Bezpečnost webových aplikací (skriptování na straně serveru, šifrování, hashování)

Bezpečnost webových aplikací

Webové aplikace a data uživatelů na serveru je samozřejmě nutné nějakým způsobem zabezpečit. Webovým aplikacím jako takovým (jejich očekávané funkčnosti a stabilitě) hrozí nejvíce obvykle uživatelé, kteří aplikaci nepoužívají správným (programem očekávaným) způsobem a následně lidé, kteří se snaží využít právě bezpečnostních děr v programu (např. pomocí SQL injection).

Data uživatelů často bývají uložena v databázi. I ta je nutné zabezpečit, zejména pak uživatelská hesla. K jejich zabezpečení se používá hashování.

Skriptování na straně serveru

Nejčastěji je využíváno právě pro běh webových aplikací. Protože je kód vykonáván na straně serveru, nemá k němu klient přístup (jako např. u JavaScriptu) a je tak zajištěna integrita kódu, zatímco při skriptování na straně klienta je možné různé kontroly zadaných hodnot obejít. Je-li kontrola prováděna na straně serveru, obejít ji nelze. Veškeré zabezpečení je nutné tedy zajistit na straně serveru (kontrola zadané hodnoty – číslo, string, enum, …), kontrola maximální délky aj…).

Zároveň je na serveru nutné zajistit ošetření dalších možných problémů, jako je například možnost provedení tzv. SQL injection – odeslání SQL kódu webovým formulářem na server, který jej namísto toho, aby s ním pracoval jako s textem, zpracuje. Data v databázi tak nejsou chráněná před neoprávněným čtením či například jejich nenávratným odstraněním.

Šifrování

Šifrování je proces převodu zprávy do tzv. šifrované podoby prostřednictvím nějakého klíče. Zašifrovaná data je možné zpětně dešifrovat.

Symetrické šifrování

Jde o šifrování, kde je k šifrování i k dešifrování zprávy používán jediný klíč. Podstatnou výhodou je nízká výpočetní náročnost (asymetrické šifrování může být až stotisíckrát pomalejší). Velkou nevýhodou však je nutnost sdílení tajného klíče, takže se na něm odesílatel a příjemce musí předem domluvit.

Asymetrické šifrování

Jde o šifrování, kde se k šifrování a k dešifrování zprávy používají odlišné klíče – klíč veřejný a klíč privátní. Veřejný klíč příjemce se používá k zašifrování zprávy a privátní klíč příjemce pak k jejímu dešifrování. Šifrovací i dešifrovací klíč spolu musejí být matematicky svázány, nezbytnou podmínkou pro užitečnou šifry však je nemožnost ze znalosti veřejného klíče zjistit klíč privátní.

Kromě šifrování dat se asymetrické šifrování používá také pro vytvoření elektronického podpisu – možnost prokázat u dat jejich autora.

Hashování

Hashování je proces převodu nějakých vstupních dat do relativně malého čísla. Výstup hashovací funkce se nejčastěji označuje jako hash nebo otisk. Hashovací funkce se používají k rychlejšímu prohledávání tabulky, porovnávání dat (např. hesel v databázi), ale například i při hledání podobných úseků DNA sekvencí v bioinformatice.

Mezi hlavní vlastnosti hashovací funkce patří následující:

1. jakékoliv množství vstupních dat vede ke stejně dlouhému výstupu (hashi)
2. malou změnou vstupních dat dosáhneme velké změny na výstupu (výsledný hash se bude na první pohled zásadně lišit od původního)
3. z hashe není možné rekonstruovat původní text zprávy (na rozdíl od šifrování)
4. v praxi je vysoce nepravděpodobné, že dvěma různým zprávám odpovídá stejný hash

Například v PHP jsou vestavěné hashovací funkce MD5, SHA-1 a spousta dalších, které je možné využít prostřednictvím funkce hash($algorithm, $data). K hashování hesel by se však měl používat pro ještě vyšší bezpečnost takový algoritmus, který zároveň uměle zpomaluje vygenerování hashe pro eliminaci možnosti využít brute-force útok k nalezení kolize[1].

[1] kolizní řetězec je řetězec, který má stejný hash jako jiný řetězec