Presentation

Presentation for characters seems like a solved problem, but it’s not: at the very least we need to get people to actually *use* the standard

Presentation

Presentation for characters seems like a solved problem, but it’s not: at the very least we need to get people to actually *use* the standard

Many programmers and programming languages still assume everything is ASCII

Presentation

Presentation for characters seems like a solved problem, but it’s not: at the very least we need to get people to actually *use* the standard

Many programmers and programming languages still assume everything is ASCII

And there’s a huge amount of legacy code and data out there that assumes ASCII

Presentation

Presentation for characters seems like a solved problem, but it’s not: at the very least we need to get people to actually *use* the standard

Many programmers and programming languages still assume everything is ASCII

And there’s a huge amount of legacy code and data out there that assumes ASCII

Exercise Other encodings are available. Find out the encodings used on various web pages from across the world

Presentation

Numbers

Another presentation problem is the byte order used for representing numbers

Presentation

Numbers

Another presentation problem is the byte order used for representing numbers

An integer is typically represented using four bytes: but how those bytes are used varies

Presentation

Numbers

Another presentation problem is the byte order used for representing numbers

An integer is typically represented using four bytes: but how those bytes are used varies

Some machines use *big endian* format: this stores the most significant byte of an integer (the *big end*) at the lowest machine address, less significant bytes at increasing addresses

Presentation

Numbers

Another presentation problem is the byte order used for representing numbers

An integer is typically represented using four bytes: but how those bytes are used varies

Some machines use *big endian* format: this stores the most significant byte of an integer (the *big end*) at the lowest machine address, less significant bytes at increasing addresses

Others use *little endian* format: the least significant byte (*little end*) is stored at the lowest machine address, more significant bytes at increasing addresses

Presentation

Numbers

If a machine receives four bytes 00 00 00 2A, does that mean the integer 42 (hex 0000002A) or the integer 704643072 (hex 2A000000)?

Presentation

Numbers

If a machine receives four bytes 00 00 00 2A, does that mean the integer 42 (hex 0000002A) or the integer 704643072 (hex 2A000000)?

Other arrangements are possible, too

Presentation

Numbers

If a machine receives four bytes 00 00 00 2A, does that mean the integer 42 (hex 0000002A) or the integer 704643072 (hex 2A000000)?

Other arrangements are possible, too

A typical solution so that everyone agrees on order is to pick a single order (*the network byte order*) and always transmit bytes in that order

Presentation

Numbers

When a machine wants to send a value, it converts it to network byte order

Presentation

Numbers

When a machine wants to send a value, it converts it to network byte order

When a machine receives a value it converts it to its native order

Presentation

Numbers

When a machine wants to send a value, it converts it to network byte order

When a machine receives a value it converts it to its native order

The *de facto* order used on most networks is big endian

Presentation

Numbers

A big endian machine has nothing to do when sending or receiving

Presentation

Numbers

A big endian machine has nothing to do when sending or receiving

A little endian machine must reverse the order of the bytes as it sends or receives

Presentation

Numbers

A big endian machine has nothing to do when sending or receiving

A little endian machine must reverse the order of the bytes as it sends or receives

A little endian machine always converts, even when connected to another little endian machine

Presentation

Numbers

A big endian machine has nothing to do when sending or receiving

A little endian machine must reverse the order of the bytes as it sends or receives

A little endian machine always converts, even when connected to another little endian machine

This is simpler than having a protocol to negotiate endianness and having separate chunks of code for each combination

Presentation

Numbers

Then there is the problem for other types of numerical data, e.g., floating point

Presentation

Numbers

Then there is the problem for other types of numerical data, e.g., floating point

Here there is not only the byte order problem, but which and how many bits are used for exponents and mantissas and so on

Presentation

Numbers

Then there is the problem for other types of numerical data, e.g., floating point

Here there is not only the byte order problem, but which and how many bits are used for exponents and mantissas and so on

Fortunately, most have plumped for the IEEE standard floating point representations

Presentation

Numbers

Then there is the problem for other types of numerical data, e.g., floating point

Here there is not only the byte order problem, but which and how many bits are used for exponents and mantissas and so on

Fortunately, most have plumped for the IEEE standard floating point representations

This fixes which bits are used for what, but leaves open the endian question

Presentation

Numbers

Then there is the problem for other types of numerical data, e.g., floating point

Here there is not only the byte order problem, but which and how many bits are used for exponents and mantissas and so on

Fortunately, most have plumped for the IEEE standard floating point representations

This fixes which bits are used for what, but leaves open the endian question

The floating point endian is usually the same as the integer endian, but doesn’t have to be!

Presentation

The End of the Line

It would be easy to think that presentation is easy and is irrelevant or has been solved: not so

Presentation

The End of the Line

It would be easy to think that presentation is easy and is irrelevant or has been solved: not so

For example: how to represent the end of a line in a text file?

Presentation

The End of the Line

It would be easy to think that presentation is easy and is irrelevant or has been solved: not so

For example: how to represent the end of a line in a text file?

- Unix-derived systems use a linefeed (LF, character 10 in ASCII)

Presentation

The End of the Line

It would be easy to think that presentation is easy and is irrelevant or has been solved: not so

For example: how to represent the end of a line in a text file?

- Unix-derived systems use a linefeed (LF, character 10 in ASCII)
- Windows systems use a carriage return (CR, ASCII 13) followed by a LF

Presentation

The End of the Line

It would be easy to think that presentation is easy and is irrelevant or has been solved: not so

For example: how to represent the end of a line in a text file?

- Unix-derived systems use a linefeed (LF, character 10 in ASCII)
- Windows systems use a carriage return (CR, ASCII 13) followed by a LF
- Pre-MacOS X used a single CR

Presentation

The End of the Line

So to copy a file from one system to another you must know whether

Presentation

The End of the Line

So to copy a file from one system to another you must know whether

- it is a text file and so you must do the translations, or

So to copy a file from one system to another you must know whether

- it is a text file and so you must do the translations, or
- it is not a text file, so you should not translate

Presentation

The End of the Line

So to copy a file from one system to another you must know whether

- it is a text file and so you must do the translations, or
- it is not a text file, so you should not translate

If we are still fumbling an issue as simple as this, just think on the general case!

Presentation

The End of the Line

Exercise Read about XDR as an encoding system

Exercise Read about the Multipurpose Internet Mail Extension (MIME)

Applications

Next we should talk about the Application layer — but time is too short to talk about things that should be reasonably familiar to you, such as the Web and email

Applications

Next we should talk about the Application layer — but time is too short to talk about things that should be reasonably familiar to you, such as the Web and email

There are very many applications that run over the IP from the well-known things like the Web and email, to the near-invisible (but very important) applications that do everyday things like serving files or controlling industrial devices

Applications

We could easily spend weeks covering application layer protocols, e.g., HTML, the protocol that fetches Web pages; or SMTP, the protocol that delivers email

Applications

We could easily spend weeks covering application layer protocols, e.g., HTML, the protocol that fetches Web pages; or SMTP, the protocol that delivers email

But Instead we will move to a subject that doesn't have a specific layer, namely security

Applications

We could easily spend weeks covering application layer protocols, e.g., HTML, the protocol that fetches Web pages; or SMTP, the protocol that delivers email

But Instead we will move to a subject that doesn't have a specific layer, namely security

Exercise Read up on your favourite applications and how they employ IP

Security

IP (both the protocol and implementations) was originally developed in a “safe” academic environment

Security

IP (both the protocol and implementations) was originally developed in a “safe” academic environment

So little thought was given to security or authentication

Security

IP (both the protocol and implementations) was originally developed in a “safe” academic environment

So little thought was given to security or authentication

And early code left a lot to be desired in programming habits, giving us some fragile implementations

Security

IP (both the protocol and implementations) was originally developed in a “safe” academic environment

So little thought was given to security or authentication

And early code left a lot to be desired in programming habits, giving us some fragile implementations

But fast development led to IP’s early acceptance and success

Security

And this also meant experimental and poorly debugged code was rapidly incorporated into a large number of systems

Security

And this also meant experimental and poorly debugged code was rapidly incorporated into a large number of systems

So there are generic bugs that tend to appear in many products

Security

And this also meant experimental and poorly debugged code was rapidly incorporated into a large number of systems

So there are generic bugs that tend to appear in many products

Some are fairly benign, such as TTL being used as a hop count

Security

Other are less so and can be exploited, perhaps to

Security

Other are less so and can be exploited, perhaps to

- crash the machine

Security

Other are less so and can be exploited, perhaps to

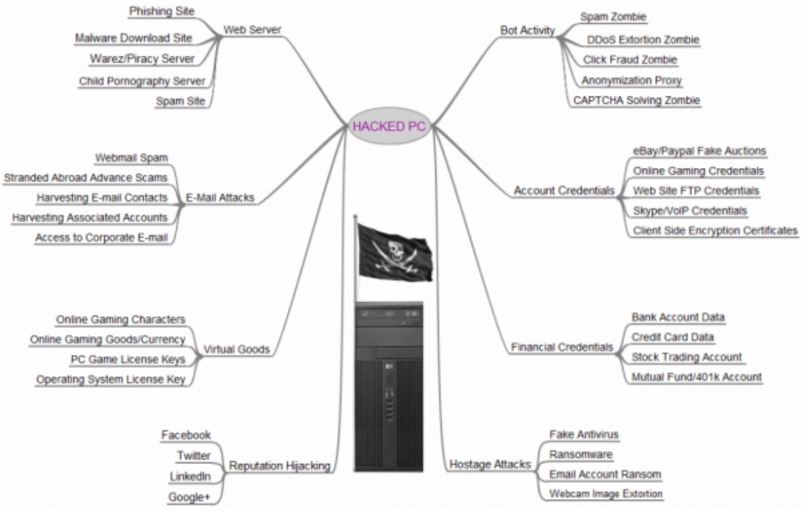
- crash the machine
- tie up the machine with so much bogus data that real traffic can't get through: called *denial of service*

Security

Other are less so and can be exploited, perhaps to

- crash the machine
- tie up the machine with so much bogus data that real traffic can't get through: called *denial of service*
- gain control over the machine, which can then be used attack a more important target or send spam

Security



Uses for a hacked PC

From: `krebsonsecurity.com`

Security

If someone says “The innocent have nothing to hide”, ask them for their credit card number and date of birth

Security

If someone says “The innocent have nothing to hide”, ask them for their credit card number and date of birth

The point is that we all have things to hide from people who would use information to harm us, financially or otherwise

Security

If someone says “The innocent have nothing to hide”, ask them for their credit card number and date of birth

The point is that we all have things to hide from people who would use information to harm us, financially or otherwise

Remember not all those looking at your traffic have your benefit in mind

Security

Today, the Internet is not a safe place

Security

Today, the Internet is not a safe place

People are trying to use it for monetary, political or other gain

Security

Today, the Internet is not a safe place

People are trying to use it for monetary, political or other gain

Or simply wanting to be a nuisance

Security

Today, the Internet is not a safe place

People are trying to use it for monetary, political or other gain

Or simply wanting to be a nuisance

Thus we must protect ourselves against these issues

Security

Technology plays a large part in this

Security

Technology plays a large part in this

But psychology of the users is just as important

Security

Technology plays a large part in this

But psychology of the users is just as important

Why bother attacking a machine when you can attack the human element?

Security

Technology plays a large part in this

But psychology of the users is just as important

Why bother attacking a machine when you can attack the human element?

Such as phoning a support engineer or administrator and pretending to be a user who has forgotten their password

Security

Technology plays a large part in this

But psychology of the users is just as important

Why bother attacking a machine when you can attack the human element?

Such as phoning a support engineer or administrator and pretending to be a user who has forgotten their password

Or sending someone an email and getting them to click a link or run some code

Security

Technology plays a large part in this

But psychology of the users is just as important

Why bother attacking a machine when you can attack the human element?

Such as phoning a support engineer or administrator and pretending to be a user who has forgotten their password

Or sending someone an email and getting them to click a link or run some code

Or simply putting fake news in a Facebook post

Security

Technology plays a large part in this

But psychology of the users is just as important

Why bother attacking a machine when you can attack the human element?

Such as phoning a support engineer or administrator and pretending to be a user who has forgotten their password

Or sending someone an email and getting them to click a link or run some code

Or simply putting fake news in a Facebook post

We shall return to this kind of attack, but shall start with some attacks on the technology

Security

Flooding Attacks

Security

Flooding Attacks

SYN floods. A denial of service attack

Security

Flooding Attacks

SYN floods. A denial of service attack

A TCP connection starts with a SYN. The server sends a SYN+ACK, which the client ACKs

Security

Flooding Attacks

SYN floods. A denial of service attack

A TCP connection starts with a SYN. The server sends a SYN+ACK, which the client ACKs

The server must save a chunk of information about the initial SYN so it can recognise the client ACK as part of the new connection; and the options, like SACK, MSS

Security

A SYN flood is where an attacker sends very many active open SYN segments and never completes the handshake

Security

A SYN flood is where an attacker sends very many active open SYN segments and never completes the handshake

The SYN segments might come from a single source, but more likely from very many hacked computers in a *distributed* denial of service attack

Security

A SYN flood is where an attacker sends very many active open SYN segments and never completes the handshake

The SYN segments might come from a single source, but more likely from very many hacked computers in a *distributed* denial of service attack

The hacked machines comprise a *botnet*, controlled by the hacker(s)

Security

A SYN flood is where an attacker sends very many active open SYN segments and never completes the handshake

The SYN segments might come from a single source, but more likely from very many hacked computers in a *distributed* denial of service attack

The hacked machines comprise a *botnet*, controlled by the hacker(s)

The individual hosts are sometimes called *zombies*

Security

Each SYN received consumes resources on the server that are not released until a suitable timeout period has passed

Security

Each SYN received consumes resources on the server that are not released until a suitable timeout period has passed

Thus the server can run out of resources and not be able to respond to real connection requests

Security

So the overload reduces the level of service for genuine users, often to zero

Security

So the overload reduces the level of service for genuine users, often to zero

This has been used many times, particularly in extortion attacks against commercial (e.g., betting) sites to get them to pay a ransom

Security

So the overload reduces the level of service for genuine users, often to zero

This has been used many times, particularly in extortion attacks against commercial (e.g., betting) sites to get them to pay a ransom

These days also used to exert political pressure against companies, people, or governments

Security

So the overload reduces the level of service for genuine users, often to zero

This has been used many times, particularly in extortion attacks against commercial (e.g., betting) sites to get them to pay a ransom

These days also used to exert political pressure against companies, people, or governments

A DDOS attack might be several GB/s of SYNs: attacks of TB/s are becoming more common

Security

> *Since the start of the [ransom DDoS] campaign, show-of-force attacks have grown from 200+ Gbps in August to 500+ Gbps by mid-September, then ballooned to 800+ Gbps by February 2021*

Akamai

Security

Note: the return addresses on the bogus SYNs will be forged to implicate some other machine(s) in the attack

Security

Note: the return addresses on the bogus SYNs will be forged to implicate some other machine(s) in the attack

And the server's handshake ACKs would go to it, instead

Security

Note: the return addresses on the bogus SYNs will be forged to implicate some other machine(s) in the attack

And the server's handshake ACKs would go to it, instead

Thus flooding a secondary target or targets

Security

Remedies include the server more aggressively dropping half-open connections when resources are low

Security

Remedies include the server more aggressively dropping half-open connections when resources are low

Say oldest first, or at random

Security

Remedies include the server more aggressively dropping half-open connections when resources are low

Say oldest first, or at random

Real connections might get dropped, but since most of the SYNs are bogus, the probabilities are that attack connections are dropped

Security

Alternatively, use *syncookies*

Security

Alternatively, use *syncookies*

Store no information for a new connection on the server, but encode it in the server's initial sequence number (ISN) for this connection

Security

Alternatively, use *syncookies*

Store no information for a new connection on the server, but encode it in the server's initial sequence number (ISN) for this connection

So the ISN is not random, but now encodes some information: it is called a *syncookie*

Security

Alternatively, use *syncookies*

Store no information for a new connection on the server, but encode it in the server's initial sequence number (ISN) for this connection

So the ISN is not random, but now encodes some information: it is called a *syncookie*

When (or if!) the client ACK gets back, we can decode the returned sequence number to retrieve the information

Security

Alternatively, use *syncookies*

Store no information for a new connection on the server, but encode it in the server's initial sequence number (ISN) for this connection

So the ISN is not random, but now encodes some information: it is called a *syncookie*

When (or if!) the client ACK gets back, we can decode the returned sequence number to retrieve the information

Now resources can safely be allocated to this presumably valid connection

Security

Alternatively, use *syncookies*

Store no information for a new connection on the server, but encode it in the server's initial sequence number (ISN) for this connection

So the ISN is not random, but now encodes some information: it is called a *syncookie*

When (or if!) the client ACK gets back, we can decode the returned sequence number to retrieve the information

Now resources can safely be allocated to this presumably valid connection

This is good as it consumes no resources in the server until they are definitely needed

Security

But it is tricky to encode enough information in the 32 bits of the ISN

Security

But it is tricky to encode enough information in the 32 bits of the ISN

And it must be encrypted to prevent spoofing

Security

But it is tricky to encode enough information in the 32 bits of the ISN

And it must be encrypted to prevent spoofing

Also it is not big enough to include any negotiated options, such as “SACK available”

Security

But it is tricky to encode enough information in the 32 bits of the ISN

And it must be encrypted to prevent spoofing

Also it is not big enough to include any negotiated options, such as “SACK available”

So syncookies are only used when the load gets high

Security

But it is tricky to encode enough information in the 32 bits of the ISN

And it must be encrypted to prevent spoofing

Also it is not big enough to include any negotiated options, such as “SACK available”

So syncookies are only used when the load gets high

Optional features, like SACK, are not used under SYN attack

Security

But it is tricky to encode enough information in the 32 bits of the ISN

And it must be encrypted to prevent spoofing

Also it is not big enough to include any negotiated options, such as “SACK available”

So syncookies are only used when the load gets high

Optional features, like SACK, are not used under SYN attack

The loss of SACK is no big deal when we have to cope with a SYN flood