-A Serious Newbie's Guide to the Underground v2-

By

ratdance

Aviator753

Killab

Mls577

Table Of Contents

Why this Paper, this Book?

A Hacker is a master of his environment, of his domain. Understands, in an intimate depth the world he is in, the computer he is on, the servers, workstations he "speaks" to and "speaks" to him, not just the GUI and pretty colors he gazes at, daily, thinking he understands his computer, but what is behind it all, what makes it "breath" and how it "breaths", from the LED's to the Beeps, to the data rushing in and through his lines, and how it arrives and is received, what happens what sent and received.

Respect where Respect is Due

I (ratdance) began this paper as a solo project as I am prone to do. In a very short period of time, my former students Killa, Aviator and MLS577, had jumped in to take over what began as some of my work, to unintentionally taking over my paper and making it a Book. And for this, I thank them all for their time and effort and in joining my Digital Crusade to Educate the Hackers that will be. There is no bigger a thank you or display of appreciate for my existence, than this. Thank all of you, thank #suidrewt, thank Hugo Cornwall and Lloyd (The Mentor).

Preface

Sometime ago, I had posted (along with a series of other docs and code) a .doc on http://haxme.org/for...pic=10155&st=10 that spread all over the net. The problem with this doc is that I intentionally didn't direct its readers to what they could read and where to begin. I really wanted others to think for themselves. And so..I will give my personal honest opinion on what I feel the Neophytes to be, should read and follow up on.

Our (digital underground) has a deep history dating back as early as 1971..and with regret, most of today's neophytes carry on to never be aware of where it all began and had led to:

http://en.wikipedia...._hacker_history

_____I strongly encourage that not only do you reference this wiki or any like it, but research what it shows you. i.e: Captain Crunch , learn more about him and why is he a big deal(and all else in this link)

http://www.mithral.c.../manifesto.html

_____Who wrote the Manifesto that is adhered to by the underground to this very day. Loyd Blankenship did..please reference and learn more

http://www.textfiles...ext/MODERN/hhbk

_____The Hackers Handbook...our bible. very, very, very, very, very few actually take time to read this. The challenge I pose to you and your mind is, what do you not see in this "Bible", there is a lesson in the answer and Hugo did it for a reason.

http://www.scribd.co...on-2-0-Module-4

_____Rhino9's reply to HHK, with technical information and intro to Remote Attacks via NetBIOS and null session.

http://www.hackcanad...isc/wardoc.html

random R9 phun

http://www.phrack.com/

_____Phrack..the undergrounds EZine(Electric Magazine) again, most aren't even aware that it exists, let alone read it. Outdated? no..not at all. it's still maintained as relevant information is provided by the underground (us) for Phrack to share with the rest of the underground. If ever there was a way to get your digital handle out there..this is it. IF you don't care to begin from Phrack 1, then at the very least, do begin at Phrack 49 File 14, by Aleph One; learn about Buffer Overflows and who Aleph is.

http://www.2600.com/

_____If you read the Hackers Handbook(then respect to you) then you know what a 2600 is...thus my next link

http://www.telephone...ephreaking.html

_____Yes, I know, a lot to read so far..I..no..WE(all hackers of the digital underground) want for you to understand the world you're now in.

What IS a Hacker?

Contrary to what the current existing Neophytes badly want to believe, a Hacker is not one who gleefully discovered remote HTTP based attacks such as RFI's, LFI's, SQLi's, JRI's,etc, as ALL of these are wildly outdated and and are near to scripted. Bottom line is, a 'Hack' is NOT a security compromise aka: breaking into a Server, Workstation, Database, short of that being done via a Hack. Let me explain...

When a Hacker locates(and usually at random) a 0day, that is a NON Documented and UNKNOWN remote or local vulnerability, NOT posted ANYWHERE before the moment of discovery, THEN writes(Codes) the required exploit code to exploit the found 0day, and executes, successfully, his personal written code for the confirmed 0day, and gains a user or root shell, then...it is a Hack. NO running to Metasploit or loading BackTrack OR ANYTHING like them OR ANY vulnerability scanner that was NOT written by you, is N O T a Hack, it is being a skidiot. YOU did NOT write ANY of that code OR even write(NO CLICKING ON SLAX TO MAKE AN PRECOMPILED OS DOESNT COUNT) the OS.

Prime examples of what a Hack IS, would be Theodore de Raadt, the writer(coder) of OpenBSD UNIX(His OS Hack) http://www.openbsd.org/ , Fyodor the writer(coder) of NMap(Sockets Hack) http://insecure.org/ or Linus Torvalds the father of Linux(OS Hack). NO because their creations are THEIR Hacks, doesn't mean YOU using it makes it a Hack, it leans back to sriptkid action if you're using THEIR work to get your lack of a Clue, accomplished.

When you take something that exists and decompile to rewrite to make it bigger, better, faster or write/make something new of significant use and recompile, and it works, it's a Hack.

When you can change your Router, Modem, Mouse, Keyboard, Tower and ANY peripherals with in to do something significant, beyond what it does normally? It's a Hack
bottom line is: Take it, Break it, Make it Bigger, Better or Faster, rebuild it, and it's a Hack, whether it be code or Hardware,etc, it's a Hack.

The common user will say "look what this can do"

The Hacker will say "But look what I can make it do..."

Who are or what is the Elite?

Those that can take what exists and make it bigger than it self and the whole world recognizes it's existence..

Theodore De Raadt founder of BSD UNIX...is Elite
Fyodor of NMAP...is Elite
Linus Torvalds of Linux..is Elite

and the list goes on far past that, but I'm confident you get the idea.

What is a "Flame"?

Stemming from Medieval England the burning of those who had a belief that was unbefitting of present societal religious and social standards (most often 'Witch!'), the accused was very often tied to a pole, dowsed in oil and set on fire (i.e: Flamed) So, it is strongly advised a Neophyte to the underground RTFM to learn to abide by the standards and be aware of the world they are now in, lest they be subject to constant Flames.

About Coding

In my previous EZine, I didn't push too much on programming. However, now that I'm here, I do encourage Low Level Languages. The closer you can get to talking, learning and understanding your computer hardware, the better. REALLY learning the world you're in. ASM..Assembler...MASM Microsoft Macro Assembler, TASM Turbo Assembler, NASM Netwide Assembler. As of today, 2010,
NASM http://www.nasm.us/ is where to go and learn both x86 and x64 ASM.

something to get you started? if you're still reading this? then OK, fair is fair:

http://asm.inightmar...e=1&location=12

http://www.emu8086.com/

_____contains a full ASM doc, that I appreciated. I wanted to direct you to a Linux/UNIX ASM, but I am aware most aren't there..yet

Along with ASM is C and Perl & Python. I learnt x86 TASM before Perl, but I suggest C, Perl or Python before ASM.

Along with coding, I very,very,very,very, strongly urge you learn Sockets and how to use Sockets with your code of choice. coding is pointless if you can't get connected to the net via TCP/IP(transmission control protocol), UDP(user datagram protocol),ICMP(internet control message protocol). really, really become intimate with TCP/IP and it's nuts & bolts such as the FLAGS, what they are and how they work.

There is little wrong with using others programs to get a task done. There is EVERYTHING wrong with using others code/programs and taking credit for what it did/does for you.

What is a Scriptkiddy aka: Skidiot ?

The term Skiddy, Scriptkiddy, Skidiot, etc all stem from the concept of movie actors following their Scripts. A Predetermined action with a known outcome and written by someone other than themselves. Well, a SkIDIOT is the very same. Someone who finds another's code or program and compiles to run or executes it, gains root as it is meant to do, then self proclaims himself a Hacker. Lame.....

And the list goes on far past that, but I'm confident you get the idea.

Wireless Networks

(Written By: Aviator753)

Wireless Networks, WiFi, 802.11x: just a few of the many names for communicating wirelessly between computers. WiFi does the same thing a wired network does, link together multiple computers in order to communicate information.

What's the secret to wireless networking? Radio waves, just like your television, car radio, microwave oven, and cell phone uses, basic two way communication:

1) A computer turns information into a radio signal and transmits it to an Access Point (AP), a wireless router, using its antenna.

2) The router receives the signal, decodes it, and sends it on through a standard physical Ethernet connection.

The process works in reverse as well, from the router to the computer's adapter.

While being similar to other devices that use radio waves, WiFi is different in a few key ways:

-The information is transmitted at 2.4 or 5 Gigahertz (GHz), a much higher frequency than other devices, allowing higher data transfer rates, but at shorter distances

-"Frequency Hopping" allows a considerable reduction in interference and the ability for multiple devices to use the same wireless connection, "Hopping" rapidly between up to three different frequency bands

-the Institute of Electrical and Electronics Engineers (IEEE) provides 802.11 networking standards.. I'll get into those a little later, these standards allow devices to move from one network to another seamlessly.

Now let's get into the different 802.1x standards. As I mentioned before, these standards were created and are maintained/updated by the IEEE (pronounced 'eye-triple-E') for carrying out wireless local area network (WLAN) computer communication in the 2.4, 3.6, and 5 GHz frequency bands.

<u>802.11x Standards</u>

802.11 - 2.4 GHz - June 1997 - allowed 1-2 Mbit/s data rates at a maximum distance of 100 meters/330 feet. Today, however, it is no longer used and nearly completely obsolete.

802.11a - 5 GHz - September 1999 - up to ~25 Mbit/s achievable speeds, distance up to 120m/390ft. The higher frequency allowed transmissions with drastically lower interference, but also a lower effective range, (the signals are more easily absorbed into walls/floors/etc).

802.11b - 2.4 GHz - September 1999 - up to 11 Mbit/s, distances up to 140m/460ft. Devices utilizing 802.11b suffered much interference from other products using the 2.4 GHz frequency band, including microwave ovens, cordless telephones, baby monitors, and the like.

802.11g - 2.4 GHz - June 2003 - up to 54 Mbit/s, distances up to 140m/460ft. Also suffers much interference, but was widely adopted by consumers prior to becoming adopted in June '03 Still widely used today, but is being replaced/phased out by the following:

802.11n - 2.4/5 GHz - October 2009 - up to 150 Mbit/s, with distances up to 250m/820ft. Improves previous standards by implementing 'MIMO' (Multiple-Input Multiple-Output), multiple antennas located on the transmitter and receiver, allowing much higher bit rates/speeds and distance/range.

Another method to reduce interference among wireless devices is for them to operate on up to 14 different channels: slightly different frequencies, ranging from 2.4000 to 2.4835 GHz. Channels 1-13 spaced only 5MHz apart, with channel 14 being 12 MHz above channel 13.

Wireless Security

Well, it's about that time, let's talk about WIFI security. In order for any wireless network to use any encryption, the client and the server must have the same encryption on or they will NOT be able to communicate. (Think of it as a person speaking Chinese trying to communicate with somebody that speaks French)

There are 4 main settings you can use as far as security goes for a router: Open, MAC filtered, WEP, or WPA/WPA2.

-Open Network - Has absolutely no security, anyone can connect and access the totally unsecured connection.

-MAC Filtering - Most Access Points (AP) have some sort of MAC Filtering that allows the administrator to only permit certain computers to connect, however MAC spoofing utilities are widely available and can easily bypass this protection.

-WEP - Now we get to the fun one, also called 'Wireless Equivalent Privacy'. WEP was the ORIGINAL ENCRYPTION STANDARD for wireless. WAS, being the keyword there. Beginning in 2001, major flaws were found and exploited. Many many many open source utilities were created to examine and decrypt the packets, successfully and quickly breaking into a WEP-secured network [In 2005, a group from the United States FBI broke a WEP encrypted network in less than 3 minutes] Another issue in WEP is key management: if enough packets can be intercepted, a person could brute force it in a matter of hours. Among WEPs weaknesses, it is still much better than an unsecured, open network: at least it keeps out mindless leeches.

-WPAv1 - 'Wi-Fi Protected Access' WPA is special in its way as it implements the security of a four-way handshake. It was created to replace WEP, in 2003 it was officially announced the successor of WEP. WPA also allows AES-CCMP algorithm, drastically increasing the protection over WEP. WPA uses a Pre-shared Shared Key (PSK) to establish protection using an 8 to 63 character password. WPA-PSK can be brute forced using an offline dictionary attack by capturing the four-way handshake when the client automatically reconnects when DE-authenticated

<u>WAPv1 Additions</u>

There are a few things that can be used alongside WPA in order to strengthen its security.

1.
-TKIP - 'Temporal Key Integrity Protocol'(pronounced 'tee-kip') TKIP uses a per-packet key mixing with a message integrity check..effectively avoiding the problems of WEP
-EAP - 'Extensible Authentication Protocol' EAP is NOT an authentication mechanism, it is an authentication FRAMEWORK. It provides common functions and negotiation of EAP methods [there are about 40 of them] EAP is not a protocol, per se, instead it defines message formats that are used for authentication.
-LEAP - 'Lightweight Extensible Authentication Protocol' LEAP is basically WEP that has been upgraded to minimalize its flaws and a sophisticated key management system. LEAP also uses a MAC Address Filtering/authentication (though, as I mentioned in a prior section, MACs can be spoofed to bypass this filtering)
-PEAP - 'Protected Extensible Authentication Protocol' PEAP allows for secure transfer of information\keys\etc without a certificate server.

-WPA2 - The primary difference between WPAv1 and v2 is the implementation of AES-CCMP (Advanced Encryption Standard-Counter Mode with Cipher Block Chaining Message Authentication Code Protocol...say that 5x fast)...which is an Encryption Protocol meant to replace TKIP. WPA2 also supports EAP\LEAP\PEAP\etc

At this point in time, WPA2 is the most secure, publicly available Wireless security Encryption.

Other Wireless Transmission methods

-Bluetooth is an open wireless protocol for transmitting data over short distances, using very short-length radio waves.. also able to connect multiple devices simultaneously, overcoming synchronization problems

-Ad-hoc - Refers to a wireless network that requires no Access Point. Computer to Computer, Cell Phone to Cell Phone, Device to Device. Each device is essentially a router and an adapter, a server and a client. Each device is independent of all others and can communicate with any other adhoc enabled device.

Cryptography

(Written By MLS577)

Encryption is converting data from a plain text into what is called cipher text. Cipher text is the information that has been transformed or encrypted using an algorithm or cipher into a character string. This data can be converted back into its original form or reverse the process is called decryption. To recover the original data that was once in plain text you need the decryption key, the decryption key will undo the process which encrypting the data has done. A decryption key is what determines the output of either the cipher or algorithm. If no decryption key is supplied then simply there will be no end result, it will simply do nothing. The cipher can be attempted to be broken through mathematics along with some other techniques that will be explained in the remainder of this book. In some cases though, you may not even need to find out the key to the cipher to gain access. The more complicated the encryption the more difficult it is to break the code (cipher).

Cryptography dates back to when communication was first established. It could be something as simple as inverting letters in the alphabet or something complex as AES encryption. Cryptography's purpose was and still is to keeping sensitive information private to others that are unauthorized to receiving this information. Though as things progress there was a need for more complex and sophisticated ciphers to protect such data. This is how some of the modern encryptions of today came to be.

Ciphers that will be cover in this book include:

- **DES and Triple DES**
- **LM**
- **NTLM**
- **MD2, MD4, MD5**
- **SHA-0 through SHA-3**

These encryptions can be decrypted through techniques such as:

- **Brute-Force**
- **Dictionary/ Word list Attack**
- **Rainbow Tables/ Cuda**

DES

DES stands for Data Encryption Standard. In 1972 it was decided that there was a need for an algorithm that could properly and securely protect classified and non-classified information by the Nation Institute of Standards and Technology (NIST). The original algorithm was called the Lucifer cipher, which was developed by the IBM Team in 1974. When it was modified from 128-bit to 56-bit by the NSA (National Security Agency) in 1976, it was then renamed DES. The DES algorithm is a block cipher which is a form of a shared secret encryption. A Shared Secret is a piece of data that can be a password, pass phrase, or randomly chosen bytes that is known by two groups that is over secured communication. It was available for public use in 1977, but only a few years later in 1998 the Electronic Frontier Foundation cracked a DES key in about 3 days. Then in 1999 the Electronic Frontier Foundation cracked the key in 22 hours and 15 minutes, which was a wake-up call that something had to be done about it. After that a new variant of the encryption was implemented called Triple DES, which will be described more below.

Triple DES

Triple DES is as mentioned above a variant of the Data Encryption Standard (DES), which was developed in estimation between 1998 -2000. It is considered that Triple DES is three times more secure than its obsolete in-secure counterpart. Though a downside of this is that Triple DES takes three times longer to compute than a DES but the security vulnerabilities it had taken care of out-weighs the processing time. Triple DES is split up into three 64-bit sub-keys or 192-bits. When the key is implemented it is, if necessary converted into 64-bit parts and then is encrypted in three separate parts which are then put together to make the cipher text. This is what makes 3 DES secure initially, though if not properly implemented could cause problems. An example of this would be if you made two parts the same (two 64-bits). This would turn Triple DES into regular DES. Though Triple DES was considered secure at the time currently it is way less secure then other more common encryptions.

LM

LM stands for LAN Manager. It is associated with Microsoft Window's operating systems, up until windows vista at least. On versions of windows that are above Windows XP have LAN Manager turned off by default. If the password is over 14 bytes (characters) it will not be stored by windows.

The process of converting ASCII into LM cipher text is:

1) Convert ASCII to all uppercase.

2) The password is split into two 7-bit hashes (cipher text).

3) The two 7-bit hashes are then converted a DES key (64-bit).

4) Each key is then DES-encrypted to constant ASCII which results in two 8-bit cipher text.

5) Those two parts then create a LM hash.

NTLM

NTLM stands for NT LAN Manager. NTLM is a variant of the original LM. It is an authentication protocol used on networks running the windows operating systems. NTLM has a lot more security than its counterpart LAN Manager Including secure login credentials using a challenge and a secure means of authenticating without the need for sending the password insecurely over the wire. NTLM credentials consist of a domain name, user name, and a password that has been encrypted with a one way hash. The way it authenticates without the password being sent over the wire is the system requests a calculation (a random number challenge) to be solved to prove it has access to the credentials. The process of authentication as listed below:

1) The client provides a user name, password (that is encrypted), and domain. Then sends the user name in plain text to the server.

2) The server then creates a challenge and sends it back to the client.

3) The client encrypts the challenge and sends it to the server along with the encrypted password (users). This is also known as a response also known as a challenge.

4) The server then sends the data given from the client to the domain controller. The domain controller retrieves the password from SAM (security Account Manager) and uses that password to encrypt the challenge.

5) The domain controller then compares the two hashes and sees if they are identical; if so then authentication is approved.

Now, within what I have written about NTLM have gone over both NTLMv1 and NTLMv2 as a whole. I left out the specifics of the changes, but I will now describe some of the differences:

· In NTLMv1 the server sends an 8-byte challenge and then the client sends back two 24-byte computation of the challenge given by the server. When in NTLMv2 the server sends a 8-byte challenge and the two 16-byte responses.

· In NTLNv1 the two 24-byte responses were split into two different encryptions. One was LAN Man and the other was MD4. As in NTLMv2 are encrypted in hmac-md5 also the first part is made up of a 8-byte client challenge with a 16-byte response makes a 24-byte part similar to NTLMv1. Then the second part consists of an NT Time format, an 8-byte value, and domain name (which are called NTv2).

MD2

MD2 stands for Message Digest 2. It is an algorithm created by Ronald Rivest in 1989. It was created for 8-bit systems, which is now obsolete to modern 32 and 64 bit systems. MD2 is defined in RFC 1319. The MD2 Algorithm is explained below:
1. The message Digest's length is extended (padded) in bytes so that it consists of 16-bytes long. This step is done in any circumstance including when the message is already equal to 16-bytes.
2. A 16-byte check sum is created of the message after step on is performed. This step uses a 256-byte mathematical permutation taken from the numerals of pi (π).
3. An MD buffer is then created, which consists of a 48-bytes.
4. An Identical 256-byte permutation as used in step 2 creates a process message in 16-byte blocks.
5. Assuming that everything went correctly the modified message digest is outputted.

MD4

MD4 Stands for Message Digest 4. This algorithm was created in 1990 by Ronald Rivest. It essential replaced it's predecessor MD2 (Message Digest 2).It was designed for 32-bit systems. It is used to generate NT hash digests for Windows NT, XP, Vista, and 7. Since MD4 and MD5 only have a few differences I decided to merge them both into one, so if you read below you will find out more information on how MD4 and MD5 work.

MD5

MD5 stands for message digest 5. It is specified in RFC 1321 and was first published in 1991 by Ronald Rivest. It is represented by a 32-digit long hexadecimal number. It was commonly used along with the rest of the message digests up until 1996 that revealed vulnerability that drew people to use other algorithms such as SHA-1, which will be explained a little later. Though the original vulnerability wasn't major it still deterred people from using it and in 2004 a major vulnerability was discovered and

led to many abandoning it.
MD4 and MD5 are described below:

1.The message's length is extended to 448-bits. Even when the message is equal to 448, this step is still performed.

2. The 64-bit message before padding is appended to the end result of the previous step.

3. A buffer is used to create a message digest. The buffer consists of four-words.

4.Three functions are defined with three 32-bit words and spits out one 32-bit word.
5. The message digest produces the output.

According to RFC 1321 the differences are as following:
• A fourth round has been added.

• 2. Each step now has a unique additive constant.

• 4. Each step now adds in the result of the previous step. This promotes a faster "avalanche effect".

• 5. The order in which input words are accessed in rounds 2 and 3 is changed, to make these patterns less like each other.

• The shift amounts in each round have been approximately optimized, to yield a faster "avalanche effect." The shifts in different rounds are distinct.

All credit goes to the authors of RFC 1321 for the “original” differences between MD4 and MD5.

SHA-0

SHA-0 stands for Security Hash Algorithm but its original name was Secure Hash Standard. It was believe to be based on Ronald Rivest’s MD (message digest) algorithms. It was first introduced in FIPS 180 in 1993 by The National Institute of Standards and Technology (NIST). SHA-0 and SHA-1 both creates a 160-bit digest derived from a message with an end length of 264 Bits. SHA-1 was then removed by the NSA; reason given was that there was serious vulnerability that was patched in SHA-1 in 1995 in FIPS 180-1.

SHA-1

SHA-1 is almost identical to SHA-0; only one thing differs between the two. This difference is in the bit wise rotation in the compression function. This was changed by the NSA fixing the so-called vulnerability in security.

SHA-2

SHA-2 like its "family" stands for Standard Hashing Algorithm 2. SHA-2 comes in a few different version including SHA-224, 256, 384, and 512 bit. SHA-2 was first introduced in 2001 in 180-2. These versions were used in quite a few things. In example SHA-256 is used in Debian Linux Software Packages. Other uses for SHA-256 and SHA-512 are used on various *nix OS's for password hashing. Though SHA-2 is widely used, it will most likely be replaced by the finished development of SHA-3.

SHA-3

SHA-3 ; which will replace it's former versions of SHA is still in development. Its set finish date is in 2012. I don't have much more information about SHA-3, so this will unfortunately end the SHA's.

Common Attacks

Brute Force is a method used to break encryption by specifying a number of certain variables to the key and trying each key until the correct key has been discovered. A key is a piece of information that will determine the output. This method is controversial on whether this attack is still useful. Now, yes they're other methods to cracking the key. These methods will be explained later, but some of the reasons for this to be ineffective are because of the time it takes to crack the key. This is a variable that can be changed through a few things like a more powerful system, using certain programs, and utilizing certain things to quicken the time. You can also reduce time by changing the character sets for example if you are brute forcing a hash and you know that the key is only in lowercase letters and numbers. Then it is not necessary to have uppercase characters included in the attack. This will initially save time and when you utilize other shortcuts, you can definitely shorten the time it takes to brute force a key. Another problem with brute force is that in some cases you can't tell whether you have successfully found the correct key or not. These problems might encourage you to pick another method of attack.

Dictionary/ Word list Attack is a method that is trying to break an encryptions key by trying all the words in the Dictionary/ Word list You may say to yourself, why don't say one or the other. Well, I don't say one or the other because frankly this attack can actually be a list of words that happen to be from a actually dictionary like Webster's Dictionary, It may just be a random list of words that you made up yourself for a specific target, or it could just be a random list made up of random phrases and alpha –numeric characters. These may also be words of another language for example if you speak English but want to pick something that somebody might not guess and you decide to use a word from another language such a Spanish. Then that might make your password more secure.

There are many word lists that can be found all around the web, it just takes a certain search on your favorite search engine and you have found a word list that you were looking for. This method can be fast depending on how far down the password is on this list. Though, a bad thing about this attack is that the password that you are looking for may not be on the list, which is really a big waste of time on your part. To increase your chances you should do some research on the target before you do this attack to help cut time. Also as mentioned above, some programs and utilities can quicken this process. One other thing that might be already built in to some programs is that it will take the word and invert it or scramble it to try to make different words. This might increase your chances on the effectiveness of the attack. Remember to do some research on the target and pick a word list that fits that situation in order to increase your chances and save time.

Operating Systems

(Written by ratdance)

Again in my original .Doc, I didn't encourage any ONE OS...and so..now..I will. http://www.slackware.com/ is by far the oldest and still maintained Linux OS, on the Net, and I guarantee most if not all Old school Hackers are on it and true aspiring hackers are headed towards it.

Then we have http://www.debian.org/ that is a Linux based on a UNIX such as http://openbsd.org/ , http://www.freebsd.org/ , http://www.netbsd.org/ I would like to take a moment to point out that OSX is in fact built on http://www.freebsd.org/ and when you spawn a BASH shell on OSX, it is a FBSD UNIX shell you're now in. I really want to get on a soapbox and preach about OBSD, FBSD and NBSD, but will refrain.

Security

(Written By: ratdance)

It's a fickle subject, security(or lack there of) Compromise.. allow me to take a moment to clarify something... where did the phrases "Black hat" and "White hat" even come from? well, dating back to the Cowboy's & Indians or Western, Movies, you would notice the social periah's(bad guys) were in Black hat's, while John Wayne and the stereotypical were in White hat's

Now, contrary to popular belief, this doesn't make us Black hat's bad anymore than it makes a White hat, good. We just take a different approach to matters.
Example:
I my self, after I've found a vulnerable server/workstation, will go ahead and bounce from a 'daisy chain' of Proxies, Wingates and VPN's then tunnel my exploit code(most often coded by me) to said server/workstation. Gain root or elevate said access to root, then create an account or add a key logger for purpose of gaining an existing users account, or to allow me to rlogin,in..hmm..OK,because I always preach about teaching what you preach about, allow me to elaborate on RLOGIN:

RLOGIN

(Written By: ratdance)

NOTE: this is NOT the only means to remote access via UNIX commands,and is only an example. Please do take the time to learn UNIX/Linux and it's commands:

Rlogin requires the user to have a file in their home directory that tells what system they can receive the rlogin from. In this file .rhosts it would look like this:

user name host name (or) host name

If you were to add to this file + + it would let any user from any host login without a password.

The file would look like this:

```
----- cut here ------
+ +
--cut here ------
```

if they already had entry's you could add the + + under their host names, but remember now they would notice seeing they would now be able to rlogin without the password. You would be targeting people that did not already have a .rhosts file.

then from your shell, it would be:
rlogin remote_host name Enter

OR

rlogin -l <compromised user acct name> Enter

WinGate

(Written by: ratdance)

Was/is a software made by Micro$oft, that is a near equivalent to a proxy. like netcat, telnet or SSH, via BASH or DOS CMD Prompt, then once connected, it is as mindless as entering the IP of the IP you next wish to connect to. And so you can imagine it's ease of use in chaining gates and proxies,together. Wingates, by default, run on port 23, but often are reconfigured to a higher location past the IANA 1024

IANA

(Written By: ratdance)

http://www.iana.org/...ts/port-numbers Internet Assigned Ports Authority. This is who decides what runs on what port or port range.

But once again, I digress.

The Black hat's seemingly malicious nature,is not inherently so. Granted, many of the newer generation will use it to justify unrequired DoS or DDOS attacks, virii, trojans,etc.

I would like to point out that DoS attacks have an authentic purpose. In example: If I get a remote & administrative access to an M$ machine, then set up a new account or program, often enough it is required that, that machine reboots before the new acct or program comes into full play..and so, should it be we were unable to manually reboot said machine, we very well may use a form of DoS/DDoS to it to do so. This is indeed a rare event, however I have experienced the need to do so in the past.

Defacements? Guilty on several accounts. It may very well be that once we have complete our root job, that we have quietly cleaned up all logs and any incriminating evidence, and don't feel the need to do the otherwise incompetent Admins job of securing his obvious insecure machine.

Further more, not wanting to lose our newly acquired user/root shell to another Hacker or...*sigh* some random hapless skiddy, we will email the admin from a spoofed email(someone remind me to come back and lecture on SMTP spoof), or via his own email from compromised machine. There is no rule written in stone, but on average we tend to wait from 3 days to 1 week before we send a 2nd email warning of said vulnerability, then same period before we flat out deface in order to scream to both to the public and admin "YOU HAVE A REMOTE HOLE LARGE ENOUGH TO PARK A MAC TRUCK INTO" and back up the index to .old and mention the same in the defacement. Most put up some impressive graphics. In an earlier time, we would all send our defacements to http://www.attrition...ity/commentary/..in fact, some of my crews defacements are buried somewhere in here. suidrewt
ex: http://attrition.org...ameri-soft.com/ , there are quiet a few..but I don't think wade will take kindly to me seemingly encouraging defacements. Some get really graphic.

The entirety of my point being,root shells and defacements means little else other than you've invested some time into researching remote & local vulnerabilities and merits little to no recognition. At best, you can use it as an example to teach how to carry out a 0day you have discovers....well OK, if you have your own 0day? THAT will get you much merit and respect.

0Day/Oh'Day

A local/remote vulnerability that has been discovered, exploited, documented and released with in a 24 hour period. it is 0 days old.

Hats

All this raises a question about the Role of the White hat community. Fact is, White hat's are significantly more productive than the Black hat, as they oft will find, report, fix then document and make information publically available to learn. such key sites as http://www.securityfocus.com/ are a gleaming example of the White hat role in the underground. Note: do sign up Bugtraq for daily vulnerability information via daily emails. You want to think twice before selecting ALL catagories as they come in swarms and are rarely all covered in less than a few hours, daily.

The Black hat's issue with the White hat release of information is that we(Black hat's) are a firm believer in 'Full Disclosure' of information, as such sites aswww.milw0rm.com where documentation and code with full explanation for How and Why the exploit is and works.

The White hats mean to control the damage and educate, the Black hat's mean to educate via damage as an example. Yes even via release of Virii, the point is to make a point of human ignorance or weak software, vulnerable hardware..and the fairly true quote "There is no cure for Human Stupidity"

Now, don't see this as a digital war between Black & White. Very, very often you will see Black and White hat's rehabilitating in the same IRC channel, Forums, DefCon's,etc. My self as an example, a certifiable Black hat educating on an otherwise White hat forum. There is little choice but to agree to disagree for both sides.

This re-invokes a prior point..DoS/DDOS. Any TRUE Hacker, Black or White hat, will be NO fan of a malicious pointless DoS or DDOS Attack as it is a blatant Denial of Information. If the Blacks and Whites have ANYTHING in common? it is to challenged our own and other's minds, learn and make bigger and better all we come across. I my self am no fan of DoS/DDoS for sake of a weak failed attempt at fame.

All of this...thus far, and I am far from done, raises its own question...."Why?" Why post all this, why fight to encourage or even enforce others to learn more and faster, why make it all a big public scene, daily..

To Quote Isaac Newton(if I may):
"If I have seen further it is by standing on the shoulders of Giants"

When I began in 1983 on my TRS80, dialing(literally...) into BBS's(Bulletin Board Systems), all we had was a static digital forum where we left notes for one another on "how to" of stuff we were all learning as we went. from BASIC programming, Basic Security,how to code our own BBS,etc, the first MB(Megabyte) hadn't yet existed, and all was in KB's. My dial up speed was measured in Baud....Baud....1 kBd = 1,000 Bd(baud) explained as symbols per second or pulses per second. I was, at that time, on a 400Baud internal modem..my modem was built INTO my keyboard as well my video, ram and all peripherals. You couldn't imagine the day I go my 12.2kb Modem! The glee..I was sure I was dialing out at break neck speeds.. then my first 56k.. then ISDN(Integrated Services Digital Network, 50 times faster than standard modems) which is still a dial up, just fastest PPP(Point to Point Protocol) dial up before my DSL(Digital Subscriber Line, 4 times faster than ISDN) and soon after ADSL!(Asymmetric DSL), then my first Oc12!(Optical Carrier 12 is a high-bandwidth "pipe" connection to the Internet operating at speeds 12 x 51.84 = 622.08 megabits per second) Oc48!! then T1(developed by AT&T Bell Labs on Twisted Copper Pair) experience at work, then T3! and..as of this paper, my Cable Ethernet at 7.7Mbps (megabits per second); and my prized memory was 128kb, built into my keyboard, now on 4Gb and 2Gb between SLI'd GeForce's...point being..all of this with in 1 life time and still going. I began at age 10-11, now 37, I've seen and lived from floppy discs, to diskettes, to CD's to DVD's, to Virtual Drives, and it goes on...

"If I have seen further it is by standing on the shoulders of Giants"
All of that Technological progress was established by the generations that be, learning from the experience and notes,documents,posts,RFC's(requests for comment) of the prior generation, allowing the next generation to slingshot forward and progress exponentially faster than the generation prior.

OK, so what do today's community get from it?
We get to live to experience the next step then live to contribute to the progress the next generation makes for the generation coming. I was born into the Kilobytes and am breathing the Terraflops(Floating point Operations Per Second) and Terrabytes. let me show you where I began and am here to experience

-kilobyte ==1,024 bytes; My memory on my TRS80 was 128kb and 400Buad Modem

-megabyte ==1,024 kilobytes;

-gigabyte ==1,024 megabytes;

-terabyte ==1,024 gigabytes; My current HDD is 1.5 Terrabyte from days of my 128kb cache memory

I crunch data and compile code faster than I would ever imagine when I was a teen I up/down load and burn entire movies with in minutes,when I used to wait days to download an mp3 or a .doc file
I code apps now, I used to dream of owning
I am fluent in more Operating Systems and Programming Languages(most low level, at that) than I speak languages.

The old school have the new school to thank for taking the passed torch and running with it, the new school have the old school to thank for running with the torch and passing it.

Yes, I do ride the new generations case, and often..and hard...can you yet see why? I'm not dead,yet, and far from it. I want to live to see what is ushered into the digital world,next, and it will be the new generation of Hackers, that start the process.

RFC

(Written by: Aviator753)

RFC - Request for Comments.. published by the IETF (Internet Engineering Task Force) describing research, advancements, behaviors, and methods that deal with the internet and systems that use the internet. The origin of RFC was in 1969 as part of the ARPANET (Advanced Research Projects Agency Network) project. [As a side note, ARPANET was the very first operational packet switching network.. the original internet] Currently, it is the official publication channel for the IETF, IAB (Internet Architecture Board), and much of the global community of computer network researchers.

In the beginning, only hard copies of the first RFCs were spread around and were written in a informal style, which is now common for RFC drafts prior to approval. In December of 1969, RFCs began being distributed via the newly created ARPANET. RFC 1,called "Host Software", was written by Steve Crocker from UCLA (University of California, Los Angeles), and was published on April 7, 1969. RFC 3 originally defined the RFC series and was attributed to the "Network Working Group" created by Crocker. From 1969-1998, Jon Postal was the RFC editor. After the ARPANET contract expired, the 'Internet Society' acted on behalf of the IETF when it contracted with the USC Information Sciences Institute, networking division in order to edit and publish.

The RFC Editor assigns each RFC a unique number. Once published, a RFC is never modified; Instead, if the document requires a change, the author(s) publish a revised document. Therefore, some RFCs replace others. Together, the RFCs compose a continuous historical record of the evolution of Internet standards. keep in mind, the term 'RFC' is also used by several other groups, however IETF is the best-known. Though most RFCs are authored by groups sponsored by institutions, individuals and small groups also have the ability to present an RFC for publishing.

Not all RFCs are internet standards, each is given one of the following statuses:
1) Informational - This status includes nearly anything, from April 1st jokes to essential RFCs (for example, see RFC 1591)

2)Experimental - Can be IETF document or individual submission; Some documents are not promoted to 'Standards Track' solely because there are no volunteers for the procedural details.

3)Best Current Practice (BCP) - Documents with this status include administrative and other texts considered the official 'rules', but do not affect over the wire data. BCP also includes technical recommendations for how to practice internet standards. (RFC 2827 refers to how to make a DoS attack more difficult) The difference between BCP and Standards Track documents is often unclear.

4)Standards Track - Documents are divided into Proposed Standard, Draft Standard, and Internet Standard

5)Historic - These documents have been replaced with newer, updated versions or been removed from use altogether.

For more details about RFCs and the publishing process, see RFC 2026: http://tools.ietf.org/html/rfc2026

Protocols

AKA: How we fling packets around and across the internet, across and at Networks, to and from LAN(local area network) to LAN, LAN to WAN(wide area network), WAN to MAN(metropolitan area network), etc; How we get an IM(instant messaging) message across to another(i.e: Yahoo to MSN,etc);How we send files to one another and usually not lose anything of the sent file in the process of doing so; How we DoS and DDoS each other off the net like cluebie skidiot champions.

IP

(Written by: Aviator753)

IP - Internet Protocol.. used for transferring data across networks using TCP/IP. IP is the primary Internet Layer protocol and is used for transmitting packets from the source host to the destination host, based solely on their IP Addresses. For this reason, IP defines several methods and structures for addressing hosts. IP encapsulates data from upper layer protocols in order to deliver it to the destination host.

In May 1974, the Institute of Electrical and Electronic Engineers (IEEE) published a paper titled "A Protocol for Packet Network
Interconnection" written by Vint Cerf and Bob Kahn. It described a protocol for transferring packets among nodes. TCP (Transmission
Control Protocol) was the main component of this model and included both connection-oriented connections and data gram services
between hosts. This was later separated into TCP at the connection-oriented layer and IP at the internet-working layer. This model
became known as TCP/IP, though it is also called the Internet Protocol Suite. The most widely used inter-networking protocol is IPv4
(as described in RFC 791 in the year 1981). The successor to IPv4 is IPv6; The most notable difference is the addressing system. V4
uses 32-bit addresses (totaling ~4.3 Billion) while v6 uses 128-bit addresses (totaling 340 undecillion).

Because of IP's encapsulation, it can be used over a mixed network. (one that contains computers connected via Ethernet, token ring, Wi-Fi, etc) As a result of each different network type using its own method of addressing, address resolution is handled by the Address Resolution Protocol (ARP) for IPv4, and the Neighbor Discovery Protocol (NDP) for IPv6. IP Addressing and routing are possibly the most complex part of the Internet Protocol. Addressing refers to how hosts are assigned IP addresses and how sub networks are grouped together. IP routing is performed by all hosts, but most crucially by inter-network routers that typically use either Interior Gateway Protocols (IGPs) or External Gateway Protocols (EGPs) to make IP packet/datagram forwarding decisions across internet connected networks.

IP's design principles assume that the network is completely unreliable at any single network element and it is dynamic in availability of links and nodes. That is to say, It doesn't matter is a router is down, a Trans-Atlantic cable is cut, etc, the information will always find a path to the host that works. However, there is no guarantee of delivery. There is no central monitoring of the state of the network. In order to reduce the complexity of the network, the information in the network is located primarily in the end nodes of each transmission (known as the end-to-end principle). This unreliable system can result in data corruption, lost packets or datagrams, data duplication, out-of-order packet delivery (Packet 5 arrives before packet 4, etc.) The only error prevention IP includes is a check sum that is checked at the routing nodes (however, this discards packets with bad headers instantly) IPv6 abandons the check sum, thus providing a faster transit through routing elements in the network.
The four numbers in an IPv4 Address are called 'octets' because they have 8 positions when viewed in binary form. These octets are used to create classes of IP addresses that can be assigned to a particular business, government, etc. based on size and need. the octets are split into Net and Host (also called Node). the Net section is always the first octet and is used to identify the network an address belongs to. The Host, or Node, identifies the specific computer on a network and contains the last octet.

There are 5 IP classes:

Class A - used for very large networks, includes addresses with the first octet 1 to 126, account for half of all IP addresses, and the first binary number in the first octet is always a 0 (zero)

Class B - used for medium-sized networks, first octet from 128 to 191, also includes the second octet as part of the Net identifier
(other two octets used to identify Host/Node), makes up a total of 1/4 all IP addresses, first binary number is 1 (one) and second is
0 (zero) in the first octet

Class C - commonly used for small to mid-size businesses, first octet from 192-223, include second and third octet as part of the Net
identifier, contains about 1/8 of all IP addresses, first binary number is 1 (one), followed by 1 (one), followed by 0 (zero) in the
first octet

Class D - Used for multicasts, first binary digits are 1-1-1-0 (one-one-one-zero), the other 28 bits are used to identify the group
of computers the multicast message is intended for, includes 1/16 of all IP addresses

Class E - For experimental purposes only, binary digits are 1-1-1-1 (one-one-one-one), the other 28 bits are used to identify the
group of computers the multicast message is intended for, includes 1/16 of all IP addresses

The OSI (Reference) Model In All Its Glory

What is the OSI Model?

"True Story"

So what is the OSI Model? OSI Model stands for the Open System Interconnection Reference model. The OSI model details how information from a software application in Box 1 moves through a network medium to a software application in another Box 2. Or for a bit more technical way to spell it out: an abstract description for layered communications and Box network protocol design. Simply put the OSI model takes a network architecture and chops it up into 7 "Layers" (the layers will be explained in depth later on).

“This 7 layer reference model defines a concept of moving information between networked Box's. It describes how information flows from one end-user application through a network into another application. The OSI model is considered the primary architectural model for inter-Box communications.”-Phillip, speedguide.net

The model doubles as a how-to so to speak. It’s a good place to start if you're looking to get into anything as far as networking, having a good understanding of the OSI model is *essential*. Especially if you're planning on getting any type of Cisco certification.

Think of the model as a factory (a very small one) and the layers are 7 people, whom work there. Let’s say they make toys, now each person has a specific task that they must complete in order for a toy to reach its final stage. The model in itself is the “housing”, so to speak for the various layers to conduct their business and operate in.

Where Did The OSI Model Come From?

Personally I see the history and evolution of anything, the most important part of the story. How can you truly know something or about something unless you're sure of its point of origin and its history? The OSI Model was designed by the International Organization for Standardization (ISO). Although the work for a network architecture started before the mid 1980`s. The International telegraph and Telephone Consultative Committee (or CCITT) and the International Organization for Standardization both took part in developing a standardized network architecture.

Almost all of the aspects of the OSI model (the design anyway), came from experiences from the CYCLADES network which was also very influential in internet design.

So How does this actually work?

This question as you may have guessed by now is not easily answered by a simple one-liner (would have been nice though). Now you know basically where its from and what's in it. Lets take a look at how it works.

Protocols enable any form of connectivity via a Box to interact with another Box using the same layer, for a task that is the same and/or different. Different layers of the OSI model can and often do interact with another layer from another Box, due to the fact that the OSI model is the network architecture for a majority of the Box's out there (mostly post-1970`s).

The above simply summarizes the OSI model, lets go a bit "deeper into the rabbit hole" to see where it leads and EXACTLY what it is its doing.

The Layers

There are 7 Layers in the OSI Model and they are as follows:
1. Physical
2. Data Link
3. Network
4. Transport
5. Session
6. Presentation
7. Application

Don't be alarmed it's not that much. Besides this is just the introduction to the layers, the *Syn* of the conversation if you will. Each layer has a different job to do at different times but like a link in a chain they all rely on each other to get said objective completed. We are going to go layer-by-layer and explain its function, describe said layer in detail, and tell you what data is called in certain layers. So let's begin!!

The Physical Layer (Layer 1)

"Without me you would be nothing".

Lets start with layer one. Layer 1 is known as the Physical Layer. Well why? It doesn't take much imagination to figure this one out. The physical layer defines electrical and physical specifications and requirements for the devices in your Box.

That's it? No, it's not. Gotcha, the physical layer does a lot more than monitor electrical signals it also, defines the relationship between a device and a physical medium.

This layer is also responsible for the layout of pins, voltages hubs repeaters, host bus adapters, cable specifications, network adapters, telephone network modems, IRDA

(Infra Red Data Association), USB (Universal Serial Bus), Ethernet, DSL (Digital Subscriber Line), Bluetooth, Fire Wire, and ether-loop. Furthermore the

The Physical layer of the OSI model defines the means of transmitting raw bits of data rather than logical data packets over a physical link and network nodes. The bit stream may be grouped into code words or symbols and converted to a physical signal that is transmitted over a hardware transmission medium. The physical layer provides an electrical, mechanical, and procedural interface to the transmission medium.

Its main function is media, signal and binary transmission.

This layer is essential due to the fact that it is a fundamental layer underlying the logical data structures of the higher level functions in a network. In layman's terms this is your foundation for your house.

Jumping a bit ahead, the Data link layer and the Physical layer ARE NOT THE SAME!

If you will imagine the physical layer as concerned primarily with the interaction of a single device with a medium. As opposed to the Data link layer who's greater concerned more with the interactions of multiple devices (at least two) with a shared medium.

Essentially the physical layer will tell one device how to transmit to the medium at the same time, the physical layer will tell another device how to receive from it. In 9/10 of these scenarios it does not tell the device to connect to the medium.

Some of the major jobs and functions of the physical layer are listed as follows;

1. Establishing and terminating of a connection to a communications medium.
2. Participation in the process whereby the communication resources are effectively shared among multiple users. For instance contention resolution and flow control.
3. Modulation, or conversion between representation of digital data in user equipment and the corresponding signals transmitted over a communications channel. These are signals operating over the physical cabling such as copper and optical fiber or over a radio link
4. Auto negotiation
5. Bit-by-bit or symbol-by-symbol delivery
6. Modulation
7. Line coding
8. Its synchronization synchronous serial connection
9. Start stop signaling and flow control in asynchronous serial communication.

10. Circuit switching
11. Multiplexing
12. Carrier sense and collision detection utilized by some level two multiple access protocols.
13. Equalization filtering, training sequences, pulse shaping, and other signal processes of physical signals.
14. Forward error correction
15. Bit rate
16. Point-to-point, multi-point or point-to-multi-point line configuration
17. Physical Networking Topology, for example bus, ring, mesh or star network
18. Serial or parallel communication
19. Simplex, half duplex or full duplex transmission mode

The physical layer is often referred to as PHY.

The Data Link Layer (Layer 2)

"Knock, Knock, Who's There?"

Now that we've fully analyze the first layer of the OSI model let's move on to the second layer. The second layer is commonly known as the data link layer. This layer transfers data between nodes on the same local area network. It provides the functionality and procedural abilities to transfer data between network entries and may also even provide the means to detect and possibly correct errors that might occur in the physical layer! It may also transfer data between adjacent network nodes in a wide area network. Some examples of data link protocols would be Ethernet for local area networks (multi-node), the point-to-point protocol (PPP), HDLC (High Data Link Control), and ADCCP (Advanced Data Communication Control Procedures) for point-to-point protocol.

The priority of the data link layer is local delivery of frames between devices on the same LAN (Local Area Network). Data link frames are what these protocol data units are called.

These do not leave the limits of the local network. Really the data link protocols focus on local delivery, addressing, and media arbitration. The data link layer can be compared to a cop due to the fact that it endeavors to attribute between parties contending for access to medium.

Delivery of frames by layer two devices is effected through the use of unambiguous hardware addresses. A frames header contains source and destination addresses that indicate which device originated the frame and which device is expected to receive and process it. In contrast to the hierarchical and routable addresses of the network layer, layer two addresses are flat, meaning that no part of the address can be used to identify the logical or physical group to which the address belongs. The datalink bus provides data transfer across the physical link. That transfer can be reliable or unreliable; many datalink protocols do not have the acknowledgments of successful frame reception and acceptance and some data link protocols might not even have any form of check sum to check for transmission errors. In those cases, higher-level protocols must provide flow control, error checking, and acknowledgment and retransmission.

What does all this means you? Well the data link layer is often implemented in software such as network card drivers. The operating system will have defined software interface between the data link in the network transport stack above this interface is not a way of itself but rather a definition on facing between layers

The data link layer also has two sub-layers The two sub-layers are commonly known as logical link control sublayer, and media access control sublayer.

If there were a hierarchy for sub-layers logical link control would have it. The logical link control sublayer multiplexes protocols running atop the data link layer, and optionally provides flow control, acknowledgment, and error notification. The logical link control also provides addressing in control of the data link that specifies which mechanisms are to be used for addressing stations were the transmission medium and for controlling the data exchange between the originator and recipient machines.

So now let's take a look at the media access control (MAC) sublayer. This as I'm sure you've guessed by now is the layer that is under logical link control sublayer. This layer determines who is allowed to access the media at any one time. Other times it refers to a frame structure with a Mac address inside.

The two main forms of media access control are distributed and centralized. The think of it is two people talking on the phone in a network made of people speaking, imagine watching a group of people, you being one of them. Now imagine trying to determine who is going to say what the next and let's say you see two people about to begin speaking at those two people speak at the same time they will back off and begin a long and elaborate game of saying no you first.

This sublayer also determines where one frame of data and seven excellent starts this is simply referred to as frame synchronization. There are four different ways that the media access control utilizes frame synchronization they are as follows; time-based, character counting, byte stuffing and bit stuffing.

Let's take a deeper look at frame synchronization that way you can truly understand what it is that the media access control sublayer actually does for you.

1. The first one that we mentioned was the time-based approach which simply puts a specific amount of time between the frames. This is a pain in the ASP oh due to the fact that new gaps can be introduced CAN be lost due to external influences.
2. Character counting simply notes the count of the remaining characters in the frames header. This method however, is easily distributed if this field gets faulty in some way, making it harder to keep up synchronization.
3. Byte stuffing this comes before the frame with a special byte sequence such as DLE STX and it succeeds it with DLE ETX. Appearances of dle (byte value 0x10) has to be escaped with another dle. The start and stop marks are detected at the receiver and removed as well as the inserted dle characters.
4. Bits stuffing simply replaces the starting blocks with flags consisting of a special bit pattern. The chances of this bit pattern in the data transmitted is avoided by inserting a bit. Hee hee example where the flag is 01111110, 0 is inserted after 5 consecutive 1`s in the data stream. The flags in the inserted 0`s are removed the receiving end. This makes for arbitrary long frames and easy synchronization for the recipient. Note that this stuffed bit is added even if the following data bit is 0, which could not be mistaken for a sync sequence, so that the receiver can run unambiguously distinguished updates from the bits.

As we did with the first player we are going to list a bunch of the data link layer services, they are as follows;

1. Encapsulation of Network layer data packets into frames.
2. Frame Synchronization.
3. Logical Link Control Sublayer
4. Error control
5. Media Access Control layer
6. Multiple access protocols
7. Physical addressing or Mac addressing
8. LAN and switching = just packet switching
9. Data packet queuing scheduling
10. Store and forward switching or cut through switching.
11. Quality of service control
12. Virtual LAN's (VLAN)

The protocol examples are as follows:

1. ARCnet (Attached Resource Computer NETwork)
2. ATM

3. Cisco Discovery Protocol (CDP)
4. Controller Area Network
5. Econet
6. Ethernet
7. Ethernet automatic protection switching
8. Fiber Distributed Data Interface
9. Frame relay
10. High-level data Link control
11. IEEE 802.2
12. IEEE 802.11 wireless LAN
13. Link Access Procedures, D channel
14. LocalTalk
15. Multiprotocol label switching
16. point-to-point protocol
17. Spanning Tree Protocol
18. StarLan
19. Token ring
20. Uni-directional link detection
21. As well as most forms of serial communication

<u>The Network Layer (Layer 3)</u>

"What do you mean, Customs stopped the package?"

A lot of the design and configuration work for inter-networks happens at layer 3. Why? Well the network layer (layer 3) defines network addresses. I am not talking about MAC address, we already covered them in layer 2 (if you where paying attention). It covers things like Internet Protocol (IP).

It also defines network addresses in a way that route selection can be determined systematically by comparing the source network address with the destination network address, and thus applying the subnet mask. This layer defines the logical network layout, for instance, routers can use this layer to determine how to forward packets. And on top of that it (layer 3) also controls switching, creating logical paths (Virtual Circuits) for transmitting data from node-to-node.

We already talked about the routing and forwarding functions of this layer however there is more! The network layer also handles error handling, congestion as well as packet sequencing (which you can do some pretty evil things with).

Relation to Other layers

As with all the above mentioned layers the network layer works with the data link layer by translating logical communication requests from the data link layer (layer 2) , into hardware specific operations to effect transmission or reception of electronic signals.

How the Network Layer Relates to TCP/IP Model

The TCP/IP Model has a layer called the Internet Layer. This is located above the Link Layer. Most of the time people will consider the Internet Layer (TCP/IP Model) as an equal or equivalent of the network layer (OSI Model). The Internet Layer is only a subset of functionality of the network layer.

The Transport Layer (Layer 4)

"All he knows how to do is DoS".

The transport layer works with the session layer and segments the data for transport across the network. One general aspect of the transport layers job is to ensure that data is delivered error-free, in the proper sequence, error recovery as well as flow control.

Its responsible for encapsulating application data blocks into data units called datagrams, or segments. Thus making them suitable for transfer to the network infrastructure from transmission the the destination host, or managing the reverse transaction by abstracting network datagrams and delivering there payload to an application. The protocols of the transport layer establish a direct, visual host-to-host communications transport medium for applications and hence the reason why they are also referred to as "transport protocols".

I am willing to bet as you read through this you may be a bit baffled, hopefully not. However you already know one of the protocols that we have been talking about here (that is if you know anything about protocols what so ever). The protocol that I am referring to is TCP.

Lets take a (very) quick look at a few protocols and how exactly they interact and what they are.

TCP: Transmission Protocol data. Responsible for segment size, flow control, the rate at which data is exchanged, and network traffic congestion. Besides the internet some of the other uses of TCP e-mail and file transfer (among many other things).

UDP: User Datagram Protocol. This protocol is also sometimes referred to as : Universal Datagram Protocol.

SCTP: This is a relatively new protocol. It stands for Stream Control Transmission Protocol. This protocol will most likely be outdated as of Ipv6, so there is no point in getting into it. Feel free however to look it up and research it. Definitely interesting, just not completely relevant.

SSL: Secure Socket Layer. This was created by Netscape. The original purpose of it was to send private data. It uses cryptographic system that uses two keys to encrypt said data. The first key was a public key that was known to everyone. The second key was a private key or “secret key” that was only known to the person whom was receiving the message.

DCCP: Datagram congestion Control Protocol. DCCP`s job is to Implement reliable connection setup, tear down, ECN, Congestion control, and feature negotiation. Through DCCP n youth team gained access to congestion control mechanisms without having to implement them at the application layer.

ECN: Explicit Congestion Notification. ECN is an extension to Internet Protocol. ECN`s job is to manage end-to-end notification of network congestion, without dropping packets.

The preceding were not all of the protocols that are used, only a few.

The Session Layer (Layer 5)

“They keep trying to add me, 1-word, Denied”.

The Session Layers (layer 5) job is to establish, manage and terminate communication sessions. So what is a communication session? Communication sessions consist of service requests and service responses that occur between applications located in different network devices. Said requests and responses are coordinated by protocols (we already went over a few protocols) implemented at the session layer.

So what are some of the protocols being implemented? Well Remote Procedure Calls (RPC), Zone Information Protocol (ZIP).

Let’s take a closer look at those two protocols.

RPC: Remote Procedure Calls, is an inter-process communication that allows a computer program to cause subroutine or procedure to execute in another address space.

ZIP: Zone Information Protocol This was the protocol that AppleTalk network numbers where associated with zone names. The “Zone” was a subdivision of the network that made sense to humans.

The Presentation Layer (Layer 6)

“I must have missed something or I am just retarded to not understand what you are trying to say.”

Have you ever been busy and had a friend type for you whether it be for an IM or something else? Did that friend ever mess up that message? Then your friend is equivalent to the Presentation Layer on a bad day. The Presentation layers job is to preserve the meaning of information sent across a network.

So how does it work its magic? It “represents” it. And no not like a hacker represents the underground, more like encodes it. It does this in more than one way however like, data compression or encryption.

Let’s take a deeper look inside the presentation layer and what it does; it mainly has the following responsibilities:

1. Data Format. Converting the complex data structures used by an application strings, integers, structures, etc. into a byte stream transmitted across the network. Representing information in such a way that communicating peer’s age to the format of data being exchanged.
2. Compressing data to reduce the amount of transmitted data.

The Application Layer (Layer 7)

“God!”

This layer is used by network applications. These programs are what actually implement the functions performed by users to accomplish various tasks over the network. The application layer is the layer with the most functions, yay. It provides services for user applications to employ.

Let’s say for instance you open up Google Chrome, now that it’s up and open doesn't mean that it resides in the application layer. It is how ever an application running on your Box. It does however use some of the protocols of some of the services that are located in the application layer. HTTP (Hyper Text Transfer Protocol) or HTTPS (Hyper Text Transfer Protocol Secure) are probably the ones that you're most familiar with.

There are a ton of protocols that reside in the application layer. You will find a lot of them in the “Protocol Reference” located below.

Protocol Reference

(written by killab)

BGP: Border Gateway Protocol, is the core routing protocol of the entire internet! BGP maintains a table of IP networks or “prefixes” which designate network reach-ability among Autonomous Systems (AS).

DHCP: Dynamic Host Configuration Protocol is a communications protocol allows network administrators centrally manage and automate the assignment of Internet Protocol (IP) addresses in an organizations network. DHCP is an extension of an earlier network IP management protocol, Bootstrap Protocol (BOOTP).

DNS: Domain Name System (or Service), that translates domain names into IP addresses. Domain names are alphabetic. Every time you use a domain name, a DNS service must translate the name into the corresponding IP address. Say the domain name http://www.packetstormsecurity.org might translate to 66.227.17.19.

FTP: File transfer Protocol, enables you to transfer files from one computer to another computer, network or the Internet. Which in-turn explains the origin of its name; it was formed as an acronym.

HTTP: Hypertext Transfer Protocol is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing of data representation, allowing systems to be built independently of the data being transferred.

IMAP: The Internet Message Access Protocol (IMAP) is one of the two most prevalent Internet Standard protocols for e-mail retrieval, the other being the Post Office Protocol (POP).

IRC: Internet Chat Relay, enables people all over the world to talk together over the internet in real-time sessions in virtual rooms.

Megaco: Media Gateway Control Protocol, is a VoIP protocol.

MGCP: Media Gateway Control Protocol, is a protocol for the control of Voice over IP (VoIP) calls by external call-control elements known as media gateway controllers (MGCs), or call agents (CAs).

NNTP: Network News Transfer Protocol, the protocol used to post, distribute, and retrieve “usernet” messages.

NTP: Network Time Protocol, is designed to synchronize the clocks of computers over a network.

POP: Post Office Protocol enables any email program anywhere on the Internet to connect to any email server to perform the usual email functions, like reading and sending, as long as they have a valid account and password.

RIP: Routing Information Protocol is dynamic, distance vector routing protocol.

RTP: Real-time Transport Protocol opens two ports for communication. One for the media stream (an even port number) and one for control (QoS feedback and media control) - RTCP. The port numbers are not hard defined, it depends very much upon the application.

RTSP: Real-Time Streaming Protocol establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video.

SDP: Session Description Protocol is a format for describing streaming media initialization parameters.

SIP: Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF) standard protocol for initiating an interactive user session that involves multimedia elements such as gaming, chat, voice, video, and virtual reality.

SMTP: Simple Mail Transfer Protocol is a TCP/IP protocol used in sending and receiving e-mail.

SNMP: Simple Network Management Protocol, is essentially a request-reply protocol running over UDP (ports 161 and 162), though TCP operation is possible

SOAP: Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of web services in computer networks.

SSH: Secure Shell or SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices.

Telnet: A terminal emulation program for TCP/IP networks such as the internet.

TLS: Transport Layer Security, is a protocol that ensures privacy between communicating applications and their users on the Internet.

XMPP: Extensible Messaging and Presence Protocol, is an open technology for real-time communication, which powers a wide range of applications including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middle ware, content syndication, and generalized routing of XML data.

ICMP

(Written by: Aviator753)

ICMP - Internet Control Message Protocol...ICMP is not used for data transmission as TCP and UDP commonly are, but rather for network error messaging; for example, a host could not be reached or a service is unavailable. ICMP messages are generally created and sent in response to errors in IP 'datagrams' (basically a packet that does not notify the sender upon a delivery failure), diagnostics, or routing purposes. Every ICMP message is included inside a single IP datagram..therefore it is unreliable (Does not deliver a reply message.. delivery is unknown to host) The ICMP header begins after the 160th bit of the IP header. Though it is inside an IP datagram, it is not treated the same. The contents of the ICMP message often require the error message to be sent back to the application that originally created the IP packet that caused the need for a ICMP message. Still with me? Good. Most common network utilities are based upon ICMP; for instance Traceroute and Ping.

Examples of ICMP Message Types
0 - Echo Reply.. used as a response to a Ping
1 - Reserved
2 - Reserved
3 - Destination Unreachable.. Transmission failure due to destination host, protocol, or port unreachable, destination host or network unknown or prohibited, route failure, or isolated source host.
4 - Source Quench.. used for congestion control
5 - Redirect Message.. includes a redirect datagram for the host, network, TOS (Type of Service) & host, or TOS & network
6 - Alternate Host Address
7 - Reserved
8 - Echo Request.. used to request
9 - Router Advertisement.. every router periodically sends this message from each interface announcing the IP Address of that interface
10 - Router Solicitation.. Router discovery/selection/solicitation (Are you there, Mr. router?)
11 - Time Exceeded.. This message is sent when a packet takes too long to reach its destination [also called TTL (Time to Live)], or when fragment reassembly takes too long
12 - Parameter Problem: Bad IP header.. this message is sent when a IP header is missing a required option, pointer indicates an error, or its length is incorrect
13 - Timestamp.. requests time for synchronization

14 - Timestamp Reply
15 - Information Request
16 - Information Reply
17 - Address Mask Request.. requests subnet mask
18 - Address Mask Reply
19 - Reserved for Security

For more information on ICMP types and codes, please reference RFC #792:
http://tools.ietf.org/html/rfc792

TCP/IP

(written by killab)

OK, so we just finished learning about the OSI Model. If you skipped that chapter and figured you already knew it I suggest you go back and read it. If you're a neophyte as that is who this book is designed for and you skipped the OSI Model section then you seriously need to go back and read it.

The Origin

To fully understand what something is you should know where it came from, this section will tell you about the TCP/IP`s origins.

So when was this Internet Protocol Suite (TCP/IP Model) made? It was created in the 1970`s by the Defense Advanced Research Projects Agency. Who are they? They are an agency of the U.S. DoD (United States Department of Defense).

Pretty original thinkers huh? Not really, the model for the TCP/IP model was a morphed hybrid of ARPANET (Advanced Research Projects Agency Network, say that 10x fast). Basically it was the world's first WAN (Wide Area Network). You're talking pre-internet here.

TCP/IP Vs. OSI

So why should you care about the TCP/IP Model? Well, the reason you should care about it is because you use it on a daily basis.
So why did I just have you read the OSI Model section? We utilize the OSI Model to educate people (not just neophytes also Network Admins, and pretty much anyone who needs to know the architectural structure of a network and computer system both as a whole and as an individual). The OSI model does a better job at laying out what each layer is doing.

As we discussed in the previous chapter the OSI Model has 7 Layers. TCP/IP only has 4. They layers are as follows: Application, Transport, Internet, and Network Access. The Why we will get into in a bit.

The OSI model was not always a tool we used for educational purposes though. In fact in the early 1970`s it was a competing standard to TCP/IP. That's right they weren't always friends! At that point in time, they were in fact competitors. For those of you not paying any attention, TCP/IP 'won' that fight. Although many speculate as to why, and think that OSI is/was a much better protocol. Why if it was so better did it loose then? Its addresses where far too complex, how so? It used hexadecimal, if you don't know what that is then 2 seconds of goggling should provide you with an answer. I find this funny as all hell. Why? Go look at an IPv6 address.

The OSI model is still used as a way to teach though. But TCP/IP is what is typically used for network communication.
TCP/IP is not just one protocol but an entire suite of protocols. If you think of GM, they have a ton of other sub-companies. Different plants that provide different products for different things. In the same way TCP/IP is a suite of protocols.

The Layers

Unlike the OSI model which has 7 layers, the TCP/IP model only has 4. So where did the 3 layers go? Nowhere, they were in fact merged into one. The reason that all of the top 3 layers are merged is because all of the top 3 layers have stuff that happens before it leaves the computer. Stuff that is only seen by the operating system.

The transport layer stays the same. The network layer was turned into the Internet layer. The data link layer and the physical layer where merged into one and in the TCP/IP model it is called Network Access Layer. In order to get a better understanding of the difference, I will list the OSI Model Layers and the TCP/IP Model Layers.

The OSI Model Layers:
1. Application Layer
2. Presentation Layer
3. Session Layer
4. Transport Layer
5. Network Layer
6. Data-Link Layer
7. Physical Layer

Now let’s compare that to the TCP/IP Model. As you will see they are different. Remember, as I discussed before, some layers are merged together to create one layer, as opposed to 3 separate layers. If you can only remember one of the models, I suggest you remember OSI.

The following is the TCP/IP Layers:

1. Application Layer (merged: Application Layer, Presentation Layer & Session Layer)
2. (Host-To-Host) Transport Layer
3. Internet Layer (merged: network layer & Data-link Layer)
4. Network Interface Layer

That's it. If you paid attention in the OSI Model chapter then you already know the jobs, and ability's of the layers individually and there is no need to restate all of them. If not then perhaps you should consider reading this book in order. Who knows maybe the authors but it in that order for a specific reason...

Shoutz

Ratdance:

#Suidrewt

Killab:

DC802
IPT
Agent X

Aviator753:

Deaftone
Aig|GreenMonster

MLS577:

#Suidrewt

Contact Info:

Ratdance: ratdance@gmail.com

Killab: threefiftysevencopkilla@gmail.com
Skype: killab6661

Aviator753: aviator753@gmail.com
Skype: Avi753

MLS577: mls577@live.com