Linux Digital Forensics Mərhələləri

Digital Forensics-in mərhələlərinin üç mərhələdən ibarət olduğunu demək olar:
Sübutların müəyyən edilməsi və qorunması
Sübutların araşdırılması və təhlili
Sübutların bildirilməsi

Sübutların müəyyən edilməsi və qorunması
Rəqəmsal məhkəmə ekspertizasının başladığı nöqtə ələ keçirmə prosesinin başladığı nöqtədir. Elektron sübut sayıla bilən istənilən qurğuda axtarış, götürmə və surət çıxarma əməliyyatları Cinayət Prosessual Məcəlləsinin 134-cü maddəsinə və “Məhkəmə və profilaktik axtarışlar haqqında” Əsasnamənin 17-ci maddəsinə uyğun olaraq həyata keçirilir.

Hadisə yerində aparılan araşdırmalar zamanı təhlükəsizlik hüquq-mühafizə orqanları tərəfindən təmin edilməlidir. Hadisə yerinə hər tərəfi görə bilmək üçün kameralar yerləşdirilməli, səhnənin müxtəlif bucaqlardan fotoşəkilləri çəkilməlidir. Cinayət yerində kompüter ekspertizasının eksperti olmalı və araşdırmalar kompüter ekspertizasının eksperti tərəfindən aparılmalıdır.

Ekspertiza zamanı sübut sayılan cihazlar aşkarlanır. Aşkar edilmiş cihazlar ayrı-ayrılıqda sübut etiketləri ilə işarələnməli və fotoşəkilləri çəkilməlidir. İnventar yaradılır və müsadirə olunan hər bir cihazın markası, modeli, seriya nömrəsi və s. məlumatlar qeydiyyat siyahısında qeyd edilməlidir. Hadisə yerində görülən bütün hərəkətlər qeyd edilməlidir. Hadisə yerində söndürülmüş cihazlar işə salınmamalıdır. Açıq cihazların müayinəsinin olacağını nəzərə alaraq, lazımi avadanlıqlar hazır olmalı və canlı müayinə zamanı həyata keçiriləcək prosedurlar qeydə alınmalı və fotoşəkilləri çəkilməlidir. Canlı analiz zamanı hər hansı bir casus proqrama qarşı diqqətli olmaq lazımdır. Alınan cihazlar ayrı-ayrı sübut torbalarına qoyulmalı və müayinə üçün laboratoriyalara aparılmalıdır.

Digital Forensics Mərhələləri

Sübutların araşdırılması və təhlili

Sübutların tədqiqi və təhlili laboratoriyalarda mütəxəssislər tərəfindən aparılır. İcra ediləcək prosedurlar ardıcıllıqla qeyd edilməlidir. Ediləcək müayinələr orijinal cihazlarda edilməməlidir. Orijinal cihazların çoxsaylı məhkəmə nüsxələri hazırlanmalıdır. Orijinal cihazın hash dəyərləri alınan məhkəmə nüsxələri ilə hesablanmalı və sübutların bütövlüyü təmin edilməlidir. Ekspertizalar məhkəmə ekspertizasının nüsxələrində aparılmalıdır. Məhkəmə ekspertizasının nüsxələri götürülərkən ümumi qəbul edilmiş kriminalistika standartlarına uyğun olaraq birtərəfli yazma qabiliyyətinə malik proqram və texniki vasitələrdən istifadə edilməlidir. Sonradan götürülmüş məhkəmə nüsxələri tədqiq ediləcək proqram təminatına yüklənməlidir. Müvafiq sübut tapmaq üçün imtahan verən bütün faylları təhlil etməlidir. Bu fayllar:

Adi fayllar
Silinmiş fayllar
Əməliyyat sistemi faylları
Şifrələnmiş fayllar
Gizli fayllar
Tətbiq faylları və s.

Sübutlar müayinə zamanı istifadə olunan proqram təminatının imkanlarına uyğun olaraq əldə edilir. Daha sonra əldə edilən sübutlar təhlil edilməlidir. Hər cür fayl, proqram, tətbiq, tarix və saat və s. üzərində olan kompüter və ya saxlama vahidləri araşdırılır. məlumatlar qeyd edilməlidir. Hadisə ilə bağlı sübutlar təhlil edilən sübutlardan ayrılmalıdır.

Sübutların Hesabatı

Rəqəmsal məhkəmə ekspertizası mərhələlərinin sonuncusu hesabat mərhələsidir. Yuxarıda göstərilən bütün proseslər ümumi qəbul edilmiş standartlara uyğun aparılmalıdır. Bütün proseslər qeydə alınmalı və hesabat verilməlidir. Sübutların qəbul edilməsi üçün onun bütövlüyü və düzgünlüyü təmin edilməlidir. Hadisə ilə bağlı əldə edilmiş sübutlar standartlara uyğun olaraq bildirilməlidir. Hazırlanmış hesabat aydın və qısa olmalı, mümkün olduqda texniki terminlərin istifadəsindən qaçınılmalıdır. Hesabatda ittiham xarakterli ifadələrdən istifadə edilməməlidir. Yalnız cinayətin törədildiyini göstərən məlumatlar açıq şəkildə ifadə edilməlidir. Zəruri hallarda istintaqı aparmış ekspert sübutlar barədə ifadə vermək üçün prokurorluq və ya məhkəmə tərəfindən çağırıla bilər.

Linux Forensics

Linux digital forensics hüquqi və ya təhqiqat prosesinin bir hissəsi kimi Linux sistemlərində tapılan rəqəmsal sübutların toplanması, təhlili və bildirilməsi prosesinə aiddir. Bu, aşağıdakı kimi geniş fəaliyyətləri əhatə edə bilər:

1) Məlumatların toplanması: Linux sistemlərindən məlumatların təhlükəsiz şəkildə toplanması, onların prosesdə dəyişdirilməməsini təmin etmək. Bu, adətən bütün fayl sisteminin bit-by-bit surətinin çıxarılmasını nəzərdə tutur.

2) Məlumatların analizi: Vacib məlumatları aşkar etmək üçün toplanmış məlumatların təhlili. Buraya fayl sistemi strukturlarına baxmaq, jurnal fayllarını yoxlamaq, silinmiş faylları bərpa etmək, sistem və istifadəçi fəaliyyətlərini təhlil etmək daxil ola bilər.

3) Zaman qrafikinin təhlili: Hadisələrin ardıcıllığını başa düşmək üçün sistem və istifadəçi fəaliyyətlərinin qrafiklərinin yaradılması.

4) Şəbəkə Təhlili: Şəbəkə bağlantılarının qeydiyyatı, şəbəkə konfiqurasiyasının təhlili və şəbəkə xidməti məlumatlarının tədqiqi daxil olmaqla, şəbəkə ilə əlaqəli fəaliyyətlərin araşdırılması.

5) Yaddaş Təhlili: Sistem vəziyyəti, işləyən proseslər və digər dinamik fəaliyyətlər haqqında məlumat toplamaq üçün sistem yaddaşının (RAM) məzmununun təhlili.

6) Artefakt təhlili: Linux sistemində qalan xüsusi artefaktlara baxmaq, məsələn, bash tarixçəsi, əmrlərin icrası, USB cihaz tarixçəsi və s. kimi istifadə nümunələrini göstərə bilər.

7) Hesabat: Nəticələrin hüquq-mühafizə orqanlarının əməkdaşları və ya hüquq mütəxəssisləri kimi texniki bilikləri olmayanlar üçün başa düşülən hərtərəfli hesabat şəklində tərtib edilməsi.

Linux Forensics zamanı araştırılmalı olan qovluqlar və fayllar

Linux kriminalistikasında əsas artefaktlar araşdırma zamanı dəyərli fikirlər verə bilən xüsusi fayllar, qeydlər və sistem məlumatlarıdır. Bu artefaktlar hadisələri yenidən qurmaq, istifadəçi hərəkətlərini başa düşmək və anomaliyaları müəyyən etmək üçün çox vacibdir. Linux kriminalistikasında diqqətəlayiq işlərdən bəziləri bunlardır:

Bash Tarixçəsi: ~/.bash_history-də saxlanılan bu fayl istifadəçilər tərəfindən Bash qabığına daxil edilmiş əmrlərin qeydini ehtiva edir. Bu fayl hücumçular tərəfindən manipulyasiya edilə bilər ona görədə önəmli fayllardan biridir. **~/.bash_profile** faylı kimi shell konfiqurasiya faylları hücumçü tərəfindən dəyişdirilə və ya zərərli kodlar əlavə edilə bilər.

Log Files: /var/log/ qovluğunda yerləşən bu fayllar müxtəlif sistem və proqram fəaliyyətlərini qeyd edir. Əhəmiyyətli qeydlərə aşağıdakılar daxildir:

auth.log və ya **audit.log**: Doğrulama və avtorizasiya məlumatlarını qeyd edir.
syslog və ya messages: Ümumi sistem fəaliyyəti qeydlərini ehtiva edir.
dmesg: Kernellə əlaqəli mesajları və səhvləri qeyd edir.
web server logları üçün **/var/log/apache2/access.log** və **/var/log/apache2/error.log** (Apache istifadə olunursa).

User və Group Məlumatları: /etc/passwd (istifadəçi hesabları) və **/etc/group** (qrup məlumatı) bölmələrində saxlanılan bu fayllar sistemdəki istifadəçilər və qruplar haqqında ətraflı məlumat verir. **/etc/shadow** faylında isə istifadəçilərin şifrələrinin sha512 və ya sha256 şifrələmə alqoritmləri ilə şifrələnmiş qeydləri vardır.

Cron job-lar: /etc/cron.* qovluğundakı fayllar və istifadəçiyə aid olan cron job-lar (crontab -l) sistemdə avtomatik və ya planlaşdırılmış hərəkətləri həyata keçirə bilən planlaşdırılmış tapşırıqları təmsil edir. **/etc/cron.d/** - **/etc/cron.daily/** - **/etc/cron.hourly/** - **/etc/cron.monthly/** - **/etc/crontab** - **/etc/cron.weekly/**

Linux Forensics zamanı araştırılmalı olan qovluqlar və fayllar

Şəbəkə Konfiqurasiyası və Qeydlər: /etc/network/, **/etc/hosts** və **/etc/resolv.conf** qovluqlarındakı fayllar şəbəkə konfiqurasiyası haqqında məlumat verir. Şəbəkə qeydləri keçmiş şəbəkə bağlantılarını və fəaliyyətlərini göstərə bilər.

SSH Qeydləri və Açarları: SSH qeydləri (/var/log/auth.log və ya /var/log/secure) və SSH açar faylları (.ssh/authorized_keys, .ssh/id_rsa və s.) sistemə uzaqdan giriş haqqında təfərrüatları təmin edir.

E-poçt və Əlaqə Qeydləri: Sistem e-poçt və ya mesajlaşma üçün istifadə olunursa, bu xidmətlərlə bağlı qeydlər və fayllar mühüm məlumatları ehtiva edə bilər.

Web Brauzer Tarixçəsi: Web brauzerlərdən istifadə edilərsə, tarix faylları ziyarət edilən vebsaytlar və onlayn görülən tədbirlər haqqında təfərrüatları təmin edə bilər.

Silinmiş Fayllar və Faylların Bərpası: Bu yaxınlarda silinmiş fayllar və ya bu cür faylları bərpa etmək cəhdləri haqqında məlumat, xüsusilə dəlilləri gizlətmək və ya silmək cəhdinin olduğu hallarda çox vacib ola bilər.

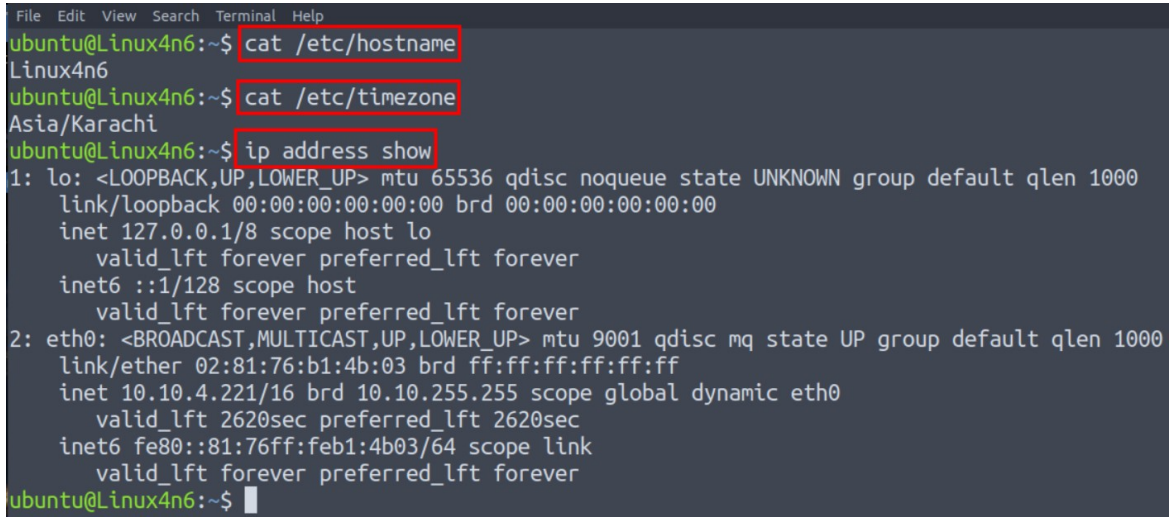
Sistem və Tətbiq Konfiqurasiya Faylları: Bu fayllar (/ və s. kataloqu) sistemin necə qurulduğu və proqramların necə konfiqurasiya edildiyi barədə kontekst təmin edə bilər.

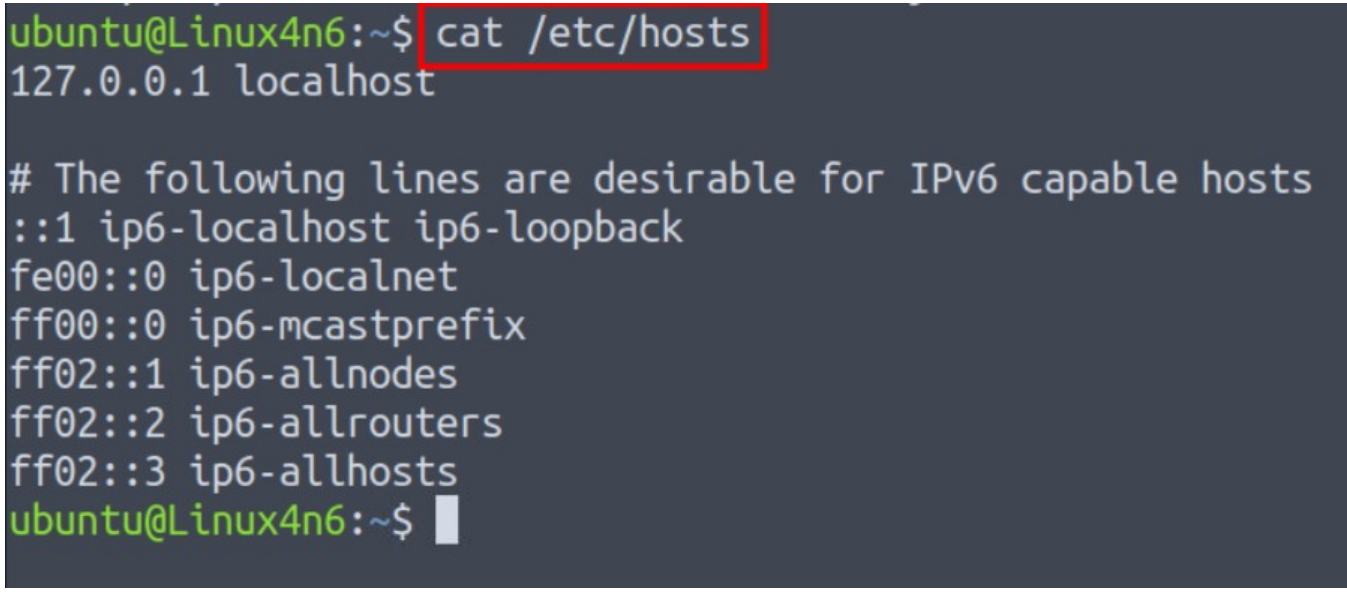
Yaddaş tullantıları: Yaddaş tullantılarının təhlili tullantıların götürüldüyü anda işləyən proseslər, açıq fayllar, şəbəkə əlaqələri və s. haqqında məlumatları aşkar edə bilər.

Linux əməliyyat sistemində Digital Forensics, bir çox sistem faylının araşdırılması ilə həyata keçirilə bilər.

Linux-da Digital Forensics Addımları

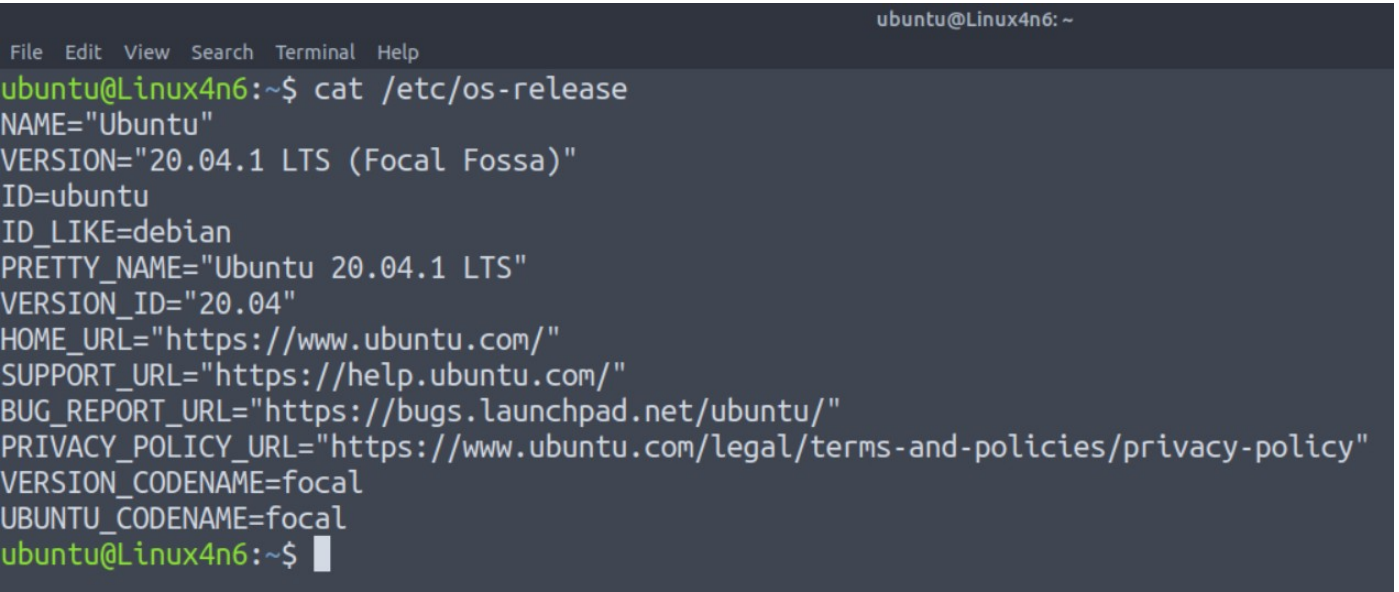
Əməliyyat sisteminin hostname-inə “**cat /etc/hostname**” commandı ilə baxa bilərik, timezone (saat qurşağı)-na “**cat /etc/timezone**” commandı ilə baxmaq olar. Daha sonra “**ip address show**” commandı ilə sistemin İP və MAC addresslərini və onların interface-lərini görə bilərik. “**cat /etc/hosts**” commandı ilə lokal DNS adının təyin edilməsi üçün olan konfiqurasiya faylını görürük, bunlar C2-ləri işarə edəcək şəkildə dəyişdirilə bilər.

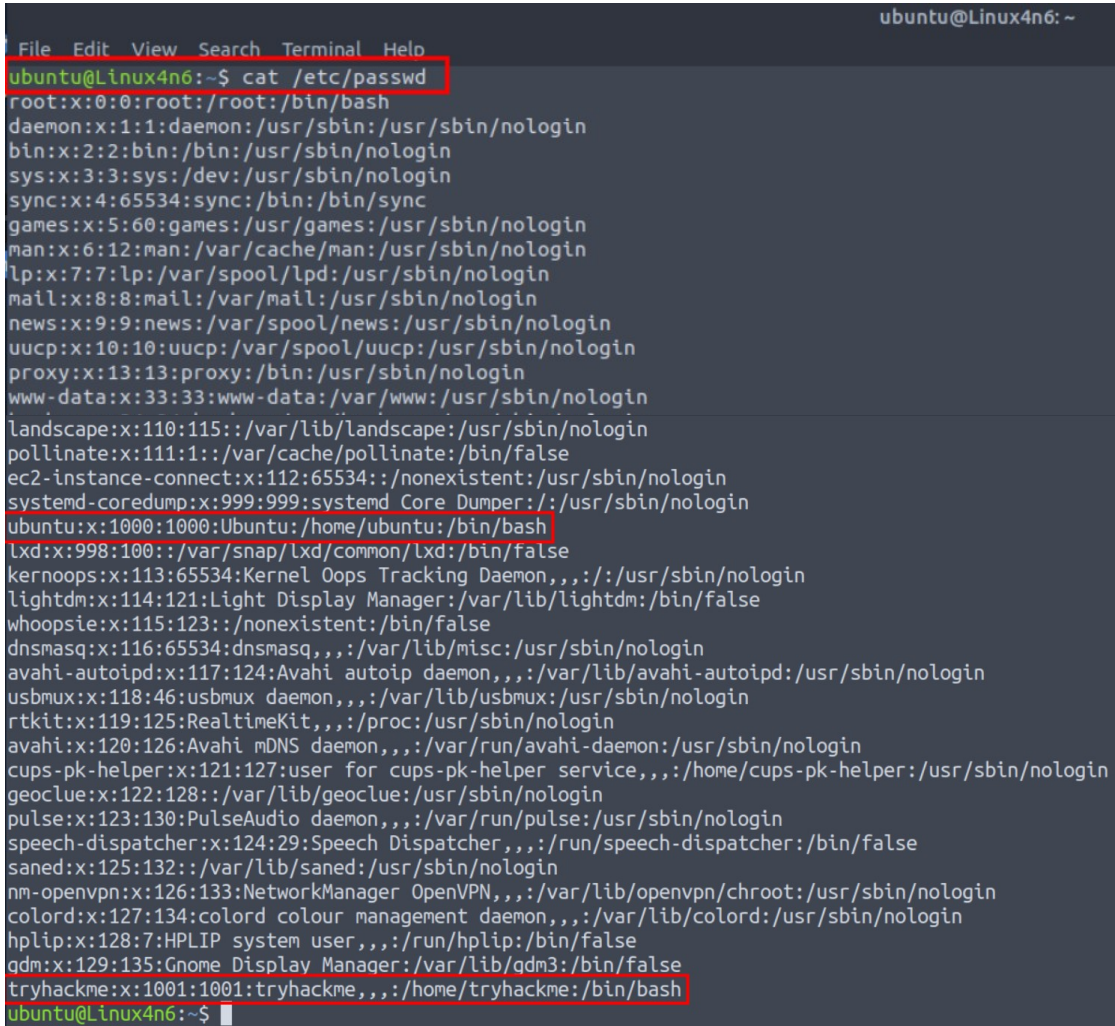




Linux-da Digital Forensics Addımları

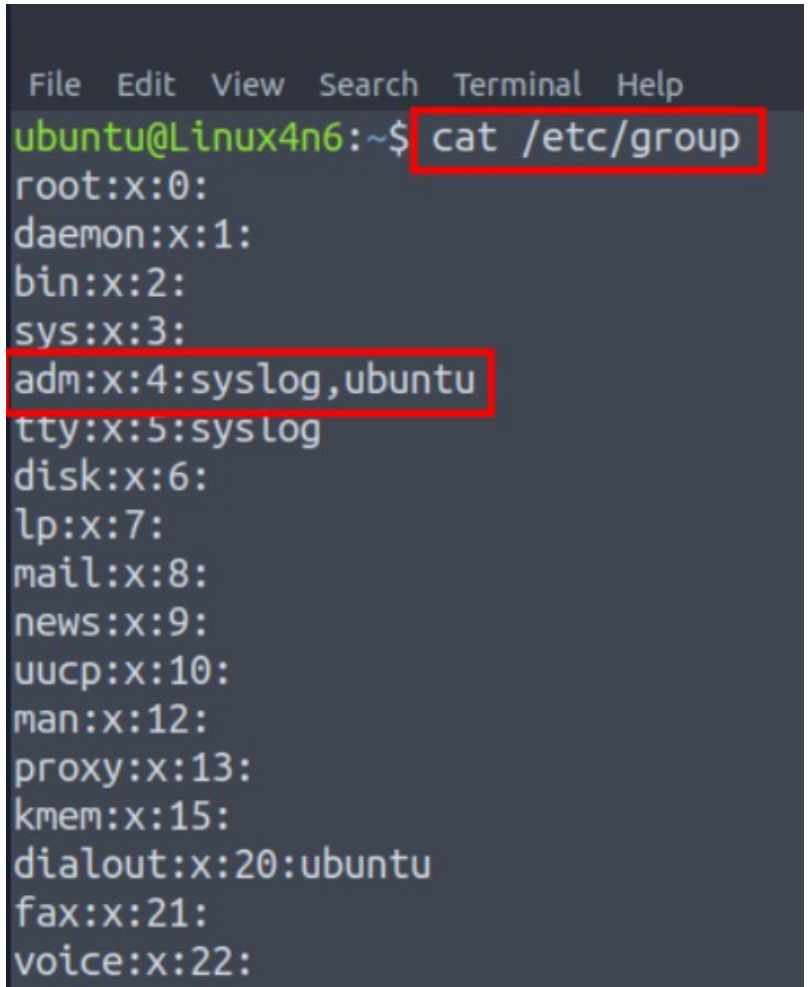
Əməliyyat sistemi haqqında məlumat əldə etmək üçün **“cat /etc/os-release”** commandını istifadə edirik. Gördüyümüz kimi Ubuntu 20.04.1 LTS Əməliyyat sistemidir. Daha sonra **/etc/passwd** faylının içinə baxaraq sistemdə olan User-ləri, İD-lərini, qruplarını və hansı Shell-dən istifadə etdikləri haqqında məlumatlar əldə edə bilərik. Ubuntu User-nə baxdıqda qarşısında x hərfini görürük bu user-in şifrəsinin **/etc/shadow** içərisində olduğunu göstərir, 1-ci 1000 rəqəmi User-in İD-ni yəni UİD-ni göstərir, 2-ci 1000 isə Group İD-ni göstərir. Daha sonra **/home/ubuntu** istifadəçinin home qovluğunu göstərir, **/bin/bash** isə istifadəçinin istifadə etdiyi shell-i bildirir.

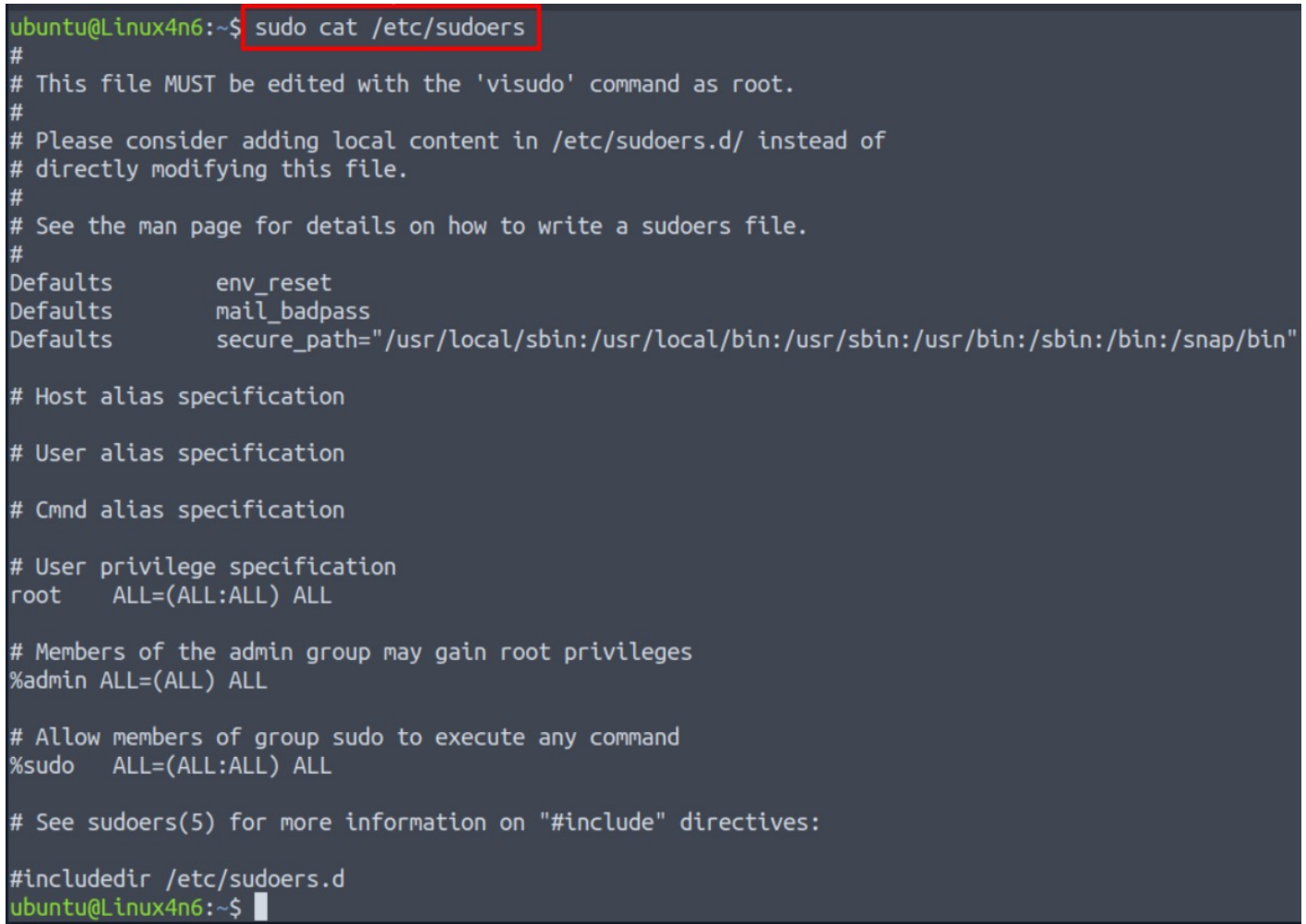




Linux-da Digital Forensics Addımları

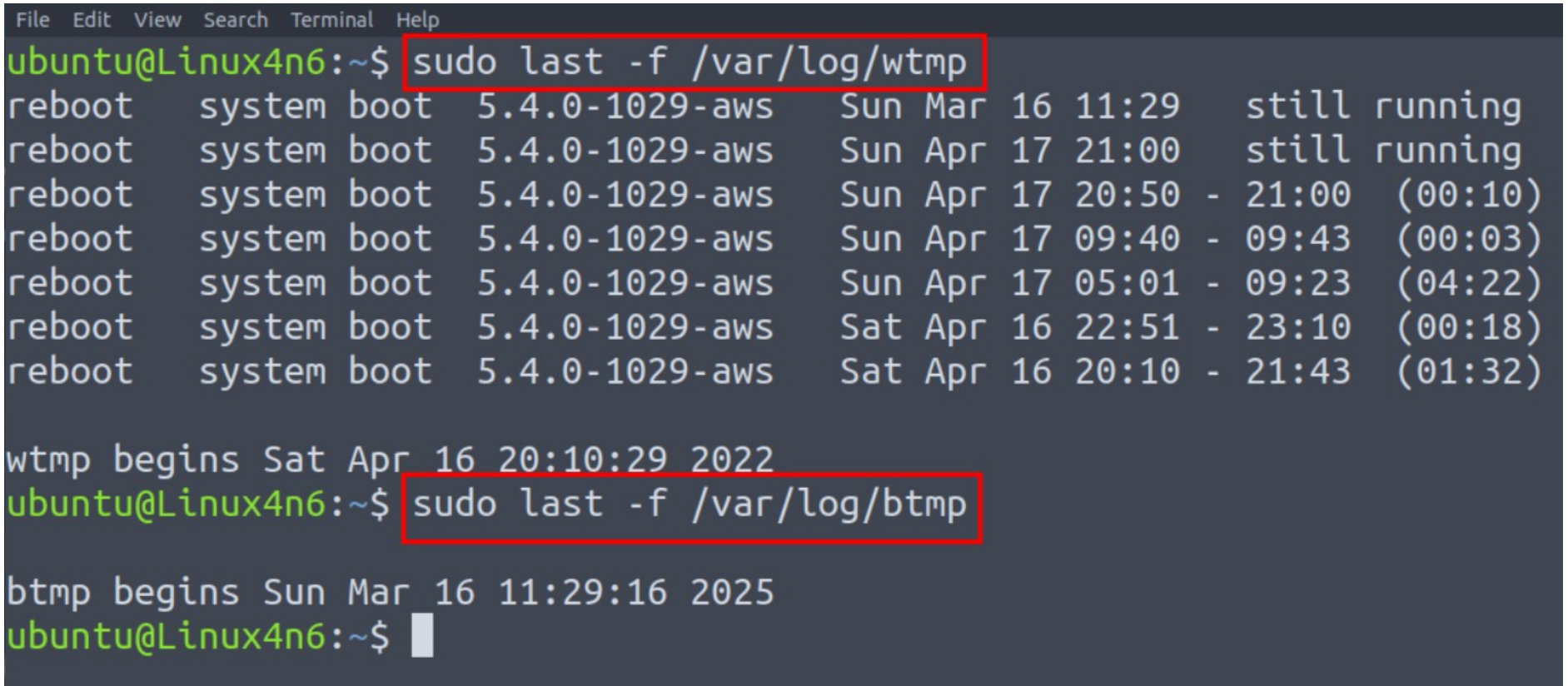
/etc/group faylına baxdığımızda ubuntu User-inin **adm** qrupunda olduğunu görürük. **sudoers** faylına baxdıqda **admin** qrupunda olan user-lərin root yetkisindən istifadə edə bildiklərini görürük, **sudo** qrupunda olan user-lər isə bütün sudo commandlarından istifadə edə bilərlər.





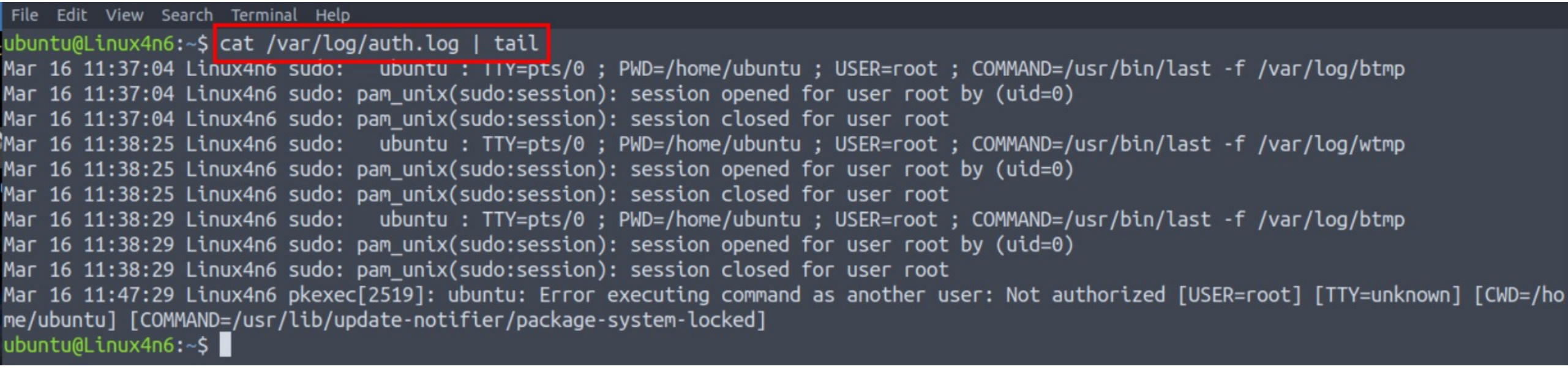
Linux-da Digital Forensics Addımları

/var/log qovluğunda biz **wtmp** və **btmp** daxil olmaqla bütün növ log fayllarını tapa bilərik. **btmp** faylı uğursuz girişlər haqqında məlumatı saxlayır, **wtmp** isə girişlərin tarixi məlumatlarını saxlayır. Bu fayllar **cat**, **less** və ya **vim** commandını istifadə edərək oxuna bilən adi mətn faylları deyil, bu faylları oxumaq üçün **last** commandından istifadə edə bilərik.



Linux-da Digital Forensics Addımları

cat /var/log/auth.log commandı ilə authetication log-larını **auth.log** faylına baxmaqla sistemə istifadəçilərin uğurlu və ya uğursuz giriş cəhdlərini görə bilərik. Burda **Ubuntu** user-inin sudo commandından istifadə etdiyini və buna görə **root** user-inin sessiyasının açılıb daha sonra bağlandığını görürük.



Linux-da Digital Forensics Addımları

“**netstat -tulpan**” commandı ilə şəbəkə bağlantılarını görmək olar. Əgər şəbəkədə bağlantılarında tanınmayan İP ilə bağlantı və bilinməyən bir proqram tərəfindən əlaqə varsa bunlar araşdırılmalıdır Linux Forensics zamanı. Linux hostunun DNS həlli üçün danışdığı DNS serverləri haqqında məlumatlara **resolv.conf** faylında baxa bilərik.

```
ubuntu@Linux4n6:~$ netstat -tulpan
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5901          0.0.0.0:*               LISTEN      928/Xtigervnc
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      1 10.10.126.70:44476      185.125.188.59:443      SYN_SENT    -
tcp        0      1 10.10.126.70:36036      185.125.188.54:443      SYN_SENT    -
tcp        0      0 10.10.126.70:80         10.100.1.179:41354      ESTABLISHED -
tcp        0      1 10.10.126.70:36028      185.125.188.54:443      SYN_SENT    -
tcp        0      1 10.10.126.70:44480      185.125.188.59:443      SYN_SENT    -
tcp        0      0 127.0.0.1:53018         127.0.0.1:5901          ESTABLISHED -
tcp        0      0 10.10.126.70:57906      172.31.65.19:443        ESTABLISHED -
tcp        0      0 127.0.0.1:5901          127.0.0.1:53018         ESTABLISHED 928/Xtigervnc
tcp6       0      0 ::1:5901                :::*                    LISTEN      928/Xtigervnc
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
udp        0      0 0.0.0.0:52447           0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
udp        0      0 10.10.126.70:68         0.0.0.0:*                           -
udp        0      0 0.0.0.0:631             0.0.0.0:*                           -
udp6       0      0 :::5353                 :::*                                -
udp6       0      0 :::40262                :::*                                -
ubuntu@Linux4n6:~$
```

```
ubuntu@Linux4n6:~$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search eu-west-1.compute.internal
ubuntu@Linux4n6:~$ resolvectl status
```

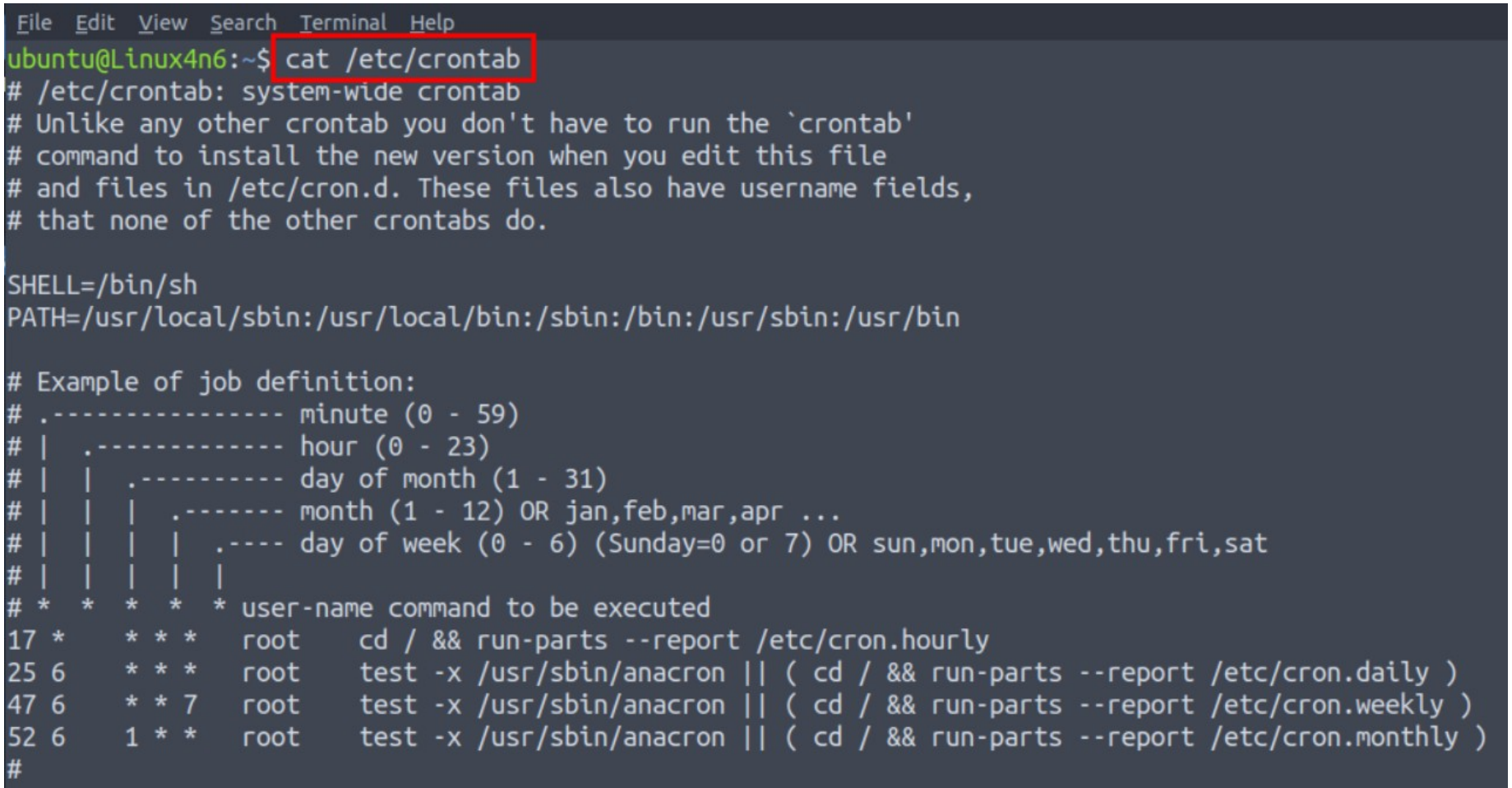
Linux-da Digital Forensics Addımları

Araşdırma zamanı “**ps aux**” commandı ilə sistemdə işləyən prosesləri baxmaq lazımdır. Əgər arxa planda bilinməyən bir proses varsa şübhəli olaraq qeydə alınaraq araştırılmalıdır.

```
ubuntu@Linux4n6:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         638  0.0  0.2 392516 11884 ?        Ssl  15:22   0:00 /usr/lib/udisks2/udisksd
root         639  0.0  0.1  13676  4956 ?        Ss   15:22   0:00 /sbin/wpa_supplicant -u -s -O /run/wpa_supplicant
avahi        648  0.0  0.0   8340   324 ?        S    15:22   0:00 avahi-daemon: chroot helper
root         677  0.0  0.3 178384 12672 ?        Ssl  15:22   0:00 /usr/sbin/cups-browsed
root         683  0.0  0.2 314328 10620 ?        Ssl  15:22   0:00 /usr/sbin/ModemManager
root         689  0.0  0.4 1382776 17460 ?       Ssl  15:22   0:00 /snap/amazon-ssm-agent/5163/amazon-ssm-agent
root         695  0.0  0.5 115708 22892 ?        Ssl  15:22   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-
root         718  0.0  0.1  25332  8032 ?        Ss   15:22   0:00 /usr/sbin/cupsd -l
lp           726  0.0  0.1  15324  6492 ?        S    15:22   0:00 /usr/lib/cups/notifier/dbus dbus://
root         770  0.0  0.0   8536  2952 ?        Ss   15:22   0:00 /usr/sbin/cron -f
whoopsie     777  0.0  0.3 175252 12092 ?        Ssl  15:22   0:00 /usr/bin/whoopsie -f
daemon       778  0.0  0.0   3792  2268 ?        Ss   15:22   0:00 /usr/sbin/atd -f
kernoops     792  0.0  0.0  11252   444 ?        Ss   15:22   0:00 /usr/sbin/kerneloops --test
root         797  0.0  0.0   7352  2400 ttyS0    Ss+  15:22   0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220
root         810  0.0  0.1 305844  7200 ?        Ssl  15:22   0:00 /usr/sbin/lightdm
kernoops     811  0.0  0.0  11252   444 ?        Ss   15:22   0:00 /usr/sbin/kerneloops
root         817  0.0  0.0   5828  1844 tty1     Ss+  15:22   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root         833  0.1  1.5 272132 63784 tty7     Ssl+ 15:22   0:01 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root
ubuntu       875  0.0  0.2  18352  9444 ?        Ss   15:22   0:00 /lib/systemd/systemd --user
ubuntu       884  0.0  0.0 103576  3544 ?        S    15:22   0:00 (sd-pam)
rtkit        915  0.0  0.0 152928  3020 ?        SNsl 15:22   0:00 /usr/libexec/rtkit-daemon
ubuntu       928  0.6  3.6 368616 145568 ?       S    15:22   0:07 /usr/bin/Xtigervnc :1 -desktop Linux4n6:1 (ubuntu) -auth /home/ubun
ubuntu       945  0.0  0.0   7104  4012 ?        Ss   15:22   0:00 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidf
root         964  0.1  0.5 121128 23040 ?        S    15:22   0:02 python3 -m websockify 80 localhost:5901 -D
ubuntu      1009  0.0  1.5 464172 60676 ?        Sl   15:22   0:00 mate-session
ubuntu      1017  0.0  0.0   7160  1972 ?        S    15:22   0:00 dbus-launch --exit-with-session mate-session
ubuntu      1018  0.0  0.0   7936  3872 ?        Ss   15:22   0:00 /usr/bin/dbus-daemon --syslog --fork --print-pid 5 --print-address
```

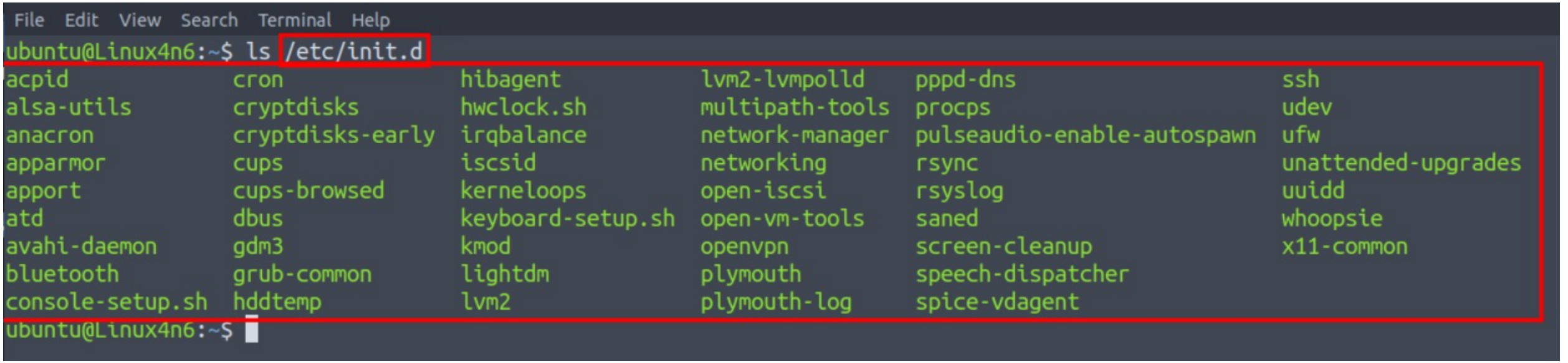
Linux-da Digital Forensics Addımları

Növbəti mərhələdə sistemdə olan Cron-lara (Windows OS-dəki Task Scheduler kimi) “**/etc/crontab**” “**/etc/cron.hourly**” “**/etc/cron.daily**” “**/etc/cron.weekly**” “**/etc/cron.monthly**” fayllarında baxa bilərik. Çünki burda qeyd olunan commandlar sistem başlanğıcında və ya təyin olunmuş bir saatda, gündə işə düşən tapşırıqlardır.



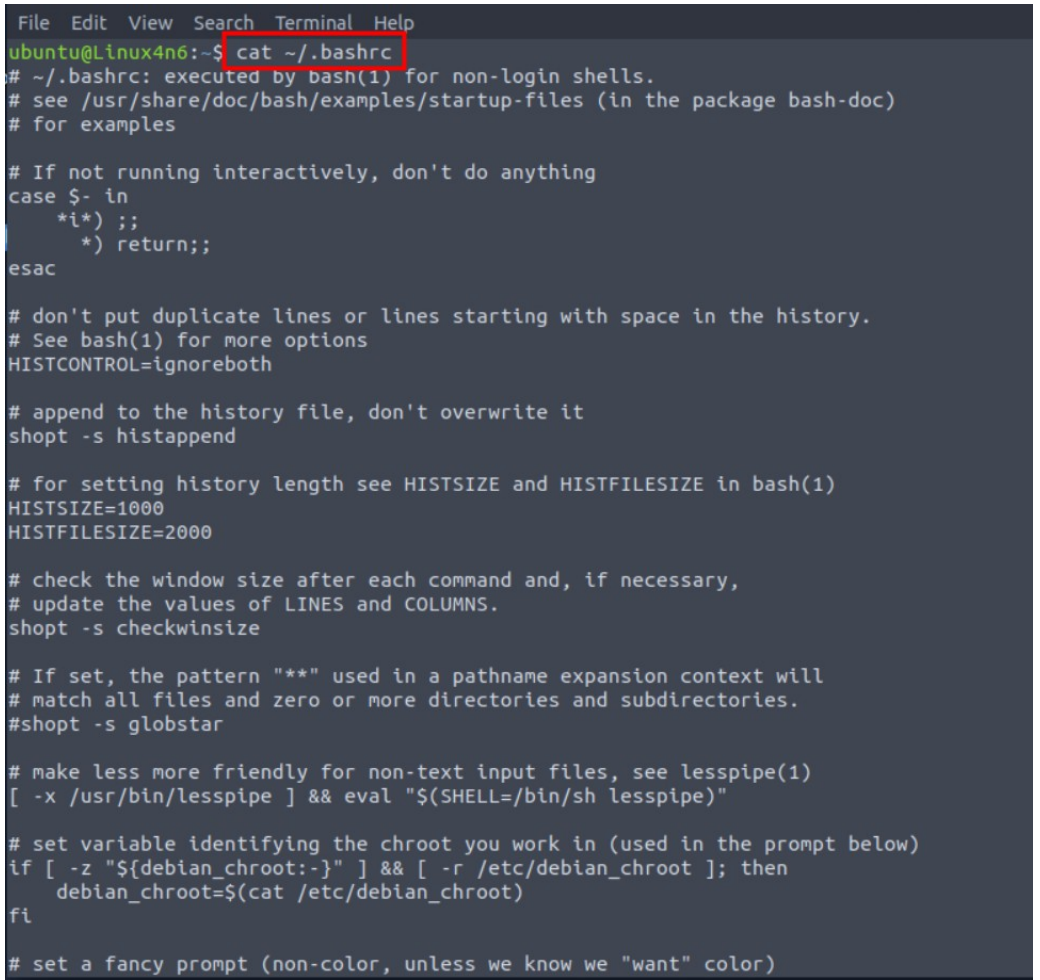
Linux-da Digital Forensics Addımları

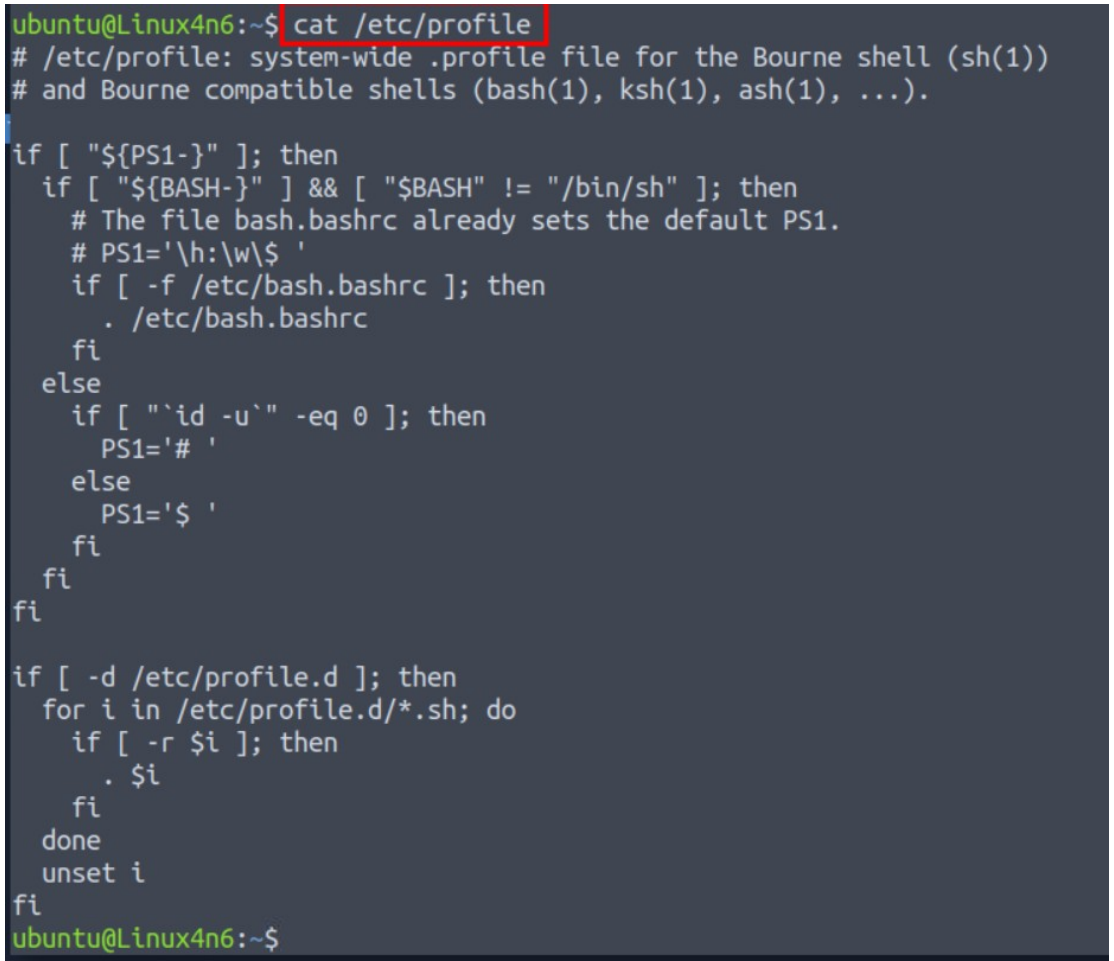
Windows əməliyyat sistemlərində sistem başlanğıcında başlayan servislər eyni qaydada Linux Əməliyyat sistemlərindədə vardır. Bunlara baxmaq üçün “**ls /etc/init.d**” commandını istifadə edirik. Burda **apparmor** (güvənlik servisi), **cron** (zamanlanmış tapşırıqlar), **gdm3** (qrafik interfeys işləməsi üçün servis), **networkmanager** (şəbəkə bağlantılarının idarəsi üçün olan servis), **rsyslog** (user hərəkətlərini loglayır), **ssh** (secure shell protocol), **ufw** (firewall) kimi bir çox servisi görmək mümkündür.



Linux-da Digital Forensics Addımları

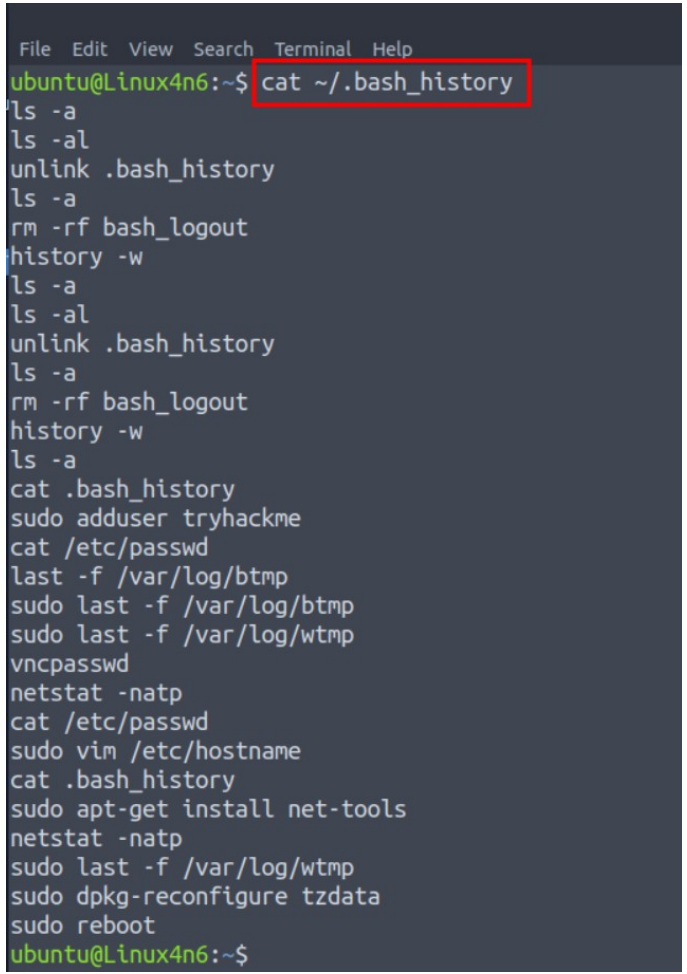
Bash qabığı yarandıqda, o, **~/.bashrc** və ya **~/.zshrc** faylında saxlanılan əmrləri işlədir. Bu fayl yerinə yetiriləcək hərəkətlərin başlanğıc siyahısı kimi qəbul edilə bilər. Buna görə də hücumçuların qalıcılıq əldə etməsi üçün istifadə oluna bilər. Bundan başqa **/etc/profile** faylınada baxa bilərik. Çünki bu cür İstifadəçi fayllarını hücumçular iz buraxmamaq üçün təmizləyə bilərlər.





Linux-da Digital Forensics Addımları

~/.bash_history faylında userin keçmişdə istifadə etdiyi əmrlər saxlanılır, bu fayla baxaraq hücumçunun hansı commandları istifadə etdiyini görə bilərik. Bunu da qeyd edim ki bu fayllar hücumçular tərəfindən manipulyasiya etmək üçün çox istifadə olunan fayllardır. **Vim** mətn redaktoru **Vim**-də açılmış fayllar üçün qeydləri home quvluğunda **.viminfo** adlı faylda saxlayır. Bu faylda açılmış fayllar üçün əmr satırı tarixçəsi, axtarış sətirləri tarixçəsi və s. var.



Linux-da Digital Forensics Addımları

Rsyslog servisi ilə loglanan qeydləri **/var/log/syslog** fayllarında görə bilərik. Burda isə əməliyyatın baş verdiyi tarix, saat, hansı servis tərəfindən və nəyin icra olunduğu haqında məlumatlar əldə edə bilərik. Misal olaraq crontab ilə 16 Mart saat 15:22-də "**python3 -m websockify 80 localhost:5901 -D**" istifadə olunduğunu görürük.

```
File  Edit  View  Search  Terminal  Help
ubuntu@Linux4n6:~$ sudo cat /var/log/syslog* | more
Mar 16 15:22:51 Linux4n6 rsyslogd: [origin software="rsyslogd" swVersion="8.2001.0" x-pid="627" x-info="https://www.rsyslog.com"] rsyslogd w
as HUPed
Mar 16 15:22:51 Linux4n6 CRON[834]: (ubuntu) CMD (sudo runuser -l ubuntu -c 'vncserver :1 -depth 24 -geometry 1900x1200')
Mar 16 15:22:51 Linux4n6 systemd[1]: Started Light Display Manager.
Mar 16 15:22:51 Linux4n6 dbus-daemon[599]: [system] Activating via systemd: service name='org.freedesktop.Accounts' unit='accounts-daemon.se
rvice' reque
sted by ':1.27' (uid=0 pid=810 comm="/usr/sbin/lightdm " label="unconfined")
Mar 16 15:22:51 Linux4n6 CRON[832]: (ubuntu) CMD (sudo python3 -m websockify 80 localhost:5901 -D)
Mar 16 15:22:51 Linux4n6 systemd-hostnamed[669]: Changed host name to 'ip-10-10-126-70'
Mar 16 15:22:51 Linux4n6 accounts-daemon[591]: started daemon version 0.6.55
Mar 16 15:22:51 Linux4n6 dbus-daemon[599]: [system] Successfully activated service 'org.freedesktop.Accounts'
Mar 16 15:22:51 Linux4n6 systemd[1]: Started Accounts Service.
Mar 16 15:22:51 Linux4n6 blueman-mechani[597]: gtk_icon_theme_get_for_screen: assertion 'GDK_IS_SCREEN (screen)' failed
Mar 16 15:22:51 Linux4n6 systemd[1]: Created slice User Slice of UID 1000.
Mar 16 15:22:51 Linux4n6 systemd[1]: Starting User Runtime Directory /run/user/1000...
Mar 16 15:22:51 Linux4n6 systemd[1]: Started Bluetooth management mechanism.
Mar 16 15:22:51 Linux4n6 systemd[1]: Finished User Runtime Directory /run/user/1000.
Mar 16 15:22:51 Linux4n6 avahi-daemon[596]: Server startup complete. Host name is Linux4n6.local. Local service cookie is 485281612.
Mar 16 15:22:51 Linux4n6 systemd[1]: Starting User Manager for UID 1000...
Mar 16 15:22:51 Linux4n6 systemd[1]: logrotate.service: Succeeded.
Mar 16 15:22:51 Linux4n6 systemd[1]: Finished Rotate log files.
Mar 16 15:22:51 Linux4n6 systemd[875]: Started Pending report trigger for Ubuntu Report.
Mar 16 15:22:51 Linux4n6 systemd[875]: Reached target Paths.
Mar 16 15:22:51 Linux4n6 systemd[875]: Reached target Timers.
Mar 16 15:22:51 Linux4n6 systemd[875]: Starting D-Bus User Message Bus Socket.
Mar 16 15:22:51 Linux4n6 systemd[875]: Listening on GnuPG network certificate management daemon.
Mar 16 15:22:51 Linux4n6 systemd[875]: Listening on GnuPG cryptographic agent and passphrase cache (access for web browsers).
Mar 16 15:22:51 Linux4n6 systemd[875]: Listening on GnuPG cryptographic agent and passphrase cache (restricted).
Mar 16 15:22:51 Linux4n6 systemd[875]: Listening on GnuPG cryptographic agent (ssh-agent emulation).
Mar 16 15:22:51 Linux4n6 systemd[875]: Listening on GnuPG cryptographic agent and passphrase cache.
Mar 16 15:22:51 Linux4n6 systemd[875]: Listening on debconf communication socket.
```

Linux Forensics Cheatsheet

System and OS information

OS release information:
Location: /etc/os-release
Can be read using cat, vim or any text editor or viewer

User accounts information:
Location: /etc/passwd
Can be read using cat, vim or any text editor or viewer

User group information:
Location: /etc/group
Can be read using cat, vim or any text editor or viewer

Sudoers list:
Location: /etc/sudoers
Can be read using cat, vim or any text editor or viewer.
Needs sudo or root permissions to access

Login information:
Location: /var/log/wtmp
Can be read using last utility

Authentication logs:
Location: /var/log/auth.log
Can be read using cat, vim or any text editor or viewer.
Use grep for better filtering.
Might also have auth.log1, auth.log2 etc as log files that have been rotated.

System configuration

Hostname:
Location: /etc/hostname
Can be read using cat, vim or any text editor or viewer

Timezone information:
Location: /etc/timezone
Can be read using cat, vim or any text editor or viewer

Network Interfaces:
Location: /etc/network/interfaces
Can be read using cat, vim or any text editor or viewer

Command: ip address show
The above command is suitable only for live analysis

Open network connections:
Command: netstat –natp
The above command is suitable only for live analysis

Running processes:
Command: ps aux
The above command is suitable only for live analysis

DNS information:
Location: /etc/hosts for hostname resolutions
Can be read using cat, vim or any text editor or viewer

Location: /etc/resolv.conf for information about DNS servers
Can be read using cat, vim or any text editor or viewer

Persistence mechanism

Cron jobs:
Location: /etc/crontab
Can be read using cat, vim or any text editor or viewer

Services:
Location: /etc/init.d/
Registered services are present in this directory

Bash shell startup:
Location: /home/<user>/.bashrc for each user

Locations: /etc/bash.bashrc and /etc/profile for system wide settings. Can be read using cat, vim or any text editor or viewer

Evidence of execution

Authentication logs:
Location: /var/log/auth.log* |grep -i COMMAND;
the grep can be used to filter the results. Can be read using cat, vim or any text editor or viewer

Bash history:
Location: /home/<user>/.bash_history
Can be read using cat, vim or any text editor or viewer

Vim history:
Location: /home/<user>/.viminfo
Can be read using cat, vim or any text editor or viewer

Log files

Syslogs:
Location: /var/log/syslog
Can be read using cat, vim or any text editor or viewer.
Use grep or similar utility to filter results as per requirement

Authentication logs:
Location: /var/log/auth.log
Can be read using cat, vim or any text editor or viewer.
Use grep or similar utility to filter results as per requirement

Third-party logs:
Location: /var/log
Logs for each third-party application can be found in their specific directories in this location

Digital Forensics İn Linux

Task üzrə hazırlanmış Report-un bu hissəsindən sonra, İnternetdə araşdırıb tapdığım bir məqaləni burda göstərmiş olacağam.

Linux sistemləri cinayət hadisələrində getdikcə daha çox rast gəlinəcək, xüsusilə də serverlər üçün üstünlük verilən əməliyyat sistemi kimi populyarlığı artdıqca. Bununla belə, Microsoft Windows əsaslı sistemlərdən sübutların bərpası üçün tələb olunan bacarıq və biliklər Linux sistemləri üçün eyni şəkildə tətbiq edilə bilməz. Məsələn, Microsoft NTFS, FAT və Linux EXT2/3 fayl sistemləri kifayət qədər fərqli işlədiyindən, birini anlamaq digəri haqqında çox az məlumat verir.

Bu məqalədə Linux əmrlər sətri utilitlərindən istifadə edərək Linux sistemlərində digital forensics prosedurlarını nümayiş etdiririk. İşləyən bir sistemdən sübutların toplanması xüsusilə vacibdir, çünki birinci dərəcəli kriminalistika mütəxəssisi canlı sübutları toplamağa üstünlük verməsə, RAM-dakı məlumatlar itə bilər.

Burada müzakirə olunan kriminalistika prosedurlarına RAM və maqnit mühitindən silinmiş faylların aşkar edilməsi və bərpası, mühüm faylların və Troyan proqramlarının müəyyən edilməsi, həmçinin gizli və adları dəyişdirilmiş faylların tapılması daxildir.

İlk olaraq, işləyən (aktiv) bir Linux sistemində RAM-dan silinmiş faylların bərpası prosesini təsvir etməklə başlayırıq. Linux sistemləri tez-tez serverlər kimi istifadə edildiyi üçün, nümunələrin əksəriyyəti sistemə müdaxilə etmiş şəxslərin tətbiq etdiyi fəaliyyətlərə və texnikalara yönəldilmişdir.

Digital Forensics İn Linux

RAM-dan Faylların Bərpası

Diskin üzərinə yazılmış olan silinmiş bir fayl hələ də bərpa edilə bilər. Rəqəmsal kriminalistika texnikasını izah etmək üçün belə bir ssenarini nəzərdən keçirək: bir hücumçu bir proqramı işə salır və sonra onun mövcudluğunu gizlətmək üçün onu diskdən silir.

Bu, məsələn, bir hücumçunun netcat utilitini işə salaraq qurbanın sisteminə arxa qapı (backdoor) qurduqda və sonra həmin utiliti diskdən sildikdə baş verə bilər. Proqram hələ də yaddaşda işləyən bir proses olaraq qaldığı müddətcə orijinal icra edilə bilən fayl bərpa oluna bilər.

Bu fayl bərpa edilə bilər, çünki Linux nüvəsi, işləyən proseslər, qoşulmuş fayl sistemləri, nüvə məlumatları və digər bir neçə yüz kritik sistem məlumatları da daxil olmaqla sistemin ümumi vəziyyətini izləmək üçün saxta bir fayl sistemindən istifadə edir. Bu məlumatlar virtual yaddaşda saxlanılır və **/proc** qovluğu vasitəsilə əldə edilə bilər.

Aşağıdakı (qismən) siyahı, işləyən bir Linux sistemində **/proc** qovluğunun məzmununu göstərir. Burada verilmiş nömrələrin hər biri bir prosessor ID-sinə uyğundur və həmin proses haqqında məlumatları saxlayan bir qovluqdur.

```
# Is /proc
1 4 4513 4703 4777 execdomains mdstat swaps
1693 40 4592 4705 acpi fb meminfo sys
2 4045 4593 4706 asound filesystems misc
2375 41 4594 4708 buddyinfo fs mm sysvipc
2429 4163 4595 4709 bus ide modules tty
2497 4166 4596 4712 cmdline interrupts mounts uptime
2764 4186 4597 4713 config.gz iomem mtrr version
29 42 4620 4715 cpufreq ioports net vmstat
```

Digital Forensics İn Linux

Silinmiş bir faylın bərpasını izah etmək üçün, bir hücumçunun parol qırıcı proqramı endirdiyini və sistem parollarını qırmağa çalışdığını fərz edək – bu, hücumçular üçün çox yayılmış bir hədəfdir. Hücumçu, john (www.openwall.com) parol qırıcı proqramını pass adlı bir faylda olan parolların siyahısı ilə işə salır. Dəvətsiz qonaq daha sonra həm işlədilə bilən faylı, həm də parolları ehtiva edən mətn faylını silir. İcra edilə bilən fayl, proses öldürülənə qədər yaddaşda işləməyə davam edir. ps komandası işləyən prosesləri göstərir. Aşağıdakı siyahı John işlədilə bilən faylının pass faylı ilə saat 10:10-da çağırıldığını, 22 saniyə ərzində işlədiyini və root istifadəçisi tərəfindən sahiblik edildiyini göstərir.

```
# ps aux I grep John
root 5288 97.9 0.0 1716 616 pts/2 R+ 10:10 0:22 ./John pass
```

Yuxarıdakı siyahıya əsasən, işlədilə bilən proses kimliyi (**PID**) **5288**-dir. Aşağıdakı (qismən) siyahıda göstərildiyi kimi, **/proc/5288** qovluğu işləyən proseslə əlaqəli məlumatları saxlayacaqdır.

```
# ls -al /proc/5288
total 0
dr-xr-xr-x 3 root root 0 Jan 17 10:11 .
dr-xr-xr-x 108 root root 0 Jan 17 04:00 ..
-r--r--r-- 1 root root 0 Jan 17 10:11 cmdline
lrwxrwxrwx 1 root root 0 Jan 17 10:12 cwd -> /j
-r-------- 1 root root 0 Jan 17 10:12 environ
lrwxrwxrwx 1 root root 0 Jan 17 10:12 exe -> /j/john (deleted)
lrwxrwxrwx 1 root root 0 Jan 17 10:12 root -> /
-r--r--r-- 1 root root 0 Jan 17 10:11 stat
-r--r--r-- 1 root root 0 Jan 17 10:12 statm
dr-xr-xr-x 3 root root 0 Jan 17 10:12 task
```

Digital Forensics İn Linux

/proc/5288 qovluğu bir neçə fayl və qovluq ehtiva edir, bunlardan ən önəmlisi işləyən parol qırıcıya sembolik bir bağlantı olan **exe** faylıdır (izinlərin ilk sütunundakı 1-ə diqqət edin). Əməliyyat sistemi (yardımçı olaraq) faylın diskdən silindiyini göstərən bir qeyd göstərir. Bununla belə, **exe** faylını qovluqdan ayrı bir qovluğa kopyalayaraq faylı bərpa edə bilərik.

```
# cp /proc/5288/exe ./john.recovered
# md5sum ./john.recovered ./john.original
83219704ded6cd9a534baf7320aebb7b ./john.recovered
83219704ded6cd9a534baf7320aebb7b ./john.original
```

Yuxarıdakı nümunədə **exe** faylını **/proc/5288** qovluğundan başqa bir qovluğa kopyaladıq və sonra işlədilə bilən faylın **MD5 hash** dəyərini **John**-un məlum bir nüsxəsinin **hash** dəyəri ilə müqayisə etdik. **Hash** dəyərlərinin eyni olduğunu görürük, bu da faylı uğurla bərpa etdiyimizi göstərir. Bu fayl bərpa etmə üsulu, proses yaddaşda qaldığı müddətcə hər cür fayl üçün işləyir.

Digital Forensics İn Linux

Faylların növünə görə fayl bərbası

Bir faylın başında olan fayl başlığı üçün ayrılmamış sahəni axtararaq faylı manual (əl ilə) bərpa edə bilərik. Məsələn, bir hücumçunun bir neçə yüz bitmap qrafikasını ehtiva edən bir qovluğu sildiyini bildiyimizi fərz edək. Bir bit səviyyə qrafikinin imzası olan **BM** ilə başlayan bir sektor üçün ayrılmamış sahədə axtarış edə bilərik. Tapdıqda, **Linux dd** komandasını istifadə edərək faylı əl ilə bərpa edə bilərik. Bu prosedurun uğurlu olduğu fərz edilir: (i) başlıq məlumatlarını müəyyən edə bildiyimiz, (ii) faylın üzərinə yazılmadığı və (iii) faylın parçalanmadığı halda. Fayl parçalanmışsa, əvvəlcə faylı təşkil edən blokları tanıya bilmədiyimiz üçün faylın yalnız bir hissəsini bərpa edə bilərik. Bu nümunədə, silinmiş bir neçə **.jpg** faylı ehtiva edən bir görüntümüz (və ya əlaqəsiz bölməmiz) var. İlk növbədə, hər bir **JPG** faylının ilk sektorunu müəyyən etməliyik, bunu da **JPG** faylında tez-tez rast gəlinən **JFIF** mətnini axtararaq edirik. Aşağıdakı siyahı, silinmiş bir **JPG** faylı üçün bir başlanğıc sektorunu göstərir. (**Qeyd**: Bir .jpg faylı üçün lazım olan başlığın tək hissəsi ilk üç baytdır: **ff d8 ff e0**. Təcrübələrdə, başlıqdakı **JFIF**-in silinməsinin tətbiqlərin fayl növünü düzgün şəkildə müəyyən etməsinə mane olmadığını, lakin ilk üç baytdan hər hansı birinin silinməsinin bunu təsir etdiyini gördük).

0004200: ffd8 ffeO 0010 4a46 4946 0001 0200 0064

Yuxarıdakı siyahı faylın **0x4200** (**hex**) ilə başladığını göstərir. Bunu ondalık ədədi olan **16,896**-ya çeviririk və **512**-yə (sektordakı bayt sayı) bölərək **33** nəticəsini alırıq, bu da faylın başlanğıc sektor nömrəsidir, yəni görüntünün başlanğıcından etibarən. Bir çox hallarda, silinmiş bir faylın tam ölçüsünü bilmirik, bu da ölçü haqqında təxmini bir qiymətləndirmə etməyimizi tələb edir. Çox aşağı bir qiymət təxmin edərək faylı az bərpa etsək, faylı öz tətbiqində göstərmək çox az sektorun bərpa edildiyini göstərəcək və fayl tam görünməyəcəkdir. Əgər çox sayda sektor bərpa etsək, həddindən artıq bərpa etmiş oluruq. Təcrübələrimizə görə çox sayda sektorun bərpa edilməsi fayla zərər vermir. Faylı bərpa etdikdən sonra, təxminimizin doğruluğunu müəyyənləşdirmək üçün uyğun tətbiqdə görüntüləyə bilərik.

Digital Forensics İn Linux

Giriş faylını görüntümüz olaraq təyin edirik (**if=image.dd**) və bərpa olunacaq fayl üçün bir ad seçirik (**of=recoveredl.jpg**). Oymaya başlamaq üçün başlanğıc sektoru göstərməliyik. Əvvəlki hesablamamıza əsasən, görüntü fiziki **sektor 33**-də başlayır. dd-də varsayılan blok ölçüsü **512 baytdır** və bunu olduğu kimi saxlayacağıq. Son olaraq, bərpa olunacaq ardıcıl blok sayını göstərməliyik. Bu nümunədə hər biri **512 bayt** ölçüsündə **30 blok** təxmin edəcəyik, beləliklə **15K** ölçüsündə faylları bərpa edəcəyik.

```
# dd if=image.dd of=recovered1.jpg skip=33 count=30
30+0 records in
30+0 records out
# file recovered1.jpg
recovered1.jpg:  JPEG image data, JFIF standard 1.01
```

JPG faylının **30** ardıcıl sektorunu uğurla bərpa etdik. Dosya əmri başlığın uğurla bərpa edildiyini göstərir. Bu bərpa üsulu, fayl başlığı məlumatları pozulmadığı müddətcə hər cür faylda istifadə oluna bilər. Bu üsulun müvəffəqiyyəti, yenə də faylda parçalanma olmamasına və faylın hər hansı bir blokunun yenidən istifadə edilib-edilməməsinə bağlıdır.

Digital Forensics İn Linux

Ümumi Fayl Bərpa Proseduru

Bir fayl başlığı üzərinə yazılıbsa və fayl əsasən mətnindən ibarətdirsə, yalnızca axtarılacaq bəzi açar sözləri bilməyimizi və əlbəttə faylın tamamilə üzərinə yazılmamış olmasını tələb edən daha ümumi bir bərpa proseduru istifadə edə bilərik. Bu göstərim üçün Linux-un ümumi günlük faylı **/var/log/messages** faylını bərpa edəcəyik. Bu fayl adətən bir hücumçu tərəfindən hədəf alınır, çünki hücumçunun izlərinə dair sübutlar ehtiva edir. Təcrübəsiz hücumçular faylın tamamını siləcəklər, bu da bir sistem idarəçisi üçün açıq şəkildə hücumun baş verdiyinə dair sübutdur. Əksinə, bacarıqlı dəvətsiz qonaqlar, icazəsiz girişi göstərən sətirləri cərrah kimi siləcək və qalan məzmunu saxlayacaqlar.

Günlük faylını bərpa etmək üçün faylda olan açar sözləri müəyyən etməliyik. İdeal olaraq, fayla xas olan açar sözləri müəyyən edirik, beləliklə yanlış müsbət nəticələrin sayını azaldırıq. Bu nümunədə, günlük faylları tez-tez dövr etdiyindən bəzi yanlış müsbət nəticələrlə qarşılaşmamız ehtimalı yüksəkdir, buna görə də axtarışımızın günlük faylının əvvəlki versiyalarından açar sözlər alması ehtimalı var. Mesajların yerləşdiyi **/var** qovluğunu ehtiva edən bölmənin bağlantısını kəsirik. Əgər **/var** öz bölməsindədirsə, bu əməliyyat olduqca sadədir:

 # umount /dev/hda3

Digital Forensics İn Linux

Əgər **/var** kök dizini ilə eyni bölmədədirsə, **Linux** önyüklənə bilən **CD** istifadə edərək sistemi yenidən başlatmalı və prosedurları önyükləmə diskindən həyata keçirməliyik. Növbəti istifadədə fiziki cihazdakı (bağlanmamış bölmə) açar sözləri axtarmaq üçün **grep** commandından istifadə edəcəyik. Fiziki cihazı istifadə edirik, çünki ayrılmamış sahəyə fiziki cihaz vasitəsilə giriş etməliyik.

 # grep -ia -f keywords -C 2 /dev/hda3

"**i**" bayrağı böyük/kiçik hərfə həssas olmayan (**case sensitive**) bir axtarışı göstərir. "**a**" bayrağı girişin (**/dev/hda3** fiziki cihazının məzmunu) **ASCII** mətni kimi qəbul ediləcəyini göstərir; bunu etməsək **grep** yalnızca faylın açar sözləri ehtiva edib-etmədiyini göstərəcək. "**f**" bayrağı, izləyənin axtarılacaq açar sözlərin siyahısını ehtiva edən bir mətn faylı olduğunu göstərir. Əsasən bir neçə açar söz üçün eyni anda axtarış aparırıq, məsələn, silinmiş faylımızın tam olaraq hansı açar sözləri ehtiva etdiyini bilməsək bunu istifadə edə bilərik. "**-C 2**" bayrağı, bir açar söz tapıldığında iki sətir əvvəl və iki sətir sonrakı konteksti istədiyimizi göstərir.

Digital Forensics İn Linux

Nəhayət, **/var** qovluğunu ehtiva edən axtarılacaq fiziki cihazı müəyyən edirik. Bu nümayiş üçün, hücumçunun root hesabına daxil olmaq üçün bir neçə uğursuz giriş cəhdi etdiyini fərz edirik - bir icazəsiz girişdə geniş yayılmış bir haldır. Bu uğursuz sessiya açma cəhdləri mesajlar günlük faylında qeydə alınacaq.

Axtarışımızın nəticələri aşağıda göstərilir:

```
Dec 18 19:13:09 gheera gdm(pam_unix)[2727]:  authentication failure;
logname= uid=0 euid=0 tty=:0 ruser= rhost= user=schmoopie
Dec 18 19:13:13 gheera gdm-binary[2727]:  Couldnt authenticate user
Dec 18 19:13:16 gheera gdm(pam_unix)[2727]:  session opened for user
schmoopie by (uid=0)
Dec 20 18:33:29 gheera gdm(pam_unix)[2752]:  authentication failure;
logname= uid=0 euid=0 tty=:0 ruser= rhost= user=schmoopie
Dec 21 18:16:55 gheera gdm(pam_unix)[2750]:  authentication failure;
logname= uid=0 euid=0 tty=:0 ruser= rhost= user=schmoopie
Dec 22 17:49:33 gheera gdm(pam_unix)[2756]:  authentication failure;
logname= uid=0 euid=0 tty=:0 ruser= rhost= user=schmoopie
Dec 22 17:49:36 gheera gdm-binary[2756]:  Couldnt authenticate user
Dec 22 17:49:48 gheera gdm(pam_unix)[2756]:  session opened for user
schmoopie by (uid=0)
```

Açar sözlər qalın yazılmışdır. Görünür ki, **schmoopie** adlı istifadəçi **18**, **21** və **22 dekabr** tarixlərində **root** kimi daxil olmağa uğursuz cəhd edib. Hər axtarış nəticəsindən əvvəl və sonra iki sətirlik məzmuna diqqət yetirin. Praktikada bu qədər məhdud bir nəticə istəməzdik: jurnalda olan bütün məlumatları bərpa etməyi üstün tutarıq. Bunu isə daha böyük bir kontekst dəyəri tələb etməklə edə bilərik, məsələn, "**-C 100**". Faylın nə qədər böyük olduğunu əvvəlcədən bilmədiyimiz üçün bu, sınaq və səhv metodu ilə aparılan bir cəhd olacaqdır.

Digital Forensics İn Linux

EXT2 disklərindən faylların bərpa edilməsi

Son bərpaetmə metodu, fayl sisteminin geniş yayılmış bir **Linux** fayl sistemi olan **EXT2** olduğunu fərz edək (baxmayaraq ki, o, daha səmərəli jurnallaşdırma fayl sistemlərinə yerini verməkdədir). Bu üsulda faylı tapmaq və bərpa etmək üçün sistemin xəta sazlayıcısından istifadə edə bilərik. Bu nümunə üçün, yaxın zamanda işdən çıxarılmış bir işçinin /home altındakı qovluğundan vacib bir faylı sildiyini fərz edək. (Bu, işdən çıxarılan işçilər üçün nadir rastlanan bir hal deyil.) Tutaq ki, faylın **ZIP** arxivi olduğu barədə məlumat əldə etmişik. Fayl bərpa prosesinə başlamazdan əvvəl, bölmələrin sayı və onların necə formatlandığı da daxil olmaqla, sərt disklərin quruluşunu müəyyən etməliyik. **Linux**-un "**fdisk -l**" əmri bu məlumatı təmin edir:

```
# fdisk -l
Disk /dev/hda:  30.0 GB, 30005821440 bytes
16 heads, 63 sectors/track, 58140 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Device Boot Start End Blocks Id System
/dev/hda1 1 41613 20972826 83 Linux
/dev/hda2 57147 58140 500976 f W95 Extd (LBA)
/dev/hda3 41613 52020 5245222+ 83 Linux
/dev/hda5 57147 58140 500944+ 82 Linux swap
```

Digital Forensics İn Linux

Tək bir **30 GB**-lıq **IDE** sərt diskimizin dörd bölmədən ibarət olduğunu görürük. İlk bölmə (**/dev/hda1**) birincil bölmə olaraq **EXT2** fayl sistemi ilə formatlanmışdır. İkinci bölmə (**/dev/hda2**) genişləndirilmiş bölmədir və iki məntiqi bölmə ehtiva edir: biri **EXT2** fayl sistemi ilə formatlanmışdır (**/dev/hda3**), digəri isə **Linux**-un dəyişdirmə (**swap**) fayl sistemidir (**/dev/hda5**). Daha sonra hansı qovluqların hansı bölmələrə qoşulduğunu bilməliyik. Bu məlumatı göstərmək üçün "**mount**" əmrini işlədirik.

```
# mount | column -t
/dev/hda1 on / type ext2 (rw,acl,user_xattr)
proc on /proc type proc (rw)
tmpfs on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda3 on /home type ext2 (rw,acl,user_xattr)
/dev/hdc on /media/cdrom type subfs ...
```

mount əmri bizə göstərir ki, **/home** qovluğu **/dev/hda3** cihazına qoşulmuşdur. Silinmiş faylın üzərinə yazılmasının qarşısını almaq üçün ya **/home** qovluğunu ayırırıq, ya da onu yalnız oxuma rejimində yenidən qoşuruq. Bu prosesi nə qədər tez həyata keçirsək, faylın üzərinə yazılma ehtimalı bir o qədər az olar. Çünki bölmə qoşulu qaldıqca, faylın üzərinə yazılma şansı artır. Qovluğu ayırmaq üçün aşağıdakı əmri yerinə yetiririk:

 # umount /home

Silinmiş faylı bərpa etmək üçün **debugfs** hata ayıklayıcıdan istifadə edirik və bölməni açırıq. Hata ayıklayıcıda "**lsdel**" əmrini icra edirik ki, bu da bölmədə silinmiş bütün faylların **inode** məlumatlarını göstərir. (**İnode**, faylın metadatasını saxlayan bir verilənlər strukturdur.

```
# debugfs /dev/hda3
debugfs 1.35 (28-Dec-2004)
debugfs:  lsdel
Inode Owner Mode Size Blocks Time deleted 272319 0 100755 3383328
828/ 828 Thu Dec 23 23:45:22 2004 1 deleted inodes found.  lines 1-3/3
(END)
```

Digital Forensics İn Linux

“**lsdel**” əmri, 272319 **inode** nömrəsi ilə təmsil olunan bir faylın **23 dekabr** tarixində silindiyini və **3 MB** ölçüsündə olduğunu (828 blokdan ibarət) göstərir. **Inode** nömrəsini əldə etdikdən sonra, daha ətraflı məlumat almaq üçün "**stat**" əmrindən istifadə edə bilərik:

```
debugfs:  stat <272319>
Inode:  272319 Type:  regular Mode:  0755 Flags:  0x0 Generation:
92194859
User:  0 Group:  0 Size:  3383328
File ACL: 0 Directory ACL: 0
Links:  0 Blockcount:  6624
Fragment:  Address:  0 Number:  0 Size:  0
ctime:  0x41cb9ee2 -- Thu Dec 23 23:45:22 2004
atime:  0x41cb9d68 -- Thu Dec 23 23:39:04 2004
mtime:  0x41cb9d68 -- Thu Dec 23 23:39:04 2004
dtime:  0x41cb9ee2 -- Thu Dec 23 23:45:22 2004
BLOCKS:
(0-11):582122-582133, (IND):582134, (12-826):582135-582949
TOTAL: 828
```

"**Stat**" commandı bizə bir sıra məlumatlar təqdim edir, o cümlədən silinmiş faylın dəyişdirilmə, əldə edilmə və silinmə tarixləri və saatları. (**NTFS** və **FAT** fayl sistemlərindən fərqli olaraq, **Linux EXT2** fayl sistemi faylın silinmə tarixini və saatını izləyir.) "**Stat**" commandı həmçinin **BLOCKS** bölməsində birbaşa, dolayı və ikili dolayı blokların sayını göstərir. Görünür ki, bütün bloklar tamamilə saxlanılıb, yəni heç bir blok üst-üstə yazılmayıb və bu da o deməkdir ki, biz bütün faylı bərpa edə bilərik. "**Dump**" commandı arqument olaraq bir **inode** nömrəsi və bərpa edilən fayl üçün bir ad alır:

debugfs: dump <272319> hda3.recovered

Debugger-dən çıxdıqdan sonra, bərpa edilmiş faylın növünü "**file**" commandı ilə müəyyən edirik. "**file**" commandı başlıq məlumatlarından istifadə edərək faylın növünü müəyyən edir.

```
# file hda3.recovered
hda3.recovered:  Zip archive data, at least v1.0 to extract
```

Digital Forensics In Linux

Bərpa etdiyimiz fayl gözlənildiyi kimi **ZIP** arxividir. Prosedurumuzun uğurlu olub-olmadığını, bərpa edilmiş faylın hash dəyərini orijinal faylın hash dəyəri ilə müqayisə edərək müəyyən edirik (bu nümunə üçün bizdə orijinal faylın hash dəyəri mövcuddur). Hash-ların uyğun gəlməsi faylı uğurla bərpa etdiyimizi göstərir. (Əgər orijinal faylın **MD5** hash-i yoxdursa, sadəcə **ZIP** arxivini açaraq da yoxlama aparmaq olar.)

 # md5sum original.file.zip hda3.recovered
ed9a6bb2353ca7126c3658cb976a2dad original.file.zip
ed9a6bb2353ca7126c3658cb976a2dad hda3.recovered

Bu prosedurun uğuru bir neçə vacib amildən asılıdır. Birinci amil, faylın silinməsi ilə bərpa edilməsinə cəhd göstərilən zaman intervalıdır. Silinmə ilə bərpa arasındakı müddət nə qədər uzun olarsa, faylın bir hissəsinin və ya bütövlükdə faylın üzərinə yazılma ehtimalı bir o qədər yüksəkdir. İkinci amil isə faylın ölçüsüdür. Daha kiçik fayllar (birbaşa bloklara sığanlar) dolayı və ikiqat dolayı blokların istifadəsini tələb edən daha böyük fayllara nisbətən daha yüksək bərpa ehtimalına malikdir.

Digital Forensics In Linux

Əhəmiyyətli Faylların və Troyanların Müəyyən Edilməsi

İzləyicilərin əsas iki məqsədi sistemə sızmaq və mümkün qədər uzun müddət qurban sistemində qalmaqdır. Sistemdə gizli qalmaq adətən **rootkit** quraşdırmaqla həyata keçirilir. **Rootkit**, bir neçə vacib sistem faylını "**Trojan**" versiyalarla əvəz edir. Bu **Trojan** versiyalar orijinal sistem faylları kimi işləyir, lakin izləyicinin izlərini - məsələn, işləyən prosesləri, açıq faylları və ya açıq soketləri göstərmir. Tez-tez Troyanlaşdırılan utilitilər arasında **ps** (sistem proseslərini göstərmək üçün), **netstat** (soketləri və şəbəkə əlaqələrini göstərmək üçün) və **top** (aktivliyə görə sıralanmış proses məlumatlarını göstərmək üçün) kimi alətlər daxildir.

Trojan olan faylları müəyyən etməyin sadə yolu **hash** analizindən istifadə etməkdir. **Hash** analizi, "önəmli" faylların tək istiqamətli kriptoqrafik **hash** dəyərlərini sistemdəki faylların **hash** dəyərləri ilə müqayisə edir. Əgər iki **hash** dəyəri uyğun gələrsə, bu, faylın **Trojan** bir versiya ilə əvəz olunduğunu göstərir.

Trojanları müəyyən etməyin ikinci üsulu bir qovluqdakı faylların **inode** nömrələrini müqayisə etməkdir. Əgər bir faylın **inode** nömrəsi həmin qovluqdakı digər faylların **inode** nömrələri ilə müqayisədə ardıcıllıqdan kənara çıxırsa, bu, faylın dəyişdirildiyini göstərə bilər.

Bir fayl sərt diskə yazıldıqda ona bir inode nömrəsi təyin edilir. Qısa müddət ərzində yaddaşa yazılan faylların **inode** nömrələri ardıcıl və ya bir-birinə yaxın olur. Bu, aşağıda göstərilən nümunədə əks olunub. Burada, **/bin** qovluğunun tərkibinin (qismən) siyahısı **inode** nömrəsinə görə sıralanmış şəkildə təqdim edilir (birinci sütunda yerləşir).

```
# ls -ali /bin | sort
130091 -rwxr-xr-x 1 root root 59100 Oct 5 11:50 cp
130092 -rwxr-xr-x 1 root root 15516 Oct 5 11:50 unlink
130093 -rwxr-xr-x 1 root root 161380 Oct 11 09:25 tar
130094 -rwxr-xr-x 1 root root 16556 Oct 5 11:50 rmdir
130095 -rwxr-xr-x 1 root root 26912 Oct 5 11:50 ln
130096 -rwxr-xr-x 1 root root 10804 Sep 30 08:49 hostname
130097 -rwxr-xr-x 1 root root 307488 Sep 21 17:26 tcsh
569988 -rwxr-xr-x 1 root root 76633 Jun 29 2004 ps
569990 -rwxr-xr-x 1 root root 92110 Jan 18 2004 netstat
```

Digital Forensics In Linux

Faylların sərt diskə hansı ardıcıllıqla yazıldığı, **inode** nömrələrinin artan sırası ilə aydın görünür. Lakin, **ps** və **netstat** əmrləri üçün **inode** nömrələrində bir anormallıq mövcuddur. Fayl başqa bir faylla əvəz olunduqda, onun **inode** nömrəsi dəyişir.

Çünki **Trojan** orijinal fayldan xeyli sonra quraşdırılıb, onun **inode** nömrəsi orijinal fayldan daha yüksək olacaq. Buna görə də, **Trojanları** müəyyən etməyin sadə üsulu, xüsusilə **rootkit**-in bir hissəsi ola biləcək fayllar üçün "**fərqlənən**" **inode** nömrələrini axtarmaqdır. Əvvəlki səhifədə göstərildiyi kimi, **ps** və **netstat** əmrlərinin **inode** nömrələri digər faylların **inode** nömrələri ilə müqayisədə əhəmiyyətli dərəcədə sırasızdır, bu da orijinal utilitin **Trojan** versiyası ilə əvəz olunması ehtimalını göstərir. Bu, yuxarıdakı **hash** analizindən fərqli olaraq, faylların mütləq **Trojan** atları olduğunu zəmanət etmir. Hər halda, əlavə yoxlama aparılmalıdır.

Digital Forensics In Linux

Uzantıları Dəyişdirilmiş Faylları Tapmaq

Bir faylı gizlətməyin sadə bir yolu, faylın uzantısını dəyişdirməkdir. Məsələn, "**chix.jpg**" faylını "**homework.doc**" olaraq dəyişdirmək, şübhəli məzmunu olan bir faylı, zərərsiz görünən bir fayla çevirir. Bu üsul, xüsusən Windows-da çox təsirli ola bilər, çünki Windows, faylın uzantısına əsaslanan bir ikon göstərəcək, faylın uzantısının fayl tipini doğru şəkildə əks etdirib-əks etdirmədiyindən asılı olmayaraq. Əvvəlcə izah edildiyi kimi, bir fayl növü başlığında (bəzən imza olaraq adlandırılır) əks olunur. Fayl başlığı tətbiqetmələrə faylın necə idarə olunacağını göstərən bir işarədir. Məsələn, bütün müasir **Microsoft Office** faylları aşağıdakı **8 baytlıq** imzalarla başlayır (qalın hərflərlə):

d0cf 11eO a1b1 1ae1 0000 0000 0000 0000

Qrafik fayllarını tapmaq üçün uzantısı dəyişdirilmiş faylları tapmağın bir yolu üç **GNU** utilitini birləşdirməkdir: **find**, **file** və **grep**. Proseduru ən yaxşı şəkildə izah etmək üçün nümunə ilə göstərmək olar.

1. **find** əmrindən istifadə edərək sabit disklərdəki bütün adi faylları tapın.
2. Bu əmrin nəticələrini **file** əmrinə verin, hansı ki, faylın başlıq məlumatlarına əsaslanaraq faylın növünü göstərir.
3. Bu əmrin nəticələrini **grep** əmrinə verin, qrafiklə əlaqəli açar sözləri axtarmaq üçün.

Aşağıda biz üç commandı birləşdirərək, uzantısı dəyişdirilmiş bütün qrafik şəkilləri müəyyən edirik:

find / -type f ! -name "*.jpg" ! -name "*.bmp" ! -name "*.png" -print0 | xargs -0 file | grep -i -f graphics.files

Bu addımlara bölündükdə daha asan başa düşülür:

1. / arqumenti, başlayacağımız kataloqu göstərir, burada kök kataloq göstərilir.

2. “**-type f**” bayrağı, xüsusi fayllar (cihazlar və ya kataloqlar kimi) deyil, adi fayllarla maraqlandığımızı göstərir. **Find** əmri, varsayılan olaraq rekursivdir, yəni **/** (kök) kataloqundan başlayaraq bütün adi faylları tapır.

3. Nida işarəsi (**!**) mötərizə içərisindəki məzmunu dəyişdirir və ***.jpg**, ***.png**, ***.bmp**, ya da ***.tiff** uzantısı olmayan faylları emal etmək istədiyimizi göstərir.

Digital Forensics In Linux

4. "**print0**" xüsusi formatlaşdırma əmri, “**find**” komandasının çıxışını növbəti əmri ötürmək üçün formatlaşdırmaq üçün tələb olunur.
5. Nəticələri, *.jpg, *.bmp və s. kimi uzantılara sahib olmayan faylların siyahısını "**xargs -0**" komandasına ötürürük, bu da hər bir fayl adını "**file**" komandasına göndərir. “**file**” hər bir faylın başlığını qiymətləndirir və faylın növü haqqında bir təsvir qaytarır.
6. Bu nəticələr **grep**-ə ötürülür, "**graphics.files**" faylında olan xüsusi açar sözləri axtarmaq üçün. "**grep**" üçün arqumentlər "**i**" (kiçik/böyük hərf həssas olmayan axtarış) və "**f graphics.files**" açar sözləri ehtiva edən fayldır: PNG, GIF, bitmap, JPEG və image.
Axtarışımız üç yanıltıcı adı və uzantısı olan fayl aşkar etdi:

```
# find / -type f !
( -name '*.jpg' -o -name '*.bmp' -o -name '*.png'
) -print0 | xargs-0 file | grep -if graphics.files
/var/cache/exec:  JPEG image data, JFIF standard 1.01
/var/log/ppp/0x12da2:  PC bitmap data, Windows 3.x format
/var/log/ppp/README.txt:  PNG image data,8-bit/color RGB
```

Axtarış düzgün olaraq üç faylı aşkar etdi: bir ***.jpg**, bir ***.bmp** və bir ***.png**, adları və ya uzantıları onların həqiqi növünü gizlətmək üçün dəyişdirilmişdi. Bu texnika düzgün işləyəcək, faylın başlığı tam qaldığı müddətcə.

Məlumatları əldə etdiyim mənbə:
https://link.springer.com/content/pdf/10.1007/0-387-31163-7_19.pdf