# A Primer on Information Security Issues in an Academic Environment

Christopher Elliott
Warewolf Labs

celliott@warewolf.net
www.warewolf.net

## Table of Contents

# Foreword

This document is intended as a first step in enumerating the most commonly exploited vulnerabilities found in various operating systems that are likely to be found in an academic setting. It is not an exhaustive list of necessary precautions and practices, and should not be used to this end. Security is not a goal with a measurable endpoint, but rather an ongoing process requiring constant vigilance to maintain.

Furthermore, this information – while designed to address an audience with varying levels of experience with information security practices and terms – is targeted primarily as a summary for the technical staff in your departments to assist them in securing your environment.

# Introduction

System and network security in an academic environment provides additional challenges when compared to a similar business setting.

In a business environment, Internet access is typically protected at the perimeter by a series of choke points and firewalls that keep out unsolicited network traffic, filter viruses from e-mail attachments, and ultimately enforce corporate information security policies.

In contrast, one of the purposes of an academic network is often to provide open and unrestricted communication with the world at large. By virtue of this freedom, it is much more difficult to secure computer systems in this setting as the emphasis is on facilitating research and academic goals for the campus community. Once a system is compromised, valuable research and data can be lost, or hours can be tied up attempting to recover from the intrusion. Furthermore, a compromised system can be used as a stepping-stone to further attack other systems and to obscure the true identity of the attacker.

Additional hurdles exist in this challenge, in that security is often inconvenient, and is thus easy to overlook. Also, security must compete for limited budget resources, and may be regarded as too expensive to consider.

This document outlines several important yet inexpensive steps that can be taken to minimize well-known vulnerabilities and to decrease the overall attractiveness of systems in terms of being easy targets.

# General

## SANS Top Ten Vulnerabilities

The SANS Institute maintains a list of the top ten vulnerabilities[1] which account for a large portion of system compromises, and which are absolutely critical to fix. Specific information on the vulnerabilities – along with recommendations on how to patch them – is provided at the site referenced below.

## Remote Access

### Telnet vs. SSH

While telnet is a very popular tool for logging in remotely to system accounts, it sends the username and password, along with all other data, in a clear text (unencrypted) form over the network. Secure Shell (SSH) should be used in place of Telnet, with a minimum of the Data Encryption Standard (DES) cryptographic algorithm, and preferably the Triple DES (3DES) algorithm.

Additionally, Secure Copy (SCP) and/or Secure FTP (SFTP) should be used in place of FTP for transferring files.

### Virtual Private Networking (VPN)

If remote access to computing systems is needed (i.e. for users who are traveling or telecommuting from remote sites) the use of Virtual Private Network (VPN) solutions is advisable. This allows two benefits. First, a secure, encrypted connection is established between the remote user and the academic systems, so that no information is sent in the clear over the network (such as sensitive or confidential research information). Additionally, it ensures that only authorized users are able to remotely access systems, rather than having to open up a system to the Internet at large and hope that the defensive measures in place were enough to keep it from being compromised.

## Firewalls

A firewall can be a standalone device or specific software running on a system that monitors network traffic and acts as a first line of defense in protecting against unauthorized system access. Basic firewalls monitor all incoming information to a system or group of system and allow or restrict it as appropriate. More advanced firewalls also monitor traffic originating from the protected systems to ensure that it is authorized to communicate with the outside world. Several free or inexpensive software firewalls are available for both Unix and Windows platforms.

---

[1] http://www.sans.org/topten.htm

## Passwords

### Creation Guidelines

- In creating strong passwords, use a combination of letters (uppercase and lowercase, if possible), numbers, and symbols (i.e. %, #, {, etc.).
- Don't use words or phrases that would be found in a dictionary.
- Don't use information that is pertinent to the user (birthday or anniversary, dates, pet's names) – while the use of this type of information makes it less likely that a user will forget a password, it increases the likelihood of the password being compromised.

### Default Passwords

Change any default passwords that are automatically created with an operating system (such as SNMP community strings), or an application package (such as database server software)

### One-time Passwords

If it is not possible to use SSH and SCP/SFTP for a particular application, and if the operating environment supports it, consider instead the use of one-time password (OTP) mechanisms such as OPIE[2]. Since each password is only good for that interactive session, it prevents the compromise of a user account by eavesdropping on a connection and the subsequent re-use of that password at a later point in time.

## Auditing and Logging

- Audit your systems with a combination of port scanners such as nmap[3] and vulnerability scanners such as Nessus[4] and SAINT[5] on a regular basis.
- Monitor your system logs for unusual activity on a regular basis, either manually or through the use of automated tools such as Psionic Logcheck/LogSentry[6] (part of the Abacus Project by Psionic Software).

## Preventative Countermeasures

- Use binary authentication programs such as Tripwire[7] to ensure that key system files have not been replaced with modified Trojan horse versions that allow attackers additional points of entry into your system.
- Additionally, consider the use of programs such as Psionic PortSentry[8] to detect possible attacks from the Internet and proactively denies access to the system from the attacking address, and Psionic HostSentry[9] which attempts to monitor for suspicious user behavior and shut down accounts which may have been compromised.

---

[2] http://inner.net/opie
[3] http://www.insecure.org/nmap/index.html
[4] http://www.nessus.org
[5] http://www.wwdsi.com/saint/
[6] http://www.psionic.com/abacus/logcheck
[7] http://www.tripwire.com, free open source version at http://www.tripwire.org
[8] http://www.psionic.com/abacus/portsentry
[9] http://www.psionic.com/abacus/hostsentry

# Microsoft Windows 95/98/ME (Windows 9x)

There are two primary areas of concern with Windows 95/98/Millennium Edition (collectively referred to Windows 9x hereafter) peer-to-peer networks. The first is with shared network drives and folders, and the second is with Trojan horse software and viruses.

## Shared Drives / Folders

Access to shared folders in a Windows 9x Peer-to-Peer (as opposed to Client/Server) environment is determined not through a username/password combination, but by individual passwords specified for each shared folder or drive. Since Windows 9x only allows a maximum of eight characters for protecting a shared folder, it is especially critical that the password creation guidelines listed previously (mixed characters, numbers, and symbols) be put into motion. Additionally, keeping up to date on known vulnerabilities and operating system fixes from Microsoft via Windows Update (9) as well as vendor-specific application fixes is especially important.

## Trojan Horses / Viruses

Trojan horse programs and viruses are received most frequently through e-mail attachments or infected software downloaded from the Internet. In addition to utilizing anti-virus software and making sure that the virus definitions are updated on a regular basis, the use of a software firewall will help minimize the chance of damage being done from these malicious programs. The most effective software firewalls will not only examine and stop unsolicited inbound network traffic destined for a computer system, but will examine all traffic attempting to leave a computer system to ensure that it is authorized (as opposed to a Trojan horse program attempting to contact an attacker to alert them to a newly-compromised system).

An added benefit of a software firewall is that it can be configured to only allow access to a specific group of computers, so that the likelihood of the passwords on shared folders being compromised is greatly diminished.

## Microsoft Windows NT and 2000

Since Windows NT and 2000 allow remote interactive logons and increased capabilities, they provide more of a security challenge than Windows 9x-based systems.

- First, disable all services that are not required for the operation of a particular computer system.

- If Microsoft File and Printer Services are not needed on a particular machine, then consider disallowing access to TCP and UDP ports 135, 137 through 139, and 445 (on Windows 2000), or restrict access to these ports to authorized local systems. Ensure that the "Enable NetBIOS over TCP/IP" option is unchecked for computers that do not need this service.

- Implement operating system hardening procedures such as those documented at the SANS Institute and elsewhere to help lock down the system and prevent exploitable vulnerabilities.

- Mandate the use of strong passwords for both user and administrative accounts, and change them regularly.

- Set up system security logs to record failed username/password attempts, and audit your logs regularly.

- Stay current with all Microsoft service packs and hotfixes[10] and Microsoft Security advisories[11], as well as advisories and vulnerability reports from sources such as NTBugtraq[12] and SecurityFocus[13].

- Consider using a software firewall and anti-virus software for the various reasons listed in the previous Windows 9x section.

[10] http://windowsupdate.microsoft.com
[11] http://www.microsoft.com/security and http://www.microsoft.com/technet/security/default.asp
[12] http://www.ntbugtraq.com
[13] http://www.securityfocus.com

## Unix / Linux & Variants

For the purposes of this paper, Solaris/SunOS, Linux, IRIX, HP-UX, AIX, other Unix variants not specifically mentioned, and most of the BSD Distributions will be collectively referred to as a Unix-based system.

These systems provide a very enticing target to outside attackers, due in part to their flexibility and power, and also due to the fact that in general the default installation of most of these operating systems is highly insecure and vulnerable to attack.

- As a starting point, disable all services that are not explicitly required for the operation of that a particular system (especially BIND/DNS, FTP, NFS, RPC, SMTP, and X Window services).

- Additionally, lock down the system to minimize its vulnerability to attack by following OS hardening guidelines available from the SANS Institute and various other sources, prior to exposing it to the Internet.

- Auditing your systems (with both port and vulnerability scanners as well as binary authentication packages as discussed previously) on a regular basis is especially important.

- Finally, keep up to date on all operating system, equipment, and application vendor advisories, as well as recent vulnerabilities and exploits found at SecurityFocus[14], and Whitehats[15].

If it is at all feasible to your specific application or needs, consider deploying a pre-hardened version of Linux (such as Immunix[16], EnGarde SL[17], or the NSA's SE Linux[18]) or OpenBSD[19] to help minimize vulnerabilities and the overall attractiveness of your system as a target to outside attackers.

[14] http://www.securityfocus.com
[15] http://www.whitehats.com
[16] http://www.immunix.org
[17] http://www.engardelinux.org
[18] http://www.nsa.gov/selinux/index.html
[19] http://www.openbsd.org

## Additional Resources

In addition to the references mentioned through this article, the following additional materials and sites contain useful information.

### Printed

Scambray, Joel, McClure, Stuart, and George Kurtz. Hacking Exposed: Network Security Secrets and Solutions, 2nd Edition. Berkeley, CA: Osborne / McGraw-Hill, 2001.

### Online

The CERT Coordination Center at Carnegie Mellon University has a wide variety of information available on system vulnerabilities and advisories, and can be found at http://www.cert.org.

The National Infrastructure Protection Center (NIPC) provides information on international threats, as well as security advisories and updates, and can be found at http://www.nipc.gov.

The ICAT Metabase is a searchable listing of vulnerabilities in a wide variety of categories (such as operating systems, application programs, hardware devices, etc.) that provides an easy method for auditing a wide range of devices for known issues from a single site. The ICAT Metabase is found at http://icat.nist.gov/icat.cfm.

## Acknowledgements

I am grateful to both Brian Flucht and John Elliott for generously contributing their time and expertise in providing feedback and helpful discussions.

## About the author

Christopher Elliott is the founder and principal member of Warewolf Labs. Started in 1997, Warewolf Labs has been providing information security services to both the private and public sector for the past several years. He can be reached at celliott@warewolf.net.