Security and Authentication in IP

VPNs

Many other VPN implementations exist

Security and Authentication in IP

VPNs

Many other VPN implementations exist

- *Crypto IP Encapsulation* (CIPE). A lightweight point-to-point protocol that layers over UDP

Security and Authentication in IP

VPNs

Many other VPN implementations exist

- *Crypto IP Encapsulation* (CIPE). A lightweight point-to-point protocol that layers over UDP
- `ssh`. This remote login protocol also has a VPN mode, but it layers over TCP

Security and Authentication in IP

VPNs

Many other VPN implementations exist

- *Crypto IP Encapsulation* (CIPE). A lightweight point-to-point protocol that layers over UDP
- `ssh`. This remote login protocol also has a VPN mode, but it layers over TCP
- OpenVPN (discussed later) tunnels over the transport layer SSL/TLS

Security and Authentication in IP

VPNs

The cryptographic quality of these varies widely: CIPE is generally judged to be not much better than PPTP

Security and Authentication in IP

VPNs

The cryptographic quality of these varies widely: CIPE is generally judged to be not much better than PPTP

In real life, PPTP and OpenVPN are common; Wireguard is growing in popularity; the others are rarely seen

Security and Authentication in IP

VPNs

The cryptographic quality of these varies widely: CIPE is generally judged to be not much better than PPTP

In real life, PPTP and OpenVPN are common; Wireguard is growing in popularity; the others are rarely seen

Exercise Read about the vulnerabilities in PPTP

Security and Authentication in IP

Transport Layer

Transport Layer security is used much more than Network Layer security

Security and Authentication in IP

Transport Layer

Transport Layer security is used much more than Network Layer security

The *Secure Socket Layer* (SSL) and its update *Transport Layer Security* (TLS) implement a security layer over the transport layer (usually TCP)

Security and Authentication in IP

Transport Layer

Transport Layer security is used much more than Network Layer security

The *Secure Socket Layer* (SSL) and its update *Transport Layer Security* (TLS) implement a security layer over the transport layer (usually TCP)

And you use this new layer instead of TCP

Security and Authentication in IP

Transport Layer

Transport Layer security is used much more than Network Layer security

The *Secure Socket Layer* (SSL) and its update *Transport Layer Security* (TLS) implement a security layer over the transport layer (usually TCP)

And you use this new layer instead of TCP

Note that SSL is no longer recommended as it has flaws in the protocols

Security and Authentication in IP

Transport Layer

Transport Layer security is used much more than Network Layer security

The *Secure Socket Layer* (SSL) and its update *Transport Layer Security* (TLS) implement a security layer over the transport layer (usually TCP)

And you use this new layer instead of TCP

Note that SSL is no longer recommended as it has flaws in the protocols

You should only use TLS, preferably versions 1.2 or later

Security and Authentication in IP

TLS

This security layer is above TCP, so it must be in the application layer

Security and Authentication in IP

TLS

This security layer is above TCP, so it must be in the application layer

It provides security of the data and authentication of the remote host

Security and Authentication in IP

TLS

This security layer is above TCP, so it must be in the application layer

It provides security of the data and authentication of the remote host

After a TCP connection has been made a TLS handshake in the application authenticates the connection and negotiates a secret key

Security and Authentication in IP

TLS

This security layer is above TCP, so it must be in the application layer

It provides security of the data and authentication of the remote host

After a TCP connection has been made a TLS handshake in the application authenticates the connection and negotiates a secret key

The key is then used to encrypt subsequent data sent over the TCP connection

Security and Authentication in IP

TLS

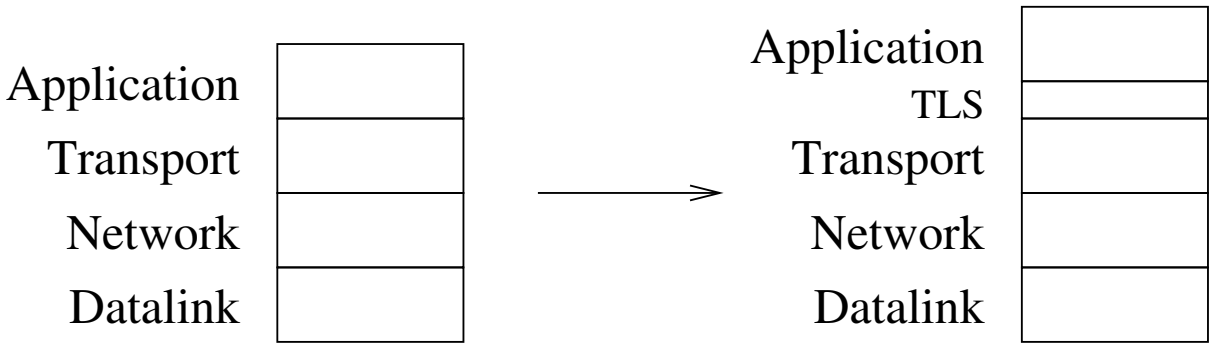
TLS provides a new transport layer that can be used very much like TCP (reliable, connection oriented, etc.)

Security and Authentication in IP

TLS

TLS provides a new transport layer that can be used very much like TCP (reliable, connection oriented, etc.)

Sometimes called a *shim* layer as it sits between two other layers



TLS Shim

Security and Authentication in IP

TLS

The client can authenticate the server through the use of public-key certificates

Security and Authentication in IP

TLS

The client can authenticate the server through the use of public-key certificates

During the handshake the client application receives a certificate from the server that it can authenticate in a variety of ways

Security and Authentication in IP

TLS

The client can authenticate the server through the use of public-key certificates

During the handshake the client application receives a certificate from the server that it can authenticate in a variety of ways

For example, Web browsers often contain a selection of master certificates from *certification authorities* that it can use to check the certificate from the server

Security and Authentication in IP

TLS

The client can authenticate the server through the use of public-key certificates

During the handshake the client application receives a certificate from the server that it can authenticate in a variety of ways

For example, Web browsers often contain a selection of master certificates from *certification authorities* that it can use to check the certificate from the server

Exercise Examine your browser to see which certification authorities it uses

Security and Authentication in IP

TLS

Similarly, if it wishes, the server can request a certificate from the client to authenticate the client a similar way

Security and Authentication in IP

TLS

Similarly, if it wishes, the server can request a certificate from the client to authenticate the client a similar way

It would be possible to use this instead of the usual “login and password” mechanism that servers often use to authenticate the client

Security and Authentication in IP

TLS

Similarly, if it wishes, the server can request a certificate from the client to authenticate the client a similar way

It would be possible to use this instead of the usual “login and password” mechanism that servers often use to authenticate the client

Unfortunately, the requirements of administration of the certificates is much beyond the skill of the average user

Security and Authentication in IP

TLS

Similarly, if it wishes, the server can request a certificate from the client to authenticate the client a similar way

It would be possible to use this instead of the usual “login and password” mechanism that servers often use to authenticate the client

Unfortunately, the requirements of administration of the certificates is much beyond the skill of the average user

Which is why login and password is still widely used to authenticate clients to the server

Security and Authentication in IP

TLS

Transport layer security is very flexible, but requires the application programmer to understand and use the function calls to set up certificate checking and the handshake

Security and Authentication in IP

TLS

Transport layer security is very flexible, but requires the application programmer to understand and use the function calls to set up certificate checking and the handshake

That is, the programmer must invoke this layer in their application: and correct use of TLS is not trivial

Security and Authentication in IP

TLS

Transport layer security is very flexible, but requires the application programmer to understand and use the function calls to set up certificate checking and the handshake

That is, the programmer must invoke this layer in their application: and correct use of TLS is not trivial

Their program can then read and write via the secure connection they get from this instead of reading and writing directly from the TCP socket

Security and Authentication in IP

```
s = socket(PF_INET, SOCK_STREAM, 0); // TCP socket
...
// Initiate TCP connection to server
connect(s, (struct sockaddr *)&addr, ... );
...
read(s, buf, 1024); // read data
...
```

Security and Authentication in IP

```
s = socket(PF_INET, SOCK_STREAM, 0); // TCP socket
...
connect(s, (struct sockaddr *)&caddr, ... );
...
ssl = SSL_new(ctx); // context contains info about ciphers
SSL_set_fd(ssl, s); // associate socket with ssl struct
...
SSL_connect(ssl); // do the SSL handshake
...
if SSL_get_verify_result(ssl) != X509_V_OK { // authenticate
... bad certificate ...
}
...
SSL_read(ssl, buf, 1024); // read data
...
```

Security and Authentication in IP

TLS

Many protocols can layer over TLS (instead of TCP) to give a secure version:

Security and Authentication in IP

TLS

Many protocols can layer over TLS (instead of TCP) to give a secure version:

- HTTPS is HTTP (the protocol to fetch Web pages) layered over TLS

Security and Authentication in IP

TLS

Many protocols can layer over TLS (instead of TCP) to give a secure version:

- HTTPS is HTTP (the protocol to fetch Web pages) layered over TLS
- SMTPS for SMTP (the protocol used to send email)

Security and Authentication in IP

TLS

Many protocols can layer over TLS (instead of TCP) to give a secure version:

- HTTPS is HTTP (the protocol to fetch Web pages) layered over TLS
- SMTPS for SMTP (the protocol used to send email)
- IMAPS for IMAP (the protocol used to read email)

Security and Authentication in IP

TLS

Many protocols can layer over TLS (instead of TCP) to give a secure version:

- HTTPS is HTTP (the protocol to fetch Web pages) layered over TLS
- SMTPS for SMTP (the protocol used to send email)
- IMAPS for IMAP (the protocol used to read email)
- Etc.

Security and Authentication in IP

TLS

Many protocols can layer over TLS (instead of TCP) to give a secure version:

- HTTPS is HTTP (the protocol to fetch Web pages) layered over TLS
- SMTPS for SMTP (the protocol used to send email)
- IMAPS for IMAP (the protocol used to read email)
- Etc.

This is a relatively easy way of making secure protocols from insecure ones: just find the parts of code that read and write from IP sockets and change them to use TLS

Security and Authentication in IP

TLS

A few people regard TLS as a presentation layer between the application and the transport layer

Security and Authentication in IP

TLS

A few people regard TLS as a presentation layer between the application and the transport layer

An interesting point of view, as TLS does rearrange your data

Security and Authentication in IP

TLS

A few people regard TLS as a presentation layer between the application and the transport layer

An interesting point of view, as TLS does rearrange your data

But not a strong point of view, as TLS does not solve the other problems a presentation layer is supposed to address, e.g., character sets

Security and Authentication in IP

TLS

A few people regard TLS as a presentation layer between the application and the transport layer

An interesting point of view, as TLS does rearrange your data

But not a strong point of view, as TLS does not solve the other problems a presentation layer is supposed to address, e.g., character sets

Most people regard TLS as a transport layer

Security and Authentication in IP

TLS

Exercise Read about STARTTLS, a protocol to negotiate a TLS connection, as used by SMTP and IMAP

Exercise Contrast HTTPS with SHTTP, which is an extension of HTTP to include security

Exercise Read about HTTP/2, the latest version of HTTP, that encourages the use of TLS

Security and Authentication in IP

TLS

We should also mention QUIC (again) at this point

Security and Authentication in IP

TLS

We should also mention QUIC (again) at this point

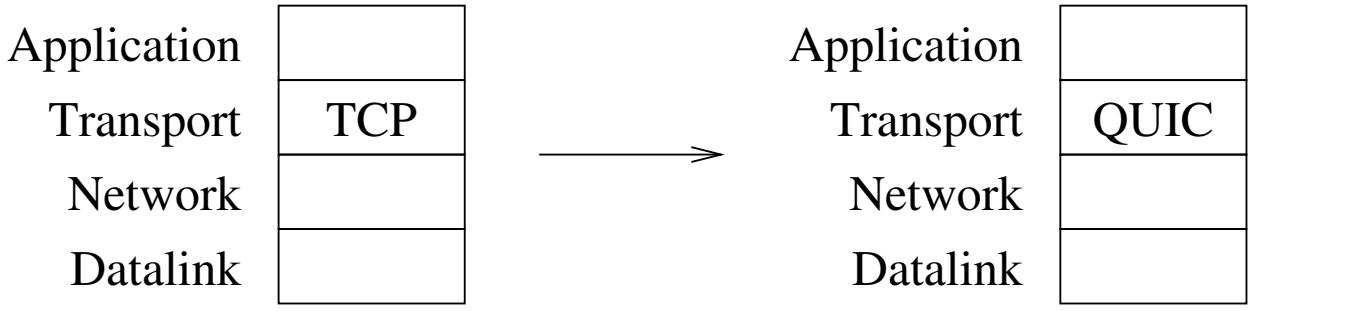
QUIC is a transport layer that replaces TCP + TLS

Security and Authentication in IP

TLS

We should also mention QUIC (again) at this point

QUIC is a transport layer that replaces TCP + TLS



insecure, unauthenticated secure, authenticated

QUIC

Security and Authentication in IP

TLS

The opening handshake of QUIC does both the reliability setup *and* the security setup

Security and Authentication in IP

TLS

The opening handshake of QUIC does both the reliability setup *and* the security setup

Thus making QUIC faster to set up

Security and Authentication in IP

TLS

The opening handshake of QUIC does both the reliability setup *and* the security setup

Thus making QUIC faster to set up

In the future, QUIC will the transport layer of the Web (HTTP/3), and possibly other applications, too (e.g., DNS)

Security and Authentication in IP

TLS

HTTPS hides the requested URL and the content of a Web page returned: this is in the encrypted data of the HTTP request; but it cannot hide the IP address of the server

Security and Authentication in IP

TLS

HTTPS hides the requested URL and the content of a Web page returned: this is in the encrypted data of the HTTP request; but it cannot hide the IP address of the server

So an eavesdropper cannot tell if you are reading `www.example.com/good.html` or `www.example.com/bad.html`

Security and Authentication in IP

TLS

HTTPS hides the requested URL and the content of a Web page returned: this is in the encrypted data of the HTTP request; but it cannot hide the IP address of the server

So an eavesdropper cannot tell if you are reading `www.example.com/good.html` or `www.example.com/bad.html`

They *can* tell you are looking at something on the host with the IP address of `www.example.com`

Security and Authentication in IP

TLS

HTTPS hides the requested URL and the content of a Web page returned: this is in the encrypted data of the HTTP request; but it cannot hide the IP address of the server

So an eavesdropper cannot tell if you are reading `www.example.com/good.html` or `www.example.com/bad.html`

They *can* tell you are looking at something on the host with the IP address of `www.example.com`

Traffic analysis of communications is a powerful tool that has been used for decades

Security and Authentication in IP

TLS

Some Websites (e.g., Tumblr) have multiple sub-sites hosted on the same IP address: called *virtual hosting*

Security and Authentication in IP

TLS

Some Websites (e.g., Tumblr) have multiple sub-sites hosted on the same IP address: called *virtual hosting*

For example, `good.tumblr.com/home.html` and `bad.tumblr.com/home.html` with both `good.tumblr.com` and `bad.tumblr.com` having the same IP address

Security and Authentication in IP

TLS

Some Websites (e.g., Tumblr) have multiple sub-sites hosted on the same IP address: called *virtual hosting*

For example, `good.tumblr.com/home.html` and `bad.tumblr.com/home.html` with both `good.tumblr.com` and `bad.tumblr.com` having the same IP address

The server name is included in the HTTP request and the server uses this to determine which sub-site the client wants

Security and Authentication in IP

TLS

```
GET /home.html HTTP/1.1
Host: bad.tumblr.com
User-Agent: curl/7.60.0
Accept: */*
```

HTTP request for `home.html` on (virtual) server `bad.tumblr.com`

Security and Authentication in IP

TLS

```
GET /home.html HTTP/1.1
Host: bad.tumblr.com
User-Agent: curl/7.60.0
Accept: */*
```

HTTP request for `home.html` on (virtual) server `bad.tumblr.com`

HTTPS runs over TLS so this is hidden from an eavesdropper

Security and Authentication in IP

TLS

But the TLS handshake (before the HTTP request) requires an authentication certificate from the server that is based on the server name

Security and Authentication in IP

TLS

But the TLS handshake (before the HTTP request) requires an authentication certificate from the server that is based on the server name

Server Name Indication (SNI; RFC6066) is part of the HTTPS handshake that asks for a cerfificate for the server name (e.g., `bad.tumblr.com`) *in the clear*

Security and Authentication in IP

TLS

But the TLS handshake (before the HTTP request) requires an authentication certificate from the server that is based on the server name

Server Name Indication (SNI; RFC6066) is part of the HTTPS handshake that asks for a cerfificate for the server name (e.g., `bad.tumblr.com`) *in the clear*

As we don't yet have a shared secret key, this can't be encrypted

Security and Authentication in IP

TLS

So accesses to such sub-sites are trackable:

- in the DNS lookup of the sub-site name
- in the HTTPS SNI handshake that contains the name of the sub-site

Security and Authentication in IP

TLS

So accesses to such sub-sites are trackable:

- in the DNS lookup of the sub-site name
- in the HTTPS SNI handshake that contains the name of the sub-site

Although the *content* of the Web pages is always hidden, which sites are being accessed can be tracked

Security and Authentication in IP

TLS

People are working on filling these gaps

Security and Authentication in IP

TLS

People are working on filling these gaps

We have already mentioned DNS over HTTPS (DoH; RFC8484) that hides the DNS lookup

Security and Authentication in IP

TLS

People are working on filling these gaps

We have already mentioned DNS over HTTPS (DoH; RFC8484) that hides the DNS lookup

Exercise Read about Encrypted SNI (eSNI) that hides the handshake

Exercise Read about Oblivious DNS over HTTPS (ODoH) that hides the DNS request from the DNS server(!)

Exercise Why are sites like Reddit that also have many sub-sites not affected by this?

Security and Authentication in IP

Aside

Think about the ways your Internet use can be tracked (by ISPs or others, for the Investigatory Powers Act; or just general snooping by bad actors) or manipulated (by bad actors, including some ISPs)

Security and Authentication in IP

Aside

Think about the ways your Internet use can be tracked (by ISPs or others, for the Investigatory Powers Act; or just general snooping by bad actors) or manipulated (by bad actors, including some ISPs)

These include:

Security and Authentication in IP

Aside

Think about the ways your Internet use can be tracked (by ISPs or others, for the Investigatory Powers Act; or just general snooping by bad actors) or manipulated (by bad actors, including some ISPs)

These include:

- Reading/manipulating your Web traffic or emails (unless you use HTTPS or an appropriate secure transport)

Security and Authentication in IP

Aside

Think about the ways your Internet use can be tracked (by ISPs or others, for the Investigatory Powers Act; or just general snooping by bad actors) or manipulated (by bad actors, including some ISPs)

These include:

- Reading/manipulating your Web traffic or emails (unless you use HTTPS or an appropriate secure transport)
- Reading/manipulating your DNS requests (unless you use DoH or similar)

Security and Authentication in IP

Aside

Think about the ways your Internet use can be tracked (by ISPs or others, for the Investigatory Powers Act; or just general snooping by bad actors) or manipulated (by bad actors, including some ISPs)

These include:

- Reading/manipulating your Web traffic or emails (unless you use HTTPS or an appropriate secure transport)
- Reading/manipulating your DNS requests (unless you use DoH or similar)
- Reading/manipulating your Server Name Indication traffic on TLS authentication certificates (unless you use eSNI)

Security and Authentication in IP

TLS

There are overheads in using TLS

Security and Authentication in IP

TLS

There are overheads in using TLS

- A one-off overhead of (re)writing the application code to use TLS

Security and Authentication in IP

TLS

There are overheads in using TLS

- A one-off overhead of (re)writing the application code to use TLS
- A per-connection overhead of TLS setup messages and the associated computation for checking certificates

Security and Authentication in IP

TLS

There are overheads in using TLS

- A one-off overhead of (re)writing the application code to use TLS
- A per-connection overhead of TLS setup messages and the associated computation for checking certificates
- A per-packet overhead of data expansion in the encryption (this effectively reduces the MTU)

Security and Authentication in IP

TLS

There are overheads in using TLS

- A one-off overhead of (re)writing the application code to use TLS
- A per-connection overhead of TLS setup messages and the associated computation for checking certificates
- A per-packet overhead of data expansion in the encryption (this effectively reduces the MTU)
- A per-packet overhead in the computation required to encrypt or decrypt the data

Security and Authentication in IP

TLS

There are overheads in using TLS

- A one-off overhead of (re)writing the application code to use TLS
- A per-connection overhead of TLS setup messages and the associated computation for checking certificates
- A per-packet overhead of data expansion in the encryption (this effectively reduces the MTU)
- A per-packet overhead in the computation required to encrypt or decrypt the data

These costs are not huge, but you must make the choice of whether they are worthwhile

Security and Authentication in IP

TLS

Big providers have mostly moved their services to TLS by default, usually HTTPS

Security and Authentication in IP

TLS

Big providers have mostly moved their services to TLS by default, usually HTTPS

For example, Google now uses it to protect all of Gmail and Web searches

Security and Authentication in IP

TLS

Big providers have mostly moved their services to TLS by default, usually HTTPS

For example, Google now uses it to protect all of Gmail and Web searches

For such a large enterprise there is a significant cost in doing so, but the security gained makes it worth doing

Security and Authentication in IP

TLS

Big providers have mostly moved their services to TLS by default, usually HTTPS

For example, Google now uses it to protect all of Gmail and Web searches

For such a large enterprise there is a significant cost in doing so, but the security gained makes it worth doing

And customers are starting to be more security conscious and are now demanding it be done

Security and Authentication in IP

TLS

Big providers have mostly moved their services to TLS by default, usually HTTPS

For example, Google now uses it to protect all of Gmail and Web searches

For such a large enterprise there is a significant cost in doing so, but the security gained makes it worth doing

And customers are starting to be more security conscious and are now demanding it be done

Exercise Compare using transport layer security against network layer security

Security and Authentication in IP

TLS

The usefulness of TLS does not stop there

Security and Authentication in IP

TLS

The usefulness of TLS does not stop there

OpenVPN uses TLS as a *datalink* layer

Security and Authentication in IP

TLS

The usefulness of TLS does not stop there

OpenVPN uses TLS as a *datalink* layer

That is, it layers IP over TLS to build its private network

Security and Authentication in IP

TLS

The usefulness of TLS does not stop there

OpenVPN uses TLS as a *datalink* layer

That is, it layers IP over TLS to build its private network

It creates a virtual network interface that the OS can pass IP packets to

Security and Authentication in IP

TLS

The usefulness of TLS does not stop there

OpenVPN uses TLS as a *datalink* layer

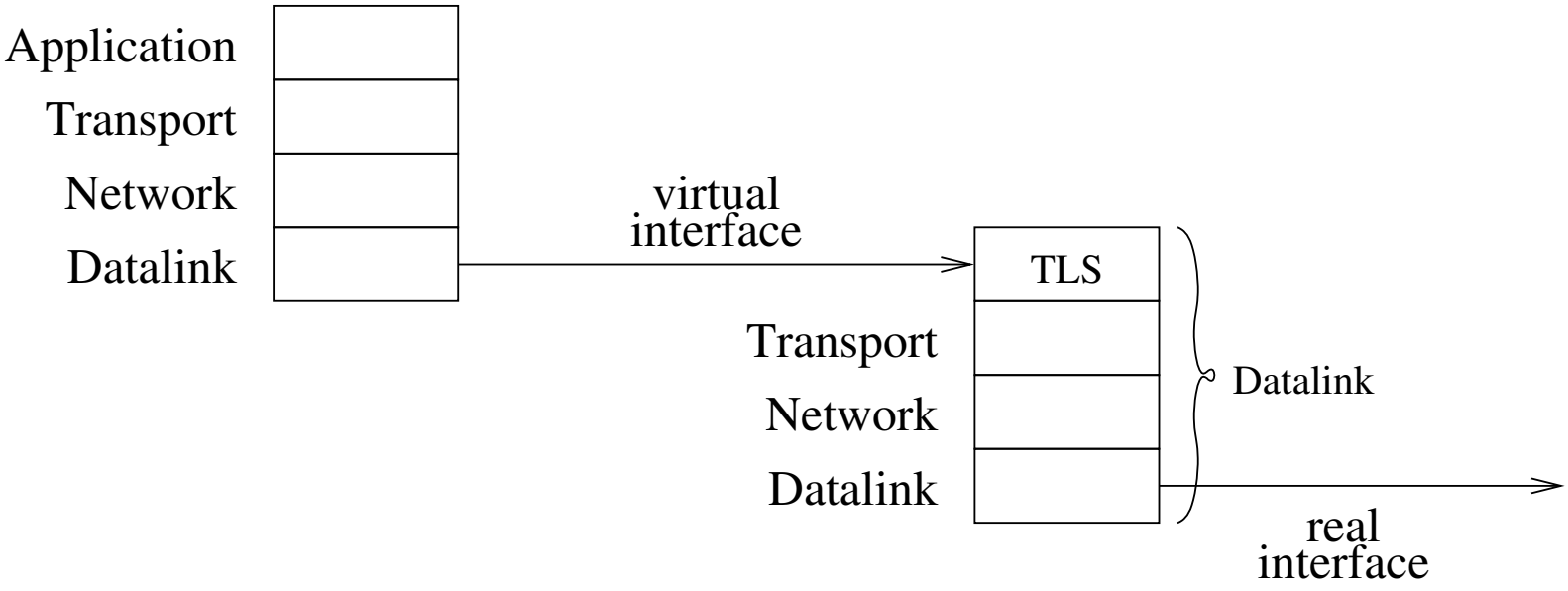
That is, it layers IP over TLS to build its private network

It creates a virtual network interface that the OS can pass IP packets to

The OpenVPN code then encrypts, authenticates and does whatever it needs (using TLS) before handing the result on to a “real” transport layer, usually UDP (as this is a VPN)

Security and Authentication in IP

TLS



TLS

Security and Authentication in IP

TLS

The encapsulated data then goes down through the normal transport and network layers and is transmitted over the real physical layer

Security and Authentication in IP

TLS

The encapsulated data then goes down through the normal transport and network layers and is transmitted over the real physical layer

At the receiving end, the real transport layer hands the data to OpenVPN which decrypts and passes the resulting IP packets to the OS to pass up the rest of the stack

Security and Authentication in IP

TLS

The encapsulated data then goes down through the normal transport and network layers and is transmitted over the real physical layer

At the receiving end, the real transport layer hands the data to OpenVPN which decrypts and passes the resulting IP packets to the OS to pass up the rest of the stack

Of course, it is layering that allows all this to work!

Security and Authentication in IP

TLS

The encapsulated data then goes down through the normal transport and network layers and is transmitted over the real physical layer

At the receiving end, the real transport layer hands the data to OpenVPN which decrypts and passes the resulting IP packets to the OS to pass up the rest of the stack

Of course, it is layering that allows all this to work!

There is a cost of about 10% overhead in practice

Security and Authentication in IP

TLS

The encapsulated data then goes down through the normal transport and network layers and is transmitted over the real physical layer

At the receiving end, the real transport layer hands the data to OpenVPN which decrypts and passes the resulting IP packets to the OS to pass up the rest of the stack

Of course, it is layering that allows all this to work!

There is a cost of about 10% overhead in practice

Exercise Compare these costs with using a Network Layer approach to VPNs, such as IPSec or WireGuard

Security and Authentication in IP

TLS

And now we have the usual benefits of a VPN: user applications can be unsecured (but remember your data is only secure while inside the VPN, not if the final destination is somewhere in the wider Internet)

Security and Authentication in IP

TLS

And now we have the usual benefits of a VPN: user applications can be unsecured (but remember your data is only secure while inside the VPN, not if the final destination is somewhere in the wider Internet)

And the usual costs (For the geeks: the TLS code runs in user mode, so the data in each packet has to go between user mode and kernel mode several times)

Security and Authentication in IP

TLS

And now we have the usual benefits of a VPN: user applications can be unsecured (but remember your data is only secure while inside the VPN, not if the final destination is somewhere in the wider Internet)

And the usual costs (For the geeks: the TLS code runs in user mode, so the data in each packet has to go between user mode and kernel mode several times)

Exercise Compare using an insecure login over a secure network against a secure login over an insecure network

Security and Authentication in IP

TLS

And now we have the usual benefits of a VPN: user applications can be unsecured (but remember your data is only secure while inside the VPN, not if the final destination is somewhere in the wider Internet)

And the usual costs (For the geeks: the TLS code runs in user mode, so the data in each packet has to go between user mode and kernel mode several times)

Exercise Compare using an insecure login over a secure network against a secure login over an insecure network

Exercise And what about using a secure login on a secure network?

Security and Authentication in IP

TLS

A web browser looking at a page secured by HTTPS on a VPN on a home network might be layering

> Web page in HTML/CSS over HTTP over TLS over TCP over IP over TLS over UDP over IP over PPP over Ethernet over Cat6a