Session Riding

Jaromír Dobiáš, Zdeněk Říha, FI MU

Pojem „Session Riding" [2] označuje skupinu zranitelností a útoků typu Cross Site Request Forgery (CSRF) [3], [4]. Session Riding bývá často ztotožňován se samotným CSRF, avšak, jak již napovídá název, Session Riding je možno považovat za podtřídu, která se týká uživatelova otevřeného a již autentizovaného sezení (tzv. session). Útoky typu Session Riding je možné realizovat díky důvěře zranitelné služby v to, že dotazy, které služba přijímá prostřednictvím protokolu HTTP z prohlížeče uživatele, jsou platnými požadavky zadanými z iniciativy uživatele. Přestože počátky odhalování a dokumentování zranitelností založených na tomto principu je možno datovat do roku 2001 a některé zdroje hovoří dokonce o dobách dřívějších, počet webových služeb a aplikací, které je dnes stále možné napadnout prostřednictvím bezpečnostní zranitelnosti typu Session Riding, je alarmující.

Pro praktické ověření tohoto tvrzení jsme se rozhodli provést test nejmenovaného webového systému z prostředí českého Internetu poskytujícího mimo jiné e-mailové účty svým uživatelům. V průběhu testu se nám podařilo využitím kombinace zranitelnosti XSS (Cross Site Scripting, viz např. [5]) a Session Riding docílit toho, že zprávy určené příjemci byly automaticky přeposílány i na náš e-mailový účet a to pouze tím, že jsme uživateli zaslali speciálně upravený e-mail a počkali, až se přihlásí do systému. Žádná další aktivita příjemce nebyla nutná. K provedení útoku stačilo pouhé přihlášení oběti do systému.

1 Princip útoku

Útoky využívající zranitelnost typu Session Riding používají ke svému provedení relaci uživatele přihlášeného ke zranitelné službě. Jednoduchý útok může být realizován například tak, že útočníkem podstrčený URL odkaz vyvolá v prohlížeči nic netušícího přihlášeného uživatele akci, kterou daný uživatel nezamýšlel provést. Tato akce je provedena v kontextu autentizovaného spojení mezi daným uživatelem a zranitelnou webovou aplikací. Příkladem takové akce by mohl být v případě zranitelné aplikace elektronického bankovnictví například převod peněz z účtu podvedeného uživatele na účet útočníka:

```
http://zranitelnaBanka.com/
prevod.php?prevadenaCastka=
1000000&naUcet=IDutocnik
```

Pokud by bankovní webová aplikace akceptovala dotaz v uvedeném formátu bez kontroly dalších prvků, útočníkovi by stačilo vytvořit dostatečně důvěryhodný e-mail, který by přiměl uživatele kliknout na daný odkaz. K provedení převodu by však došlo pouze tehdy, pokud by byl uživatel v jiném okně případně v některé záložce daného prohlížeče přihlášen ke zranitelné aplikaci elektronického bankovnictví.

Útoky tohoto typu lze provádět především proto, že řada služeb verifikuje po úspěšném přihlášení uživatele pouze jeho identitu. Nekontroluje však, že požadavek, který pod danou identitou server obdržel, je skutečně zamýšleným požadavkem daného uživatele. Identita je v případě webových aplikací obvykle ověřována na základě identifikátoru session ID, který prohlížeč zasílá automaticky prostřednictvím HTTP metody GET, POST nebo nejčastěji v Cookie. Tím, že se identifikátor posílá při komunikaci automaticky, nemá uživatel prakticky žádnou možnost kontroly nad akcemi, které jsou vyvolávány z jeho prohlížeče. Pokud akce směřují do autentizované oblasti, zranitelná aplikace nedokáže rozeznat, zdali přicházejí skutečně od uživatele nebo z jiného zdroje využívajícího jeho prohlížeč. Vůči zranitelnosti typu Session Riding jsou náchylné také webové systémy, které používají k autentizaci uživatele metodu „Basic Authentication" podporovanou nativně protokolem HTTP. Při použití tohoto autentizačního mechanizmu jsou totiž autentizační údaje zasílány rovněž automaticky.

Jakmile se uživatel přihlásí do autentizované sekce zranitelné webové služby, mají veškeré následné dotazy odeslané z prohlížeče přihlášeného uživatele také možnost ovlivňovat stav uvnitř autentizované oblasti. Tato možnost existuje až do okamžiku, kdy se uživatel odhlásí nebo dojde k vypršení platnosti relace. V praxi to

znamená, že pokud uživatel v době svého aktivního přihlášení obdrží podvržený hypertextový odkaz, který směřuje do autentizované sekce, provede zranitelná webová aplikace akci podstrčenou útočníkem pod identitou přihlášeného uživatele. Útočník k tomu obvykle využívá sociálního inženýrství a snaží se nalákat uživatele na aktivaci nevinně vypadajícího hypertextového odkazu. Typicky se k propagaci podvrženého hypertextového odkazu využívá e-mail, chat nebo webové fórum.

Útočník však má také silnější zbraň, která mu umožňuje aktivovat podvržený odkaz v prohlížeči uživatele bez nutnosti jeho přímé spolupráce. Útočník například vyláká oběť na svoji stránku, která obsahuje HTML kód pro načítání externích objektů (např. HTML element IMG pro načítání obrázků). Kód za normálních okolností odkazuje na zdroj externího objektu (např. obrázku, souboru definic kaskádových stylů nebo JavaScriptu), odkud je jeho obsah načítán, avšak trik útočníka spočívá v tom, že místo původního obsahu vyvolává hypertextový odkaz požadovanou akci v autentizované sekci zranitelné webové aplikace. To sice může způsobit podezřelé anomálie v načítané stránce (například ikonku chybějícího obrázku), avšak prohlížeč ve snaze o načtení externího objektu voláním podstrčeného odkazu způsobí vykonání útočníkem zamýšlené akce, a to v kontextu aktuální relace uživatele! Byl-li například uživatel již přihlášen ke zranitelnému e-shopu, lze takto podvrhnout objednávku bez jakýchkoliv viditelných stop indikujících, že něco není v pořádku. K rozšíření těchto lstivých „objektů" mohou sloužit například různá webová fóra či blogy, kde je povoleno přímé načítání externích objektů.

Útoky vedené prostřednictvím zranitelnosti typu Session Riding bývají často přirovnávány k útokům krádeže relace a následné impersonaci, kdy útočník převezme plnou kontrolu nad napadeným účtem. Zranitelnost Session Riding však lze považovat v tomto směru za nebezpečnější, neboť její efekt může být v podstatě stejný, při vynaložení menšího úsilí a zanedbatelné možnosti detekce útoku. Při zneužití prohlížeče oběti se totiž podvržený požadavek útočníka jeví navenek jako právoplatný požadavek oběti, nevykazující atypický projev, jakým je například podezřelá změna IP adresy požadavku v případě útoku krádeže relace (tzv. session hijacking).

2 Příklady zranitelností

Původem zranitelnosti je samotná aplikace poskytující patřičnou funkčnost autentizovaným uživatelům. Obecně lze říci, že čím důležitější je webová aplikace, která je náchylná vůči útokům typu Session Riding, tím závažnější důsledky má útok využívající zranitelnosti v této aplikaci. Kromě zmiňovaného výskytu zranitelnosti v aplikacích elektronického bankovnictví patří mezi další rizikové potenciální zdroje například sociální sítě, aukční portály, portály pro nákup a prodej akcií a cenných papírů, webová rozhraní správy/administrace nejrůznějších systémů, nebo aplikace, které na základě jediného přihlášení poskytují přístup k širokému spektru webových služeb (tzv. Single Sign-On).

Session Riding ohrožuje především uživatele zranitelné služby, samotná webová služba je jen zprostředkovatelem útoku. To také může být důvodem pro značné rozšíření tohoto problému a malý zájem ze strany administrátorů a webových návrhářů a programátorů.

Ve světě se již objevily případy zranitelnosti tohoto typu ve službách elektronického bankovnictví renomovaných bankovních institucí. Známým se stal případ nalezení této zranitelnosti na portálu INGDirect.com týmem bezpečnostních výzkumníků z Princetonské Univerzity [6]. Jejich odhalení šokovalo veřejnost tím, že pomocí existující zranitelnosti bylo jednoduše možné provést převod peněz z účtu oběti na účet útočníka a to i přesto, že relace byla zabezpečena šifrovaným spojením SSL [1].

Jiným příkladem z praxe muže být například zranitelnost, která byla odhalena ve firmware verze 4.30.9 bezdrátového přístupového bodu Linksys WRT54GL [7]. S využitím této zranitelnosti bylo možné provést neautorizované změny v konfiguraci tohoto přístupového bodu. Bylo tedy možné například návštěvou určité stránky způsobit deaktivaci firewallu tohoto zařízení.

3 Jak se bránit?

Existuje celá řada více či méně úspěšných mechanismů, které bývají v boji proti zranitelnosti typu Session Riding používány. Ve snaze zabránit jejímu výskytu bývá jako obranný mechanismus často nasazována kontrola HTTP parametru „Referer" (URL předchozí stránky z níž jsme se dostali na aktuální stránku). Tímto způsobem se inkriminovaná aplikace snaží hlídat, zda požadavek na provedení autentizované akce pochází z očekávaného umístění rozhraní (obvykle z lokace, kde je webový formulář způsobující nastavení hodnot). Tato metoda není zcela vhodným řešením, jelikož nemalé množství uživatelů si parametr Referer blokuje. Tento mechanismus navíc není schopen zabránit útokům, které z očekávaného umístění pocházejí. Jiným obranným mechanismem, který bývá často nasazován jako protiopatření vůči zmiňované zranitelnosti, bývá nahrazení HTTP metody GET metodou POST při přenosu řídících parametrů. Tento mechanismus eliminuje kupříkladu možnost použití načítání externího objektu jako prostředku provedení útoku. Útočník však může obejít i tento mechanismus, zejména v případě výskytu zranitelnosti XSS, a docílit tak vykonání akce v kontextu uživatelského účtu oběti (např. využitím neviditelného rámce IFRAME obsahujícího podvrženou stránku). Útok je sice pracnější a méně efektivní než v případě nastavení hodnot prostřednictví URL, nicméně pokud dokáže útočník zkonstruovat dostatečně přesvědčivý scénář, pak s využitím technik sociálního inženýrství může být dopad útoku srovnatelný.

Často používanou obranou proti zranitelnostem webových služeb je použití potvrzovacích dialogů. Tato metoda je však v drtivé většině případů pouze na obtíž uživatele, navíc je útočník většinou schopen obejít i tento mechanismus, a to tak, že postupnou návštěvou patřičných hypertextový odkazů, které odpovídají akci potvrzení, simuluje kroky uživatele. I v případě, že je aplikace ošetřená sofistikovanější logikou potvrzovacích dialogů je schopen útočník simulovat akce uživatele například i vložením umělého zpoždění mezi jednotlivé dotazy v případě možnosti vyvolání kontextu JavaScriptu (nejčastěji prostřednictvím zranitelnosti XSS).

Vhodným přístupem, který řeší popisovaný bezpečnostní problém, je validace požadavků uživatele ve třech krocích. V prvním kroku se ověří, že uživatel je držitelem patřičných autentizačních údajů. V druhém kroku se ověří, zdali jsou v dotazu přítomny veškeré požadované argumenty. Ve třetím kroku se ověří, zdali uživatel opravdu použil patřičné webové rozhraní pro vytvoření a odeslání daného požadavku. K ověření požadavku je vhodné použít sdíleného tajemství mezi rozhraním na straně uživatele a aplikací na straně serveru. Toto tajemství je vhodné generovat pseudonáhodně na straně serveru v době, kdy uživatel úspěšně ověří svou identitu vůči němu. Tajemství není možné ukládat do cookie uživatele, jelikož cookie se zasílá serveru automaticky. Běžnou praxí je přidávat tento parametr do skrytého pole formuláře, který je proti útokům typu Session Riding ochráněn. Uvedený mechanismus zabraňuje útočníkovi zasílat validní dotazy prostřednictvím přihlášeného uživatele, neboť konstrukce platného dotazu vyžaduje znalost časově proměnného tajemství. Pokud by bylo možné tajemství automatizovaně zjišťovat jiným způsobem (například pomocí útoku XSS), byl by útok přece jen proveditelný, v každém případě se však použitím popsaného obranného mechanismu výrazně zvyšuje náročnost provedení úspěšného útoku.

4 Závěr

Se zranitelnostmi a útoky typu Cross Site Request Forgery se setkáváme již řadu let. Kategorie útoků „Session Riding", která se týká spojení zachovávajících stav mezi jednotlivými dotazy, je nebezpečná v tom, že zneužívá důvěru webové aplikace v platnost požadavků pocházejících z prohlížeče autentizovaného uživatele. Se službami zranitelnými vůči tomuto typu útoku se setkáváme relativně často. Existují sice metody, jak se webové služby mohou bránit, jejich implementace však nemusí být vždy snadná.

Článek je krácenou a upravenou verzí článku [8].

Literatura

[1] K. J. Higgins. *CSRF Flaws Found on Major Websites*. DarkReading, 2008.

http://www.darkreading.com/security/appsecurity/showArticle.jhtml?articleID=211201247

[2] Thomas Schreiber. *SESSION RIDING: A Widespread Vulnerability in Today's Web Applications*. SecureNet whitepaper. prosinec 2004. http://www.securenet.de/papers/Session_Riding.pdf

[3] *OWASP Testing Guide: Testing for CSRF*. http://www.owasp.org/index.php/Testing_for_CSRF

[4] D. Stuttard, M. Pinto. *The Web Application Hacker's Handbook – Discovering and Exploiting Security Flaws*. ISBN: 0470170778

[5] Wikipedia. *Cross-site scripting*. http://en.wikipedia.org/wiki/Cross-site_scripting

[6] W. Keller, E. W. Celtem. *Cross-Site Request Forgeries: Exploitation and Prevention*. září 2008. http://www.freedom-to-tinker.com/sites/default/files/csrf.pdf

[7] T. Bratusa. *Linksys WRT54 GL Session Riding (CSRF)* http://www.securiteam.com/securitynews/5TP032ON5U.html

[8] J. Dobiáš, Z. Říha. *Session Riding*. DSM, Praha: Tate International, s.r.o., XIII, 1, s. 40–42. ISSN 1211-8737. 2009 □