Keystroke Recorder Attack on a Client/Server Infrastructure

Randy Marchany, Tom Wilson
VA Tech Computing Center
Blacksburg, VA 24060
8/2/96

Abstract

This paper describes and analyzes a simple attack on a client/server infrastructure. We had no knowledge of the software being used by the client/server systems but did have moderate knowledge of the Macintosh computer. We built a trojan program that installs keystroke recorder software on target client systems without the owner's knowledge. The recorder software captured all characters entered by the client software users. The recorder log was then transferred back to a Mac system for analysis. This attack can be Internet or Intranet based. This is a demonstration of the ability of a novice attacker to use sophisticated tools with success on a typcial client/server infrastructure.

"No true hacker would ever admit to his peers that he could break into a Mac. What skill is there in that?"

-Internet Truism

1.0 Introduction

In general, standard PC and Mac operating systems do not employ adequate access controls to prevent anyone from installing unwanted software on the system. This poses one of the greatest threats to a company's client/server structure. PC's and Mac's are vulnerable to a wide range of virus, trojan horse, and socially engineered software attacks.

There has been considerable discussion on the security features of the client/server projects. IS staffs have anticipated that network sniffer software would be employed to intercept traffic between the client and server systems. A number of interlocking defenses such as data encryption and subnet isolation are being employed or considered by the IS staffs. Tools such as Kerberos or Oracle's SNS are being evaluated as methods of

defending against sniffer attacks. In this demonstration, we used a Macintosh system simply because we had one in our office. We would like to emphasize that we could have done the same attack using a PC system. The attack we present in this paper shows the weakest point in a typical client/server architecture now becomes the client system itself.

2.0 Attack Methodology

After reviewing possible attack scenarios, we used the Netscape browser program to connect to a well known Internet search engine at *www.altavista.digital.com*. and searched for any WWW site that had the phrase "Macintosh Hacking" in its WWW pages. This search engine has references to over 15 million WWW pages on the Internet. The search yielded over 50 sites on the Internet that had such tools. Figure 1 shows one of the pages that resulted from our search. We selected a site that had keystroke recording software. This software (available in Mac or PC flavors) is designed to record every character that is entered by a user. It stores the log in a hidden area on the system which could then be retrieved manually or over the network.

We were surprised to find both shareware and commercial versions of Keystroke Recorder software. Apparently, companies use these products to record which commands are being executed by their employees. Therefore, one cannot claim that this type of software was developed by and for hackers only. We found PC, Mac and Unix versions of these commercial tools.

We downloaded one software package to a test Macintosh and examined the code. The code (called Invisible Oasis) came in 3 parts: an Installer, a README file containing a description of the application and the extension code. Figure 2 shows the components of the keystroke recorder kit. The README file is given in Appendix A of this report. The software was packaged for easy and quick installation. Total installation time was less than 2 minutes.

This software is capable of recording keystrokes, titles and menu selections at a *very* low level. It captured data from the Finder screen or from any application window (NCSA Telnet, Oracle Forms Runtime, Fetch, Netscape). In most cases, it also displayed Window Titles and menu selections so one could reconstruct the menu access path. It even displayed access to the Powertalk Keychain.

Using the Invisible Oasis software, we quickly developed a trojan horse program that captures keystroke, titles and menus to demonstrate a passive attack scenario for unprotected client machines. We would rely on a social engineering attack to activate the software on the target machine. The social attack would be in the form of a mail message that asked the recipient to install an "upgrade" to some client software component. We picked the Eudora mailer as the component but it could have been any client software

component such as Oracle Forms, MacTCP, or new anti-virus software. Appendix B shows other ways the trojan program could be packaged.

The total time spent to design, build and package the trojan software was approximately 3 hours. The time spent on each phase was:

1. Conceptualize the attack - 30 minutes
2. Design the attack tools - 30 minutes
3. Prototyping - 30 minutes
4. Packaging production version, debugging - 90 minutes

We would like to stress that all of this time was spent to make the software appear to be something else. The attack software was ready to use and would take less than 1 minute to install on a target machine.

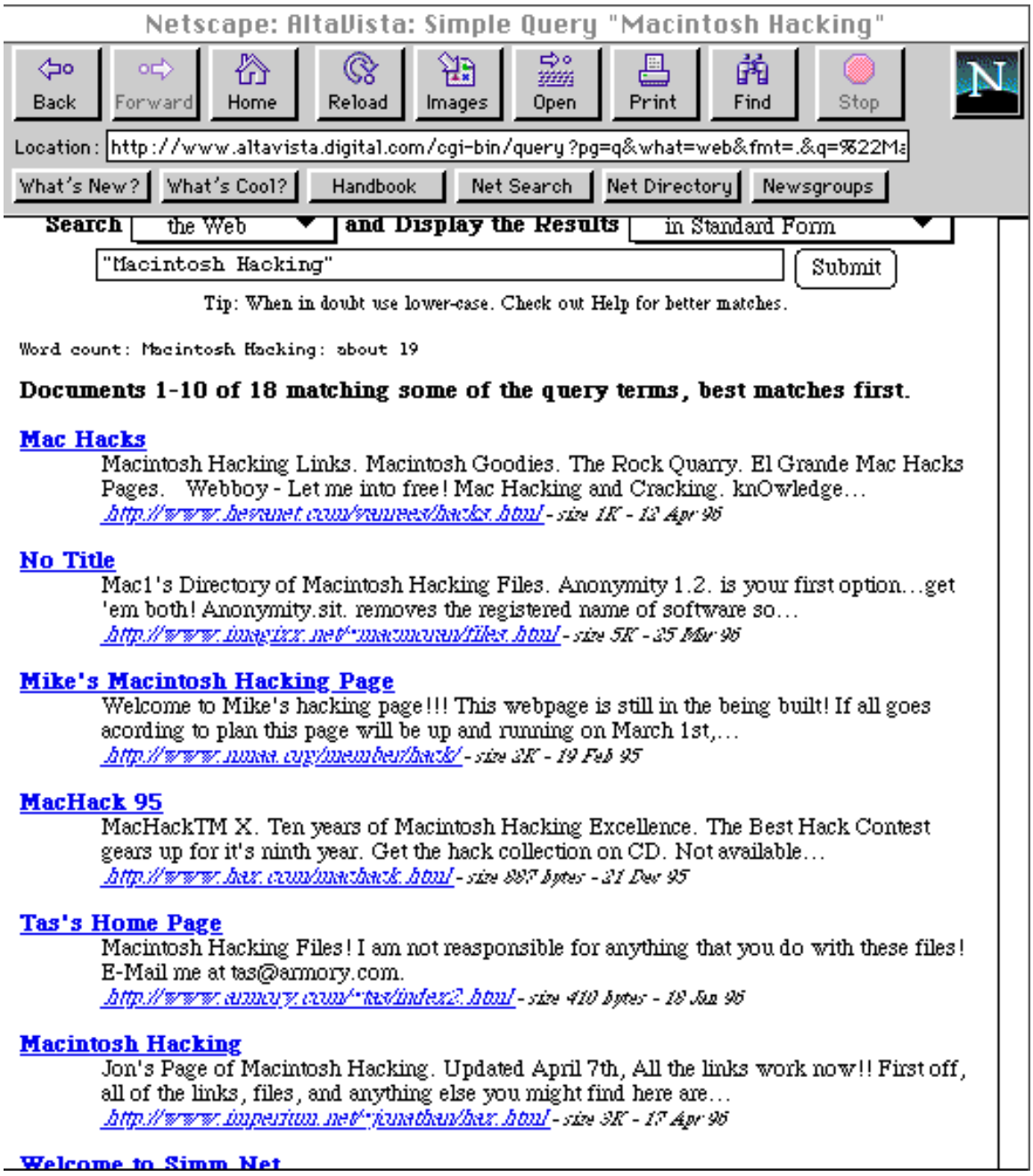


Figure 1. Sample Page Showing Altavista Search for Hacking Tools

2.1 Physical Access Attack

Our attack scenario is to simply walk up to an unprotected Mac with a floppy disk containing the recorder software. We loaded the diskette, installed the software and removed the disk, restarted the system to activate the recorder software and left the site.

Elapsed time was 30 seconds. We came back later to retrieve the log and uninstall the software. Elapsed time for this effort was approximately 1 minute.

No one disputes the fact that unenhanced PC and Mac physical security is inadequate. Even when software products (At Ease, On Guard, etc.) are used to help protect certain areas of the system, we find there are tools available to anyone on the Internet that are designed to break through this level of security. Most of these tools can be found on the Internet.

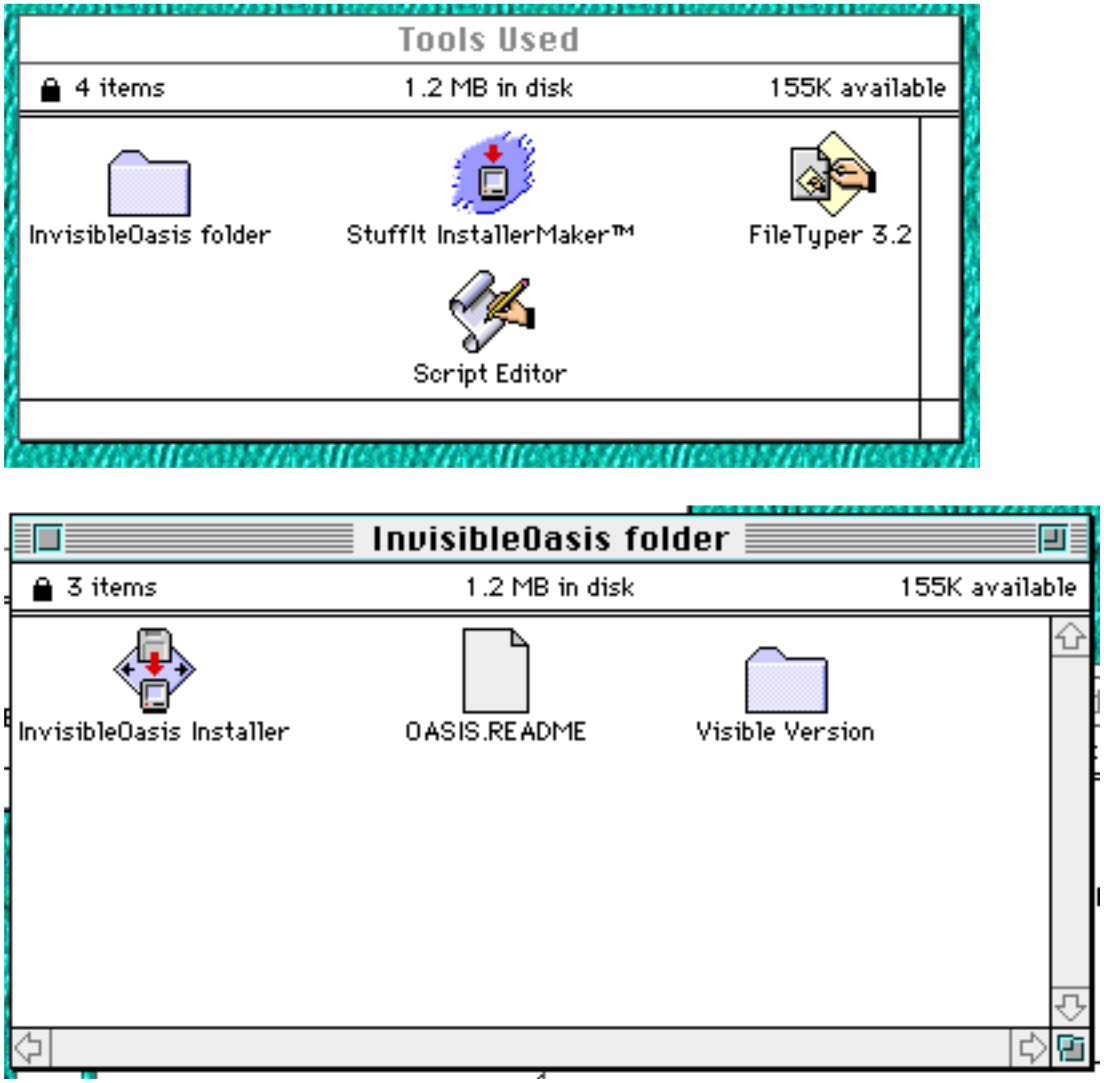


Figure 2. Tools Used to Construct the Trojan Program

2.2 Network Based Attack

A network based attack is possibly the more dangerous scenario. We wanted to see if we could download the trojan to a target system and a) trick the user into executing the program b) have the program load itself without any user intervention.

2.2.1 Introducing the Trojan via Mail

We decided to see if we could design the Trojan in a manner that would allow us to get the user install it for us. We packaged the recorder software as an alleged Eudora update, made it an attachment in a mail note and sent the note to the owner of the target machine. The mail note advertised a new Eudora upgrade and asked the recipient to install it immediately. We built the trojan package by doing the following:

1. We used the Stuffit Installer Maker program to build a new installer package. We made sure the installation process operated in "silent mode". Figure 2 shows the tools used by us to build the trojan program.
2. We created a bogus Eudora Plug In icon and moved this to the trojan installer kit so it would appear to be a valid Eudora Plug In module. The trojan icons as they appear in a Mac window are shown in Figure 3. The Invisible Oasis Installer Icon is visible for the purposes of this demonstration. Normally, it would not be visible in our attack scenario.
3. We built an Applescript program that removed the disk version (keeping the only copy running in memory) of the recorder software on system boot. The keystroke buffer is closed when the system is shutdown so the Applescript program (invoked at startup) would copy this buffer to a remote machine and erase the local log when the system was booted. Figure 4 shows the text of the Applescript program.
4. We then searched the network for a Mac that had write access to its system enabled through commonly available "Public Folders". In our first version, we simply used that system to store the keyboard logs. We anticipate that this is what a real attacker would do. For the purposes of this demonstration, we used a Mac in our office area.
5. The recorder software installed itself on the client machine as soon as the mail recipient opened the mail note and double clicked on the attachment. Double clicking

the trojan icon would install the keystroke recording software in the Preferences, Extensions and Startup Items Folders on the system.

2.2.2 Introducing the Trojan via Servers

We did not attempt to implement an server based attack at this time. We believe it would be quite simple to set up an alias to a well-known client/server program and include the trojan package as a front end to the real package. Double clicking on the "alias" would invoke the trojan installer program and then invoke the real program.

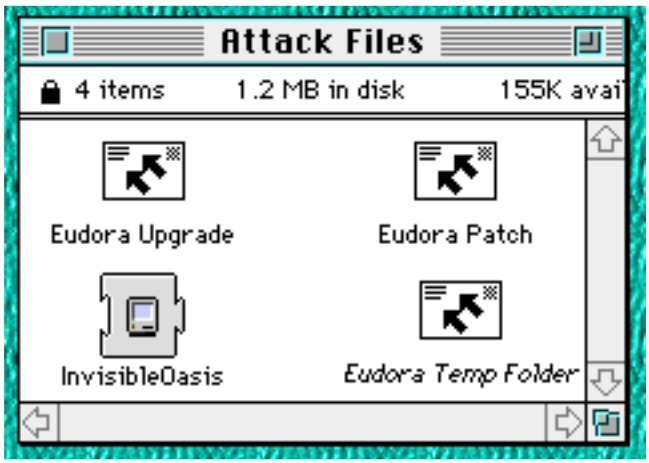


Figure 3. Trojan Program shown as Eudora Plug Ins.

2.3 Obtaining the Stolen Information

The Applescript program (Figure 4) was invoked at system startup. It opens a connection to a remote Mac and copies the keyboard log to a file on that system. It would then erase the buffer files from the victim's system. We then examined the log to look at all the keystrokes typed in by the user. A sample log output is shown in Figure 5.

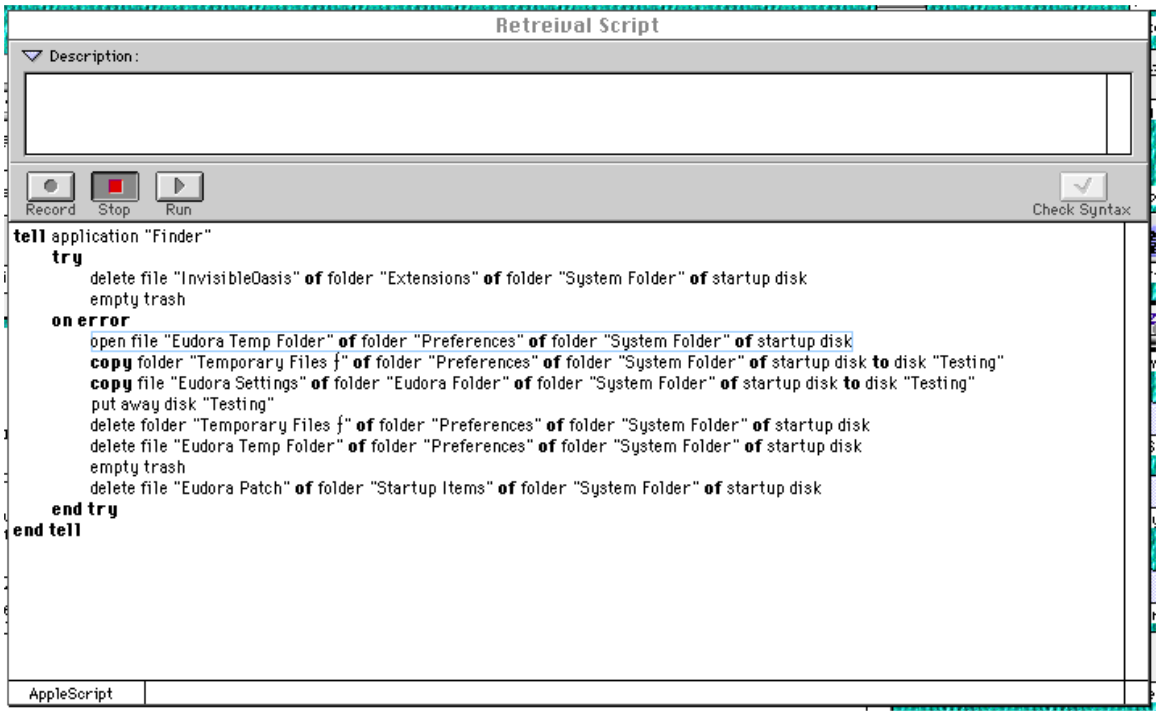


Figure 4. Applescript Retrieval Program

§New Connection…§
□□vtusd.cc.vt.edusomeuser person

§error DLOG§
□
§New Connection…§
bogus
§Netscape§
ftpvttserf.cc.vt.edu@v
§§
ftpbogus
§New Dialog§
vtusd.cc.vt.edu
§vtusd.cc.vt.edu 1§
randymarchany
telnetbogus
§Sharing Setup§
appleboguspassword
§§

```
s
§Preferences§
finderfinder
```

Figure 5. Sample Keystroke Recorder Output

3.0 Discussion

We were extremely surprised at the ease of constructing the attack and its success. Lack of adequate access controls on the client systems, up until now, was deemed to be an acceptable risk in most environments. However, the existence of keystroke recorder software severely threatens this assumption.

Security architects realize that it is important to be able to determine WHO has accessed the client system. There is "locking" software designed to prevent unattended access (At Ease, On Guard etc.) but even this software is a layer that may be subverted by tools freely available on the Internet. We found attack tools by simply entering "Mac hacking tools" as an entry to a standard Internet search engine.

We found locking tools do provide some protection. We tried to attack a Mac that was "protected" by On Guard with no success. The locking tool prevents a general user from installing the software. However, the same type of social engineering attack would probably work against the privileged user. It should be noted that NONE of the shareware antivirus software for Macintoshes detected our trojan program.

The recorder software we used came preconfigured to run immediately on a target system with little or no knowledge required of the attacker. Indeed, most of the time spent on this exercise was on the presentation portion of the attack package. A relatively novice hacker could obtain these packages and inject them into the network in under 30 minutes.

Captured keystroke logs were very complete in the information they obtained. Most window titles as well as the userids and passwords of the standard Mac Internet tools (Fetch, NCSA Telnet) are displayed. The keystroke software also logged access to the Powertalk Keychain. This allows the attacker to masquerade as a valid person and present the proper commands in the proper sequence to a client/server application.

This exercise demonstrates the need to install anti-tampering tools on the client machines. Users must be trained to resist socially engineered attacks. These tools are not the complete answer but are certainly a good beginning.

Additional measures including search programs that identify a list of hacking tools on the client machines will have to be developed and run to sweep the client systems. These measures should be in place before the current version of the client/server project can be operated in a safe and secure manner.

4.0 Conclusion

We make the following recommendations:

1. Anti-tampering tools must be installed on all client machines. No client system without an anti-tampering tool installed on it should be allowed to connect to sensitive server systems.
2. Phone or face-to-face confirmation of modifications to client programs including those NOT directly related to a project should be required. This should prevent an "Software Installation" attack similar to the one we discuss in this paper. This will most likely require an additional training section on security.
3. Network security tools should be installed on all of the server systems. We've demonstrated that physical security isn't the ONLY threat to a server system. Adequate access controls, source change and network access logging capabilities need to be developed and installed on the server systems.
4. Periodic sweeps or software inventories of the client systems will need to be performed. Any new software should be listed and verified.

We did this exercise after spending years talking about attacks on our systems and not having a real-life demonstration of such an attack. Hopefully, this demonstration will confirm and provide justification for the continuing efforts to improve overall network and system security infrastructures.

APPENDIX A

Keystroke Recorder README File

The following is the README file from the keystroke recorder file used in this attack. It is included to give the reader an idea of the sophistication of the tool and the lack of knowledge required to install it. “Fatal Error” is the nickname of the person who packaged the kit.

(Note: This was not written by the author)

Oasis is a simple extension, place it in your victims extension folder, and upon the next restart, it will record all keystrokes and place them in a folder called Temporary Files in the victims Prefs folder. This is useful for grabbing passwords, or intercepting love-letters for black-mail, or material which can be sold for fun and profit ;)

-Fatal Error

InvisibleOasis

...is based on the original Oasis. It collects all keystrokes, and saves them in the System Folder/Preferences Folder/Temporary Files folder of the computer it's placed on. To install, just double-click on the icon...an idiot could to that. The original Oasis you could see in Extensions Manager, and this one you can't. It's that simple. It creates a new file on each startup. All deletes are shown with a box, so if you saw

username: johndoo<box>e
password: aolsuxz<box>

it would really be

username: johndoe
password: aolsux

This program was originally found at Knight Hawk's FTP site.
ftp.winternet.com:/users/nitehwk in the /hack/mac directory.
Please check out the site, u/l please, no "w4r3z" please.
Also, check out the new ezine entitled PHAUN at his ftp site in

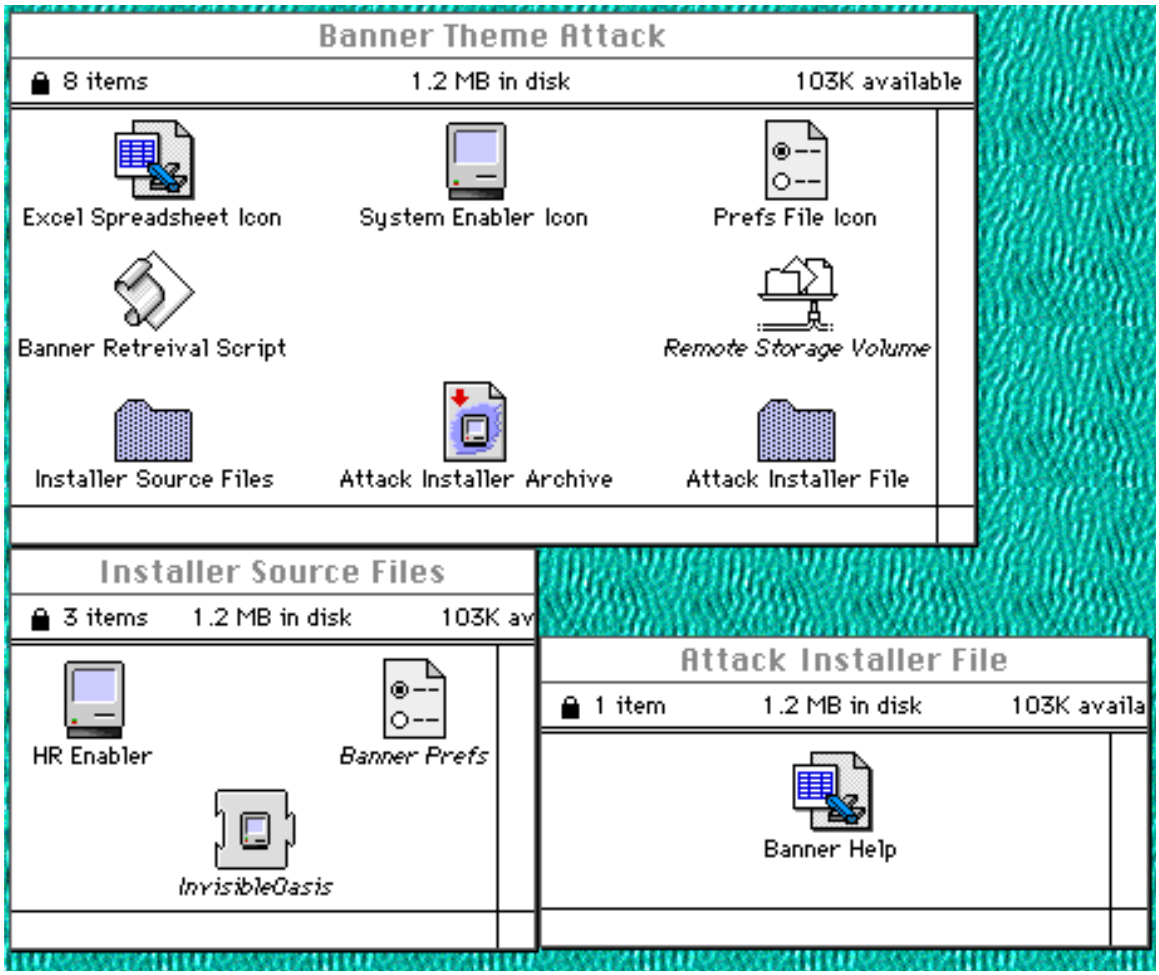
/zines/phaun directory.
PHAUN - - Phreakers, Hackers, & Anarchists Underground Nation

File written by:
Mr Phaun & staff of PHAUN

APPENDIX B

Other Ways to Package the Trojan Program

The trojan program could be packaged to look like an Excel spreadsheet, Banner Help file or anything else.



Title: Keystroke Recorder Attack on a Client/Server Infrastructure

Type: PAPER

Authors:

1) **Randy Marchany**, VA Tech Computing Center, 1700 Pratt Dr., Blacksburg, VA 24060

Phone: 540-231-9523

FAX: 540-231-7413

Email: randy.marchany@vt.edu

2) **Tom Wilson**, VA Tech Computing Center, 1700 Pratt Dr., Blacksburg, VA 24060

Phone: 540-698-6014

Email: tom.wilson@vt.edu

Point of Contact: Randy Marchany