Defeating Security With Security

In the last couple of years ICT security awareness in companies has substantially increased. There are many factors which are responsible for this, but probably the most important are the following three:

1. the average security knowledge basis of ICT personnel has greatly increased;
2. companies are sustaining considerable economical losses due to the lack of ICT security;
3. laws, regulations and standards are obliging companies to address ICT security, and probably this is in practice the most important factor.

So, due to these and other reasons, we have seen and we are seeing a lot of activity to introduce or improve ICT security within companies.

But the main point of this article is to understand if what is usually done is really increasing the security level of ICT within companies or it is just moving the vulnerabilities from one target to another without really increasing the security, if not decreasing it, while at the same time making us feeling more secure anyway. It should be obvious that feeling and believing to be secure when the contrary is true, is the worse possible situation.

To understand what is going on, we have to balance various factors. On one side the fact that the management realizes that ICT needs security is a big step forward. Investments are done, jobs appointed and results expected mostly in a modified form of ROI (Return Of Investment) which can be summarized as: *we spend today not to loose more money tomorrow*.

This is good and opens completely new scenarios for ICT and particularly for ICT security and audit.

But if we check what actually is done at the technical level, the situation doesn't look so good. Indeed the most common approach to security can be summarized as: *buy the licences for the anti-virus and the hardware for the firewall, but nothing else*. Unfortunately too often the management perceives ICT security as to *have* the anti-virus and the firewall and to produce enough paper to comply with any regulation, certification or law requiring some sort of ICT security.

It is true that ICT is generally leading towards being a commodity, but ICT security cannot definitely be considered as a hardware lock on a door which you just buy, install and forget about it. As we very well know, ICT security is a process which constantly changes and which is difficult to implement. It requires organization skills, a complete understanding of the needs and implementation of ICT within the company and an understanding also of the principal technical issues.

It is interesting to note that in an attempt to clarify better what it is ICT security for a company, international standards are getting more detailed. The high level approach requires in practice only to follow the well known cycle of finding the vulnerabilities, evaluating the risks, and implementing the countermeasures, cycle which can be interpreted in many ways, from the most generic to the most technically detailed. But obviously in practice the generic approach is usually followed. To guarantee that some technical points are always implemented, standards, regulations and laws are adding more detailed requirements and a good example of this is the PCI-DSS standard. This standard, which applies to the companies and the ICT systems which deal with credit card payments, is forcing many financial and commercial companies to face some concrete security issues.

To make this discussion more clear and to try to understand if our efforts in implementing ICT security are effective or not, in the reminder of this article we will discuss how to comply with a practical requirement now present in many standards, regulations and laws. We can generally formulate the requirement as:

> Network connections used for remote managing of ICT hosts and applications, must be protected by Strong Authentication and Encryption.

From the security point of view, this is a basic pillar of correct ICT system administration. But to an ICT manager not skilled in security, this could sound just as a useless and expensive complication. Too often in my personal experience I keep hearing complaints that strong authentication and encryption are not needed for many reasons which can be summed up by saying that the company LANs are already enough secure. (It is widely accepted that WAN administrative communication must be protected, so from now on we consider only LAN communication.)

In some cases, the reason why LANs are considered secure and the administrative traffic should be allowed in clear, is based on the wrong assumption that switches are security devices. Indeed some believe that using switches instead of hubs can guarantee that network packets can be seen only by the rightful sender and receiver. It is true that a switch in general does not broadcast packets to all hosts but sends them only to a single destination host. More precisely, in a switched LAN the switch sends to a host connected to one of its ports only packets addressed to one of the MAC addresses which the switch has detected being active at that moment on the cable attached to that port. But notice that the assignment of MAC addresses to ports is usually a dynamical process. Indeed usually if a PC is disconnected from one port and reconnected to another, it keeps working since the switch immediately realizes the change of the port to which the PC is connected and sends the packets down to another cable. At first sight it could look like that it is impossible to intercept packets

addressed to another host, but this is not true. There exist many kinds of attacks, some fully automated, which allow an attacker on a LAN to redirect traffic and/or intercept the traffic to/from another host: some of these attacks are the following: the Switch Port Stealing attack, Arp Poisoning, DHCP poisoning, CAM table overflow, STP Mangling, VLAN Hopping etc. All these attacks require the attacker to act from an host directly connected to the LAN without firewall or router in between. If one of these attacks succeeds and the administrative traffic is not protected end-to-end, the attacker will be able to gather the administrative credentials, modify on the fly commands issued by administrators and so on. One practical defence against these kinds of attacks is given by the so called Network Access Control (NAC) infrastructures, or similar name depending on the vendor. The idea of NAC is to allow a device to connect to a LAN only after some kind of authentication at the application level, that is identifying the user and not the machine. Once the user is authenticated, the host is given, usually via the DHCP protocol, an IP address and is allowed to connect to the LAN through a specific physical port of the switch. If the host is connected to another port, the application level authentication has to be repeated. The problem with NAC is that its main purpose is to prevent infected PCs to connect to the LAN and not to prevent other kinds of attacks to the LAN. For example, printers are usually not protected by NAC and in this case an attacker can easily obtain the MAC address of the printer, program this MAC address on his portable PC and connect it to the LAN instead of the printer.

In conclusion it is often easy to find attack vectors of this kind which can be promptly demonstrated to the management and which require a low level of technical skills.

A second complaint against the protection of the administrative traffic is that it is too expensive with respect to its benefits. This is usually not true either since administrative traffic is little, its encryption does not require hardware support and can be easily implemented using free or very cheap (compared to normal enterprise software licence fees) software tools, as we will see.

A third complaint is that this security issue is not a security priority for the company ICT system, since usually there are other vulnerabilities much more crucial that have to be fixed first. This is absolutely true, but if the administrative traffic is protected, it is usually possible just by this to prevent an escalation of the exploiting of the vulnerability from a local one to a global one, that is one which allows the attacker to fully control the company network.

Assuming that it has been decided to protect the administrative traffic with strong authentication and encryption, the technical solutions are easy to find. The most used tools are based on the Secure Shell (SSH) protocol, as specified in RFC-4250 ... 4256 , which has been developed exactly to this purpose.

The SSH protocol and the tools based on it, allow to create encrypted communication between

hosts, tunnels, redirection of tcp ports and services, terminal sessions and so on, and offer various authentication schemes. The most common authentication schemes are the usual username+password and private+public keys.

Username+password cannot be considered a strong authentication mechanism and we all know how difficult it is to correctly create and manage passwords. For obtaining strong authentication it is then usually adopted the use of private+public keys and cryptographic algorithm like RSA. We do not describe here how public-key cryptographic algorithms like RSA achieve strong authentication by means of using private and public keys, the important point for us is that they are considered secure, they are implemented in tools like SSH and easy to use. Indeed in their simplest set-up private+public key authentication does not require the user to remember or manage keys or passwords and allows for instant and secure communication to all hosts.

But is it really secure?

Let us discuss more in details how SSH works and how it should be used versus how it is usually used.

To use a SSH tool with private+public key authentication, one first has to create a private+public key pair. This is often done just by clicking on a button. The private key is the one that identifies the user and should be kept protected only on the host (the personal workstation) from which the user connects to the other machines. In principle the private key should reside on a smart-card or similar device, but this would make much more expensive and difficult to manage the solution. So for the moment we do not consider the use of smart-cards or similar devices to protect the private key, as this is the most common situation.

The public key is then distributed to all hosts and all accounts to which the user needs to connect to and inserted, for example, in a file called authorized_keys in the destination user account. Having done this, the user from his personal workstation can connect directly to any other host in a secure way without using a password. Indeed the remote host, using a cryptographic algorithm, verifies that the user connecting workstation has the private key which corresponds to the local public key which authorizes the login. After this strong authentication procedure has succeeded, all communications between the two hosts are encrypted.

But what if the user needs to connect from an host to another, neither of which is his personal workstation? In principle the user should create a private-public key pair for each account to/from which he can connect on every host and he should distribute the public keys to any other account on all hosts. But this is the well known key distribution problem of public-key cryptography! The situation easily gets out of hand since managing all these keys on all hosts and accounts becomes immediately a daunting job. To address this problem, Public Key Infrastructures (PKI), which in

simple words means digital certificates and Certification Authorities, have been introduced. But this solution is surely not acceptable for the problem at hand since it is definitely too complex to manage and too expensive.
So what is it usually done?
Quite simple! Since the private key is in a file and not locked in a smart-card, the user just copies both the private key and the authorized-key file (containing the public key) to all accounts on all hosts. Since on all accounts there is the same private key, from any account the user can connect to any other account without using a password but with strong authentication and encryption! And even better if the same private key is used both for personal and administrative accounts since this gives instant access to everything.
(Notice that sound security practices would require that a user always first logs-in in a host in a personal account and then acquires locally the administrative privileges. But this second step requires usually a local username+password and too often, *to make life easier* and use SSH to its full extents, users connect directly as administrators to the remote hosts.)

If we could say that this solution implements to the letter the requirement of administrative traffic with strong authentication and encryption, we must admit that it definitely violates its spirit and that it introduces new vulnerabilities and attack vectors.

Indeed, in the situation just described, if an attacker is able to break into an account so configured, he has instant access to everything, whereas before the introduction of SSH he would have had to obtain the passwords for example by sniffing on the network. Another example is if the attacker has physical access to the unlocked workstation of an administrator, again in this situation the attacker would have instant access to all systems without need of knowing any password.
Looking at it from this point of view, the situation just described looks quite hazardous, to say the least.
Obviously the experts who have designed the SSH protocol had realized these risks and indeed there are features in SSH which prevent and reduce these risks.

But we should stress that the problem is not with the protocol itself but with how it is used in practice. This is a really important point we want to make and which has lead some companies to use SSH with username+password and not private+public key considering less risking the managing of passwords that this way of managing keys.

But the SSH protocol offers also the possibility of protecting (i.e. encrypting) the private key with a pass-phrase (oh yes, another password!). Now assume that the private key of the workstation of the administrator is in a smart-card, protected with a PIN and not exportable, and that each account has a private key protected/encrypted with a pass-phrase too. The previous scenario is completely changed:

1. we have again to manage pass-phrases, that is passwords
2. but now all communications are encrypted and nothing can be sniffed on the network
3. moreover passwords are not exchanged on the network because pass-phrases are used only to decrypt the private key on the host originating the connection
4. access to one account does not allow connections to any other account if the pass-phrase is not known
5. even stealing/copying the private key is useless without knowing the pass-phrase (but, as with any password, the pass-phrase is subject to the risk of being easy to guess).

Finally, as a practical help to manage pass-phrases, many tools provide a *key-agent* which is able to store securely the pass-phrase for the duration of a terminal session. The way to use the key-agent is to load the pass-phrase in it at the beginning of a terminal session after which the key-agent provides the pass-phrase to SSH every time SSH needs to use the private-key.
Obviously this introduces the risk, smaller than the previous one, of an attacker entering in an active terminal session to steal the pass-phrase. But this is much more difficult to do and often requires special tools like a key-logger or the presence of a trojan on the workstation of the administrator, which by themselves are by far greater problems.
(For completeness we should mention that the key-distribution problem does not have a real solution and that some compromises must be found, like the one to use the same private+public key only on a group of hosts and accounts and not all. Notice also that the same private key can be protected/encrypted with different pass-phrases on different hosts/accounts.)

So it is possible to use SSH securely and by this to really increase the security of the company ICT system. The real question is if SSH is used correctly or not, and unfortunately too often the answer is not.

Obviously we can extend these considerations to almost any other security measure, procedure, tool, device, protocol etc. etc. The lessons that we must learn and remember are

1. a security mechanism does not *automatically* increase security;
2. incorrect use of a security mechanism can actually decrease security by opening new

vulnerabilities and attack vectors;

3. Security is hard.

Andrea Pasquinucci
PhD CISA CISSP
http://www.ucci.it/