National Security Agency | Cybersecurity Information

Update and Upgrade Software Immediately

Small Delay, Big Issue

Enterprises that do not take advantage of software updates and upgrades leave themselves needlessly vulnerable. Malicious actors rapidly develop exploits by analyzing and reverse engineering software patches, and these exploits are often available days after patch release [1], [2]. Updating software as soon as possible is the only way to address this threat.

Furthermore, effective new security technologies arrive only in major software upgrades. New security technologies block entire classes of exploitation techniques, and industry continually introduces new features that make exploitation more difficult. Software upgrades and timely deployment of patches are both critical to network defense [3].

Why Patch?

The Common Vulnerabilities and Exposures (CVE) database at http://cve.mitre.org demonstrates the sheer volume of vulnerabilities that are publicly reported by the cybersecurity industry. The true volume of vulnerabilities patched is even greater because many updates contain security fixes that are never captured in a CVE or otherwise documented publicly. Responsible enterprises – and malicious adversaries – act on this information.

Some enterprises delay or ignore patches due to the fear that patches may break or slow down applications. But vendors perform significant testing of these patches prior to the deployment of updates in order to ensure disruption remains rare [4]. For those enterprises with residual concerns about applying updates in their environment, the costs of pre-deployment testing are miniscule compared to the devastating costs incurred from a security breach.

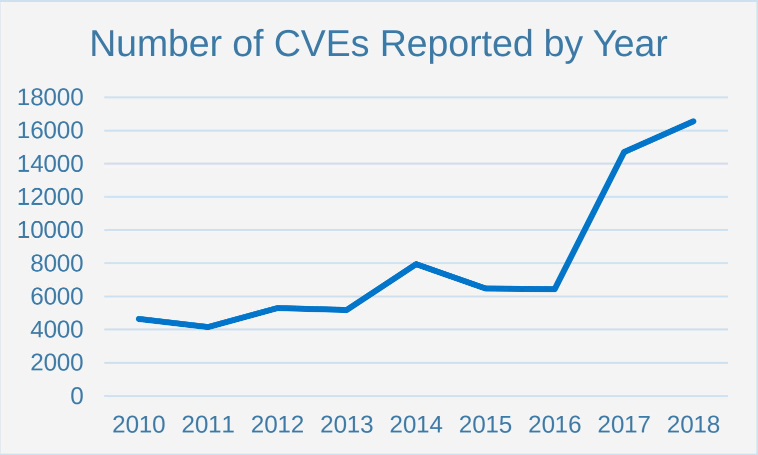

Figure 1: CVEs by Year, https://www.cvedetails.com/browse-by-date.php

Other enterprises pick and choose which patches to apply based on vulnerability databases. While these databases are informative, they are not sufficiently comprehensive to be the sole foundation for a patch management strategy. Enterprises that choose to apply only specific patches based on CVE lists remain exposed to known vulnerabilities. Applying all vendor patches is the best patch management strategy available for enterprises to secure their networks.

Patching for operating systems and applications is becoming increasingly streamlined, further decreasing the costs of deploying updates. Many operating systems and applications, including Apple iOS®[1] and macOS®, Microsoft Windows®[2], Red Hat®[3] and other Linux®[4] distributions, Google Chrome™[5], and Mozilla Firefox®[6] provide automated update features to

[1] iOS and macOS are registered trademarks of Apple.

[2] Microsoft Windows is a registered trademark of Microsoft.

[3] Red Hat is a registered trademark of Red Hat, Inc.

[4] . Linux is a registered trademark of Linus Torvalds.

[5] Chrome is a trademark of Google

[6] Firefox is a registered trademark of Mozilla.

facilitate easy and timely update deployment. Automated updates are already the status quo for most mobile devices, with patches coming directly from the vendor to the device.

An automated and comprehensive patch management strategy, executed promptly upon patch release, is the only viable strategy for risk mitigation. Instead of relying on administrators to manually deploy updates, automatic updates minimize the human factor and free up scarce enterprise IT resources. Enterprises that have blocked URLs to prevent automatic updates can use the HTTP-Connectivity-Tester tool to identify what resources need to be unblocked to facilitate the implementation of automatic updates [5].

Why Upgrade?

Only major software upgrades incorporate new security features that effectively address entire classes of attack. Figures 2 and 3 below demonstrate features implemented via upgrades in Microsoft Windows® and Apple iOS®, respectively. Deploying software with the latest anti-exploitation features forces an adversary to spend considerable resources to bypass defenses or find new vulnerabilities. These features have the potential to make known and unknown vulnerabilities difficult or impossible to exploit.

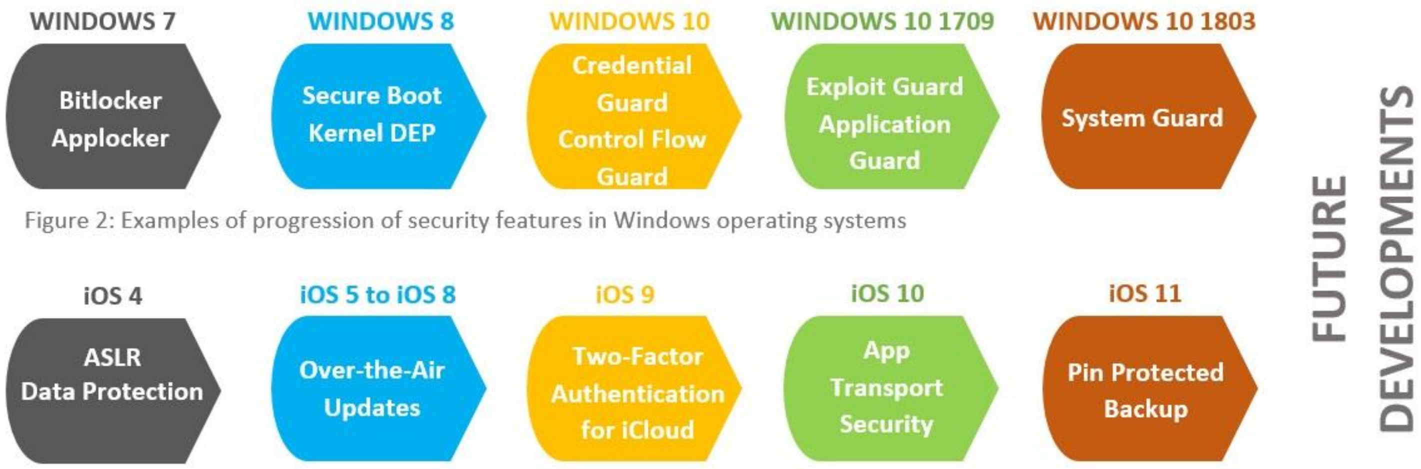

Figure 2: Examples of progression of security features in Windows operating systems

Figure 3: Examples of progression of security features in Apple iOS systems

Examples of anti-exploitation features in operating system software include:

- Exploit Guard: Provides built-in intrusion prevention capabilities to reduce the attack/exploit surface of applications
- App Transport Security: Enforces best practices in the secure connections between an application and its back-end
- Secure or Trusted Boot: Verifies that only the intended operating system is loaded

New features often require low-level architectural changes, so their implementation in older versions is unlikely or incomplete. Additionally, vendors often can only support patches for a limited number of older versions before security updates become infeasible. By upgrading systems, enterprises make themselves dynamic, and therefore more difficult targets. A study by the RAND Corporation, which analyzed 207 exploitable zero-day vulnerabilities over a 14 year period, found that nearly a quarter of the patched security vulnerabilities had no public documentation and the only way to address them is by upgrading to the latest product version [6]. Although upgrading to newer operating systems and applications incurs costs, these costs pale in comparison to losses of sensitive information from a compromise.

Patch immediately. Upgrade regularly. If cybersecurity is truly a priority, there is no other choice.

Works Cited

[1] T. Rains, D. Weston, and M. Miller. “Exploitation Trends: From Potential Risk to Actual Risk.” Presented at RSA© Conference, San Francisco, CA, 2015. Available: https://www.rsaconference.com/writable/presentations/file_upload/br-t07-exploitation-trends-from-potential-risk-to-actual-risk.pdf

[2] M. Shahzad, et al. “A Large Scale Exploratory Analysis of Software Vulnerability Life Cycles.” Proceedings of the 2012 International Conference of Software Engineering, pages 771-781. June 2012.
[3] NIST Special Publication 800-40 Rev.3, “Guide to Enterprise Patch Management Technologies.” Available: nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf
[4] Charles Reis, Adam Barth, and Carlos Pizano. “Browser Security: Lessons Learned from Google Chrome.” June 18, 2009. Available: http://queue.acm.org/detail.cfm?id=1556050
[5] HTTP Connectivity Tester. Available: https://github.com/nsacyber/HTTP-Connectivity-Tester
[6] RAND Corporation. “Zero Days, Thousands of Nights.” March 9, 2017. Available: https://www.rand.org/pubs/research_reports/RR1751.html

Disclaimer of Warranties and Endorsement

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

iCloud, iOS, and macOS are registered trademarks of Apple. Microsoft Windows, Windows 7, Windows 8, Windows 10, AppLocker, and BitLocker are registered trademarks of Microsoft. Red Hat is a registered trademark of Red Hat, Inc. Linux is a registered trademark of Linus Torvalds. Chrome is a trademark of Google. Firefox is a registered trademark of Mozilla.

Contact

Client Requirements / General Cybersecurity Inquiries: Cybersecurity Requirements Center, 410-854-4200, Cybersecurity_Requests@nsa.gov
Media inquiries / Press Desk: 443-634-0721, MediaRelations@nsa.gov