is
installed.  It contains information used by the FrontPage client to communicate with the
FrontPage server extensions installed on this web server.  You should not delete this file."
--></p>

<h1>FrontPage Configuration Information </h1>

<p>In the HTML comments, this page contains configuration
information that the FrontPage Explorer and FrontPage Editor need to
communicate with the FrontPage server extensions installed on
this web server. Do not delete this page.</p>
</body>
</html>
```

This html page references the web page remote authoring and admin applications. If an attacker can hack into these applications, or if he can change directories to the vti bin directory, he may be able to get a prompt or exe access.

The FrontPage application processes web forms. If the FrontPage application gets strange requests through this portal, it can become confused and shut down, creating a denial of service.

The NIMDA worm also gives this a whack, trying to create a buffer overflow and execute a command prompt and directory traversal.

GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir

HTTP/1.0

3.6 Correlations:

There are two CVE's currently registered for FPSE CVE-2001-0096 , CAN-2002-0692. The first one is for the web form submission exploit. The second will be described below.

Here is the CERT advisory

http://www.cert.org/advisories/CA-2001-26.html

This is test source code which demonstrates a DOS against FrontPage

/data/vulnerabilities/exploits/fpse2000ex.c

This article describes a SmartHTML interpreter DOS vulnerability.

> Microsoft advises patch on Internet servers to stave off malicious code.
>
> A flaw in the SmartHTML Interpreter contained in Microsoft FrontPage Server Extensions (FPSE) could enable an attacker to run malicious code or to instigate a denial of service attack, Microsoft said in a security advisory late Wednesday. The flaw affects FrontPage Server Extensions 2000 and FrontPage Server Extensions 2002. Previous versions of this software are no longer supported, and may or may not be affected by these vulnerabilities, Microsoft said in the advisory. Microsoft categorized the security hole as critical on Internet servers, moderate for intranet servers and no threat to client systems.[*]

Also see Microsoft article Q280322

3.7 Evidence of Active Targeting:

This web server is being actively targeted. In addition to the fact that only one other even similar attack was found in the weekly logs, (the rpc attack listed below) we have to keep in mind that this attack is no fun unless it returns information about the server which can enable the attacker to run NIMDA or a buffer overflow or some other attack on it.

3.8 Severity:

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

Criticality = 5

---

[*] Legard, David

Web servers are generally important. Although many machines have IIS and FrontPage installed on them that are not Web servers, either by mistake or for other purposes, this Web server was accessible from the outside. It may be very visible and important. Microsoft categorizes the threat to un-patched servers 'critical'. There is a very good chance our hacker is going to try to follow up with an attack since he has actively targeted this system.

Lethality = 4
The amount of possibilities available to the attacker is pretty high, if this scan produces a positive hit. Website defacing Worms, DOS attacks, root access, are all possible results.

System Countermeasures = 2
On average, most IIS servers and even many NT or Win2k servers have this vulnerability exposed. Guessing, I would say the average network is a 2.

Network Countermeasures = 4
It is good that there is a catch-all rule here blocking any kind of access to the FPSE inf. FrontPage is such an issue that an entire category of attacks is dedicated just to it. Another rule for FPSE was triggered, as shown in the logs:

[**] WEB-FRONTPAGE _vti_rpc access [**]

SEVERITY = 3

3.9 Defensive Recommendations:

If front page authoring is needed, it really needs to be locked down against a variety of attacks. Front Page extensions only need to be installed to create this vulnerability. If the network admin is not using Front Page, it should remove these shared files. Check all machines with IIS. Remember, FPSE doesn't have to be running to be affected.

If one needs Front Page, patches are available. One came with Win2k SP2, however, Microsoft has just released another one. There is a catch of course. You have to upgrade to FPSE 1.3 and if you are running Win2k, you must upgrade to SP4. Once you have done that you can download the patch. The file Fp4awel.dll is upgraded for NT and Win2k. For XP, a whole slew of files is upgraded.

- Microsoft Windows NT 4.0: http://download.microsoft.com/download/fp2000fd2000/Patch/1/W9XNT4Me/EN-US/fpse0901.exe
- Microsoft Windows 2000:

http://download.microsoft.com/download/win2000pro/Patch/Q324096/NT5/EN-US/Q324096_W2K_SP4_X86_EN.exe

- Microsoft Windows XP: http://download.microsoft.com/download/whistler/Patch/Q324096/WXP/EN-US/Q324096_WXP_SP1_x86_ENU.exe

Also, be sure that exe and directory browsing permissions are restricted to the admins who need them if remote admin is necessary.

The extensions can be found in the Microsoft shared files folder and under Inetpub\wwwroot_vti_bin. Manual removal of the executables or folders is one option, but you can also uninstall FPSE. Securing the file permissions is another option.

Here is another attack on the network, which is very similar and is an example of why it would behoove the admin to lock down or eliminate FrontPage completely.

```
[**] WEB-FRONTPAGE _vti_rpc access [**]
06/10-03:30:56.784488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x181
194.78.64.131:17988 -> 46.5.180.133:80 TCP TTL:47 TOS:0x0 ID:33417
IpLen:20 DgmLen:371 DF
***AP*** Seq: 0x8F579EB4  Ack: 0xA8C23575  Win: 0x1C84  TcpLen: 20
50 4F 53 54 20 2F 5F 76 74 69 5F 62 69 6E 2F 73  POST /_vti_bin/s
68 74 6D 6C 2E 65 78 65 2F 5F 76 74 69 5F 72 70  html.exe/_vti_rp
63 20 48 54 54 50 2F 31 2E 31 0D 0A 44 61 74 65  c HTTP/1.1..Date
3A 20 4D 6F 6E 2C 20 31 30 20 4A 75 6E 20 32 30  : Mon, 10 Jun 20
30 32 20 30 38 3A 32 36 3A 35 37 20 47 4D 54 0D  02 08:26:57 GMT.
0A 4D 49 4D 45 2D 56 65 72 73 69 6F 6E 3A 20 31  .MIME-Version: 1
2E 30 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20  .0..User-Agent:
4D 53 46 72 6F 6E 74 50 61 67 65 2F 34 2E 30 0D  MSFrontPage/4.0.
0A 48 6F 73 74 3A 20 77 77 77 2E 58 58 58 58 2E  .Host: www.XXXX.
63 6F 6D 0D 0A 41 63 63 65 70 74 3A 20 61 75 74  com..Accept: aut
68 2F 73 69 63 69 6C 79 0D 0A 43 6F 6E 74 65 6E  h/sicily..Conten
74 2D 4C 65 6E 67 74 68 3A 20 34 31 0D 0A 43 6F  t-Length: 41..Co
6E 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C  ntent-Type: appl
69 63 61 74 69 6F 6E 2F 78 2D 77 77 77 2D 66 6F  ication/x-www-fo
72 6D 2D 75 72 6C 65 6E 63 6F 64 65 64 0D 0A 43  rm-urlencoded..C
6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D  onnection: Keep-
41 6C 69 76 65 0D 0A 43 61 63 68 65 2D 43 6F 6E  Alive..Cache-Con
74 72 6F 6C 3A 20 6E 6F 2D 63 61 63 68 65 0D 0A  trol: no-cache..
0D 0A 6D 65 74 68 6F 64 3D 73 65 72 76 65 72 2B  ..method=server+
76 65 72 73 69 6F 6E 25 33 61 34 25 32 65 30 25  version%3a4%2e0%
32 65 32 25 32 65 34 37 31 35 0A                 2e2%2e4715.
```

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

3.10 Multiple Choice Question:

Which one of these attacks does NOT exploit FPSE in some way?

A. NIMDA
B. RED WORM
C. VTI RPC Access
D. SubSeven
E. They all exploit FrontPage

Answer D

SECTION THREE - ANALYZE THIS

Executive Summary

This network has been attacked by a few different categories of attacks. Many of these categories can be eliminated from worry completely if comprehensive steps are taken to lock down these commonly exploited services. To give a big picture view of this network assessment, my recommendations will be along these lines.

Not only is this a comprehensive assessment, but also a method for maintaining the network on a weekly basis by spending a reasonable amount of time analyzing data.

I purposely designed simple *efficient* code and on the fly UNIX commands. I do this by using short modular, batched Perl scripts, which contained no nested loops and are linked together with batch files. I also used UNIX grep, cut, and sort. If I was the full-time security admin at GIAC U, I would want to produce a report weekly based on these logs. Time would be critical. Lots of interesting comparisons and charts are neat to look at but we have to remember what our primary goal is here – **Produce data which doesn't take long periods of time to create but will still help us keep out the bad guys.**

I chose the following files for my assignment:

(Logs from October 11 – 15, 2002)

alert.021002	scans.021002	OOS_Report_2002_10_2
alert.021003	scans.021003	OOS_Report_2002_10_3
alert.021005	scans.021005	OOS_Report_2002_10_5
alert.021006	scans.021006	OOS_Report_2002_10_6
alert.021009	scans.021009	OOS_Report_2002_10_9

THE FOLLOWING IS AN ALERTS CHART SORTED BY NUMBER OF OCCURANCES:

Reference number / ALERT	OCCURANCES	TARGET
1 Portscan (Connection limit exceeded)	407,352	general recon
spp_portscan (One-to-One Connections across to many hosts)	199,407	general recon
2 ICMP SRC and DST outside network	72,263	
3 spp_http_decode	55,894	webserver
4 Watchlist 000220 IL-ISDNNET-990517	45,096	
5 SMB Name Wildcard	20,578	fileshares
6 Possible trojan server activity	3,488	multiple targets through covert channel
7 FTP DoS ftpd globbing	2,368	ftp server
8 High port 65535 udp - possible Red Worm - traffic	1,403	multiple targets through covert channel
9 IDS552/web-iis_IIS ISAPI Overflow ida nosize	1,255	webserver
10 Tiny Fragments - Possible Hostile Activity	1,028	
11 Queso fingerprint	842	OS recon
12 IRC evil - running XDCC	560	
10 Incomplete Packet Fragments Discarded	442	
4 Watchlist 000222 NET-NCFC	433	
13 External RPC call	388	RPC
14 EXPLOIT x86 NOOP	340	OS
8 High port 65535 tcp - possible Red Worm - traffic	278	multiple targets through covert channel
13 SUNRPC highport access!	262	RPC
5 Null scan!	162	fileshares
15 Port 55850 tcp - Possible myserver activity - ref. 010313-1	127	covert channel

16 TFTP - Internal UDP connection to external tftp server	109	
5 SMB C access	91	fileshares
15 Port 55850 udp - Possible myserver activity - ref. 010313-1	88	Network - DDOS covert channel
1 NMAP TCP ping!	68	general recon
14 EXPLOIT x86 setuid 0	44	OS
17 Bugbear@MM virus in SMTP	31	IE
14 EXPLOIT x86 setgid 0	29	OS
18 connect to 515 from outside	16	LPR
14 EXPLOIT x86 stealth noop	13	OS
2 TCP SRC and DST outside network	12	
19 RFB - Possible WinVNC - 010708-1	10	Remote Access Program
20 EXPLOIT NTPDX buffer overflow	6	OS
16 External FTP to HelpDesk MY.NET.70.50	6	ftp server
16 HelpDesk MY.NET.70.49 to External FTP	5	ftp server
21 Russia Dynamo - SANS Flash 28-jul-00	5	Covert channel?
22 DDOS shaft client to handler (port 20432)	4	Network - DDOS covert channel
16 TFTP - Internal TCP connection to external tftp server	3	ftp server
16 TFTP - External UDP connection to internal tftp server	3	ftp server
16 HelpDesk MY.NET.70.50 to External FTP	2	ftp server
16 Attempted Sun RPC high port access	2	RPC
11 Probable NMAP fingerprint attempt	2	OS
16 TFTP - External TCP connection to internal tftp server	2	ftp server
16 HelpDesk MY.NET.83.197 to External FTP	2	ftp server
16 External FTP to HelpDesk MY.NET.70.49	2	ftp server

THIS CHART SORTS THE ALERTS BY THEIR TARGETS:

ALERT	OCCURANCES	TARGET
15 Port 55850 tcp - Possible myserver activity - ref. 010313-1	127	covert channel
21 Russia Dynamo - SANS Flash 28-jul-00	5	Covert channel?
5 SMB C access	91	fileshares
5 Null scan!	162	fileshares
5 SMB Name Wildcard	20,578	fileshares
16 HelpDesk MY.NET.70.50 to External FTP	2	ftp server
16 TFTP - External TCP connection to internal tftp server	2	ftp server
16 HelpDesk MY.NET.83.197 to External FTP	2	ftp server
16 External FTP to HelpDesk MY.NET.70.49	2	ftp server

16 TFTP - Internal TCP connection to external tftp server	3	ftp server
16 TFTP - External UDP connection to internal tftp server	3	ftp server
16 HelpDesk MY.NET.70.49 to External FTP	5	ftp server
16 External FTP to HelpDesk MY.NET.70.50	6	ftp server
7 FTP DoS ftpd globbing	2,368	ftp server
17 Bugbear@MM virus in SMTP	31	IE
18 connect to 515 from outside	16	LPR
8 High port 65535 tcp - possible Red Worm - traffic	278	multiple targets through covert channel
8 High port 65535 udp - possible Red Worm - traffic	1,403	multiple targets through covert channel
6 Possible trojan server activity	3,488	multiple targets through covert channel
22 DDOS shaft client to handler (port 20432)	4	Network - DDOS covert channel
15 Port 55850 udp - Possible myserver activity - ref. 010313-1	88	Network - DDOS covert channel
11 Probable NMAP fingerprint attempt	2	OS
20 EXPLOIT NTPDX buffer overflow	6	OS
14 EXPLOIT x86 stealth noop	13	OS
14 EXPLOIT x86 setgid 0	29	OS
14 EXPLOIT x86 setuid 0	44	OS
14 EXPLOIT x86 NOOP	340	OS
11 NMAP TCP ping!	68	RECON, GENERAL
1 spp_portscan (One-to-One Connections across to many hosts)	199,407	RECON, GENERAL
1 Portscan (Connection limit exceeded)	407,352	RECON, GENERAL
11 Queso fingerprint	842	RECON, OS
16 Attempted Sun RPC high port access	2	RPC
13 SUNRPC highport access!	262	RPC
13 External RPC call	388	RPC
19 RFB - Possible WinVNC - 010708-1	10	Software, remote access program
9 IDS552/web-iis_IIS ISAPI Overflow ida nosize	1,255	webserver
3 spp_http_decode	55,894	webserver

1

SCANS

High volume recon attempts to find available ports, subnets and machines. Easily found in the scans log and alert log because of their massive amounts of traffic.

The TCP Ping is a scan that attempts to find active hosts.

2

TCP source and dest outside of network, UDP source and dest outside of network

This is either misconfigured router activity or spoofing. There is no reason that people should be connecting from the outside to another outside address through your network.

3

HTTP DECODE,

This is an exploit which works by bypassing a checking system in IIS for the universal code, "Unicode" language and character interpreter. It results in directory traversal and can result in executable access of the cmd.exe command line. It is also the goal of two of the attacks described in my detects. See section two.

Additional references:

http://xforce.iss.net/alerts/advise68.php
http://www.securityfocus.com/archive/1/140091
CAN-2000-0884.

4

Watchlist 000222 NET-NCFC

A watchlist is nothing more that a list of troublemakers. Any activity from them is logged, and in fact a lot of their activity, at least in logs I've viewed, has indeed been suspicious or at least weird.

These came from the Computer Network Center Chinese Academy of Sciences.

10/03-00:07:00.090643 [**] Watchlist 000222 NET-NCFC [**]
159.226.66.158:1167 -> MY.NET.145.18:80

10/03-00:07:00.091490 [**] Watchlist 000222 NET-NCFC [**]
159.226.66.158:1167 -> MY.NET.145.18:80

5

Netbios and SMB, SMB C exploit

These services share a type of exploit called a null user exploit whereby fileshares are hacked by sending blank commands to them in the user field where user and password are required. Netbios is available on port 137 and 139. These can all be blocked at the router. Typically, file sharing is not done through remote access. Remote access is implemented through other services.

The C exploit attempts to access the administrator's share on Windows -- C$ -- which gives access to the entire C drive with admin privileges. These attacks need defending from the inside too. This is where NFAT tools and strong passwords come into play. There is no realistic way to turn off these shares on Windows machines.

Additional references:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0288

6

Trojan server

Look for activity on port of 27374 Many Trojans work on it. Sub Seven and Back Orifice are two. They establish covert channels and report back to HQ to alert the master machine that it is ready to be infected. ICQ and email can be used for this. If you find ICQ connections en masse, you need to take immediate action.

Additional references:

http://www.sans.org/newlook/resources/IDFAQ/subseven.htm
http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html

7

DOS FTP FILEGLOB

Additional references:

http://www.digitaltrust.it/arachnids/IDS487/event.html

This is not DOS as in the operating system, it's Denial of Service. This attack tries to bring ftp servers to a hault by sending a huge request for file listings. This was found in a packet

ls */../*/../*/../*/../*/../*/../*/../*/../*/../*/../*

Certain Unix machines vulnerable. They are too numerous to list and some platforms are currently in the process of investigation. See Cert reference listing all of them and current information.

http://www.cert.org/advisories/CA-2001-07.html

8

RED WORM

This is a tricky worm because although it usually operates on port 65535, its code allows it to be altered easily. This worm scans hosts for the LPRng, BIND Wu-FTPd and rpc vulnerabilities, and the exploits them. Then it creates a bastardized PS binary file and hides the original in /usr/bin/adore. Then it tries to download user account files by sending them through email. It's very dangerous. One recommendation is to scan for these changed files and block the outgoing emails via SMTP queues. There is a tool for cleaning up infected machines:

http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm

adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com, adore9001@sina.com are the email addresses

9

IDS552/web-iis_IIS ISAPI Overflow ida nosize

This is an ISAPI IDQ buffer overflow allowing arbitrary code to be run remotely through exploits in almost all versions of IIS Indexing service.

Additional references:

CAN-2001-0500

http://www.eeye.com/html/Research/Advisories/AD20010618.html
http://www.cert.org/advisories/CA-2001-13.html

10

Tiny Fragments - Possible Hostile Activity, Incomplete Packet Fragments Discarded

Sending crafted tiny fragments can cause a host of problems. Tiny fragments can create DOS attacks, or can be reassembled on the other side of the firewall to create dangerous data streams. The firewall may have been performing stateful packet analysis but could easily miss this data because it was not reassembling the whole message. This is the big flaw in the tool known as packet grepping.

They are used for different attacks, but these trigger false alarms also, such as from faulty NIC cards or mis-configured gateways. There were a tremendous amount of them, but again, if a router is out of order, it will spew out a lot of garbage. Port zero is a dead giveaway of something weird going on.

10/03-00:20:57.713480 [**] Tiny Fragments - Possible Hostile Activity [**]
68.83.182.149 -> MY.NET.150.220
10/03-00:20:57.713525 [**] Tiny Fragments - Possible Hostile Activity [**]
68.83.182.149 -> MY.NET.150.220

10/09-12:05:50.237319 [**] Incomplete Packet Fragments Discarded [**]
211.115.216.106:0 -> MY.NET.168.68:0
10/09-12:05:51.331357 [**] Incomplete Packet Fragments Discarded [**]
211.115.216.106:0 -> MY.NET.168.68:0

Some attacks require that certain packets be dropped purposely and other attacks depend on invalid sequencing, and this leaves gaps in the data stream. The data is of course useless. Careful IDS will drop the whole stream before reassembly is attempted.

11

Probable NMAP fingerprint attempt, Queso Fingerprint, IDS28 "PROBE-

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap fingerprint attempt";flags:SFPU; reference:arachnids,05; classtype:attempted-recon; sid:629; rev:1;)

10/09-18:45:54.675038 [**] Probable NMAP fingerprint attempt [**] 203.204.57.59:1952 -> MY.NET.113.7:63022

10/09-18:53:19.886583 [**] Queso fingerprint [**] 209.116.70.75:45831 -> MY.NET.100.217:25

Fingerprinting is the attempt to discover a system's setup or manufacturer, hardware or software, by listening for responses from a scan. The response will include tell-tale signs of a certain kind of system, such as TTL value, frame size, etc. of the packet it generated in response. Notice the combination of two ephemeral ports for the NMAP, and notice the connection to port 25 with Queso. Queso often uses port 6699.

Additional references:

http://www.whitehats.com/info/IDS05

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454

12

IRC evil - running XDCC

No information available

13

Attempted Sun RPC high port access

Watch for activity on port 111, which is Sun's RPC, especially if it originates outside the network. Sun RPC is also available through backdoors, such as the listening port, 32771.

14

IDS284 "Shellcode-x86-setgid0", noop setuid x86 any<-> port UDP, EXPLOIT x86 stealth noop

This is a possible attempt to make a system call to setgid on the x86 platform, but we would need to get the complete packet to see what ascii commands were sent. It works on any<->any port on TCP.

10/09-01:16:12.333675 [**] EXPLOIT x86 setgid 0 [**] 202.102.139.246:1049 -> MY.NET.111.145:2304

10/09-20:00:57.928218 [**] EXPLOIT x86 NOOP [**] 64.4.36.250:80 -> MY.NET.153.193:1408

These two appeared together:

10/09-10:22:04.227731 [**] EXPLOIT x86 stealth noop [**] 202.103.69.65:80 -> MY.NET.88.144:1248
10/09-10:22:20.140241 [**] EXPLOIT x86 NOOP [**] 199.93.170.190:80 -> MY.NET.143.63:1316

The mechanism of the NOOP is a buffer overflow to gain this access.

Additional references:

http://www.whitehats.com/info/IDS291, *http://www.whitehats.com/info/IDS436*

15

Possible MyServer Activity

Snort watches for activity on 55850. MyServer is a DDOS so it needs to establish a covert channel. It installs bastardized versions of UNIX files to create a stealthy subversive network.

10/09-08:45:54.340069 [**] Port 55850 udp - Possible myserver activity - ref. 010313-1 [**] MY.NET.87.233:55850 -> 10.0.1.1:192
10/09-10:48:26.761733 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 168.143.179.114:80 -> MY.NET.168.43:55850

Notice that the first alert is coming from the inside to a private network address - (possibly spoofed). There may already be communication established. Source routed and private network addresses should not be allowed into the network as these are signs of spoofing.

Our alerts showed this activity cascading around the network. There were several such sessions and attempts, followed or proceeded by scans. All of this initiating activity occurred from MY.NET

bigalert.txt:10/03-13:51:20.213917 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.53.72:55850 -> 203.199.83.131:80
bigalert.txt:10/03-13:51:21.159366 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 203.199.83.131:80 -> MY.NET.53.72:55850
bigalert.txt:10/03-13:51:21.159848 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 203.199.83.131:80 -> MY.NET.53.72:55850
bigalert.txt:10/03-13:51:21.160709 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.53.72:55850 -> 203.199.83.131:80
bigalert.txt:10/03-13:51:21.161273 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 203.199.83.131:80 -> MY.NET.53.72:55850
bigalert.txt:10/03-16:47:57.555457 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 61.63.15.66:80 -> MY.NET.198.245:55850
bigalert.txt:10/03-16:47:57.555589 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 61.63.15.66:80 -> MY.NET.198.245:55850
bigalert.txt:10/03-16:47:57.555644 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 61.63.15.66:80 -> MY.NET.198.245:55850
bigalert.txt:10/03-16:47:57.597215 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.198.245:55850 -> 61.63.15.66:80
bigalert.txt:10/03-19:37:43.976162 [**] Port 55850 udp - Possible myserver activity - ref. 010313-1 [**] 204.183.84.240:55850 -> MY.NET.137.7:53
bigalert.txt:10/03-22:10:13.347228 [**] Port 55850 udp - Possible myserver activity - ref. 010313-1 [**] MY.NET.87.233:55850 -> 10.0.1.1:192
bigalert.txt:10/05-06:09:41.092293 [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.6.40:55850 -> 67.99.104.17:25

16

FTP

These FTP alerts are basically cataloguing what the admin has decided is unworthy traffic on his network to and from FTP servers. Remember that anonymous FTP machines must be very locked down and somewhat replaceable.

17

Bugbear Worm/Virus

Additional reference:

http://news.zdnet.co.uk/story/0,,t281-s2123098,00.html

This Trojan horse steals passwords and credit card information by using an exploit in IE 5.01 or 5.5. The MIME header needs a patch. It can be distinguished not by sender or subject line but by length – 50,688 bytes. Port 36794 is used to transmit covert information back to the hacker which has been stored in keystroke logging program files. It installs the Trojan, changes the registry and adds itself to the startup folder in Windows.

18

515 Access

This is the print spooler network port. It should never be accessed remotely and simple blocked both directions at the firewall.

Version 3.6.25 or below in LINUX or BSD are vulnerable attacks on 515.

19

RFB WINVNC

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"INFO VNC server response"; flags: A+; content:"RFB 003.003"; depth:12; classtype:misc-activity; sid:560; rev:2;)

WINVNC is a remote desktop access program, so it's obviously very important to keep it protected. Its passwords are stored in a registry key, which can be manipulated.

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\

CVE: CAN-2000-1164

20

Exploit NPTDX Buffer Overflow

alert udp $EXTERNAL_NET any -> $HOME_NET 123 (msg:"EXPLOIT ntpdx

overflow attempt"; dsize: >128; reference:arachnids,492; reference:bugtraq,2540; classtype:attempted-admin; sid:312; rev:2;)

NTP is Network Time Protocol. It synchronizes date and time between computers on a network over UDP. An older version, version 3 is also called XNTP. It is subject to buffer overflow attempts, which crash the daemon or can gain access to the system and execute commands, even to the point of giving the attacker root access. It is a very serious exploit. Windows now uses a version of NTP but I don't think it's the same. Port 123 must be watched.

Cisco, and various versions of UNIX including OSX for the MAC are all vulnerable.

Additional Reference:

http://online.securityfocus.com/bid/2540/discussion/

21

Russia Dynamo - SANS Flash 28-jul-00

This is traffic from port 6699 to 2478 leaving our network. The address it is going to is 194.87.6.38, which is a Russian address known for this. So this is more of a watch-list event. In my logs all of them involved these three ports.

10/03-09:38:17.901898 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.131:1427 -> MY.NET.150.133:1214
10/03-09:38:18.603893 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.131:1427 -> MY.NET.150.133:1214
10/03-09:38:18.604219 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.150.133:1214 -> 194.87.6.131:1427
10/03-09:42:12.862231 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.131:1673 -> MY.NET.150.133:1214
10/03-09:42:12.862616 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.150.133:1214 -> 194.87.6.131:1673

Here is the SANS FLASH on this subject:
http://archives.neohapsis.com/archives/sans/2000/0068.html

SANS recommends blocking this network from any access to your network.

22

Shaft DDOS client activity on port 20432

In November 1999, the Shaft DDoS tool became available. A Shaft network looks conceptually similar to a trinoo; it is a packet flooding attack and the client controls the size of the flooding packets and duration of the attack. One interesting signature of Shaft is that the sequence number for all TCP packets is 0x28374839.[*]

Additional Reference:

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

[*] Kessler, Gary C.

My method of Analysis:

These are my goals:

1. Consider all of the categories of alerts and what services they attack.
2. Find ports to filter or watch on the firewalls.
3. Keep an eye on the dangerous hackers.
4. Do this with as simple a process as possible.

This is my method for achieving them:

1. Briefly familiarizing myself with the attacks on the alert list, as to which ports they are attacking.
2. Top talkers in the alert, OOS and scan files are analyzed.
3. I compare a list of standard 'interesting ports' and ports with high volume of the attacks in the alert log, with the attacks and attackers.
4. Whatever selected ports, not obvious as to their use and how they were attacked, based on what has been reviewed up to this point are analyzed against the alert, OOS and scan logs to see if there is a trail of activity.
5. Any remaining anomalies related to the alerts, and attacked ports are found and either dismissed or catalogued.
6. Top talkers from all three lists are linked to their activities and those who are deemed to be most active and/or dangerous are made candidates for the detailed watch-list or suggestions for firewall filtering, if outside the network. Unusual traffic patterns are analyzed.
7. Current trends are analyzed based on a summary of the data.
8. The rest of the data is discarded.

This covers a surprisingly large percentage of total events, but not all of them. In the business of analysis of millions of logged events, we can't look at them all. Step 8 is the reality of this business.

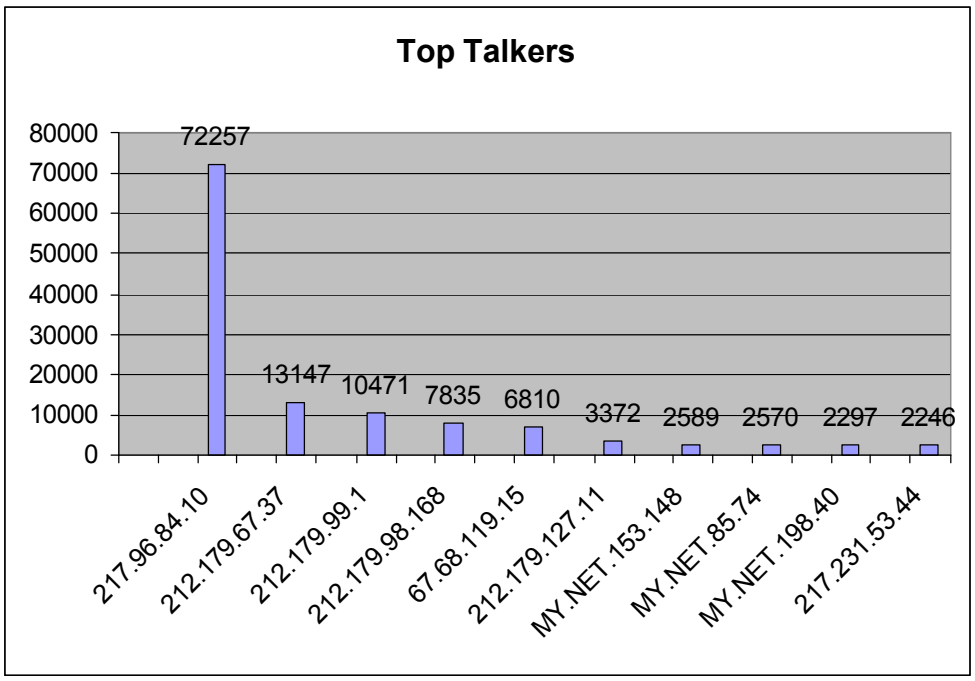
Scans: Top Talkers

976507CLASS.B.111.140
287562CLASS.B.83.146
140568CLASS.B.84.137
86715CLASS.B.70.207
81337CLASS.B.91.240
78129CLASS.B.87.50
57184CLASS.B.198.40

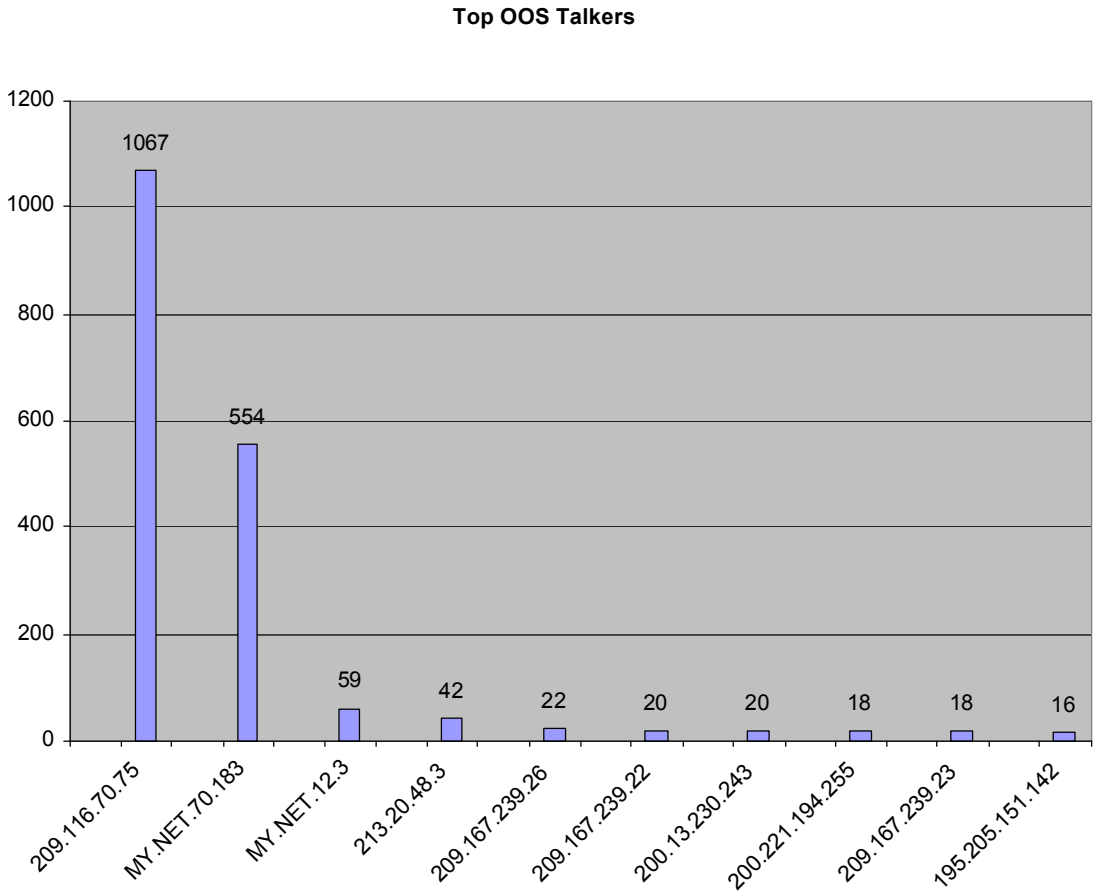
45264CLASS.B.111.216
35819CLASS.B.88.155
31298CLASS.B.137.7
24369CLASS.B.198.204
24343CLASS.B.84.178
18679CLASS.B.84.147
16588CLASS.B.70.176
13418CLASS.B.111.214
12476CLASS.B.71.173
11633CLASS.B.114.45
10649CLASS.B.150.113

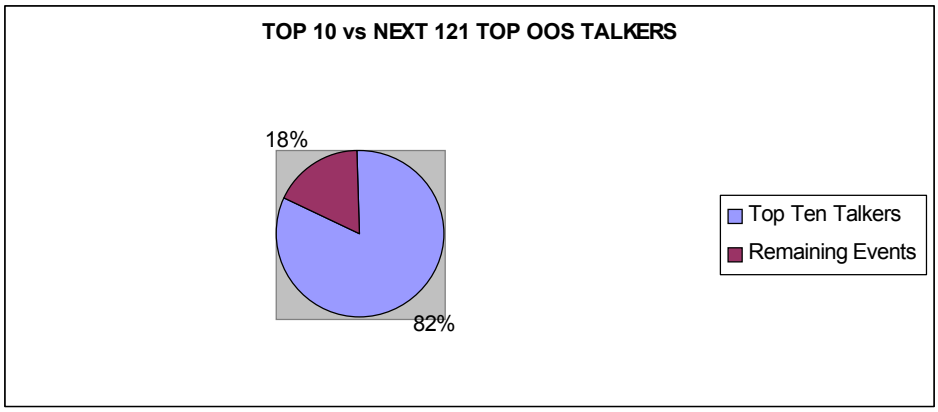
A lot of scanning activity is taking place on our network (sanitized as CLASS.B). None of these addresses match the other Top Talkers lists. This may be a college campus where the hackers move around a lot and a lot of scanning activity goes unchecked.
Hint: Remember the very Talking-est Talker for later.

Alerts: Top Talkers



None of the top OOS talkers have anything to do with the top alert talkers. Apparently the packet crafting people and the script kiddie/canned exploit people do not mingle.





Of course we have to keep in mind, that the best hackers are not always the noisiest. An expert attacker will be able to do recon, get in and get out under the radar with a minimum of communication. Nevertheless at the end of our study, we will keep this list in mind for analysis of five addresses to research. More

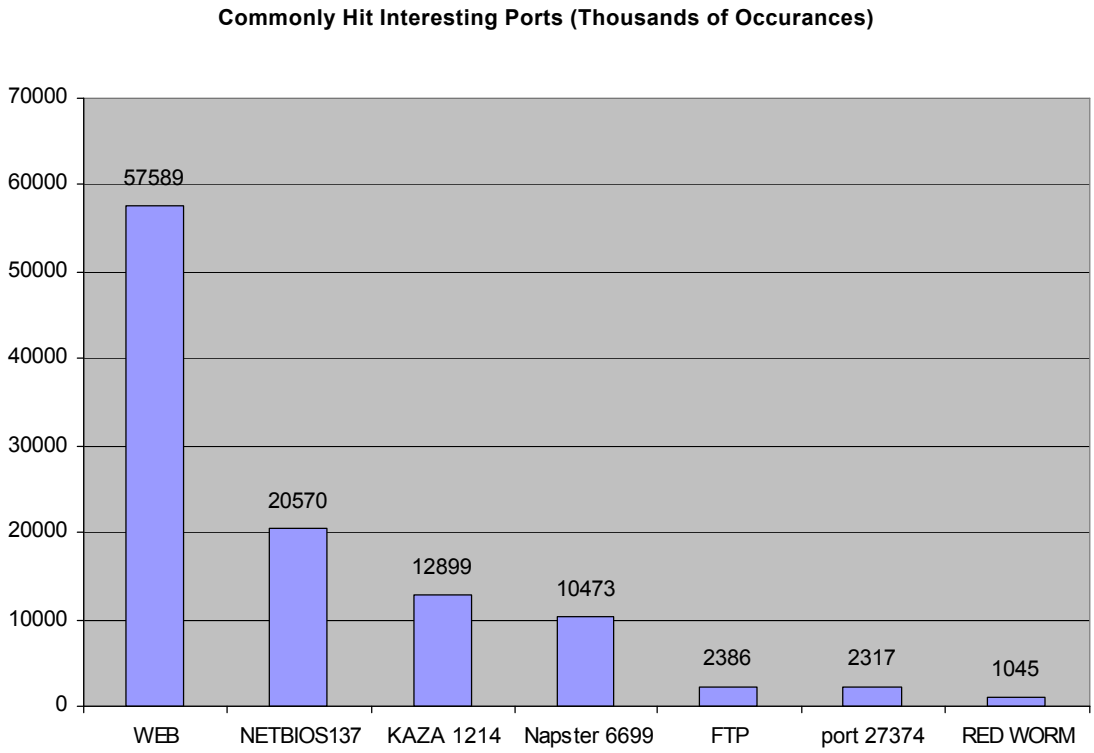
research is required to learn what they are doing, and OOS hackers that are crafting packets are arguably among the most sophisticated hackers.

TOP DESTINATION PORTS

A total of 474 ports were hit in the alert logs. We're analyzing the most 'interesting' and most popular.

These tables speak to how much traffic in our alerts is generated by well-known TCP ports. Two criteria were used to select what I refer to as 'interesting ports'.

1. Firewalls frequently require them to be open to some degree for business purposes.
2. They are the targets of well-known exploits.

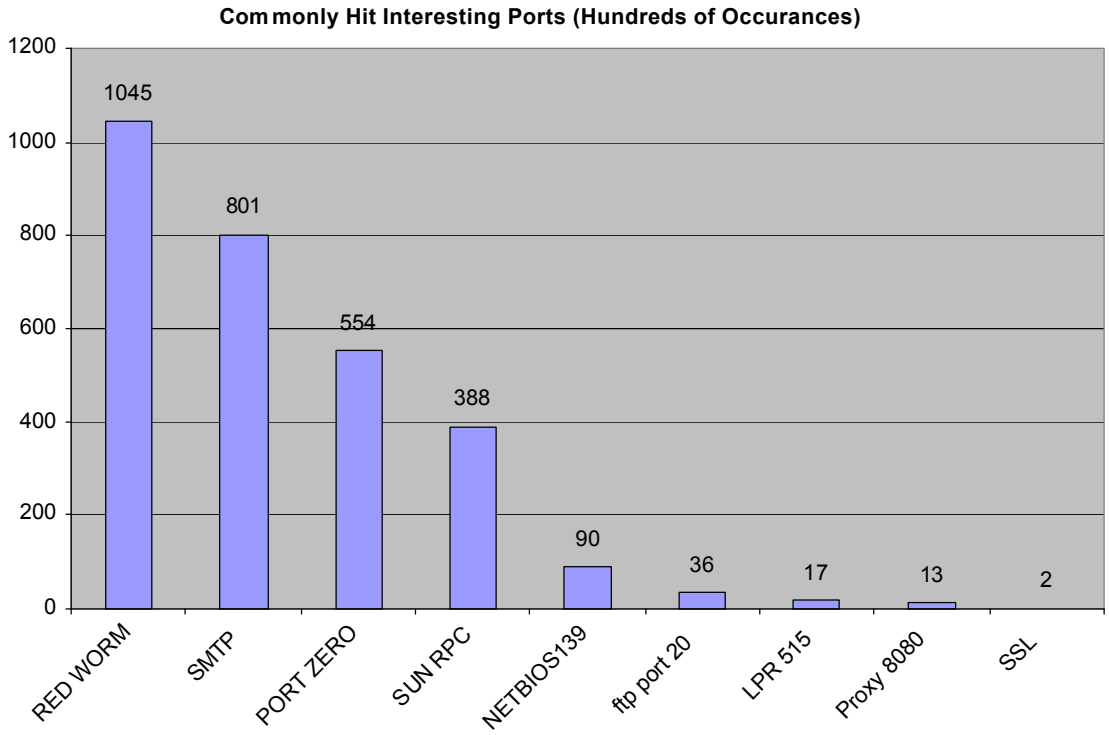


- Port 27374 - It has been used by W32.leave.worm and Sub Seven attacks.

- Netbios - It is accounted for by SMB wildcard attacks and should be

blocked at the router.

- Kaza - 1214 appeared in the Russian Dynamo and Red Worm traffic.
- Napster 6699 - This port is also used for more than one exploit mentioned above and if nothing else is a neusance for administrators as most Napster activity ends up just taking up fileserver space and productive time in a business.
- FTP - Many of these alerts were informational and depend on what traffic is allowed, however there was a lot of file globbing attack attempts. Fortunately they were blocked.
- Red worm - Port 65535 was working in tandem with 1214 on many alerts



Red Worm has been repeated to add perspective from the preceding chart. I found it odd that SSL had only been attempted twice. Especially since there was a new high-profile case of a 'flaw' found in the MS implementation of SSL. *Hint: Make a mental note of this for later.*

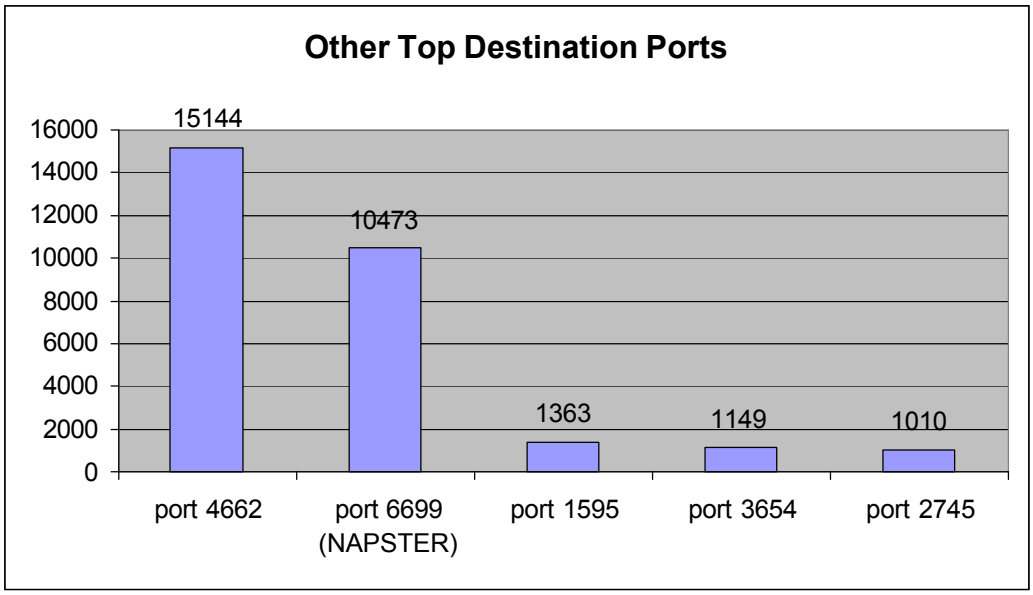
- SMTP - This port is needed for traffic and there is no shortcut to providing carefully configured SMTP queues for your network.
- Port 0 is obviously interesting because it's reserved and never used for legitimate network traffic!
- Sun RPC was discussed above in the highport access alert
- Netbios - See entry for port 137 in the first chart
- FTP - See entry for port 21 in the first chart
- LPR - this was discussed above and should be blocked at the router
- Proxy - These are the alert entries for an attack on a Webserver. Maybe some of them tried to sneak through a proxy.

```
10/02-03:24:13.358038  [**] spp_http_decode: CGI Null Byte attack detected [**]
MY.NET.163.146:1064 -> 207.105.75.42:8080
10/02-03:24:13.358038  [**] spp_http_decode: CGI Null Byte attack detected [**]
MY.NET.163.146:1064 -> 207.105.75.42:8080
10/02-03:24:13.358038  [**] spp_http_decode: CGI Null Byte attack detected [**]
MY.NET.163.146:1064 -> 207.105.75.42:8080
10/02-03:24:13.358038  [**] spp_http_decode: CGI Null Byte attack detected [**]
MY.NET.163.146:1064 -> 207.105.75.42:8080

10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808  [**] spp_http_decode: IIS Unicode attack detected [**]
```

MY.NET.91.103:1935 -> 211.63.185.26:8080
10/02-10:34:10.490808 [**] spp_http_decode: IIS Unicode attack detected [**]
MY.NET.91.103:1935 -> 211.63.185.26:8080

This graph represents the remainder of destination ports that were hit over 1000 times.



Port 3654 was being used to connect to 27374, which is very not good as we commented above. This was being done by 12.249.72.167.

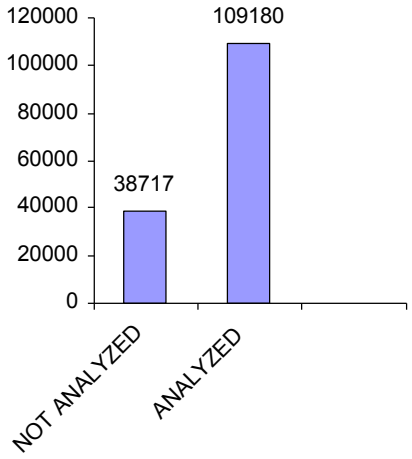
Name: 12-249-72-167.client.attbi.com
Address: 12.249.72.167

212.179.48.2 was on the watchlist for port 2745.

A few of these ports were hit incidentally during portscans. This accounted for the rest of the incidents.

Except for 4662, that covers the Top Destination Ports and all "Interesting Ports." As you can see, we have already covered about 3/4 of the total number of events in the alert log.

Attack Breakdown By Port



I decided to check the OOS files for more information on 4662 as the alert log didn't give me much to go on. It was just a lot of scanning activity. But why the interest on 4662?

****

THE PORT 4662 MYSTERY

In the OOS files, a few connections on port 4662 showed up.

grep :4662 bigoos.txt

10/02-09:31:15.562278 217.81.50.191:4662 -> MY.NET.111.215:13
10/02-22:40:17.723350 211.176.140.23:4662 -> MY.NET.111.216:3387
10/02-23:35:08.764336 80.14.123.83:2704 -> MY.NET.168.253:4662
10/02-23:35:11.758208 80.14.123.83:2704 -> MY.NET.168.253:4662
10/03-09:57:13.063747 61.221.38.18:3543 -> MY.NET.111.216:4662
10/03-14:30:41.159838 217.225.59.42:21127 -> MY.NET.82.50:4662
10/03-15:46:19.991654 213.141.40.189:38827 -> MY.NET.82.50:4662
10/03-17:33:29.847917 X.Y.152.208:42519 -> MY.NET.71.173:4662
10/03-22:21:09.191585 213.141.40.189:40915 -> MY.NET.82.50:4662
10/09-19:35:03.934259 218.21.82.139:4662 -> MY.NET.111.214:3897
10/09-19:36:06.325787 218.21.82.139:4662 -> MY.NET.111.214:3897
10/09-19:36:17.567190 218.21.82.139:4662 -> MY.NET.111.214:51f

Thinking the attack was coming from the outside, I grepped for each address that showed up in the OOS log in the scan file and found only a few matches.

grep $addresses bigscans.txt

Oct 3 09:57:13 61.221.38.18:3543 -> MY.NET.111.216:4662 NULL ********
Oct 3 12:10:58 213.141.40.189:46619 -> MY.NET.82.50:4662 SYN 12****S*

However, one address had many entries on port 4665.

Oct 2 08:51:30 MY.NET.111.214:2134 -> X.Y.152.208:4665 UDP
Oct 2 08:53:00 MY.NET.111.214:2134 -> X.Y.152.208:4665 UDP
Oct 2 20:31:14 MY.NET.111.214:4466 -> X.Y.152.208:4665 UDP
Oct 2 21:07:36 MY.NET.168.186:1193 -> X.Y.152.208:4665 UDP
Oct 3 16:33:54 MY.NET.111.214:2805 -> X.Y.152.208:4665 UDP
Oct 3 22:19:54 MY.NET.71.173:1833 -> X.Y.152.208:4665 UDP
Oct 3 23:21:46 MY.NET.71.173:1833 -> X.Y.152.208:4665 UDP
Oct 3 23:22:54 MY.NET.71.173:1833 -> X.Y.152.208:4665 UDP
Oct 3 23:24:45 MY.NET.71.173:1833 -> X.Y.152.208:4665 UDP
Oct 5 15:41:28 MY.NET.168.186:1122 -> X.Y.152.208:4885 UDP

I found something else. The MY.NET address was showing a lot of activity. So I did some more checking for it in connection with port 4665. I found it was scanning for this port and sending SYN******S packets to port 443 for a long list of machines.

With grep, I found 976,512 scans from MY.NET.111.140 address to ports 443, methodically going through huge lists of large subnets on the internet, looking for port 443. 2,148,065 total entries were found for the MY.NET class subnet.

Here is a sample:

Oct 2 01:51:02 MY.NET.111.215:2301 -> 80.131.122.216:4665 UDP
Oct 2 01:51:02 MY.NET.111.215:2301 -> 217.83.188.79:4665 UDP
Oct 2 01:51:03 MY.NET.111.215:2301 -> 80.131.132.180:4665 UDP
Oct 2 01:51:03 MY.NET.111.215:2301 -> 172.185.2.117:4665 UDP
Oct 2 01:51:03 MY.NET.111.215:2301 -> 217.80.110.228:4665 UDP
Oct 2 01:51:04 MY.NET.111.215:2301 -> 80.132.22.239:4665 UDP
Oct 2 01:51:04 MY.NET.111.215:2301 -> 62.4.19.197:4665 UDP
Oct 2 01:51:04 MY.NET.111.215:2301 -> 192.168.0.1:4665 UDP
Oct 2 01:51:04 MY.NET.111.215:2301 -> 217.68.165.4:4665 UDP
Oct 2 01:51:06 MY.NET.111.215:2301 -> 210.117.67.218:4665 UDP
Oct 2 01:51:06 MY.NET.111.215:2301 -> 212.185.224.133:4665 UDP
Oct 2 01:51:06 MY.NET.111.215:2301 -> 217.226.49.236:4665 UDP
Oct 2 01:51:06 MY.NET.111.215:2301 -> 80.131.128.99:4665 UDP
Oct 2 01:51:07 MY.NET.111.215:2301 -> 217.80.110.204:4665 UDP
Oct 2 01:51:07 MY.NET.111.215:2301 -> 80.131.70.139:4665 UDP
Oct 2 01:51:07 MY.NET.111.215:2301 -> 213.23.37.13:4665 UDP
Oct 2 01:51:07 MY.NET.111.215:2301 -> 80.130.146.176:4665 UDP
Oct 2 01:51:08 MY.NET.111.215:2301 -> 80.135.78.237:4665 UDP
Oct 2 01:51:08 MY.NET.111.215:2301 -> 217.84.1.93:4665 UDP
Oct 2 01:51:08 MY.NET.111.215:2301 -> 80.128.68.210:4665 UDP
Oct 2 01:51:08 MY.NET.111.215:2301 -> 80.133.60.82:4665 UDP
Oct 2 01:51:08 MY.NET.111.215:2301 -> 62.245.160.134:4665 UDP
Oct 2 01:51:09 MY.NET.111.215:2301 -> 80.131.107.149:4665 UDP
Oct 2 01:51:09 MY.NET.111.215:2301 -> 217.82.90.116:4665 UDP
Oct 2 01:43:34 MY.NET.111.140:4665 -> 117.135.245.139:443 SYN ******S*
Oct 2 01:54:13 MY.NET.111.140:4665 -> 119.49.41.240:443 SYN ******S*
Oct 2 01:58:24 MY.NET.111.140:4665 -> 119.49.78.129:443 SYN ******S*
Oct 2 02:15:25 MY.NET.111.140:4665 -> 149.55.204.29:443 SYN ******S*
Oct 2 02:18:59 MY.NET.111.140:4665 -> 17.161.3.222:443 SYN ******S*

Oct 2 02:19:38 MY.NET.111.140:4665 -> 17.161.8.218:443 SYN ******S*
Oct 2 02:21:58 MY.NET.111.140:4665 -> 86.31.71.13:443 SYN ******S*
Oct 2 02:30:47 MY.NET.111.140:4665 -> 17.161.108.31:443 SYN ******S*
Oct 2 02:27:18 MY.NET.111.140:4665 -> 17.161.77.13:443 SYN ******S*
Oct 2 02:36:42 MY.NET.111.140:4665 -> 17.161.159.182:443 SYN ******S*
Oct 2 02:37:54 MY.NET.111.140:4665 -> 17.161.170.68:443 SYN ******S*
Oct 2 02:40:47 MY.NET.111.140:4665 -> 100.185.171.18:443 SYN ******S*
Oct 2 02:48:30 MY.NET.111.140:4665 -> 175.84.7.97:443 SYN ******S*
Oct 2 02:44:20 MY.NET.111.140:4665 -> 27.32.11.112:443 SYN ******S*
Oct 2 03:00:53 MY.NET.111.140:4665 -> 122.179.91.75:443 SYN ******S*
Oct 2 03:19:47 MY.NET.111.140:4665 -> 88.121.1.149:443 SYN ******S*
Oct 2 03:08:35 MY.NET.111.140:4665 -> 27.32.224.229:443 SYN ******S*
Oct 2 03:22:11 MY.NET.111.140:4665 -> 52.23.47.101:443 SYN ******S*
Oct 2 03:22:44 MY.NET.111.140:4665 -> 52.23.52.97:443 SYN ******S*
Oct 2 03:31:37 MY.NET.111.140:4665 -> 119.131.171.56:443 SYN ****

By the date-stamps I was able to piece together what happened

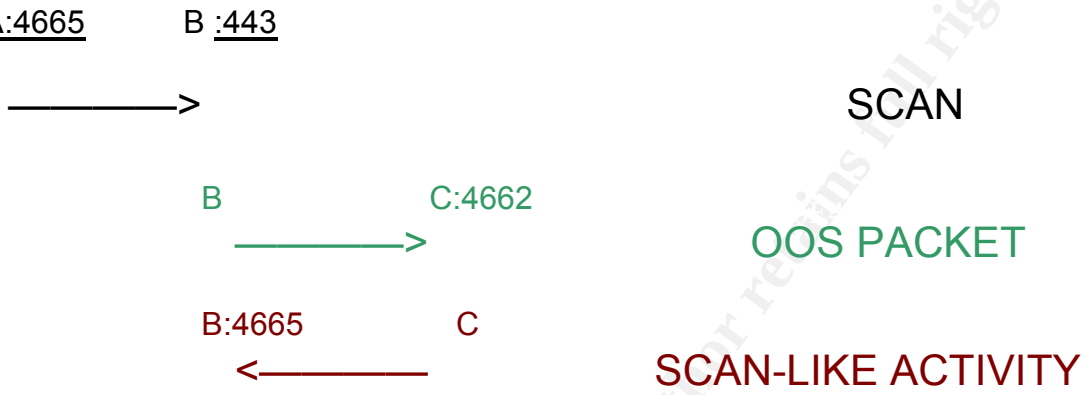
1. Machine A (MY.NET) was scanning for 443 hosts through GIAC U network, trying to connect through 4665.
2. He found one on Machine B (x.x. 152.208).
3. The next day machine B connected to machine C (My.Net.71.173) on port 4662 and it responded on 4665. No alerts, or OOS logs report any attempted access at any web servers on port 443 on the MY.NET network, but it is possible something slipped under the radar.

It would probably be a good idea to watch our secure servers closely and log entries. Keeping up with the latest news on IIS and SSL security should be a priority.

Fortunately he accounts for most of the 443 scanning. There were 7 others who tried from outside the network to scan for 443 on MY.NET. None of them appeared on the Alerts logs or the OOS logs. In the end. was looking for reasons 4662 appeared on the alerts list and wound up finding out a lot about SSL activity.

The moral of this story is you'll never know what you'll find if you start digging. This is just one example. This guy from our campus may be trying to use covert channels or some sort of spoofed connection to exploit SSL servers.

LINKCHART



POSTSCRIPT:

So what was all that 4662 activity?

It turned out to be almost entirely watchlist activity from 212.179.67.37. 4662-4665 are available channels. They are ephemeral ports also, which makes them ideal to create covert channels. They can't be blocked by the firewall because they may be needed for client sides of server sessions and they're probably not closely monitored because they have no official use.

Port 1595 was also being used by the same address to connect to 4662. This is the port for RADIO.

http://isc.incidents.org/ lists port 4665 as one of the most popular ports right now but they do not know why.

****

CORRELATIONS

Correlations were made to alert log files with similar systems, ie. mixed UNIX and Windows alerts. The first was from logs six months old. The second was from logs approximately a year old. The goal was to figure out how attacks are evolving and what new attacks are out, as well as finding which attacks had died out in popularity and which had resurfaced. Also a comparison between levels of alerts was made. My alerts were more forgiving. The other sets of logs had more informational messages included.

Correlation to GIAC from logs six months old
Gary Smith GIAC - logs from late march 2002

I notice in this log, first of all that ICMP was generating a lot of alerts where mine didn't. We are not interested in ICMP whereas he was. His alerts also feature INFO messages, which catalogue more innocuous activity such as ping requests, failed logins and Napster clients' login, which mine does not. Other than that, there are only a few differences in his alerts from mine that I would call significant.

Some of the activity picked up was the result of scans, which should show up in the scans log such as SQUID, but some would not because the scans were too low and slow such as the FIN scans. Some of the ICMP activity was not so innocuous such as specific echo requests like ICMP BSD TYPE requests, and traceroute activity, and possible IRC access from outside. Other successful events such as successful FrontPage Administrator access through the web were also not catalogued on my alerts. The weakness in my alerts is that if a hacker succeeds in breaking into a service such as IIS, I will not have any way of doing forensics, because informational alerting was not turned on. I would have to rely on IIS log files to trace attackers. Successfully logging in may be a red flag if you are keeping careful track of such things, but this creates a lot of log entries and requires the admin to be on his toes. If we are logging one IIS server for admin access, this is not so daunting a task. If we have a few hundred servers, it could be.

Other events were so similar to mine that they would probably be found, such as the many exploits of FrontPage and the NOOP buffer overflow. These are follow-up attacks for the FPSE access and recon tests that were discussed and detected in the detects section. It's a judgment call what to include as different enough to record. I have removed some alerts because they were similar to mine. I argue they would have been caught by my alert set. Many times, an attack can be detected by more than one alert, and it's just a matter of which rule is higher in the search order.

This chart represents the significant differences, and the examples mentioned above, which are highlighted in yellow. Some very infamous attacks either are not being used any more, have been adapted and renamed, or went undetected in my alerts. These are highlighted in red.

SNMP public access	35563	26	146	SNMP
MISC Large UDP Packet	21139	31	19	DOS
WEB-IIS view source via translate header	783	40	2	IIS
WEB-FRONTPAGE _vti_rpc access	359	117	2	IIS
INFO Possible IRC Access	96	23	16	
WEB-CGI rsh access	49	1	1	Web
INFO Inbound GNUTella Connect accept	**35**	**6**	**21**	**GNUTella**
WEB-MISC http directory traversal	33	6	2	Web
WEB-CGI csh access	30	1	1	Web
WEB-MISC compaq nsight directory traversal	23	5	5	Compaq Web
MYPARTY - Possible My Party infection	**18**	**3**	**1**	
WEB-CGI scriptalias access	17	2	1	Web
Back Orifice	**11**	**5**	**7**	
FTP CWD / - possible warez site	11	1	11	FTP
SMB CD...	10	1	1	fileshare
WEB-COLDFUSION administrator access	5	1	1	Web
ICMP Echo Request BSDtype	5	2	1	scan
SCAN FIN	4	1	1	scan
WEB-FRONTPAGE author.exe access	2	1	1	legit
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	2	2	2	OS
WEB-CGI formmail access	2	2	1	Web
WEB-MISC whisker head	2	2	1	
RPC tcp traffic contains bin_sh	2	2	2	OS
MISC traceroute	2	1	1	Recon
BACKDOOR SIGNATURE - Q ICMP	1	1	1	

BACKDOOR NetMetro Incoming Traffic	1	1	1	
IDS50/trojan_trojan-active-subseven	1	1	1	
Virus - Possible scr Worm	1	1	1	

Only the following four attacks are new:

- IRC evil - running XDCC
- Bugbear@MM virus in SMTP
- EXPLOIT NTPDX buffer overflow
- DDOS shaft client to handler (port 20432).

This is good news. Over a six month time, learning about four new attacks is a reasonable about of research to do. Bugbear is a new exploit. Shaft DDOS is not. The NTPDX buffer overflow was first posted to Bugtraq on April 4th of last year, so it's not new either.

Correlation to GIAC from logs one year old.
Ruth Kizlyk GIAC – logs from late November early December 2001

Once again removing informational messages and ICMP messages, I dramatically reduced the number of alerts to the significant alerts and found that she still had a very large list of alerts. Her list differed from mine significantly. She had roughly 40 alerts that I did not. She had quite a large number of alerts to begin with. Most of these have vanished, with still very few new appearing on my list. Even though I was able to sort mine into 22 different categories, I think my list is at the outer limit of attacks an admin can reasonably keep track of in a week. Ruth had 157 total alerts. I had 45.

FIVE LOOKUPS

This address was one the OOS Top Talkers.

C:\>nslookup 209.116.70.75
Server: dns.my.dns.server
Address: x.y.z.a

Name: vger.kernel.org
Address: 209.116.70.75

WHOIS result:

Registrant:
Transmeta Corporation (KERNEL2-DOM)
3940 Freedom Circle
Santa Clara
CA,95054
US

Domain Name: KERNEL.ORG

Administrative Contact:
kernel.org hostmaster (KO1380-ORG)
HOSTMASTER@KERNEL.ORG
Transmeta Corporation
3940 Freedom Circle
Santa Clara , CA 95054
US
(408) 919-3000
Fax- (408) 919-1199
Technical Contact:
Transmeta Hostmaster (TH11-ORG)
HOSTMASTER@TRANSMETA.COM
Transmeta Corporation
3940 Freedom Circle
Santa Clara, CA 95054
US
(408) 919-3000
Fax- (408) 919-1199

Record expires on 08-Mar-2003.
Record created on 07-Mar-1997.
Database last updated on 28-Oct-2002 14:27:03 EST.

Domain servers in listed order:

NS2.KERNEL.ORG 204.152.189.113
NS1.KERNEL.ORG 64.158.222.226
NS1.TRANSMETA.COM 63.209.4.198
NS2.GIMP.ORG 195.92.249.252
NS.VGER.KERNEL.ORG 209.116.70.75

This person might be a LINUX guy. This domain has a web page with Linux information. Check for LINUX vulnerabilities. He's been sending a lot of 12*S* flag TCP bytes to port 25 to what may be the mail server. He's using reserved bits in the TCP header to do something***.

2.

This address was among two from a domain which were the only other traceable OOS Top Talkers.

C:\>nslookup 209.167.239.22
Server: dns.dnsmachine.org
Address: x.y.a.z

Name: out12.greatoffrs.com
Address: 209.167.239.22

Query by number produced this:

The previous information has been obtained either directly from the registrant or a registrar of the domain name other than VeriSign. VeriSign, therefore, does not guarantee its accuracy or completeness

inetnum: 0.0.0.0 - 255.255.255.255
netname: IANA-BLK
descr: The whole IPv4 address space
country: NL
admin-c: IANA1-RIPE
tech-c: IANA1-RIPE

status: ALLOCATED UNSPECIFIED
remarks: The country is really worldwide.
remarks: This address space is assigned at various other places in
remarks: the world and might therefore not be in the RIPE database.
mnt-by: RIPE-NCC-HM-MNT
mnt-lower: RIPE-NCC-HM-MNT
mnt-routes: RIPE-NCC-NONE-MNT
changed: bitbucket@ripe.net 20010529
changed: bitbucket@ripe.net 20020625
source: RIPE
role: Internet Assigned Numbers Authority
address: see http://www.iana.org.
e-mail: bitbucket@ripe.net
admin-c: IANA1-RIPE
tech-c: IANA1-RIPE
nic-hdl: IANA1-RIPE
remarks: For more information on IANA services
remarks: go to IANA web site at http://www.iana.org.
mnt-by: RIPE-NCC-MNT
changed: bitbucket@ripe.net 20010411
source: RIPE

Not very helpful.

3.

Although the watchlist and many practicals and lists have the Russian Dynamo address recorded already, I thought it might be a good idea to do a fresh query to see if anything new appeared.

C:\>nslookup 194.87.6.131
Server: dns.my.server.org
Address: x.y.z.a

Name: 131.6.87.194.dynamic.dol.ru
Address: 194.87.6.131

inetnum: 194.87.6.0 - 194.87.6.255
netname: DEMOS-DOL-DIALUP
descr: DEMOS-Online Dialup
descr: Demos-Internet Co.
descr: Moscow, Russia
country: RU
admin-c: DNOC-ORG
tech-c: DNOC-ORG
status: ASSIGNED PA

mnt-by: AS2578-MNT
remarks: **********
remarks: Please send abuse reports to abuse@demos.su
remarks: **********
changed: rvp@demos.net 20020911
source: RIPE
route: 194.87.0.0/19
descr: DEMOS
origin: AS2578
notify: noc@demos.net
mnt-by: AS2578-MNT
changed: noc@demos.net 20000927
source: RIPE
role: Demos Internet NOC
address: Demos Company Ltd.
address: 6-1 Ovchinnikovskaya nab.
address: Moscow 115035
address: Russia
phone: +7 095 737 0436
phone: +7 095 737 0400
fax-no: +7 095 956 5042
e-mail: ncc@demos.net
admin-c: KEV-RIPE
admin-c: RPS-RIPE
admin-c: GVS-RIPE
tech-c: KEV-RIPE
tech-c: RPS-RIPE
tech-c: GVS-RIPE
nic-hdl: DNOC-ORG
notify: hm-dbm-msgs@ripe.net
notify: ncc@demos.net
notify: ip-reg@ripn.net
mnt-by: AS2578-MNT
changed: evgeny@demos.su 20021021
source: RIPE

Abuse hotline is recorded. It's worth at least a phone call or email.

4.

I looked up the other computer in the 4662 mystery to either investigate or warn him.

C:\>nslookup 207.6.152.208
Server: dnsssp100.ncr.disa.mil
Address: 164.117.82.7

Name: aoi291gy3gc.bc.hsia.telus.net
Address: 207.6.152.208

There was no WHOIS entry for any part of the domain, including searching for just telus.net, so I did a tracert. Top snipped for sanitization.

…

6	40 ms	50 ms	50 ms	198.26.119.81
7	40 ms	60 ms	50 ms	wdc-edge-07.inet.qwest.net [63.148.66.221]
8	40 ms	50 ms	60 ms	wdc-core-03.inet.qwest.net [205.171.24.129]
9	50 ms	50 ms	50 ms	dca-core-03.inet.qwest.net [205.171.8.213]
10	50 ms	50 ms	50 ms	dca-core-01.inet.qwest.net [205.171.9.9]
11	121 ms	120 ms	120 ms	svl-core-02.inet.qwest.net [205.171.8.202]
12	120 ms	121 ms	120 ms	svl-core-03.inet.qwest.net [205.171.14.126]
13	120 ms	120 ms	121 ms	pax-brdr-02.inet.qwest.net [205.171.205.30]
14	120 ms	130 ms	120 ms	plalca01gr00.bb.telus.com [154.11.3.13]
15	140 ms	141 ms	150 ms	sttlwa01gr02.bb.telus.com [154.11.10.1]
16	171 ms	160 ms	170 ms	nwmrbc01br01.bb.telus.com [209.53.75.177]
17	161 ms	170 ms	160 ms	vancbc01br01.bb.telus.com [209.53.75.221]
18	160 ms	170 ms	160 ms	clgrab21br01.bb.telus.com [154.11.10.22]
19	160 ms	160 ms	170 ms	clgrab31br01.bb.telus.com [154.11.10.165]
20	160 ms	180 ms	171 ms	edtnabxmbr01.bb.telus.com [154.11.10.149]
21	160 ms	170 ms	171 ms	edtnabxmgr01.bb.telus.com [154.11.10.141]
22	170 ms	171 ms	420 ms	edtnabkddr00.bb.telus.com [205.233.111.133]
23	170 ms	160 ms	170 ms	161.184.255.147
24	160 ms	171 ms	170 ms	cityweb.telus.net [198.161.157.214]

A whois on 154.11.10.0 produces this:

inetnum: 0.0.0.0 - 255.255.255.255
netname: IANA-BLK
descr: The whole IPv4 address space
country: NL
admin-c: IANA1-RIPE
tech-c: IANA1-RIPE
status: ALLOCATED UNSPECIFIED
remarks: The country is really worldwide.
remarks: This address space is assigned at various other places in
remarks: the world and might therefore not be in the RIPE database.
mnt-by: RIPE-NCC-HM-MNT
mnt-lower: RIPE-NCC-HM-MNT
mnt-routes: RIPE-NCC-NONE-MNT
changed: bitbucket@ripe.net 20010529
changed: bitbucket@ripe.net 20020625

source: RIPE
role: Internet Assigned Numbers Authority
address: see http://www.iana.org.
e-mail: bitbucket@ripe.net
admin-c: IANA1-RIPE
tech-c: IANA1-RIPE
nic-hdl: IANA1-RIPE
remarks: For more information on IANA services
remarks: go to IANA web site at http://www.iana.org.
mnt-by: RIPE-NCC-MNT
changed: bitbucket@ripe.net 20010411
source: RIPE

Whoever this destination is, they've disguised themselves very well. Maybe Rip.Net has some answers. Is hop number 22 referring to a backdoor? And If so, a backdoor to what?

5.

Since very little information was available about the WINVNC attack, I decided to research the initiator of the alert.

C:\>nslookup 68.33.45.145
Server: dns.machine.org
Address: a.b.d.c

Name: pcp02465960pcs.chrchv01.md.comcast.net
Address: 68.33.45.145

This is probably a dynamically assigned cable modem address. Comcast is a cable carrier. A call to their abuse line may be in order.

Recommendations

1. Step by step IIS configuration can be found in many books and on the Internet. It is imperative to lock down IIS and enable automatic update service on all Web Servers. Some basic things to get started are:

    - Keep the IIS directory on a separate physical hard drive.
    - Make good use of several Virtual Directories to segregate Web content which is on the same server.
    - Remove FrontPage from your servers. See Network detect section

on information on why.
- Make sure that IIS has its current patches installed.
- Make sure the scripts directory denies directory traversal permission.
- Log All SSL activity verbosely.

2. Block ports 4662-4665 incoming and outgoing.

3. Many services do not need to be run from outside the network and therefore their accompanying ports should be blocked both ways. These are RPC, filesharing, and printing:
   137, 139, 32771, 111, 515,

4. Block port 6699. Sell the idea to management as a way to prevent users from visiting Napster instead of working and filling up the file server with illegal music files.

5. Block port 27372, 65535, and 55850 (from the Top Ports list.)

6. Monitor the SMTP queue and prevent message from being sent to adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com, adore9001@sina.com. (Red Worm)

7. FTP servers are unfortunately very needed in a college campus and create huge amounts of traffic. Make sure to separate anonymous read, authenticated write and read only FTP partitions as much as possible, even so far as to put them on entirely different machines or virtual servers. Make these servers expendable and secure them from accessing other servers. See this link for information on file globbing vulnerabilities.

   http://www.cert.org/advisories/CA-2001-07.html

8. Block internal address blocks from the router. They are not needed and are a sign of MyServer activity.192.168.0.0.-192.168.255.255, 172.16.0.0.-172.31.255.255, 10.0.0.0-10.255.255.255.

9. Block watchlist ports 1167 and 1214.

10. Add to startup scripts a section to check for and remote Trojan horses. On the UNIX side, have the script install the patch from Dartmouth listed above, search for /usr/bin/adore directories and report back any finds until RED WORM and its variants die down in popularity. Run this script daily.

11. On the Windows side, have the startup script remove standard users and everybody permissions from this registry key on all Windows machines HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\.

12. Make sure Network Time Protocol is blocked at the router.

13. Install file monitoring and logging on all servers requiring access from port 443.

14. Segregate the WAN into small LANs with different purposes and limited access to each other. The top scanning talkers should not be able to get responses from anything outside the computer labs they are probably using. Consider using more dumb-type terminals using Windows Policies that do not allow installation of software or command line access. The routers or VLANS in the Library, Computer Lab and dorms should only allow Web access outside their segments. The scanning is coming all from the inside of the GIAC U. Consider using sign-in sheets for computer use to catch these people. A lot of the most serious activity, the 443 scans, possible myserver activity, and other events are all being initiated inside the network.

15. Aside from the ports we have discussed, http://isc.incidents.org/ lists these ports as additional ports that are currently popular for attacks. If possible, log or block them inside and out. They are ports145, 445, 1433

16. If you need to manage Windows security, here's a place to start: http://online.securityfocus.com/infocus/1629

17. Install IE 6 and all of its patches or upgrade MIME header with the following patch http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp. Upgrading to IE 6 will cure other ills as well.

APPENDIX I ---- SOURCE CODE

All of my scripts were basically variations on these two scripts that I wrote. The first one sorts, and the second one “uniquifies”. I batched them together. I used Developers Pad for Windows and processed the data on SUN, LINUX, WINNT 4 and Win2k machines.

```
#!/usr/bin/perl
#squash into one filehandle
#split to get description
#print unique items
```

```
$outputfilename = ">sorteddestports.txt";

@files = qw( alert1.txt alert2.txt alert3.txt alert4.txt alert5.txt) ;
open(OUT, $outputfilename) || die "could not open $!";

#qw means quoted words, so we dont' have to write ""

foreach $file (@files)

{


        print "file= $file \n";
        open(FH, $file) || die "could not open $!";



        while ( <FH> )
                {


        ($datefield, $descriptionfield, $addrfield)=split(/\[\*\*\]/,$_) ;
        @description[$x] = $descriptionfield;
        $x++;
        ($src_and_port, $dest_and_port) = split (/\-\>/, $addrfield);
        @source_and_port[$z1] = $src_and_port ;
        $z1++;

        @dest_and_port[$z2] = $dest_and_port ;
        $z2++;

        ($dest, $port) = split (/\:/, $dest_and_port);
        @destport[$z3] = $port ;
        $z3++;
#       print ( " port" . $port   ."\n") ;
#debugging

                }

        close(FH);
        print "close files...\n" ;
}

        print "count... \n";
        %seen={};
        @uniqueport = grep ( $seen{$_}++, @destport) ;
        foreach $item (@uniqueport)

                {

        print OUT ( $seen{$item} . "     occurrences for destination port   " . $item . "\n") ;

                }
```

```
print "done!!!!!!!! \n";



#!/usr/bin/perl
# parser_template.pl
# rips out unique destination and uniquifies


$datafilename = "sorteddestports.txt";
$outputfilename = ">sorted_unique_dest_ports.txt";


open(FH, $datafilename) || die "could not open $!";
open(OUT, $outputfilename) || die "could not open $!";

while ( <FH> )
        {


                @a[$x] = $_;
                $x++;


        }

close(FH);

print "uniquify....\n" ;

#sort the values
#descending

@sorted_a = sort { $b <=> $a } @a;


foreach $item (@sorted_a)
        {
  unless ($seen{$item}) {
     # if we get here, we have not seen it before
     $seen{$item} = 1;

print OUT ( $item, "\n") ;

  }
}
```

This three liner found all of the scanners' Ip addresses and put them in a list.

cat scan* > bigscansfile.txt
cut –d" " –f5 bigscansfile.txt > cutscans.txt
cut –d: -f1 cutscans.txt | uniq –c > scans.txt

APPENDIX II - BIBLIOGRAPHY

Legard, David "'Critical' FrontPage Security Flaw Found". IDG News Service. Thursday, September 26, 2002

Vamosi, Robert." Help & HowTo: Bugbear worm". October 2002.Zdnet UK News.
URL:http://news.zdnet.co.uk/story/0,,t281-s2123098,00.html

Kessler, Gary C. "Defenses Against Distributed Denial of Service Attacks". November 29, 2000

Gary Smith. GCIA Practical. June 2002
URL:http://www.giac.org/practical/Gary_Smith_GCIA.zip.

Ruth Kizlyk. GCIA Practical. January 2002
URL:http://www.giac.org/practical/Ruth_Kizlyk_GCIA.zip.

Voemel Christof. GCIA Practical. September 2001.
URL:http://www.giac.org/practical/Christof_Voemel_GCIA.txt

Royds Bill. GCIA Practical. September 2000.
URL:http://www.giac.org/practical/Bill_Royds.zip

Turkia Miika. GCIA Practical.
URL:http://www.giac.org/practical/Miika_Turkia_GCIA.html

The following sites were used extensively for research:

www.cert.org
www.microsoft.com
www.symantec.com
ics.incidents.org
www.whitehats.com
www.onlinesecurityfocus.com
cve.mitr.org
www.sans.org
www.digitaltrust.it/arachnids