Global Information Assurance Certification Paper

Public Domain FTP Buffer Overflow
Vulnerabilities Feb. - Oct. 1999.

Ralph Durkee
June 14, 2000

Summary:

Multiple buffer overflow vulnerability exists in older (1999) versions of popular public domain FTP Daemon software which may be exploited to run arbitrary code on the server under the user privileges used to run the ftpd, which is usually root. Specifically CERT® Advisories CA-99-03-FTP-Buffer-Overflows and CA-99-13 Multiple Vulnerabilities in WU-FTPD describes the risks and platforms affected. Although WU-FTPD (originally from Washington University) is specifically mentioned, other public domain ftp server derivatives of WU-FTPD, including ProFTPD are vulnerable as well.

Recommendation:

Vulnerable software should be disabled, and replaced with newer versions. The current (as of June 2000) recommended version of WU-FTPD is 2.6.0 released Oct. 1999. Please check the WU-FTPD support site http://www.wu-ftpd.org/ for up-to-date security and version information and downloads. The recommended version of ProFTPD is at least 1.2.0pre9, while a newer version 1.2.0pre10 is also available from http://www.proftpd.net/ Red Hat and many other Linux systems include WU-FTPD by default, while most commercial UNIX systems were not vulnerable to this specific exploit. Read the CERT Advisory and check with your system software vendor or freeware provider to be sure.

Checking Vulnerability:

The version of an ftpd software can be checked by running the command with the appropriate option. The WU-FTPD takes an uppercase -V option to display the version information, while ProFTPD will display version information with a lowercase -v option. With default configuration settings these FTP daemons will also announce their name and version when serving a connection prior to login. You should also check the /etc/inetd.conf file or the /etc/rc* files or to be sure of the correct path to the ftpd which is actually in use.

Several potential overflow bugs were fixed in code reviews according to the WU-FTPD change log . This specific exploit requires connecting to the ftp server as an anonymous or as a regular user, and requires access to any writeable directory. Obviously, ftp servers which allows anonymous connections should be chroot'ed and not be allowed to put files in any directory. Many ftp servers used for business-to-business or home-to-work file transfers as well as most ISP's allow authenticated users to write in assigned directories. In these situation there would be significant usage of WU-FTPD and ProFTPD where incoming transfers must be allowed for authenticated users. These situations would be vulnerable to exploitation of root privileges from legitimate but possibly untrusted users, or from others if the user passwords were sniffed, stolen or guessed.

A realpath() Buffer Overflow Exploit:

An exploit posted by Mixter to bugtraq on 1 May 1999 with comments "Coded by smiler and cossack" involves creating a very long path of directories within directories using nested long directory names of 194 characters each. The length of the total path is calculated specifically to overflow the buffer which holds the path string. One final directory is created (and possibly deleted) but the name of the last directory actually includes binary precompiled exploit code which will spawn a shell when executed. The exploit shell code will be executed if the buffer overflows allowing the return address to be overwritten with the location of the shell code. The

overwriting of the return address redirects the flow of execution to the exploit code (sent as data) instead of returning to the calling function. If the exploit succeeds, the cracker will have a socket connection to a shell spawned by the ftpd and running with the same privileges as ftpd (usually root). A directory created by the exploit will look somewhat like the following with random uppercase characters used. The ellipses (...) represent repetitions removed. In this case the system created the final directory name containing the precompiled code without a buffer overflow. The leading bytes in the final directory will be no-ops instructions suitable for the platform, followed by the code to execute a shell process.

AAA...AAAA/VVV...VVVV/DDD...DDDD/SSS...SSSS/AAA...AAAA/???...???1Û?Ø°?Í?ëf^?ó?Ã?9ó|??+?þ
...

Security and public domain software:

If public domain FTP server software is less secure than commercial ftp daemons, then why use the public domain FTP servers such as WU-FTPD and ProFTPD? The WU-FTPD and ProFTPD provide additional functionality, such as anonymous style chroot for real user logins. Allowing untrusted users chroot-ed ftp access to their own home directory while restricting access to the rest of the system is important for a number of applications such as ISP's with shared hosting servers. Buffer overflow vulnerabilities have been problematic across the board, and are not unique to public domain software. Having the source code available to the crackers may be seen as a liability to some, but also enhances the the security of the software in the long run. In either case the security benefits and liabilities of open source is a topic which will have to be left for another discussion.

General Prevention:

Upgrading your ftpd software, if you were using a vulnerable version, will prevent this exploit, and several others that were found, but what else can be done to protect your system from other yet-to-be-announced buffer overflow vulnerabilities? Other than good programming practices, there are no silver bullets yet for buffer overflows on most systems, however basic best-practices in network security will provide necessary redundancy, augmenting each other in ensuring the security of the systems. There are a few system architectures which enforce stack pro Specifically, keeping up-to-date with security announcements such as http://www.SANS.org as well security notices from your system and software vendors or freeware developers is a good start for defense. Intrusion detection systems or software along with daily auditing and monitoring of the normal system usage will provide an additional line of defense. There are also projects in the works for developing tools to protect or detect a stack corrupted by a buffer overflow. Such projects include

Stack Shield http://www.angelfire.com/sk/stackshield/
Stack Guard http://www.cse.ogi.edu/DISC/projects/immunix/StackGuard

There are criticisms from some on the viability of these stack protection techniques. Certainly the efforts are to be critiqued and applauded accordingly.

Sources:

CERT® Advisory CA-99-03 FTP Buffer Overflows, http://www.cert.org/advisories/CA-99-03-FTP-Buffer-Overflows.html (June 13 2000)

CERT® Advisory CA-99-13 Multiple Vulnerabilities in WU-FTPD, http://www.cert.org/advisories/CA-99-13-wuftpd.html (June 13 2000)

Aleph One, "Smashing The Stack For Fun and Profit", Phrack Magazine (Vol VII, Issue 49), http://phrack.infonexus.com/search.phtml?view&article=p49-14, (June 14, 2000)

WU-FTPD Development Group, "Change History", http://www.wu-ftpd.org/ ftp://ftp.wu-ftpd.org/pub/wu-ftpd/CHANGES, (June 13, 2000)

Smiler and Cossack, posted by Mixter, "wuftp2.4.2academ beta 12-18 exploit", BugTraq e-mail Archive - 1 May 1999, http://www.securityfocus.com/templates/archive.pike?list=1&date=1999-05-1&msg=Pine.LNX.4.04.9905012157010.543-100000@aviation.net (June 14, 2000)

Vendicator (vendicator@usa.net), "Stack Shield - A "stack smashing" technique protection tool for Linux", http://www.angelfire.com/SK/stackshield/ (June 16, 2000)

Immunix Project at Oregon Graduate Institute of Sci. and Tech., "StackGuard: Protecting Systems From Stack Smashing Attacks", http://www.cse.ogi.edu/DISC/projects/immunix/StackGuard/ (June 16, 2000)