



Patching Placebo

By Mitchell Ashley, CTO and VP of Customer Experience at StillSecure

January, 2006

It has become a regular, reoccurring event; patching our servers, desktops and even the network infrastructure on a monthly and sometimes even daily basis.

Microsoft Tuesday or Black Tuesday as many have come to call it, is at the top of the list for corporate IT organizations. It has even become so common place that many just refer to it as "patch Tuesday." Whether you follow rigorous configuration management and change control processes or you simply rely on Windows Update to update servers and desktops with current security fixes, patch Tuesday is a corporate IT fact of life. In addition to Microsoft, many other major software companies such as Oracle have adopted use of the monthly patch release process.

There certainly are advantages to the process of a monthly scheduled release of patches from major vendors. Security patches are rated by vendors and security organizations as to the security risk posed to users of the software. Now organizations regularly schedule their change control processes around these monthly releases so they can adequately test new updates and release them into production as change control processes allow.

While we've all grown accustomed to weekly and sometimes daily announcements of vulnerabilities there are also some disadvantages to relying on vendor released patches. Patch releases usually don't cover all current security vulnerabilities, and patches from the major software vendors don't necessarily mean that there are not other ways to exploit these vulnerabilities. In this article we will examine examples where just relying on vendor patching strategies can still present a security threat and methods for balancing these risks.

Vulnerability or Misuse?

In August of 2005, Igor Fanchuck discovered a condition in the Windows Registry that would allow probable attackers to hide malicious software through the use of an overly long value name within the registry. Fanchuck then reported it to security researchers at Secunia. The Windows Registry Concealment vulnerability was rated as not critical by Secunia and low risk by the French Security Incident Response (FrSIRT) at the time of its discovery.

This vulnerability is not actually a security flaw in the Windows Registry or the operating system but a flaw in software programs that scan and edit the Windows registry. This is true of both Windows REGEDIT utility and some other third party registry and security scanner tools. This caused many to see the problem not as a security vulnerability, but rather as a capability in the operating system that could be misused by someone. That's a very fine line of distinction.

Some registry editors and scanners possess a flaw in their programming that causes them to believe they have reached the end of that portion of the registry when they encounter a registry

key longer than 255 bytes. While it is not 'illegal' to put very long entries in the registry, (this is something Windows allows for), some software wasn't built to expect it. In this situation, hidden values can be placed after the long registry value name which would not be noticed by registry scanners and editors. Why is this an issue? Autorun registry keys directing Windows to run an executable at start up can be placed after the long registry value name, concealing the run key from view by the administrator and registry viewing software.

The SANS Internet Storm Center helped coordinate the investigation of the Registry Key Concealment vulnerability, requesting feedback from security researchers about susceptible anti-virus, anti-spyware, registry scanners, editors and vulnerability testing tools. The ISC also conducted in-house testing of the vulnerability along with end-user education. Robert Danford a StillSecure Security Researcher recommended that everyone watch for product updates when the problem became known. "Security is about cat and mouse. Many cleaning tools rely heavily on the registry keys. The hidden keys can prevent detection and hinder cleanup" said Danford. For the complete ISC diary entry please link to: <http://isc.sans.org/diary.php?date=2005-08-25>.

Since the discovery of the Registry Key Concealment Vulnerability, malware using this specific exploit have been seen in the wild. One example is a vulnerability discovered on April 22, 2005 called [Backdoor.Ripgof](#). While no wide-spread infections of malware using this technique/strategy have been seen to-date, one incidence of a keystroke logger missed by anti-virus software on the wrong machine within the organization is all it takes for this to jump from a low rating to a very critical issue.

Security fixes to all security tools won't come from Microsoft, they need to be supplied by the individual software makers. Even though the threat may be rated as low, IT security professionals must still track and repair these types of vulnerabilities to make sure they are receiving an accurate picture of the security posture of their network.

Is this the next worm to sneak under the wire and infiltrate networks? It's not likely to spread massively since it isn't a vulnerability directly accessible via the network, but it very easily could be used as a secondary method to leave behind a Trojan to later be used to spread, capture sensitive information or damage devices on the network.

Fix Date Unknown

Even when a security vulnerability is contained within the realm of one software manufacture, it doesn't mean the problem will be fixed right away. Part of what every software vendor does is perform triage on newly reported vulnerabilities; assessing their potential damage, how easily they can spread, do they require actions by an end user to become effective or proliferate, and is there code in the wild utilizing this exploit.

Every vendor goes through this process but the most visible are the major software manufactures. This means that all of the security vulnerabilities in commonly used software may not get fixed right away, and you may wait weeks or months before a fix is provided. In fact there are thousands of undiscovered/unreported vulnerabilities.

A recent example of this is the Microsoft Jet Database Engine Malformed Database File Buffer Overflow vulnerability. According to *Security Focus*, the Microsoft Jet Database Engine can be compromised due to a buffer overflow situation, which is due failure in the software for not properly checking the bounds of user-supplied database file contents. If exploited this

vulnerability could be used to execute arbitrary machine code when attempting us use a malicious Jet database file. The vulnerability is reported to exist in the 'msjet40.dll' library, version 4.00.8618.0 and may possibly affect older versions. More information on this vulnerability can be found at <http://www.securityfocus.com/bid/12960>.

The vulnerability was reported in March 2005. Since then many more malware programs have appeared using this exploit. The first was discovered on April 19th; a Trojan horse called [Backdoor.Ryejet](#).

The Trojan can deceptively hide traces of itself as a rootkit on the compromised machine and hide the presence of its files on the compromised computer. It can be distributed to other machines since the Trojan embeds itself in a Microsoft Jet Database.

Recently, another Trojan horse utilizing this exploit was discovered. The [Backdoor.Hesive](#) (Bugtraq ID 12960) is a Trojan horse that opens a backdoor on the compromised device and gives remote attackers unauthorized access. Again, this is made possible by an unpatched vulnerability in the Microsoft Jet Database.

Release of a security patch for this vulnerability is unknown. While this vulnerability has not led to a widespread infection, it still is a vulnerability which can lead to a serious security incident. This situation demonstrates that just relying on patches means there has to be a patch available or forthcoming. We've used a Microsoft vulnerability for this example but this could have easily been many other software vendors. Without a patch what do we do in the meantime?

Good Security Is Still Good Security

It would be all too easy to fall back on vendor patches as a primary means to manage security risks. While this certainly is important to ultimately resolving the core security vulnerability in software, not every problem can be solved via a patch from Microsoft, Oracle, or another major software supplier.

As we've seen, security risks can fall into that grey area of 'misuse' or not actually a true vulnerability in the operating system. Software not directly part of the operating system and administration tools themselves can aid in masking problems. Security tools themselves can also contain flaws that may mask or hide existing security issues.

Patching is not an end in itself but must be a part of an overall security program. Security best practices recommend following a layered approach to security, one that provides for checks and balances and does not rely on any single method or tool for maintaining security. A good way to define layered security is a security strategy that has defensive, proactive and compliance elements.

Beyond just relying on patches to solve all software security issues, a well executed and comprehensive vulnerability management program gives the security team a proactive view into security vulnerabilities across the spectrum; those that can be patched, those that require configuration or change management, and those that must be managed until a patch becomes available.

Despite what the security and patching vendors might tell you, vulnerability management is not a technology but an organizational process that is driven by the security team. Assessing the

organization's vulnerability to exploits and determining if the proper security remediation steps (firewall policies, endpoint security, network access control, patch application, and verification testing) are put into place are crucial.

Single vulnerabilities are typically the easiest to manage. Change the devices configuration or apply a software fix and the vulnerability is taken care of. Blended threats, such as those vulnerabilities that may not be directly exploitable until the attacker has direct access to device, must be closely managed as well. These blended threats can be even more dangerous because the immediate impact may not be apparent. The initial compromise of a device may leave behind dormant code to be executed at some later time, even after the device has been remediated.

Tracking available vulnerabilities allows the security team to determine interim changes, such as the access control list (ACL) on a router could effectively prevent or limit the effects of an un-patchable vulnerability.

A truly effective vulnerability management program must also account for devices outside the purview of the IT organization. These may be servers that are managed and maintained by non-IT departments, such as when someone in finance or radiology manages their own files servers and applications. Vendor equipment running commonly available operating systems and file server software can also pose the risk of compromise and become the launch pad for attacks inside the network perimeter.

Network access control (NAC) technologies can add an addition layer of security to a good vulnerability management program. NAC can ensure that end user devices connecting to the network are quarantined before being allowed to access full network resources. Endpoint devices can be checked for the up-to-date security patches, anti-virus and threatening software such as spyware, P2P, and messaging programs. This is especially valuable for assessing the security posture of unmanaged endpoint devices, those whose security is not managed by the internal security organization. Unmanaged or foreign endpoints are usually thought of as the computer of a visitor, contractor, or a work-at-home employee.

Vulnerability management and endpoint network access control are excellent additions to traditional network security approaches such as perimeter firewalls and intrusion detection systems. Managing the gaps, which are the less known vulnerabilities that don't exactly make the front pages of the latest IT trade magazine, becomes more palatable when automated vulnerability management processes can drive the patching processes. This allows security teams to make good decisions about where gap engineering should be performed and how best to protect the network when patches are not immediately available. Recognizing that patching doesn't solve all of our software security vulnerabilities is the first step understanding where latent threats may exist and how best to mitigate the risks they pose.

About the Author

Mitchell Ashley is CTO and VP of Customer Experience at StillSecure where he is responsible for the product strategy and development of the StillSecure suite of network security products. Mr. Ashley has more than 20 years of industry experience holding leading positions in data networking, network security, and software product and services development. Mr. Ashley can be reached at mashley@stillsecure.com or 303-381-3830.