Global Information Assurance Certification Paper

Security Implications of Update Agent Software
Shaun Glaim
October 5, 2000

Introduction

Many of today's programs employ methods to keep their code up to date. How do the update methods used by these programs ensure code validity, and what sort of vulnerabilities exist that could be exploited. This paper will look at the different methods used by some of today's more popular programs and examine some items that may give cause for concern.

Background

In the past, software was upgraded by the local support personnel and generally delivered on media. Most times the source of the upgrade was known to be valid in origin, and presented little in the way of a security threat to the end user, and the corporate network.

In today's digital world however there is an increasing use of automatic upgrade programs to patch software, deliver new software packages, and update operating systems components. Most end users will generally trust that the files they are receiving via these update agents have not been corrupted, and that they come from the source that is claiming to deliver them.

However, as this may not necessarily be the case, most of the update programs in use today utilize some form of data encryption or hash to create a signature for the software that is being updated This is done in order to provide an additional form of package validation

In some cases the upgrade agent also utilizes a secure download communication channel as a method of source verification.

Methodology

What are the risks inherent in today's auto update programs?

In order to highlight some of the methods that are used today, and their risks, I will use the following procedure:

> Outline how the update agent performs the download of its software packages. Examine the method that is used by the agent for package verification. Once that has been established I will provide a known example, if one exists, of how the agent may be vulnerable. I will then examine what sort of risk this may pose.

In order to do this I have opted to examine two well know anti-virus products. I will also examine the methods used by two of today's Operating Systems: Microsoft's Windows Update, and Linux's RPM method.

Specific examples:

Norton Live Update

Method Used:

> Norton's Live Update can use two methods to download its packages: Http and FTP (as a fallback method). The retrieval location of the updates can be controlled from the users workstation, and can be set to get updates from locations other than Symantec's servers. The update program verifies downloaded packages by checking the digital signature of the update. This includes checking to see that the name of the downloaded package is reflected in the digital signature. In the case of Norton's tools, the filenames change with every update of the software and usually include the month and day that the package was released.

Known vulnerabilities:

> None came to light during research.

Security implications:

If a way can be found to emulate the digital signature of the downloaded packages, it would be possible for an attacker to customize the location from which the download would take place. This would in effect allow for a rapid deployment of any such trojaned files.

McAfee AutoUpdate

Method Used:

McAfee's AutoUpdate uses two methods to download its packages: FTP and local Network shares. The retrieval location of the updates can be controlled from the users workstation, and can be set to get updates from locations other than McAfee's servers. The download location is controlled via a registry key in Windows, and is easy to change.

The update program, which checks for the digital signature inside a known .ini file, verifies downloaded packages.

Known vulnerabilities:

Testing of Netshield 4.5 and VirusScan 4.5 indicated that both products were shown to have a potential security hole in the AutoUpgrade portion of the product.

As Richard Fry pointed out in his July 11, 2000 email to the NTBugtraq list:

> "If the directory pointed to by the mechanism (controlled via a registry key) is insecure and has the correct format (PKGDESC.INI, SETUP.ISS included) any file, which is placed there with the name of SETUP.EXE, will be run in local administrator context."

Security implications:

In this case a Trojan program could be renamed as SETUP.EXE, and would be executed with full privileges. This type of hole has very serious implications for the security of any system that is using this product. Any type of malicious program could be installed and, in the worst cases, allows the attacker to gain a foothold into the System, and the networks that it is attached to.

Windows Update

Method Used:

When you log on to Microsoft's Windows Updates page, an HTTPS connection is established to the site. During this process the windows update site presents it's digital certificate for validation. Once the site has been verified, an ActiveX control is downloaded and the machine is scanned to see what update packages may be needed. All of the packages have been signed with Microsoft's Authenticode technology. Authenticode consists of a hash algorithm, an encrypted key specification, and a site certificate specification.

Known vulnerabilities:

There have not been any documented cases of vulnerabilities within the Windows Update Service. However, since this service utilizes ActiveX and Java Script in order to scan for updates, it may be possible to take advantage of holes in this software.

ActiveX has been demonstrated to have many security holes and in the case of the "Exploder Control", it was shown that even an Authenticode signed control could still be malicious.

Security implications:

Although no know holes exist in the Windows Update service there are known holes in the underlying technologies being used. It may be possible to leverage these services as part of an attack.

RPM Update

Method Used:

RPM update is used by various Linux platforms to install updates. In most cases, packages are downloaded from a mirror site via FTP; however, the site that the download comes from may not be a secure or valid site. Therefore when the RPM packages are created, they are given an MD5 checksum and the contents of the RPM are listed as part of the header. After the packages have been downloaded, they can be checked for authenticity by validating their checksum and viewing the package list.

In addition, the Security teams from some of the Linux companies are utilizing a PGP key and they are signing RPM packages to further authenticate that the packages are valid and from a proper originating source.

Mandrake Update and Redhat Update Agent are two specific examples of these types of agents. Each handles the update process in a slightly different manner.

When invoked, Mandrake Update shows what packages are available from a mirror site that is selected by the operator. The update packages are then downloaded. Next, a check is done to see if the current operator running the upgrade program is the root operator. If they are not, they are then prompted to login as root to complete the upgrades.

When run, Redhats Update Agent checks to see if the current operator asking for the updates is the root operator. If this validation completes, the agent then connects to a secure known site where the RPM packages are stored and the selection and download process is started.

Known vulnerabilities:

For this type of upgrade agent, the process is controlled as only the root operator is allowed to install package updates (on a properly configured system). However, there are exploits of the RPM agent itself that utilize buffer overflows to sieze elevated access on the machine.

Security implications:

A great amount of security consideration was given to this type of design. Packages are given a digital signature and a Public Key, and the package contents are listed. Some of the distributions even utilize secure sites for downloading the packages. However, this process still relies upon a properly configured machine and a knowledgeable systems administrator to check the validity of the downloaded packages.

Conclusions

Most automatic upgrade software is not designed with security foremost in mind. In fact, all of the upgrade agents still rely on the end user to know what is taking place.

Even if the process of doing the upgrade has been designed with security in mind the underlying technologies that are being used to deliver and verify the content may not be secure.

Although auto-updating software provides convenience there continues to be a trade off with regards to security concerns.

Ensuring that the content is not corrupt, and that the security of a system is intact, still requires a knowledgeable person to manage the update process.

Resources Used

Symantec. " Symantec Support Site."
URL: http://www.symantec.com/techsupp/index.html

Symantec. Norton Antivirus 2000 Documentation.
URL: http://www.symantec.com/techsupp/files/nav/nav2001.html

Mcaffee. " How to update your DAT files", Mcaffee online help manual
URL: http://www.nai.com/asp_set/services/technical_support/docs.asp?pCode=NSHIN

Microsoft. "About Windows Update" Microsoft Windows Update.
URL: http://windowsupdate.microsoft.com/R553/V31site/x86/nt5/en/Ie5/about.htm

Microsoft. " Frequently Asked Questions About Authenticode."
URL: http://msdn.microsoft.com/workshop/security/authcode/signfaq.asp

Microsoft. " Security – Certificate and Authenticode."
URL: http://www.microsoft.com/technet/security/authtech.asp

RedHat Linux, Working with Update Agent, **The Official Red Hat Linux Reference Guide**
URL: http://www.redhat.com/support/manuals/RHL-6.2-Manual/ref-guide/ch-up2date.html

RedHat Linux, Package Management with RPM,**The Official Red Hat Linux Reference Guide**
URL:http://www.redhat.com/support/manuals/RHL-6.2-Manual/ref-guide/ch-rpm.html

SANS, **ActiveX Controls - Risk and Control,** Kam Ng
URL: http://www.sans.org/infosecFAQ/activex.htm

Mandrake Linux, MandrakeSoft Security Advisory
URL: http://www.linux-mandrake.com/en/security/MDKSA-2000-034.php3

NTBugtraq, "Potential Vulnerability in McAfee Netshield and VirusScan 4.5" Richard Fry
URL: http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0007&L=ntbugtraq&O=D&P=2753