Network Security Protocols

Table of Contents

Network Security Protocols -1

The original concept for the Internet had minimal security.

Various protocols have been created over the years to address the notion of security.

These protocols have been stacked into the OSI and TCP/IP model depending on what they protect and how they do it.

**147 Some network security protocols that we need to pay attention to. So, remember, originally none of this stuff needed to worry about security. We said well we just want to actually get it up and running. We have to look at these security protocols to help us protect. So, you've heard about the regular protocols.

Now let's make them secure.

Network Security Protocols -2

Layer 1	
	None, but physical security controls can be implemented and types of cabling used can make a difference
Layer 2	
	PPTP, Layer 2 Forwarding, Layer 2 Tunneling Protocol, wireless network security, MPLS
Layer 3	
	GRE, IPSec
Layer 4	
	SSL, TLS, WTLS, SSH, SOCKS
Layer 5+	
	Application dependent, S-RPC, DNSSEC, S-HTTP

**148 At layer one we really say none. What we say here is it's none, but really what the answer is that we do physical security protections. We do conduit, those kinds of things.

Layer two we could use encryption like PPTP, or some sort of wireless network security. Now L2TP doesn't have any security in it. But it is a tunneling protocol that helps us. And it supports IPsec.

Or we could use MPLS. Now, standard MPLS is not a security protocol in and of itself. But it has some authentication mechanisms in it that we can use.

At layer three, general route encapsulation or IPsec. We'll talk about IPsec in more detail.

In layer four, this is where everything really happens. We use SSL or TLS. Those are our primary two protocols. But we could use others.

And at layer five, well we've got a web protocol HTTP. And what we use is a lower level protocol to bolt on to it. Normally, it's HTTPS. But here is also another protocol. There is another. It is SHTTP. That is a real protocol. It is a different protocol than HTTPS on the other end, separate protocol.

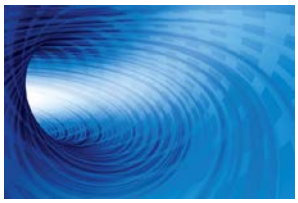
PPTP and L2F

PPTP – Point to point tunneling protocol

- PPTP → PPP → IP encapsulation for TCP/IP, IPX, and NetBEUI
- No encryption, but extended with RC4, PAP, CHAP, and EAP
- Single-factor authentication; weak implementation
- Nearly all Windows based; obsoleted by L2TP and IPSec

L2F – Layer 2 forwarding

- Tunnels at, surprise, layer 2
- Not IP dependent, supports ATM and frame relay
- Relies on PPP for authentication (designed to tunnel PPP traffic)
- Used for VPNs
- No encryption by itself



**149 Let's talk PPTP and layer two forwarding. PPTP is relatively old at this point. It encapsulated any kind of IP traffic. It didn't matter what was above it. And we used point to point tunneling protocol along with point to point protocol to communicate. There wasn't any encryption. But what we did is we did authentication with point to point tunneling protocol. And that worked pretty well for a long period of time. It's pretty much obsolete at this point.

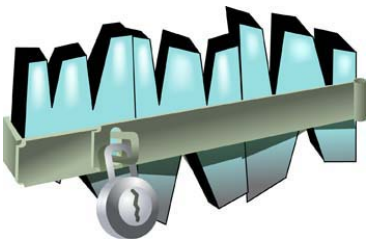
Now, specific to a particular vendor is layer two forwarding. Cisco came up with this concept. It tunnels at layer

two. It even says two in the thing. But it's not used that often. It is used for VPNs, but not that often. I mean, we're going to live and die by IPsec.

IPSec and GRE

IPSec and GRE

IPSec – Internet protocol security



- Encapsulates at Layer 3
- Mutual node authentication
- Can authenticate users, but requires L2TP
- Crypto implementation agnostic
- Client-to-client, or node-to-node (bulk)
- Mandatory for IPv6 implementation
- Does not work with NAT, unless NAT-Transversal (NAT-T) is used

GRE – Generic Route Encapsulation

- Encapsulates layer 3 packets in IP tunnel
- Used to secure VPNs
- Creates a virtual point-to-point link with destination
- Supports multicast protocols – IPSec doesn't!

**150 Okay, here's IPsec. IPsec, it encapsulates layer three. So, is it a layer two protocol? Well, not it's kind of a shim protocol that fits in between layer two and layer three. It's above the IP address in most cases unless we decide to abstract the IP address. We'll get into that a little bit later.

We can use user authentication. It can use L2TP. That depends on the

implementation. Some Cisco implementations don't. And some Microsoft implementations do. Check your local operating system for the configuration near you.

It can be done client to client to that's host to host. Or we can do it network to network using router configurations or remote access hosts. And we'll get into those later on.

Now, it is mandatory for V6 in that there is a next header for it. But it's not required that you have IPsec for IP6. You have the next header option. You can bolt it on. But it's not like everything is encrypted for IPv6.

Next is generic route encapsulation. And this encapsulates layer three packets in an IP tunnel. It is used to secure VPNs. It creates those virtual point to point links.

One of the things that it does that IPsec doesn't do is it supports multicast protocols. But what I will ask you is is when you are doing multicast protocols and encryption at the same time? Usually, we do a single point to point. Now, there are a whole bunch of other implementations from vendors out there that use combinations of IPsec and GRE and point to point and layer two tunneling protocol. And you need to talk to those vendors.

And one of the problems that we run into is those vendor specific protocols work really well if you're using all the

same gear. But when you start mixing and matching gear, things kind of blow up in your face. So, be aware of that.

IPSec -1

IPSec -1

Authentication Header (AH) – prove identity of source node (host) and provide integrity, through hashing packet data and pre-shared secret key

Encapsulating Security Payload (ESP) – encrypts packets, contains fields for header, payload, trailer (optional), and authentication (optional)

Security Association (SA) – describes protocols in use, algorithms, keys, and mode of operation; stored in Security Association Databases (SADs), one required for each communication direction

**151 Let's talk a little more about IPsec. It has two major protocols within it, AH and ESP. AH is authenticated header. And all this gives us is integrity. There is no encryption here. But what we do is we get integrity of the packet. And that's a nice thing if we need it as far as our communication is concerned. Most of the time, we don't turn on AH. But it's there if we need it.

What do we use most of the time? Well, we usually use ESP.

Encapsulated security payload is encryption. That's really what we want out of a good IPsec tunnel. That's really what we care about.

Along with that, we have the ability to configure. And the way we configure is in simplex connections through something called a security association.

So, let's talk about simplex connections through security associations. I demand that we use the strongest, most impossible encryption that's ever possible. And he says, "I just need a little bit of encryption here. That's all I really care about." So, when I talk to him, and I communicate to his network, I just use a little bit of encryption. But when he talks to me, he uses a lot of encryption.

So, in security associations, we can create what are called simplex connections. When you talk to me inbound, I require this. When I talk to you outbound, you require this. And I can change those. Now, what that did was that really messed with a lot of people when they were doing IPsec implementations. And it just drove them just crazy.

So, what ends up happening is that we end up saying let's all use really, really super-duper encryption, which is not a bad thing, except for if he's got a really small enterprise over

there. And he can't afford all that CPU utilization.

IPSec -2

IPSec -2

Mode – transport or tunnel

- Transport – IP payload is protected (usually client to server)
- Tunnel – IP payload and header are protected (usually network to network, like remote office to home office)

Internet Key Exchange – negotiates key materials for SADs

- IKMP – Internet key management protocol; builds on ISAKMP and Oakley implementations
- IPSP – IP Security Policy, establishes source, destination and type of traffic that is permitted
- Phase 1 – end-point authenticate to each other using a pre-shared secret, public keys, or a “revised” public key method
- Phase 2 – establishes a Security Association and a tunnel to secure the rest of the key exchanges

**152 Now, there are two different ways that you can communicate via IPsec, transport mode and tunnel mode. In transport mode, we go from host to host. Now, at this point, the payload is protected, but my source IP address and your source IP address are known to our adversaries. So, that might be inappropriate.

What happens if you've got your network, and I've got my network? And my host really can't figure out

how to do all this IPsec configuration stuff. But we want to have privacy between us. So, what we'll say is all your people can talk to your VPN concentrator. And all my people will talk to my VPN concentrator. And then from my network to your network, we'll make sure that we encrypt all that traffic.

What does the IP addressing scheme look like to the rest of the world? My live source IP address, your live source IP address, and then this encryption that goes around it.

What about their IP address, and your IP address back there? It doesn't know anything about it. And that's what's beautiful about tunnel mode. We can go from network to network.

Now, one thing is is what happens when we go from a single host on the Internet to a VPN concentrator? Well, since there's one host involved, we only go transport mode. How you can remember, its tunnel mode is from network to network is there's two Ns in tunnel. So, therefore, Ns and networks fit together.

Now, we can go beyond that configuration. And we can start talking about the Internet key exchange, what we call IKE, or Internet Key Management Protocol. And in this case, what we can do is we can start playing with the configurations.

How far can we carry this? Well, what's amazing to me in Internet key exchange is we can say I want to rekey every X number of seconds. So, if an adversary is listening, and they're working on the first key of our communication, and they decrypt that one piece of our conversation thinking they can jump into our stream, by then we've moved on to a new key. So, we can, in Internet key exchange, rekey very often.

We can also choose the type of keys that we use for setting things up. It could be a pre-shared secret where I call you up on the phone, and I say the password is password. Or it could be that we're in the same Kerberos realm. And we can use Kerberos tickets between us, which is in our Kerberos servers. That's if we are on the same Kerberos realm. From one Microsoft Active Directory to another or to a UNIX Kerberos network, that's between realms. And that won't work.

We could use something this is universal for us all. And that's public key infrastructure. We could use PKI certificates to exchange. And we could set up those PKI keys and then generate keys from there. So, there's all these configurations in IPsec to make it so that we can make it super-uper crazy secure. And that's really what we want.

Telnet, SSH, and SSL/TLS

Telnet – TCP/IP Terminal Emulation Protocol (T:23)

- Plain text terminal program, running over TCP port 23
- Basic authentication only

SSH – Secure shell (T:22)

- Creates tunnels that other applications can use (encapsulation)
- Provides server and client authentication and encryption
- Replaced Telnet

SSL – Secure Sockets Layer / TLS – Transport Layer Security

- Encrypts client-server communication
- Used by protocols (e.g., HTTPS) for security
- Server authentication to client mandatory
- Client authentication to server optional

SSL/TLS Handshake includes
- Encryption negotiation
- Identification of server and/or client
- Key exchange

**153 Let's switch into some little easier protocols.

Well, enter Secure Shell. The open source free tool on the client side for Secure Shell is called Putty. One the server side, we use open SSH if we're doing on it Linux. And then there's one for every one of the platforms out there. Secure Shell literally creates a secure shell on your machine. It gives me terminal access to you if you're got it set up that way.

What we could use is terminal emulation on port 23. The problem is is that that's clear text. It's plain text terminal, basic authentication, clear

text. And we don't want to use Telnet. So, what we do is we say we'll set up a secure shell, and then we'll do Telnet inside of that. And that will work out for us.

Now, the other protocol besides IPsec that you really want to invest some time in is Secure Sockets Layer or what is now known as Transport Layer Security, or TLS. It's using the encryption of PKI certificates to actually establish communications back and forth. This is exactly what the S in HTTPS is for. A lot of protocols, when you see the S bolted onto the end, it more than likely means that they took the original protocol and added TLS or SSL to it.

When we talk about client authentication, there are two ways to do this. The first way is always in done in ecommerce. What happens is you all come to my website. And I say if you want to communicate securely with me, here is my public key certificate. Validate it on your end, and then use to encrypt any communications that you're going to send back to me. On your side, you pick up that certificate. You check your list of keys that you have or your root certificates to see if I'm in the root certificates list.

And then what you do is you encrypt the communications coming back to me using my public key. Since I'm the only one with the private key, then that means that I'm the only one that can decrypt that communication with you.

What do you send me? Do you just keep on encrypting with the private key? No because that's too expensive? What we do, and again we'll talk about this in cryptology, what we do is we transmit back a session key that's going to be used for the rest of our communications.

What you should be doing when you're doing this as a VPN, is you should also be sending me your certificate that you've created and that is through a valid registrar all the way at the top of your food chain. If you do that, and you send it back to me, I should validate it on my side.

Ecommerce-wise, it doesn't matter what you send me. Yeah, do you want to buy my stuff? I'm not validating this. And that's exactly what happens. You gave me something encrypted, the session key that we're going to use, and we transmit back and forth using that session key for all of our encrypted transactions. And there you go. You get to buy your Kewpie doll. There. It's done.

See, the problem is that you didn't validate my certificate. You just asked for it. And I didn't validate your certificate. So, we are susceptible to man-in-the-middle unless you're creating real certificates and working with a real VPN. All ecommerce servers go hmph. And the throw away your certificate. So, it doesn't matter.

S-RPC and DNSSEC

S-RPC – Secure Remote Procedure Call

- Based on DES encryption algorithm
- Uses public key scheme for encryption
- Vendor specific

DNSSEC – Domain Name System Security

- Provide authentication and integrity of DNS answers
- Designed to protect against cache poisoning
- Uses public key scheme, but does not do encryption

**154 There are two other protocols in here that we want to pay attention to, secure RPC and DNSsec. Now, secure remote procedure call is vendor specific. It's only for Sun. It is secure. But it's based on the Sun platform. It uses the data encryption algorithm. So, that's a little bit dated at this point. And it uses public key schemes for encryption of that des key that's being transmitted. We don't use it that often. But from an option, this would be a possibility if you were talking Sun to Sun. there's nothing wrong with that.

In DNSsec, what we're doing is is we're staying the integrity of the

response of the servers that sends us back a resolution for fully qualified domain names to IP addresses is insufficient. The integrity of this response is in question. So, therefore, what we will do is we will upgrade to DNSsec which has an integrity of response.

It is not encrypted DNS communications. It is integrity DNS communications. It uses public keys that are attacked and signed by root certificate-- by DNS roots as it goes down to each one of the daughters or sons underneath of that particular DNS root server.

Notices

Notices