Beginner's Guide to Computer Security

This document is an overview of computer security appropriate for someone with no previous knowledge of the subject. An understanding of computer security will help you use your computer in a safe and efficient manner and help protect you from viruses, spam and computer crime. Acquiring an understanding of computer security also makes you a good citizen by stopping your computer spreading these problems to other people and by increasing the cost of crime. Although this document looks mainly at how you can protect yourself, by doing so you will be helping other people too. This document concentrates on the needs of home users and small businesses. Large organizations may have different priorities.

Your Valuables

To choose appropriate methods of protection you need to consider what valuable things you have on your computer or which may be affected by your computer and how these things might be lost or stolen. Here is a list in order of importance for a typical individual. Your circumstances and the information on your computer will influence the order of this list.

Information Loss

You can lose information by accidentally deleting or modifying files. Information loss can also be caused by hard disk failure, computer theft, fire, or other people using your computer.

Computer Theft

The theft of portable computers from public places is a frequent problem, but the risk of burglary of all computer equipment should be considered.

Loss of Your Time

Avoid the mistake of ignoring the value of your time. Even if you are well-organised it could take days of work to recover information or computer functionality after an incident. Your time can also be consumed in more subtle ways, such as by having to deal with adverts delivered using spam or advertising programs.

Identity Theft

This can be as simple as theft from your credit card or bank account by someone making unauthorised internet purchases. More seriously it could extend to obtaining loans in your name or the use of your bank account for money laundering.

Telephone Bill Theft

Money can be stolen from your telephone account by making your computer call 'premium rate' phone numbers owned by the thieves.

Bandwidth Theft
This is use of your computer and its internet connection without your permission. It can include sending spam, viruses or other unpleasant files at your expense and in your name.

Information Release
Saleable information such as lists of passwords, credit card numbers or customer names and addresses are obvious targets. Personal information and confidential business information can also be worth protecting.

You need a variety of complementary measures to protect these valuables properly. Below is an overview of the techniques and knowledge required, which applies to all types of computer.

Your Information is Valuable - Copy It
Consider what you would lose if your computer died at the end of this sentence. Weeks of hard work? Copy all your information on to a removable disk, such as a CD or DVD, now. Work at making this process easy, because it will affect how often you take a copy of your information and consequently how much you lose when there is a problem.

If you are not familiar with 'files', 'directories' and 'disks' get someone to show you how they work. Almost all your valuable information will be held in 'files'. A 'file' is a long list of related information usually magnetically imprinted on a rotating disk. Files are given names and are grouped together in 'directories' to make them easy to find and copy. You will need to understand your files and how they are arranged in order to make copies of your information. Learn also how to check how much free space you have on your disks. Running out of free space can corrupt the files you are using and destroy your work.

It is best to make three levels of copy :

- A daily copy of the files you are working on into another directory on your computer, preferably to a second hard disk. This will protect against accidental modification or deletion and against partial hard disk failure. To do this you could make a complete copy of a directory you are working in into a directory named 'Save'. Edit the new subdirectory name to add the date to the end so you know when it was saved and can keep several copies of the same directory made at different dates.

- Every few days or after reaching an important point in your work copy these saved directories on to CD or DVD kept in the same building as the computer. This will protect against hard disk failure, computer theft and viruses. A firesafe in a different room is the ideal place to store the disks. As with your copies on the hard disk, write a date on the disk and keep several old disks, not just one.

- Occasionally take a complete copy of all the files on your computer and put it in another building in case of fire or theft. If you do not have another building available put a copy in the garden shed, your car, or any place that is unlikely to be destroyed in the same fire. This copy could be taken using special 'backup' software, but be careful to keep your information in a format which can easily be read on a different computer. Try retrieving a file from the copy on a different computer, to make sure it works.

Passwords

Choosing and using passwords is crucial to many aspects of computer security. Passwords are the keys to your digital homes. When you are asked to choose a new password the objective is to create something you can easily remember, but nobody else could guess even if they knew all the other passwords you have ever used and set their computer to work trying millions of different guesses. This sounds difficult, but with a little thought you can invent memorable and impressively secure passwords.

Ideally your password should be more expensive to guess than the value of the thing it protects. If you are protecting information of declining value, a good password is one which takes so long to guess that the information becomes worthless. How difficult does a password need to be? Suppose your computer could test 1,000 guesses at a password every second and you want to keep some information safe for two years against an adversary that has 10,000 similar computers at their disposal. In two years they can test 600 million million guesses. If your password is composed entirely from randomly selected lower case letters each character could be discovered by trying all 26 possible letters. Every extra character added to the length of your password multiplies the guesses they need by 26. So to force 600 million million guesses your password should be 11 characters long. For most purposes a password of 11 random lower case letters is unbreakable.

To make the password shorter but just as effective you can use the 26 uppercase letters, 10 digits and 30 punctuation symbols as well as the 26 lower case letters. This would force them to try 92 different characters in each position, thus reducing the required password length to just 8 characters. But allowing for some predictability in the characters you choose, a password that is 10 characters long is unbreakable. Typing 10 characters is quick and easy, but the real problem is how do you remember these 10-character passwords.

If you apply some ingenuity and memory games you can invent various ways to create and remember fairly random 10 character passwords. A good method is to pick a recent very memorable event in your life. Something you won't forget. Next make up a sentence about this event. Take the first letter of each word in the sentence and you get a password. So for example 'Sally and I went to the pictures on Wednesday' gives a password of SaIwttpoW. A fairly good password which, if the event was important to you, would be easy to remember. A test of

how good the sentence is for constructing your password is to consider whether a search of the web for this exact phrase would find any pages. Clearly the more private or bizarre the thoughts you include the less likely anyone is to guess any of the words you might put in the sentence. The structure of sentences and the initial letters of words are to some extent predictable, but passwords about 10 characters long should be unbreakable. This method will work well for the two or three important passwords you use regularly.

To remember lots of passwords, or infrequently used passwords, you need to add a different technique. Because you should never use the same password in different places, nor re-use an old password, one way to avoid having to write down your passwords is to split them into a common stem, say the first few characters, which you change on all your passwords at the same time and invent using the above method and a shorter distinguishing portion, say the last few characters, which is unique to each password but doesn't normally change. So the distinguishing parts might be 'My very expensive bank' Mveb, 'The accounts system' Tas, giving passwords of SaIwttpoWMveb and SaIwttpoWTas etc. You will need to keep a list of all the places you use passwords and change them all simultaneously so you only have to remember one stem at a time. This method does, however, increase the risk that someone who breaks into one system will be able to break into the others by knowing the password stem. So a longer distinguishing portion is needed for important passwords.

In practice it is better to write any infrequently used or unimportant passwords on a piece of paper. Writing down the passwords has the advantage that you can use even longer passwords with lots of punctuation and digits. If you cannot lock the paper somewhere secure you could tweak the way you write the passwords to make them difficult to read, for example by adding a character to the front of each password.

Every time you type a password there is a chance it will be seen by someone, either physically or electronically. Try to avoid typing passwords on insecure or public computers. If you think a password has been seen by somebody else, change it, and if possible check whether it has been misused, for example by studying the transactions on your account.

Programs and Data

The security of your computer depends on your ability to distinguish 'programs' from 'data'. A 'program' is a list of instructions written by someone for the computer to follow slavishly, 'data' is any other information such as the text of this document. The word 'program' is usually spelt 'program' rather than 'programme'. Programs can be dangerous because the instructions might tell your computer to do something you do not want it to do. The word 'data' can mean one bit of data or a whole load of data. Data is relatively harmless. Data just sits there, like a book on a shelf, until an instruction in a program tells the computer to read the data and use it somehow.

Absolutely everything your computer does is achieved by it 'running' one of the hundreds of programs you have on it. Programs are harmless until they are 'run'. You may also meet the term 'execute', this means exactly the same as 'run'. When you 'run' a program you are telling your computer to follow the list of instructions and do whatever they tell it to do. So by 'running' a program you give total power over your computer to the person who wrote the list of instructions. This power is obviously not something you want to give to someone who will misuse your computer. The term 'attacker' is applied to someone who is trying to do something on your computer you have not given them permission to do. You are the 'defender' who tries to stop the attackers. The usual objective of an 'attacker' is to run their program on your computer, because this gives them complete control over your computer. So the attackers win if they run their program and you win if you stop them.

Actually not all programs are created equal. Some run with more power than others. For a simplified picture of how this works meet the 'kernel' and the 'sandbox'. A 'sandbox' is a confined area containing a safe set of tools the program can play with. Instructions in the program about plastic buckets and spades are obeyed, but those requesting chainsaws are ignored. The program runs inside the sandbox and can safely do whatever it likes inside, but it is not allowed to escape. The 'kernel' works similarly but the other way round. Not all computers have a 'kernel' but for those that do it is the most powerful program the computer runs. It doesn't do anything very exciting from the user's perspective - it is usually in charge of the memory, hard disks and clock - but all the other programs are subordinate to and are controlled and terminated by the kernel. The kernel effectively runs all your 'user' programs like word processors and web browsers in sandboxes, to prevent them misbehaving or fighting with each other. Although kernels and sandboxes are very reliable ways to constrain the power of programs an attacker can gain huge power if they can find a flaw in the mechanism and write a program that breaks out of a sandbox into a user program, or out of a user program into the kernel. Attackers spend much effort trying to discover these flaws, while defenders do their best to eliminate them. The ultimate victory for an attacker is to get their program to run as part of the kernel with unfettered power over the computer.

There are various other mechanisms for running programs with fewer privileges, but they generally work in a similar way with a more privileged program starting the execution of a less privileged program. It may also be possible to restrict access to files on your disks to particular users.

A Wolf in Sheep's Clothing

Because of their size programs always contain mistakes. These mistakes are called 'bugs'. Bugs are dangerous because they may allow an attacker to give the program some data which is unexpected or will confuse its instructions. This could stop the program working, allow access to your files, run another program,

or in the worst case make the program treat the data as if it were part of its own program and run it. Apparently harmless data can become a dangerous program that the attacker has managed to run on your computer because of a bug in your program.

To defend yourself you need to know a little more about bugs. Programs are written as short lines of text, called 'source code' or just 'code', and are often measured roughly in terms of the number of these lines. A very small program might only have 10 lines, but an ordinary program could have 10,000 lines and a very large set of programs 2,000,000. Every 1,000 or so lines a mistake will be made when the program is being written, so the larger the program the more bugs it is likely to contain, and you could guess there might be 2,000 bugs in our very large set of programs. From this you can see that the larger the programs you allow an attacker to feed data into, the more bugs are available for them to use against you.

A bug that can be used in some way by an attacker is called a 'vulnerability'. When a new vulnerability is discovered a race starts between the attackers and the defenders. The attackers want to use the vulnerability and the defenders want to prevent it from being used. To use the vulnerability the attackers need an 'exploit'. An exploit might be a small program embedded in some data that will be run when that data is fed into the vulnerable program. The defenders must either find some way of stopping the bug being accessible to the attackers or get the bug corrected and replace the faulty program with a fixed version. When the bug is fixed you may be able to get the fix in the form of a 'patch' which can be 'applied' to the original program to solve the problem rather than having to download a whole new copy of the program. The latest version of a program will have all the patches already applied. By using recent versions of any programs that are accessible to attackers you reduce the number of exploits that can be used against you.

This leads us to consideration of which programs are accessible to an attacker. Which programs can an attacker most easily feed data into? By far the most accessible and therefore dangerous place to put a program is in a place where everyone on the internet can feed data into it whenever they want. Much less accessible, but still exposed to places you obtain information from, are the programs you use to read internet information such as your web browser. And fairly safely positioned are all the other programs which never talk to the internet, but may occasionally be fed some data you have been sent or downloaded.

Thus large accessible programs are dangerous even if they are updated every week, but small or inaccessible programs can be used quite safely for years. For safety purposes your objective is to minimise the accessibility, size and age of the programs running on your computer.

Mis-configuration

The behaviour of many programs can be radically altered by changing their 'configuration'. It is usually very easy to change the configuration of a program through its menu options or by editing small text files. It is also very easy to get the configuration wrong or leave it in a dangerous default state. For an inaccessible program this may just result in some slight inconvenience when you are using it, but for a program that can be reached by people on the internet it is not impossible to accidentally give the world access to all your files. Any kind of remote access or network program needs to be studied and configured very carefully.

The configuration of a program often gives you the ability to switch off large chunks of that program. Switching off the features you do not need can significantly reduce the number of vulnerabilities that are accessible to an attacker.

Stop the Attackers Running a Program

The attackers win if they manage to run their program on your computer and you win by stopping them. To win you need to be able to recognise programs in their many guises, understand what will start a program running, and make informed decisions about which programs you do run.

To recognise programs, start by studying the different types of files on your computer. Understanding the different file types will help you decide whether a file you receive, by whatever means, is a potentially dangerous program file or a relatively harmless data file. Programs are referred to by many names depending on the kind of program : executable, script, macro, batch, command, application, library etc. You do not need to know the distinctions between the types of program - just learn how to recognise the common types of files so you will be able to distinguish programs from data. If you receive a file of an unknown type it should be classified as a program until you have studied it thoroughly and found it is just a data file. The game for an attacker is to entice or fool us into running their program. The better you are at recognising programs the safer you become.

As well as recognising a program when you receive one, you also need to know what things may start the program running. Any mechanism you have on your computer for receiving files such as email or CD is also likely to have a quick and easy way of running any program files you receive. This is convenient and useful, but until you understand what triggers the mechanism and how you can prevent it running a program it is also dangerous. Study the configuration of any programs you have that receive files, and make sure they will warn you before running any programs. Less convenient but safer, switch off their ability to run any programs.

Having received or found a program and recognised it as such, you must decide whether to run it. The question to ask yourself is 'Do I want to invite the people who wrote this program to do whatever they want on my computer?'. If the possible gain from fun, education or productivity outweighs the potential

detrimental effects, you run the program. But before you do run it consider how you will clear up the mess if anything goes wrong. Now is a good time to make copies of files or a backup of the whole computer. The ability to get your computer back quickly to the state it was in before you ran the program allows you to take more risks when running new programs.

Another consideration is that generally the more programs you run, even if they are well-intentioned programs, the less stable your computer is likely to become and the more likely you are to have to have to restore it to a previous state. For both safety and stability it is best to minimise the number of programs you run on your computer.

Firewalls

Your security is improved when you remove opportunities for attackers to feed data into the programs on your computer. Your network connection is usually the easiest way for attackers to interact with the programs on your computer, so is an important area to study and secure. Your primary means of securing a network connection is a 'firewall'. The main purpose of a firewall is to reduce the accessibility of your programs to attackers over a network. It can also save you from the consequences of accidental mis-configuration and may alert you when a malicious program is run on your computer. A firewall lets you decide whether a program on your computer is allowed to use the network connection, restrict the computers it can talk to, and prevent computers on the network initiating these conversations.

A 'firewall' is a program, but the same term is also used to refer to a 'hardware firewall' which is a computer allocated exclusively to the task of running a firewall program. A firewall program does not need much computing power, and running a firewall on your computer is useful whether or not you have a hardware firewall between your computer and the network. Businesses often have more than one firewall because each extra layer adds to the protection. Firewalls give cheap and effective security.

The general approach to firewall configuration is to make it stop all access to the network, then selectively open that access just enough for each of your programs to communicate as required. Try running a program which needs network access, and if it doesn't work then determine what it is trying to do that the firewall is stopping and open this aspect of the firewall. Repeat this process until all your programs work properly. You can go to great technical lengths configuring a firewall, but most of the benefits can be gained with only a simple knowledge of networks. A little configuration work gives you considerable protection.

Although firewalls can be used on all kinds of networks, the most likely place for your firewall is between your computer and the internet. The rest of this section will concentrate on this application.

Your wired or wireless connection to the network will be totally controlled immediately inside your computer by a program called the 'network stack'. This is a small program that sends and receives all the information going through your network connection. If you run a firewall on your computer it gives you control of the network stack, thus the ability to stop information going through your network connection. Some understanding of what the network stack does will help you configure your firewall.

To send information to a computer on the internet the network stack needs to be given the address of the computer to send it to. This address is an 'Internet Protocol' address or 'IP address', and is the internet equivalent of a telephone number. You will see them expressed in 'dotted quad' format such as 81.2.66.10, that is four numbers separated by full stops. Every computer on the internet has its own unique IP address.

Your internet conversations are conducted using three main 'protocols'. A 'protocol' is a set of rules that governs how computers conduct their conversation. These three protocols are 'TCP' (Transmission Control Protocol), 'UDP' (User Datagram Protocol) and 'ICMP' (Internet Control Message Protocol). 'TCP' is a two-way conversation, 'UDP' is a one-way sending of information, and 'ICMP' is normally used to report problems with UDP or TCP conversations. All internet conversations are conducted in small chunks of information called 'packets'. When sending a large file the network stack chops it up and puts it into small packets, which are then transmitted. When they arrive the packets are reassembled by the network stack on the receiving computer. The protocols in use determine how big the packets can be and what information is put in them. Just like posting letters, your packets may get lost or delayed somewhere on the internet, but your network stack solves these problems for you by resending packets and sorting them into the right order. Other types of network use other protocols, and usually these should not be allowed out on to the internet. Stop all protocols except TCP, UDP and ICMP with your firewall unless you are certain you need them.

Almost all your information will be transmitted using TCP. Because TCP is the most important protocol and uses the 'Internet Protocol' to handle IP addresses, you will often see references to this combination of two protocols expressed as 'TCP/IP' which means using TCP and IP together. Because the main purpose of the 'network stack' is to communicate using TCP/IP, it is often called the 'TCP/IP stack'. It is called a 'stack' because the protocols often sit on top of each other in combinations like TCP/IP, where TCP is in charge and gives IP tasks to do.

You may be running several programs on your computer that all want to hold conversations through the network stack at the same time. For the network stack to pass the information it receives from the network to the right program, TCP and UDP use a number called a 'port'. You can think of a 'port' as a room in a

building. The IP address gets your letter to the right building (computer) and the port number gets the letter to the correct room, which will be occupied by someone who receives it (a program). Some ports are by convention always allocated to the same program, such as a website using port 80, but others may be given any free port number. When configuring your firewall you should ideally block all the ports, then selectively open them as you find programs on your computer that need to use them.

There are essentially two ways a program can use a port - one is dangerous and the other is safe. Some programs will use a port to 'actively' initiate a conversation with a particular computer. This is equivalent to your program phoning the other computer. This is safe because only the computer your program has decided to phone will be able to talk to your program. If you browse a website your web browser will only use ports in this safe fashion because your browser knows exactly which computer it wants to talk to every time it wants information. Other programs will open a port and 'passively' wait for any computer on the internet to start a conversation. This is like answering the phone when it rings - it could be anyone in the world who wants to speak to you. This is dangerous because it makes the program which opens the port accessible to an attacker from any computer on the internet. Instead of only having access to the program if they control the one computer your program has chosen to speak to, the attacker can attack you from any of the millions of computers on the internet. A program that is this accessible has to be very secure to survive the constant attacks to which it will be subject. An example of this type of program is the 'web server' which your web browser contacts to get a web page. A web server will have a port open permanently for any computer which wants to contact it. But most programs should not use ports in this 'passive' fashion. If you look at the incoming attempts to talk to your computer from the internet, you will see every minute or so an attacker trying to initiate conversations through passively open ports where they expect to find a vulnerable program. There are several ways to defend against these attacks. Obviously you can thwart them by not running programs that open passive ports to allow incoming connections. Since many computers run mis-configured programs that allow incoming connections, you can also reduce the risk by studying and correcting the configuration of any programs that may use the network. And your outer level of defense is to run a firewall, which can be set to refuse incoming connections. Running a firewall that refuses all incoming connection attempts will massively increase your security on the internet.

Viruses and other nasties

Prevention is much better than cure. Your objective should be to prevent the attackers running their program on your computer. But it is worth considering what may happen and what you can do if you lose this battle. The malicious programs you are most likely to notice are the ones that 'self-propagate' - they contain a mechanism that allows them to copy themselves on to other computers. A self-propagating program is called a 'virus' if it requires your

assistance to propagate, a 'worm' if it propagates unassisted using a network, or possibly 'spyware' if its purpose is to gather information. Note that these are all just programs, and should be kept off your computer in the normal ways. However, to help detect and remove these programs you can run an 'anti-virus' program sometimes know as a 'virus checker', or a similar program that fights spyware. These can run continuously on your computer looking for viruses, worms and spyware. Their primary method of detecting a virus is to look for a small piece of the virus, called a 'signature', that the anti-virus company has decided by manual inspection of a captured virus is unique to that virus. Occasionally a mistake is made by an anti-virus company and the same signature appears in a legitimate program and the anti-virus program will falsely accuse it of containing a virus, but this is very rare. A more likely problem is that the virus will be missed by the anti-virus program because the virus has been written after the signatures were updated. Virus writers release frequent changes to their viruses to stop the old signatures detecting them. For this reason you should configure your anti-virus program to obtain daily updates of its signatures from the internet. An anti-virus program may also be able to remove a virus from your computer. This relies on the anti-virus company fully understanding the virus, which is not always an easy task. Viruses are intentionally written to be difficult to understand and remove. After an attacker has run a program on your computer you will probably have to completely format and reload the computer to be certain it is clean.

One of the difficulties of recovering from a successful attack is that all may not be what it seems. The attacker is unlikely to want you to know about the attack because continued control of your computer is valuable, and they will anyway not want to get caught. So the game is to replace your usual programs with impostors that look the same but do something subtly different. Three examples of this are the 'root kit', 'backdoor' and 'Trojan horse'.

The 'root kit' is the extreme example because it replaces core programs in your computer's 'operating system', such as Windows or Linux, with look-alikes that try to convince you by every means possible that they are the originals. Thus if you try to check for a root kit by looking at the files on your disk, the program in your operating system that usually gives you this list of files could be part of the root kit held in a file on the disk, innocently telling you that there are no root kit files on the disk. What used to be a truthful answer from the operating system is now a lie from the root kit.

A 'backdoor' is an alternative means of entry that bypasses the normal restrictions. This might be as simple as a password created by an attacker for later use, but it could be something less obvious such as an adjustment to a program that checks passwords to make it always accept a particular password.

A 'Trojan horse', named after Greek mythology, is a program used for gaining access to a computer rather than maintaining access. Like a root kit, a Trojan

horse is a program that pretends to be something else. Any program you use could be a Trojan horse without you noticing, so consider where you obtain your programs from as well as what they appear to do when you run them.

Secret Keys and Public Padlocks

When you send important and confidential information to someone there are three problems: how you stop other people reading the information, how the recipient can check it has not been altered in transit, and how they can be certain it was you that sent it. If you post a cheque to someone you solve these problems by putting it in an envelope to stop others reading it, writing the value in figures and words to stop anyone changing the value, and signing it with your unique signature to prove you wrote it. To do the same things on a computer you have to have more cunning mechanisms. If you have visited a secure web page or paid for something using a 'chip and PIN' smartcard you have probably already used all these mechanisms without realising it.

Central to all these mechanisms is something called a 'key'. A password is type of 'key' that is short and easy for you to remember and type. But there are other keys which are created and only ever used by computers, and so can be much longer and more random without becoming bothersome. If an attacker doesn't know the key you are using, the only way they can gain access to the information is to try every possible key until they guess the right one. Like passwords, the longer the key the more combinations an attacker will have to try before they guess correctly. Similarly a short key can be safe if it takes lots of effort to test each key to see whether it is the right one. Rather than insisting on a long password from the user, programs that ask for passwords should increase the amount of work the computer has to do to check whether a password is correct.

To stop other people reading the information you can 'encrypt' it. 'Encryption' uses a key to scramble the information. To unscramble the information again you 'decrypt' it. Every way of encrypting information is paired with its own unique way of decrypting it which, when given the right key, exactly reverses the process and recreates the original information. This encryption-decryption pair is called a 'cipher'. The wonderful thing about ciphers is that even a very slow computer can easily encrypt and decrypt information in a way that the fastest computers are unable to break into unless they are given the right key. Instead of posting your letter in an envelope you can post it in a bank vault by encrypting it.

There are two types of cipher - 'symmetric' and 'asymmetric'. 'Asymmetric' ciphers are slow and generally only used for small things like keys. 'Symmetric' ciphers are old-fashioned but fast and reliable, so they are used for large amounts of information such as big files. Symmetric ciphers are called symmetric because they use the same key to decrypt the information as was used to encrypt it. You might employ a symmetric cipher to password protect a file on your disk, but they are not very good for sending information over a network. To understand this, a useful analogy is to imagine encrypting the information by

locking it in a strong box using a padlock. You post the padlocked box to someone on the other side of the world but you also have to send them the key to open the padlock, and the only way you can do this is by posting the key. The 'man in the middle', the one you are trying to protect this information from, now has the key and can unlock the box.

A solution to this problem is for your correspondent to send you an open padlock of the kind that locks when you close it. He keeps the key to this padlock so he can open the box you send back to him containing your information locked with his padlock. The key is never put in the post so the 'man in the middle' is unable to open the box. Even you are unable to retrieve the information once the padlock has been closed. This is the essence of an 'asymmetric' cipher.

An 'asymmetric' cipher needs two related keys - one is an encryption key which can be shown to everyone, and the other is the corresponding decryption key which must be kept secret. The padlock can be shown to everyone but you must not give anyone its key. These are called the 'public key' and the 'private key'. The asymmetric cipher you will use has another feature which unfortunately doesn't fit this analogy, but is very useful. When information is encrypted with a 'private key' it can be decrypted with the corresponding 'public key'. This gives someone who has a 'private key' a way to prove they are the person who has the 'private key' without showing the key to anybody else. This allows everyone who generates a pair of keys, publishing one and keeping the other private, to provide a 'digital signature' the purpose of which is similar to signing a cheque.

To make it easy to prove that information has not been changed you can use something called a 'secure hash'. You could just take a complete copy of the original information and match every bit of it against the other copy, but there is a more efficient method to check that nothing has changed. Just like the total on a shopping bill, an ordinary 'hash' is a small amount of information calculated from all of a very large amount of information. If the price of a single item on a bill is altered, you know something is wrong unless the total is altered to match. A 'secure hash' works by encrypting the information, then taking a hash of the result. This makes it impossible to create some information that will produce a known secure hash and even a tiny change to the original information will produce a large and totally unpredictable change in the hash. Thus if you calculate a secure hash for some information and get the same answer as last time it was calculated, you can be certain the information has not been changed in any way. You can use secure hashes for 'digital signatures' and for checking files have not been tampered with.

There are three problems with using other people's public keys, all to do with whether you trust the public keys you have been given. A 'man in the middle' could substitute their own public key when you are expecting somebody else's, or a private key could be stolen by an attacker either with or without the knowledge of its owner, or the owner might lose their private key. Any good system of

handling public keys needs to allow the owner of a public key to cancel the public key when they think the corresponding private key has been lost or is no longer secret. It should also allow someone who receives a public key to withdraw their trust in it when they think the private key has been stolen. And the system needs to thwart a 'man in the middle' substitution of a public key. The primary means of solving these problems is to use digital signatures. You can use your private key to sign someone else's public key, possibly along with other information relating to them, and pass this signed information on to another person as a sort of introduction. Long signed sequences of these public keys and their related information can be established, where at each step they are signed by the previous person in the sequence. There are two different mechanisms you are likely to meet which do this - one is used for web pages and the other for emails. Your main concern with asymmetric ciphers, however, is how to keep your own private key out of the hands of attackers rather than whether you should trust the public keys you are given.

Information release

By accident or intention your personal or valuable information can fall into the hands of people you do not want to have copies. By stopping attackers running programs on your computer and carefully configuring your network programs and firewall you can block the most likely paths through which information can be released. The remaining routes to consider are the physical storage and the transmission of your information.

All the physical media you use to hold information can release your information, so hard disks, CDs, DVDs, tapes and memory cards need to be overwritten or destroyed before they are placed somewhere other people can access them. Re-writable media can be wiped fairly effectively by deleting all the files, then completely filling them with useless files by repeated copying. A special hard disk cleaning program may be needed for hard disks that contain sensitive information. Physical destruction of a hard disk is easy and quick for those too old to be re-used. A minute with a screwdriver to open the case and bend and scratch the disks will put the information beyond reasonable recovery. CDs and DVDs can be fed through a heavy duty shredder or scratched and broken.

Any form of encryption, even simple encryption with an easy-to-guess password, will drastically reduce the number of people willing to expend the effort required to access the information, and may completely prevent them reconstituting a fragment of a file to which they might gain access. Unless the rest of your security is exceptionally good, any very personal or confidential information should be encrypted. This is especially important where the physical storage could be mislaid or stolen, such as a laptop computer. Solid encryption and password management can allow you to safely store copies of your information on poorly-secured physical media.

Be aware that looking at or typing confidential information, such as typing a password, on a computer which is not yours exposes it to any programs running on the computer. A simple program called a 'keystroke logger' can easily record everything you type, and a 'screen capture' program can record the display.

Wireless connections will probably leak your information, but only to people within transmitting range. The equipment to intercept wireless transmissions is readily available and easy to use. Wireless tends to be slow and very difficult to secure compared with its wired counterparts, so unless you are an expert it may be better to use wires instead.

Email

Anyone on the internet can feed data into your email program by sending you an email, so your email program is a frequent point of attack. As with most programs you can reduce the amount of program code and functionality accessible to attackers by careful configuration. You should obviously switch off any kind of automatic running of programs you receive, but you should also turn off images that load from the web and automatic replies. By doing this you can deprive the sender of any indication that you have received or looked at the email, and so you will be doing your bit against spammers. The easiest way for us to reduce spam is to stop responding in any way to their emails - spam is only sent because people visit their websites or buy their products as a result of it.

It is easy to send an email that looks as if it has come from someone else. This is called 'spoofing'. A 'spoof' email can look exactly like an original. Only a small amount of the 'header' of an email is always likely to contain true information because it will have been added by the computer from which you obtain your emails. Email is normally presented to you in a way which does not show the real information that you have received. Learning how to read the complete unformatted information may allow you to determine the true origin of the email and identify 'spoofing' attempts and viruses.

Most spoof emails will be obvious attempts to obtain information such as your bank details or a password. A quick web search for a phrase used in the email may expose the deception because others are likely to have been sent a similar email. If after careful consideration you think the email may be genuine, you should still protect yourself against the possibility that it may be a spoof. You can do this by using a web or email address you have stored and know to be genuine. Failing this you should use the results of a web search rather than rely on any information in a potentially spoofed email.

Emails are not usually encrypted for you during transmission, and they pass through and are temporarily stored on many computers before arriving. This means they are fairly public unless you encrypt them yourself. A famous program for email encryption is Pretty Good Privacy (PGP). This provides a practical

implementation of asymmetric and symmetric ciphers and key handling. With PGP you can encrypt and digitally sign emails.

Web

Like emails, your web page requests and the information returned to you are not normally encrypted, and pass through and are temporarily stored on many computers, so they are fairly public information. But unlike email programs all web browsers have a simple standard way to encrypt this information. This is called Secure Sockets Layer (SSL). For ordinary purposes both SSL and PGP are impregnable if used properly. Whether your web requests and the pages they return are encrypted is decided by the person running the web server. Ordinary web pages that contain no sensitive information are not encrypted. This reduces the work the computers have to do and allows copies of popular web pages to be saved around the web for reuse, thus reducing the number of times the same information is transmitted. A web page that is not encrypted is therefore fast and efficient. Any page containing sensitive information, however, especially one where you type your password or credit card details, should always use SSL. Your browser indicates whether the page is using SSL - this is commonly indicated by a padlock icon which is locked on an encrypted page. Another distinguishing feature of an encrypted page is a web address starting with 'https' rather than 'http', where the 's' stands for secure. You should not trust a website that deals with sensitive information unless it uses SSL for all the important pages.

A website can also be 'spoofed' very easily. There are several ways to protect against being fooled by a spoof website. The best method is to get to the site from known good information such as your saved web addresses - sometimes known as bookmarks - rather than from something potentially unreliable like a web address in an email. Always bookmark any important pages so you can return to them and have an accurate record of the web addresses. If a page uses SSL you can also examine its 'certificate'. This is its public key and other digitally signed information which should relate to the site. Anything wrong with the certificate should make you suspect that the site is insecure and possibly spoofed. You can also detect a spoofed web page by studying its Uniform Resource Locator (URL) - its web address. A URL starts with 'http://' followed by a 'domain name' such as 'www.millstream.com' followed by '/'. The domain name is the same as the one you see on the end of an email address such as 'r.kelsall@millstream.com', and you can check on public websites to see who owns a domain name. The domain name reads from right to left in terms of importance. So 'com' in this case is the top level domain, 'millstream.com' is a sub-domain of the 'com' domain, and 'www.millstream.com' is a sub-domain of 'millstream.com'. The owner of a domain name can create whatever sub-domains they like under their domain name. Identical domain names in two URLs normally means both pages come from the same web server run by one person or organisation. Looking at the domain name of a web page will often help you identify the source of the information.

A 'cookie' is a small piece of data stored on your computer by your browser at the request of a web page you visit. The main purpose of a cookie is for the website to be able to know that it's still you when you move to another web page on the site. This allows the website to remember what you have in your shopping basket or to retain some preference you have set. Cookies are harmless except that they can link together the different actions you take on the web. Your browser can be configured to reject cookies - which may make shopping and similar sites not work properly, but will give the maximum privacy - or as a compromise between privacy and functionality set your browser to delete all its cookies when the program exits.

Wish List

Thank you for visiting this web page - your hit has been appreciated. If you are feeling generous I would like a small gift in return for this carefully composed information. Pick something from my wish list:

- Email me r.kelsall@millstream.com with your views about this page, or suggest a small improvement, clarification, correction or link.
- Send an email recommending this page to the people in your address book who will find it useful and interesting.
- Add a link to this page to your website or blog.

Links

Beginner's Links

The web will help you solve any security problem. You will have met Google but have you read Google Help and experimented with the advanced search options? If so you could try Copernic or read Fravia. You can test your ports using the Shields Up feature of Steve Gibson's website. For a better web browser visit the Mozilla website and try the Firefox browser. If you don't have a firewall on your Windows PC install Zone Alarm.