Addressing Cybersecurity Risks with Patch Management

In these "field it fast, fix it later" times, patch management has become extremely important. [1]

It is also on the verge of undergoing significant change, driven by both the importance of a digital economy that claims an increasingly large percentage of overall GDP and the ever more harrowing risks posed by unchecked cybersecurity threats.[2]

The damage from these threats is difficult to estimate, but one study finds that the total cost of cybercrime per US organization rose an average of 29% from 2017 to 2018 and that the global economic value at risk from cybercrime over the next five years stands at $5.2 trillion.[3]

Nor, of course, is the damage merely financial. Cyberthreats pose very real risks to public health and safety as well as to civil liberties and the energy, communications, transportation, and defense capabilities on which modern societies depend.

Responses to these cyberthreats take two different yet complementary approaches:

1 Change how software and hardware systems are initially developed

Some argue that the "field it fast, fix it later" paradigm must give way to more responsible, accountable technology development, even arguing that tech vendors should meet certain cybersecurity standards which, if unmet, should trigger full-blown product/service recalls.

Short of that, some government agencies are pressing for technology transparency — e.g., the US Commerce Department's Software Bill of Materials disclosure process that requires software and IoT (Internet of Things) vendors to share details about underlying components, libraries, and dependencies of their wares.

Meanwhile, California — the fifth largest economy in the world — has passed the IoT law AB1906, taking effect in January 2020, that delineates security features required in all digitally connected devices, including device attestation, code signing, and low-level firmware component security auditing.

Other, more familiar regulatory efforts include the European Union's GDPR (General Data Protection Regulation) and the California Consumer Privacy Act (CCPA), which holds organizations accountable for the digital security of personal data, and China's national cybersecurity law specifying data protection requirements and rules about cross-border data flows and critical information infrastructures.

2 Double down on securing fielded software and systems — with automated patch/upgrade management

While we wait for enhanced security and better transparency in software and systems, we must rely on what has long been called patch management and embrace automation and processes that help make it more effective and repeatable as a first line of cyberdefense.

By leveraging patch management automation and routine auditing of inventory to patch as much of the environment as possible, these approaches will eventually ease patching hassles. But as vendors begin to push larger patches and version upgrades (e.g., Microsoft's 1903 version of Windows) businesses are expected to enforce a mature test regiment prior to deploying to production environments.

However they're labeled, patches and/or upgrades will continue to do double duty after devices, systems, and apps are in operational environments, fixing vulnerabilities, flaws, and bugs as well as adding features and capabilities that improve functionality and performance.

The importance of patching/upgrading is most obvious when it is not performed — just ask anyone victimized by WannaCry or the corporate counsel of any organization whose failure to patch led to cybercrimes that spawned legal liability, regulatory fines, and/or reputational damage.

Nine tips for a successful patch/upgrade

Because it's easy for a patch/upgrade to be handled poorly or missed altogether, even smaller organizations should implement a formal, proactive patch/upgrade management program that sustains patching and upgrading processes, procedures, and schedules. To be most effective, such a program should utilize automated tools which ensure the security and optimal operation of core infrastructure and systems.

Patch/upgrade management is never-ending, and its complex lifecycle should be handled with these nine best practices:

1. Inventory and categorize your IT assets

A complete list of all IT assets — network, servers, workstations, PCs, mobile and IoT devices, operating systems, applications, etc. — should be categorized based on their exposure to risk so you can determine what needs immediate critical patching/upgrading (timeframes in hours/days) and what requires standard patching/upgrading (timeframes in weeks).

2. Ensure that all operating systems in your environment are supported

Not just Windows, but also Mac, Linux, and Unix.

3. Embrace a multi-layered security model and enforce WAF

This requires WAFs (web application firewalls) that analyze web app transactions to prevent attacks in transit and, without needing to modify the app's source code, stop malicious exploits before they reach vulnerable web apps.

4. Maintain a patch/upgrade/update information database

You need to know all your vendor's latest patches, upgrades, and fixes, including additional information regarding classifications and compatibilities.

5. Establish a patch/upgrade rollback plan

You may need to reverse patches and/or upgrades, returning your environment to its pre-patched/upgraded state, so create a plan with procedures to handle that.

6. Formalize monthly patch/upgrade deployment with a schedule

Your organization's needs will determine your patching/upgrading routine — all at once over a weekend, in weekly pre-set increments — so you avoid adverse impacts.

7. Test your patches and upgrades on a staging system

You need to test patches and upgrades to make sure they don't break anything before deploying them to operational systems; include system reboots in your staging system testing so unexpected post-patch/upgrade reboots don't affect the business.

8. Deploy those patches and upgrades

Automated patching/upgrading tools can help you prevent unintended effects that might disrupt your operations.

9. Assess every patch/upgrade deployment and mitigate for exceptions as necessary

Use a post-patch/upgrade deployment assessment to analyze logs and exceptions, formally verifying that all planned deployments are executed correctly and implementing your rollback in the face of significant issues. When you leave a system or app unpatched or not upgraded, be prepared to deploy appropriate additional security measures (e.g., removing direct internet access, locking down user permissions, whitelisting apps to prevent execution of untrusted payloads).

One of the best ways to deal with the challenges of "field it fast, fix it later" patch/upgrade management is to engage a services provider experienced in patch and upgrade management that offers both a customizable PMaaS (patch management as a service) solution and the expertise to help you implement the complex, cyclical patching/upgrading processes so essential to protecting and securing your enterprise.

1 https://www.cigionline.org/articles/patching-our-digital-future-unsustainable-and-dangerous

2 https://blogs.wsj.com/cio/2017/11/03/gdp-doesnt-work-in-a-digital-economy/

3 https://www.accenture.com/_acnmedia/PDF-96/Accenture-2019-Cost-of-Cybercrime-Study-Final.pdf