# 1.4t: transcript

Hi, welcome to this practical on WireShark. So we're going to be using WireShark to analyze the network frames and packets that traverse our network.

WireShark is a software tool, a protocol analyser or packet sniffer application.

Used for network troubleshooting analysis, software and protocol development, and education.

As data streams travel back and forth over the network, the sniffer captures each protocol data unit and can decode and analyze its contents. First thing you're going to have to do is, obviously, go to the WireShark homepage.

This here I've cheated already...

I've already gone to it and you're going to download WireShark so you can see big download button. We now get different options now – stable release current one is 3.2.1. - obviously if you're doing this little bit later on in the year at some point this will have changed, so download the most current stable release.

There's lots of options here for various operating systems. I'm going to be using a Windows installer 64bit. You will obviously choose the correct one for your own operating system.

So, all we do is click. Do I wish to run, save or cancel it?

I'm going to save it and this will now start downloading. 57.3MB

I'm not going to talk through that while it downloads, so I'm just going to pause the video for a moment and then I'll restart once it's downloaded. I suggest you do the same as well.

Okay as you can see here, I've downloaded the version of WireShark that I require.

So, double-click to install...

We're still going to run it...

and you can see here - "do you want to allow this device..." usual thing on Windows, so I'm going to say yes.

So, welcome to WireShark - standard set up, click 'next'.

So, there's three parts to this - various parts of the license, the actual license and other acceptable

licenses. So again, I'm going to agree. Obviously, I suggest you always read through the licensing conditions.

Leave everything at the default, click 'next'. Where do you want it? I'm just going to leave in it the start menu, I'm not bothered about the desktop icon but again it's entirely up to you.

Don't uncheck the associated file extensions, leave all of those alone for now.

Okay. there's the location, install Npcap.

You'll need to leave this alone okay so just, again leave it in. It's actually used as part of WireShark to

capture the various packets as they pass to and fro by our network card.

Again, I'm going to pause the video while it does it's install rather than you sit there and watch the screen rolling by.

During the install as well you'll see the Npcap copyright agreement pop-up. It's by NMap.org. Click agree to that and let your installation continue. There are certain options to choose supporting raw 802.11 traffic and monitor mode for wireless adaptors. I'm going to check that because I like

using it.

Install Npcap in WinPcap API compatible mode. I'll install that as well but you don't have to, it depends on your own systems. And it will install.

It will carry on. So, again I'm going to pause the video until I need to tell you something else.

Okay, so once you've installed WireShark and you actually run the application, you'll see a screen similar to this as well. I use the word similar because obviously this is dependent on your machine, of what network card you have installed and what you're using.

Now in my case I'm obviously on a laptop, so I'm using Wi-Fi. So, if you see there's a little graph there at the side of it. It is doing no more than just telling me that that's the active card that's bringing packets to and from my machine. If there were others on here, I could select those as well but I'm using Wi-Fi.

Now I'm going to go up and I'm going to start a capture.

Now, one of the things you going to notice is there's going to be a great flood of information very, very quickly, so if I go to capture and I hit start...

It's starting to pull all information.

Now this gives you an idea of how quickly packets are actually being passed to and fro to my machine. All these packets means different things.

Now just while this is all going on, we'll just have a look at a few different things. We can see here we've got the time. So, it's showing our time that they've arrived or left us, they're showing us the source, and they're showing me the destination, an IP.

These are IPv4 addresses over here. Here they're also showing me what the protocol is. You can see TCP popping up. There's also a bunch of other things popping up here as well.

It shows me the length and it gives me some form of information as well. This obviously is very, very difficult for us to read to do anything with at all. So, what we need to do is either stop it and have a look, or we can actually apply filters.

If I go to this red light here and we'll just stop where we are for a moment, so there we go, we have a pause for a moment, and we can actually see there's various forms of communication.

If I scroll down towards the last packets, there are some TCP going on here, some conversations going backward and forward. These are acknowledgments of various sequences - this is called windowing. I mention that in one of the earliest sets of slide presentations where we talk about this idea of synchronisation.

ACK, which is all windowing - this idea that our host and our receiver will acknowledge things for TCP. This is why TCP is known as reliable.

So, if I scroll up you can see there is an awful lot of information which you would expect. These are packets they've come down our stack and they've been broadcast out on to the wire. You can actually see here. Duplicate ACKs because obviously at times things get missed and they have to be resent and so on. So, lots and lots and lots and lots of information. It really is at this point far too much for us to try and digest, and take in one go.

However, this is where WireShark is really, really very useful. Up here, we have a filter.

Now, if I apply a filter of ICMP, you can see the bar goes green which means it understands what ICMP is, it gives me an option between version six and just ICMP, We'll stick with just ICMP. So that filter is now set.

I'm just hit the return key there and it's cleared it out because I've got no ICMP packets.

Now, ICMP is used for various forms of communications within a system.

What we're actually going to do is the most simplest of ICMP is what's called a ping test.

So, the first thing you need to do is we need to launch a command prompt.

If I drag this over onto this window so you can see it. So, there's my DOS prompt. I'm sitting there - ipconfig.

Again, lots of information but the one we're really interested in at the moment is our IPv4 address. There it is: 192.168.1.115

It also shows you what my default gateway is. How do I get out of this network? So, if we've actually filtered on ICMP in our Wireshark, if I now ping from this machine and I ping:

192.168.1.254 - my gateway,

you'll see I start getting replies. Okay? Now WireShark didn't do anything because I'd paused it. Let's do it again now. I'll move the command prompt out of the way and we'll start another capture. So it says 'do I want to save' or 'continue without saving' so we're going to continue without saving because we're not interested in all those packets we had a few moments ago.

So you can see it's sat there very, very quietly, nothing is actually happening...

until I run the ping again.

So, in my DOS prompt, my command prompt here, I can actually see I'm getting a reply from 192.168.1.254. It's telling me it's time in milliseconds and it's time to live is 64. We'll talk about that in future modules where we start digging more into networks, but we can actually see the reply coming to us. However, if we look closely in WireShark, which is still recording but there are no more pings, there are no more ICMP messages.

So, we'll stop recording but we can actually see: there is our request, there's our reply. So if we click on our request: 192.168.1.115 my source, my computer. Destination 192.168.1.254 the gateway I've decided to ping. Protocol, ICMP, and the length, and so on. Now if we look at this middle part of the screen here...

we can actually start to dig into that individual packet. That individual packet I've got highlighted there; I can now have a look at.

So, if I go down to Internet Control Message Protocol and expand that out, we can actually see it is a particular type. It is type 8, and type 8 is a ping request. We can actually start to identify types of packets that are traveling around our system.

At the next layer up - remember we talked about stacks... Internet protocol version, source, and destination, and there is our header length - 20 bytes. So, it actually gives us lots of information surrounding what is inside our header.

We can go to another layer as well. We talked about the Mac layer. The Mac layer inside a frame will actually have the Mac address of my local station and where it's going to, source and destination again. So, we can actually seek a lot of information.

If we moved to the next packet which is a reply, again all of this already expanded out, but we see down here Internet Control Message Protocol type zero, a ping  reply.

So, it's a different type and that's how we start to identify the different types as they move back and to. We can take this slightly further - how about we change our filter to TCP.

Now we mention TCP earlier, didn't we? Again, let's start our capture, continue without saving...

and we're waiting for TCP information.

We can now see there is lots of TCP. Now,TCP carries lots of various information around, so we can actually see all forms that it's actually using the TCP protocol for.

If we want as well, if I could stop that and change our filter, this time to UDP, start recording, hit enter, so I'm capturing UDP this time. We start to see various other things coming up. SSDP stands for Simple Service Discovery Protocol.

It's a network protocol based on the internet protocol suite for advertisement and discovery of network services.

It's just another protocol that happens to be around and it can piggyback on UDP so again we can see lots of other things kicking around. There are obviously other things, other filters that we've not discussed, other protocols.

I'll give you an example of one called DNS: domain name services. If I hit enter and apply that and start our capture.

Again, continue without saving.

I've got some very quiet DNS traffic. Let me clear the filter and just see what's going on with that.

Start a capture, continue and save with no filter at the moment.

Everything is starting to come through again.

Lots and lots of packets. Okay. let me apply a filter

DNS...

I'll just force it to actually try and resolve a name

just to see what is going on. So if by ping www.cisco.com

There we go - DNS. So, it's resolving the name www.cisco.com

it is turning it into an IP address, that is what DNS does for us so I can actually force my machine to resolve names but it can't do it by itself. It actually goes out and does queries so we can actually see here that it is doing a standard query for www.cisco.com and it will eventually get responses, this is how we can actually then find various websites and servers across the Internet.

Okay, so your challenge for today, if you like, is to have a look around WireShark. Apply various filters and see what you can learn, see what you can understand. There's obviously an awful lot out there.

Don't forget. If you are following a networking cyber route with us, you'll certainly be coming back to more of this at some point in time, so don't worry too much about it now. It's really a case to give you that feel and understanding that when we get to virtualisation, a lot of it is built upon our current hard standards if you like.

So more and more to learn as we go along. So, enjoy WireShark, get interested in it and have a play about.

Thanks