Jak dostat webshell na server

Vložil/a RubberDuck [1], 22 Leden, 2010 - 17:45

Cílem tohoto dokumentu bude shrnout (nikoliv detailně popsat) 9 běžných způsobů, jakými útočníci vytváří na serveru webshelly, které jim umožňují dále pracovat se serverem.

Obsah:

Otázka, kterou si dřív nebo později položí každý útočník-začátečník. Jako vždy záleží jen na kreativitě a vědomostech útočníka.

Způsob první: Nezabezpečený upload

K tomu není moc co dodávat. Nejjednodušší forma útoku těžící z důvery autora. Někdy bývá problém najít přesné umístění složky, ve které se uploadovaný soubor/skript nachází.

Způsob druhý: Remote File Inclusion (RFI)

Dříve poměrně hodně populární způsob kompromitace systému oblíbený pro svou nenáročnost. Je způsoben špatným ošetřením vstupů pro funkce typu include (resp. include_once), require (resp. require_once) a jiné, jež zahrnují obsah souboru uvedeného v parametru do souboru, ve kterém se funkce nachází. Někteří administrátoři se z lenosti uchylují k opatření, kdy zakazují načítání souborů mimo daný server, což se většinou rovná pokusu lovit slona jehlou: Teoreticky to jde, prakticky se to nikomu ještě nepovedlo.

Způsob třetí: Local File Inclusion (LFI)

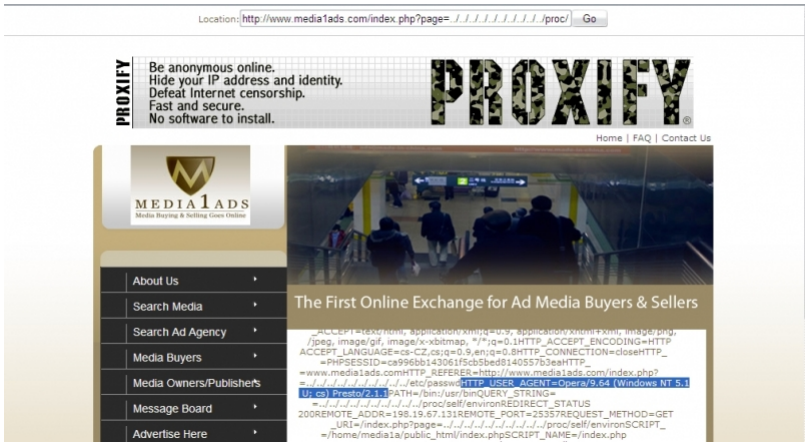
Obdonmý případ jako u RFI s tím rozdílem, že útočník provádí útok ze stejného serveru. Největším zrádcem je v tomto případě adresář /tmp. Ten je v drtivé většině případů společný pro všechny. Z toho vyplývá možnost sem uložit webový shell, který budeme pomocí zranitelnosti LFI inkludovat.

Způsob čtvrtý: Využití funkce system a jí podobných

Někdy se může stát, že útočník nebude mít kvůli direktivě open_basedir přístup k adresáři, kde plánuje umístit svůj webshell. Může se pokusit využít funkci system (resp. exec, passthru, popen, escapeshellcmd, pcntl_exec) a vytvořit webshell s pomocí systému.

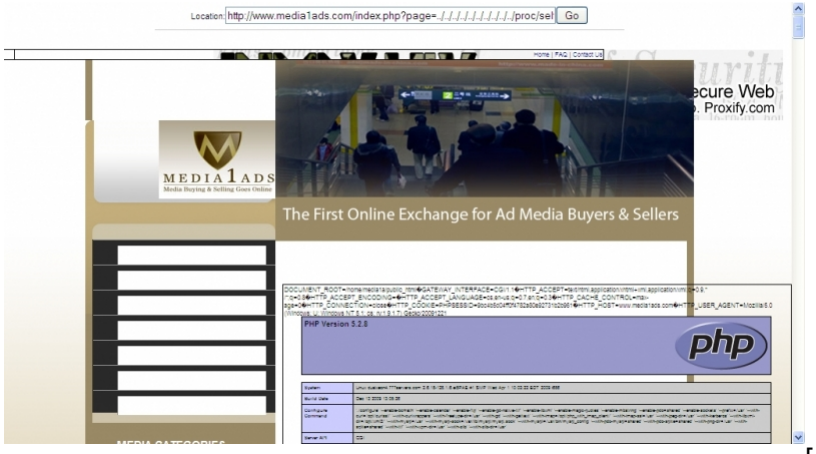
Způsob pátý: Využití prostředí

Adresář /proc/self/environ je defaultně je přístupná jen účtu s root oprávněním. Obsahuje některé informace o systému, stejně jako informace o zaslaném požadavku, mezi kterými může být i user agent daného browseru. Pokud je tento soubor načten přes LFI a prohlížeč má pozměněného user agenta tak, že obsahuje PHP kód (např.: <?php phpinfo();?>), může být tento PHP kód vykonán.



[2]

Náchylná stránka



[3]

Po injekci PHP kódu

Způsob šestý: Využití /etc/passwd

Někdy nemá útočník v daném účtu možnost zapisovat. To ovšem neznamená, že to nepůjde v nějakém jiném účtu. Přes /etc/passwd si útočník zjistí seznam všech účtů na serveru a postupně zjišťuje, ve kterém z nich může zapisovat.

Způsob sedmý: Error Log Injection

Bývá zvykem logovat všechny chyby do nějakého error_logu. Útočník může zaslat neplatný HTTP požadavek (řekněme, že bude obsahovat PHP kód <?php phpinfo(); ?>), který nutně povede k chybě a zaznamenání popisu této chyby. Pokud pak bude tento error_log načten pomocí LFI, může dojít k vykonání PHP kódu.

Způsob osmý: Apache Log Injection

Má podobný průběh jako Error Log Injection. Občas bývá problém nalézt lokaci, kam se tento log

ukládá.

Způsob devátý: MySQL Injection

Tento způsob jsem popisoval ve svém článku o SQL injection [4]. Využívá schopnosti MySQL ukládát vybraná data z tabulky do souboru.

Poznámka: Plánuji tento článek pozvolna aktualizovat. Podle nálady :)

URL článku: https://www.security-portal.cz/blog/jak-dostat-webshell-na-server

Odkazy:
[1] https://www.security-portal.cz/users/rubberduck
[2] https://www.security-portal.cz/sites/default/files/proc1.jpg
[3] https://www.security-portal.cz/sites/default/files/proc2.jpg
[4] https://www.security-portal.cz/clanky/sql-injection-full-paper