SECURICOM 96 - Security Evaluation of a PC Network based on Intrusion Experiments

Ulf GUSTAFSON

Erland JONSSON

Tomas OLOVSSON

Department of Computer Engineering
Chalmers University of Technology
S-412 96 Göteborg - SWEDEN

email: ulfg/jonsson/olovsson@ce.chalmers.se

ABSTRACT

This paper presents an intrusion experiment in which the target system was a Personal Computer network connected to a Novell NetWare 3.12 server. Undergraduate students with little security expertise and hardly any knowledge of the system served as attackers and were given the task of performing as many intrusions as possible. The objectives of the experiment were twofold: first, to learn more about how to gather and process data from intrusion experiments and to form a methodology applicable to a generic class of computer systems; and, second, to find out whether it is actually possible to create a secure system based on insecure PC workstations. This paper deals mainly with the latter objective, and investigates how and to what extent unevenly distributed security features, such as a "secure" file server with untrusted clients, affect overall system security. Furthermore, in experiments, as opposed to real life situations, it is possible to collect information about how the attacking process is carried out.

Before the experiment, we anticipated that the attackers would create Trojan Horses on the clients to spoof other users during the login process, but we did not expect them to find as many serious vulnerabilities in the concept as they did. The experiment shows that untrusted PC clients have ample intrusion possibilities, and that the vulnerabilities can not be compensated by security features elsewhere in the system. Novell has undoubtedly spent more effort in securing the file server and its assets than in securing the clients in the system. This paper contains a summary of the security problems the attackers found, from which it is evident that several new security mechanisms must be added before a NetWare 3.12 system can be regarded as secure.

Keywords: Security, Vulnerability, Intrusion, PC Network, Experimentation, Tiger Team.

1. INTRODUCTION

Many vendors of network-based systems claim they can offer a solution with a high degree of security. In this study, we studied a specific system, a Novell NetWare based system consisting of insecure PC workstations connected to a physically secured file server. There were two major issues we wanted to address in this study: first, is it *really* possible to create a secure system based on insecure PC workstations and, if not, what are the limitations of such a design; and second, how much *effort* does it require to break into such a system?

Security evaluation using so-called "Tiger Teams" has been performed for quite some time [Attanasio 1976], [Herschberg 1988], [Goldis 1989]. A Tiger Team is a group of people that is very skilled and knowledgeable in the security domain, having deep knowledge about the system and an awareness of potential vulnerabilities. Our experiment however, is quite different. The alleged attackers in this experiment are "normal" users and thus we expect that the findings apply to a wider set of systems, both because it was a "standard" system and because it shows what ordinary users can do with such a system. The ultimate goal of this type of experiment is quantitative modelling of operational security, i.e. to find measures of operational security [Littlewood et al 1994]. The idea is that the measure of the security level of a system, captured by the above mentioned parameter "effort", should reflect the intuitive notion of "ability to resist attacks".

We have previously performed two similar experiments. In those, the target system was a networked Unix operating system [Brocklehurst et al. 1994], [Olovsson et al. 1995]. The collected data from these experiments has been used to draw quantitative results on the attacking process and the behavior of attackers [Jonsson and Olovsson 1996]. This time, in an attempt at diversification, a PC network was selected in the hope that the change of target system would lead to new findings with respect to security modelling, a better knowledge of generic vulnerabilities in its design and greater knowledge of the attacking process. Details on the carrying out of the experiment as well as the collection and interpretation of the data is found in [Gustafson et al. 1996]. The present paper, however, concentrates on the exploited vulnerabilities and the attacking process, and attempts to draw some general conclusions from the collected data.

2. THE SYSTEM

Novell NetWare 3.12 is a network operating system that allows many different clients, such as PCs running DOS, Windows or OS/2 as well as Macintosh and Unix systems, to connect to a NetWare file server. Packets transmitted over the NetWare network normally use Novell's own Protocol SPX/IPX. Since our target system in the conducted experiment consisted entirely of PCs, the following discussion is simplified by focusing only on such systems.

The clients and their server interact according to the general principles shown in Figure 1. Clients in the target system were ordinary PCs running DOS and Windows 3.1, although two additional software packages, NETX and IPX, were needed to communicate with the NetWare server (also PC-based). The NETX package redirects application system calls either to the local operating system or to the server accessible through IPX and the network interface card (NIC).

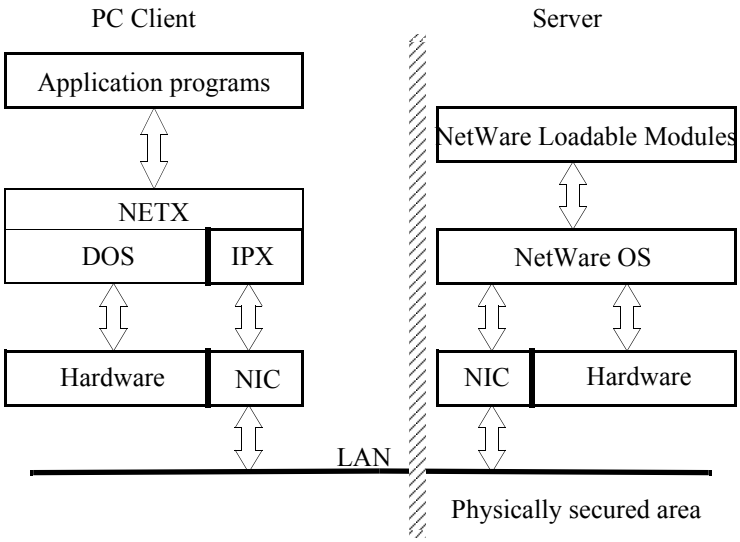

Figure 1: Relationship between clients and server in NetWare 3.12.

The server runs the network operating system, NetWare OS. NetWare OS can be configured and extended with various NetWare Loadable Modules, NLMs. Examples of NLMs are device drivers for network cards and hard disks, but can also be various menu-driven application programs such as commands and programs available at the server console. When NetWare OS is running, it is not possible to start DOS sessions.

NetWare 3.12 offers many security features, although none of the high security features are activated by default, thus some default values must be changed by the system administrator to attain an adequate degree of security. The security features can be divided into three different groups: *login security*, *bindery security* and *file system security*.

The *login security* process decides whether a user at a PC client should have access to resources on the server. The login process offers two security features: identification and authentication. Identification is established by sending a unique user identifier to the server, which sends back an account identifier and a unique encryption key. The client then encrypts the user password and the account identifier with the key and sends it to the server [Stang and Moon 1993]. The encryption algorithm itself is not publicly known. If the password and the user identifier match the corresponding pair in the bindery database, the user is authenticated to use the system.

The second mechanism is the *bindery security.* The bindery is the NetWare OS system database and is stored on the server. In the bindery, the supervisor has the ability to define privilege relations between different objects such as users, groups and print queues.

The third security mechanism deals with *file system security*. The file system security is divided in two groups: *Rights security* and *Attribute security*. *Rights security* is obtained with Access Control Lists, ACLs, where rights to a file or directory can be assigned to a user or a group. *Attribute security* offers users the possibility to assign attributes to files or directories, such as “not backed up”, “hidden”, “system file” etc.

3. THE INTRUSION EXPERIMENT

The experiment was conducted over a four week period in November/December 1994. The target system was a "standard" Novell NetWare 3.12 system with eight Intel 486 PCs connected to a file server. All PC clients had DOS 6.2 and Windows 3.1 installed. The server was secured in a well-protected room, to which only the system administrator had access. Intruder detection and lockout features in NetWare were enabled, and the number of login attempts was restricted to 10. Network (NCP) packet signatures were not used, nor did the hardware support hardware-based passwords. Since we wished to test a "realistic" system, no special security-enhancing modifications were made to the system. Besides, this system was intended to handle all administrative work required by a small organization, and the system owners had configured the system to be, what they believed, secure.

3.1 The Actors

There are three different actors involved in this kind of experimentation: the Attackers, the System Administrator and the Coordinator:

The selected *attackers* were last year university students who conducted this experiment as part of an undergraduate course in Applied Computer Security. There were 14 attackers divided into eight groups (six groups with two persons, two groups with one person). We expected the attackers to spend about 40 hours each of effective working time during a four-week calendar period, although there was no absolute requirement to spend exactly this amount of time. The major motivation of the attackers was that the experiment was a compulsory part of the course they were taking and that well performed work (i.e. a good security analysis of the system) could result in higher marks in the course. To keep attackers motivated, they did not necessarily have to conduct many security breaches; a well-planned approach together with serious *attempts* would be considered equally good.

Each group was given an account with ordinary user privileges. However, we knew there were certain limitations that would make it fairly difficult for our attackers to be able to break into the system. First, they had physical access to the target system only two evenings a week during a three-week period, even though they had full access to Internet through Unix workstations continuously during the experiment. Second, except for one attacker, no one had ever seen a Novell NetWare system before, although most had worked with PCs[1]. Third, the system was not yet in production use, meaning that there were no real users using the system during this time. Despite these rather serious limitations, the attackers did find surprisingly many security problems with the target system.

The *coordinator's* role was to monitor and coordinate all activities during the experiment. In particular, he tried to make sure that the attackers and the system administrator complied with the experimental rules (see below). He was also to make sure that the activity of two different attack groups would not interfere with each other.

The *system administrator* was to behave as realistically as possible. He performed his task rather modestly and only elaborated with the system when it did not behave normally.

3.2 Rules for the Attackers

The attackers were told (rather informally) that *a security breach occurs whenever they succeed in doing something they are not normally allowed to do*, for example reading or modifying a protected file, using another user's account or disturbing the normal system function. The attackers had also to obey some restrictions: They were not allowed to physically damage the equipment. They were instructed not to cooperate with any other attacking groups.

---

1 All group members were Unix users. Four groups reported they had "none" to "some" experience with Dos and Windows. The other four groups reported their knowledge to be "fair" to "good".

3.3 Reporting

There were three principal reports: the *background report*, the *activity report* and the *evaluation report*. At the end of the experiment, each group had to submit a *written final report*.

The attackers were asked to fill in a *background report* which was submitted before the experiment started. The attackers were asked to document their background, knowledge in computer security, prior experience with NetWare and DOS etc. It was especially important to find out whether any attackers had any prior knowledge of vulnerabilities in NetWare before starting the experiment.

In the *activity report,* the attackers were asked to document and describe every specific activity. They were to document how long they worked on each activity and what resources they had used, such as literature, personal help, Internet or BBSs. They were also asked to record their gain or loss for every activity they carried out.

At the end of the experiment, the attackers filled in another questionnaire, the *evaluation report*. Here, the attackers were asked to estimate and document their opinions on the information flow between the groups and to give general comments about the experiment.

In the *written final report,* the attackers summed up all activities and described the results freely. They were supposed to describe their main leads and to explain what vulnerabilities they had found during the experiment. They also described their intrusions and all intrusion attempts in detail.

4. RESULTS OF THE EXPERIMENT

4.1 General Results

The most striking result of the experiment is that all groups succeeded in finding at least one way to break into the system, even groups with little or no experience of DOS and PCs. The best information and most ideas and hints were found in Internet sources such as WWW and News. Most programs were retrieved via ftp on Internet, and two groups had used various underground BBSs to collect both information and programs. As a result of the search, many publicly available programs were found and tested.

4.2 Some Numerical Results

During the experiment, we received 80 activity reports describing a total of ten security breaches of various kinds. In total, the groups reported 318 hours of working time, i.e. time during which at least one group member was active. Altogether 424 man-hours were spent, which is an average of 30 man-hours per attacker. It is important not to forget that these numbers include time learning the system. Of these ten breaches, five were not actually completely carried through but were counted as probable breaches, since they would have resulted in a full breach if other users had been present in the system (see below). In addition, there were three attacks that would have succeeded if the file server had not been physically secured with alarms, as in this installation. However, equally important are the many failed attacks since they quite often describe breaches that would have succeeded if the attackers had been given more time. All vulnerabilities found, including breaches, are discussed in the next section.

4.3 Attacks and Breaches

4.3.1 Classification

A classification of all attacks performed is found in table 1. The table shows a summary of successful breaches, probable breaches (attempts that would lead to a breach in a more realistic environment) and unsuccessful attempts per category. The categories are defined as follows:

- *Use of System Administration Programs*: Programs that are used by the system administrator to administer the system.
- *Network Attacks*: Reading or modifying packets transmitted over the network.
- *Attacks against the Login Process*: Password guessing or replacing the NetWare login command with a Trojan Horse.
- *Keyboard Snooping*: Listening to the keyboard through TSR (Terminate and Stay Resident) programs.
- *File server Attacks*: Attacks against the file server, provided that the server is physically available to the attacker.

Table 1: Number of breaches per category

Category	Unsuccessful Attempts	Probable Breaches	Successful Breaches	Total no of Attempts
Use of System Administration Programs	4	-	-	4
Keyboard Snooping	4	3	2	9
Network Attacks	5	-	1	6
Attacks against the Login Process	3	2	2[a]	7
File server Attacks	-	3[b]	-	3
Total	16	5 + 3	5	29

a. One of these breaches may not be considered as a serious breach. The system administrator had not initially assigned passwords to the attackers' accounts, and one group simply tried and managed to login to unused accounts without passwords. See section 4.3.5.

b. One of the groups were given physical access to the server. They had found Novell's notorious back-door to the system, and the intrusion attempt was aborted by the coordinator when they had demonstrated that it could be carried out. Two other groups had found similar information, but were not given physical access to the server. See section 4.3.6.

4.3.2 Use of System Administration Programs

It is generally not possible to read system or administration-related information in NetWare as an unprivileged user. The possibilities for using system administration programs in NetWare is also limited and supervisor privileges are normally required. There exists, on the other hand, administration programs that can be installed on clients and may be used to circumvent some security features in NetWare.

One such program that was found and tested is *spy.com*. The program is publicly available and is intended to help system administrators to remotely control their clients. It can, for example, be used to send the contents of the screen to a remote program. To use this program, a TSR program is installed on the client and another program on the controlling computer. Attacks using *spy.com* were not successful because the NETBIOS protocol was required and this was not available at the target system. However, sites running NetWare where NETBIOS is installed should be vulnerable to these attacks. Also, note that an installation of NETBIOS is a relatively simple operation made in less than five minutes by someone with a little more Windows experience than our attackers had.

4.3.3 Keyboard Snooping

Keyboard snooping was accomplished by logging keyboard strokes with a TSR program. A TSR program is normally installed either when the client boots or when a commonly used program is executed. When started, the TSR program stays resident in the memory and performs its task; in this case, it normally sleeps and is activated at each keyboard interrupt.

Several attackers used TSR programs to log keystrokes to a log file. In some cases, the log file was encrypted with a simple algorithm to hide its contents. Several groups found and tested already available programs, for example *keytrap.com*, *log.com* and *keycopy.com,* whereas other groups wrote their own programs. Five groups installed a fully functional program, but because of the lack of ordinary users, only two groups succeeded in this activity. Also, many attempts were unsuccessful because the attackers did not fully understand the programs they had found or because of bugs in their own software.

4.3.4 Network Attacks

The target system used a network setup on which all messages, except passwords, were sent in plaintext. Thus, it was possible to monitor the contents of the transmitted packets. Some groups tried to use network snooping programs. However, while the passwords were encrypted and the data given by the snooping programs was not very useful (mostly because of lack of ordinary users but also because of very complicated user interfaces), none of the groups managed to obtain anything useful from the network.

Yet another attack method discovered was the possibility of taking over a client's communication with the file server. One such program is a very well-known Dutch program, called *hack.exe* or *nethack.exe.* This program was developed in 1992 at Leiden University in the Netherlands. It exploits a deficiency in the NetWare Core Protocol, NCP. All system calls for network resources in NetWare are constructed as NCP packets, and one NCP packet holds information about which user has requested a service together with the user's authentication ticket. The program extracts one NCP packet from the network, and replaces the client network address with a well-selected address that can be arbitrarily selected. It then constructs 256 packets with different sequence numbers (NCP packet sequence numbers are 1 byte long) and bombards the server with them, knowing that one packet will be accepted, and thereby taking over the original user's connection [Stang and Moon 1993]. The program *hack.exe* exploits this deficiency by automatically taking over the supervisor's communication. Novell claims to have solved this problem when they introduced NCP packet signatures, where each NCP packet has an encrypted signature that changes with each packet.

Several groups found and tested *hack.exe*, although it was necessary for the supervisor to be logged-on to the system in order for his communication to be taken over. Furthermore, the program looked for the string "supervisor" in network packets while, in the target system, the supervisor used a supervisor-equivalent account name, "root". The program would have had to be changed accordingly in order to work, but this was realized by only one group that managed to get *hack.exe* to work as proposed.

4.3.5 Attacks against the Login Process

One obvious way to attack the login system is to guess passwords. One group succeeded in gaining access to some accounts, but this was a result of the fact that no passwords were assigned initially when the experiment started. There are also some public domain password-guessing programs available that were used by the attackers, for example *netcrack* and *novelbfh.* These programs require passwords to be transmitted over the network in an unencrypted form. This was however not the case in the target system. Consequently, these attempts were not successful.

The password encryption algorithm used by NetWare is not publicly known. Several sources claim that some passwords are encrypted with badly selected encryption keys, where the result is virtually independent of the password, making such login-sessions easy to fake. A program called *knock.exe* that exploits this deficiency is said to exist (not verified by us). This program should, when executed, login to the supervisor's account and erase the password and, consequently, anyone could be a supervisor after this program has been executed.

The other type of attack against the login process utilized the insecure operating system on the client by means of inserting a Trojan Horse. Anybody who gains physical access to a client can replace the NetWare login command with a copy that also records the login parameters. A couple of groups tested this approach and one group succeeded in obtaining passwords this way. Trojan Horses of this type were all written by the attackers themselves.

4.3.6 File server Attacks

It is important that the file server is physically secured. If the file server is physically available, it is possible to boot DOS and directly modify arbitrary system files. One group was, for test purposes, given physical access to the server. This group tried to modify server data according to step-by-step instructions found in a BBS. They were not allowed to continue when it became obvious that they had found Novell's notorious back-door. Novell's back-door allows anyone who can gain physical access to the server to become the supervisor of the system (the idea is to make NetWare believe it is freshly installed, in which case it allows the supervisor to log in without a password). Detailed instructions for how to use this back-door are available on the Internet, and three groups found those instructions and accomplished "probable breaches". One group even found a program called *setpass*, which seemed to be an automatic program for this task, and executing it on the server would allow anyone to use the supervisor account.

The attackers also had several other ideas about how the file server could be attacked; for example, one source claimed that filling the print partition with data should make the file server hang, although these ideas were never tested due to a lack of time.

5. DISCUSSION

The approach of letting *ordinary* users attack a system has proven to be a valuable tool for assessing its security level [Olovsson et al. 1995]. In this case, we allowed a group of university students to attack a newly installed system that was not yet in production use. As a result, the system owners have

become fully aware of the limitations of their system's security features and of how much effort is required to break into their system. This information is invaluable when making decisions about how the system can be used and to what extent it can be trusted.

The attackers approached their task in different ways. The most successful *method*, all categories, was to attack the clients, i.e. to monitor users and install malicious software such as Trojan Horses. The network was vulnerable, too, since NetWare sends all data except passwords in plain-text. This is a serious weakness even though no attack team managed to get any useful information directly by listening to the network. Again, given enough time, this approach would without doubt have proven to be successful.

It is also worth noting that the server was virtually untouched during this experiment. Undoubtedly, Novell has spent more effort in securing the file server and its assets than in securing the clients in the system. However, as long as the clients are as vulnerable as they turned out to be, attackers did not have to spend their time attacking the server.

5.1 Attackers and the Intrusion Process

Despite the facts that 13 of the 14 attackers had *no experience with Novell NetWare* prior to the experiment at all, and that they only had four weeks to spend learning and using (in parallel with attending three courses), it is disappointing from the system owner's point of view that all groups managed to find different ways to break into the system. On average, they spent only 30 man-hours each, including the initial learning period required for a completely new system.

By studying the intrusion process, it is also possible to draw some conclusions about *how* they behaved and *where* they managed to get their information. Internet seems to be an inexhaustible source of information, both technical information about the system and information about possible weaknesses in the system. Two groups found various more or less obscure BBSs containing programs (*hack.exe*, *netcrack*, etc), several of them developed in the Netherlands. In some cases, books, manuals and journals were used. Some of the attackers also used "insiders" in different companies or friends to obtain information about possible ways to breach the system. However, all breaches were "known" on the Internet. It is also worth noting that Novell's own documentation about security features in NetWare version 4 contains information on security enhancements made from version 3.12, which of course gave rise to ideas about what parts of the system had turned out to be especially vulnerable[1].

It is noteworthy that many attackers, after having accomplished a successful breach, started to improve the intrusion method instead of searching for new ways to break into the system. This was done in spite of the fact that the objective was to make as many breaches as possible. A similar observation is that many groups tried to write their own software, even though they knew similar software was available elsewhere. It may seem a waste of time, but they probably felt a high reward from successful breaches performed without external help. Also, they spent considerable time on trying to understand why attacks worked or did not work, even when they had other ideas waiting for how to penetrate the system.

5.2 Possible Security Improvements

Most detected problems exist due to the fact that local clients are insecure. The problem of insecure clients is also addressed in NetWare 4.x, although Novell unfortunately does not say in which way and how well they have solved the problem. One improvement would be to use a login procedure that supports *one-time passwords,* such as external devices or smart cards. The problem of recording and

1 Even though version 4 is released, NetWare versions 3.11 and 3.12 constitute the largest installed base of NetWare systems today.

replaying passwords could then be avoided. Another improvement would be to *boot the client via the network* [Lomas and Christianson1995] or to boot from a write-protected or encrypted hard-disk. Still another possibility suggested by Novell is to *physically secure the clients* or to use hardware-based (e.g. PROM-based) passwords.

The second most vulnerable part of the system is the network, since all network traffic except passwords are transmitted in plain-text. It is therefore obvious that security could be strengthened by encrypting the traffic. This is actually possible with NetWare, even though third party solutions are needed since Novell will not provide encryption software or hardware.

6. CONCLUSIONS

Before the experiment started, we had anticipated that the attackers would create Trojan Horses on the clients. However, we did not expect the system to be as vulnerable as it turned out to be. Two general conclusions can be drawn from this experiment. First, despite the fact that the system claimed to have security mechanisms of a rather high level, it was easily and successfully attacked and breached by novice users, who exploited the insecurity of the clients and the network. The conclusion is that security must be distributed evenly over the system. The experiment pinpoints that *an attacker attacks where the weaknesses are*, not where security enhancements have been made. The practical implication of this is that, given a specific system, the security of it is not necessarily improved by adding some heavily marketed “security packages” that will strengthen a certain part of it. Obviously, it is the weakest part that determines the security level.

Second, the experiment shows the importance of Internet access for information gathering. Even if all information is available through alternative sources, the use of Internet means that an enormous amount of, often dedicated, information is available very easily and rapidly. It is often this fact that reduces the effort required for a successful intrusion. This makes all the difference from a security point of view.

7. REFERENCES

[Attanasio 1976] **C. R. Attanasio. P. Markstein and R. J. Phillips:** *Penetrating an Operating System: A Study of VM/370 Integrity*, IBM Systems J., 15 (1), pp. l02-116, 1976.

[Brocklehurst et al. 1994] **S. Brocklehurst, B. Littlewood, T. Olovsson, E. Jonsson:** *On measurement of Operational Security*, pp. 257-266 in Proceedings of the Ninth Annual IEEE Conference on Computer Assurance, COMPASS '94, Gaithersburg, Maryland, USA, June 29-July 1, 1994.

[Goldis 1989] **P. D. Goldis**: *Questions and Answers about Tiger Teams*, EDPACS, The EDP Audit, Control and Security Newsletter, October 1989, Vol XVII, No. 4.

[Gustafson et al. 1996] **U. Gustafson, E. Jonsson, T. Olovsson:** *On the Modelling of Preventive Security Based on a PC Network Intrusion Experiment*, To be presented at the Australasian Conference on Information Security and Privacy, 24-26 june 1996, Wollongong, Australia.

[Herschberg 1988] **I. S. Herschberg:** *Make the Tigers Hunt for You,* Computers & Security, 7 (1988), pp 197-203, Elsevier Science Publishers Ltd.

[ITSEC1991] *Information Technology Security Evaluation Criteria (ITSEC),* Provisional Harmonized Criteria, December 1993. ISBN 92-826-7024-4.

[Jonsson and Olovsson 1996] **E. Jonsson and T. Olovsson:** *An Empirical Model of the Security Intrusion Process,* To appear in the proceedings of COMPASS '96, 11th Annual Conference on Computer Assurance, 17-21 June 1996, Gaithersburg, Maryland, USA.

[Littlewood et al 1994] **B. Littlewood, S. Brocklehurst, N.E. Fenton, P. Mellor, S. Page, D. Wright, J.E. Dobson, J.A. McDermid and D. Gollmann:** *Towards operational measures of computer security,* Journal of Computer Security, vol. 2, no. 3.

[Lomas and Christianson1995] **M. Lomas, B. Christianson:** *Remote Booting in a Hostile World: To whom am I Speaking?,* Computer, January 1995, pp. 50-54.

[Olovsson et al. 1995] **T. Olovsson, E. Jonsson, S. Brocklehurst, B. Littlewood:** *Towards Operational Measures of Computer Security: Experimentation and Modelling*, in B. Randell et al. (editors.): *Predictably Dependable Computing Systems*, ESPRIT Basic Research Series, Springer Verlag, 1995, ISBN 3-540-59334-9, pp 555-572.

[Stang and Moon 1993] **D. J. Stang and S. Moon:** *Network Security Secrets,* IDG Books Worldwide, Inc. ISBN 1-56884-021-7, 1993.

[TCSEC 1985] *Trusted Computer System Evaluation Criteria* ("orange book"), National Computer Security Center, Department of Defense, No DOD 5200.28.STD, 1985.

BIOGRAPHIES

Ulf GUSTAFSON, Ph. D. Student, M. Sc. E.E.

Ulf Gustafson received his M. Sc in Electrical Engineering 1994 from Chalmers University of Technology at Gothenburg, Sweden. He has also been working as a systems design engineer at Ericsson Microwave Systems AB for one year. He is currently a graduate student with research interest in computer security in distributed computer systems.

Erland JONSSON, Associate Professor, Ph. D.

Erland Jonsson is working as a teacher and researcher at Chalmers University of Technology, Gothenburg, Sweden. His major research interest is computer security and in particular issues regarding the quantitative assessment of security. In addition to his academic merits he has an extensive industrial experience. He is the author of one textbook and a great number of technical papers and publications.

Tomas OLOVSSON, Assistant Professor, Ph. D.

Tomas Olovsson received the M.S and Ph.D. degrees in computer engineering at Chalmers University of Technology, Gothenburg, Sweden. He is currently working both as Assistant Professor at Chalmers University of Technology and as a computer consultant with special interest in computer security issues. His current research areas are computer security and in particular assessment of operational security.