Why Vulnerabilities Go Unpatched

Lori MacVittie, 2008-19-06

The good folks at Verizon Business who recently released their 2008 Data Breach Investigations Report sounded almost surprised by the discovery that "*Intrusion attempts targeted the application layer more than the operating system and less than a quarter of attacks exploited vulnerabilities. Ninety percent of known vulnerabilities exploited by these attacks had patches available for at least six months prior to the breach.*"

This led the researchers to conclude that "*For the overwhelming majority of attacks exploiting known vulnerabilities, the patch had been available for months prior to the breach. [...] Also worthy of mention is that no breaches were caused by exploits of vulnerabilities patched within a month or less of the attack. This strongly suggests that a patch deployment strategy focusing on coverage and consistency is far more effective at preventing data breaches than "fire drills" attempting to patch particular systems as soon as patches are released.*"

There's actually a very valid reason why vulnerabilities go unpatched for months in an organization, regardless of how frustrating that reality may be to security professionals: reliability and stability.

The first rule of IT is "Business critical applications and systems shall not be disturbed." When applications are the means through which your business runs, i.e. it generates revenue, you are very careful about disturbing the status quo because even the smallest mistake can lead to downtime, which in turn results in lost revenue. For example, it's estimated that Amazon lost $31,000 per minute it was down. It was down long enough for the revenue lost to jump into six digit figures.

Patching a system requires testing and certification. The operating system or application must be patched, and then all applications running on that system must be tested to ensure that the patch has not affected them in any way. IT folks have been burned one too many times in the past by patches to simply "fire and forget" when it comes to changing operating systems, platforms, and applications in production environments. That's why multiple environs exist in the enterprise - an application moves from development to quality assurance and testing to production, and why we've all sat around eating pizza and watching the clock late on a Saturday night, holding a printed copy of our "roll back" plans just in case something went wrong.

Patching even a vulnerability takes time, and the more applications running on a system the more time it takes, because each one has to be tested and re-certified in a non-production environment ***before*** the patch can be applied to a production system. This process evolved to minimize the impact on production systems and reduce system downtime - which usually translates into lost revenue.

Thus, it's no surprise that Verizon's researchers discovered such a high percentage of vulnerability exploits could have been prevented by patches issued months prior to the breach. It's possible that many of those breaches occurred while the patch was being tested and simply hadn't been rolled out to production yet.

It's certainly not that IT professionals are unaware that these patches exist. It's simply that there are so many moving parts on a production system with higher risk factors than your average system that they aren't willing - many would say rightfully so - to apply patches that may potentially break a smooth running system.

This is one of the reasons we advocate the use of web application firewalls and intelligent application delivery networks. The web application firewall will usually be updated to defend against a known web application vulnerability before the IT folks have had a chance to verify that the patch issued will not negatively affect production systems. This provides IT with the ability to take the time necessary to ensure business continuity in the face of patching systems without leaving systems vulnerable to compromise. And if the web application firewall isn't immediately updated, an intelligent application delivery platform can often be used as a stop-gap measure, filtering requests such that attempts to exploit new vulnerabilities can be stopped while IT gets ready to patch systems.

An application delivery platform can also, through its load balancing capabilities, enable IT to patch individual systems without downtime by intelligently directing requests to the systems available that are not being patched at the moment.

IT Security Professionals understandably desire patches to be applied as soon as possible, but the definition of as soon as possible is often days, weeks, or even months from the time the patch is issued. IT and business folks don't want to experience a breach, either, but the need to protect applications and systems from exploitation must be balanced against revenue generation and productivity. Employing a web application firewall and/or an intelligent application delivery network solution can bridge the gap between the two seemingly diametrically opposed needs, providing security while ensuring IT has the time to properly test patches.

Imbibing: Coffee

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com