Experiences of Internet security in Italy.

Alessandro Berni, Paolo Franchi, Joy Marino
Department of Telecommunications, Computer and Systems Science
University of Genova
Via Opera Pia, 11a
16145 Genova, Italy

E-mail:
{ab,pan,joy}@dist.unige.it

Abstract

While the diffusion of the open systems culture is nowadays marking the Italian networking panorama, a growing number of issues regarding Internet security need to be addressed to ensure the proper balance between connectivity and safety.

This paper illustrates two cases occurred in the past and gives the picture of the current situation.

1 Background

The first Internet connection between the United States and Italy was established in 1986 using the now dismissed SATnet, a medium speed satellite network connecting Europe to the ARPANET core: the satellite link, sponsored by DARPA, by the Italian Ministry of Defense and by the Italian National Research Council (CNR) was shared with two other sites, one in England and one in Norway, while the site selected to host the connection was the CNUCE Institute of the CNR, in Pisa.

Till then academic networking was not widely spread and the existing connections were based on proprietary protocols (mainly VMS/DECNET in the High Energy Physics community and VM/RSCS for EARN/BITNET sites): there were however some *TCP/IP islands*, that is, institutions that adopted TCP/IP for in-house networking. The obvious step for them was to reach somehow Pisa and then make use of the international link. In this framework the University of Genova was the first italian university to obtain the Internet Connected Status, shortly followed by the University of Bologna.

The success of these experiments led the Ministry of University and Scientific Research to sponsor the support of the IP stack in the GARR network, whose aim was to provide a high speed multiprotocol backbone that could serve all the government-funded research institutions.

The GARR network consists of E1 (2 Mbit/s) lines connecting the main hubs situated in 6 italian cities: regional networks obtain access to GARR connecting to one of the hubs. The following table [1]shows the evolution of the number of TCP/IP hosts in Italy since the GARR network was started. [2]

[1] Courtesy of Marten Terpstra, RIPE.

[2] It's out of discussion that the growth of the world Internet has strong relations with the diffusion of UNIX systems: the smaller *density* of TCP/IP hosts in Italy (compared to other european countries), in spite of a modern network infrastructure, is sign of a prevalence of the culture of proprietary environments against open systems.

Month	Year	IP hosts
October	1990	526
November	1990	649
December	1990	640
January	1991	929
March	1991	1031
April	1991	1198
June	1991	1298
August	1991	1495
September	1991	1822
October	1991	2094
November	1991	2413
February	1992	3289
April	1992	3657

A side effect of the extension of the network to different institutions, only a few of which had experience with TCP/IP networking, has been a continuing growth of unauthorized activities. In the following sections we will analyse two past cases that served to increase the sensibility to security issues among italian networkers.

2 Two case studies.

2.1 Case #1, December 1990.

On Dec. 3, 1990, at about 10 am EST, a site in the US noticed from the *syslog* files that a host on network 131.175.0.0 was trying to issue sendmail DEBUG and WIZ commands on TCP port #25 (SMTP). Later on the same day, at around 8 pm, another site noticed unauthorized activity from a different machine on the same network, consisting of

- attempts to use *tftp* to obtain system sensitive files (e.g. */etc/passwd*)
- attempts to exploit the VAX *fingerd* bug in the same way as the Internet worm
- attempts to use an old *ftpd* bug to obtain privileged access to the system.

Both sites informed the CERT/CC, that took the proper steps to get in touch with the administrator of the offending system that, according to the DDN NIC *whois* database, belonged to the University of Milan.

University officials were immediately contacted by e-mail and a carbon copy of the message was sent to us in Genova in case we could be of help in contacting them.

This contact proved to be quite difficult: in fact we were not able to speak to the administrators on the machine originating the intrusion attempts, but only to the network manager. As a matter of fact, nobody seemed to know precisely to whom the offending machine belonged.

In the same time the attacks were resumed and directed to hundreds of Internet hosts, using the same techniques described before: considering the seriousness of what was happening we started considering to filter all packets from/to network 131.175 crossing our routers, thus isolating the whole University of Milan from the rest of the world.

We subsequently decided to keep this option as an extreme alternative, while we continued to try to get in touch with the administrators of the offending site.

The following day the postmaster on our mail machine noticed about 2000 error messages directed to the same user at the University of Milan:

```
To the Postmaster:

We are a group of researchers and students of the
state university of Milano (Italy), computer science
Dept.  that are working on security.
```

```
for what we know, there is a common bug in some ARPA
services or their installations.  We are now trying to
identify hosts that may be subject to this bug, in order
to inform them as soon as we finish collecting the data.

Nobody is going to try to intrude these systems.

for any further explanation, please send mail to
miners@host.unimi.it, and we will reply as soon as
possible.
```

The situation was now much clearer: we used our *syslog* data to derive the addresses of the original recipients of that message. This list proved to be a subset of the HOSTS.TXT file as found on *nic.ddn.mil*: according to the list, the intruders had tried to exploit UNIX security problems connecting to non-UNIX hosts, such as MILNET TACs (that for sure do not accept SMTP connections: that's why many of the e-mail messages had been rejected).

The list, containing all the attacked sites, was sent to the CERT/CC, that distributed a Cert Advisory CA:90-11 about the *Miners* probes.

In the next days it was finally possible to get in touch with the administrators of *host.unimi.it* [3](restricting the search to the CS department): they had not realized what was happening until they received a number of complaints from US sites that reported unauthorized activity originating from their machine.

They proved to be very cooperative in suspending all the *Miners* probes and making their software available to both the GARR and CERT/CC for analysis, but all this happened a week after the first attempts had started: in different situations this could have been simply too late.

2.2 Case # 2, August 1991.

In August 1991 we noticed a series of connections for user *user1* originating from the computer that hosts the primary name server for domain .IT: a quick check showed that *user1* had been out of town for the whole week, with little chance for him to put his hands on a keyboard.

We immediately changed his shell to */usr/misc/freeze* [4]and informed the administrator of the other machine: he reported that he had noticed something *going wrong* in the last weeks and had already changed the *root* password, checking also for suspect *setuid* programs (without finding anything relevant).

In the next days we recorded attempts to log in as *user1* on several machines within our LAN, originating from a terminal server at the MIT or, in alternative, the .IT server host. A *finger* to this machine while a connection attempt was being made showed as sole user logged on *gallina* [5] (from an MIT terminal server): as soon as someone logged on to see what was happening *gallina* disappeared (and that's not surprising since nobody of the administrators had heard of that user before).

What was really surprising was to find out that *gallina*'s entry in */etc/passwd* had *uid* zero, granting him (her?) superuser access at any time.

There weren't doubts left that the intruder had found his way to become superuser any time he wanted: a deeper analysis of the file system (comparing the size and date of all binaries with those found on the OS distribution media) showed that */usr/ucb/telnet* had practically doubled its size.

This newer version of *telnet* did not contain strings that could help in finding out what its *enhancements* were: however, using the standard *trace* command we were able to notice that at the beginning of the remote session a file in */usr/spool/mqueue* was being opened with filename *AA(pid)* (where *(pid)* was the current process id). In this way a a casual observer would not have

[3] The actual names of users and machines have been changed since they are unnecessary in this context.

[4] *freeze* prints a message at *login* informing the user that his account has been suspended and that he is invited to call the system manager for further explainations.

[5] by the way, that's the italian word for *hen* (!)

immediately noticed what was going on: the naming convention adopted by the patched *telnet* was in fact similar to that of *sendmail* while handling temporary files.

It goes without saying that this patched version of *telnet* recorded every single keystroke the unaware user was entering.

The following is an example of how the *troyan* telnet recorded the sessions:

```
write (1, "Trying...\n", 10) = Trying...
(...)
write (1, "Connected to nic.ddn.mil.\n", 26) = Connected to nic.ddn.mil.
(...)
open ("/var/spool/mqueue/AA19365", 01001, 0666) = 5
(...)
write (5, "Connected to nic.ddn.mil.\n", 26) = 26
close (5) = 0
write (1, "Escape character is '^]'.\n", 26) = Escape character is '^]'.
```

When we browsed through the files in */usr/spool/mqueue* we were able to find 64 "recordings" made by *telnet*: in particular we found passwords for the great majority of the GARR routers as well as computers both in Italy and abroad.

Of course there was also a recording of a session made once by *user1* on one of our systems (and that explains the intrusion we recorded): it's not clear however why the intruder did not care to delete those files after having examined them (he had for sure the occasion to do so).

From an old *wtmp* file we were able to determine that the intruder(s) had first succeeded in obtaining access to that machine three months earlier: a high number of *ftp* logins concentrated in a very short amount of time suggested that the security hole could be related to the anonymous ftp server active of the same host. In this the network proved to be a valuable source of information (pointing to a well known *ftp* bug that enabled the guest user to obtain unauthorized privileges). After having rebuilt the system from past backups a newer version of *ftpd* was obtained and the problem solved.

3 Our test.

In this last year we have recorded a number of unauthorized logins on our systems: no harm was done (only once the intruder used our computing resources to run a password cracking program) and all the cases were solved within a day or two, simply monitoring every access from the outside.

Instead then trying to solve our problem using brute force (like installing a firewall system) we decided to take measures of passive defense, like putting a wrapper around our network daemons or improving the consciousness of users regarding to password security.

The question we posed ourselves was to find out how difficult it is to break into a totally foreign system: of course we wanted to avoid doing anything illegal, but also had the awareness that potential intruders never make a great quantity of considerations regarding ethics. We also felt that our experience could help other sites that were (or still are) victims of those intrusions to build a safer networking environment, without having to abandon the openness that should mark academic institutions.

Our decision was of adopting a soft approach, avoiding as much as possible the use of special purpose programs and using only standard UNIX commands or simple shells scripts. As a prerequisite to our tests we needed to be *root* on at least one system [6]

We did not consider this condition as too restrictive for the following reason: in a rather loose environment (as we find in many academic institutions) it's not impossible to gain privileged access on one or more computers (maybe that workstation that arrived last monday and that is not being used extensively); in addition to that, with the diffusion of small UNIX boxes (e.g. 386 with 4.3 BSD or low cost workstations) the number of systems that need to be administered is

[6] The idea of this test dates back to February 1992, while its realization has been completed in May.

increasing every day. It's not uncommon to have the administration of a system delegated to a researcher or a student that is not experienced with network security.

What we did [7]was to obtain a listing of all the italian IP hosts making recursive queries to the DNS: from the total of more than 3000 hosts we were able to discriminate and discard all routers, PCs and Macintoshes (that usually don't accept connections from the outside). The next step was to use the HINFO resource record to obtain particular host information, like CPU type and OS version: we decided to concentrate our search on computers built by one single manufacturer, in consideration of the even distribution over the country of these workstations and servers.

The list, consisting of 220 hosts (about 7% of the grand total), was passed to a simple shell script that attempted to obtain the */etc/passwd* file using remote commands: while the data was being gathered, at periodic intervals, the file with the results was transferred with *uucp* over a *hidden* serial line to a secure machine kept isolated from our Ethernet. This to avoid to leave this sensitive information on a machine shared with other users and thus potentially insecure.

At the end of our test 48 machines had not refused us access (21.8% of our list, about 1.5% out of the grand total as of Febrary 1992)

We joined all the password files in a single one, over 1500 lines long, and fed it into the *Crack* program running on a machine that was kept off the network for the occasion. The dictionary we used consisted of 60453 italian words (and was a *souvenir* of a previous visit of an intruder to our systems).

The results of *Crack* after a 14 days run on a dedicated Sparcstation ELC were as follows:

		Percentage on total
Locked	414	27.6
Users with NO password	180	12
Users with "easy" password	44	2.9
Users with simple vocab. password	40	2.6
Users with vocab. password + number	4	0.25
Success ratio		17.6

4 Lessons learned.

In spite of the abundant literature related to security available through the Internet, we have noticed that it is rather easy to obtain unauthorized access to a foreign system making use of well known *problems*, consisting in most cases in improper configurations.

Before informing the sites we succeeded in attacking of their security problems, we made a further test, in order to determine how security information is dealt with in different sites.

Starting from a security checklist received from the CERT/CC, we prepared an information file, in both italian and english, giving information on how to solve "some common security problems", that is, giving precise directions on how to fix the holes that had granted us access.

The file has been sent by mail to both the *root* and the *postmaster* account of "our" 48 systems: 24 of them received the italian version, while the remaining half received the english one.

We allowed them 15 days for taking the proper steps to correct their problems and then tried again our script: our aim was to see how many sites had acted to patch their situations and whether the reception of the information file in the native language had brought to an increased degree of responsiveness.

The result is summarized by the following table:

	Italian	English
No action done	15	18
Holes fixed	7	4
Unreachable	2	6

[7] Of course before starting the test we informed the GARR/IP administrators that a security test would be taking place in the following days.

It seems that the reception of the information file in the native language has determined a slightly higher grade of awareness regarding security: however, the narrowness of the sample and the persistent unreachability of some hosts involved in the test do not allow us to bring a definite conclusion to the question.

5 Conclusions.

From our experience we can say that obtaining and maintaining a reasonable level of network security is fairly simple: what is not offered by the standard operating systems distributions can be easily obtained over the network.

Packages like Wietse Wenema's TCP Wrapper offer powerful instruments for detecting and keeping off potential intruders: other programs, like *Crack* could both solve and cause security problems. Up to the release of *Crack*, one of the few password crackers available was Dave Curry's *NSA*: this program was not available to the general public, but only to the *root* user of Internet hosts registered in the DDN NIC *whois* database. This minimal level of control, helped to prevent the wide circulation of the program.

The problem with the wide availability of *Crack* is that while not all system administrators would want to use it, every cracker would surely do so: to achieve the parity in this game, every system administrator should check periodically the password file of every machine under his control. This can prove to be very expensive in terms of both CPU and manpower.

The best solution so far is that of installing a shadow password file, in order to vanify the intruder attempts.

It goes without saying that prevention is extremely important in this field: the establishment of the CERT/CC has made possible the existence of an important *culture repository* related to Internet security: the present (centralized) structure of this organization make it most effective in disseminating information and *responding* to user needs, while the preemptive and direct action on the end user results extremely difficult.

For this reason we feel that the establishment of CERTs among all the different communities that make up the world Internet would bring to a more direct way of *making order* in the network: this is precisely the framework that constitutes the Forum of Incident and Response Teams (FIRST). The european networking organization RARE is already considering the establishment of its own response team, and so is doing the pan-european network EUnet. At a local level, response teams are being built in some european countries, for example in The Netherlands: for what pertains to Italy we are ready to offer our experience to help to create a similar structure for the benefit of all networkers.

Acknowledgements.

The authors would like to acknowledge the staff at the IUnet Network Operations Center in Genova and CNUCE Institute of the CNR for their precious collaboration and helpful discussions.

APPENDIX: Information file (English version).

Dear system administrator,

please review the following information from CERT. Further information about CERT can be obtained with anonymous ftp from ftp.iunet.it or directly from cert.org.

1) Compare /bin/login against a known good version. If none is available, you can try running "strings /bin/login" and look for a likely looking password name and try logging in as any user with that password. So far, the trojan passwords have shown up this way, but that could easily change.

2) Check other machines on your local networks for signs of intrusion. Any site you share yellow pages (NIS), NFS or /etc/hosts.equiv with is at risk. Any sites your users share .rhosts access with is at risk.

3) Check your /usr/ucb/telnet and su programs.They may be trojans collecting legitimate user sessions (with machine names, accounts, passwords). In one case these transcripts were kept in a /usr/spool/lpd/.lpd directory.
(Again, the "strings" program may identify the directory being used.)
If you find this, then other machines in the transcripts are at risk. Since the files would likely have been cleaned out periodically, other machines your users regularly access could be at risk.

4) Check mount and umount to be sure they aren't set-uid to root. In general, check for other setuid programs via

```
     find / -perm -4000 -print
```

5) Check /etc/hosts.equiv and all users' .rhosts files for inappropriate "+" entries or non-local machines (especially ~root, ~uucp, and other system accounts).

6) Check /usr/share/lib/me for a subdirectory called "..." which has been used as a home base for the intruder on some systems. Actually, "..." has been a favorite directory name and it would be worth a

```
 find / -name ...  -print
```

to look for others.

7) Install wrappers on your tcp services to log connection attempts. Source code for a package which does this is available via anonymous ftp from cert.org in the pub/network_tools directory.

8) Check /etc/inetd.conf for new entries which provide services you do not wish to offer. Especially look for new "services" which are not familiar to you.

9) Make sure all system passwords (especially uucp) are set to reasonably hard-to-guess strings.