The perspectives of the Amateur University Networks – AMUNETs (Part 2)

MIROSLAV SKORIC
Faculty of Technical Sciences
University of Novi Sad
Skoric, P. O. Box 52, 21102 Novi Sad
SERBIA
skoric@uns.ns.ac.yu http://tldp.org/HOWTO/FBB.html

Abstract: - This paper discusses results of the recent experiments, which intention was to test security in accessing e-mail servers and radio relay systems within the amateur radio digital infrastructure. There are many opportunities in available safety measures, which support the integrity of user's and system administrator's credentials. This work suggests various methods which aim is to bridge the gap between the improved safety and comfort in regular end user's and system administrator's activities within an amateur radio computer network.

Key-Words: - amateur radio, AMUNET, communication, computer, security, privacy, university, education

1 Introduction

One of the frequently asked questions after the amateur radio presentations at technical conferences and similar events – is how to ensure the safe access to end-user email accounts – using the amateur radio waves. In fact, the amateur radio traffic travels as the 'opened text', which means all active amateurs are capable to read it. That includes transmission of passwords as clear text. Following the international amateur radio regulations, some of which were mentioned in [1], any 'ham' (= amateur radio) individual has to transmit messages and other correspondence in clear text. The rules also suggest what types of topics and discussions are acceptable or not. For example, it is completely common to communicate related to the following:

- installation of antenna systems,
- building amateur radio receivers and transmitters,
- power supply and grounding facilities,
- programming the parameters of amateur radio hardware and software,
- fixing small technical problems with computers and amateur radio stations,
- etc.

That does not mean that general educational topics are not interesting for the local radio amateur community. Discussions about preparing technical conferences, papers and tutorials, or recent technical expeditions and interesting school projects as well as non-classified details of scientific research and/or master and doctoral studies are completely fine. In opposite, it is not acceptable to discuss on things that include political, racial, national, social, sexual, professional and similar potentially provocative themes. On the other side, there are not strict distinctions between more or less priorities in the amateur radio communications. It is clear that, according to the laws, emergency cases have priority, particularly when it comes to save human's lives or proprieties. But, in any occasion, one can be sure that amateur radio conversations are as 'private' as talks in, say, public transportation systems, which means there is not much 'privacy' in the amateur radio. Every user of a local amateur radio email server should be aware that unknown amateurs could read the content of his/her messages – either during an end-user exchange of traffic with the nearest email server, or during the exchange of email between 'store & forward' systems.

In such a relatively 'unsafe' environment, it is obvious that most countries have allowed the amateur radio communications primarily for exchanging results of the radio- and computer-related experiments that do not include commercial parts, like advertisements related to selling computers, various shop pricelists or like. The major goal is to establish an ordinary '2-way' communication link between two or more enthusiasts who might also be the local school's kids and persons of same age in other educational institutions, their teachers, friends and relatives. The idea is to increase the popularity of engineering and technology in young people and motivate them to continue education in technical professions – electronics, electrical and mechanical engineering, computer science, hardware, software etc. When it comes to commercial or other topics that are not appropriate for the amateur radio channels, it is the right time to switch to commercial email and similar communicating systems.

2 Privacy in communication

In most cases, the radio amateurs communicate by voice- and computer-related modes and they are mostly capable to differentiate themselves after a few spoken words or even by the few lines sent by their computer systems. In fact, the majorities of amateur radio enthusiasts who live in some area usually know each other, so only newcomers are unknown to the existing 'ham' population. This is just one of the reasons why all amateur radio communications travel in clear regime. In voice communications, the involved correspondents are obliged to exchange their unique identifiers ('*callsigns*' in the radio jargon) every now and then – to inform about their presence on a working radio frequency.

The amateur enthusiasts, who experiment with 'packet radio' – the most popular digital communication mode ("A" and "B" in Fig. 1), are also required to identify with their *callsigns*. The same goes for connections to a local 'ham' radio email server. Communication programs, which are used by the amateur population, have to be properly configured – in away to transmit their callsign every ten to fifteen minutes (depending on the local radio regulations). To ensure that ability, the callsign is a parameter of every end-user class of the amateur radio software. It is the obligation for every particular user to ensure that his/her own callsign is inserted in the program's configuration file(s). There is no logical control if the callsign written into a program's file is a legal identifier of the particular person or not, which means that any software is going to accept any possible callsign. As a result, the actual callsign being used is not only transmitted from time to time, as a flashlight in the dark, but it is also used as a 'user name' for accessing the amateur radio mailboxes. Furthermore, the email server software is going to accept any incoming callsign as a completely valued identifier of the connected user. Following the connecting procedure, after the first initial contact of an end-user with the server, his/her callsign becomes the 'primary key' of a new record, added to the user database, which is maintained within the email server software. All amateur radio server programs keep the track of users' activities that, for example, prevents a user to list or read the same messages he/she had read during the previous visit. The users' database records include not only the callsigns, but also related personal names, their cities' names, postal zip codes etc. Although these databases are not completely standardized, neighboring email servers often automatically exchange their contents between themselves, with the idea of helping their local users to handle e-mail. Practically speaking, if a Russian amateur wants to post a message to an American 'ham', he/she only needs to know the correspondent's callsign that goes to the "To:" field of a new message header. The database will take care to add the recipient local ('home') server's address and location, the shortest path to it etc. That set of records is known as a *WP* (*W*hite *P*ages) database and it is continually refreshed with both new and updated records, [3].

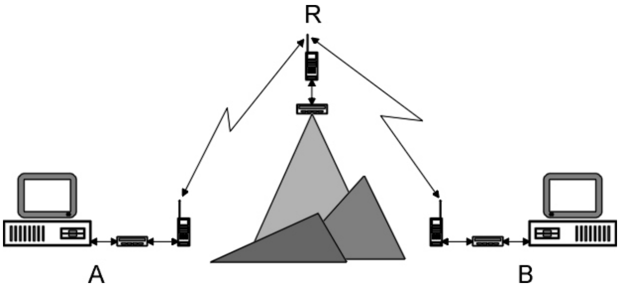


Fig.1 Digital communicators using a radio-relay

Having in mind the friendly environment, mentioned earlier, the radio amateurs do not have (significant) problems related to intentionally misusing callsigns, which means the member of the amateur radio community perform their email procedures in the proper way. Of course, that does not mean that potential amateur radio 'pirates' do not appear on the horizon from time to time.

2.1 Possible problems

If the end-user program parameters are not set properly, various problems might occur. If a wrong technical parameter controls the behavior of the modem or radio station, it is possible not to be capable to establish the communication with the remote station at all. When it comes to 'non-technical' parameters, the wrong callsign means the mistaken identity. That could lead to malicious misuse the third-party traffic, sending non-authored messages to unknown recipients including the use of bad words or some racial and political speech, deleting unread personal email etc.

The procedure for obtaining an amateur radio license (the permission to transmit radio signals) is relatively complex. It also keeps motivating people to conform to the domestic, international laws, and related regulations. For example, in Serbia it is easier to buy a vehicle – regardless the car owner posses a driving license or not, than to buy an amateur radio transmitter. That means a candidate for ownership of the 'ham' radio station must prove that he/she is already qualified enough to handle the system, which, in turn, means that he/she has to take the amateur radio classes, pass an exam and obtain the license. During the educational part of the courses, one of the most important points is to respect the rules, because if not – the radio frequency might get 'clogged' and unusable for ordinary communications. The experience confirms that there are ethically illiterate consumers who do not hesitate to misuse technology. In the amateur radio computer-related communications, malicious users can take actions, which do not contribute our digital systems to grow and develop.

3 Solutions

In this chapter, we will discuss about some existing tools and procedures that are possible solutions for the most popular amateur server software FBB, described in [2], as well as for an alternative program called AA4RE. We are going to base the examples given in this paper on several experiments, either in the real world or in a simulated radio environment.

Whatever operating system is in use (MS DOS, Windows, Linux), the software FBB allows for installing additional tools (commonly called 'servers' or 'PG' programs), which insert more comfort to both system administrators and remote users. One of those tools is a connection filter (*c_filter*) called Protus. By using Protus the system administrators are capable to significantly improve the way their users access the server. That means, in addition to a callsign which plays the role of a *username*, now it is possible to set an optional or mandatory password for each or all users (various combinations are possible). If such a password is really a secret for all but its user and system operator, the integrity of the user access is significantly better than before.

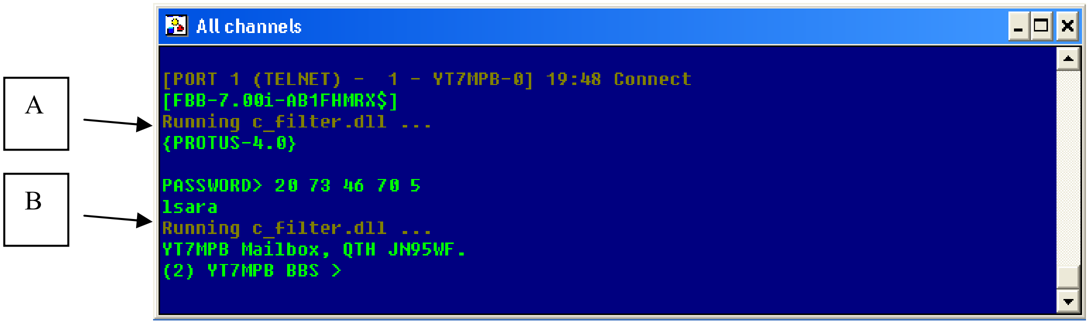
3.1 Experiment 1

The procedure for connecting the server with a password is slightly different from the procedure without the password. That means, after an initial connection is established, the server looks in its password database and checks if there exist a password line for the connecting station. The secret password lines are usually in form of a large string (80 characters or more, see Fig.2). The string might be any possible composition of small and capital letters from the English alphabet and numbers 0-9. If the password line is not set for an incoming callsign and if the server is generally set for the 'open access', the callsign will be given full access without further questions. In opposite, if the callsign is given a password, the server is going to compute a *challenge* (session A in Fig.3), which is a random series of five numbers, who represent positions of the alphanumeric characters from the secret string. The user's task is to return the proper answer – also called a *response* (B in Fig.3), which consists of five alphanumeric characters, 'translated' from the server's *challenge*. The easiest way for an end-user to perform this translation is to compare the received *challenge* with the numbered positions of the characters in the string, like bellow:

```
umoransamodkafanavolimpivoirakijuumoransamodkafanavolimpivoirakijuumoransamodkaf

12345678901234567890123456789012345678901234567890123456789012345678901234567890
1       10        20        30        40        50        60        70        80
```

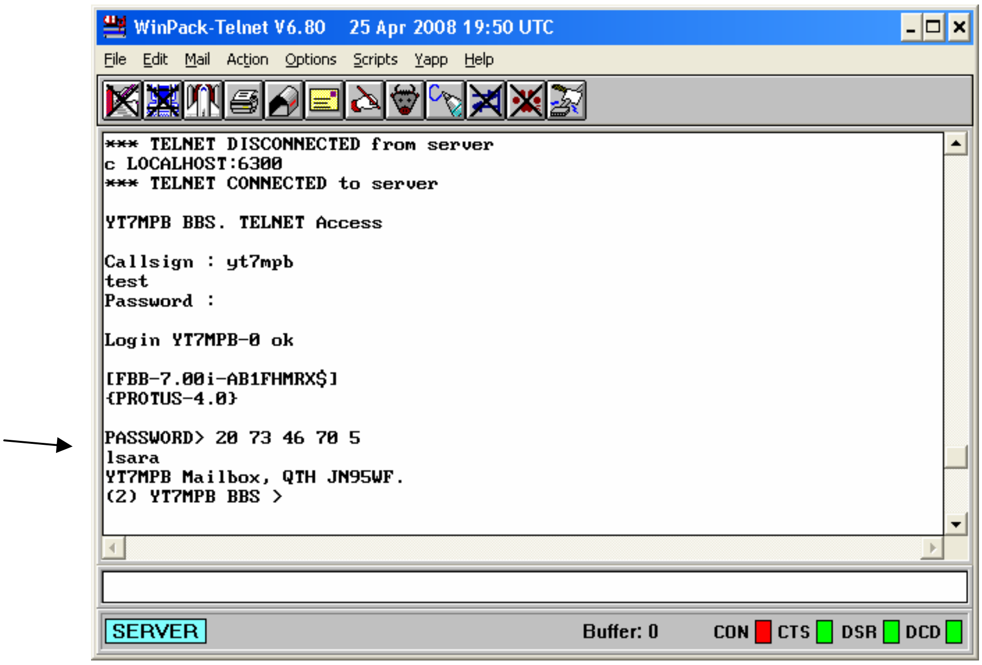
Fig.2 A sample of a large string

The most important factor in using a password system like this one is that the exchange of the large string between any two parties should be a secret activity. That means the users have to find other secure methods in communication: a post letter, a personal contact with the system administrator, or something else *except* the particular communication mode the password is about to ensure. Having that in mind, *packet radio* itself is *not* going to be a safe way for secret string distribution here. However, using passwords as described above is safe for an undefined period – depending on the regularity and frequency in particular users' connections. For those who access their mailboxes, say, ones a day, it should pass a relatively long time to disclose all 80 (or more) characters from the large string. For those who access their mailboxes more frequently, it would be possible that someone performing a thorough surveillance of a radio channel will learn the secret elements of the large string(s) and compromise the passwords much quicker.



```
[PORT 1 (TELNET) -   1 - YT7MPB-0] 19:48 Connect
[FBB-7.00i-AB1FHMRX$]
Running c_filter.dll ...
{PROTUS-4.0}

PASSWORD> 20 73 46 70 5
lsara
Running c_filter.dll ...
YT7MPB Mailbox, QTH JN95WF.
(2) YT7MPB BBS >
```

Fig.3 Two phases at the server's *c_filter*: A) preparing the *challenge*, B) checking the *response*



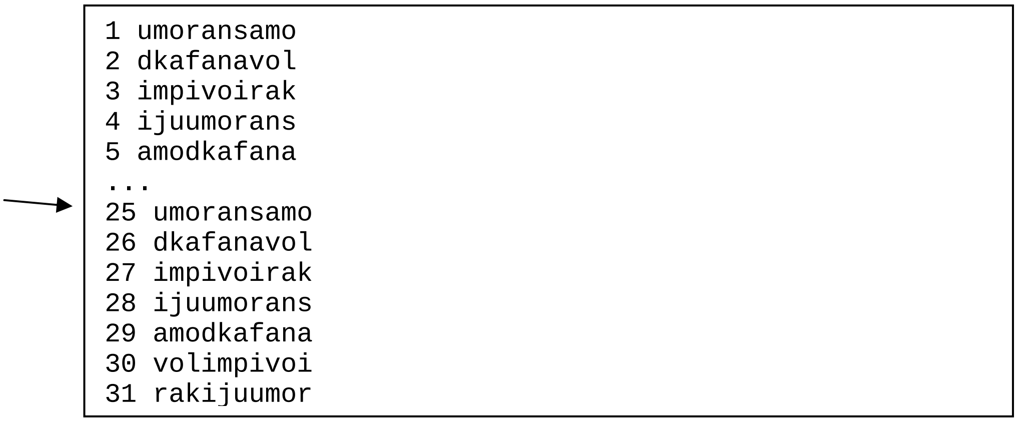
```
*** TELNET DISCONNECTED from server
c LOCALHOST:6300
*** TELNET CONNECTED to server

YT7MPB BBS. TELNET Access

Callsign : yt7mpb
test
Password :

Login YT7MPB-0 ok

[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}

PASSWORD> 20 73 46 70 5
lsara
YT7MPB Mailbox, QTH JN95WF.
(2) YT7MPB BBS >
```

Fig.4 The end-user's view: Password '**lsara**' is a *response* to a *challenge* '**20 73 46 70 5**'

One of the possible solutions for preventing early disclosing the elements from the secret string, is to intentionally replace the elements in either regular or irregular intervals (weekly, monthly or so). It is important that the users perform this change in a manner that any potential 'pirate' is not aware of it. The easiest method which is our proposal, is to 'change' the content of the string in a way that the most left character (i.e. position #1) is moved to the end of the string and the remaining series of 79 characters is moved just one place to the left. Using that approach which, in turn, can be negotiated between the administrator and a user, would keep all existing elements within the string at slightly modified positions, so a 'pirate' will hardly be capable to recognize any change – even after a prolonged period of the radio channel surveillance.

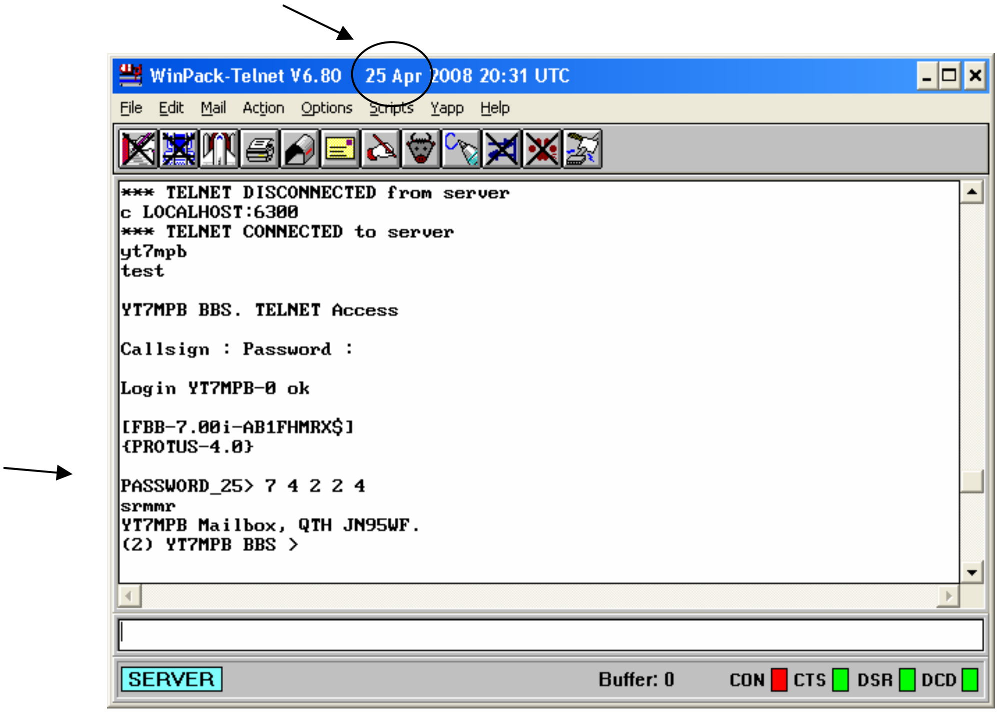
3.2 Experiment 2

Protus *c_filter* has an interesting option: A specific 'table' having 31 rows of text can replace the secret string. Every row represents a day of the month and program is capable to use only the representing row each day. With such a system, the software uses the first row only once a month etc. This method is going to disclose the secret elements of the rows after a very long time of thoroughly performed radio surveillance. Fig. 5 shows a matrix-like table, which uses the elements from the large string, described in the previous example. With a 'table' password, the authenticating procedure is a little bit different. The *c_filter* looks only in the line that corresponds with a particular day in a month.



```
1 umoransamo
2 dkafanavol
3 impivoirak
4 ijuumorans
5 amodkafana
...
25 umoransamo
26 dkafanavol
27 impivoirak
28 ijuumorans
29 amodkafana
30 volimpivoi
31 rakijuumor
```

Fig.5 A sample of the table with 31 rows



WinPack-Telnet V6.80 25 Apr 2008 20:31 UTC
File Edit Mail Action Options Scripts Yapp Help
SERVER
Buffer: 0
CON 🟥 CTS 🟩 DSR 🟩 DCD 🟩

```
*** TELNET DISCONNECTED from server
c LOCALHOST:6300
*** TELNET CONNECTED to server
yt7mpb
test

YT7MPB BBS. TELNET Access

Callsign : Password :

Login YT7MPB-0 ok

[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}

PASSWORD_25> 7 4 2 2 4
srmmr
YT7MPB Mailbox, QTH JN95WF.
(2) YT7MPB BBS >
```

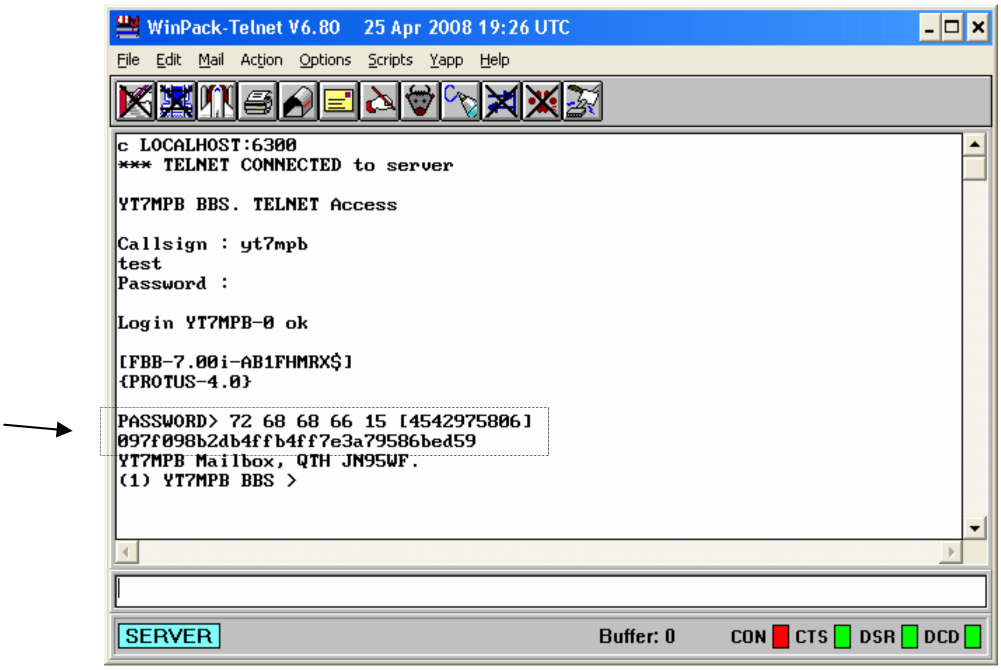
Fig.6 Password ' **srmmr**' is a *response* to a *challenge* '**7 4 2 2 4**'

We performed an experiment on April 25, and on that day, Protus used only the 25th line within the table, see Fig. 5 and Fig. 6. The password syntax routine changed into a format of "**PASSWORD_25**" that means something like "Please, use only the line #25". The implementation of a 'secret table' instead of a 'secret string', significantly improves the reliability of the password safety, because any potential invader will register of only five characters per month (if the particular user accesses his/her mailbox once a day). As a result, the users do not need to replace the content of the table so often. In addition, the use of such format is not complicated from the point of view of an end-user.

For the MS Windows operating system users, it is possible to use client amateur radio software, called *WinPack*, which is capable to prepare the answer automatically by reading its own copy of a 'secret table' file. That simplifies connecting procedures and gives a user more time for reading and writing emails.

3.3 Experiment 3

An even better approach that Protus *c_filter* offers is the implementation of the *MD* (*M*essage *D*igest) algorithm, [4]. In our test, it looks similar to described Experiment 1 (a single large string), with an addition: The server's *challenge* also has five numbers which represent the positions of alphanumeric characters within the secret string, but now it includes a 10-digit **[square brackets]** sequence, see Fig. 7. (In our example, it is **[4542975806]**). When the client's software (which must be capable to understand *MD* cipher technology) establishes the link and receives such *challenge*, it uses the sequence in square brackets to compute its *response*. In our case, the *response* is **097f098b2db4ffb4ff7e3a79586bed59**. Then the client's program sends the calculated *response* back to the server, which, in turn, makes a similar computation on its side. Finally, the server compares results. If the results are the same, the server grants the access to the connecting callsign. In opposite, if the results are different, an immediate disconnection occurs, following a warning message to the system administrator.



```
c LOCALHOST:6300
*** TELNET CONNECTED to server

YT7MPB BBS. TELNET Access

Callsign : yt7mpb
test
Password :

Login YT7MPB-0 ok

[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}

PASSWORD> 72 68 68 66 15 [4542975806]
097f098b2db4ffb4ff7e3a79586bed59
YT7MPB Mailbox, QTH JN95WF.
(1) YT7MPB BBS >
```

Fig.7 *WinPack* enables its user to prepare an *MD response* automatically

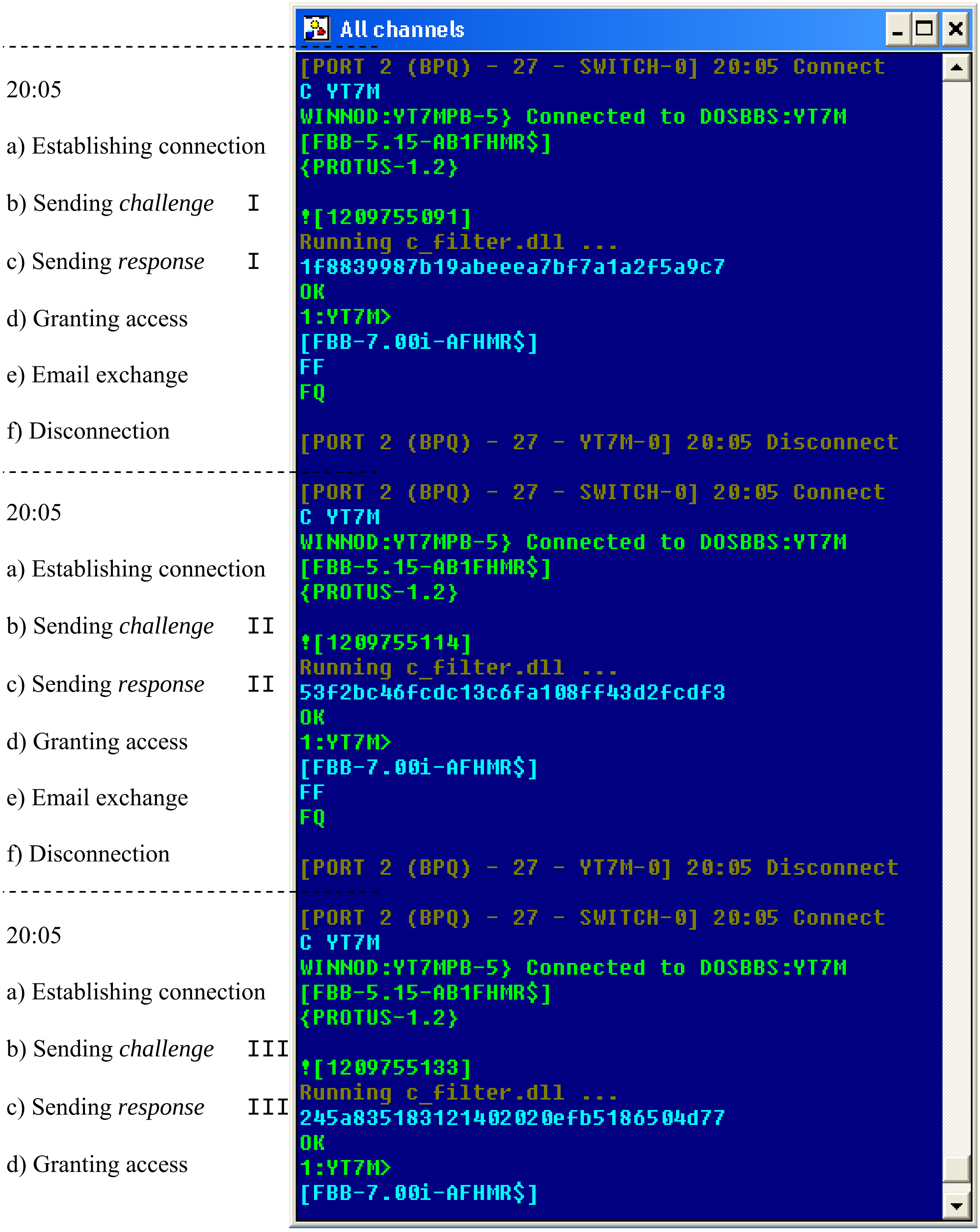
As we can see from all three examples, the implementation of Protus connection filter offers the system administrators a whole spectrum of various safety mechanisms to ensure the integrity of their users' email activities. In the same time, the level of responsibility in all participants of the amateur radio traffic, including not only the end-users but also the system administrators, is increased.

3.4 Experiment 4

If any two neighboring system administrators use Protus, it is possible to establish a special automatic BBS-to-BBS[1] protected forwarding session between two servers. That mode does not expect any manual input from the system operators and allows the fast mutual 'recognition' if the two systems implement the same password-authorizing tool. Every next connection results in completely new *challenge* and *response*, which never contain visible elements of the secret string of alphanumeric characters. In practice, that means such a system is completely satisfying any possible amateur radio safety

[1] *BBS* stands for the *B*ulletin *B*oard *S*ystem (the email server).

requirements, because the potential 'pirate' cannot produce the secret key from the recorded transmissions between the two parties. Our next experiment describes a network simulation of three short sessions between email systems YT7M and YT7MPB. A 'local' server YT7M performs the authentication of a 'remote' server YT7MPB, by using a shortened version of the *MD2* algorithm. As it can be seen, the first session's exchange of the *challenge* **![1209755091]** and its *response* **1f8839987b19abeeea7bf7a1a2f5a9c7** is fully automatic and time-efficient, which allows the partnering stations to authorize their credentials for even three times in just a single minute – without any supervisors' interaction! (Fig. 8)



```
[PORT 2 (BPQ) - 27 - SWITCH-0] 20:05 Connect
C YT7M
WINNOD:YT7MPB-5} Connected to DOSBBS:YT7M
[FBB-5.15-AB1FHMR$]
{PROTUS-1.2}

![1209755091]
Running c_filter.dll ...
1f8839987b19abeeea7bf7a1a2f5a9c7
OK
1:YT7M>
[FBB-7.00i-AFHMR$]
FF
FQ

[PORT 2 (BPQ) - 27 - YT7M-0] 20:05 Disconnect

[PORT 2 (BPQ) - 27 - SWITCH-0] 20:05 Connect
C YT7M
WINNOD:YT7MPB-5} Connected to DOSBBS:YT7M
[FBB-5.15-AB1FHMR$]
{PROTUS-1.2}

![1209755114]
Running c_filter.dll ...
53f2bc46fcdc13c6fa108ff43d2fcdf3
OK
1:YT7M>
[FBB-7.00i-AFHMR$]
FF
FQ

[PORT 2 (BPQ) - 27 - YT7M-0] 20:05 Disconnect

[PORT 2 (BPQ) - 27 - SWITCH-0] 20:05 Connect
C YT7M
WINNOD:YT7MPB-5} Connected to DOSBBS:YT7M
[FBB-5.15-AB1FHMR$]
{PROTUS-1.2}

![1209755133]
Running c_filter.dll ...
245a835183121402020efb518650 4d77
OK
1:YT7M>
[FBB-7.00i-AFHMR$]
```

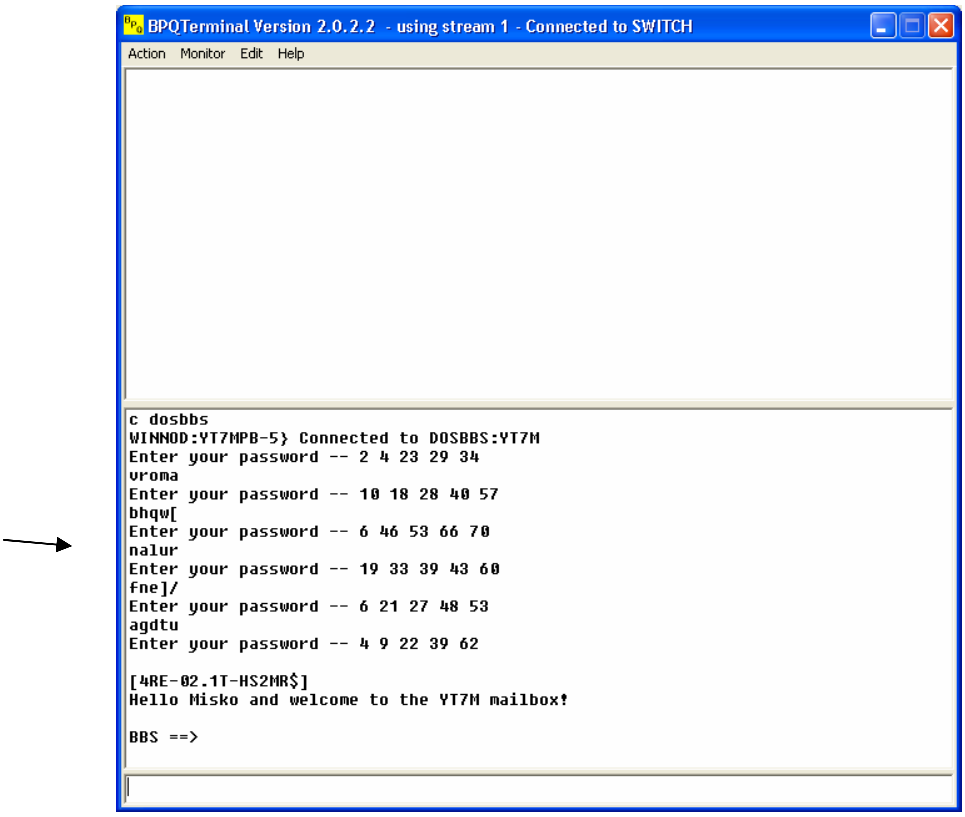
Fig.8 Three successful *MD2* authentications between FBB e-mail servers

3.5 Experiment 5

In the next primer, we were testing secure connections to alternative e-mail server software, called AA4RE, which is capable to run on 'vintage' computers of types PC XT or PC AT, equipped with CPU 8086 or 80286 and 640 kilobytes of RAM memory. Despite its maturity and date of manufacture – early nineties, the newer versions of AA4RE are "Y2K" compliant, which was a general requirement for reliable e-mail *store & forward* programs. Actually, authors of AA4RE understood the needs of many radio amateurs who kept using ancient computers in their amateur labs. To be precise, FBB has also solved the "Y2K" issue on time – but only in its newer versions, which are not capable to install on the older computers mentioned in this work.

During the tests, we were not capable to confirm eventual compatibilities between systems implementing AA4RE and those running FBB + Protus, what we found as a disadvantage of the former server program. On the other side, AA4RE includes an option that has been in use for accessing amateur radio-relay stations ('digipeaters'). Those stations are often positioned at remote locations (mountaintops etc) that are not always accessible. In the same time, relay stations do not contain computers, so their administrators could re-program the electronic circuits only when on site. That means a frequent replacement of a secret string is almost impossible. To avoid such situations, radio amateurs needed to invent solutions that would ensure safe remote administrative access from radio networks and would not require frequent change of the elements in the secret alphanumeric string. A solution that proved as reliable is similar to the one from Experiment 1, which returns the right answer of five alphanumeric characters, but now the right answer can be inserted into a longer 'word' – so a potential intruder would intercept an unexpectedly longer user's *response*.

Besides that, AA4RE gives an opportunity to accept and analyze not only a single row (a single line) with the right answer, but a 'composition' of several lines. Such approach transforms (disguises) a relatively simple phrase to a more complex 'table', described in Fig. 9. AA4RE ignores everything but the right part of the answer. The e-mail server knows that the real end of such a 'composition' occurs after receiving an empty line, which happens after pressing the *Enter* key twice.


BPQTerminal Version 2.0.2.2 - using stream 1 - Connected to SWITCH
Action
Monitor
Edit
Help

```
c dosbbs
WINNOD:YT7MPB-5} Connected to DOSBBS:YT7M
Enter your password -- 2 4 23 29 34
vroma
Enter your password -- 10 18 28 40 57
bhqw[
Enter your password -- 6 46 53 66 70
nalur
Enter your password -- 19 33 39 43 60
fne]/
Enter your password -- 6 21 27 48 53
agdtu
Enter your password -- 4 9 22 39 62

[4RE-02.1T-HS2MR$]
Hello Misko and welcome to the YT7M mailbox!

BBS ==>
```

Fig.9 A disguised *response* '**nalur**' is easily recognized by AA4RE e-mail server

4 Discussion

4.1 Apparatus

For described experiments, one can use the following equipment:

YT7M
- Computer PC AT 80286 CPU clock 12 MHz, 1 MB RAM,
- Operating system MS DOS 5.0,
- Network node software BPQ 4.08a,
- E-mail server software DosFBB 5.15c, AA4RE 2.13t,
- Protus 1.2 and Protus 3.3 for DOS.

YT7MPB
- Computer P2 Celeron CPU clock 400 MHz, 224 MB RAM,
- Operating systems MS Windows XP, MS Windows 2000, Linux Mandrake 9.1,
- Network node software BPQ 4.10d for Windows, Linux Node,
- E-mail server software WinFBB 7.00i for Windows, LinFBB 7.03g for Linux,
- Protus 4.0 for Windows, Protus 4.1b2 for Linux,
- Terminal software WinPack 6.80 for Windows.

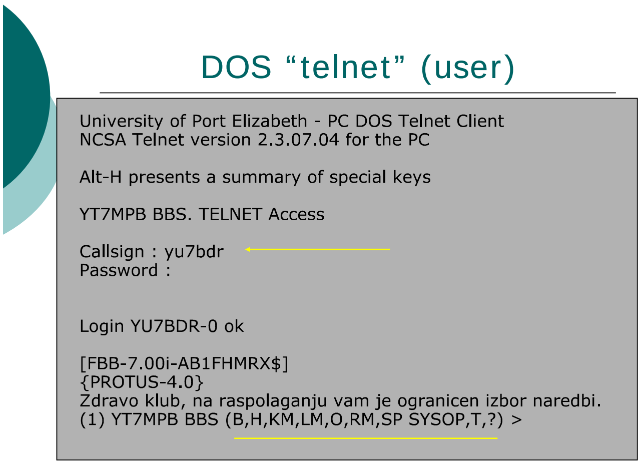
As shown above, we did not invest in brand new equipment. Instead, we rather experimented with, say, average personal computers capable to run Windows and Linux operating systems, and some outdated machines for MS DOS or PC DOS. If both parties ran Windows or Linux versions of FBB software (accompanied with updated versions of Protus *c_filter*) then an improved *MD5* algorithm would bring even more safety to the customers.

Other differences between versions of Protus for DOS and for Windows/Linux are also visible in some other features, mainly related to their abilities to understand other parties' *challenges*. For example, Protus for DOS v. 1.2 seems not to be capable to compute a *response* to the Protus for Windows' *challenge* – which results in a broken link. Such situation occurs when a DosFBB server initiates an outgoing connection request to a WinFBB system. (In the opposite direction, when a WinFBB attempts to establish the link with a DosFBB machine, described handshaking goes smoothly.)

4.2 Additional options

Besides our main goal in the experiment with password tool modes intended to give more security to the mutual BBS-to-BBS interactions, the implementation of the automated authentication gives us the opportunity to save our working radio frequencies from overloading and increased traffic. In fact, during an exchange session between two systems ('forwarding'), running on the international HF radio waves – mainly intended for the automatic store & forward activities, it is important to ensure that on the same channel there are no other users (the end-user 'intruders' who want to access the mailboxes manually). There are several reasons for that policy: The HF bands are prone to fading and even a slight fade is enough to cause data loss. Besides that, bad weather conditions, noise or interference, or that entire combined, can give a lot of frustration to the system administrators, [5]. In that manner, it is possible to set only the passwords for collaborating servers on the otherwise 'closed' systems, so the other connecting stations would be disconnected immediately. In addition to such a rather rigid decision, it might still be possible to reserve some time slots for the 'open' access of those end-users, provided the automated exchange sessions are finished. According to the good practice and common amateur policies, which are included in so called 'ham spirit', there is a variety of available system messages within the Protus tools – signaling their users that the safety measures are taking place.

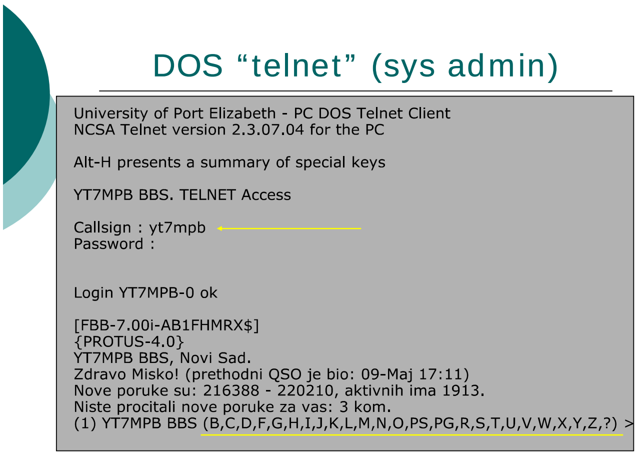
The amateur packet radio might be an interesting educational tool for increasing motivation in young generations for studying engineering and technology, [6]. If the amateur radio infrastructure is linked to the computer network of a school or university, it is important to take care of privileges given to the students. Program FBB is fully capable to differentiate low-risk privileges intended for ordinary '*telnet*'-users within a LAN, from the high-level administrative tasks, Fig. 10-11. The slides show that lower privileges give only restricted access to the user's personal email account: ListMine (LM), ReadMine (RM), KillMine (KM) are the most suitable commands for a student (calssign YU7BDR, Fig. 10), while the system administrator (callsign YT7MPB, Fig. 11) is allowed to do everything else, including removing inappropriate content (if any), accessing radio gateways (if installed) etc. Note that different privileges can be configured not only for the users within a LAN, but also for those who come through the radio channels or Internet connections, and so on.



DOS “telnet” (user)

```
University of Port Elizabeth - PC DOS Telnet Client
NCSA Telnet version 2.3.07.04 for the PC

Alt-H presents a summary of special keys

YT7MPB BBS. TELNET Access

Callsign : yu7bdr
Password :


Login YU7BDR-0 ok

[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}
Zdravo klub, na raspolaganju vam je ogranicen izbor naredbi.
(1) YT7MPB BBS (B,H,KM,LM,O,RM,SP SYSOP,T,?) >
```

Fig.10 Low-privileged user



DOS “telnet” (sys admin)

```
University of Port Elizabeth - PC DOS Telnet Client
NCSA Telnet version 2.3.07.04 for the PC

Alt-H presents a summary of special keys

YT7MPB BBS. TELNET Access

Callsign : yt7mpb
Password :


Login YT7MPB-0 ok

[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}
YT7MPB BBS, Novi Sad.
Zdravo Misko! (prethodni QSO je bio: 09-Maj 17:11)
Nove poruke su: 216388 - 220210, aktivnih ima 1913.
Niste procitali nove poruke za vas: 3 kom.
(1) YT7MPB BBS (B,C,D,F,G,H,I,J,K,L,M,N,O,PS,PG,R,S,T,U,V,W,X,Y,Z,?) >
```

Fig.11 High-privileged user

4.2 Conclusion

The examples given in this paper intended to inform potential newcomers to the exciting world of amateur radio that many counter fighting measures can be successfully used against eventual would-be 'ham pirates'. One can be sure that there exist varieties of methods that help our daily amateur radioactivities to be safer. Depending on ideas in the amateur radio programmers' community, there appear various principles for making 'ham' digital communications more secure. Some approaches are better with specific server and client software; the other might be universal.

We encourage the readers to explore new horizons by their own, because there is not much literature on this topic available.

Final tips and reminders:

- System administrators who do not have so many users (and even less password-protected ones), would do the best if implement the simplest security mechanism of a secret 'large string'. If needed, from time to time, the content of the string can be fully exchanged or slightly modified (as described) – if proven to become compromised.

- Those 'sysops' who want to experiment with approaches that are more complex and whose customers use newer communicating programs shall test the system of monthly 'tables'. At first, you can try to use the same 'large string' as a foundation for the daily lines within the table. If it proves not to be safe enough (i.e. because of just a few elements in a row), you can try with longer daily lines. Make some more tests and compare the results.

- If the password-protected users are equipped with communicating software that integrate automated safety routines (as *MD* algorithms), use it whenever possible, because the computing process itself would be efficient and fully transparent for your end-users.

- If you and your collaborating neighboring partner(s) implement updated Protus tools, enable some of the automatic BBS-to-BBS modes for corresponding callsigns. You would not need to replace the content of the secret key for a very long period.

- Keep an eye on daily routines in mailbox activities of your new users, to get familiar with their behavior. Keep them informed about the existence of the safety facilities you have implemented within your server.

- Get familiar with the safety lessons within the amateur radio classes and courses in your area. Advise their lecturers about the safety measures, which are the counter fighting methods against poorly educated beginners in the 'ham' radio community.

- Should you, as a scholar or student, consider implementing the amateur radio technology in your educational environment, you do not have to be a computer expert! If you can read operating manuals and follow instructions, you can easily become an active *packeteer*. "Like any other facet of Amateur Radio, you're going to have to learn some new concepts, but that's part of the enjoyment", says Steve, WB8IMY, [7].

References:

[1] M. Skoric, The new amateur radio university network – AMUNET, 9th WSEAS Int. Conf. on Computers, *Proceedings*, 2005.

[2] M. Skoric, *FBB Packet-radio BBS mini-HOWTO*, The Linux Documentation Project, http://tldp.org/HOWTO/FBB.html, 2000-2008.

[3] G. Jones, *Packet Radio: What? Why? How?*, TAPR, 1996.

[4] RSA Data Security, Inc., MD2/MD5 Message-Digest Algorithm.

[5] S. Ford. *Your HF Digital Companion*. ARRL, 1995.

[6] M. Skoric, The Amateur Radio as a Learning Technology in Developing Countries, *Proceedings*, IEEE ICALT-TEDC, Joensuu, Finland, 2004.

[7] S. Ford. *Your Packet Companion*. ARRL, 1995.