

CyberSoft White Papers

The Plausibility of Unix Virus Attacks

Prescript, April 1996

I am still amazed at the number of people who somehow believe that UNIX is immune to software attack. Recently I was the subject of a heckler at a conference in which I was speaking on this subject. It appears that this is a subject that still angers some people so much that they become obnoxious. Days later, a high level technical manager of a very savvy firewall company made the statement that UNIX viruses don't exist and thereby killed an opportunity to port VFind (VFind is a "virus scanner" that executes on UNIX systems and searches for UNIX, MSDOS, Macintosh and Amiga attack programs) directly to their firewall. I can only state that those individuals who work hard and diligently at remaining ignorant of the world around them have themselves as their most appropriate punishment.

This paper was first written as a rebuttal to a paper published by a Senior Scientist within the US Federal Government. Doctor Fred Cohen (fc@all.net) also published a rebuttal which, at that time, was distributed with this paper. I will give the scientist who wrote the paper we rebutted credit, because after reading our papers, he retracted his. In fairness, since he retracted his paper, I have removed all references to him and his paper from this document.

This paper is essentially the 1993 paper with sections that have become self-evident eliminated and new updates added to bring it current. In addition, I refer the reader to additional works on the same subject which may be of interest. Two of these papers have been written by me, "Computer Viruses In UNIX Networks - 1995, 1996" and "Heterogeneous Computer Viruses In A Networked UNIX Environment - 1991, 1996". There are also many excellent papers written by Doctor Cohen, Tom Duff, M. Douglas McIlroy, N. Derek Arnold and Mark Ludwig on this subject. Most of these papers include working examples of UNIX viruses so there is no need to include working examples here. Refer to the Reference Materials section of this paper for details.

One final note about the applicability of the concepts contained this paper. Since this paper was first published in 1993, the world of operating systems has changed. There are now more operating systems that look like UNIX at the functionality level than existed at that time. I refer the reader specifically to the Linux and Microsoft NT operating systems. Everything contained in this paper that is valid for UNIX will also be valid at the concept level for these newer systems. These systems share more functionality than they are dissimilar.

Magical Immunity

The promotion of the concept of "magical immunity" to computer viral attacks surfaces on a regular basis. This concept, while desirable, is misleading and dangerous since it tends to mask a real threat. The latest paper to surface (removed by author) was published by a respected technical organization of great reputation. This paper asserted that UNIX and Amiga computer systems were immune to viral attacks because they made use of hardware instructions that provide a Supervisor mode of operation. Supervisor mode is a concept that requires access to restricted services in order to perform certain functions. It was implied that this mode imbued the operating system with protection. On the surface, this argument is academically stimulating, however, upon consideration, the argument becomes transparent and fails. The use of Supervisor mode is not necessary for viral infection, therefore the argument is moot. In addition, access to Super User Mode is easily obtainable through the many holes that are common in all operating systems. These facts are supported by the existence of viruses that infect the UNIX and Amiga systems.

ROM Based Operating Systems Do Not Provide Protection

In the paper "From Little Acorns Mighty Viruses Grow" by Alan Glover of Pineapple Software, it is disclosed that the Acorn Archimedes computer which holds all of its operating system and windowing systems locked in hardware based Read Only Memory has been successfully infected by computer viruses. This is an extreme case of hardware based protection and yet it failed. As of January 1994 there were 52 virus families totaling 84 viruses affecting the system. When compared to the Acorn computer, UNIX and NT, systems have very little chance of magical immunity.

Real Examples

Scholarly reports in separate papers by Tom Duff and M. Douglas McIlroy of AT&T Bell Laboratories contained in the USENIX 1989 Volume 2 journal not only attest to the existence of viral code for UNIX, but provide full source code for a few examples. These examples are provided in the Bourne shell script language, however, Mr. Duff also provided the information necessary for the infection of UNIX system binaries. The existence of these papers

in 1989 puts to an end, for all time, the plausibility that UNIX is, or ever has been, immune to viral attack.

Having disproved the immunity of UNIX to virus attacks by referencing known UNIX viruses, I turn the discussion to the virility of these attacks. Past experiments by Doctor Fred Cohen [1984] in which he used a UNIX system user account, without privileged access, yielded total security penetration in 30 minutes. Doctor Cohen repeated these results on many versions of UNIX, including AT&T Secure UNIX and over 20 commercial implementations. These results have been confirmed by independent researchers.

In McIlroy's paper, he attributes Highland [1988] with the statement, "Most computer programmers, aside from virus researchers, have ... difficulty in writing the code to make a virus replicate itself and secure itself to another disk." McIlroy then references Thompson [1984] that "Despite the claim, programs that reproduce themselves are not hard to make." This has also been my experience.

Operating System Components and Attack Payloads

Those components of an operating system that are deemed necessary for practical use, such as copy, append, change permission settings and hundreds of other basic functions are the only necessary building blocks for viral code. Many simple and normal functions that may pass a security screen, when combined, implement a virus. A simple example of a virus would be a program that located files, targeted hosts and then proceeded to infect them. This can be easily accomplished by "find / -type f -exec file{} \; | grep command | sed ---". The options for "sed" were withheld. A virus of this type could potentially carry a payload of "/bin/rm -rf / > /dev/null 2>&1". This payload can be set for a specific activation time and would be both silent and devastating. In fact, the recursive bin remove attack is the most common payload of virus, time bomb and Trojan Horse attacks in UNIX. Even in systems that are well protected, it is a common practice for users to have their own files unprotected, (permission setting 777 octal). If the remove attack was executed by a standard user account, without privilege, it will remove many of the user data files from the system. I suggest that the reader not experiment with this form of attack.

Script Viruses Are Simple

Many of the examples provided for UNIX viruses have been written in shell script. As proof that a relatively unsophisticated shell language can be easily used for writing virus code, Richard B. Levin published the source code to an MSDOS ".bat" virus in his book, "The Computer Virus Handbook", 1990 Osborne-McGraw-Hill. On page 9 of the book he demonstrated that a bat virus can be reduced to one line:

```
for %%fin (*.bat) do copy %%f + bf.v.bat
```

Virus Technology Is Easily Available

The simplicity of writing virus code is further aided by the existence of virus "cook books". Some of the books provide direction for the design, writing and implementation of computer viruses. One book by Mark Ludwig, "The Little Black Book of Computer Viruses", [1991 American Eagle Publications] contains full source code for sophisticated MSDOS executable viruses. The reader can also obtain the source, hex listings and compiled samples on diskette. A second book by Mark Ludwig, "The Giant Black Book of Computer Virus", [1995 American Eagle Publications] contains the source code for two UNIX companion viruses written in the C language. The book "UNIX Security, A Practical Tutorial" by N. Derek Arnold [1993 McGraw Hill] dedicated all of Chapter 13 to the explanation of viral activity under UNIX, including a working example in C language source code.

Information of this type is easy to obtain even from sources that do not intend to. The book "The PC Virus Control Handbook" by Robert V. Jacobson, [Second Edition 1990, Miller Freedman Publications] contains enough information about fighting virus infections to write a virus. All of the information, skills and techniques of virus writing is transferable between operating systems.

Productivity Tools Amplify Ability

Productivity tools that amplify a programmer's ability work equally well on constructive as well as destructive projects. Virus computer aided design and manufacturing programs V-CAD/CAM programs exist in the MSDOS environment. At least one V-CAD/CAM program is graphically enabled thereby allowing the user to select virus attributes using a mouse. Automated auditing and penetration testing (attack) programs have existed for many years in the UNIX environment, COPS, Tiger Script, SATAN, Root Kit and Crack. Since both systems are known to support hostile , it is only a small jump to understand that all V-CAD/CAM ability is portable as a working idea from MSDOS to UNIX.

All of these programs have been available via computer bulletin boards and at least one underground network (Nuke Net) for many years but has moved to the Internet with its new wide spread popularity and ease of use. It is not hard to locate a library of "hacker" tools on the Internet using any of the publicly available Internet search

engines. [This was predicted in the first printing of this paper with the statement, "The advent of Nuke Net will pale in significance once viral authors discover the Internet".]

All of this technology is applicable to UNIX and any other complex operating system such as Microsoft NT. In general, technology and ideas move from simple systems to complex systems. In this case, from MSDOS to UNIX and Microsoft NT.

Why Not More UNIX Attacks

In the paper, "Computer Virus Awareness for UNIX" NCSA News Volume 3, Issue 3 May/June 1992, I stated that the reason there have not been more UNIX attacks is because virus programmers could not afford the hardware necessary to execute the UNIX system. This is no longer true. UNIX is widely available at Universities, offices and libraries. The cost of used UNIX workstations such as Sun Microsystems Sparc 2 systems are selling at the same price as new PC based systems. In addition, the advent of Free BSD, BSDI, Linux and the newer lower cost versions of SCO UNIX and SCO UnixWare have made full function UNIX available on low-end PC systems. Due to the new popularity of the Internet, whose backbone and most of the servers are UNIX systems, it is no longer considered unusual to find UNIX systems in people's homes. The availability of UNIX, especially Linux, is now the same as or greater than any other system. Rarity will no longer provide any level of protection for UNIX.

A second reason that there have not been more UNIX attacks is that attacks that are made are not reported. Nothing inspires success as well as success and the lack of publicity authors of UNIX attack programs have received has had a beneficial damping effect. The two reasons that there has been very little publicity is because some of the organizations that track these attacks have made a policy not to report them hoping in effect to not fan the flames and because the media circus surrounding the Internet Worm and Michelangelo attacks has left the press gun shy. [Yes, even the press doesn't want to look foolish by being alarmist, although the nightly television news may convince you otherwise.] In addition, there is no reason to publicize anything but the most spectacular events such as the Internet Worm. Since then, there was a major European university infected with a UNIX script virus in 1992, rumors of a virus infection at a major American oil company in 1993, an international computer network using PC UNIX systems died in 1995 from the Michelangelo virus with a repeat performance in 1996. There were also many infections not worthy of special note and examples of the Typhoid Mary Syndrome that occurred in actual real world operation. None of this was reported to the general public, which while having a desirable and beneficial effect, also left many system administrators in the dark about the risks they may be facing.

What to Expect

The sophistication of computer viruses and virus programmers are increasing. There is no effective way to turn back the clock and legal measure will not help. Making the possession of viruses or other attack code illegal may, in fact, make dealing with the problem significantly more complex while removing useful penetration testing tools from the hands of legitimate users.

Complacency caused by a lack of understanding, publicity and a desire to not acknowledge problems that may exist in relation to the UNIX system will insure that when the next major incident occurs, it will be of global scale. The interconnection of the world's computer networks via the Internet will insure that no one is spared and that the entire event will occur worldwide before anyone knows that it has happened.

In the book, "Computer Security" by Ralph Roberts and Pamela Kane, the authors state that information is today's gold and that "the ultimate responsibility for protection of yourself and your property rests with you". Well said.

Postscript April 1996

It appears that everyone is from Missouri, the "show me" state. Very few people have bothered to follow the references given in this paper since its release in 1993. Consequently, people are no better educated about or prepared to deal with software based attacks than they were three years ago.

The reason that I did not give more explicit examples in the original paper was that it was an industry practice to not do so. I don't believe in this practice, but it was intended to fight the rumors that the anti-virus industry was creating viruses for it to fight. These rumors were ridiculous and appear to have all but disappeared. No one in the anti-virus industry has the time, energy or money to do so and no one is willing to take the risk, especially since the problem is already so large. For these reasons and because I believe that "Security Through Obscurity Is Insecurity" (for philosophical reference read: "Rudimentary Treatise on the Constructions of Locks", 1853 by Charles Tomlinson. Contained on page 144 of "Firewalls and Internet Security" by William R. Cheswick and Steven M. Bellovi.) I will now provide a nonfunctional, weak but educational example of code fragments that will allow the reader to understand the actions of a virus attack on a UNIX system. The reader should not make the mistake of believing that this example is the only method that such code may take because examples of code

found "in the wild" have used stronger algorithms.

In honor of Mr. Tomlinson whose philosophical treatise has improved my understanding of the world and because it is always convenient to name code fragments for future reference, this example is named the "1853 UNIX Example Virus".

Many parts of the UNIX operating system are written in script languages. It is therefore desirable to write viruses in a script language. An additional benefit of writing the attack in a script language is that script programs are portable between different manufacturer's systems while executable binaries are not. It is therefore necessary for an attacking script virus to identify other script programs as potential targets. This can be done using the command,

```
find / -type f -exec file {} \; |grep command |awk {print $2} .....
```

The first line of a program written in script normally controls which script language it executes in. This line appears as a comment if contained anywhere else in the body of the program. It is therefore necessary for an attacking virus to preserve the first line of the target program. This can be done using the "head -1 \$target > /tmp/trash" command.

Assuming the virus is the first nine lines of code following the first line of the program, then the virus can be extracted from the attacking host using the following code fragment.

```
head -10 $0 > /tmp/trash
```

```
tail -9 /tmp/trash > /tmp/trash2
```

The file "/tmp/trash2" now contains the virus body. To complete the attack and infect a target file, the code fragments may be assembled somewhat like this:

```
head -10 $0 > /tmp/trash
tail -9 /tmp/trash > /tmp/trash2
head -1 $target > /tmp/trash3
cat /tmp/trash3 /tmp/trash2 > /tmp/trash4
cat /tmp/trash4 > $target
/bin/rm -f /tmp/trash*
```

The results of infection will appear as following:

Original Target Code	Infected Code
#!/bin/sh	#!/bin/sh
[body of target program]	[virus body]
	#!/bin/s
	[body of target program]

Typical payloads such as a recursive bin remove (/bin/rm -rf / > dev/null 2>&1) or the insertion of a back door (cp /bin/sh /tmp/gotu ;chmod 4777 /tmp/gotu) can be carried in the body of the virus.

I hope that this illustration of the mechanical operation of a virus ends the discussion on the plausibility of UNIX viruses. UNIX viruses can and do exist. They have been found infecting sites "in the wild" and are not curiosity items.

Reference Materials

Reference materials are listed in date of publication. This is not a full or extensive reference but a resource guide for the reader who wishes to continue investigations into this subject.

Rudimentary Treatise on the Constructions of Locks 1853 by Charles Tomlinson Contained in "Firewalls and Internet Security" (see below)

Computer Security by Ralph Roberts and Pamela Kane 1989, Compute! Publications, Inc. ISBN 88-63151

Experience with Viruses on UNIX Systems by Tom Duff Spring 1989 Volume 2 Number 2, USENIX Computing Systems ISBN 0895-6340

Virology 101 by M. Douglas McIlroy Spring 1989 Volume 2 Number 2, USENIX Computing Systems ISBN 0895-6340

A Short Course on Computer Viruses by Doctor Frederick B. Cohen 1990, ASP Press, Inc. ISBN 1-878109-01-4

The Little Black Book of Computer Viruses by Mark Ludwig 1990, American Eagle Publications, Inc. ISBN 0-929408-02-0

The PC Virus Control Handbook by Robert V. Jacobean Second Edition 1990, Miller Freedman Publications ISBN 0-87930-194-5

Heterogeneous Computer Viruses In A Networked UNIX Environment by Peter Radatti 1991, 1996, CyberSoft, Inc.

Computer Virus Awareness for UNIX by Peter V. Radatti May/June 1992, NCSA News - Volume 3, Issue 3, Page 8

UNIX Security, A Practical Tutorial by N. Derek Arnold 1993, McGraw-Hill, Inc. ISBN 0-07-002560-6 {PBK}

Firewalls and Internet Security by William R. Cheswick and Steven M. Bellovin 1994, Addison-Wesley Professional Computing Series ISBN 0-201-63357-4

From Little Acorns Mighty Viruses Grow by Alan Glover, Pineapple Software February 1994, Virus Bulletin ISSN 0956-9979

Computer Viruses In UNIX Networks by Peter V. Radatti 1995, 1996, CyberSoft, Inc.

The Giant Black Book of Computer Viruses by Mark Ludwig 1995, American Eagle Publications, Inc. ISBN 0-929408-10-1

Papers by Doctor Fred Cohen Multiple dates, Available at <http://all.net>.

[Back](#)

[Home](#) | [Products](#) | [Support](#) | [Purchase Info](#) | [About Us](#) | [News](#) | [Contact](#)

Questions or comments about this website? Please contact the webmaster. 

© Copyright 2009 CyberSoft, Inc. All rights reserved.



This site certified 508 Compliant