Chraňte svá data na webu

O kriminálnících, kteří ke svým nekalým činům využívají počítačových sítí, jste už mnohokrát slyšeli. Dokáží napáchat obrovské škody, avšak mnoho provozovatelů počítačových systémů si to stále zcela neuvědomuje. Čím více se na internet přesouvají další a další lidské aktivity, tím víc je třeba o bezpečnosti přemýšlet. Ohroženy jsou peníze, informace, soukromí, dobrá pověst, a v některých případech dokonce i hospodářská infrastruktura.

S neustálým růstem složitosti sítí je zajištění jejich bezpečnosti stále komplikovanějším úkolem. Jediná bezpečnostní chyba vede ke kompromitaci celé soukromé sítě. Tento článek se zaměřuje na to, jak proti hackerským útokům co nejlépe zabezpečit webové aplikace, které mohou být cílem útoku samy o sobě nebo které se mohou stát prostředkem průniku do vnitřní sítě. Webová aplikace je systém několika kooperujících částí. Jedná se o webový server, interpreter jazyka aplikace a databázový server. Proti útokům je proto nutné ošetřit nejen zdrojové kódy, ale i webový a databázový server, případně vhodně konfigurovat interpeter jazyka.

HACKER CHODÍ HLAVNÍM VCHODEM

Bránou do WWW aplikace je internetový prohlížeč. Běžný uživatel ho používá k nakupování, psaní a čtení (svých) e-mailů, ke komunikaci s úřady, s knihovnou, školou atd. Sedne-li si k prohlížeči hacker, je možné, že se mu podaří krást v elektronických obchodech, číst a psát cizí e-maily, změnit stránky politické partaje, která ho ve večerních zprávách poněkud naštvala, a provést mnoho dalších útoků. Hackerská komunita disponuje pro svou "práci" specializovanými nástroji, kterými lze útok provést, ale nejsou výjimkou ani případy útoku pouze s internetovým prohlížečem.

Aby mohl hacker na webovou aplikaci zaútočit, potřebuje jí alespoň povrchně porozumět a odvodit tak možné slabiny celého systému. K tomu mu prohlížeč dobře poslouží. Hackera například zajímá, na jakém typu webového serveru aplikace "běží", zajímá ho použitá databáze, jazyk v němž je aplikace napsána, struktura souborů aplikace. Kde konkrétně tyto informace získá, již možná sami tušíte. URL v adresové řádce často napoví typ použitého jazyka a parametry, s nimiž aplikace pracuje. V HTTP hlavičce bývá kromě jiných zajímavých informací uveden i název webového serveru. Princip správy uživatelských relací je možné zjistit z cookies a z URL. Databázový server lze odhalit vyvoláním chybového hlášení vhodnou změnou některé z částí URL.

Na základě takové analýzy se pak útočník pokusí najít slabá místa aplikace. Vy se ale můžete pokusit hackerům takové techniky zkomplikovat, popřípadě úplně znemožnit.

ZDROJOVÉ KÓDY BEZ BEZPEČNOSTNÍCH DĚR

Protože hacker pravděpodobně využije k "hacknutí" aplikace prohlížeč, aplikace by měla vždy sama prověřit vstupní údaje přebírané z prohlížeče a zásadně jim nedůvěřovat.

Měla by prověřit délku i obsah každého vstupního parametru. Nikdy nenechávejte kontrolu vstupních dat na klientských skriptech, jako je například JavaScript - ten totiž lze v prohlížeči vypnout nebo pozměnit. Ani vlastnosti nastavené u prvků formuláře nemohou kontrolu vstupu v žádném případě zabezpečit. Určení maximální délky vstupního pole nebo předávání vstupních údajů ve skrytých polích je sice na první pohled praktická věc, ale není to vůbec bezpečné. Uživatel, vlastně hacker, totiž může celou stránku před odesláním uložit k sobě na počítač, změnit některé údaje v kódu a pak teprve změněný formulář odeslat aplikaci. V aplikaci by se měly mezi jejími jednotlivými stavy předávat jen nezbytné údaje a vše ostatní byste měli vždy znovu načítat z databáze. Argument o rychlosti aplikace zde neuspěje!

Kromě důkladného ověření vstupních hodnot je hlavní zásadou důkladné ošetření chybových stavů aplikace. O každé chybě, která nastane, by měla aplikace zaslat do prohlížeče krátké upozornění, které neprozradí víc, než je nutné. Nespoléhejte se na generování hlášení samotným serverem, může hackerovi prozradit užitečné informace (viz výše).

NA INTERNETU NIKDO NEVÍ, ŽE JSI PES

Tento citát z kresleného vtipu, ač starý deset let, platí i dnes doslova. Podaří-li se hackerovi zmocnit se něčí identity, mívá to pro poškozeného velmi nepříjemné následky.

Aplikace, která poskytuje určité uživatelské soukromí, nejprve nabídne přihlašovací formulář a po úspěšné autentizaci pak po celou dobu přihlášení udržuje tzv. relaci, jejímž primárním účelem je odlišit jednotlivé přihlášené uživatele. Důvodem tvorby relací je bezstavový princip komunikace mezi serverem a prohlížečem. Chce-li se neoprávněná osoba dostat na cizí webový účet, musí buď prolomit přihlašovací formulář, nebo oklamat skript pro správu relace.

Na prolomení autentizace existují programy nebo si hacker napíše vlastní skript - částečně se mu dá zamezit dodržováním pravidel pro tvorbu hesel. Používání minimálně šestimístných (raději delších) hesel, která jsou kombinacemi písmen, čísel a znaků, může prolomení přinejmenším prodloužit.

Uživatelská relace bývá obecně realizována tak, že se po přihlášení vygeneruje identifikátor, který se pak předává a ověřuje ze stránky na stránku, tak jak uživatel prochází aplikací. Identifikátory se většinou předávají v cookies, skrytých polích či v URL řetězci. Protože cookies může uživatel v prohlížeči vypnout, je jejich použití většinou ještě kombinováno s předáváním v URL. Podaří-li se hackerovi rozluštit způsob generování identifikátoru a nahradí svůj identifikátor identifikátorem jiné osoby, dojde ke krádeži relace a útočník se tak přímo ocitá na cizím účtu.

Bránit se proti uhádnutí principu relace lze mnoha způsoby. Tvůrce aplikace by se měl snažit, aby tvorba identifikátoru byla pokud možno obtížně rozluštitelná. Vhodné je také při přechodu ze stránky na stránku identifikátor měnit. Ověřování identifikátoru by mělo být kombinováno ještě například s kontrolu IP adresy či jiných údajů.

ZABEZPEČENÍ WEBOVÉHO SERVERU A DATABÁZE

Pokud využíváte webhosting, moc toho s bezpečností serveru a databáze nenaděláte a nezbude vám než se spolehnout na firmu, u které aplikaci na serveru máte. Na druhou stranu si ale také ušetříte mnoho starostí i peněz.

Jestliže je webová aplikace umístěna na vašem vlastním serveru, je potřeba se o jeho zabezpečení neustále starat. Nedobytný software bohužel neexistuje, i když se vám to možná nějaký prodejce pokusí tvrdit - velmi záleží hlavně na konfiguraci softwarového i hardwarového vybavení.

Informace o konkrétním nastavení webového a databázového serveru od výrobce je vhodné doplnit zkušenostmi ostatních uživatelů z diskusních fór a z webových stránek zabývajících se bezpečnostní problematikou. Každopádně je nutné neustále sledovat aktualizace programového vybavení. Tyto "záplaty" (patche) výrobci většinou zveřejňují na svých WWW stránkách.

NUTNÉ, ALE NE POSTAČUJÍCÍ ZABEZPEČOVACÍ TECHNOLOGIE

Existuje řada technologií, které přinášejí větší obranu proti útokům. Žádná z nich bohužel také nefunguje stoprocentně, ale to není důvod, proč je nevyužívat. Samozřejmě, každý server by měl mít svou "vstupní bránu", firewall, který brání vnitřní síť proti nepovolenému druhu přístupu zvenčí, popřípadě má i další funkce. Firewall však nemusí být jako zabezpečení proti útoku na samotnou webovou aplikaci účinný, protože propouští webovou komunikaci na portu 80.

Detektor průniku (IDS - Intrusion Detection Systems) je zařízení, které má za úkol na základě monitoringu činností v síti odhalit webový útok a zamezit mu. Mezi komerčními programy je i jeden open source detektor - jmenuje se Snort a více informací o něm hledejte na adrese www.snort.org.

Údaje, které by měly zůstat utajeny, například hesla, je vhodné posílat zašifrované. Nejčastěji se pro zakódování používá SSL (Secure Sockets Layer). Podpora SSL se konfiguruje na serveru. Komunikace mezi počítači pak neprobíhá protokolem HTTP, ale HTTPS. Pro provoz je třeba získat od certifikačního serveru digitální certifikát. Takto zašifrovaná data je možné odposlouchávat stejně jako nezašifrovaná, ale rozšifrovat je lze pouze se znalostí certifikátu.

Jednorázové heslo (OTP - One-Time Password) je účinný lék na odposlouchávání a krádeže hesel. Pokud ho vetřelec získá, je mu k ničemu. Toto heslo totiž může být použito pouze jednou, potom jeho platnost končí. Jistota, že uživatel, který právě žádá aplikaci o autentizaci, je skutečně tím, za koho se vydává, je pak mnohonásobně vyšší. Heslo vygeneruje uživateli zařízení, které má u sebe a které je synchronizováno s technickým vybavením na cílovém serveru.

TEST SLABÝCH MÍST

Abyste zjistili "obrněnost" své aplikace proti útokům, nemusíte čekat na nezvané hosty, ale aplikaci i ostatní související části, například SQL server, si můžete otestovat. Na internetu jsou k dispozici testovací nástroje (např. www.nextgenss.com), popřípadě je možné si zdarma stáhnout programy pro hackery - hackeři si totiž v rámci své komunity nástroje ochotně sdílejí. Existují také společnosti, které vaše stránky za nevelký peníz na bezpečnostní chyby otestují (např. http://netcraft.com/).

NA ZÁVĚR

Na závěr vám chci připomenout, abyste si pro případ útoku nezapomněli svá data zálohovat, a popřát vám co nejvíce odražených útoků.

Soňa Žáčková

ODKAZY, KDE LZE ZÍSKAT PRAKTICKÉ INFORMACE

http://packetstormsecurity.nl
www.owasp.org
http://neworder.box.sk
www.defcon.org (Nejznámnější hackerské centrum, buďte opatrní!)