

Forum:[Hack Forums](#)

Thread: [\[Tutorial\] ~ Intro to Packet Fingerprinting with Wireshark ~ Detecting NMap Scans!](#)

Goodies - October 12, 2013, 11:09 pm

I spent a little over 2 hours writing this, so I'd really appreciate feedback :) Thanks!

This is going to be a fairly long tutorial on Wireshark. This is if you have had some sort of experience with wireshark and nmap, and you should have an above-average understanding of some basic protocols.

We will be covering some major scans including:

TCP Scans (Transmission Control Protocol)

~ TCP SYN scan === nmap -sS

Explanation: Sends a SYN packet to an IP's ports

~ TCP Christmas Scan === nmap -sX

Explanation: Sends a TCP packet with the flags PSH, URG, and FIN set

~ TCP Null Scan === nmap -sN

Explanation: Sends NO TCP flags (easy to detect)

~ TCP FIN Scan === nmap -sF

Explanation: Sends only a TCP FIN header

We will ONLY be covering TCP as it is the most common. If you're interested in SCTP/UDP scans, feel free to PM me!

Make a New Color Scheme

We will be making our OWN color scheme. The default wireshark theme is never meant to be used. It's malleable and meant to be played with.

Here is how we do that:

[img]http://i.minus.com/ijAruJPOAJyOI.gif[/img]

I'm going to set some of the default colors to my own.

Protocol - Background Color / Text Color

TCP - Dark Green / White

UDP - Dark Blue / White

ARP - Orange / Black

I have also set HTTP to Dark Green / Yellow off-camera.

This should color QUITE A BIT of the packets green and blue since 90% of them run on either TCP or UDP.

Here's how:

[img]<http://i.minus.com/ik9XJOSlm4HNT.gif>[/img]

Run an NMap Scan and Look at Packets

1. Nmap SYN Scan (nmap -sS -v -n 192.168.1.1):

Alrightm so here is what the scan looks like in wireshark:

Let's look at the coloring rule and see why each is which. This is very easy.

Make sure the "Packet Details" panel is available. Click View -> Packet Details

The red is simple. It's just a RST (reset) or RST ACK packet

Alright, now click on the packet you'd like to see and expand the "Frame" tag. Then you can look at the colorize rule and see exactly why it's colored the way it is.

We can see that all of this grey is a TCP SYN packet. Hopefully you know that SYN requests are very common through a network. When initiating a TCP connection (every time you visit a website), a SYN request is sent. So what makes this SYN request different than a normal SYN request?

NMAP SYN scans have a TCP window size of:

- 1024
- 2048
- 3072
- 4096

while normal SYN window sizes are usually much larger and vary constantly. So this is our key! We can also note that the FIN header is always on.

So, let's write a rule. Here are some things you should note:

&& = and. Ex: tcp && udp = that will show something that has both TCP and UDP protocols

|| = or. Ex: tcp || udp = this will show anything that is either TCP OR UDP

! = not. Ex: !tcp = this will show everything that is NOT TCP protocol

That's pretty much it!

So we know that it is TCP so let's just make sure it's TCP:

Code tcp

Alright. So we also know that the TCP FIN header is also on. So let's write that as well:

Code tcp && tcp.flags.fin ==1

Alright, this may be just a little confusing if you're not a coder. Basically, we want to see if the window size is any of the aforementioned window sizes. If any of them match (they can't all match)

then this will definitively prove a SYN scan without much chance for false positive. We are going to set this aside with () and will use || inside them. Let's see:

```
Code tcp && tcp.flags.fin ==1 && (tcp.window_size==1024 || tcp.window_size==2048 || tcp.window_size==3072 || tcp.window_size==4096)
```

Here I show a real SYN request (grey) and a NMap SYN request (orange).

[img]http://i.minus.com/ibgdhP8pXd4DrT.gif[/img]

There we go! We have finished our first scan!!! Now we can detect a very basic TCP SYN Scan!!! Hoorah!

2. Christmas Scan (nmap -sX -v -n 192.168.1.1):

Alright, let's see the packet as we scan it. Open any packet you find and look at the packet details :)

What do we notice? Flag size = 0x29 in hex in EVERY PACKET

PSH, URG, FIN set in EVERY PACKET

It runs on TCP

Alright, It's go ahead and write a simple rule based on what we've noticed:

It's TCP only, right? Yes.

Code tcp

Now lets make sure the flag size is 0x29

Code tcp && tcp.flags==0x29

Now, lastly, we need to specify the headers that are set:

Code tcp && tcp.flags==0x29 && tcp.flags.fin==1 && tcp.flags.push==1 && tcp.flags.urg==1

And that's our rule! That will detect a Xmas scan!

[img]http://i.minus.com/ibpjEgncCyz3C4.gif[/img]

3. TCP Null Scan (nmap -sN -v -n 192.168.1.1):

This is a very obvious one to detect...

What do you notice?

TCP Flags is 0. There are NONE SET

Winddow size is the same as all the others (1024, 2048, 3072, 4096)

Again, let's write the rule.

Code tcp

Code tcp && tcp.flags==0x00 && (tcp.window_size==1024 || tcp.window_size==2048 ||

tcp.window_size==3072 || tcp.window_size==4096)

Now you can follow the last 2 tutorials to add your own color-coded scheme :)

This one I want you to do yourself!

First write what you notice. Once you're done, open the spoiler and see if you get the same things I have:

Spoiler [\(Click to View\)](#)

I noticed:Flags = 0x001
Only the FIN bit is set
Runs on TCP
Windows size (1024, 2048, 3072, 4096)

Okay, once you've done that, try and write a simple rule!

Spoiler ([Click to View](#))

Here is my final result

```
Code tcp.flags==0x01 && tcp.flags.fin==1 && (tcp.window_size==1024 || tcp.window_size==2048 ||  
tcp.window_size==3072 || tcp.window_size==4096)
```

Hopefully you've got this too ;D If not, you should re-read the tutorial. I hope you guys enjoyed this!

Tutorial by Goodies. I Hope You Enjoyed!

~ Goodies ?

DrDinosaur - October 13, 2013, 12:09 am

Nice tutorial Goodies. Hopefully many users can apply these techniques.

T0ucaNSaM - October 13, 2013, 5:51 am

Nice didnt know about the default color setting!

PinkPanther - October 13, 2013, 11:12 am

Wireshark is indeed a very versatile tool.

xSpectrex - October 13, 2013, 2:41 pm

Nice tutorial man, looks like you put a lot of work into it :)
Much appreciated.

Goodies - October 13, 2013, 2:53 pm

(Oct 13, 2013 02:41 PM)xSpectrex Wrote: [\[CITE\]](#) Nice tutorial man, looks like you put a lot of work into it :)

Much appreciated.

Thank you very much :) I spent a very long time doing this. Many of the GIFs I had to record multiple times xD

stdio - October 14, 2013, 9:37 pm

You put a lot of effort to it so thank you :) and very HQ thread
i however feel, you should also put a little motivation so people could actually know what its all bout.
thanks again for this great work.
Regards

Cybertooth - October 14, 2013, 10:56 pm

Thank you very much for an outstanding Intro.

weedy69 - November 6, 2013, 1:21 am

Thanks a lot for this great tutorial, really nice and detailed. ?

SIM-5561 - November 6, 2013, 7:32 am

Nice tut. Effective use of gifs :) Repped.

Letoll - November 6, 2013, 9:25 am

Forgot the -sV scan.

Nice tutorial on using wireshark though. Definitely a tool more people need to become familiar with, but have little idea on how to use it to its potential :).

Goodies - November 6, 2013, 2:15 pm

(Nov 6, 2013 10:25 AM)Letoll Wrote: [\[CITE\]](#) Forgot the -sV scan.

Nice tutorial on using wireshark though. Definitely a tool more people need to become familiar with, but have little idea on how to use it to its potential :).

Oh you're right...

Well, -sV is an ADDON for a TCP or UDP scan, but yeah I could show the difference between those packets ;)

~ Goodies ?

Goodies - November 14, 2013, 11:38 pm

I'm going to this. I hope people become interested in this kind of stuff ;)

~ Goodies ?

Goodies - March 2, 2014, 2:46 pm

Boomping for wireshark :) Anyone interested in this?

Mad76e - March 2, 2014, 2:51 pm

Hell yeah! This brings back some fine memories! Good job Goodies!

Zapu - March 3, 2014, 2:55 pm

Personally, I'd rather just use Snort with port scanning rules, but, we already know IDSes can be circumvented with some pretty nifty ways.

Nice tutorial, Goodies.

rwj2010 - March 4, 2014, 9:39 am

Nice Tutorial...been looking for something like this. Thanks.

HellaCrow - March 25, 2014, 4:42 pm

Great tutorial and well done. I will be reading all of your threads on networking. :)

mrtwothree - April 6, 2014, 8:17 pm

Great Tutorial boss. One of my PCs was infected with some persistent rat/trojan so this help alot. Im wondering if you know more filters for detecting IRC commands because Im sure that machine has a botnet infection. Either way thanks for the knowledge, it will be put to use.

_Tech - April 7, 2014, 1:53 am

Thank you nice detailed tut. Liking the use of the gifs. :)

handcastle - April 13, 2014, 4:51 pm

Another great tutorial, bookmarked!

Asp1r3-At0m - April 16, 2014, 10:12 pm

Sorry for reviving this old thread, but this is great information, I always think about how you could detect nmap with wireshark and then I found this, thank OP for this useful information you shared :D
I will read more about wireshark, nmap and tcp/ip protocol

Goodies - April 16, 2014, 10:16 pm

Hey bud! Thanks for reading! I'll actually have an ebook coming out that will be well over 100 pages that will be an introduction to pentesting and networking :) It won't be ready for likely several weeks or months, but you may like it!

Dorrance - April 16, 2014, 11:26 pm

Dam Goodies so much people can learn from you on a daily.

Goodies - April 17, 2014, 12:08 am

(Apr 16, 2014 11:26 PM)Mr. Unpleased Wrote: [\[CITE\]](#) Dam Goodies so much people can learn from you on a daily.

haha that's my goal! I know I'm far from professional, but if I know something interesting, why the hell would I want to keep it to myself? Whatever. I'm just glad people are enjoying it!

etchasako - April 28, 2014, 7:24 pm

This will really help me a lot! Thank you very much!

Rackzen - April 29, 2014, 2:28 pm

This guy rocks .. i simply appreciate all your tutorials and believe man , i pay a lot of attention on your tutorials , always HQ !
Good job 1 more time !

Computer Forensics - May 5, 2014, 8:35 pm

Very nice tutorial keep up the good work!

T-rom - June 3, 2014, 3:55 am

This is very high quality, I enjoyed the read. Hopefully gunna help some new people out :)

POLY MATH - June 3, 2014, 9:44 am

Another Great Tutorial from Goodies. Please keep up the good work. A suggestion try to set sections to keep it organized.

FluxHax - June 4, 2014, 5:46 pm

Thanks for the work. I appreciate the effort too. I will follow the tutorial.

EDIT:

Sorry if is asking too much. Can you write the hex colors, the "gifs" are terrible and I like the combination.

Thank you ?

ced.feludas - July 19, 2014, 11:39 am

Very very handy and detailed tut. Thanks mate.

53CUR1TY - July 19, 2014, 11:14 pm

Appreciate the tutorial, keep it up!

Ibizaganger98 - September 3, 2014, 2:08 am

Learned a lot of this. Thanks man, great tut!

freemium - August 20, 2015, 12:08 pm

just got wireshark and commView for wifi, now I can play with them, good tutorials:)

freemium - August 21, 2015, 12:35 pm

I saved all your tutorials for offline use, I hope you don't mind...

I tryed with linux and it worked...when I entered usb...was a password prompt, I dont know how I did it, but I used plop boot manager to run backtrack, and I was in...I could explore the HDD... thanks:) you're the best:)where's the like button ?

UnKnOwN333 - August 21, 2015, 1:10 pm

(Oct 12, 2013 11:09 PM)Goodies Wrote: [\[CITE\]](#)

I spent a little over 2 hours writing this, so I'd really apprciate feedback :) Thanks!

This is going to be a fairly long tutorial on Wireshark. This is if you have had some sort of experience with wireshark and nmap, and you should have an above-average understanding of some basic protocols.

We will be covering some major scans including:

TCP Scans (Transmission Control Protocol)

~ TCP SYN scan === nmap -sS

Explanation: Sends a SYN packet to an IP's ports

~ TCP Christmas Scan === nmap -sX

Explanation: Sends a TCP packet with the flags PSH, URG, and FIN set

~ TCP Null Scan === nmap -sN

Explanation: Sends NO TCP flags (easy to detect)

~ TCP FIN Scan === nmap -sF

Explanation: Sends only a TCP FIN header

We will ONLY be covering TCP as it is the most common. If you're interested in SCTP/UDP scans, feel free to PM me!

Make a New Color Scheme

We will be making our OWN color scheme. The default wireshark theme is never meant to be used. It's malleable and meant to be played with.

Here is how we do that:

[img]<http://i.minus.com/ijAruJPOAJyOI.gif>[/img]

I'm going to set some of the default colors to my own.

Protocol - Background Color / Text Color

TCP - Dark Green / White

UDP - Dark Blue / White

ARP - Orange / Black

I have also set HTTP to Dark Green / Yellow off-camera.

This should color QUITE A BIT of the packets green and blue since 90% of them run on either TCP or UDP.

Here's how:

[img]<http://i.minus.com/ik9XJOSlm4HNT.gif>[/img]

Run an NMap Scan and Look at Packets

1. Nmap SYN Scan (nmap -sS -v -n 192.168.1.1):

Alrightm so here is what the scan looks like in wireshark:

Let's look at the coloring rule and see why each is which. This is very easy.

Make sure the "Packet Details" panel is available. Click View -> Packet Details

The red is simple. It's just a RST (reset) or RST ACK packet

Alright, now click on the packet you'd like to see and expand the "Frame" tag. Then you can look at the colorize rule and see exactly why it's colored the way it is.

We can see that all of this grey is a TCP SYN packet. Hopefully you know that SYN requests are very common through a network. When initiating a TCP connection (every time you visit a website), a SYN request is sent. So what makes this SYN request different than a normal SYN request?

NMAP SYN scans have a TCP window size of:

- 1024

- 2048

- 3072

- 4096

while normal SYN window sizes are usually much larger and vary constantly. So this is our key! We can also note that the FIN header is always on.

So, let's write a rule. Here are some things you should note:

&& = and. Ex: tcp && udp = that will show something that has both TCP and UDP protocols

|| = or. Ex: tcp || udp = this will show anything that is either TCP OR UDP

! = not. Ex: !tcp = this will show everything that is NOT TCP protocol

That's pretty much it!

So we know that it is TCP so let's just make sure it's TCP:

Code tcp

Alright. So we also know that the TCP FIN header is also on. So let's write that as well:

Code tcp && tcp.flags.fin ==1

Alright, this may be just a little confusing if you're not a coder. Basically, we want to see if the window size is any of the aforementioned window sizes. If any of them match (they can't all match) then this will definitively prove a SYN scan without much chance for false positive. We are going to set this aside with () and will use || inside them. Let's see:

Code tcp && tcp.flags.fin ==1 && (tcp.window_size==1024 || tcp.window_size==2048 || tcp.window_size==3072 || tcp.window_size==4096)

Here I show a real SYN request (grey) and a NMap SYN request (orange).

[img]<http://i.minus.com/ibgdhP8pXd4DrT.gif>[/img]

There we go! We have finished our first scan!!! Now we can detect a very basic TCP SYN Scan!!!
Hoorah!

2. Christmas Scan (nmap -sX -v -n 192.168.1.1):

Alright, let's see the packet as we scan it. Open any packet you find and look at the packet details :)

What do we notice? Flag size = 0x29 in hex in EVERY PACKET

PSH, URG, FIN set in EVERY PACKET

It runs on TCP

Alright, It's go ahead and write a simple rule based on what we've noticed:

It's TCP only, right? Yes.

Code tcp

Now lets make sure the flag size is 0x29

Code tcp && tcp.flags==0x29

Now, lastly, we need to specify the headers that are set:

Code tcp && tcp.flags==0x29 && tcp.flags.fin==1 && tcp.flags.push==1 && tcp.flags.urg==1

And that's our rule! That will detect a Xmas scan!

[img]<http://i.minus.com/ibpjEgncCyz3C4.gif>[/img]

3. TCP Null Scan (nmap -sN -v -n 192.168.1.1):

This is a very obvious one to detect...

What do you notice?

TCP Flags is 0. There are NONE SET

Window size is the same as all the others (1024, 2048, 3072, 4096)

Again, let's write the rule.

Code tcp

Code tcp && tcp.flags==0x00 && (tcp.window_size==1024 || tcp.window_size==2048 || tcp.window_size==3072 || tcp.window_size==4096)

Now you can follow the last 2 tutorials to add your own color-coded scheme :)

This one I want you to do yourself!

First write what you notice. Once you're done, open the spoiler and see if you get the same things I have:

Spoiler ([Click to View](#))

I noticed:Flags = 0x001

Only the FIN bit is set

Runs on TCP

Windows size (1024, 2048, 3072, 4096)

Okay, once you've done that, try and write a simple rule!

Spoiler ([Click to View](#))

Here is my final result

Code `tcp.flags==0x01 && tcp.flags.fin==1 && (tcp.window_size==1024 || tcp.window_size==2048 || tcp.window_size==3072 || tcp.window_size==4096)`

Hopefully you've got this too ;D If not, you should re-read the tutorial. I hope you guys enjoyed this!

Tutorial by Goodies. I Hope You Enjoyed!

~ Goodies ?

The Best HQ Member Pentesting Expert !!!!

Goodies - August 21, 2015, 1:14 pm

(Aug 21, 2015 01:10 PM)UnKnOwN333 Wrote: ([CITE](#)) The Best HQ Member Pentesting Expert !!!!
There's no need to quote the entire thread.

Also, this is very old and not at all advanced, but I appreciate the sentiment.

UnKnOwN333 - August 21, 2015, 1:19 pm

(Aug 21, 2015 01:14 PM)Goodies Wrote: ([CITE](#)) There's no need to quote the entire thread.

Also, this is very old and not at all advanced, but I appreciate the sentiment.

bro have you skype ? can i add you in skype :) whats your username or you add me my name :
Razor-Unknown

Goodies - August 21, 2015, 1:35 pm

(Aug 21, 2015 01:19 PM)UnKnOwN333 Wrote: ([CITE](#)) bro have you skype ? can i add you in skype :)
whats your username or you add me my name : Razor-Unknown

<https://keybase.io/goodies>

Skype is goodieshq

radialhat - October 20, 2015, 3:14 pm

Amazing tutorial Goodies, I've put my time to learn all you explained here during this evening...
Cheers mate

SIM-1858092 - January 22, 2016, 10:39 pm

This is interesting! Thanks Goodies -- HQ user as always.

Jazer - *January 23, 2016, 3:08 am*

This will help a lot of users thanks for contributing

Goodies - *January 23, 2016, 1:22 pm*

Wow, I was doing this shit 3 years ago?

I've come a long way :)

hackeaziq - *October 17, 2017, 12:46 pm*

This is so cool tutorial, i hope i can go further in my career. Hope can see many kind of tutorial like this (Y)