Session

Session (v překladu *relace*, méně častěji *sezení* nebo *seance*) v informatice představuje permanentní síťové spojení mezi klientem a serverem, zahrnující výměnu paketů.

Session v různých komunikačních protokolech

U protokolů jako je telnet nebo FTP session odpovídá spojení na úrovní nižšího protokolu TCP. V případě použití protokolů které žádnou podporu pro sessions nemají (UDP), nebo kde spojení typicky trvají velmi krátkou dobu (HTTP), jsou session udržovány přímo aplikačním programem, a k tomu nutné informace jsou vkládány do přenášených dat.

Session v HTTP

Session v protokolu HTTP dává webovému serveru možnost uložit si libovolné (většinou však ne příliš obsáhlé) informace o uživatelích, kteří k němu přistupují, a to o každém zvlášť. Protokol HTTP ze svého principu (a způsobu komunikace stylem požadavek - odpověď) postrádá kontext o jednotlivých klientech, a právě session ho webovým aplikacím dokáže dát.

Předávání session

Session je v HTTP předáváno dvěma nejrozšířenějšími způsoby:

- v URL cílové stránky – jako její proměnná
- jako HTTP cookie

Konfigurace session

To, jakým způsobem se session bude přenášet a pod jakým názvem, nastavuje webový server (například u Apache se název session ukládá v konfiguraci pod proměnnou nastavení `session.name`; použití cookies pak nastavují proměnné `session.use_cookies` a `session.use_only_cookies` popř. `session.cookie_httponly`, maximální délka platnosti se nastaví v proměnné `session.gc_maxlifetime`, která je implicitně 1440 sekund – tzn. pokud uživatel k webovému místu nepřistoupí do 24 minut od posledního požadavku, session ztrácí).

Výhody a nevýhody jednotlivých způsobů přenosu

Nevýhodou přenosu přes cookies je to, že některé archaické internetové prohlížeče je nepodporují (těch je ovšem mizivé procento) a ty soudobé dovolují cookies vypnout (opět, podíl uživatelů s vypnutými cookies je zanedbatelný). Jinou potenciální nevýhodou může být to, že pokud by potenciální útočník získal přístup k adresáři na serveru, do kterého se cookies ukládají, dostal by se i k session. Výhodou je to, že se nepřenáší v těle požadavku, nikolik jeho URL.

Naopak, session v URL danou adresu znepřehledňují, činí ji příliš dlouhou, nezapamatovatelnou a působí proti principům SEO (mohou dokonce „rozmělňovat“ odkazovatelnost toho kterého odkazu). Při práci s aplikací s tímto způsobem předávání se jednotlivá session navíc ukládají i do historie prohlížeče a záložek. Je-li například (proti bezpečnostním zásadám) hodnota session generovaná na základě přístupových údajů uživatele, může se k nim útočník dostat snadněji než v případě ukládání do cookies. Některé webové aplikace (např. phpMyAdmin) proto při přihlašování vyžadují možnost ukládat cookies.

Session předávané jako cookie

Typickým příkladem je použití HTTP cookie k uložení jednoznačného identifikátoru (pojmenovaného **SESSIONID**, **SESSID**, **SID** apod., jehož hodnota je při startu session náhodně vygenerována). Podle takto uloženého HTTP cookie pak server ve své paměti najde potřebné informace, například o přihlášeném uživateli, jeho úrovni přístupu a podobně. Podstatné je, že samotná data se již nepřenášejí (ani v URL, ani v samotném požadavku).

Pokud se klient může připojit k libovolnému serveru z clusteru, je třeba mezi jednotlivými servery informace o sessions buď sdílet, nebo zajistit, že se stejný klient vždy připojí ke stejnému uzlu. V opačném případě by se klient mohl spojit se serverem, který o zahájené session neví, a tak přijít o přihlášení, stav nákupního košíku a podobně.

Session pro skriptovací jazyky

Z hlediska skriptovacích jazyků pro programování intranetových/internetových aplikací, session představuje množinu proměnných (někde přístupnou přes sadu funkcí, jinde přes globální proměnnou), které dovolují uchovávat hodnoty, které jim byly nastaveny, po dobu připojení (tj. se znovunačtením stránky se neztratí).

Například, v jazycích ASP a ASP.NET jsou přístupné automaticky jako jednotlivé prvky pole `Session("klíč")`, v PHP pak pod tzv. superglobálním polem `$_SESSION["klíč"]`. V obou příkladech lze do session ukládat „jednoduché“ typy (celá čísla, racionální čísla, řetězce, null) i z nich složená pole (teoreticky libovolné složitosti). Hodnoty se ukládají tzv. serializované (převedené na řetězec podle formátu, ze kterého jej lze zpětně načíst (přibližně jako třeba JSON)).

Session a bezpečnost

Zabezpečení session funguje na tom, principu, že přístup k proměnným konkrétního uživatele je možný přes jeho identifikátor v podobě dostatečně dlouhého, náhodně generovaného, a tedy neuhádnutelného klíče. Druhé specifikum je to, že session má poměrně krátkou dobu platnosti (implicitně bývá 24 minut, není-li nastaveno jinak). Samotné použití session ovšem automaticky nezaručuje, že se k datům pomocí nich uložených dostane pouze uživatel, jemuž jsou určeny. Webové aplikace – pokud session využívají – by současně s ním měly zavést konkrétní principy, které rizko zneužití session minimalizují.

Session hijacking

Následující metody umožňují útočníkovi získat přístup k session své oběti. Některé z nich využívají sociální inženýrství. Všechny jsou možné, pokud je cílová webová aplikace proti těmto metodám nezabezpečená. Obecně se nazývají **session hijacking**.

- **Session fixation –** útočník na internetu uloží stránku s odkazem na zamýšlenou aplikaci a svou oběť/oběti – např. přes ICQ, e-mail, … – aby na tento odkaz klikly a přihlásily se do aplikace. Daný odkaz v cílové URL adrese již obsahuje (konstantní) HTTP cookie, která se (pokud aplikace není zabezpečena proti tomuto útoku) použije. Útočník poté použije stejný odkaz a získá stejná oprávnění jako před tím přihlášený uživatel.
- **Session sidejacking –** pokud se SESSIONID přenáší spolu s URL adresou, může útočník využít packet sniffing (sledování packetů), ze kterých hodnotu SESSIONID přečte a (dokud se původní uživatel neodhlásil a jeho relace nevypršela) může se pokusit přihlásit s touto SESSION.
- Pokud se session ukládají v cookies, data z nich jsou uloženy na serverovém pevném disku. Dostane-li se tedy útočník k nim, může je (ty z nich, jejichž platnost ještě nevypršela) zneužít.
- Útočník může pro získání session zkombinovat sociální inženýring s útokem typu cross-site scripting, kdy přiměje uživatele již přihlášeného do aplikace spustit odkaz se záškodným kódem, jenž získá SESSIONID oběti a odešle ho útočníkovi.

Prevence

- Nejstarší webové aplikace využívající session si jejich identifikátor často stanovovaly samy. V některých případech jako zašifrované údaje související s údaji daného uživatele. Tento přístup je považován jako bezpečnostní chyba – SESSIONID by vždy měly být zcela nahodilá a dostatečně dlouhá, aby odolaly útokům hrubou silou. Současné skriptovací jazyky toto automaticky nabízejí.
- Dodatečné ověřování – po přihlášení uživatele sledovat též IP adresu klientského počítače a řetězec identifikující webový prohlížeč (označovaný jako ***User-agent***) a v případě, že přijde další požadavek z jiné IP adresy nebo z jiného prohlížeče, session ukončit a její proměnné vymazat. Tato metoda se doporučuje implementovat, přestože nemusí být 100% účinná (v případě, že útočník a oběť mají sdílenou IP adresu (např. ve stejné firmě), a používají stejný typ prohlížeče). Nicméně řeší více metod ukradení a následného použití SESSIONID a dokáže útočníka zastavit v poslední fázi, ať už se k cizímu session dostal jakkoli.
- Proti session fixation se doporučuje regenerovat po přihlášení hodnotu SESSIONID (např. v PHP funkcí `session_regenerate_id`).
- Alternativně k tomu některé služby regenerují SESSIONID s každým požadavkem na server. To výrazně redukuje dobu, po kterou má útočník příležitost nabytou hodnotu SESSIONID využít.
- Šifrování dat přenášených mezi jednotlivými stranám, zejména pak SESSIONID – toto je v praxi využíváno v internetovém bankovnictví a jiných e-commerce aplikacích.

Odkazy

Zdroje

V tomto článku je použit překlad textu z článku Session hijacking [1] *na anglické Wikipedii.*

Související články

- HTTP cookie
- HTTP
- HTTPS

Reference

[1] http://en.wikipedia.org/wiki/En%3Asession_hijacking?oldid=360241235

Zdroje článků a přispěvatelé

Session *Zdroj*: http://cs.wikipedia.org/w/index.php?oldid=8612218 *Přispěvatelé*: Che, Hidalgo944, Premek.v, 2 anonymní úpravy

Licence