Jak a proč se dělají záplaty

Aktualizaci systémů a programů se na svém počítači nevyhnete

VOJTĚCH BEDNÁŘ

Už jste někdy záplatovali počítač? To znamená, zda jste do něj instalovali nějakou aktualizaci programu? Pokud máte operační systém od Microsoftu a používáte internet, pak skoro určitě ano. Ale ani dalším programům a systémům od jiných výrobců se potřeba záplatování nevyhýbá. Co ale ve skutečnosti taková záplata představuje? Jak se dělá a jakými mechanismy se distribuuje? A proč by její výroba neměla být otázkou několika desítek minut?

Svět je plný počítačů a ty jsou plné aplikací. Každý program, než je uvolněn pro veřejnost, prochází testováním. Počínaje výrobkem nadšence, který si jej ověří alespoň u dvou kamarádů, zda nepadá, až po profesionální produkt softwarové společnosti s tisíci zaměstnanci, která jej kontroluje mnohem sofistikovanějšími metodami. V každém případě aplikace, která se dostane do volného oběhu nebo prodeje, by měla být alespoň minimálně funkční. Mnoho chyb však nelze odhalit ani tím nejpodrobnějším testováním. A tak se čas od času stane, že je nějaký nedostatek v programu objeven až po nějaké době při jeho používání. Jak si ukážeme dále, nemusí jít vždy nutně pouze o chybu, která by souvisela s nešikovností programátora nebo toho, kdo aplikaci navrhoval. Problematické místo může v aplikaci vzniknout mnoha dalšími způsoby nebo si jej tam dokonce lidé mohou přimyslet.

V každém případě pak je nutné problém nějak vyřešit. U jednoduchých programů se to dělá vydáním aktualizované verze. Problém v aplikaci prostě odstraníme tím, že ji odinstalujeme a nahradíme takovou verzí, kde se již tento nedostatek nevyskytuje. Vydávání aktualizovaných verzí aplikací však má také svoje nedostatky. Hlavním z nich je, že aktualizovaný program je potřeba nejprve odinstalovat – to často znamená vymazat i jeho data a přijít o jeho nastavení. Nová verze také nemusí být nejmenší a především představuje zásah do fungujícího prostředí (byť s chybou).

A tak je zde druhá možnost, kterou je vydávání jednotlivých aktualizací. Tedy jakýchsi balíčků, jež nahrazují vybrané části programu. Lidově se jím říká záplaty a v poslední době se s nimi můžeme setkat takřka na každém kroku. Proč? Protože, jak se ukazuje, dokonalá aplikace nebo dokonalý systém neexistuje a záplatovat se chtě nechtě musí. Vydávání aktualizací (záplat) nicméně kromě řešení problémů také přináší a vyvolává mnoho dalších obtíží.

Typy problémů

Pokud čtete elektronické či papírové časopisy o počítačích, víte, že takřka každý druhý den je vydána aktualizace nějakého programu. Tyto aktualizace obvykle řeší „bezpečnostní chyby". Stalo se módou posledních několika let hledat tyto chyby v oblíbených operačních systémech a aplikacích. Ačkoliv „nejděravější" jsou v tomto smyslu operační systémy z rodiny Windows, není to pouze vina jejich výrobce – společnosti Microsoft – ač právě na ni je vina často svalována. Opravované chyby jsou často dílem kutilů, kteří je schválně vyhledávají. Takto záplatované „problémy" by bez svých objevitelů pravděpodobně neexistovaly a jejich řešení může v některých případech přinést více neštěstí než pozitiv. Bezpečnostní chyby navíc ani zdaleka nejsou jediným typem problému, který by vyžadoval aktualizaci. Základním předpokladem je, že problém je nalezen až po oficiálním vydání aplikace. V zásadě můžeme události, jež vedou k vydávání záplat, kategorizovat takto:

Funkční chyby jsou nedostatky ve fungování aplikace nebo její části. Komponenta (funkce či něco podobného) většinou díky nedokonalému testování za určitých okolností selhává, nebo nefunguje tak, jak by fungovat měla. Některé části aplikace nejsou za určitých okolností dostupné nebo jsou nepoužitelné. Příčinou funkční chyby je většinou chyba na straně programátorů a softwarových inženýrů.

Vnitřní kolize vznikají tehdy, když si dvě části stejného programu nebo systému uvnitř nerozumí. V důsledku toho může docházet k selhávání části aplikace nebo celku. Aplikace (systém) se může chovat neočekávaně, může padat, může docházet ke ztrátě dat. Vnitřní kolize se nejčastěji vyskytují tehdy, když různé části programu vyvíjejí různé týmy a dostatečně nekomunikují (nebo nepřipraví dostatečnou dokumentaci). V současné době je tento typ kolizí u nových aplikací vzácností a někdy vzniká jako důsledek předchozích aktualizací. Celkově množství výskytu těchto chyb klesá.

Vnější kolize neboli nekompatibilita. Aplikace může být nekompatibilní buď s jinou aplikací, nebo se součástí operačního prostředí, ve kterém funguje, případně s hardwarem. K nekompatibilitě může dojít vlivem nedokonalého testování (vina je na programátorech), ale také vlivem časového posunu. V tom případě se program stane nekompatibilním s „moderním" operačním systémem ne proto, že by se změnil on, ale proto, že se změnil operační systém. Nekompatibilita je tedy vyvolána někým jiným a původní programátoři za ni nemohou. Podobně mohou kolidovat i jednotlivé aplikace mezi sebou. Ne vždy je taková kolize odstranitelná a ne vždy má smysl se ji snažit odstranit. Příkladem jsou antivirové systémy. V manuálu ke každému z nich najdeme, že jej nesmíme používat současně s jiným systémem.

Změna použití se týká především operačních systémů. Funkce, která byla zamýšlena pro určité použití, se v praxi používá jinak. Na takové použití ale není konstruovaná, a tak je potřeba tuto funkci přepracovat, aby vyhovovala novému účelu.

Exploit – konečně to nejoblíbenější. V nějaké části aplikace nebo systému je objeven nedostatek a je popsán způsob, jak tohoto nedostatku využít k překonání ochrany aplikace a pro přístup k informacím v ní uloženým, který nebyl tvůrci plánován. Většinou se to týká jednotlivých programů, v krajním případě ale může úspěšný exploit jediné aplikace vést až ke stavu, kdy ten, kdo jej využije, získá přístup k celému počítači nebo počítačovému systému.

Interpretační chyby – nejedná se o chyby v pravém slova smyslu, ale o nedorozumění. Něco, co je ve své podstatě naprosto bezproblémové, se stává problémem v důsledku toho, jak je to interpretováno. Nedorozumění mezi tvůrci aplikace nebo systému a mezi jeho uživateli nebo tvůrci podřazených aplikací vede až k tomu, že se objeví některá z předchozích chyb a interpretační selhání se tak stává například důvodem, proč něco objektivně nefunguje jak má, nebo je dokonce vystaveno riziku napadení. Interpretační chyby lze považovat za hlavní zdroj jak vnějších, tak i vnitřních kolizí.

Toto byly pouze některé z problémů, které vyvolávají potřebu aktualizací. Pokusili jsme se vytvořit seznam generických chyb. Tento seznam není úplný a nemusí být ani přesný, nicméně by měl poskytnout dostatečný přehled o základních a nejčastějších nedostatcích. Běžný počítačový program je v současné době již poněkud složitou záležitostí. V jeho jádru se díky použití sofistikovaných vývojových nástrojů dokonce – ač je to paradoxní – nemusí vyznat ani jeho tvůrci. Především z hlediska bezpečnostních problémů je možné, že nedostatek vznikne naprosto mimo kontrolu autorů aplikace, aniž by na něj mohli včas přijít. To je důvod, proč je mnoho nedostatků i přes intenzivní testování aplikací objeveno až mnoho měsíců po jejím oficiálním uvolnění.

Jak doktor vyměnil nohu

Jak jsme si řekli, pokud se v nějakém systému najde chyba, existují dvě cesty, jak z toho ven. První možnost je, řekneme-li to trochu zjednodušeně, vymazání této aplikace a její nahrazení novou, bezchybnou verzí. To je ale velmi často problém – tím větší, s čím složitější aplikací pracujeme (zkuste kvůli chybě v kalkulačce reinstalovat Windows). Proto softwarové firmy vydávají balíky aktualizující pouze jednotlivé části jejich dílek. Bohužel, takový balík – záplata řešící problém nebo aktualizace, dodávající novou funkci či vlastnost – s sebou nese nejedno riziko. Především je potřeba zvládnout aktualizační mechanismus. Pokud se to nepovede, může nainstalování updatu vést k většímu problému než byl ten, který měla záplata řešit. Mezi další rizika patří „pooperační" problémy. Například opravená komponenta se nebude snášet s něčím jiným. Vzniku podobného problému se výrobci softwaru snaží zabránit tím, že stejně jako aplikace, která má být opravována, prochází i záplaty velmi přísným testováním na oba dva typy kolizí a na další možnosti, jak by mohly negativně interferovat s okolním prostředím. To ale neznamená, že se vždy podaří všechny nedostatky objevit. Celý proces by se dal přirovnat k tomu, kdybychom chtěli zraněnému člověku přišít novou končetinu od někoho jiného. V případě, že se to podaří, bude tento člověk znovu chodit. Pokud ale nikoliv, možná umře. Rizika, která jsou spojena se záplatováním aplikací a především pak operačních systémů, se v případě, že takový systém, respektive stejné či přímo kooperující komponenty jsou záplatovány vícenásobně, mnohonásobí. Vzniká pak totiž zmatek vycházející z toho, že současně může existovat mnoho různých verzí prakticky téhož. Tím se sice (spekulativně) může zvyšovat bezpečnost, protože heterogenní systém není možné napadnout tak snadno jako systém víceméně jednolitý. Současně ale klesá kompatibilita se součástkami a s aplikacemi, které jsou na takovém systému závislé, a dále dochází k problémům i v jeho vlastním rámci. Například při nesprávném používání některých z funkcí systému podřízenými programy může i při takové změně, která by „papírově" neměla být nekorektní, dojít větší aktualizací jeho vnitřních částí k zásadnímu narušení konzistence běžících aplikací. Takové chybě se říká „Guru Meditation" a můžeme se s ní setkat takřka vždy, když je vydána nová verze operačního systému pro jakýkoliv počítač. Přichází ale často i s jednotlivými aktualizacemi – například s nedávným vydáním balíku SP2 pro Windows XP od Microsoftu. To je ovšem trochu odbočka někam jinam.

Jak se dělá aktualizace?

Je půlnoc a automatický systém ve vašem počítači zjistil, že je k dispozici nová verze jedné ze síťových komponent. Během několika minut dojde k jejímu stažení z aktualizačního serveru Microsoftu a nainstalování do vašeho počítače. Že se uvnitř něco změnilo, se díky plně automatickému nastavení v podstatě nedozvíte. V době, kdy se aktualizace stahuje, totiž spíte a ráno počítač funguje stejně, jako fungoval večer, i když s novou komponentou. Jak je to možné? Proces, který na první pohled vypadá jednoduše a nic od vás nevyžaduje, je výsledkem poměrně dlouhého vývoje plného omylů. A sám o sobě rozhodně jednoduchý není. Pojďme se tedy podívat, jak taková modelová aktualizace vzniká.

- **Na počátku je nalezení problému.** Řekněme, že nezávislý bezpečnostní auditor našel (česky a mnohem přesněji bychom řekli, že vyšťoural) zranitelnost v komponentě webového prohlížeče. Při použití určité syntakticky sice správné, ale nesmyslné struktury skriptovacího jazyka na webové stránce se stane, že prohlížeč v operačním systému zareaguje chybně. Jeho část se zacyklí a pokud je mu následně dodán odkaz na spustitelný soubor, vykoná jej, aniž by se zeptal uživatele. To je krajní možnost chyby a navíc velice generalizovaná, nicméně právě takto některé z objevených bezpečnostních nedostatků vypadají. Auditor dále zjistí, že k problému je náchylná pouze jedna verze webového prohlížeče, přitom stejný operační systém může obsahovat nejméně tři různé verze. Podle toho, kdy byl vydán, zda byla již aktualizována jeho jiná část a také podle toho, v jaké jazykové verzi v daném systému existuje.

Webový prohlížeč se skládá z mnoha komponent. Když problém zobecníme (velmi silně), pak tyto komponenty můžeme najít na disku počítače na různých místech nejčastěji v podobě dynamických knihoven – souborů s příponou .dll. Každá z těchto knihoven obsahuje číslo své verze. Toto číslo je, když se na něj podíváte, poměrně podrobné. Identifikuje totiž jednak generaci dané komponenty, jednak její samotnou verzi a jednak číslo sestavení, tak zvaný build. Chyba se pak může vyskytovat v určitém rozsahu sestavení, řekněme tedy, že je potřeba aktualizovat soubory, které mají číslo verze v rozsahu 6.0.2400.1200 – 6.0.2400.1300. Čísla mají vnitřní význam, který může být u různých produktů různý a rozdíl jedné stovky v čísle buildu rozhodně neznamená, že na světě je sto různých verzí. V každém případě víme, kde chybu hledat.

Jak ji ale opravit? Výrobce musí přesně vědět, za jakých okolností je možné nalezeného bezpečnostního nedostatku využít. Jedině simulováním ta-

▲ **LiveUpdate společnosti Symantec je samostatná aplikace.**

▲ **Na této obrazovce vidíme nalezené aktualizace.**

▲ **LiveUpdate stahuje aktualizace do produktů a komponent Symantecu.**

kových okolností je možné problém v komponentě prostudovat a zjistit, co přesně jej způsobuje. Mnoho aplikací včetně webového prohlížeče se programuje ve vývojových balících, pracujících na tak zvané vysoké úrovni. To znamená, že programátoři se k nejnižší vrstvě aplikace již prakticky nedostanou, ale právě tam dochází k největšímu počtu chyb. Zranitelná komponenta je spouštěna v prostředí, které umožňuje její přesné monitorování a na základě jejich reakcí je vyhledán jak potenciál selhání, tak i přesný mechanismus, jenž k němu vede. Následně jsou navrhovány změny, které je potřeba provést, aby došlo k odbourání zjištěné závady. Tyto změny ale musí být navrhovány s ohledem na další věci, mezi něž patří zejména:

Zachování funkčnosti. To znamená, že všechno, co má prohlížeč správně umět a co fungovalo před vydáním záplaty, musí fungovat i posléze.

Zachování konzistence. Aktualizací jedné části nesmí vzniknout chyba v jiné nebo nesmí dojít ke kolizi s jinou.

Zachování kompatibility. Programy určené pro webový prohlížeč by měly fungovat i poté, co se některá jeho část změní. To je problém, protože do duše jejich autorů programátoři prohlížeče nevidí.

V některých případech stačí provést drobnou změnu a potřebnou část překompilovat. V okamžiku, kdy se tak stane, dojde ke zvýšení čísla build a případně subverze upravované komponenty. Jindy je změna výraznější nebo je dokonce třeba celou komponentu přepsat. Zde si musíme uvědomit, že co na disku počítače představuje soubor o velikosti několik desítek kilobajtů, je ve skutečnosti moře řádků zdrojového kódu v programovacím jazyce. Vyznat se v něm není zdaleka jednoduché, zejména ne v případě, kdy chybu způsobuje až kompilátor – program, který převádí zdrojový kód komponenty do podoby, která je srozumitelná počítači a dalším kompilovaným programům.

Změna v jedné komponentě si také může vynutit změnu v dalších částech. Ty sice opravovanou chybu neobsahují, nicméně by posléze mohly se změněnou částí programu prohlížeče interferovat, a tak je nutné zajistit, aby k tomu nemohlo docházet. Zalepení jediné chyby tak vede k nutnosti vyměnit řekněme dvě další knihovny.

● **Testování aktualizovaných částí.** Řekněme, že výrobce úspěšně aktualizoval jednu část webového prohlížeče a vyřešil tak potenciálně zneužitelné místo. Aktualizované části se dostávají na testovací pracoviště. Zde je třeba zjistit, zda jejich nasazení – to znamená výměna starších verzí za novější v potenciálně napadnutelných systémech – nemůže způsobit nějaký problém. Komponenty se testují na množství počítačů a jsou schválně uváděny do takových situací, u nichž se předpokládá, že by mohly způsobit problémy, interference, nebo by mohly být vyvolány hackerem ve snaze najít další bezpečnostní nebo funkční nedostatek. Velmi často se stane, že při takovém testování je skutečně další problematická část objevena a je nutné s celým procesem vytváření aktualizace jít prakticky vzato zase na začátek. To se může několikrát opakovat, až testovací středisko dospěje k názoru, že aktualizovaná verze je v rámci možností bezchybná.

V minulosti došlo u produktů Microsoftu, ale i dalších výrobců několikrát k tomu, že při programování záplaty, respektive aktualizované komponenty byla zanedbána fáze testování. Výsledkem bylo, že aktualizovaná komponenta způsobila mnohem větší problémy než ty, které vlastně měla řešit. V současné době probíhá velmi detailní testování každé připravované aktualizace. V případě bezpečnostních záplat ale toto testování silně komplikuje faktor času. Dochází totiž k tomu, že čas mezi objevením nedostatku a jeho praktickým zneužitím se zkracuje. Červi, založení na nově objevených nedostatcích, se objevují hodiny poté, co byly tyto nedostatky zveřejněny. Strategie založené na utajování bezpečnostních problémů až do okamžiku, kdy jsou vydány záplaty, se ukazují jako ne zrovna účinné, a tak se testovací týmy bezpečnostních, ale i funkčních aktualizací dostávají do časového stresu. Tím pochopitelně stoupá pravděpodobnost, že vydaná záplata obsahuje chybu. Existuje několik různých koncepcí co s tímto stavem dělat a jejich popis by zde byl nad naše možnosti.

Následuje fáze přípravy samotné aktualizace. Až doposud jsme totiž hovořili pouze o aktualizované verzi komponenty, u níž byla objevena chyba, a o nové verzi několika dalších, které, jak se ukázalo, s touto komponentou bezprostředně souvisí. Zkušený uživatel nebo počítačový nadšenec by si byl schopen ve zranitelném webovém prohlížeči tyto komponenty vyměnit sám. To ale rozhodně neplatí o uživateli nezkušeném a už vůbec ne o správci, který má na starosti řekněme několik desítek až stovek počítačů. U každého z nich by musel zjistit, zda obsahuje zranitelnou verzi, a teprve kdyby zjistil, že ano, měl by ji nahradit za novou, bezchybnou. To ale také může znamenat nutnost pracovat s procesy v počítači, některé z nich zastavit, aby došlo k „odblokování" souboru a ten mohl být přepsán, eventuálně restartování počítače a znovuspuštění procesů. Proto je vytvořena aktualizace – balíček. Ta se skládá jednak z aktualizovaných souborů a jednak ze seznamu akcí, které je potřeba provést, aby byla aktualizace nainstalována. Vzniká záplata v takové formě, jak ji známe. Tato záplata je následně instalována do počítače.

Instalace záplaty – ať už ji zprostředkovává k tomu určený systém, třeba Windows Update, nebo ji provádíme ručně – vypadá následovně.

Nejprve instalační balíček (z praktických důvodů mívá formu jednoho spustitelného souboru) zkontroluje, zda nedošlo například při jeho stahování z internetu nebo zápisu na médium, ze kterého jej spouštíme, ke změně obsažených dat. Pro tyto účely jsou v něm uloženy kontrolní součty všech souborů aktualizace i příslušných skriptů, které se v něm nacházejí. Spustitelná část balíku jej extrahuje do počítače a porovná, zda to, co bylo extrahováno, odpovídá přesně tomu, co tvůrci zabalili a vypustili do světa. Aktualizace jsou totiž potenciálním cílem hackerů. Může také dojít k jejich narušení během přepravy – instalování jakkoliv porušené komponenty do jinak fungující aplikace nebo operačního systému by mohlo mít katastrofální následky.

Pokud je všechno v pořádku, musí instalátor zjistit, zda počítač, ve kterém byl spuštěn, aktualizaci vůbec potřebuje. Jestliže instalujeme aktualizaci ze spustitelného souboru, tedy ručně, děje se tak až po jejím extrahování. Jestliže ale používáme aktualizační systém, je tento proces převrácen a provádí se úplně na začátku, ještě před stažením updatu ze serveru výrobce softwaru. Nutnost aktualizace se zjišťuje podle čísel verzí komponent. Existuje několik přístupů, uveďme si tedy některé z nich:

▲ **Windows Update je aktualizace ovládaná z webového rozhraní.**

▲ **Systém Windows Update hledá dostupné aktualizace.**

– V počítači je komponenta určité verze (ekvivalence).
– V počítači komponenta určité verze chybí, nebo je k dispozici verze nižší (škála).
– V systému najdeme několik komponent daných verzí (výčet).

Na základě těchto pravidel se rozhoduje o nutnosti aktualizace. Kromě verzí však mohou rozhodovat i další faktory. Těm, které musí být splněny, aby mohlo dojít k nainstalování záplaty, říkáme podmiňující. Je to například určitá jazyková verze operačního systému – některé záplaty totiž nelze nainstalovat na verzi jinou. Existují ale i takové, které nainstalování záplaty, byť je pro daný systém určena, zabrání – jsou jejími kontraindikátory. Může jít například o přítomnost jiného softwaru nebo komponenty, která se s tou aktualizovanou vylučuje. Možností je ovšem pochopitelně ještě o něco více.

Následně je třeba zajistit, aby instalátor záplaty měl právo přístupu k souborům, které má změnit. Prakticky to znamená zastavení procesů, jež tyto soubory používají. V moderních Windows lze tuto akci provádět za plného chodu systému, ve starších verzích však byla výměna systémového souboru nerozlučně spjata s restartováním operačního systému a s následnou operací, která byla sice technicky velmi zajímavá, ale současně poměrně riskantní. Po této operaci byla nutná výměna samotných souborů, které předcházelo zálohování jejich původních verzí. Je to proto, aby bylo možné v případě potřeby aktualizaci (záplatu) zase odinstalovat a vrátit se k původnímu stavu vyměněných komponent. Zajímavý problém se v tomto okamžiku objevuje tehdy, jestliže je jedna komponenta aktualizována vícenásobně.

Po provedení výměny souborů je většinou potřeba ještě upravit konfiguraci daného programu. V případě webového prohlížeče to znamená změnit záznam o jeho verzi, aby ji mohlo být možné při dalších aktualizacích správně identifikovat, případně pro potřeby aplikací, které jej využívají. Záznamy o provedených záplatách se ukládají do speciálních katalogů. Ty slouží k odlaďování programů, sledování jejich změn a mají význam i z hlediska počítačové bezpečnosti.

Excesy se záplatami

Záplaty aplikací a systémů problémy účinně řeší, ale jak jsme si zde již několikrát řekli, mohou je i způsobovat. Totéž bohužel platí o záplatovancích aplikacích. Chyby již byly nalezeny v aktualizačním mechanizmu společnosti Symantec, ale také v několika dalších, menších. Dalším faktorem je, že „doporučené", tedy plně automatické nastavení v podstatě zcela vynechává z procesu aktualizace počítače uživatele. Ten tak sice není odpovědný za potenciální problémy, k nimž může dojít, ale ztrácí kontrolu nad tím, co přesně se s jeho systémem děje. Většina aktualizačních mechanismů má různá střední nastavení, nicméně tato nastavení bývají označována jako expertní a běžní uživatelé jsou od nich (v zásadě správně) odrazováni. Několik nedávných problémů ovšem ukazuje, že ani aktualizacím nelze stoprocentně věřit. Naštěstí se již softwarovým výrobcům podařilo vyřešit zásadní problém aktualizačních systémů – totiž otázku, jak má aktualizace opravit sama sebe.

▲ **Zde se nachází seznam aktualizací, jejich stav a datum provedení.**

Automatické aktualizační systémy

Aktualizační systémy jako například Windows Update nebo Live Update společnosti Symantec pomáhají udržovat produkty, pro něž jsou určeny, v aktuálním stavu. To znamená, že průběžně zjišťují verze nainstalovaných komponent a pomocí speciálního protokolu je porovnávají s těmi nejnovějšími dostupnými. Pokud je vydána aktualizace, pak se tyto systémy postarají o její korektní stažení a nainstalování. Pravdou je, že ne vždy je možné používat jejich služeb, ale práci při aktualizacích počítačů výrazně usnadňují. V případě velkých podniků jsou aktualizace některých komponent testovány ještě dříve na úrovni k tomu určeného oddělení firmy, až pak jsou nainstalovány jednotlivým uživatelům. V tom případě se například u produktů Microsoftu aktualizace jednotlivých počítačů neprovádí ze serveru Windows Update, ale z k tomu určených serverů v rámci firmy. Totéž se děje i tam, kde se více lpí na tom, aby všechny počítače fungovaly naprosto spolehlivě a také tam, kde právě dvakrát nedůvěřují výrobcům softwaru.

▲ **Obrazovka nastavení automatických aktualizací ve Windows.**

Záplaty potřebujeme

Bez aktualizací softwaru (jak funkčních, tak i bezpečnostních) se v současné době již prakticky neobejdeme. Jejich existence a vývoj má svá pozitiva, ale přináší také bohužel negativní důsledky. Těchto negativních důsledků bychom si měli být vědomi. Lze předpokládat, že v dohledné době množství objevených bezpečnostních mezer, především v operačních systémech, bude dále stoupat. Jestliže přitom hovoříme pouze o záplatování Windows, dopouštíme se veliké chyby. Jak se totiž na desktopové počítače dostávají alternativní systémy, především různé distribuce OS GNU/Linux, stoupá počet chyb, které jsou objeveny i v nich a tím pádem stoupá i potřeba aktualizací. Pro ty, kteří si zvykli hanět Microsoft, že jeho produkty jsou vadné, se v souvislosti s uvedeným trendem blíží do jisté míry „doba pravdy". Například aktualizace Linuxu je v mnoha ohledech technicky obtížnější než aktualizace systémů Windows, především díky dodnes ne zcela dokonale zvládnutému systému balíčků a jejich vzájemných závislostí, které se často týkají jednotlivých verzí.

Záplaty nás tedy provázejí a ještě provázet budou. Doufáme, že tento text přinesl alespoň základní informaci o tom, jak vypadá jejich vývoj, testování a distribuce. V mnoha ohledech by se dalo říci, že jde o proces, který je obtížnější než proces vývoje původního softwaru, který je následně záplatován. Někteří odborníci tvrdí, že vývoj záplaty se trochu podobá psaní dalšího dílu rozjetého televizního seriálu – víte přesně, co děláte, ale naprosto vše musí sedět. Za vším tím tichým stahováním a instalováním komponenty někdy o půlnoci nebo ve dvě hodiny ráno je ukryt vskutku poměrně složitý proces, který jsme zde mohli spíše jen naznačit, než popsat do všech důsledků. Pokud vás zajímají další informace, obraťte se na weby určené pro programátory. Najdete jich tam mnoho – avšak informace budou mnohem více technického rázu.

5 0321/FEL ❑