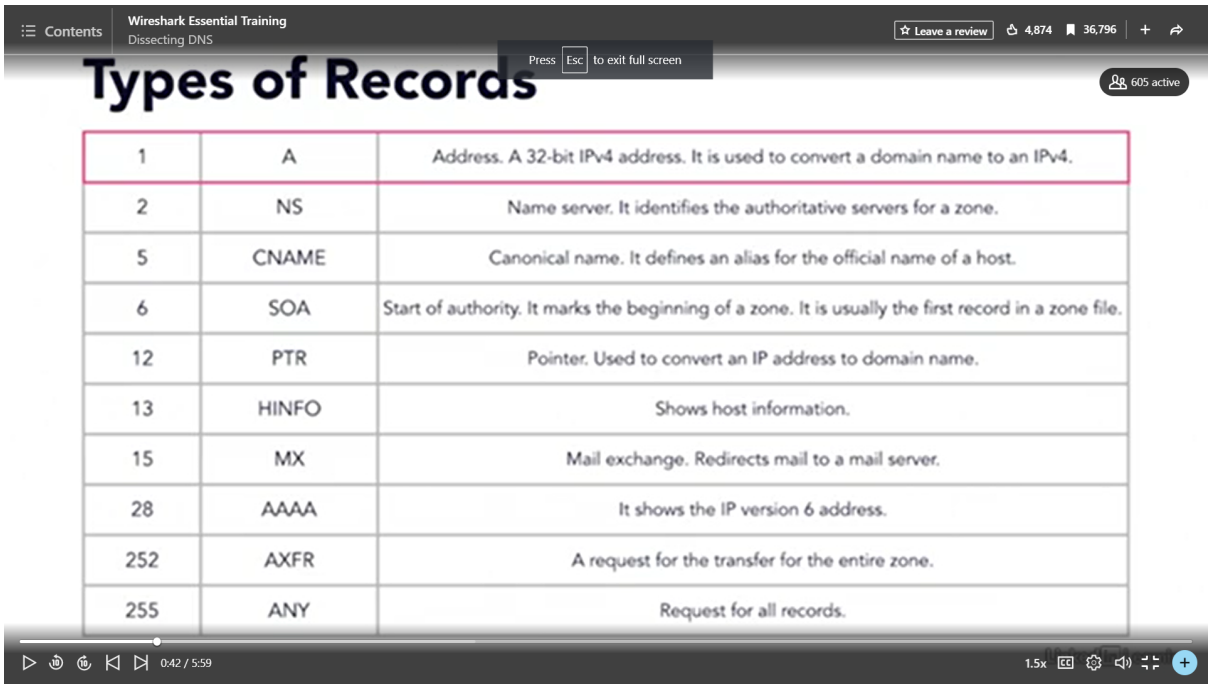
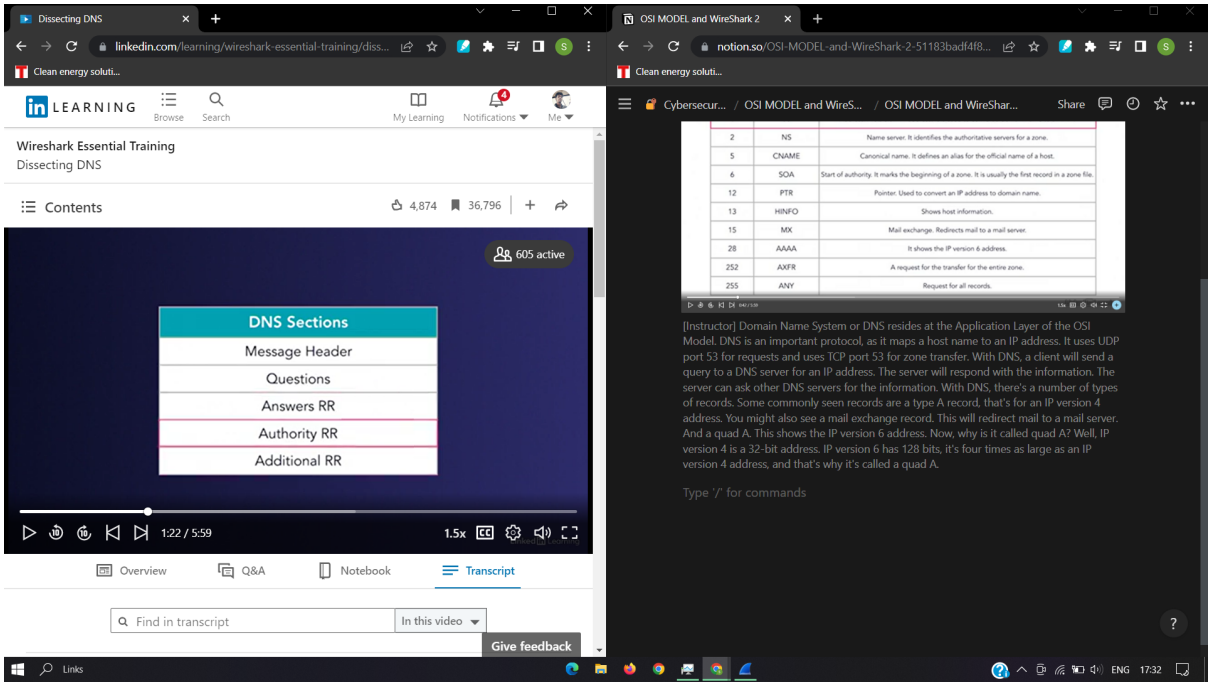
OSI MODEL and WireShark 2

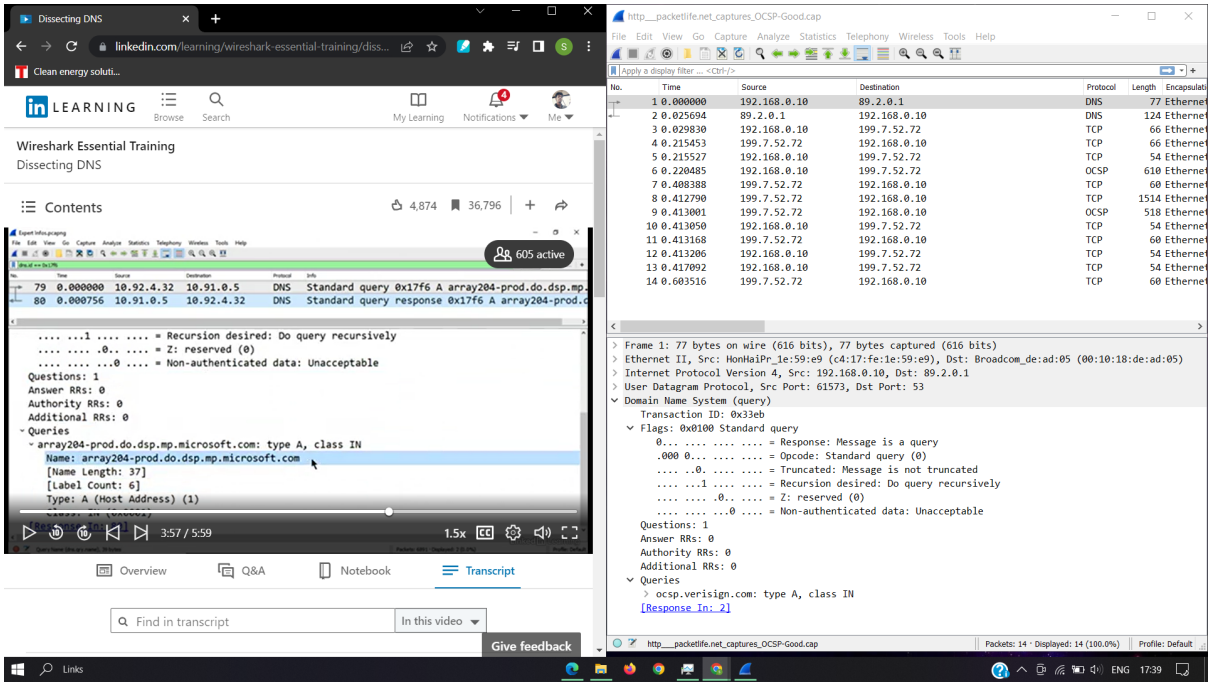
Dissecting DNS:



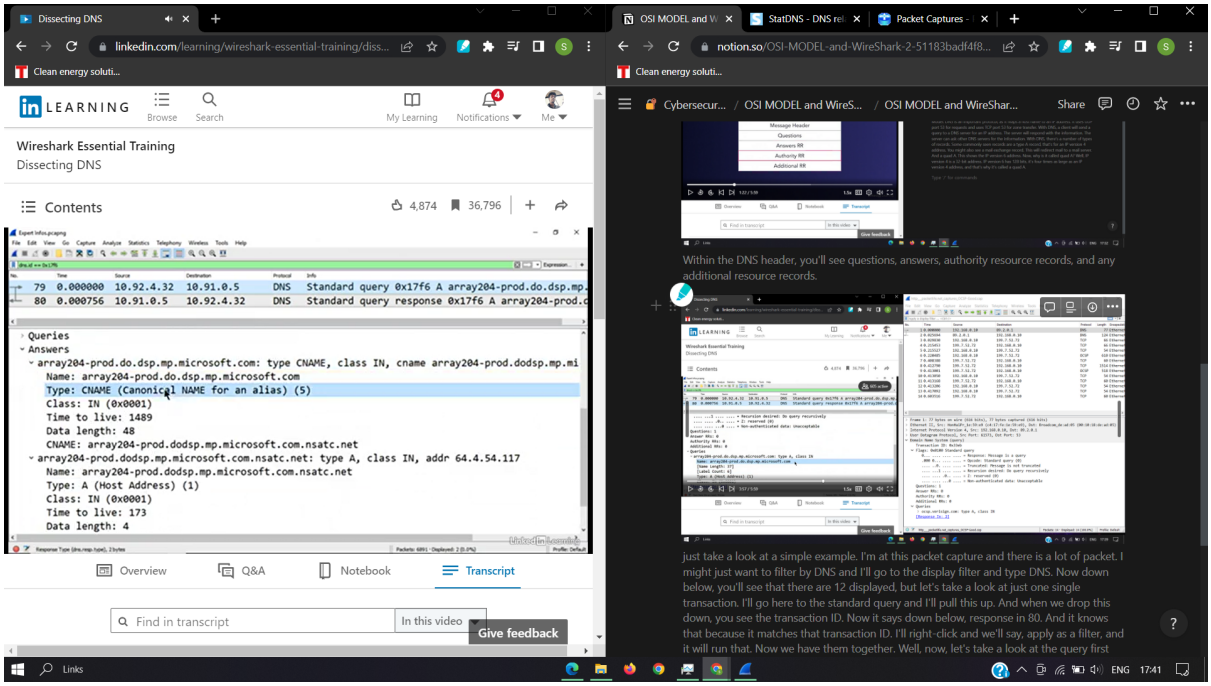
[Instructor] Domain Name System or DNS resides at the Application Layer of the OSI Model. DNS is an important protocol, as it maps a host name to an IP address. It uses UDP port 53 for requests and uses TCP port 53 for zone transfer. With DNS, a client will send a query to a DNS server for an IP address. The server will respond with the information. The server can ask other DNS servers for the information. With DNS, there's a number of types of records. Some commonly seen records are a type A record, that's for an IP version 4 address. You might also see a mail exchange record. This will redirect mail to a mail server. And a quad A. This shows the IP version 6 address. Now, why is it called quad A? Well, IP version 4 is a 32-bit address. IP version 6 has 128 bits, it's four times as large as an IP version 4 address, and that's why it's called a quad A.



Within the DNS header, you'll see questions, answers, authority resource records, and any additional resource records.



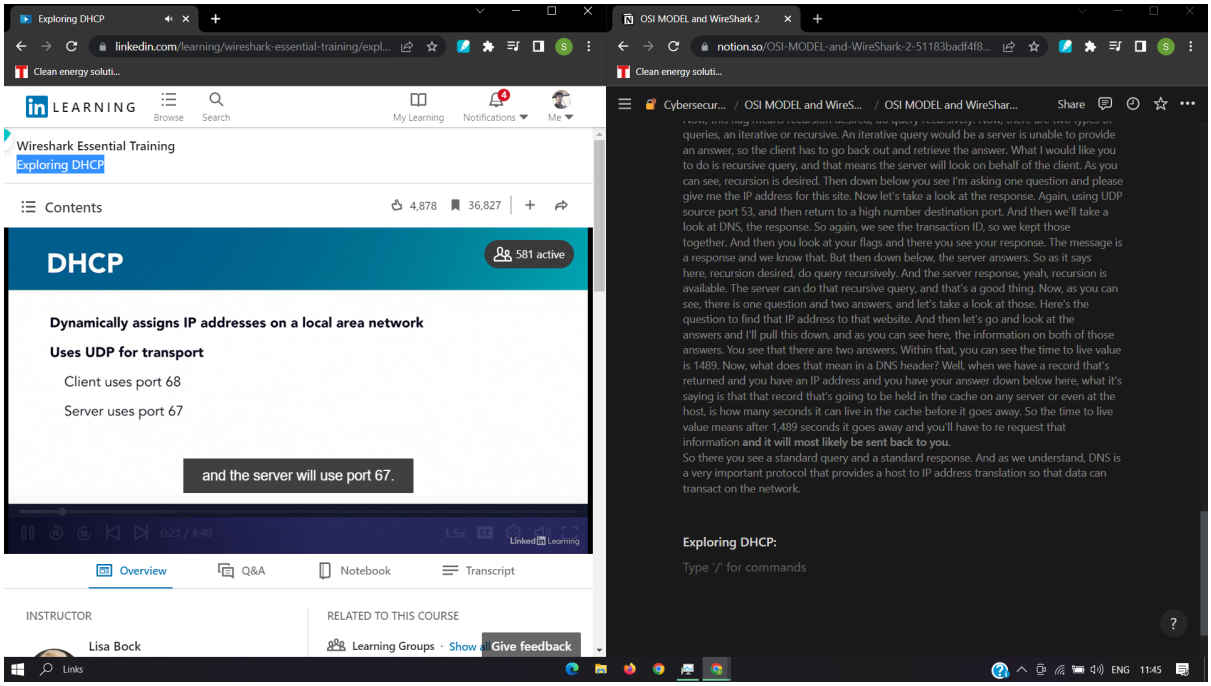
just take a look at a simple example. I'm at this packet capture and there is a lot of packet. I might just want to filter by DNS and I'll go to the display filter and type DNS. Now down below, you'll see that there are 12 displayed, but let's take a look at just one single transaction. I'll go here to the standard query and I'll pull this up. And when we drop this down, you see the transaction ID. Now it says down below, response in 80. And it knows that because it matches that transaction ID. I'll right-click and we'll say, apply as a filter, and it will run that. Now we have them together. Well, now, let's take a look at the query first from the client. So using UDP, you see that it offers a high number port, and then it goes to the server on port 53 for a query. Now, we'll drop this down and here we see the query. And first we see the transaction ID. And then we drop this down and these are the flags that are set. Now, first we see that the first one is a zero, that means that message is a query. Now, this flag means recursion desired, do query recursively. Now, there are two types of queries, an iterative or recursive. An iterative query would be a server is unable to provide an answer, so the client has to go back out and retrieve the answer. What I would like you to do is recursive query, and that means the server will look on behalf of the client. As you can see, recursion is desired. Then down below you see I'm asking one question **and please give me the IP address for this site.**



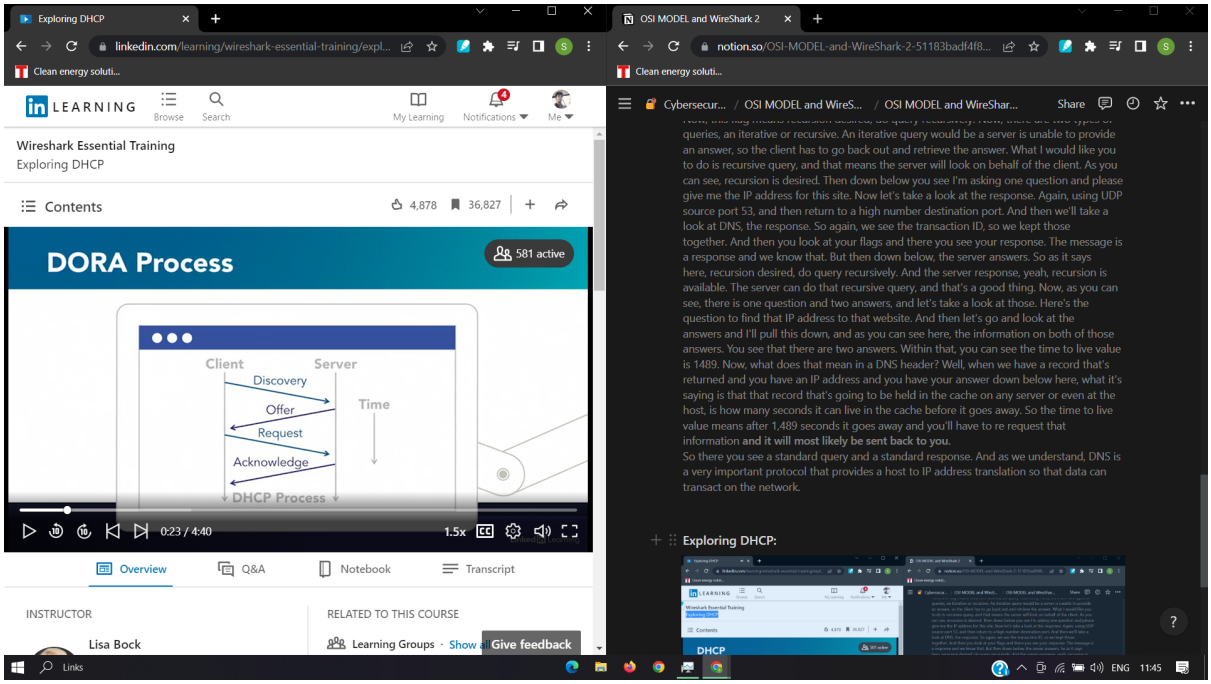
Now, this flag means recursion desired, do query recursively. Now, there are two types of queries, an iterative or recursive. An iterative query would be a server is unable to provide an answer, so the client has to go back out and retrieve the answer. What I would like you to do is recursive query, and that means the server will look on behalf of the client. As you can see, recursion is desired. Then down below you see I'm asking one question and please give me the IP address for this site. Now let's take a look at the response. Again, using UDP source port 53, and then return to a high number destination port. And then we'll take a look at DNS, the response. So again, we see the transaction ID, so we kept those together. And then you look at your flags and there you see your response. The message is a response and we know that. But then down below, the server answers. So as it says here, recursion desired, do query recursively. And the server response, yeah, recursion is available. The server can do that recursive query, and that's a good thing. Now, as you can see, there is one question and two answers, and let's take a look at those. Here's the question to find that IP address to that website. And then let's go and look at the answers and I'll pull this down, and as you can see here, the information on both of those answers. You see that there are two answers. Within that, you can see the time to live value is 1489. Now, what does that mean in a DNS header? Well, when we have a record that's returned and you have an IP address and you have your answer down below here, what it's saying is that that record that's going to be held in the cache on any server or even at the host, is how many seconds it can live in the cache before it goes away. So the time to live value means after 1,489 seconds it goes away and you'll have to re request that information **and it will most likely be sent back to you.**

So there you see a standard query and a standard response. And as we understand, DNS is a very important protocol that provides a host to IP address translation so that data can transact on the network.

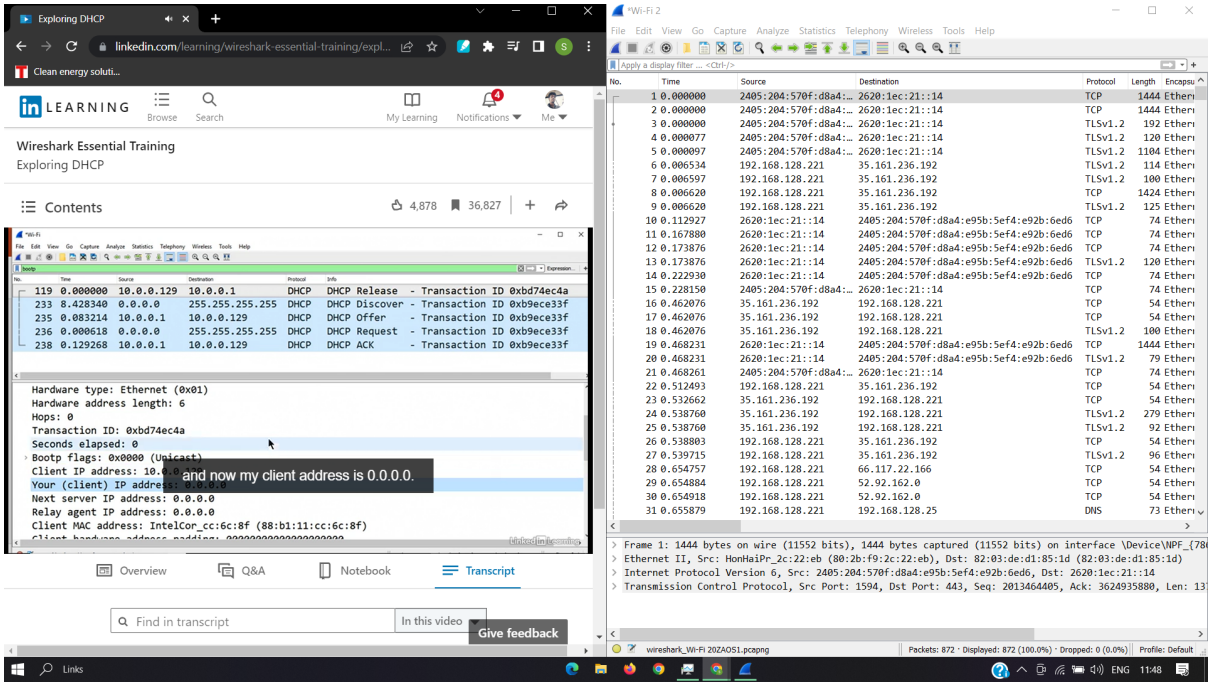
Exploring DHCP:

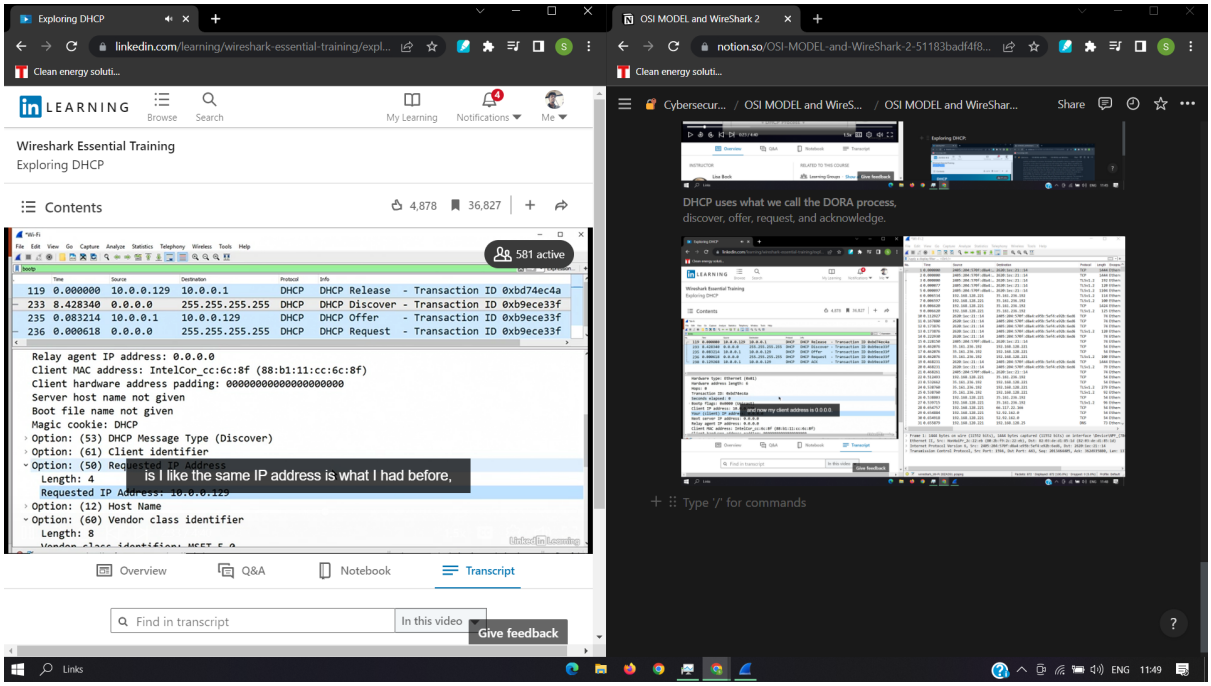


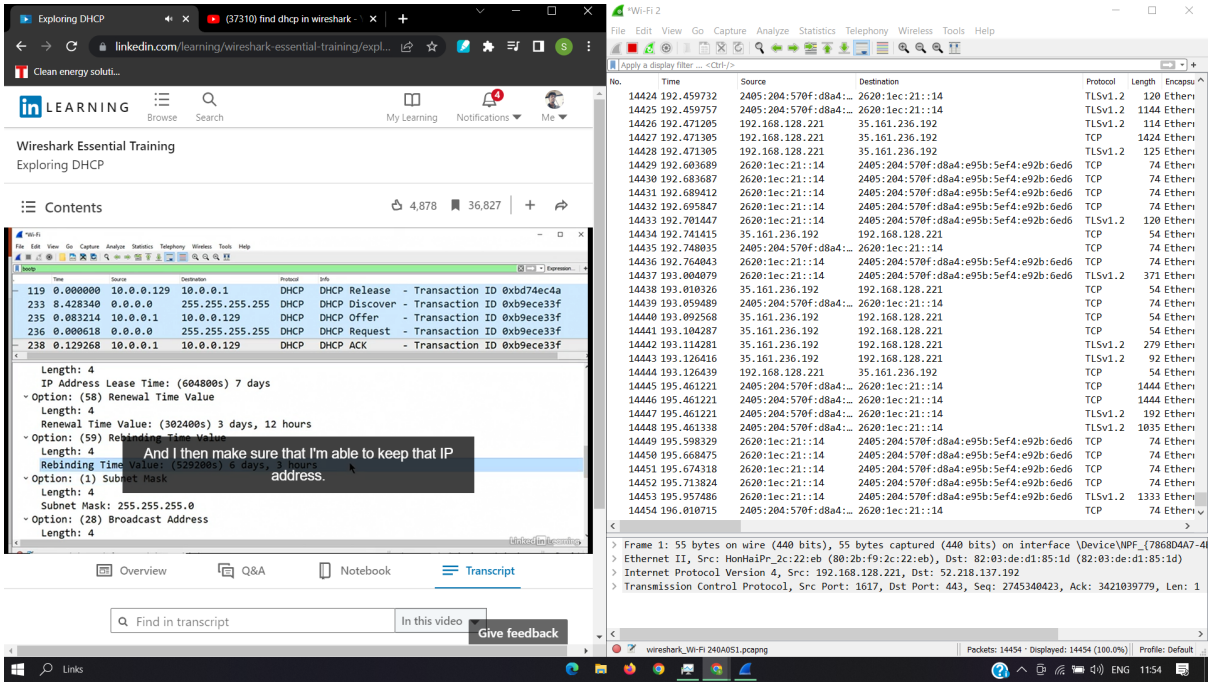
[Narrator] Dynamic host configuration protocol works at the application layer of the OSI model. DHCP dynamically assigns IP addresses on a local area network. DHCP uses UDP for transport, and the client will use port 68, and the server will use port 67.



DHCP uses what we call the DORA process,
discover, offer, request, and acknowledge.


Exploring DHCP
linkedin.com/learning/wireshark-essential-training/expl...
LEARNING
Wireshark Essential Training
Exploring DHCP
Contents
4,878
36,827
and now my client address is 0.0.0.0.
Overview
Q&A
Notebook
Transcript
Find in transcript
In this video
Give feedback
*Wi-Fi 2
Frame 1: 1444 bytes on wire (11552 bits), 1444 bytes captured (11552 bits) on interface \Device\NPF_{78
Ethernet II, Src: HonHaiPr_2c:22:eb (80:2b:f9:2c:22:eb), Dst: 82:03:de:d1:85:1d (82:03:de:d1:85:1d)
Internet Protocol Version 6, Src: 2405:204:570f:d8a4:e95b:5ef4:e92b:6ed6, Dst: 2620:1ec:21::14
Transmission Control Protocol, Src Port: 1594, Dst Port: 443, Seq: 2013464405, Ack: 3624935880, Len: 13
wireshark_Wi-Fi 20ZAOS1.pcapng
Packets: 872 · Displayed: 872 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
ENG
11:48

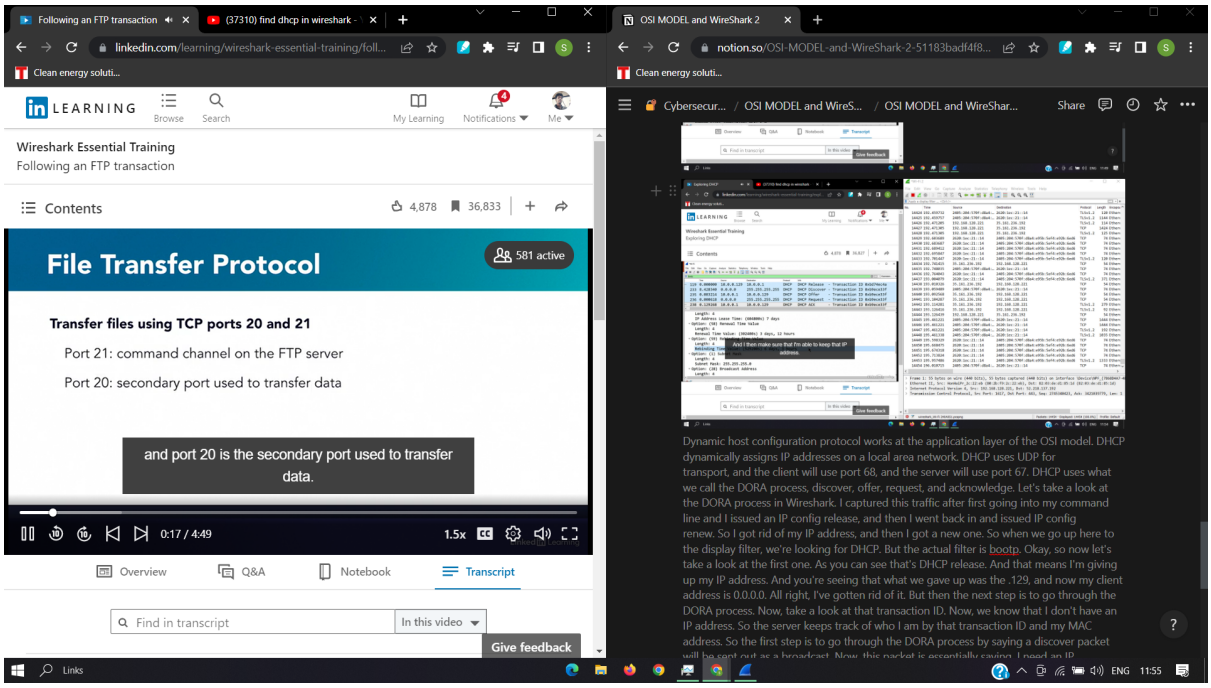

Exploring DHCP
linkedin.com/learning/wireshark-essential-training/expl...
LEARNING
Wireshark Essential Training
Exploring DHCP
Contents
4,878
36,827
581 active
is I like the same IP address is what I had before,
Overview
Q&A
Notebook
Transcript
Find in transcript
In this video
Give feedback
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f8...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share
DHCP uses what we call the DORA process,
discover, offer, request, and acknowledge.
Type '/' for commands
ENG
11:49

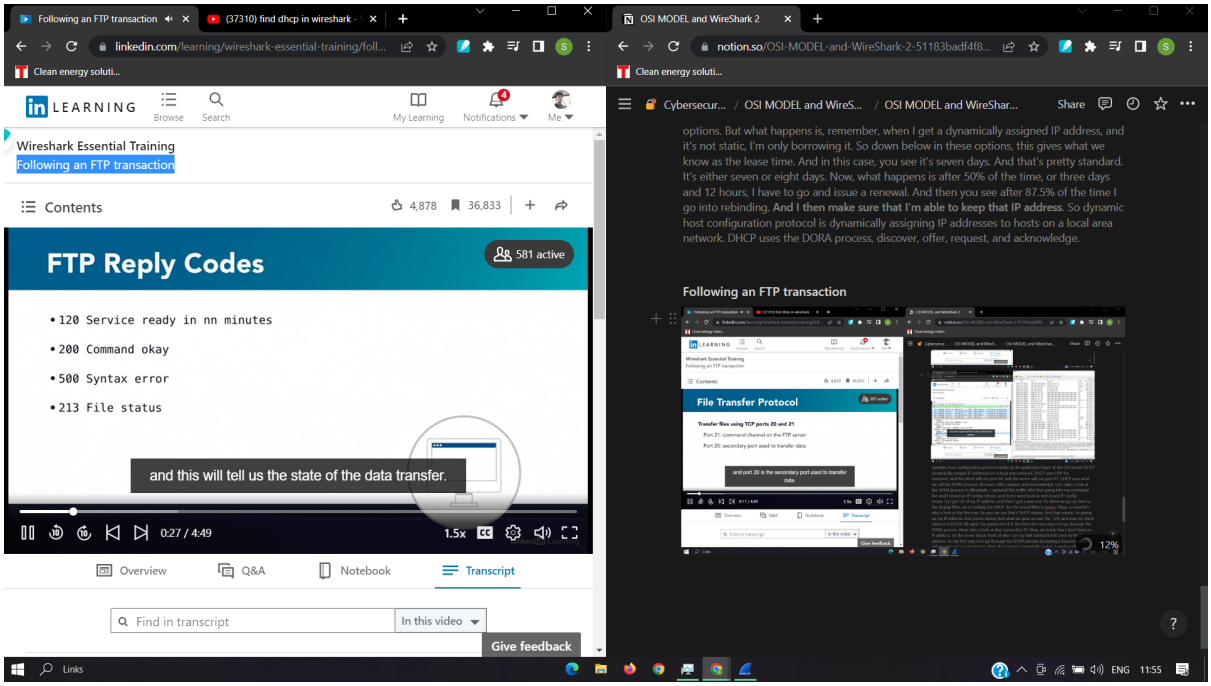


Dynamic host configuration protocol works at the application layer of the OSI model. DHCP dynamically assigns IP addresses on a local area network. DHCP uses UDP for transport, and the client will use port 68, and the server will use port 67. DHCP uses what we call the DORA process, discover, offer, request, and acknowledge. Let's take a look at the DORA process in Wireshark. I captured this traffic after first going into my command line and I issued an IP config release, and then I went back in and issued IP config renew. So I got rid of my IP address, and then I got a new one. So when we go up here to the display filter, we're looking for DHCP. But the actual filter is bootp. Okay, so now let's take a look at the first one. As you can see that's DHCP release. And that means I'm giving up my IP address. And you're seeing that what we gave up was the .129, and now my client address is 0.0.0.0. All right, I've gotten rid of it. But then the next step is to go through the DORA process. Now, take a look at that transaction ID. Now, we know that I don't have an IP address. So the server keeps track of who I am by that transaction ID and my MAC address. So the first step is to go through the DORA process by saying a discover packet will be sent out as a broadcast. Now, this packet is essentially saying, I need an IP address. Anybody? Does anybody have an IP address? Now, this goes out as a broadcast in hopes that a server will hear and respond. So as you can see that the message type is the boot request. I'll pull this up and I will tell you that there are a lot of different options with DHCP. But the one I want to show you with that discover packet is requested IP address. What I'm essentially saying in that original discover packet is I like the same IP address is what I had before, that .129. Now we see the offer. The DHCP offer comes from a server. Now, understand, on an enterprise network, you most likely will see two offers responding. So, it'll come from two different servers. Now the offer is saying, I have an IP address and you probably will like it. Why don't you try it, .129? It knows I want the same one. And it says, you know, no one's using it. Go ahead. Why don't you try it? The next step is for the client then to go back out and say, you know, yeah, I think I do want that one. And let's go to it. So I'm going to request it. And again, I want the requested IP address, and I would like that IP address. Could I have it? Well, sure. So the server responds with the final step in the DORA process by sending an acknowledgement. And now I'm sending my reply. And again, the same transaction ID. And here we say, your client IP address is .129. Now, if we go down, it of course gives the MAC address and then the options. But what happens is, remember, when I get a

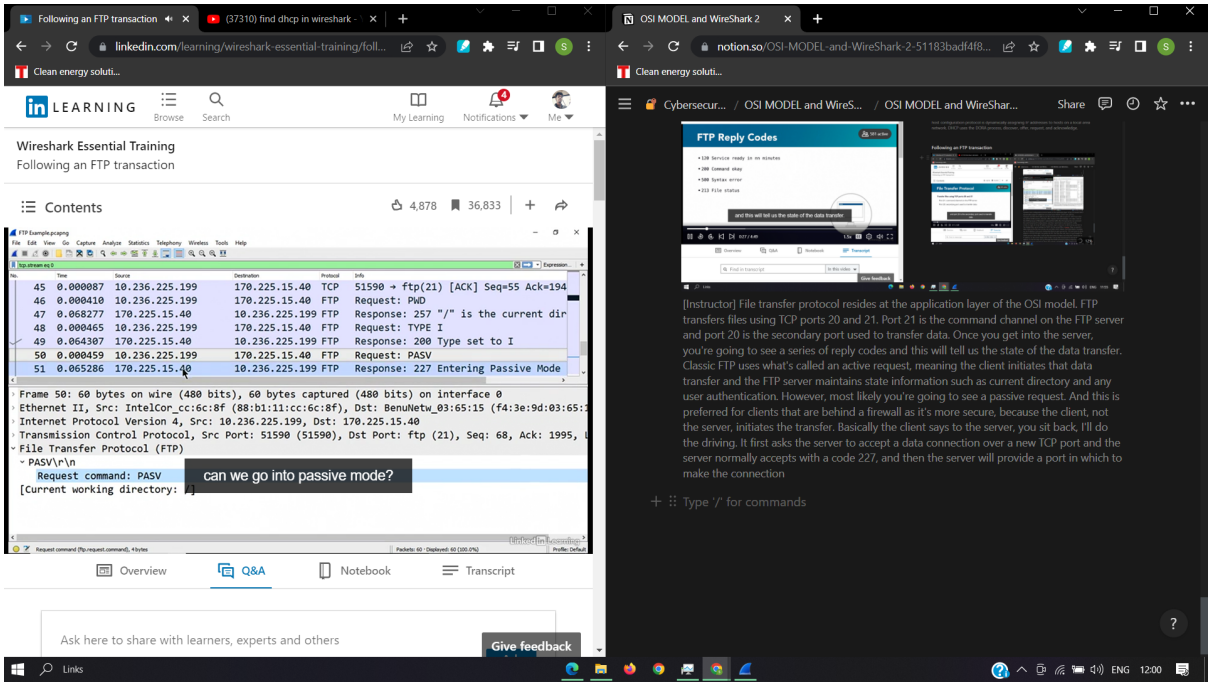
dynamically assigned IP address, and it's not static, I'm only borrowing it. So down below in these options, this gives what we know as the lease time. And in this case, you see it's seven days. And that's pretty standard. It's either seven or eight days. Now, what happens is after 50% of the time, or three days and 12 hours, I have to go and issue a renewal. And then you see after 87.5% of the time I go into rebinding. **And I then make sure that I'm able to keep that IP address.** So dynamic host configuration protocol is dynamically assigning IP addresses to hosts on a local area network. DHCP uses the DORA process, discover, offer, request, and acknowledge.

Following an FTP transaction

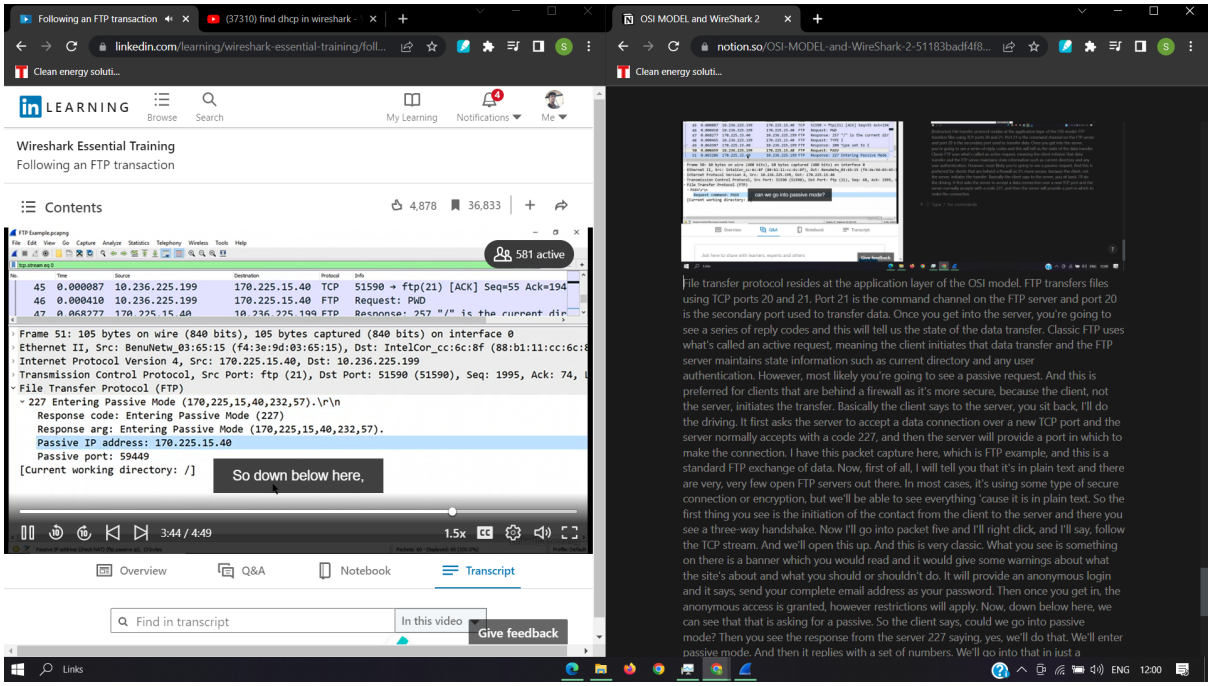




[Instructor] File transfer protocol resides at the application layer of the OSI model. FTP transfers files using TCP ports 20 and 21. Port 21 is the command channel on the FTP server and port 20 is the secondary port used to transfer data. Once you get into the server, you're going to see a series of reply codes and this will tell us the state of the data transfer. Classic FTP uses what's called an active request, meaning the client initiates that data transfer and the FTP server maintains state information such as current directory and any user authentication. However, most likely you're going to see a passive request. And this is preferred for clients that are behind a firewall as it's more secure, because the client, not the server, initiates the transfer. Basically the client says to the server, you sit back, I'll do the driving. It first asks the server to accept a data connection over a new TCP port and the server normally accepts with a code 227, and then the server will provide a port in which to make the connection

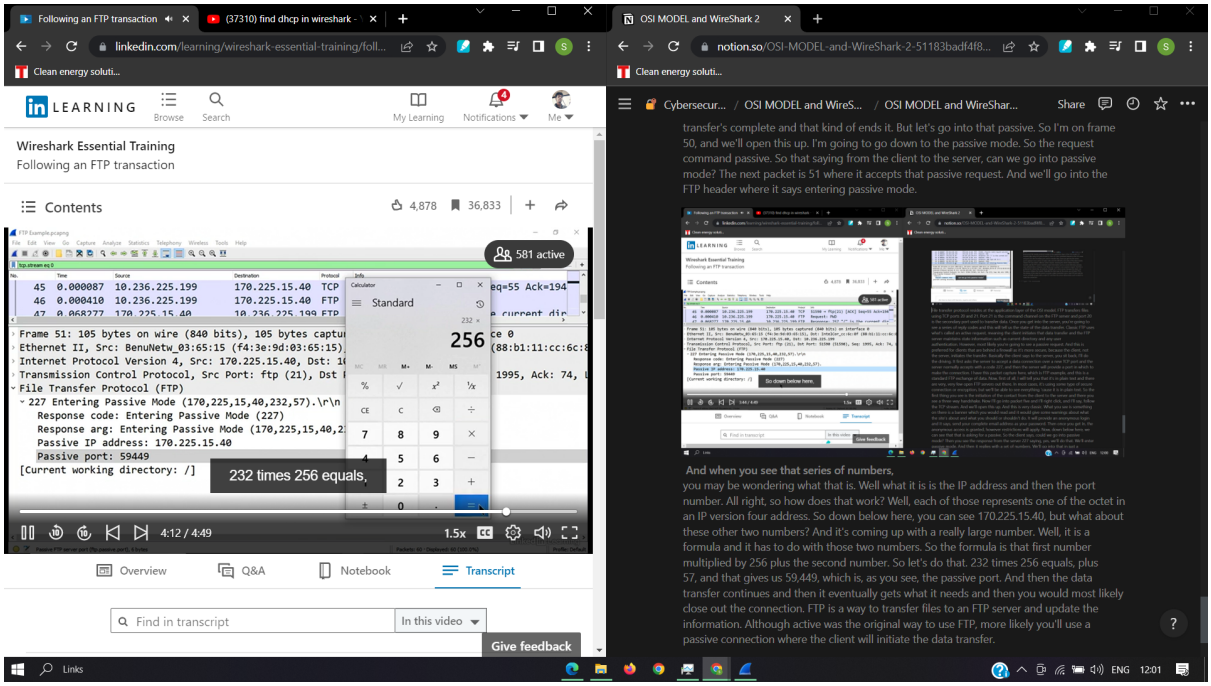


File transfer protocol resides at the application layer of the OSI model. FTP transfers files using TCP ports 20 and 21. Port 21 is the command channel on the FTP server and port 20 is the secondary port used to transfer data. Once you get into the server, you're going to see a series of reply codes and this will tell us the state of the data transfer. Classic FTP uses what's called an active request, meaning the client initiates that data transfer and the FTP server maintains state information such as current directory and any user authentication. However, most likely you're going to see a passive request. And this is preferred for clients that are behind a firewall as it's more secure, because the client, not the server, initiates the transfer. Basically the client says to the server, you sit back, I'll do the driving. It first asks the server to accept a data connection over a new TCP port and the server normally accepts with a code 227, and then the server will provide a port in which to make the connection. I have this packet capture here, which is FTP example, and this is a standard FTP exchange of data. Now, first of all, I will tell you that it's in plain text and there are very, very few open FTP servers out there. In most cases, it's using some type of secure connection or encryption, but we'll be able to see everything 'cause it is in plain text. So the first thing you see is the initiation of the contact from the client to the server and there you see a three-way handshake. Now I'll go into packet five and I'll right click, and I'll say, follow the TCP stream. And we'll open this up. And this is very classic. What you see is something on there is a banner which you would read and it would give some warnings about what the site's about and what you should or shouldn't do. It will provide an anonymous login and it says, send your complete email address as your password. Then once you get in, the anonymous access is granted, however restrictions will apply. Now, down below here, we can see that that is asking for a passive. So the client says, could we go into passive mode? Then you see the response from the server 227 saying, yes, we'll do that. We'll enter passive mode. And then it replies with a set of numbers. We'll go into that in just a second. Then you'll see the complete exchange and then at the end where it says the transfer's complete and that kind of ends it. But let's go into that passive. So I'm on frame 50, and we'll open this up. I'm going to go down to the passive mode. So the request command passive. So that saying from the client to the server, can we go into passive mode? The next packet is 51 where it accepts that passive request. And we'll go into the FTP header where it says entering passive mode.



And when you see that series of numbers,

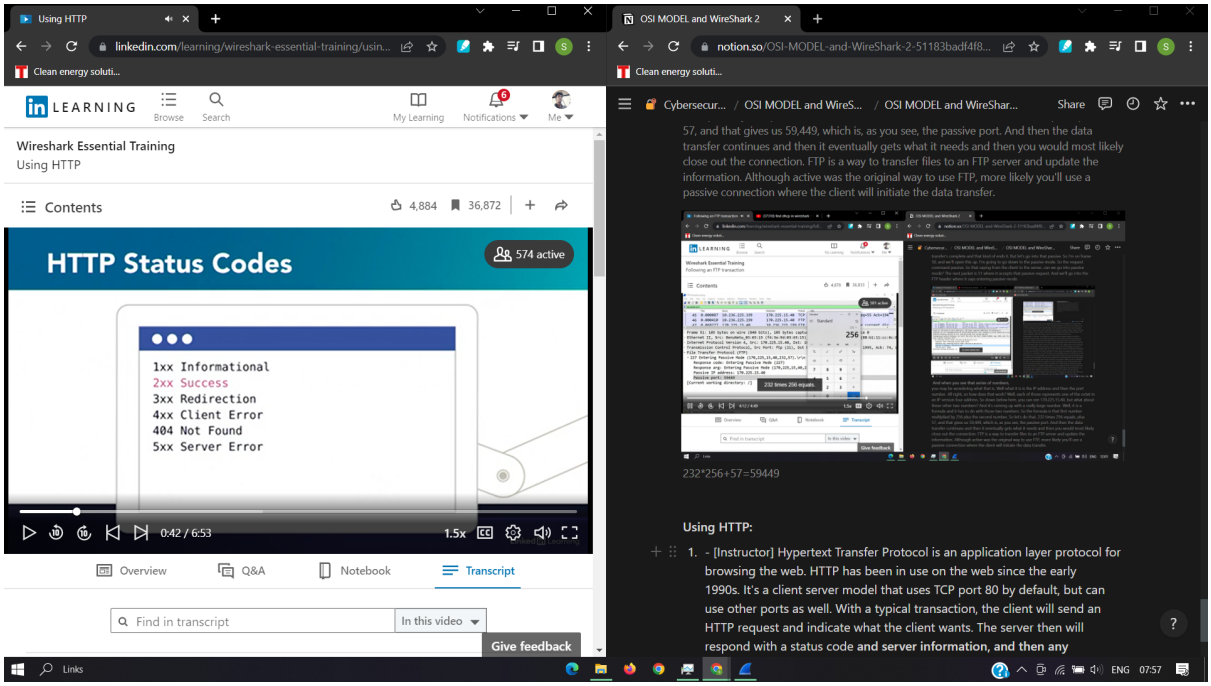
you may be wondering what that is. Well what it is is the IP address and then the port number. All right, so how does that work? Well, each of those represents one of the octet in an IP version four address. So down below here, you can see 170.225.15.40, but what about these other two numbers? And it's coming up with a really large number. Well, it is a formula and it has to do with those two numbers. So the formula is that first number multiplied by 256 plus the second number. So let's do that. 232 times 256 equals, plus 57, and that gives us 59,449, which is, as you see, the passive port. And then the data transfer continues and then it eventually gets what it needs and then you would most likely close out the connection. FTP is a way to transfer files to an FTP server and update the information. Although active was the original way to use FTP, more likely you'll use a passive connection where the client will initiate the data transfer.



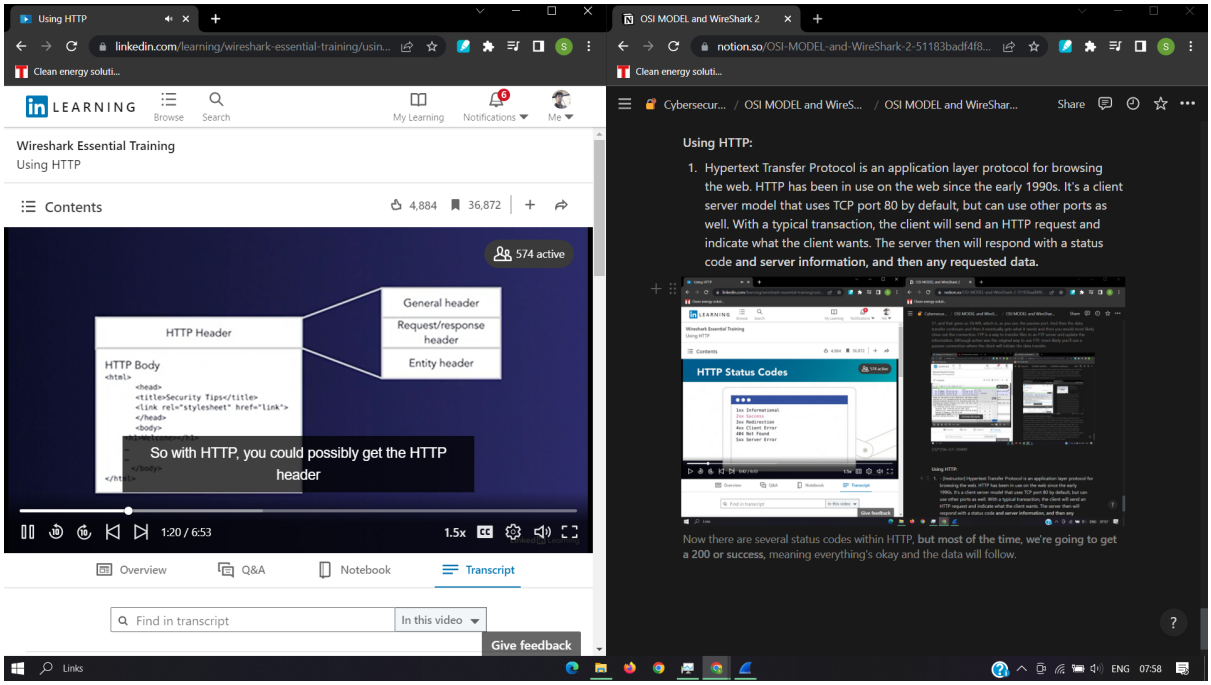
232*256+57=59449

Using HTTP:

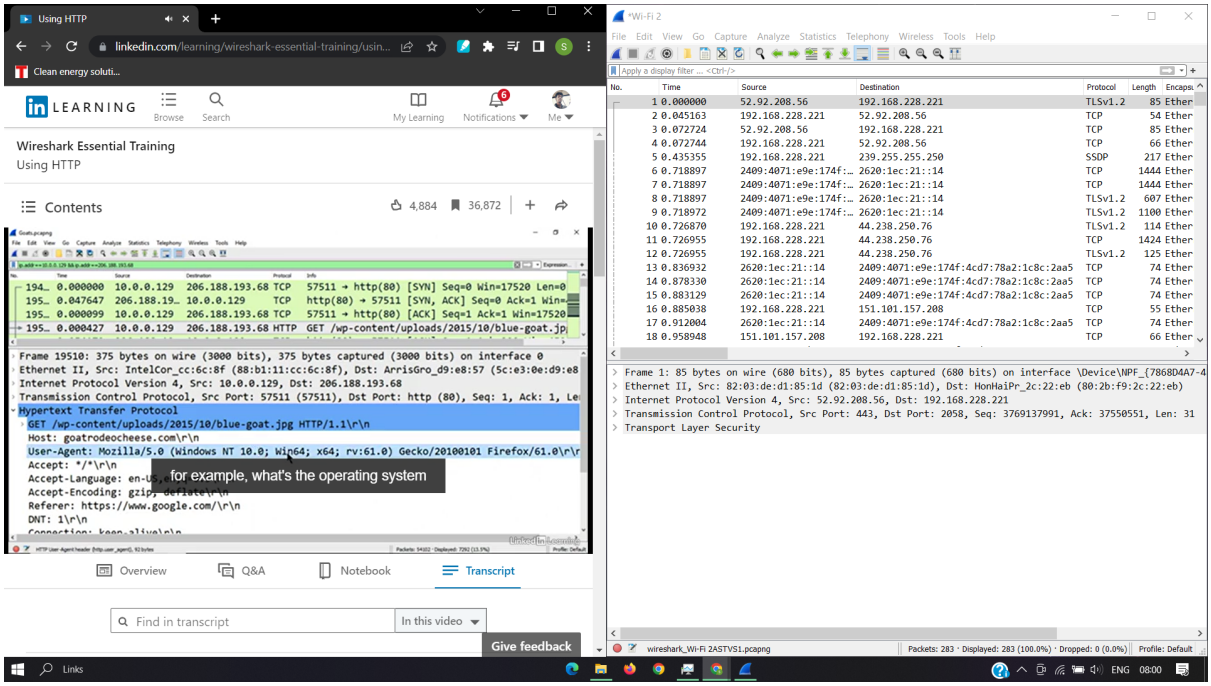
1. Hypertext Transfer Protocol is an application layer protocol for browsing the web. HTTP has been in use on the web since the early 1990s. It's a client server model that uses TCP port 80 by default, but can use other ports as well. With a typical transaction, the client will send an HTTP request and indicate what the client wants. The server then will respond with a status code **and server information, and then any requested data.**



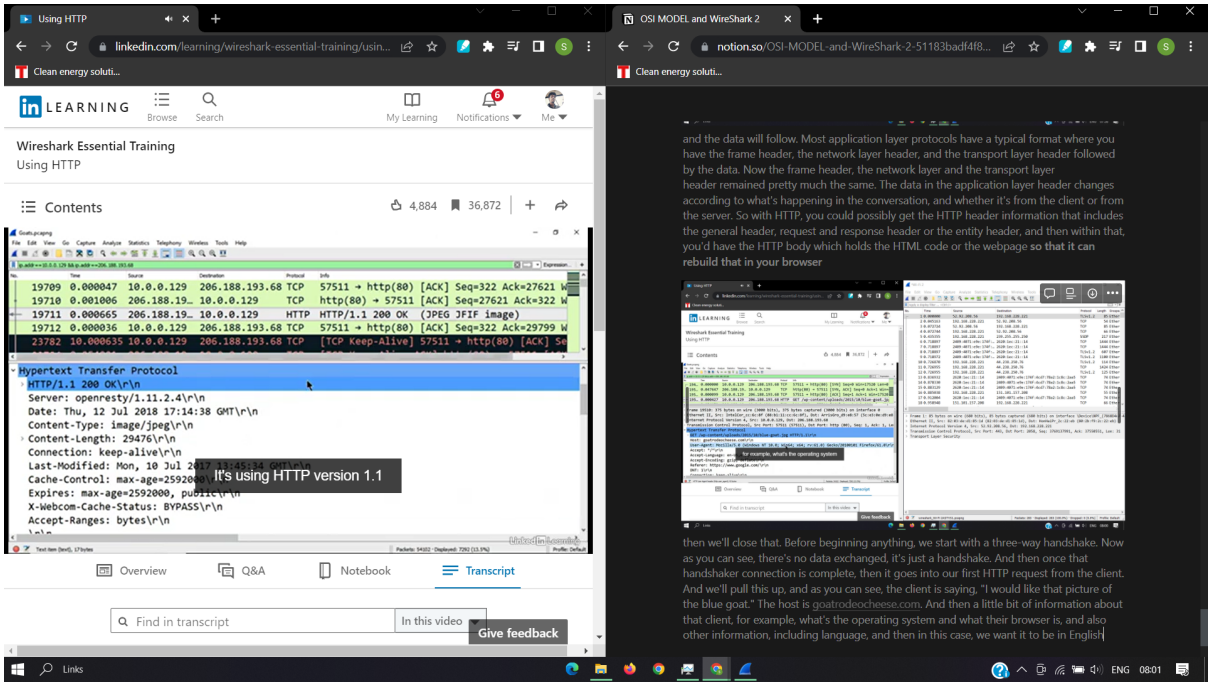
Now there are several status codes within HTTP, **but most of the time, we're going to get a 200 or success,** meaning everything's okay and the data will follow.



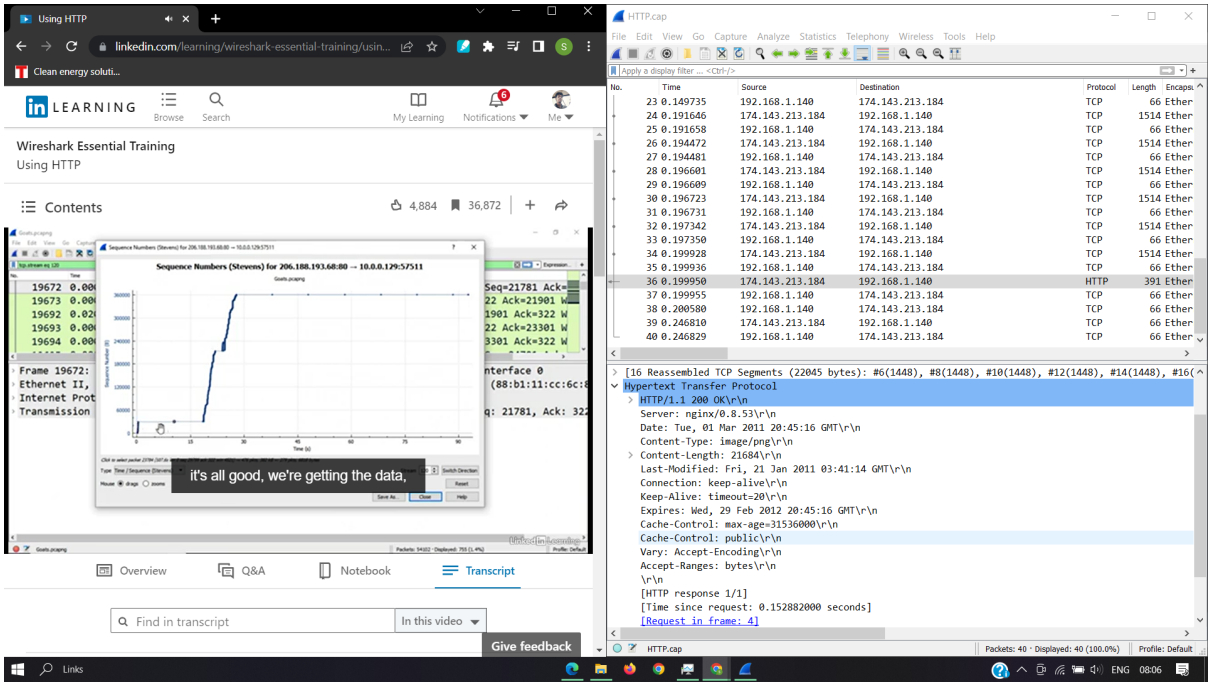
and the data will follow. Most application layer protocols have a typical format where you have the frame header, the network layer header, and the transport layer header followed by the data. Now the frame header, the network layer and the transport layer header remained pretty much the same. The data in the application layer header changes according to what's happening in the conversation, and whether it's from the client or from the server. So with HTTP, you could possibly get the HTTP header information that includes the general header, request and response header or the entity header, and then within that, you'd have the HTTP body which holds the HTML code or the webpage **so that it can rebuild that in your browser**



then we'll close that. Before beginning anything, we start with a three-way handshake. Now as you can see, there's no data exchanged, it's just a handshake. And then once that handshaker connection is complete, then it goes into our first HTTP request from the client. And we'll pull this up, and as you can see, the client is saying, "I would like that picture of the blue goat." The host is goatrodeocheese.com. And then a little bit of information about that client, for example, what's the operating system and what their browser is, and also other information, including language, and then in this case, we want it to be in English



and then in this case, we want it to be in English. Then down below, we now see the response from the server, and it's saying it's a 200 status code, which means it's okay. Everything is good and now I can start to give you what you need. It's using HTTP version 1.1 which is the most commonly used version, and the server is openresty, which is the third largest web server in use today behind a patchy and NGN IX. **It says, well, what the content is,**
and it's an image or JPEG. Another information about the server.

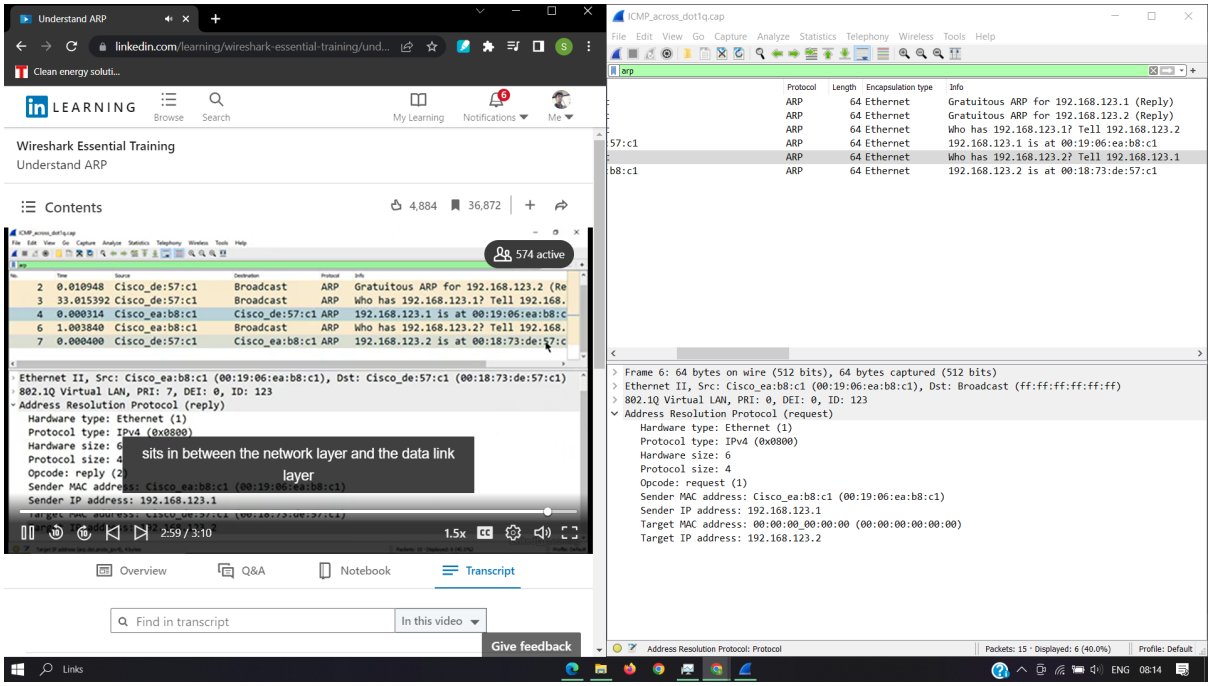


Hypertext Transfer Protocol is an application layer protocol for browsing the web. HTTP has been in use on the web since the early 1990s. It's a client server model that uses TCP port 80 by default, but can use other ports as well. With a typical transaction, the client will send an HTTP request and indicate what the client wants. The server then will respond with a status code and server information, and then any requested data. Now there are several status codes within HTTP, but most of the time, we're going to get a 200 or success, meaning everything's okay and the data will follow. Most application layer protocols have a typical format where you have the frame header, the network layer header, and the transport layer header followed by the data. Now the frame header, the network layer and the transport layer header remained pretty much the same. The data in the application layer header changes according to what's happening in the conversation, and whether it's from the client or from the server. So with HTTP, you could possibly get the HTTP header information that includes the general header, request and response header or the entity header, and then within that, you'd have the HTTP body which holds the HTML code or the webpage so that it can rebuild that in your browser. I've opened up this packet capture and we can take a look at some HTTP traffic. There's a lot of packets here, and one thing we can do is take a look at the conversations by going to statistics, and then conversations. Now once you open this, there's a lot of column headers. And what you could do, and in this case, we're going to sort by bytes so that we get the top talker. Now as you can see now, that highlighted area, we can do something with that, for example, I can apply a filter. I can do this, I can either select the entire conversation from A to B, that would be from the client to the server, from just the client to the server, or from just the server to the client. I'll select all of that and then that will run that filter, and then we'll close that. Before beginning anything, we start with a three-way handshake. Now as you can see, there's no data exchanged, it's just a handshake. And then once that handshaker connection is complete, then it goes into our first HTTP request from the client. And we'll pull this up, and as you can see, the client is saying, "I would like that picture of the blue goat." The host is goatrodeocheese.com. And then a little bit of information about that client, for example, what's the operating system and what their browser is, and also other information, including language, and then in this case, we want it to be in English. Then down below, we now see the response from the server, and it's saying it's a 200 status code, which means it's okay. Everything is good and now I

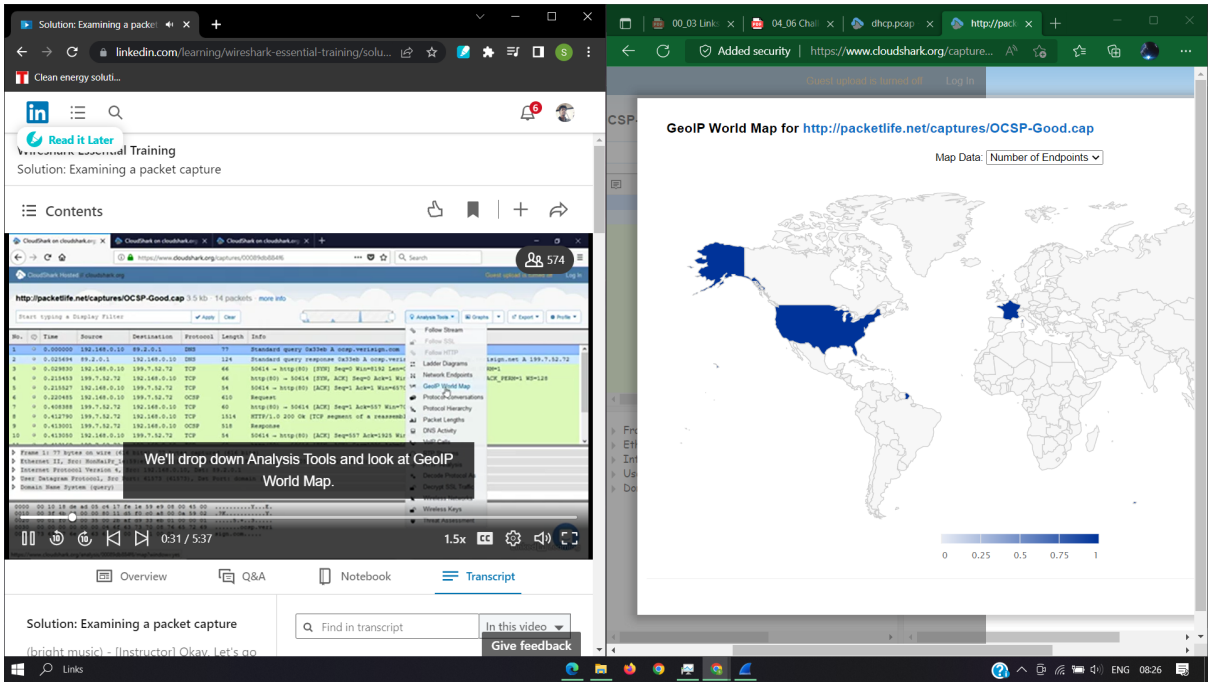
can start to give you what you need. It's using HTTP version 1.1 which is the most commonly used version, and the server is openresty, which is the third largest web server in use today behind a patchy and NGN IX. It says, well, what the content is, and it's an image or JPEG. Another information about the server. Now, what I can do is right click, and I can say, follow the TCP stream. Now, in this case, this shows all of the connections that are happening within that single stream, and there's more than one. And in fact, when you're retrieving data and objects off of a web page, you'll have more than one stream most of the time because you want to be able to retrieve those objects quickly and rebuild the web page in your browser quickly. Now as you can see, the red is from the client and the blue is from the server. And I'm going to scroll down and you'll see that there's a number of requests and responses, here's another one, get, and then the HTTP status report and the HTTP status code, and then the response. And there are a few of those within this single stream. All right, now close that, and then just one more thing we'll take a look at. When you're getting this data from the web server, which is very common, you see a number of areas that are black. And if this is going to happen consistently, you might want to look into it. One thing we can do is go to statistics and then run a stream graph, all use the TCP stream graph time sequence or Stevens graph, and we'll open this up. Now as we see, that's the server to the client and we can switch directions just so you can see, but we want it to be from the server to the client. All right, so when we follow this, this is showing this line's going up, it's all good, we're getting the data, and then, we're not. So, this is showing nothing's happening here. As you see, that's a ten second delay. And I'll move this down, and as you can see, it's a keep alive. Nothing's happening, it's just still waiting. And we go here, and there's a seven second delay. Then we have the data is coming in, it's good, oh, a little hiccup here and up, and now, we have a nine second delay, then a 10 second delay, 10 second delay, and more ten second delays. Well, really nothing's happening. So I'll close this and you can take a look at this and see that it's pretty much stalled. But at some point at the end of the conversation here, you see that the client and the server both agree they both have what they need, and then you see the exchange of the FIN and the ACK packets from the client to the server and the server to the client, and that effectively closes the conversation. So HTTP is an application layer protocol **for browsing the web that's been in use**
since the early 1990s. It uses a client server model using TCP port 80, but can use other ports as well.

Understand ARP:

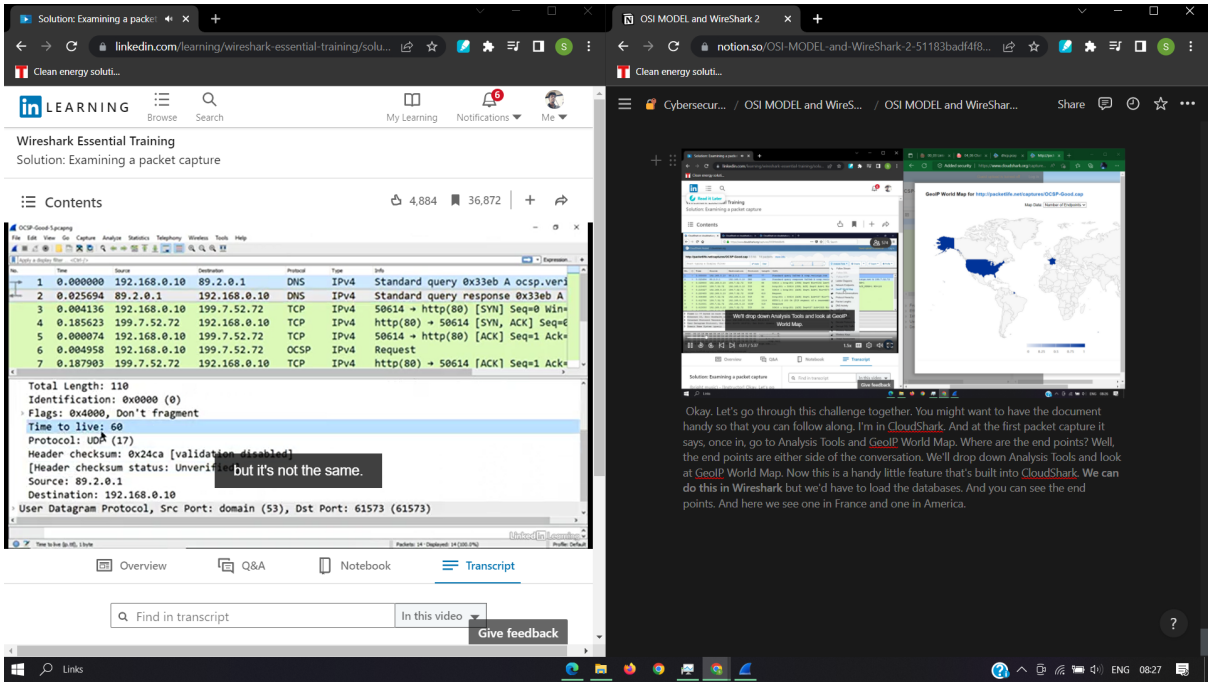
1. [Instructor] On a local area network, the data link layer uses the MAC address of the destination machine rather than the IP address. Address Resolution Protocol resolves an IP address to a MAC address on a local area network. Let's take a look at where ARP resides in the OSI model. It resides in between layer three and layer two. And that's because when we take a look at it, it doesn't have a network layer header, it doesn't have a transport layer header, it's simply doing a resolution. In addition to the ARP request, ARP replies, you might also see a gratuitous ARP, which is a test for duplicate IP addresses on a local area network



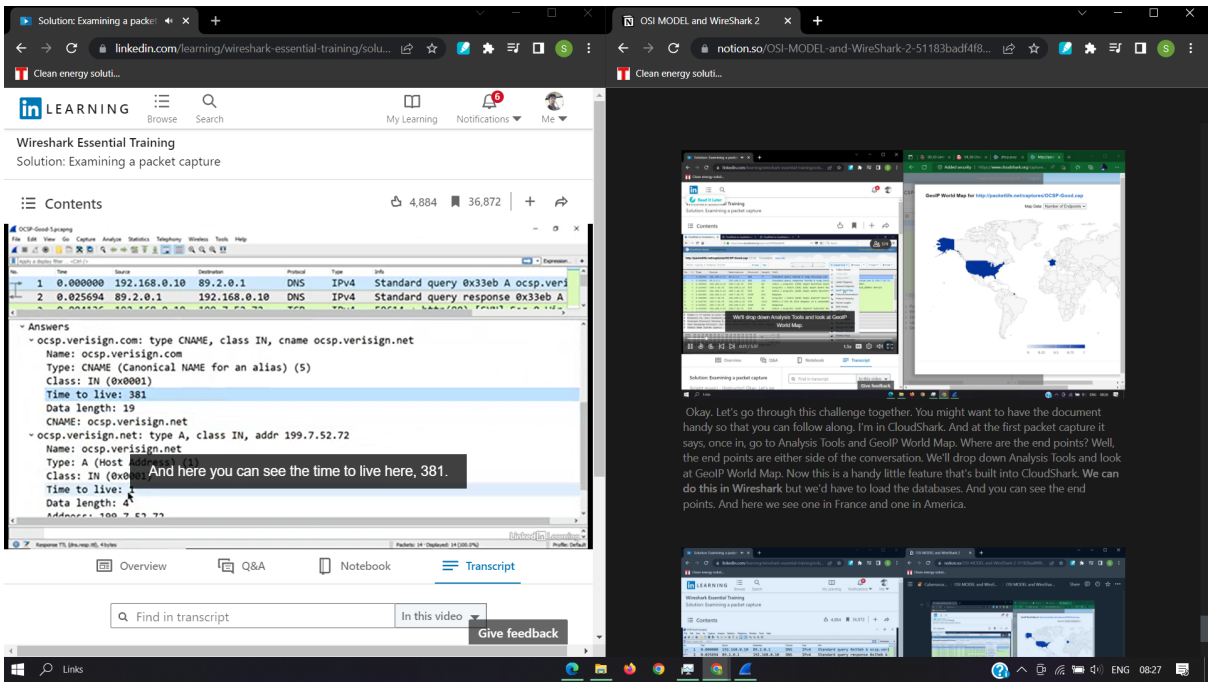
On a local area network, the data link layer uses the MAC address of the destination machine rather than the IP address. Address Resolution Protocol resolves an IP address to a MAC address on a local area network. Let's take a look at where ARP resides in the OSI model. It resides in between layer three and layer two. And that's because when we take a look at it, it doesn't have a network layer header, it doesn't have a transport layer header, it's simply doing a resolution. In addition to the ARP request, ARP replies, you might also see a gratuitous ARP, which is a test for duplicate IP addresses on a local area network. If you'd like to follow along, go to packetlife.net and I selected ARP and this last one here ICMP across dot1Q.cap. I've opened it up in Wireshark and there really aren't a lot of packets in this capture, but we'll just take a look at ARP. We'll put a display filter in and you see there are a couple of types. One is the gratuitous ARP. And as we talked about that is simply a test for duplicate IP addresses on a local area network. But then down below here, we see the ARP request and ARP reply. Now, let's take a look at the request and then we'll drop this down. And the hardware type is the type of data link type in use. And here we can see it's ethernet and we see the protocol type IP and the hardware size which is six bytes and that is the length of the hardware address and the protocol size is the length of the protocol address. Here, we see the opcode and it is a request. Now it's from the sender here which you can see, this is the sender's MAC address, and this is the sender's IP address. The target MAC address is a broadcast and remember it doesn't know the MAC address, so it goes out to everyone on the network, everyone on the network hears it, but only one should respond. And then down below here, you see this is the IP address. So looking up here in packet three it says, who has this IP address? Could you tell me who has this IP address? Then we see a reply and it has the same fields except it does provide the answer. Now we see the opcode is a reply and the sender MAC address is words coming from, and there's the IP address and then it's responding to the one that sent the ARP request. So, Address Resolution Protocol sits in between the network layer and the data link layer **to resolve an IP address to a MAC address** on a local area network.



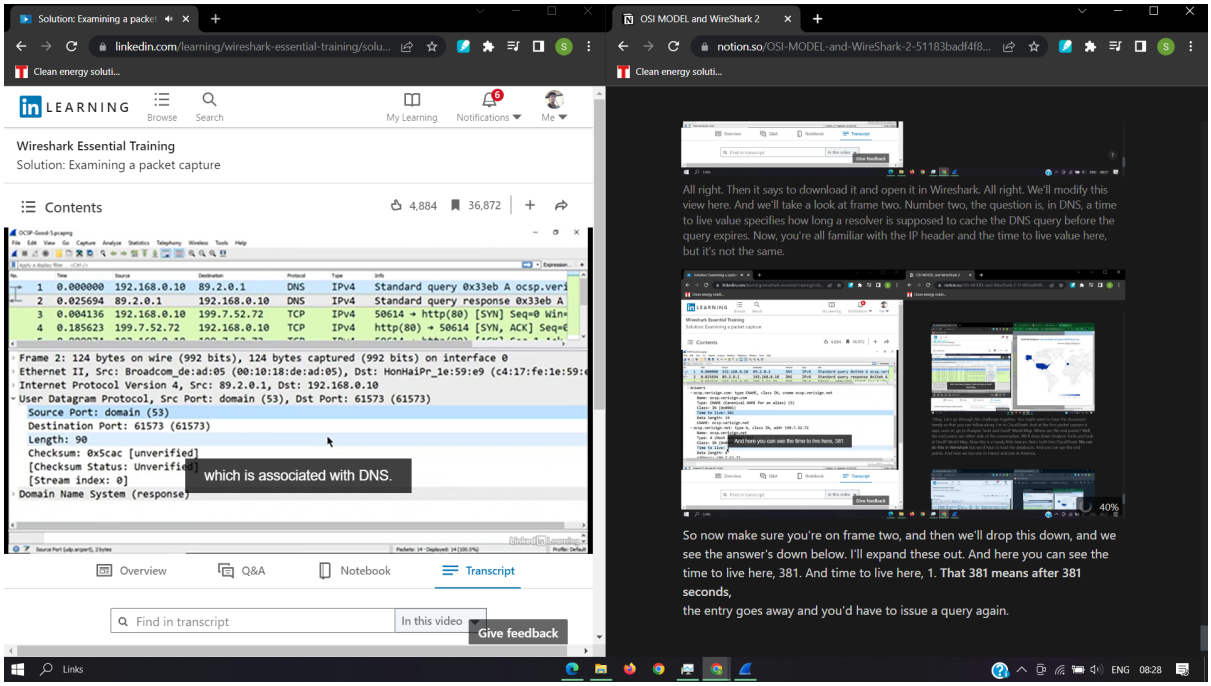
Okay. Let's go through this challenge together. You might want to have the document handy so that you can follow along. I'm in CloudShark. And at the first packet capture it says, once in, go to Analysis Tools and GeoIP World Map. Where are the end points? Well, the end points are either side of the conversation. We'll drop down Analysis Tools and look at GeoIP World Map. Now this is a handy little feature that's built into CloudShark. **We can do this in Wireshark** but we'd have to load the databases. And you can see the end points. And here we see one in France and one in America.



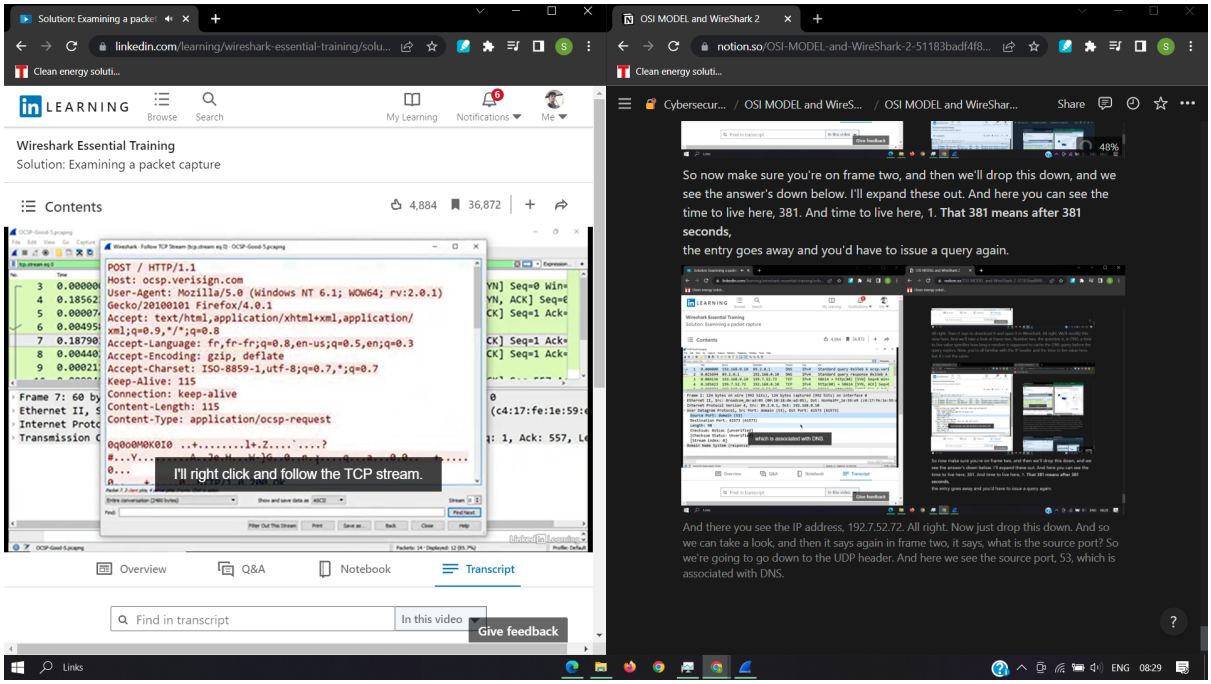
All right. Then it says to download it and open it in Wireshark. All right. We'll modify this view here. And we'll take a look at frame two. Number two, the question is, in DNS, a time to live value specifies how long a resolver is supposed to cache the DNS query before the query expires. Now, you're all familiar with the IP header and the time to live value here, but it's not the same.



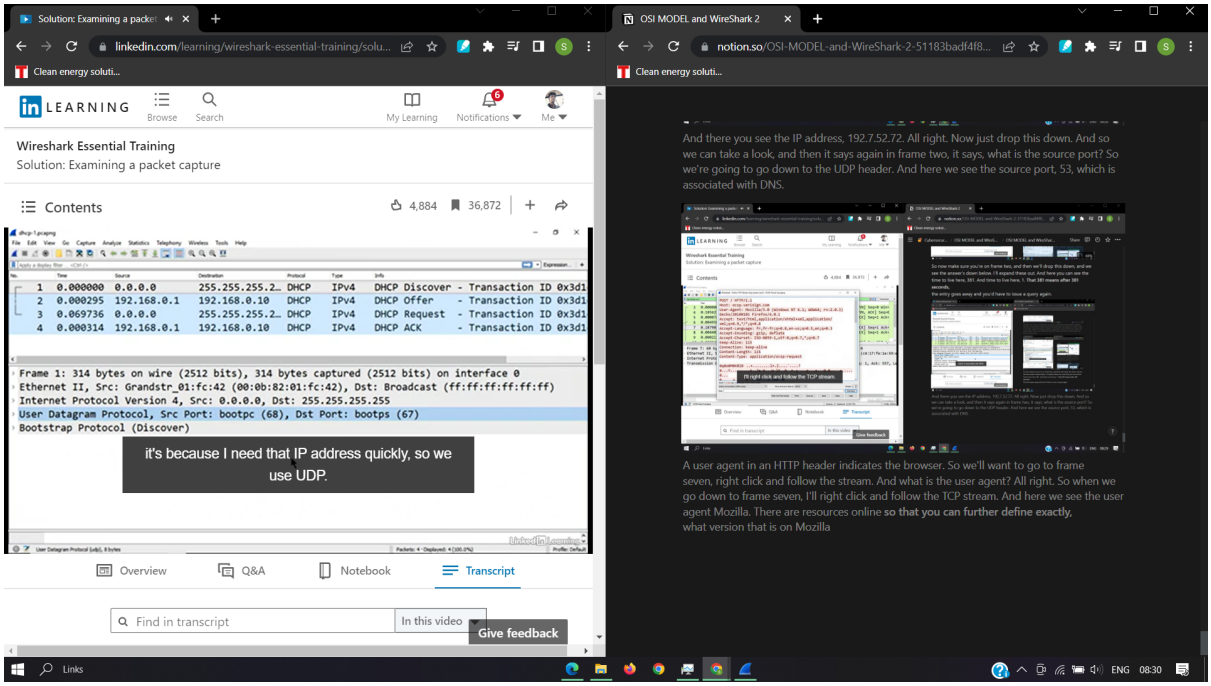
So now make sure you're on frame two, and then we'll drop this down, and we see the answer's down below. I'll expand these out. And here you can see the time to live here, 381. And time to live here, 1. **That 381 means after 381 seconds,** the entry goes away and you'd have to issue a query again.



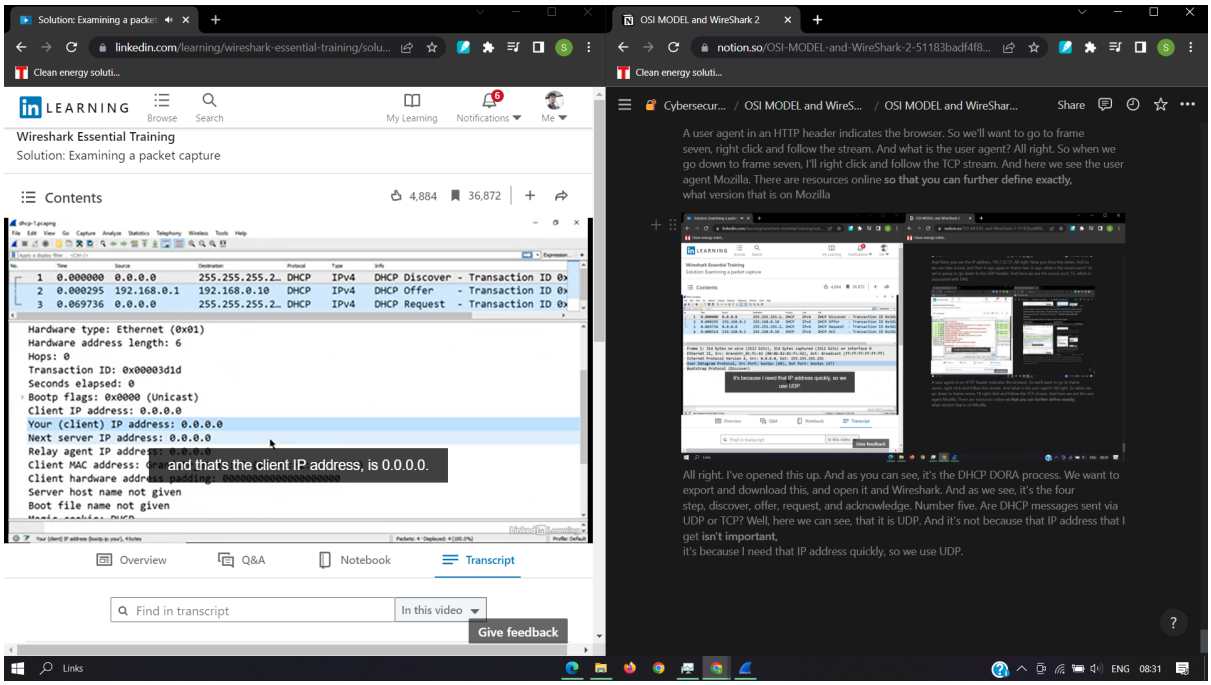
And there you see the IP address, 192.7.52.72. All right. Now just drop this down. And so we can take a look, and then it says again in frame two, it says, what is the source port? So we're going to go down to the UDP header. And here we see the source port, 53, which is associated with DNS.



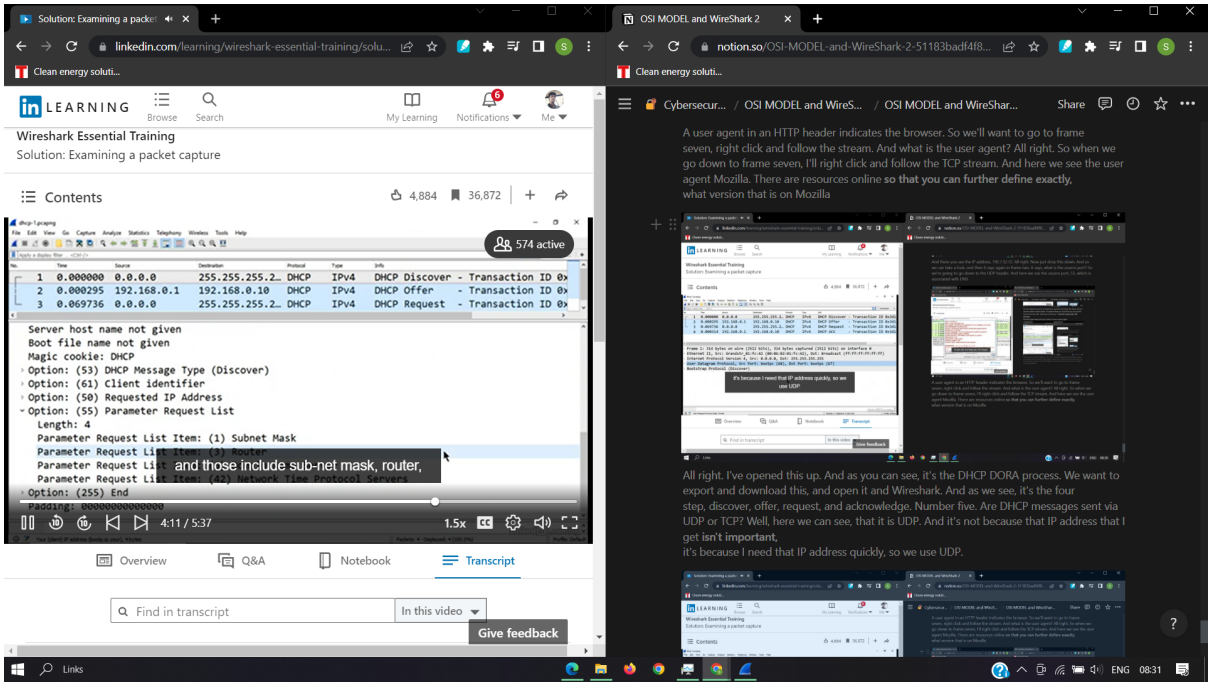
A user agent in an HTTP header indicates the browser. So we'll want to go to frame seven, right click and follow the stream. And what is the user agent? All right. So when we go down to frame seven, I'll right click and follow the TCP stream. And here we see the user agent Mozilla. There are resources online **so that you can further define exactly,**
what version that is on Mozilla



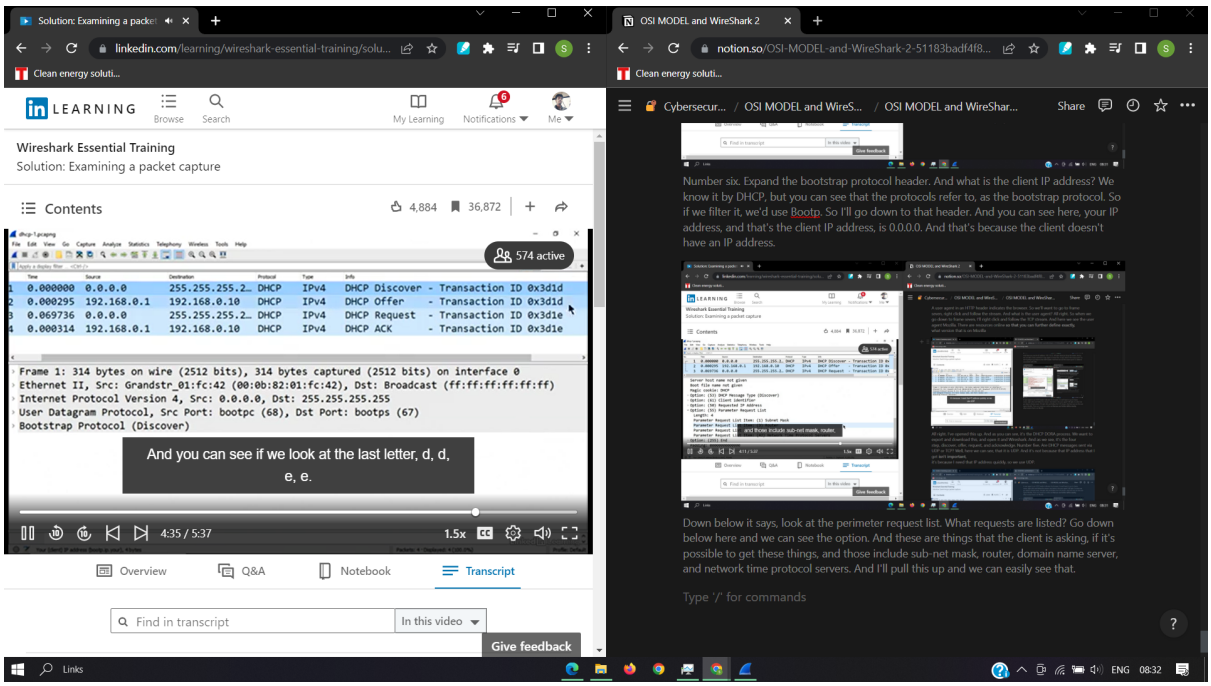
All right. I've opened this up. And as you can see, it's the DHCP DORA process. We want to export and download this, and open it and Wireshark. And as we see, it's the four step, discover, offer, request, and acknowledge. Number five. Are DHCP messages sent via UDP or TCP? Well, here we can see, that it is UDP. And it's not because that IP address that I get **isn't important,** it's because I need that IP address quickly, so we use UDP.



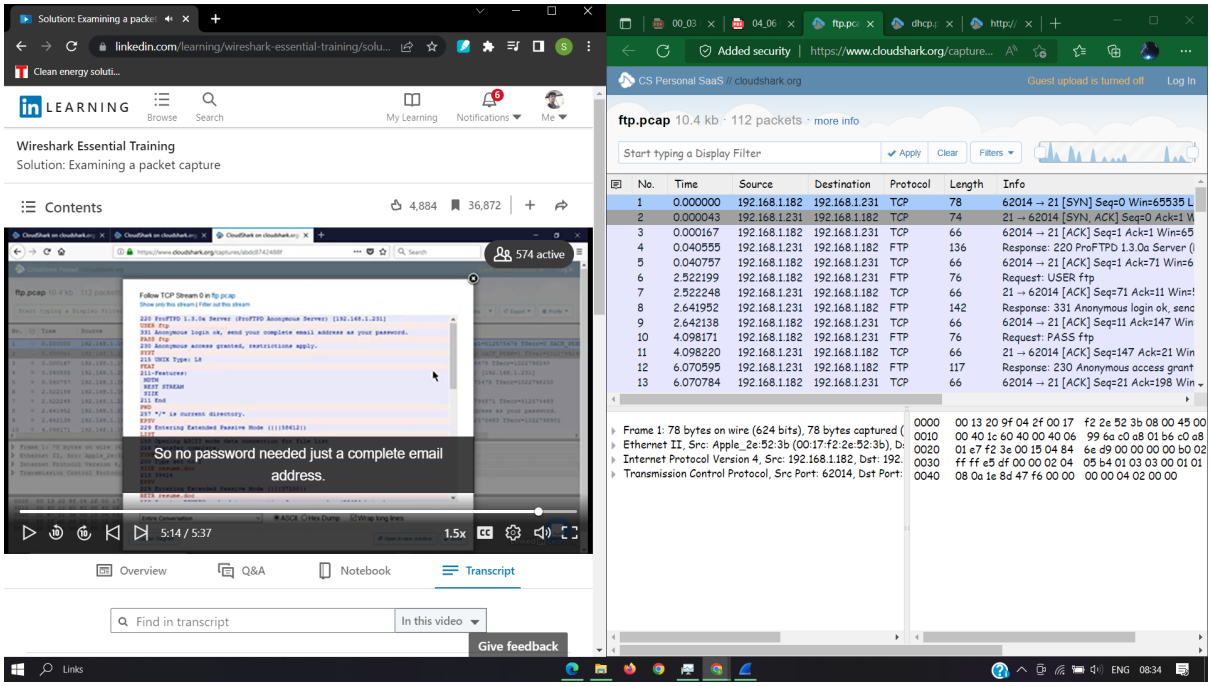
Number six. Expand the bootstrap protocol header. And what is the client IP address? We know it by DHCP, but you can see that the protocols refer to, as the bootstrap protocol. So if we filter it, we'd use Bootp. So I'll go down to that header. And you can see here, your IP address, and that's the client IP address, is 0.0.0.0. And that's because the client doesn't have an IP address.



Down below it says, look at the perimeter request list. What requests are listed? Go down below here and we can see the option. And these are things that the client is asking, if it's possible to get these things, and those include sub-net mask, router, domain name server, and network time protocol servers. And I'll pull this up and we can easily see that.

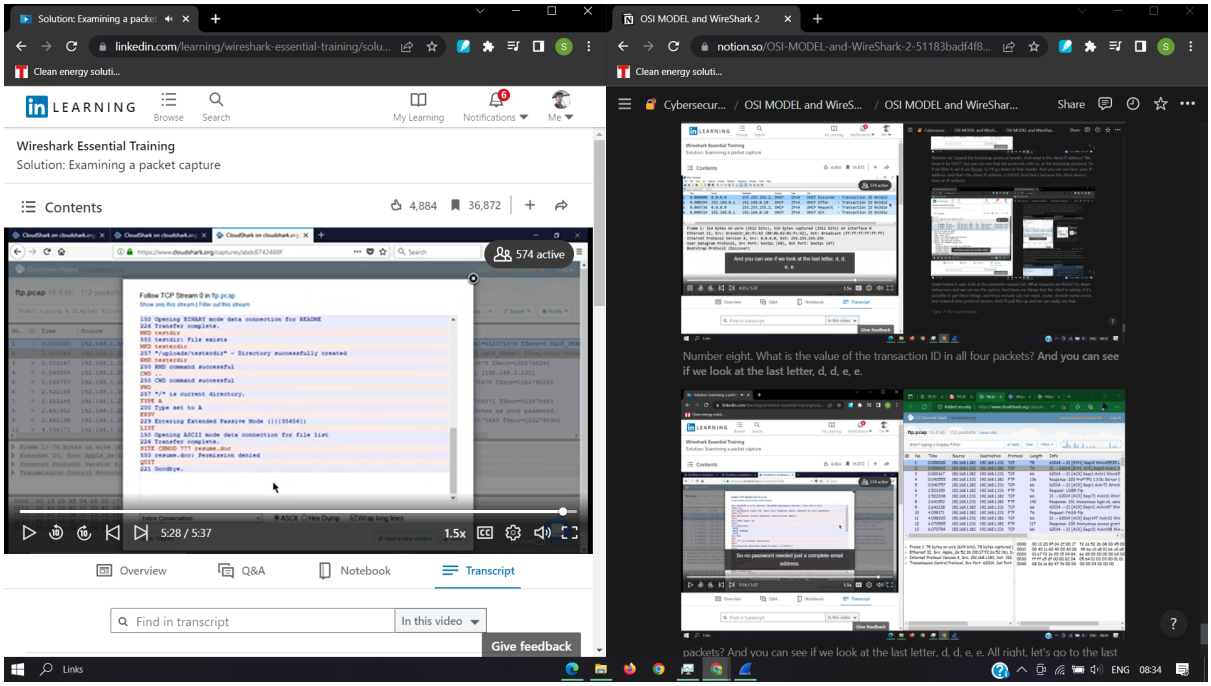


Number eight. What is the value of the transaction ID in all four packets? **And you can see if we look at the last letter, d, d, e, e.**

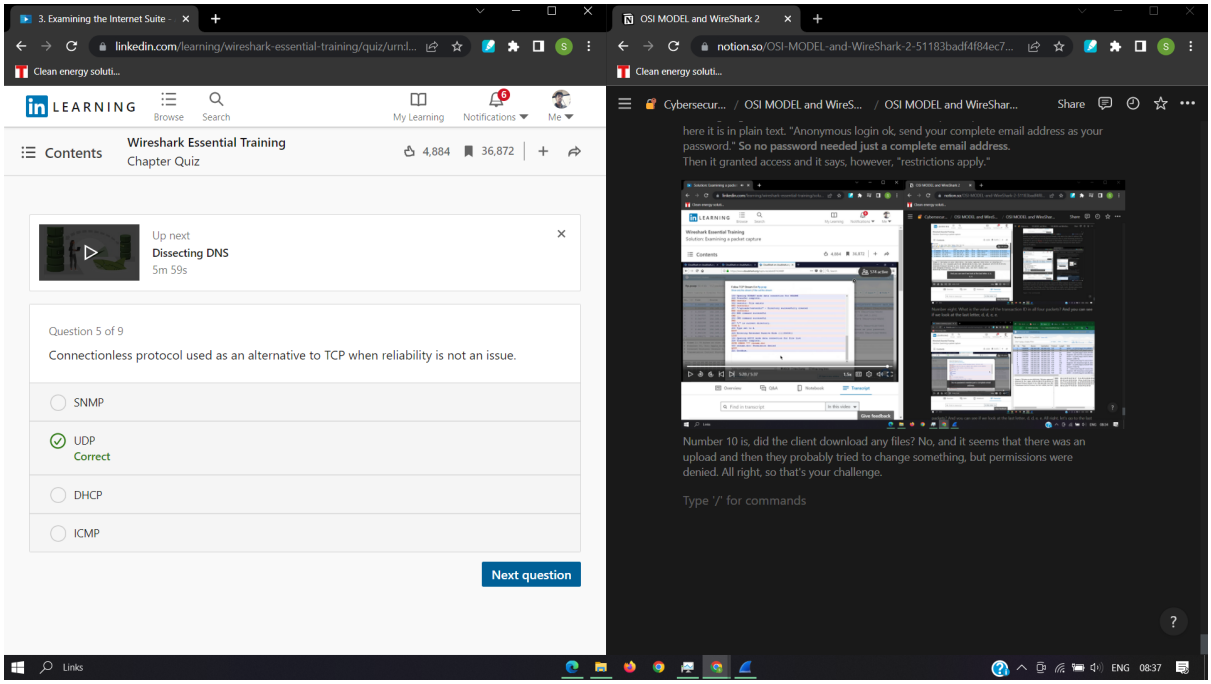


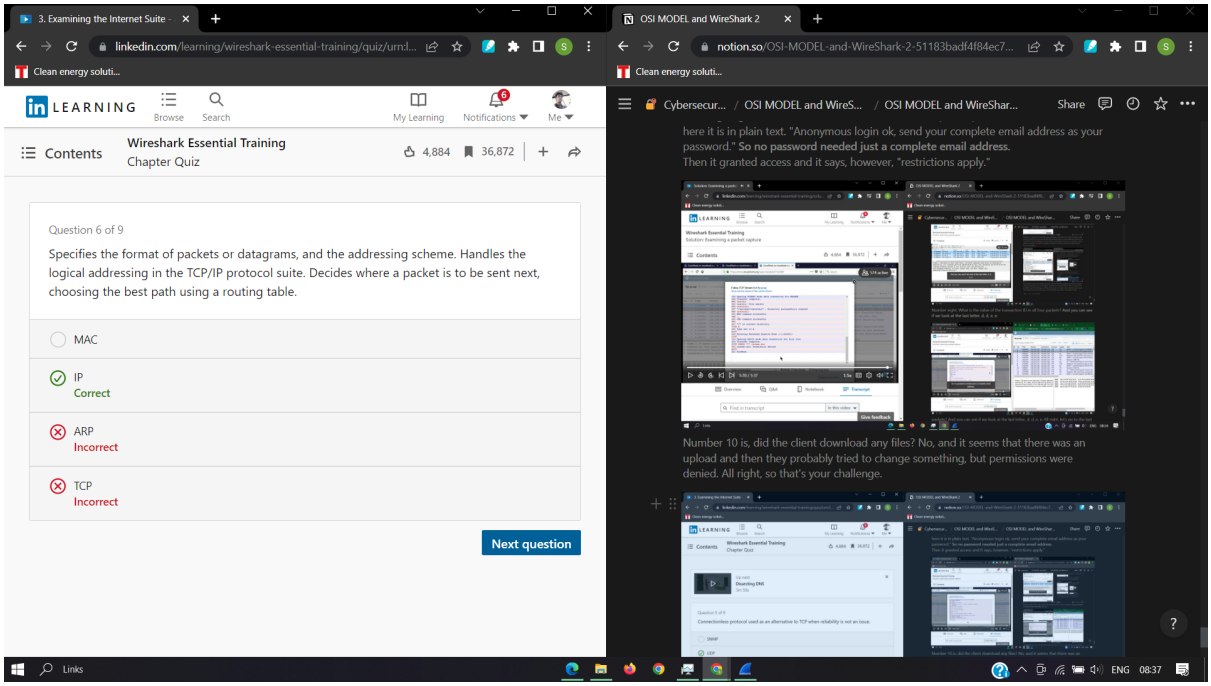
packets? And you can see if we look at the last letter, d, d, e, e. All right, let's go to the last packet capture in CloudShark. Now within this, we can see it's an FTP request and response, but what really is going on? So we need to follow the stream. So I'll go to Analysis Tools and say, follow the stream. Now there's no filter, but we can see pretty much what's going on. Number nine. Did the FTP server require a password? Well, we can see here it is in plain text. "Anonymous login ok, send your complete email address as your password." **So no password needed just a complete email address.**

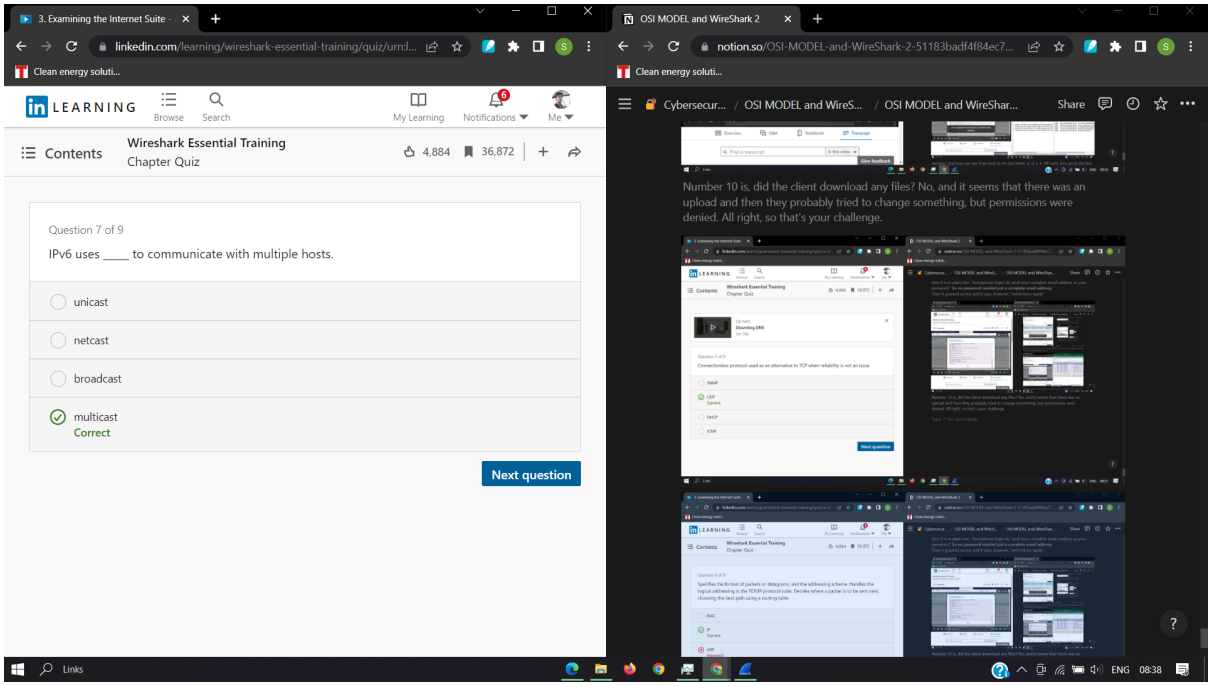
Then it granted access and it says, however, "restrictions apply."

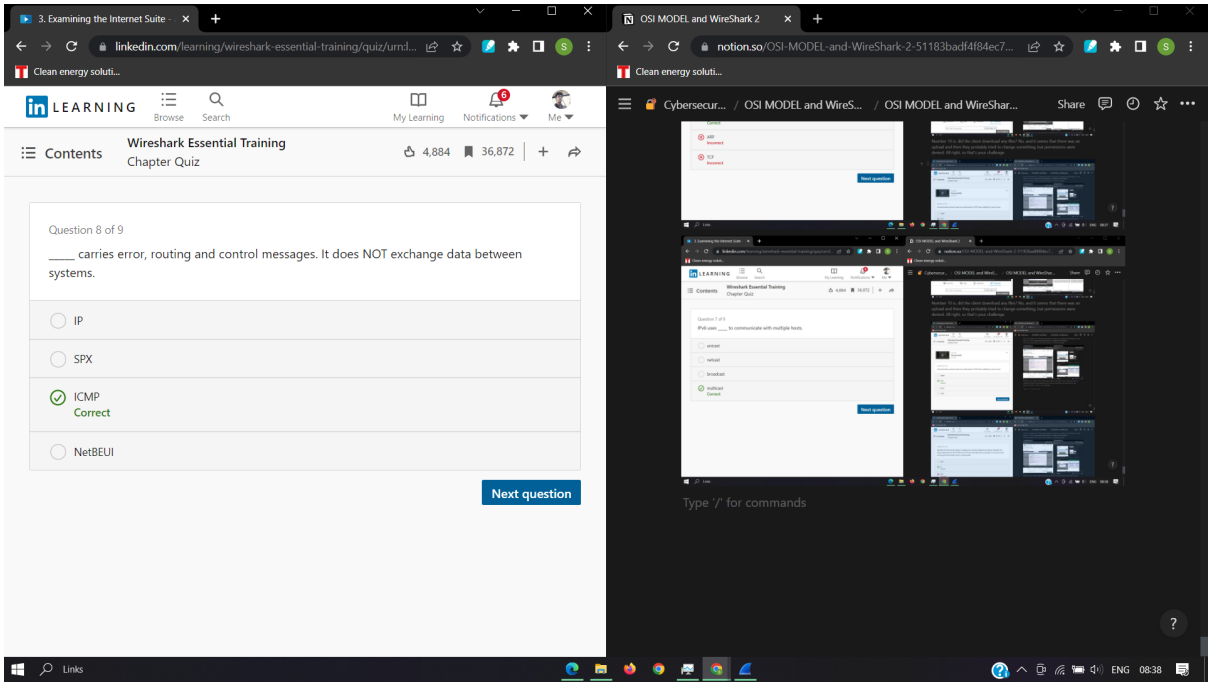


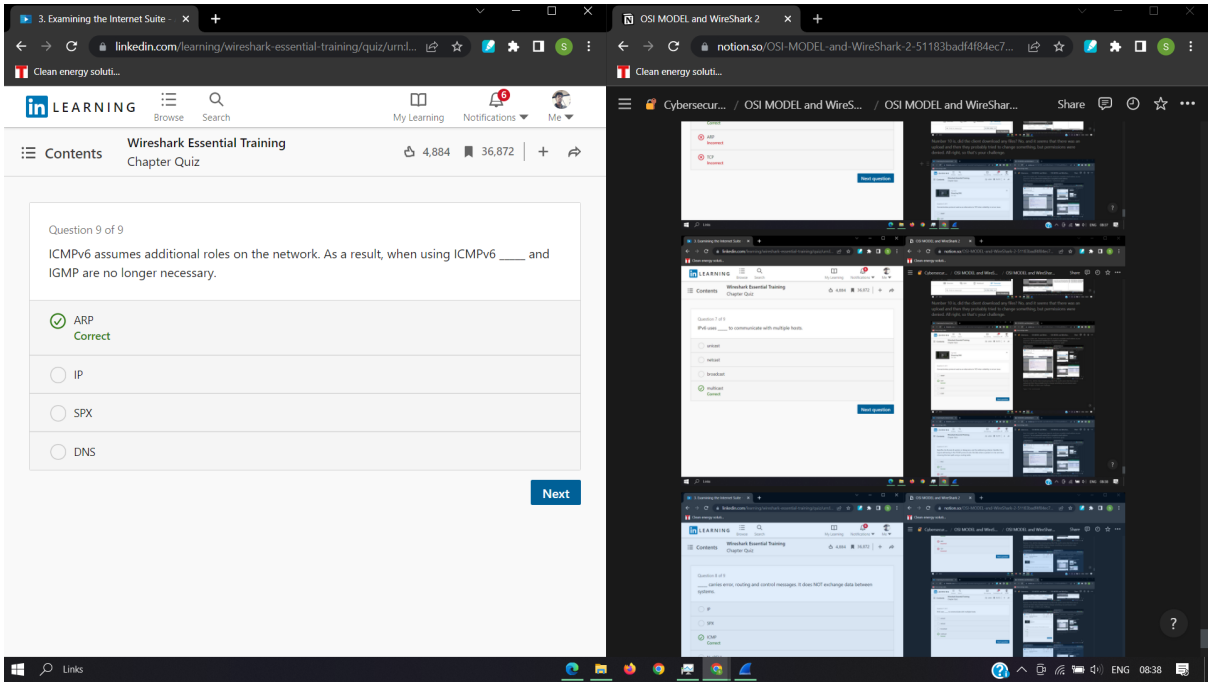
Number 10 is, did the client download any files? No, and it seems that there was an upload and then they probably tried to change something, but permissions were denied. All right, so that's your challenge.

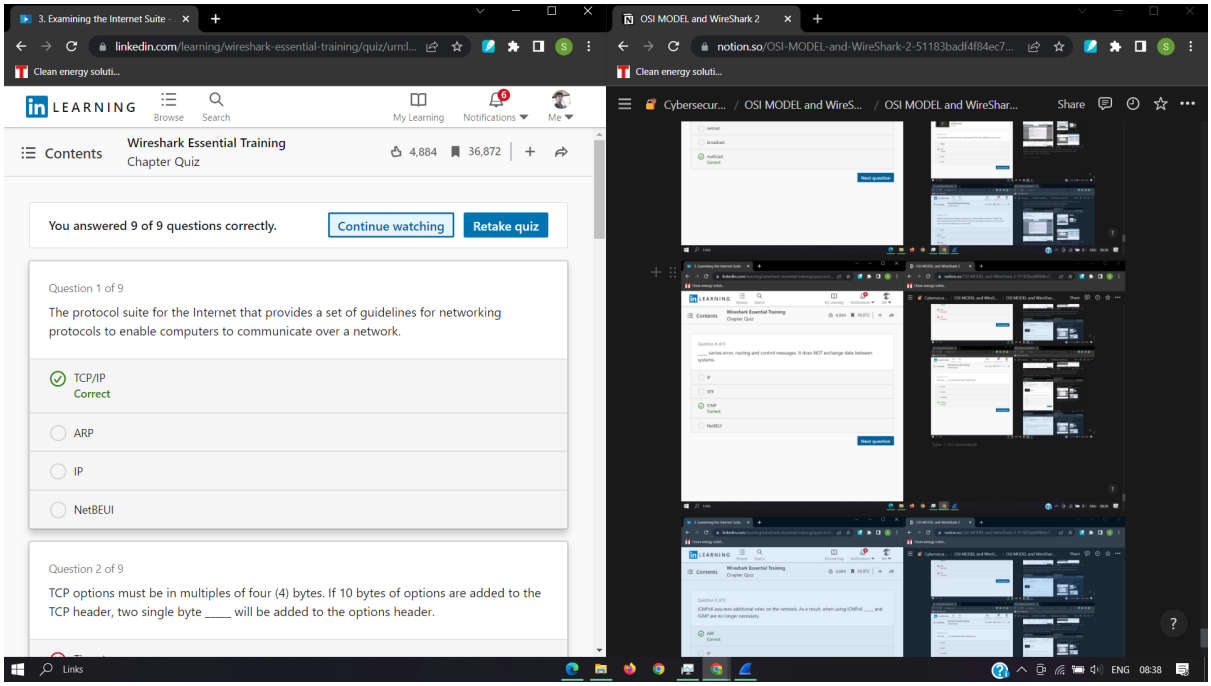


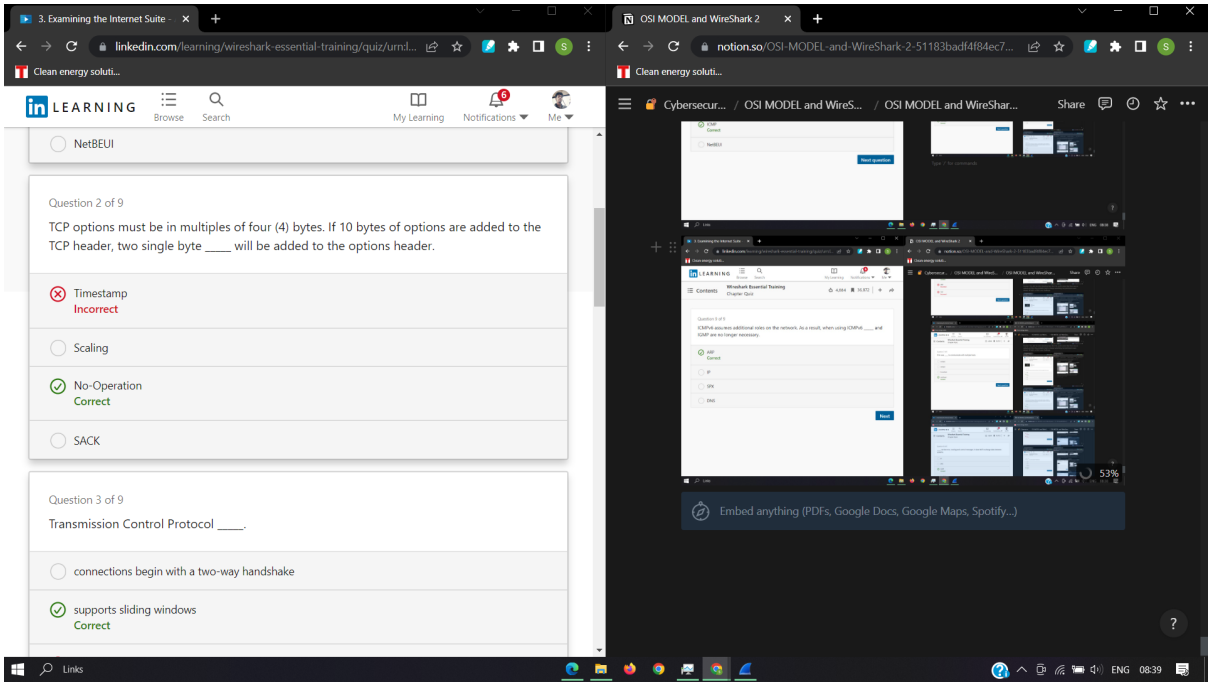

3. Examining the Internet Suite -
linkedin.com/learning/wireshark-essential-training/quiz/urn:l...
Clean energy soluti...
LEARNING
Browse
Search
My Learning
Notifications
Me
Contents
Wireshark Essential Training
Chapter Quiz
4,884
36,872
Question 6 of 9
Specifies the format of packets or datagrams, and the addressing scheme. Handles the logical addressing in the TCP/IP protocol suite. Decides where a packet is to be sent next, choosing the best path using a routing table.
MAC
IP
Correct
ARP
Incorrect
TCP
Incorrect
Next question
Links
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f84ec7...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share
here it is in plain text. "Anonymous login ok, send your complete email address as your password." So no password needed just a complete email address.
Then it granted access and it says, however, "restrictions apply."
Number 10 is, did the client download any files? No, and it seems that there was an upload and then they probably tried to change something, but permissions were denied. All right, so that's your challenge.
ENG
08:37

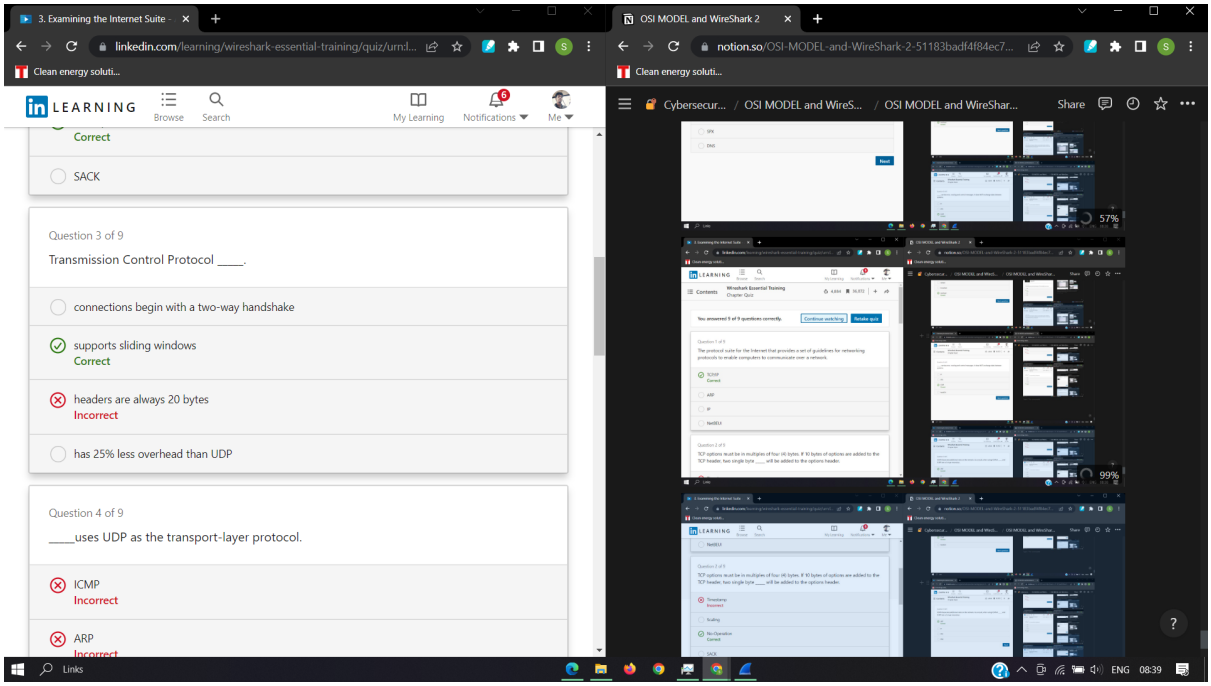

3. Examining the Internet Suite -
linkedin.com/learning/wireshark-essential-training/quiz/urn:l...
Clean energy soluti...
LEARNING
Browse
Search
My Learning
Notifications
Me
Contents
Wireshark Essential Training
Chapter Quiz
4,884
36,872
Question 7 of 9
IPv6 uses ____ to communicate with multiple hosts.
unicast
netcast
broadcast
multicast
Correct
Next question
Links
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f84ec7...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share
Number 10 is, did the client download any files? No, and it seems that there was an upload and then they probably tried to change something, but permissions were denied. All right, so that's your challenge.
ENG
08:38

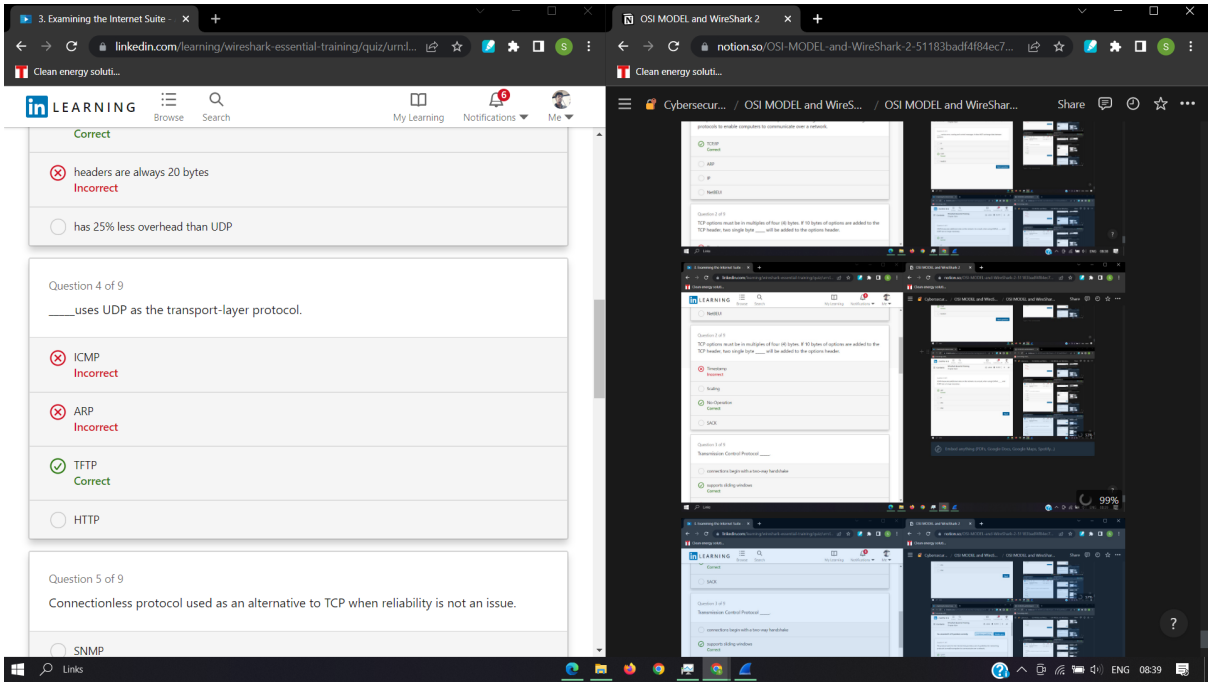

3. Examining the Internet Suite -
linkedin.com/learning/wireshark-essential-training/quiz/urn:l...
Clean energy soluti...
LEARNING
Browse
Search
My Learning
Notifications
Me
Contents
Wireshark Essential Training
Chapter Quiz
4,884
36,872
Question 8 of 9
____ carries error, routing and control messages. It does NOT exchange data between systems.
IP
SPX
ICMP
Correct
NetBEUI
Next question
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f84ec7...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share
Type '/' for commands
Links
ENG
08:38


3. Examining the Internet Suite -
linkedin.com/learning/wireshark-essential-training/quiz/urn:l...
Clean energy soluti...
LEARNING
Browse
Search
My Learning
Notifications
Me
Contents
Wireshark Essential Training
Chapter Quiz
4,884
36,872
Question 9 of 9
ICMPv6 assumes additional roles on the network. As a result, when using ICMPv6 ____ and IGMP are no longer necessary.
ARP
Correct
IP
SPX
DNS
Next
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f84ec7...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share
Links
ENG
08:38


3. Examining the Internet Suite -
linkedin.com/learning/wireshark-essential-training/quiz/urn:l...
Clean energy soluti...
LEARNING
Browse
Search
My Learning
Notifications
Me
Contents
Wireshark Essential Training
Chapter Quiz
4,884
36,872
You answered 9 of 9 questions correctly.
Continue watching
Retake quiz
Question 1 of 9
The protocol suite for the Internet that provides a set of guidelines for networking protocols to enable computers to communicate over a network.
TCP/IP
Correct
ARP
IP
NetBEUI
Question 2 of 9
TCP options must be in multiples of four (4) bytes. If 10 bytes of options are added to the TCP header, two single byte ____ will be added to the options header.
Links
ENG
08:38
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f84ec7...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share


3. Examining the Internet Suite -
linkedin.com/learning/wireshark-essential-training/quiz/urn:l...
Clean energy soluti...
LEARNING
Browse
Search
My Learning
Notifications
Me
NetBEUI
Question 2 of 9
TCP options must be in multiples of four (4) bytes. If 10 bytes of options are added to the TCP header, two single byte ____ will be added to the options header.
Timestamp
Incorrect
Scaling
No-Operation
Correct
SACK
Question 3 of 9
Transmission Control Protocol ____.
connections begin with a two-way handshake
supports sliding windows
Correct
Links
ENG
08:39
OSI MODEL and WireShark 2
notion.so/OSI-MODEL-and-WireShark-2-51183badf4f84ec7...
Cybersecur... / OSI MODEL and WireS... / OSI MODEL and WireShar...
Share
53%
Embed anything (PDFs, Google Docs, Google Maps, Spotify...)





OSI MODEL and WireShark 3