The Good, the Bad and the Ugly : or how a chain of insignificant events leads to a dramatic outcome

Autor(en): **Grundschober, Stéphane / Etienne, Pascal**

Objekttyp: **Article**

Zeitschrift: **Comtec : Informations- und Telekommunikationstechnologie = information and telecommunication technology**

Band (Jahr): **76 (1998)**

Heft 12

PDF erstellt am: **21.07.2024**

Persistenter Link: https://doi.org/10.5169/seals-877347

The Good, The Bad and The Ugly

or how a chain of insignificant events leads to a dramatic outcome

STÉPHANE GRUNDSCHOBER
AND PASCAL ETIENNE, BERN

Segmentation Fault: Core Dump" followed by "Kernel Panic" were the last two error messages the computer spitted out before freezing... It was not a good programming evening for *Count Zero,* as he likes to be called in the underground scene. Nevertheless, he is quite respected by the other crackers. They meet electronically via IRC (Internet Relay Chat), a system that lets them communicate in real time via the screen and keyboard of their computers. He proved that he was able to penetrate computer networks, often because of a faulty configuration of the system. There is no money in the game, it's just for the fun. He especially enjoys modifying the home page of a corporate web server, usually with something unpleasant for the corporate image. The best thing is: Everybody can see his work, his skills, and he laughs when thinking at the poor system administrator that must explain that to the board of directors!

The computer just crashed as he got a phone call from *Dexter.* He is a newbie, a young teenager who wants to crack into systems, but does not know anything. He absorbs any information he can, and is very obstinate at this job ... to the misfortune of Count Zero who cannot have peace without leaking out some interesting tool or method to crack computer networks! But this time, Dexter has something unusual for Count Zero: a SecurID[1] card he has found in the wallet someone has lost.

"This is great Dexter! This card is a strong authentication mechanism, used by remote access servers (RAS) to allow entry to a company network via modem."

"What does that mean, strong authentication? I thought that if I want to connect my computer to a network, all I have to do is to type my username and the corresponding secret password. I am the only one who knows the password, so the access server can trust that its really myself and not someone else who is using my username. What is the problem with this method?"

"Indeed, that's the simplest way to authenticate the identity of a user. But this has some drawbacks. The first one, is that it's always the same password that you provide. Imagine that someone watches over your shoulder when you are typing your password and memorize it. He can reuse it without a problem!"

"Oh... I don't know how to avoid that, except by ensuring that no one is behind you..."

"Come on Dexter, I'm sure you will find at least one solution to that problem!", said Count Zero teasing Dexter.

"Hey, I'm not that dumb... Let me think... OK, the user should change his password regularly, so that even if someone steals it, it would not last for ever."

"Good! That's actually the right solution. But this works only in theory. In the everyday use, people do not like to change their passwords. Indeed, they have to remember a new one, they do not type it as fast as the old one, ... They do not like that. And if the administrator insists on users changing passwords very often, they will simply add a number at the end of their password..."

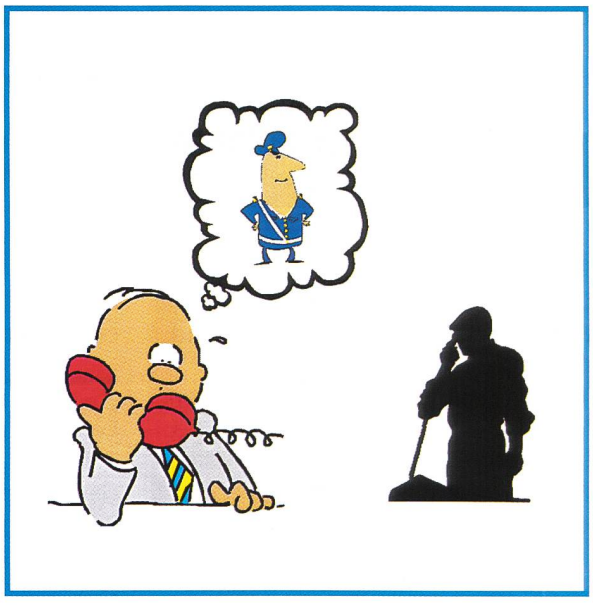
"That's really stupid, or just less stupid than the system that accepts this new password ... The guy who stole it can try to add some numbers at the end, with a good probability to find one." said Dexter proudly.

"At this point you will understand the usefulness of the SecurID card. The card displays a number that is used as password. More precisely, the user who wants to connect to his corporate network will enter his name, a secret password and the number displayed on the card. Only if these three items of information match, will access be granted. So far, nothing very new. The key point of these cards, is that they change the displayed number every minute, as the server does. Even if someone gets a valid number, one minute later it will be useless! Moreover, these numbers are one-time passwords. That means that once you used one for a successful authentication, you cannot reuse it. You have to wait one minute. This really doesn't help the usual crackers, but today we have the card, so we should be able to do something!"

"And what about the secret number? We do not have it ... On the card there is only the name of the user, Mr. Littelstone, and the corporate name written."

"You said the name of the user is written on the card? Give me 10 minutes, and we have the secret number! I will show you an example of *social engineering."*

Indeed, the first step is to find the home phone number of the careless user. Then, a little bit of acting skills does the rest:



Social engineering.

"Hello Mr. Littelstone, this is the computer center. Are you currently connected to the corporate network?"

"No, not at all, I lost my SecureID card today. Why are you asking?"

[1] SecurId, a product of Security Dynamics, is one among many authentication products. Other companies propose similar products.

"Mmmh. That's what I suspected. We are seeing unusual activity here, and it seems that it's someone using your identity. We are witnessing a hacker attack, this is serious. I hope it is not too late to block him... I need your personal password and username to block the RAS access, so that you will not be held responsible for this attack."
"Oh my goodness! My username is *little1* and my password is *8z4p,* and please hurry!"
Et voila! It seems the security manager has some communication problems with his users ... Too bad!
Thus, when Count Zero rejoined Dexter, he had all the information necessary to impersonate Littlestone. One piece is still missing, the remote access number.
"This should not take long to find it," said Count Zero. "It's quite a big company and they must have their own phone exchange. Do you see the phone number of the company in the phone book? Any phone number with the same prefix should belong to them. With my *War Dialer* software, we will try one after the other the numbers in that range. If a modem answers, we may have found the Remote Access Point. It may be possible to get a modem connection to something else as a RAS, like a fax-modem. But if our identification is rejected, we can proceed with the remaining of the scan. The other phone attempts will not get noticed, because it's late now, and nobody should be working there."
Twenty minutes later, Count Zero and Dexter are logged on the corporate network by way of a PPP (Point to Point Protocol) link.
"Wow!", says Dexter, excited about his first successful break-in.
"Well, this is the first step, but it doesn't let us do much... We need to go further. Let's check the remote network to see what machines are running."
Count Zero brought with him a couple of small programs. One of them is a *port scanner* that probes any remote computer, and checks what services are running on it.
"Look at that machine: There are lots of services running on it. If we have luck, it's a new UNIX machine that has been poorly configured. UNIX is a very powerful operating system (what makes a computer run), but absolutely not fool proof. And fools abound... because either they don't bother or simply don't know how to configure their machines securely. Let's try it. First, I check what disks are exported via NFS. I use the *showmount* command."
"Wait, what is NFS?"
"Oh, Dexter, I thought you knew that already... NFS stands for Network File System. It lets you have access to a remote file directory as if it were local. In technical jargon, we say that we are *mounting an NFS drive.* Usually, you can define what computer (or client) has the right to access such a remote file directory. But if it is badly configured, everybody can mount it."
"Like that one?" Dexter was pointing the name of a drive on the screen that apparently is shared to *the world.*
Dexter is right, and Count Zero did not wait long before mounting that filesystem. But what a surprise: this is the *home* filesystem where all the personal directories of the users of that machine

Glossary

Bug:
An unwanted and unintended property of a program or piece of hardware, esp. one that causes it to malfunction.

Brute force attack:
A brute force attack tries every possibility until success. For example, a brute force attack against a passworded authentication system is to try every possible password.

Cracker, or Dark-Side Hacker:
A criminal or malicious hacker. From George Lucas's Darth Vader, "seduced by the dark side of the Force". The implication that hackers form a sort of elite of technological Jedi Knights is intended.

DoS, Denial of Service:
Attack with the sole purpose to disable a service, a computer or a whole system.

Exploit:
A small program that exploits a bug in the operating system of a computer to get super-user (administrator) privileges.

Firewall:
A dedicated gateway machine with special security precautions on it, used to service outside network connections and dial-in lines. The idea is to protect a cluster of more loosely administered machines hidden behind it from crackers.

Hacker:
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.

IP Spoofing:
Forge Internet Protocol data packet with the source IP address of another computer. Efficient attack when the target computer does authentication based on the source IP address.

Port scanner, network scanner:
A program that checks remotely the presence of a computer and what network services are running.

Sniffer:
A sniffer is a program looking passively at every packet transmitted on a network for sensitive information like passwords.

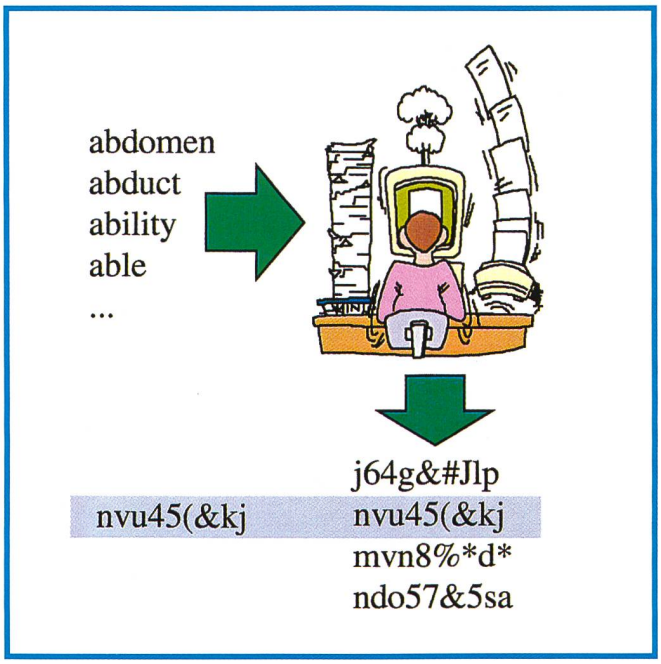
Social engineering:
The aim is to trick people into revealing passwords or other information that compromises a target system's security.

War Dialer:
A cracking tool, a program that calls a given list or range of phone numbers and records those which answer with handshake tones.

The Hacker Dictionary: http://earthspace.net/jargon/

are stored. All their files, their e-mail, their reports are accessible. But this is not what interests Count Zero. He wants to write to a specific configuration file of one of these users, the *.rhosts* file.
"What's that file?" asks Dexter, trying to remember were he saw that name.
"This file will be our way into that machine. It's a feature of the UNIX operating system: Usually, to connect to another computer, you have to enter your name and password. This can be annoying if you do it many times a day. The rhost mechanism lets you tell the system to accept connections originating from a specific host and user without prompting for a password."
"I missed something. How do you configure that rhost mechanism?"
"Let's take an example. Say I have an account on two computers. One is in my office, and I have the username *countzero*. The second is in a laboratory, configured with my account as *countlab*. I have to write a report, on my office machine, and need to access data on my laboratory machine. It's painful to enter a password each time when I log to the lab. Therefore I edit an *.rhosts* file on my lab account, where I say that the user *countzero* from the *office computer* can log in without a password. Once it is done, I can use the command *rlogin – l countlab lab-computer* from the office computer, and I am logged in directly without having to enter the *countlab* password! There are a couple of other commands, the 'r' commands, that work under the same principle: *rcp* to copy files from one system to another, *rsh* to execute a program on the remote computer, ..."
"All right, it's a pretty useful system. But what is the problem with it?" said Dexter, wondering if his mind is not devious enough to find the flaw...
"If the corporate network is well secured, then that system is OK. But as you know, our computer was not intended to be on that network... Look: I just have to configure my computer, and now I can add remotely '+ +' to that file: anybody is now authorized to connect from any other computer. The trick is that normally we shouldn't be able to write to the *.rhost* file. It's the bad NFS configuration that let us write to that file, and now we can log in from our computer!"

"...I know I've made some very poor decisions recently, but I can give you my complete assurance that my work will be back to normal... I've still got the greatest enthusiasm and confidence in the mission, and I want to help you...Dave...

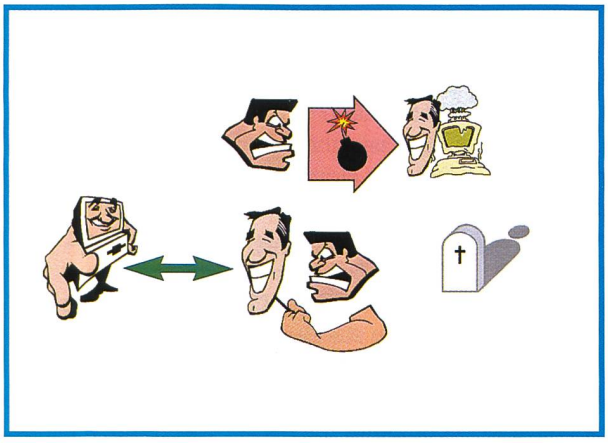


Password cracking.

stop...stop, will you...stop, Dave...". Arroway was watching on TNT his favorite Science-Fiction movie when he received a message on his pager. As a system administrator, he has to be reachable in case there is a problem with his corporate computer network. This time, it was a message automatically sent by an Intrusion Detection System he installed a couple of days ago: "IDS – Portscanning detected". "Mmmh... another false alarm from the System," he thought. "I already received two other alarms today. I should check the configuration tomorrow morning..."

"Now that we are logged on that machine, we should try to get super-user privileges. This will let us take entire control of this computer. To do that, we need to find one system program with a bug. The sendmail program is known to have some programming errors." said Count Zero, which seems to find this system increasingly interesting.
"Oh yes, I know." said Dexter, happy to show he wasn't completely ignorant. "Sendmail is the program that manages the transmission of e-mail from one computer to another, but it's complex and large. Most complex programs have bugs somewhere..."
"Exactly, and we are lucky: The version of sendmail installed on that machine, version 8.8.4, is not the latest one. I have a small program that will exploit a tricky bug and let us get the identity of the root user. And the *root* user is the super-user, with all the privileges!"
"How does it work?"
"Well, it's a little bit complicated, but it uses the fact that sendmail runs with these same high privileges. The script will trick sendmail to run an other small program that we provide and that will open the door for us. This kind of exploit is very common. Although they are rapidly patched by the manufacturers, new ones are constantly discovered."
As he explains this, Count Zero launches the exploit, and one minute later he has the famous '#' prompt, indicating he has gained super-user privileges.
"What are we doing now?" asks cluelessly Dexter. "Is there something interesting on that computer?"
"It doesn't seem so. A badly configured machine usually does not contain sensitive information. Otherwise the administrator would have better taken care of it... But we can try something, and that will perhaps let us go further into the network. We will sniff the network traffic."
"Sniff? Like a retriever? If you are hungry, we can order a pizza, you know..." said Dexter almost laughing...
"No stupid, *I* will not sniff the network, but a program will. And you are right, it's like a retriever: The program will look at specific information. Indeed, a computer physically connected to a network can see all the data packets transmitted on that section of the network. We say that the network is shared, because it's one copper wire that goes through a couple of computers to connect them together. When a computer wants to transmit

data, it listens first to the wire. If nobody is sending data, then it starts transmitting. Otherwise it waits a random time before trying again."
"OK, I understand now why the sniffer can get all the packets transmitted on the wire. But why do we need to be super-user to do that?"
"Usually, a computer records only the packets that are for him. When we have high privileges, we can instruct the electronic of the network interface that we want to capture all packets. Not only those destined for us, but also the packets sent between other computers."
"You said it will look for specific information. What info and to do what?"
"You know that all these protocols, that define how computers talk to each other on the Internet, are quite old. You weren't born when people implemented the first versions. And in those days, network confidentiality was not an issue. The whole Internet and it's protocols were designed for high availability. As a confidentiality, the authentication procedures are quite simplistic. Take the *Telnet* protocol, that lets you connect to another computer. The remote system asks you for your name and password. If these two match his database, it grants



Spoofing.

you access. The problem is that your password is transmitted "as is" on the network! With our sniffer we can catch these packets, and record username/password pairs... And it's not the only protocol that is so naïve!"
"OK, that's for these old protocols. What about Windows? I've heard that the passwords are hashed. Your sniffer doesn't help in that case. The hash function is a one-way function with the mathematical property of not being reversible. Even if you can get the right packet, you cannot find the original password back!"
"You are right, from the hashed password you cannot get the original directly. That's what the programmer thought. But there is an other method: You make the supposition that the password is a word that you can find in a dictionary (or any other collection of words), or a combination of two of these words. You then hash this dictionary like Windows would do (the algorithm is known), and compare each result with the hashed password you sniffed. If there is a match, then you know that the original password is the one you used!"
"But this must take an eternity to hash all these words!"
"Not that much. It takes less than a minute on a Pentium 200MHz to check about 800 000 words... And you know, if there is something you can find everywhere..."
"... it's foolish people, that would choose a common word as password!" interrupted Dexter in a burst of laughter.
"You learn fast! But I have to concede that everybody is not that dumb. Some use incomprehensible passwords, like 'audgrem'. To crack these passwords, you can do what is called a *brute force attack*. It's the same principle than the dictionary based attack, but instead of using a list of words, you search exhaustively every possibility of passwords. For example, you could try to test every password composed of six letters. This would take approximately 6 hours."
"Wow, that's much longer than the dictionary attack. But what happens if people use upper and lower case letters, along with numbers and special characters like '$#&'?"
"Then we cannot do much about that. It would take months or years to check all the possibilities. On the other hand, we need only one user with a bad password to get into a system... But let the sniffer do his work. We will come back tomorrow morning to gather the password harvest."

Arroway's brain was a little bit fuzzy this morning. The movie last evening extended too much in the night to let him wake up completely today. A good double expresso with a high caffeine level – one of the rare legal drugs without too much taxes – should activate the remaining sleepy neurons.
The timetable for today has not changed: Archive the backup tapes in the safe, restart some servers that expectedly crashed, complain to the manufacturer about this new network equipment 'that will be delivered tomorrow' for two weeks now, learn the last voodoo tricks to make PC's programs do what the user wants and not the opposite, water the green plants, get the latest patches regarding the year 2000 problem, ...

"Look Dexter: 23 passwords in twelve hours! Issued from Telnet sessions, but also FTP and POP ones. I told you there are many unsecured protocols in daily use..."
"It's a nice program your sniffer, I see it records also the IP addresses of the communicating computers. Did you see that one? The addresses are quite different of the others..."
"That could be interesting. As you know, each computer has its own IP address (which stands for Internet Protocol) that permits the data packets to reach the correct destination. These addresses are then grouped in subnets. This helps the administration of the network, and usually reflects the physical architecture of the network. In that way, the IP address you spotted seems to be on a separate subnet. Let's try to connect it with Telnet."
"Hey, it doesn't work... The connection failed... What does that mean?"
"That's strange... Even a *ping* command, that checks if a computer is reachable, does not work. It's like if our data packets cannot reach the destination, as if they were filtered out..."
"I must have missed something... How can the sniffed computer establish a connection to that other subnet if all the packets are magically filtered out?"
"First it's not magic. The special computer – the router – that connects the two subnets is able to filter what packet it will transmit and what other it will reject. We call such a specially configured router a *firewall*. Second, it seems that the packets coming from our hacked computer are rejected, and not from the one the sniffer has seen."
"So we cannot go further until we manage to make the firewall think we

are that other machine..."

"Exactly! And how does the router know our identity? Simply by looking at the source IP address: Every packet contains two IP addresses. One indicates the destination of the packet, and the other indicates who has sent that packet. It's useful for the destination computer in order to send the replies. Therefore, if we want to make the firewall think we are an authorized computer, we will forge packets with a false source address. This technique is called *IP spoofing.*"

"But the replies will not reach our computer! They will go directly to the impersonated machine!"

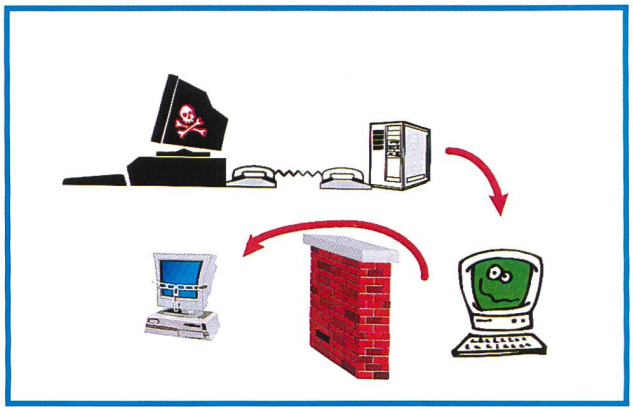
"Yes, that's true. And this has two repercussions: First, as you say, the answers will not go to our computer. This is not too big a problem. Indeed, we will see these packets passing. We will then be able to forge new packets as if we received these packets and follow the protocol."

"Really? What does the protocol look like?" asked Dexter innocently.

"You are going to kill me with your questions! OK, I will not go in all the details, but it works like that: To open a reliable connection, the system uses a protocol called TCP, Transport Control Protocol. This protocol assures, after a setup phase called *three-way handshake,* that the data transmitted over multiple packets is correctly retrieved on the remote computer. Indeed, if a packet is lost, TCP will make the sender retransmit that packet. TCP will then insert the retransmitted packet at the correct place in the stream before it delivers the data to the processing program."

"And you want to generate *by hand* each packet, simulating that protocol? It must be very difficult!"

"It's a little bit painful, but I will not do it, a program will. And computers love to do painful tasks!" said Count Zero self-confident. "The second problem is when the response packets from the target computer will reach the real machine. It will not understand what that packet is: acknowledgments for a connection it never started... It will think there is something wrong and sends a message saying 'stop! I did not start a session!'. And all our impersonation efforts will be lost, the connection will be closed. We have to prevent this to happen. We should in a way or in another 'surgically' disable the service dealing with these specific remote connections. One solution to this problem is to flood the real computer with open connection requests but without completing the protocol, without acknowledging the *Connection responses* that the flooded machine will send. This has the effect of consuming lots of resources. The computer will be so overloaded that it will not notice the packets coming from the computer we are spoofing, and will not close the connection in our back."



Bypassing security sences.

"I see. It's a DoS attack, a Denial of Service against the machine we want to impersonate so that it lets us do our spoofing attack against the target... Tricky..."

"And it's a subtle one, disabling just what we need to. This attack is called *SYN Flooding* because the connection request packets are commonly called SYN packets. There are other DoS attacks much more powerful, that take a whole computer down, but using that kind of attack is not a good idea if you want to do a stealthy intrusion! But first I have to download and compile some helper programs that will do these two attacks. This will take a couple of minutes."

"Hello Mr. Arroway, this is Mr. Littlestone. What's new with the hacker?"

Arroway thought he had just entered the fourth dimension when he received that phone call. Who is Littlestone, and what hacker?

"Yes, you must remember, you phoned me yesterday evening. I told you I lost my SecurID card and you said that you were hunting a hacker that was using my identity."

"I did not call you. And what did you loose? Your Sec..." Arroway's blood decided at that moment it was not really necessary to irrigate his face...

Indeed, the beeper message last night was not a false alarm. Someone was really attacking the network with the benediction of the system, confident in the SecurID authentication... This is a red alert case. What did this cracker do? Has he destroyed sensitive information, crashed servers, or blocked access to legitimate users? It seems not yet, otherwise he would have already received hundreds of phone calls from angry users... But this may still happen!

The first task is to check the remote access log. "It's not a bad dream. There is an entry yesterday evening... But what is this? There is a second entry, and it's now! The fool, he is connected right now!"

Arroway is about to force the disconnection of the cracker when he thought: "Wait, I don't know what he has done, what computers he has compromised. I should try to spy on him now, to get clues about his actions. I could always disconnect him out if I keep an eye on him." It's not easy to trace the intruders tracks. Arroway first sets up a kind of network sniffer to get a dump of all the connections, and especially the ones belonging to the cracker. From this trace, he will know to what computer inside the corporate network the bad guy is connected to.

"OK, the helper programs are compiled, we can start with the flooding."

At the time Arroway got the name of the computer the hacker was using, a blinking red icon on the network management console signaled an overloaded computer. He has to think fast. He cannot simply close the cracker's connection. He has to know what the pirate is doing, but he has to find that quickly before the cyber-gangster disable the whole network. There is a tool, *ttywatcher,* that will display exactly what the intruder sees on his screen, along with what he types. This is the perfect spying tool on a Unix server. Here is what the tool displayed:

```
# spoof_telnet 10.162.240.65
business12
```

"Ah ha, we are interested in the business department. Too bad, it's filtered! You will not get through!"
But the spying tool dumped two lines unexpected by Arroway:

```
UNIX(r) System V Release 4.0
(business12)
login:
```

"What??! How the hell did he go through my firewall??!!" Arroway rushed to the firewall configuration, looking at the rules, trying to find what he did wrong. But he did nothing wrong, except thinking that a firewall was THE solution to protect a network.

By the time he was looking at the configuration file, Count Zero and Dexter penetrated the machine without difficulty by reusing the sniffed passwords they collected.
"That's it, we are in! Let's look around to see what we have."
Exploring the filesystem, Count Zero quickly discovered some Word documents, with appealing titles. One of them took only ten seconds to dump on the screen and saved at the same time. These ten seconds were enough to reduce all the security measures to nothing, as this confidential document details the new restructuring plan of the company. The delivery of this information to the public, and especially to the stock exchange, will have significant consequences.

When Arroway looked again at the screen, it was too late. Although he quickly disconnected the hacker and removed the Remote Access Account, the confidential document was already out of the company. And that should really not have happened...
"Well, I think it's time for me to clean up my Curriculum Vitae and find a company that does not require a letter of recommendation ..."

This is fiction, but reality is behind your door. Remember, your personal password must remain secret. Think about what you can do in your everyday business to break such a chain of events... [1]

Stéphane Grundschober *studied Communications Systems at the Swiss Federal Institute of Technology (EPFL, Lausanne) and Eurecom (Sophia-Antipolis, France) from 1993 to 1998. He made his Diploma thesis at the IBM Research Laboratory Rüschlikon in the field of Intrusion Detection. Since 1998 he is working at Swisscom CIT-CT-TPM on computer security and fraud in telecommunication services.*

Pascal Etienne *followed an apprenticeship of electronics at the «Ecole des Métiers» of Fribourg. He then got his diploma of electrotechnique (telecommunication option) at the Engineer School of Fribourg. Since 1997 he is working at Swisscom CIT-CT-TPM on computer and Internet security.*