

Kovářova kobyla ...

Radoslav Bodó

Západočeská univerzita v Plzni

Centrum informatizace a výpočetní techniky

email: `bodik@civ.zcu.cz`

24. dubna 2007

Abstrakt

V dnešní době rychlého a hojně rozšířeného internetu jsou hackerské útoky na denním pořádku. Díky kvalitním vyhledávačům je navíc velmi snadné najít si různé informace o zdokumentovaných technikách, hotových nástrojích a nezřídka i potenciálních obětech útoku. V tomto příspěvku ukážeme případovou studii forenzní analýzy napadeného počítače s OS Linux.

1 Úvod

Bezpečnostní incidenty začínají v praxi několika způsoby. Buď přijde oznámení cizího subjektu o útocích nějakého stroje, nebo hlášení uživatele o podezřelém chování, nebo přijde hlášení od některého z IDS¹ systémů domovské sítě. Podle důležitosti daného zařízení provede bezpečnostní technik průzkum (automatický nebo osobní), na jehož základě je vydáno patřičné rozhodnutí o dalších akcích vedoucích k vyřešení události. Tomuto průzkumu a interpretaci nalezených údajů se říká *forenzní analýza*.

2 Získání dat

Nejdříve je nutné vyhledat uživatele daného PC a vyzpovídat ho, kdy a po jakých akcích se závadné chování objevilo, nebo alespoň na jaké období by mohl mít podezření. Dále se musí zjistit k jakým účelům byla stanice používána a případně porovnat uvedená data se statistikami datového provozu v domácí síti. Ta mohou odhalit IP útočníka nebo dalších obětí.

V tomto případě bylo hlášeno podezřelé chování uživatelem a v IDS Snort[8] domovské sítě byl nalezen záznam o úspěšném spuštění programu `bin/id` v rámci komunikace s webovým serverem, který na daném stroji běžel.

¹Intrusion Detection System

K vytvoření duplikátu je potřeba nějaké LiveCD⁴ (Knoppix, Helix, ...) a úložiště kam data uskladnit (jiné PC popřípadě externí disk). Po spuštění LiveCD, se pomocí `dd` a `nc` (ev. `ssh`) vytvoří kopie lokálních disků a zapíše se informace o rozložení dat na nich uložených. Je vhodné (nikoli však nutné) sejmut obrazy jednotlivých oddílů zvlášť (kvůli snazší manipulaci) a pořídit si i kontrolní součty zkoumaných dat. Z kopií je možné nakonec uložená data číst a provádět libovolné analýzy.

```
sklad$ nc -l -p 1234 > /tmp/mistik.hda1
```

```
mistiKnoppix$ dd if=/dev/hda1 | nc sklad 1234
mistiKnoppix$ cat <hda1>/etc/fstab >sklad> mistik.fstab
mistiKnoppix$ fdisk -l /dev/hda >sklad> mistik.fdisk
mistiKnoppix$ md5sum /dev/hda1 >sklad> mistik.hda6.md5
```

```
sklad$/dukazy/mistik# mount -o ro,loop,noexec -t ext3 mistik.hda5 mounts/usr
```

3 Odhalení průnikové cesty

Výchozími body pro forenzní analýzu jsou: typ bezpečnostního incidentu, zachovaný stav OS, očekávaný stav OS a dochované záznamy o činnosti – logy. Použité vyšetřovací nástroje závisí na zvyklostech vyšetřujících, ale nejčastější jsou jimi klasické utility: `ls`, `od`, `grep`, `find`, `md5sum`, `ldd`, `strings`, `dd`, `file`, `lsattr`, `objdump`, ..., případně specializované nástroje pro hledání dat v neobsazeném prostoru na disku (Coroner's Toolkit, Sleuth Autopsy)[1].

Jako první se doporučuje prohlídka systémových logů, avšak s vědomím, že informace v nich nemusí být pravdivé. Další kontroly se měly zabývat následujícími položkami:

- systémová nastavení (`etc`)
- spouštěcí skripty (`rc`, `inittab`, ...)
- kontrola záměny systémových nástrojů (`ls`, `ps`, `netstat`, ...)
- `suid` a `sgid` programy a adresáře nejlépe pomocí ověřených kontrolních součtů z distribučního média
- důležitá místa jako `/tmp`, `/bin`, ...
- moduly jádra operačního systému
- moduly autentizačního subsystému
- profily a historie příkazů všech uživatelů
- skryté či nezměnitelné (`immutable`) soubory
- soubory podezřelých vlastníků (numerické hodnoty)
- soubory a adresáře vlastněné provozovanými službami
- ...

Podle výše uvedeného je prvním krokem v popisovaném případě prozkoumání logu webového serveru a speciálně vyhledání záznamů související s IP nalezenou v IDS (plus

⁴Lokální systém mohl být pozměněn tak, aby útočníka schovával

následné prohledání souborů souvisejících se službou www na typické útoky php-injection⁵, sql-injection a soubory touto službou/uživatelé vytvořené). V protokolu jsou nalezeny požadavky na soubor, který do systému zřejmě nepatřil (na forenzním duplikátu navíc nebyl vůbec nalezen).

```
---- cut var/log/apache2/access.log ---
access.log:195.34.xxx.96 - - [13/Jan/2007:17:34:23 +0100] "POST /ftp/incoming/z.php HTTP/1.0" 200 34840
access.log:195.34.xxx.96 - - [13/Jan/2007:17:34:49 +0100] "POST /ftp/incoming/z.php HTTP/1.0" 200 34974
...
access.log:195.34.xxx.96 - - [13/Jan/2007:17:59:59 +0100] "GET /ftp/incoming/z.php HTTP/1.0" 200 31099
---- cut var/log/apache2/access.log ---
```

Jinými slovy z toho vyplývá, že útočník dostal pravděpodobně nějak na cílový stroj vlastní skript/program, který mohl vzdáleně volat (úspěšně viz 200 OK HTTP/1.1). Z uvedených záznamů je patrné, že byl umístěn někde v adresáři, který souvisel s další službou na stroji provozovanou – FTP serverem. Jeho nastavení ukázalo, že adresář byl přístupný pro čtení i zápis anonymním uživatelům.

```
---- cut var/log/auth.log ---
Dec  5 17:11:49 mistik proftpd[7734]: mistik.zcu.cz (proxy.lipetsk.ru[195.34.xxx.96]) -
      ANON anonymous: Login successful.
Jan  8 04:31:56 mistik proftpd[29866]: mistik.zcu.cz (crawl-66-249-66-10.googlebot.com[66.249.xxx.10]) -
      ANON anonymous: Login successful.
---- cut var/log/auth.log ---
```

V tomto okamžiku je téměř jisté, že útočník ovládl stroj pomocí *konfigurační chyby*. Přes FTP nahrál na server program, který byl schopen spustit pomocí WWW serveru a pod jeho identitou. V nastavení serveru a interpretru PHP nebylo omezeno prakticky vůbec nic (chroot, safe_mode, open_basedir, disabled_functions ...), takže činnost útočného skriptu/programu nebyla ničím limitována a mohla pohodlně využít všech dostupných prostředků a SW vybavení.

Průzkum dalších logů (*wtmp*, *xferlog*, *auth.log*) dokazuje, že útočník nebyl zřejmě pouze jeden a navíc že o této nebezpečné shodě nastavení dokázal informovat i Google kohokoliv, kdo věděl co má hledat (viz obrázky 1 a 2). Víc vodítka v podobě podezřelé IP adresy neposkytlo.

Následný podrobný průzkum logů a systému odhalil skript */tmp/back*, který zevnitř stroje otevře shell po tcp na zadanou adresu a tím obejde jak nastavení většiny současných lokálních firewallů, tak i autentizaci potřebnou k přístupu k terminálové službě – tzv. *backconnect*. Dále prohlídka odhalila masivní volání skriptu *r57.php*, který byl na disku nalezen a podroben zkoumání. V poslední řadě pak odhalil informaci, že jeden z útočníků používá prohlížeč Opera v anglickém jazyce.

4 Nalezené nástroje

Nalezený skript *r57.php* (dále jen R57) je velmi pěknou ukázkou práce současných hackerů. R57 je jedním z veřejně dostupných útočných PHP skriptů⁶, které mají za úkol zjednodušit

⁵Například `egrep -i "http://" /var/log/apache2/*`

⁶C99, PHPshell, mambot, ...



Obrázek 1: Vyhledání oběti na Googlu

```
---- cut var/log/apache2/access.log ---
error.log.1:[client 195.34.xxx.96] script '/mnt/parta/ftp/incoming/r57.php' not found or unable to stat,
referer: http://www.google.com/search?q=allinurl:r57.php&hl=en&lr=&client=opera&rls=en&hs=22Y&start=20
access.log.2:87.126.xx.17 - - [17/Dec/2006:14:17:38 +0100] "GET /ftp/incoming/r57.php HTTP/1.1" 200 32295
---- cut var/log/apache2/access.log ---
```

Obrázek 2: Ukázka logu typu apache combined

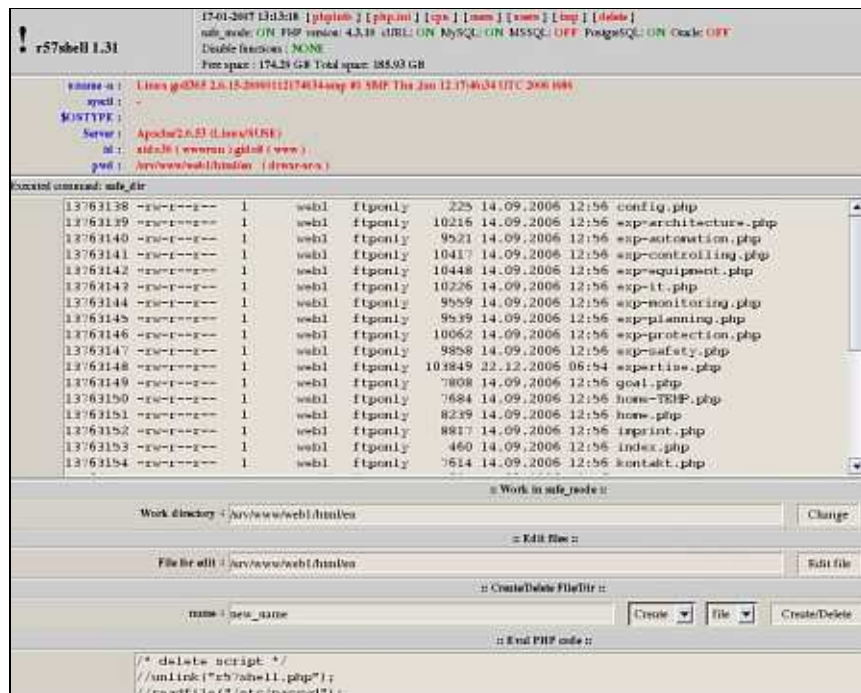
ovládání napadeného stroje, umožnit prvotní průzkum napadeného systému a urychlit další rozšiřování sama sebe. R57 je rovnou psán dvojjazyčně (anglicky a rusky) a uživateli umožňuje (obrázky 3 a 4):

- listování adresářů
- editování stávajících a nahrávání nových souborů
- pohodlné vyhledávání souborů pomocí předdefinovaných příkazů find
- procházení databází Mysql, MS-SQL, PostgreSQL
- ftp a email klient, lámání ftp hrubou silou
- automatické promazání /tmp

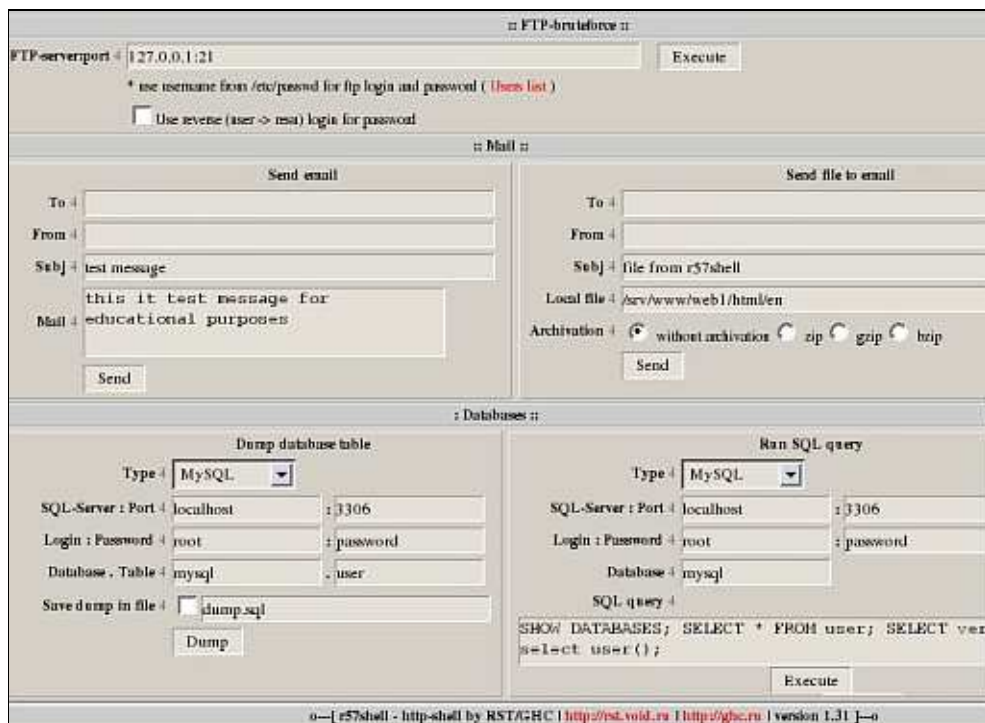
Navíc obsahuje R57 užitečné funkce v podobě zdrojových kódů programů (pro C a Perl) umožňující:

- připojit /bin/sh na určený lokální TCP socket — *bindport*
- připojit /bin/sh na spojení iniciované zevnitř stroje na zadaný cíl — *backconnect*
- vytvořit TCP tunel skrz napadený server — *datapipe*

Tyto utility jsou v R57 uloženy v kódování Base64, aby se mohly dobře integrovat do nosného skriptu a jednoduše tak obejít problémy se závorkováním a uvozovkováním a teoreticky může R57 v této podobě nosit i binární programy. V sobě mají implementováno ukrývání svých pravých názvů přepisováním proměnné *argv[0]* a funkce v nosném skriptu umožňují jejich pohodlnou kompilaci a spuštění přímo z webového rozhraní.



Obrázek 3: Náhled r57shell.php - 1



Obrázek 4: Náhled r57shell.php - 2

Hypoteticky lze vytvořit další skript (za pomoci funkce *datapipe*), který dokáže vytočit krycí spojení přes několik uzlů v internetu, a navíc tyto uzly najít online pomocí Googlu. Tyto dotazy se Google snaží blokovat.

Nicméně ani tvůrci tohoto útočného skriptu nedají nic jenom tak. Ve veřejné variantě R57 jsou další dvě pole, která obsahují javascriptový kód (též v Base64), odmaskované kódy jsou vloženy při zobrazení v prohlížeči do stránky a uživatel skriptu se tak sám nahlásí autorům a to pomocí veřejného webového počítačidla. V konečném důsledku mohou autoři sledovat jak napadené servery, tak některé uživatele svého výtvoru.

5 Získání superuživatelských práv

Pokud je napadený stroj důležitou součástí infrastruktury sítě, případně obsahuje-li citlivá data pro napadenou organizaci, bývá zvykem ve vyšetřování dále pokračovat i po prokázání počátečního napadení. Důvodem je odhad možných škod, které útočník napáchal, či odhad důsledků jeho činnosti. Technicky bývá další hledání zaměřeno na důkazy o prolomení *uid=root*, s jehož oprávněním je útočník schopen odcizit veškeré dokumenty na stroji uložené, pokusit se zachytit uživatelská hesla nebo odposlouchávat síťový provoz na lokálním segmentu, ... Tento postup se v zásadě neliší od postupu popisovaném v kapitole 3.

Ve zkoumaném systému byla nalezena zadní vrátka v podobě nastaveného hesla pro uživatele *uid=99* (*apache*), ten nebyl v systému oprávněně a dokazuje to, že měl útočník superuživatelská práva. V popisovaném případě bylo však původně napadnuté *uid=33* (*www-data*), webová služba.

Pomocí příkazu `'find / -user www-data'` byl nalezen adresář, v němž si útočník nechal další nástroj:

```
---- cut 'ls -lia /sbin/apache' ---
total 19
30230 drwxrwxrwx 2 root    root    1024 2007-01-14 10:54 .
20098 drwxr-xr-x 3 root    root    5120 2007-01-15 16:21 ..
30320 -rw-r--r-- 1 www-data www-data 1362 2007-01-13 18:27 pack.tgz
30313 -rw-r--r-- 1 www-data www-data 1790 2007-01-14 18:25 passwd
30370 -rwsrwxrwx 1 root    root    5295 2007-01-14 09:03
s~30369 -rw-r--r-- 1 www-data www-data 1460 2007-01-14 18:25 shadow
30374 -rw-r--r-- 1 www-data www-data 722 2007-01-14 10:54 shadow.tgz
---- cut 'ls -lia' ---
```

Je zde k vidění *suid* program */sbin/apache/s*. Atribut *suid* je označení umožňující uživateli vykonat takový program s dočasně jinými právy, než jaká byla uživateli přidělena (viz *uid* vs. *euid*). Pomocí *strings* (vytažení tisknutelných řetězců z binárního souboru) je možné odhadnout jeho účel. Zkoumaný */sbin/apache/s* je velmi krátký a dle získaného výpisu slouží nejpravděpodobněji ke spuštění jiného programu s právy *roota*, viz volání *setuid()*, *setgid()* a *system()*. Totéž prokáže i zpětný překlad (*disassembling*). Zbývá zjistit, jak příslušný program útočník vyrobil, čili je třeba najít důkaz, nebo alespoň nalézt nějakou lokálně zneužitelnou chybu zkoumaného systému. Těch existuje daleko více než vzdálených a pátrání se v tomto případě zaměřilo na ostatní *suid* programy a vyhledání jejich publikovaných chyb:

```

---- cut ---
$ find . -type f -perm -04000 -ls
20124 16 -r-sr-xr-x 1 root root 15000 ^en 28 2004 ./root/sbin/unix_chkpwd
30370 6 -rwsrwxrwx 1 root root 5295 led 14 09:03 ./root/sbin/apache/s
29829 36 -rwsr-xr-x 1 root root 35512 srp 12 20:05 ./root/bin/login
...
990617 796 -rwsr-xr-x 1 root root 809836 pro 8 23:29 ./usr/bin/gpg
...
$ find . -type f -perm -02000 -ls
...
---- cut ---

```

Dle výpisu mohlo jít tedy o zneužití jedné ze dvou chyb v GPG. Instalovaný SW je datovaný na 8.12.2006 a o den později vyšlo DSA-1231⁷ oznámení o dvou chybách, které mohou vést ke spuštění podstrčeného kódu pomocí chyby přetečení zásobníku. Nicméně důkaz o zneužití této chyby nebyl nalezen a nebyl nalezen ani publikovaný exploit nebo jeho proof-of-concept.

Nakonec vše prozradil *process accounting* v podobě databáze programu *atop*. Z výpisu je možné vidět spouštění programu *getsuid* identitou *www-data*, program do systému nepatří a jeho název naznačuje vytvoření programu */sbin/apache/s*.

```

---- 'atop -r var/log/atop.log' ---
12556 www-data www-data 2007/01/08 15:36:59 -- S~0% apache2
22461 www-data www-data 2007/01/10 18:28:45 -- S~0% apache2
...
? www-data www-data 2007/01/13 17:35:25 NE 0 E 0% <chmod>
? www-data www-data 2007/01/13 17:35:38 NC 11 E 0% <getsuid>
? www-data www-data 2007/01/13 17:35:38 NE 0 E 0% <getsuid>
? www-data www-data 2007/01/13 17:34:06 NE 0 E 0% <id>
? www-data www-data 2007/01/13 17:34:07 NE 0 E 0% <ls>
? www-data www-data 2007/01/13 17:34:24 NE 0 E 0% <ls>
? www-data www-data 2007/01/13 17:34:49 NE 0 E 0% <ls>
---- 'atop -r var/log/atop.log' ---

```

Podle jména je pak jednoduché najít hotový exploit za pomoci vyhledávače:

- <http://www.ykzj.org/article.php?articleid=2009>
- <http://www.securityfocus.com/bid/18874>

Tento exploit využívá lokální chybu linuxového jádra v systémovém volání *prctl()* (process control). Chyba umožní vytvoření souboru *core*⁸ i v adresáři, ve kterém nemá běžící program právo zápisu. Exploit tedy vytvoří dva procesy, z nichž jeden nechá spadnout v adresáři */etc/cron.d* tak, aby cron (běžící s rootovskými právy) dokázal interpretovat text v *coredump*. V padajícím programu je zakompilován řetězec:

```

---- cut getuid.c ---
char *payload="\nSHELL=/bin/sh\nPATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin\n
* * * * * root chown root.root /tmp/s ; chmod 4777 /tmp/s ; rm -f /etc/cron.d/core\n";
---- cut getuid.c ---

```

⁷Debian Security Advisory - <http://www.debian.org/security/2006/dsa-1231>

⁸coredump - obraz procesu po pádu určený ke zjištění příčiny chyby v debuggeru

Program `/tmp/s` byl na forenzním duplikátu nalezen. Nakonec bylo potřeba pouze prokázat použití tohoto exploitu. To se povedlo nalezením smazaného coredumpu (je nalezen blok ve kterém je řetězec obsažen):

```
---- cut ---
sklad @ /dukazy/mistik# grep -iab "/etc/cron.d/core" mistik.hda1
29158161:* * * * *   root    chown root:root /tmp/s ; chmod 4777 /tmp/s ; rm -f /etc/cron.d/core
---- cut ---
```

Funkčnost exploitu byla úspěšně odzkoušena ve virtuálním pískovišti⁹. V této chvíli bylo prokázáno, že k získání superuživatelských práv byla zneužita lokální chyba jádra *Linux Kernel PRCTL Core Dump Handling Privilege Escalation Vulnerability* [10].

6 Závěr

K tomuto incidentu došlo díky nedostatečnému zabezpečení provozovaných služeb:

- anonymní přístup na FTP server s možností zápisu do adresáře publikovaného pomocí webové služby,
- nezabezpečené PHP,
- webový server neběžel v odděleném prostředí, které by útočnickovi ztěžovalo průnik do systému (jail/chroot),
- neaktualizované jádro operačního systému.

Stroj byl napaden již v listopadu roku 2006, tedy minimálně dva měsíce před odhalením. V této konfiguraci byl však provozován cca 3 roky, čili přesné datum průniku není možné přesně určit.

K odhalení došlo víceméně náhodou (provozní problémy stroje podnítily hlubší průzkum), také k tomu velmi dopomohl fakt, že útočníci po sobě prakticky nezametali.

Služba IDS Snort, kterou v domovské síti provozujeme, přinesla jeden z prvních praktických výsledků a bude nadále rozšiřována, protože mohla přinést varování o průniku automaticky.

Troufám si tvrdit, že chybu jádra CVE-2006-2451 obsahuje velké procento současných produkčních strojů napříč celým internetem. Proti této a jiným podobným chybám jsou jedinou obranou včasná varování od systémů IDS, správná konfigurace a zabezpečení služeb tak, aby nedošlo k úvodnímu napadení.

Jako poznámku na závěr a zamyšlení, bych si dovolil podotknout, že vyhledávací služby jsou šavle broušené na mnoho stran ...

⁹testovacím prostředí VMware

Literatura a odkazy

- [1] Bc. Josef Kadlec: *Forenzní analýza unixových systémů*
<http://jose.dump.cz/diploma.html>
- [2] Dave Dittrich: *Basic Steps in Forensic Analysis of Unix Systems*
<http://staff.washington.edu/dittrich/misc/forensics/>
- [3] Sam Stover, Matt Dickerson: *Using memory dumps in digital forensics*
;login:, vol. 30 no. 6, pp. 43-48
- [4] Glenn M. Brunette, Jr.: *Hiding within the trees*
;login:, vol. 29 no. 1, pp. 48-54
- [5] Boris Loza: *Under Attack*
;login: 2005, vol.30. num. 3
- [6] Boris Loza: *Finding trojans for fun and profit*
;login:, vol. 30 no. 5, pp. 19-22
- [7] Michal Vyskočil: *LiveCD a jejich použití*
Sborník příspěvků z XXVIII. konference EurOpen.CZ, 2006
- [8] Snort: Intrusion Detection system
<http://www.snort.org/>
- [9] BASE: Basic Analysis and Security Engine
<http://sourceforge.net/projects/secureideas>
- [10] Linux Kernel PRCTL Core Dump Handling Privilege Escalation Vulnerability
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2451>
<http://www.securityfocus.com/bid/18874>