Patch (computing)

From Wikipedia, the free encyclopedia

A **patch** is a piece of software designed to update a computer program or its supporting data, to fix or improve it.[1] This includes fixing security vulnerabilities[1] and other bugs, with such patches usually called **bugfixes** or **bug fixes**,[2] and improving the usability or performance. Although meant to fix problems, poorly designed patches can sometimes introduce new problems (see software regressions). In some special cases updates may knowingly break the functionality, for instance, by removing components for which the update provider is no longer licensed or disabling a device.

Patch management, is a part of lifecycle management, and is the process of using a strategy and plan of what patches should be applied to which systems at a specified time.

Contents

Types

Patches for proprietary software are typically distributed as executable files instead of source code. This type of patch modifies the program executable—the program the user actually runs—either by modifying the binary file to include the fixes or by completely replacing it.

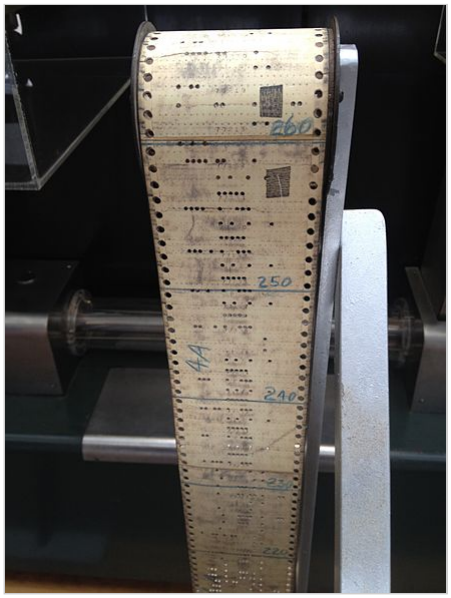
Patches can also circulate in the form of source code modifications. In this case, the patches usually consist of textual differences between two source code files, called "diffs". These types of patches commonly come out of open-source projects. In these cases, developers expect users to compile the new or changed files themselves.

Because the word "patch" carries the connotation of a small fix, large fixes may use different nomenclature. Bulky patches or patches that significantly change a program may circulate as "service packs" or as "software updates". Microsoft Windows NT and its successors (including Windows 2000, Windows XP, Windows Vista and

Windows 7) use the "service pack" terminology.[3] Historically, IBM used the terms "FixPaks" and "Corrective Service Diskette" to refer to these updates.[4]

History

Historically, software suppliers distributed patches on paper tape or on punched cards, expecting the recipient to cut out the indicated part of the original tape (or deck), and patch in (hence the name) the replacement segment. Later patch distributions used magnetic tape. Then, after the invention of removable disk drives, patches came from the software developer via a disk or, later, CD-ROM via mail. With the widely available Internet access, downloading patches from the developer's web site or through automated software updates became often available to the end-users. Starting with Apple's Mac OS 9 and Microsoft's Windows ME, PC operating systems gained the ability to get automatic software updates via the Internet.



A program tape for the 1944 Harvard Mark I, one of the first digital computers. Note physical patches used to correct punched holes by covering them.

Computer programs can often coordinate patches to update a target program. Automation simplifies the end-user's task – they need only to execute an update program, whereupon that program makes sure that updating the target takes place completely and correctly. Service packs for Microsoft Windows NT and its successors and for many commercial software products adopt such automated strategies.

Some programs can update themselves via the Internet with very little or no intervention on the part of users. The maintenance of server software and of operating systems often takes place in this manner. In situations where system administrators control a number of computers, this sort of automation helps to maintain consistency. The application of security patches commonly occurs in this manner.

Application

The size of patches may vary from a few kilobytes to hundreds of megabytes; thus, more significant changes imply a larger size, though this also depends on whether the patch includes entire files or only the changed portion(s) of files. In particular, patches can become quite large when the changes add or replace non-program data, such as graphics and sounds files. Such situations commonly occur in the patching of computer games. Compared with the initial installation of software, patches usually do not take long to apply.

In the case of operating systems and computer server software, patches have the particularly important role of fixing security holes. Some critical patches involve issues with drivers.[5] Patches may require prior application of other patches, or may require prior or concurrent updates of several independent software components. To facilitate updates, operating systems often provide automatic or semi-automatic updating facilities. Completely automatic updates have not succeeded in gaining widespread popularity in corporate computing environments, partly because of the aforementioned glitches, but also because administrators fear that software companies may gain unlimited control over their computers. Package management systems can offer various degrees of patch automation.

Usage of completely automatic updates has become far more widespread in the consumer market, due largely to the fact that Microsoft Windows added support for them, and Service Pack 2 of Windows XP (available in 2004) enabled them by default. Cautious users, particularly system administrators, tend to put off applying patches until

they can verify the stability of the fixes. Microsoft (W)SUS support this. In the cases of large patches or of significant changes, distributors often limit availability of patches to qualified developers as a beta test.

Applying patches to firmware poses special challenges, as it often involves the provisioning of totally new firmware images, rather than applying only the differences from the previous version. The patch usually consists of a firmware image in form of binary data, together with a supplier-provided special program that replaces the previous version with the new version; a motherboard BIOS update is an example of a common firmware patch. Any unexpected error or interruption during the update, such as a power outage, may render the motherboard unusable. It is possible for motherboard manufacturers to put safeguards in place to prevent serious damage; for example, the upgrade procedure could make and keep a backup of the firmware to use in case it determines that the primary copy is corrupt (usually through the use of a checksum, such as a CRC).

Video games

Video games receive patches to fix compatibility problems after their initial release just like any other software, but they can also be applied to change game rules or algorithms. These patches may be prompted by the discovery of exploits in the multiplayer game experience that can be used to gain unfair advantages over other players. Extra features and gameplay tweaks can often be added. These kinds of patches are common in first-person shooters with multiplayer capability, and in MMORPGs, which are typically very complex with large amounts of content, almost always rely heavily on patches following the initial release, where patches sometimes add new content and abilities available to players. Because the balance and fairness for all players of an MMORPG can be severely corrupted within a short amount of time by an exploit, servers of an MMORPG are sometimes taken down with short notice in order to apply a critical patch with a fix.

In software development

Patches sometimes become mandatory to fix problems with libraries or with portions of source code for programs in frequent use or in maintenance. This commonly occurs on very large-scale software projects, but rarely in small-scale development.

In open-source projects, the authors commonly receive patches or many people publish patches that fix particular problems or add certain functionality, like support for local languages outside the project's locale. In an example from the early development of the Linux kernel (noted for publishing its complete source code), Linus Torvalds, the original author, received hundreds of thousands of patches from many programmers to apply against his original version.

The Apache HTTP Server originally evolved as a number of patches that Brian Behlendorf collated to improve NCSA HTTPd, hence a name that implies that it is a collection of patches ("a patchy server"). The FAQ on the project's official site states that the name 'Apache' was chosen from respect for the Native American Indian tribe of Apache. However, the 'a patchy server' explanation was initially given on the project's website.[6]

Variants

Hotfix

A hotfix or Quick Fix Engineering update (QFE update) is a single, cumulative package that includes information (often in the form of one or more files) that is used to address a problem in a software product (i.e., a software bug). Typically, hotfixes are made to address a specific customer situation. Microsoft once used this term but has stopped

in favor of new terminology: General Distribution Release (GDR) and Limited Distribution Release (LDR). Blizzard Entertainment, however, defines a hotfix as "a change made to the game deemed critical enough that it cannot be held off until a regular content patch".

Point release

A point release is a minor release of a software project, especially one intended to fix bugs or do small cleanups rather than add significant features. Often, there are too many bugs to be fixed in a single major or minor release, creating a need for a point release.

Program temporary fix

Program temporary fix or Product temporary fix (PTF), depending on date, is the standard IBM terminology for a single bug fix, or group of fixes, distributed in a form ready to install for customers. Customers sometime explain the acronym in a tongue-in-cheek manner as *permanent temporary fix* or more practically *probably this fixes*, because they have the option to make the PTF a permanent part of the operating system if the patch fixes the problem.

Security patches

A *security patch* is a change applied to an asset to correct the weakness described by a vulnerability. This corrective action will prevent successful exploitation and remove or mitigate a threat's capability to exploit a specific vulnerability in an asset. Vulnerability management, is a part of patch management, and is the cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities.

Security patches are the primary method of fixing security vulnerabilities in software. Currently Microsoft releases its security patches once a month, and other operating systems and software projects have security teams dedicated to releasing the most reliable software patches as soon after a vulnerability announcement as possible. Security patches are closely tied to responsible disclosure.

Service pack

A service pack or SP or a feature pack (FP) comprises a collection of updates, fixes, or enhancements to a software program delivered in the form of a single installable package. Companies often release a service pack when the number of individual patches to a given program reaches a certain (arbitrary) limit, or the software release has shown to be stabilized with a limited number of remaining issues based on users' feedback and bug tracking such as bugzilla. In large software applications such as office suites, operating systems, database software, or network management, it is not uncommon to have a service pack issued within the first year or two of a product's release. Installing a service pack is easier and less error-prone than installing many individual patches, even more so when updating multiple computers over a network, where service packs are common.

Unofficial patches

An unofficial patch is a non-commercial patch for a commercial software created by a third party instead of the original developer. Similar to an ordinary patch, it alleviates bugs or shortcomings. Examples are security fixes by security specialists when an official patch by the software producers itself takes too long.[7][8] Other examples are unofficial patches created by the game community of a video game which became unsupported abandonware.[9][10]

Hot patching

Hot patching, also known as *live patching* or *dynamic software updating*, is the application of patches without shutting down and restarting the system or the program concerned. This addresses problems related to unavailability of service provided by the system or the program.[11] A patch that can be applied in this way is called a *hot patch*. This is becoming a common practice in the mobile app space.[12] Companies like Rollout.io use method swizzling to deliver hot patches to the iOS ecosystem.[13] Another method for hot-patching iOS apps is JSPatch.[14]

Slipstreaming

In computing, slipstreaming is the act of integrating patches (including service packs) into the installation files of their original app, so that the result allows a direct installation of the updated app.[15][16]

Slipstreaming can save time and money. Initially, it involves some work, but can save time later on in re-installation terms. This is especially significant for administrators that have to manage a large number of computers, where the default case for installing an operating system on each computer would be to use the original media and then update each computer after the installation was complete, as opposed to using a more up-to-date (slipstreamed) source, and having to download/install a minimal number of updates.

However, not all patches can be applied in this fashion and one disadvantage is that if it is discovered that a certain patch is responsible for later problems, said patch cannot be removed without using an original, non-slipstreamed installation source.

See also

- Software release life cycle
- Software maintenance
- Backporting
- Patch (Unix)
- Porting
- Delta encoding
- SMP/E
- Automatic bug fixing

References

1. "Microsoft issues biggest software patch on record". *Reuters*. 2009-10-14. Retrieved 14 October 2009.
2. "What is a Bug Fix? – Definition from Techopedia". *techopedia.com*. Retrieved 2015-07-29.
3. "Service Pack and Update Center". *windows.microsoft.com*. Retrieved 2015-06-01.
4. http://www.tavi.co.uk/os2pages/glossary.html
5. Liu, Ashok (2012). *Computercare's Laptop Repair Workbook: The 300 Cases of Classic Notebook Computers Troubleshooting and Repair*. AuthorHouse. p. 591. ISBN 9781477205402. Retrieved 2015-01-08. "Uninstall High Definition Audio driver patch KB835221 & KB888111 [...]"
6. Wayback Machine link to the Apache website, 1997 (http://web.archive.org/web/19970615081902/http://www.apache.org/info.html)
7. Barwise, Mike (2007-10-16). "Unofficial patch for Windows URI problem". The H Security. Retrieved 2012-01-29.
8. "Another unofficial IE patch offered to counter critical flaw". Computer Weekly. 2006-03-30. Retrieved 2013-07-09. "*Another unofficial patch has been released to counter a critical flaw in Microsoft's Internet Explorer browser.*"

9. Wen, Howard (2004-06-10). "Keeping the Myths Alive". linuxdevcenter.com. Retrieved 2012-12-22. "*[...]fans of the Myth trilogy have taken this idea a step further: they have official access to the source code for the Myth games. Organized under the name MythDevelopers, this all-volunteer group of programmers, artists, and other talented people devote their time to improving and supporting further development of the Myth game series*."
10. Bell, John (2009-10-01). "Opening the Source of Art". Technology Innovation Management Review. Retrieved 2012-12-30. "*[...]that no further patches to the title would be forthcoming. The community was predictably upset. Instead of giving up on the game, users decided that if Activision wasn't going to fix the bugs, they would. They wanted to save the game by getting Activision to open the source so it could be kept alive beyond the point where Activision lost interest. With some help from members of the development team that were active on fan forums, they were eventually able to convince Activision to release Call to Power II's source code in October of 2003*."
11. "Oracle Magazine". Oracle.com. Retrieved 2013-01-04.
12. "Hot or Not? The Benefits and Risks of iOS Remote Hot Patching « Threat Research Blog". *FireEye*. Retrieved 2016-10-26.
13. Perez, Sarah. "Rollout.io Puts Mobile Developers Back In Control Of Their Apps". *TechCrunch*. Retrieved 2016-10-26.
14. "bang590/JSPatch". *GitHub*. Retrieved 2016-10-26.
15. Karp, David (14 July 2008). "Build an XP SP3 Recovery Disc". *PC Magazine*. Ziff Davis.
16. Thurrott, Paul (7 May 2008). "Slipstreaming Windows XP with Service Pack 3 (SP3)". *Supersite for Windows*. Penton.

External links

- *The Jargon File* version 4.4.7 entry for *patch* (http://www.catb.org/~esr/jargon/html/P/patch.html)
- A detailed masters dissertation dealing with security patches (http://singe.za.net/blog/archives/763-Limiting-Vulnerability-Exposure-through-effective-Patch-Management-a-thesis.html)
- Official Linux kernel patch format (http://linux.yyz.us/patch-format.html)
- 0-day-patch - Metric comparing patch performance of Microsoft and Apple (http://www.techzoom.net/publications/0-day-patch)