

[TOOL] r57-pid-check

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-04/msg00011.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxxx>
 - *Date:* 5 Apr 2006 14:18:15 +0200
-

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

r57-pid-check

SUMMARY

DETAILS

r57-pid-check is a tool written in perl for Unix based operating systems that finds hidden PID including rootkits, by using the system calls: kill() and setpriority().

Tool Source:
#!/usr/bin/perl

```
use Getopt::Std;  
use Fcntl;  
use Time::HiRes qw(usleep);
```

```
$r00t=0;  
$SUCCESS=0;  
$clean=0;  
$pause=50000;  
$scanner="pidcheck";
```

[TOOL] r57-pid-check

```
$method='kill -9';
$sf0="/tmp/r35u1t0";
$sf="/tmp/r35u1t";
$sf2="/tmp/r35u1t2";
$sys="/var/log/messages";
$start = time();
@jail=();
@jail2=();
@only=();
@only_proc=();

getopts("m:bhr");

sub usage
{
print"Usage: $0 -m [MAX number PID]\n";
print"Now type: $0 -h\n\n";
}

sub help
{
print qq!
[~] First - you must create CLEAN, UNPRIVILEGED user pidcheck,
home - /dev/null, shell - /bin/sh, with locked password

[~] For interactive check type $0 -m [number PID to scan]
default method kill() - command "kill -9"
# $0 -m 5000
and you check all PID to 5000

[~] You can use method setpriority() system call, is
command "renice -20", option -r
# $0 -m 5000 -r

[~] Background check (output in /var/log/messages)
use option -b
# $0 -m 5000 -b &

If nothing found, then in log:
[+] r57-pid-check.pl: Check PID, hidden PID not found.
[+] Time check: some time
Else all info write to system log.

[~] Testing on:
Linux 2.4.x (rootkits: adore-0.42, adore-ng-1.41)
Linux 2.6.x - quite possible work
FreeBSD 5.x - quite possible work
OpenBSD 3.x - quite possible work
\n\n!
}
```

```
sub head
{
print qq!
~~~~~
Find hidden PID, even rootkit installed.
Use system call's: kill(), setpriority().
Gr33tz: blf, 1dt.w0lf, edisan, foster,
Pengo, Dr_UF0_51.
(c)oded x97Rang, RST/GHC 2006
http://rst.void.ru
http://ghc.ru
~~~~~
\n!
}

sub get_os
{
if(!$opt_b) {print"[+] OS: $^O\n";}
if($^O eq "freebsd" || $^O eq "openbsd")
{
$mode = ">";
}
elseif($^O eq "linux")
{
$mode="&>";
}
else
{
print"[-] Test only FreeBSD, OpenBSD and Linux\n";
exit;
}
}

sub get_uid
{
if($< != $r00t)
{
print"[-] For use this you need UID=0\n";
exit;
}
system("id pidcheck $mode /dev/null");
if($? != $SUCCESS)
{
print"[-] You must add to system user pidcheck, type $0
-h for help\n";
exit;
}
}
```

```

sub do_it
{
if($opt_m =~ m/\d+/ && $opt_m > 20){
if(!$opt_b) { print"[~] Begin scan PID's to $opt_m\n";}
if(!$opt_b && !$opt_r) { print"[~] Try use kill()\n";}
if(!$opt_b && $opt_r) { print"[~] Try use setpriority()\n";
$method='renice -20';}
if($opt_b && $opt_r) { $method='renice -20';}
for($n=20;$n!=$opt_m;$n++)
{
if(!$opt_b) {status();}
system("ps aux | awk '{print \$1,\" \",\$2}' | grep -w $n |
grep $scanner > $sf0");
if(-s $sf0){ next;}
system("su $scanner -c '$method $n' 2> $sf");
usleep $pause;
sysopen(TF, $sf, O_RDONLY) or die "Couldn't open $sf for
reading: $!\n";
while ($line=<TF>)
{
if($line =~ m/permitted/)
{
system("ps ax -o pid | grep -w $n >
$sf2");
if(-z $sf2)
{
$clean=1;
push(@jail,$n);
}
}
}
close(TF);
}
} else { &usage; exit;}
}

```

```

sub in_proc
{
{
if(!$opt_b){ print"\n[~] Check vfs /proc\n";}
for($n=1;$n!=$opt_m;$n++)
{
if(!$opt_b) {status();}
system"test -d /proc/$n";
if($? == $SUCCESS)
{
system("ls -F /proc/ | grep '$n/' > $sf2");
if(-z $sf2)
{
$clean=1;
push(@jail2,$n);
}
}
}
}

```

```
}  
}  
}  
}  
}  
  
sub last_chance  
{  
if($clean != $SUCCESS)  
{  
for($i=0;$i<=#jail;$i++)  
{  
system("ps aux | awk '{print \$1, \" \", \$2}' | grep -w  
$jail[$i] | grep $scanner > $sf0");  
if(-s $sf0){ next;}  
system("su $scanner -c '$method $jail[$i]' 2> $sf");  
sleep(1);  
sysopen(TF, $sf, O_RDONLY) or die "Couldn't open $sf  
for reading: $!\n";  
while ($line=<TF>)  
{  
if($line =~ m/permitted/)  
{  
system("ps ax -o pid | grep -w  
$jail[$i] > $sf2");  
if(-z $sf2)  
{  
push(@only,$jail[$i]);  
}  
}  
}  
close(TF);  
}  
  
if($jail2[0])  
{  
for($j=0;$j<=#jail2;$j++)  
{  
system("test -d /proc/$jail2[$j]");  
sleep(1);  
if($? == $SUCCESS)  
{  
system("ls -F /proc/ | grep  
$jail2[$j]' > $sf2");  
if(-z $sf2)  
{  
push(@only_proc,$jail2[$j]);  
}  
}  
}  
}  
}
```

```

}
}

sub show_res
{
if($only[0])
{
for($k=0;$k<=$#only;$k++)
{
if($opt_b)
{
$a=time();
$b=localtime($a);
sysopen(LOG, $sys, O_WRONLY|O_APPEND) or die
"Couldn't open $sys for writing: $!\n";
print LOG "##### WARNING
#####\n";
print LOG "[!] r57-pid-check.pl\n";
print LOG "[!] Time check: $b\n";
print LOG "[!] Found invisible PID - $only[$k]\n";
close(LOG);
}
else
{
print "\n[!] Found invisible PID: $only[$k]\n";
}
}
}
if($only_proc[0])
{
for($l=0;$l!=@only_proc;$l++)
{
$x=$only_proc[$l];
if($opt_b)
{
$a=time();
$b=localtime($a);
$who=`cat /proc/$x/cmdline`;
$where=`ls -l /proc/$x/cwd`;
sysopen(LOG, $sys, O_WRONLY|O_APPEND) or die
"Couldn't open $sys for writing: $!\n";
print LOG "##### WARNING
#####\n";
print LOG "[!] r57-pid-check.pl\n";
print LOG "[!] Time check: $b\n";
print LOG "[!] Found hide PID in /proc - $x\n";
print LOG "[!] Running program: $who\n";
print LOG "[!] Current working directory of the
process: $where\n";
close(LOG);
}
}
}

```

```

else
{
print "\n[!] Found hide PID in /proc: $x\n";
print "[!] Current working directory of the
process:\n";
system("ls -l /proc/$x/cwd");
print "\n[!] Command line is:";
system("cat /proc/$x/cmdline");
print "\n[!] More info about running program:\n";
system("cat /proc/$x/status");
}
}
}
}

sub is_clean
{
unlink($sf0);
unlink($sf);
unlink($sf2);
if(!$only[0] && !$only_proc[0] && !$opt_b)
{
print "\n[+] r57-pid-check.pl: Hidden PID not found.\n";
}
if(!$only[0] && !$only_proc[0] && $opt_b)
{
$a=time();
$b=localtime($a);
sysopen(LOG, $sys, O_WRONLY|O_APPEND) or die "Couldn't open
$sys for writing: $!\n";
print LOG "[+] r57-pid-check.pl: Hidden PID not found.\n";
print LOG "[+] Time check: $b\n";
close(LOG);
}
}

sub over
{
$end = time();
$allt = $end-$start;
$nt = $allt/3600;
printf("\n[~] Time of work: %.3f h\n", $nt);
print "[~] Done.\n";
}

sub status
{
$status = $n % 5;
if($status==0){ print "\b\b/"; }
if($status==1){ print "\b\b-"; }
if($status==2){ print "\b\b\"; }
}

```

[TOOL] r57-pid-check

```
if($status==3){ print "\b\b|"; }
}

if($opt_h)
{
&head;
&help;
exit;
}
elseif(!$opt_m)
{
&head;
&usage;
exit;
}
else
{
if(!$opt_b){ &head;}
&get_os;
&get_uid;
&do_it;
if($^O eq "linux") { &in_proc;}
&last_chance;
&show_res;
&is_clean;
if(!$opt_b){ &over;}
}
# EOF
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:ghc@xxxxxx>> GHC.
To keep updated with the tool visit the project's homepage at:
<<http://rst.void.ru/download/r57-pid-check.txt>>
<http://rst.void.ru/download/r57-pid-check.txt>

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@xxxxxxxxxxxxxxxxx
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxxx

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.