# LINUX FORENSIC

# İçindekiler Tablosu

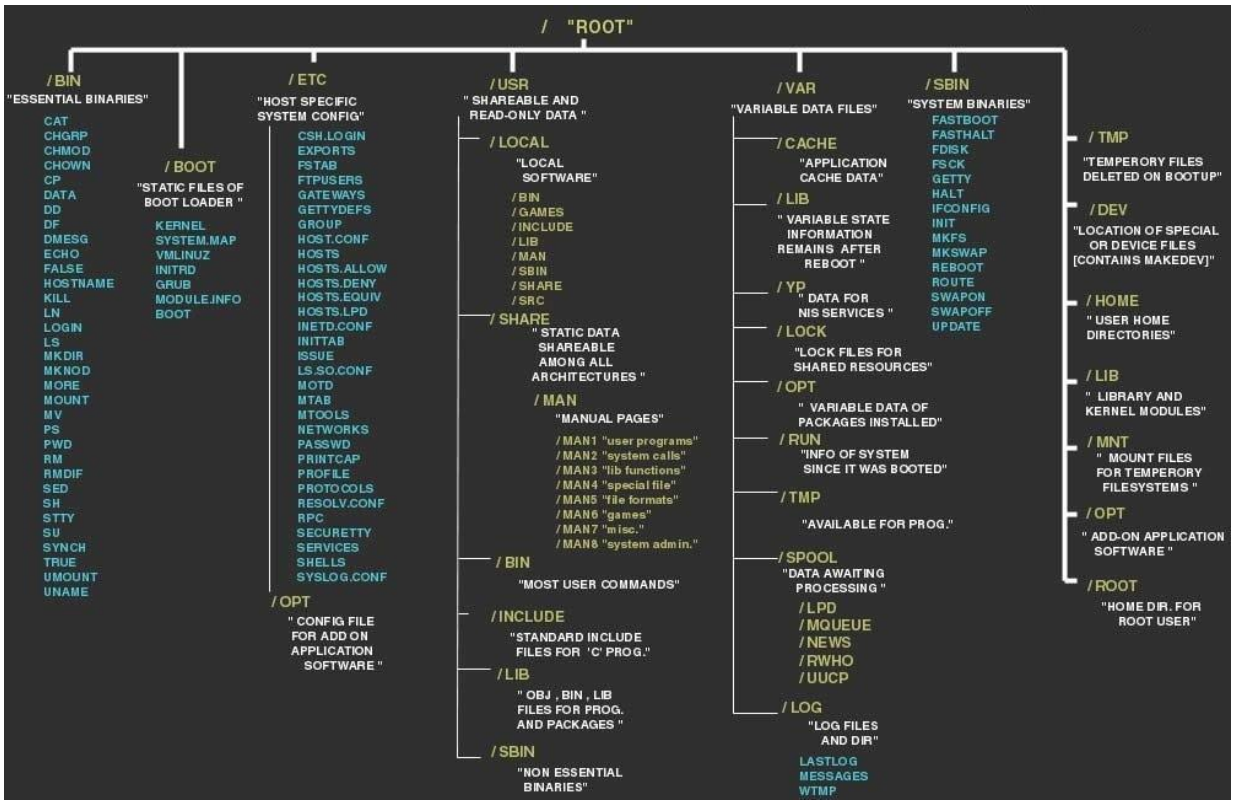
# ÖN SÖZ

Linux forensic, sistem olaylarını analiz ederek dijital suçlarla ilgili kanıt toplama sürecidir. Bu süreçte sistem logları, dosya yapıları, kullanıcı etkinlikleri ve ağ bağlantıları gibi veriler incelenir. Örneğin, `/var/log/` dizinindeki log dosyaları, kullanıcı aktiviteleri ve başarısız giriş denemeleri gibi kritik bilgileri içerir. SSH logları ve cron job'lar, uzaktan erişim ve zamanlanmış görevlerin analizi için kullanılırken, paket yönetim logları izinsiz kurulumları tespit etmeye yardımcı olur.

Adli inceleme sırasında, sistemin dikkatle yedeklenmesi ve güçlü Linux araçlarının (örn. `journalctl`, `netstat`) kullanımı önemlidir. Linux'un esnek yapısı, adli incelemeyi kolaylaştırırken aynı zamanda uzmanlık gerektirir. Bu alan, hem suçların aydınlatılmasında hem de güvenlik açıklarının kapatılmasında kritik bir rol oynar.

Linux adli bilişim süreci, Linux sistemlerinde depolanan verilerin incelenmesi, analiz edilmesi ve suç teşkil eden ya da olayın aydınlatılmasına yardımcı olabilecek kanıtların toplanmasıyla ilgilidir. Sistem günlükleri, kullanıcı etkinlikleri, ağ trafiği ve dosya sistemleri gibi bir dizi bileşen dikkatlice incelenerek, olayın nasıl gerçekleştiği ve kimlerin sorumlu olduğunu belirlenmeye çalışılır. Linux'un açık kaynak yapısı, güçlü analiz araçları sunarken iz bırakmadan kötü niyetli faaliyetlere de olanak tanıyabilir, bu nedenle adli süreç dikkat ve uzmanlık gerektirir.

## LINUX DIZIN YAPISI




Linux'da tüm veriler birer dizin olarak görünür. Diskteki verilerden tutunda sisteme takılan hoparlör gibi aygıtlara kadar. Yukarıda bu dizinlerin bir şeması verilmiştir. Bu dizinleri tek tek açıklamak gerekirse:

**/ (Root-Kök):** Diğer tüm dizinleri kapsayan, en büyük ebeveyn dizin / (root) dizinidir. Altında 15-16 dizin bulunur. / (root) dizininin silinmesi tüm sisteminizin silinmesi demektir.

**/bin:** Binary yazımının bir kısaltmasıdır. Burada sistemde çalıştırılan komutların derlenmiş halleri bulunur. Sistem için kritiktir. Bu dizin silinirse sistemde hiçbir komut çalıştırılamaz. Temel sistem komutları bu dizin içinde bulunur.

**/boot:** Kernel ve açılış dosyalarını içerir. Sistem ilk başlarken bu dosyalardan yüklenilir.

**/etc:** Sistemin yapılandırma dosyalarının birçoğu bu dizinde bulunur. Sistemde bir ayar yapılacağı zaman genellikle buradan yapılır.

**/opt:** Kendi kütüphane dosyalarını yükleyen uygulamaların kütüphane dosyalarını tutar. Silinmesi halinde bağlı uygulamalarda hata oluşabilir.

**/usr:** Çalıştırılabilir dosyalar, kaynak kodlar, dökümanlar, kütüphaneler gibi dosyalar içerir. Kullanıcıların uygulamaları burada depolanır. /usr/bin ve /usr/sbin dizinleri de temel sistem komutlarını içerir.

**/lib:** Kütüphane ve kernel modülleri tutulur.

**/sbin:** Sistem çalıştırılabilir dosyaları saklanır.

**/var:** Variable(değişken) ifadesinin kısaltılmışıdır. İçerisinde sürekli değişecek bilgileri tutar. Örneğin loglar veya mail bilgileri gibi.

**/tmp:** Geçici dosyaların bulunduğu kısımdır. Burada root kullanıcısı dışında her kullanıcı kendi dosyalarında değişiklik yapabilir.

**/proc:** Sistemle ilgili bilgilerin alınabileceği bölümdür. Aslında boştur fakat sistem çalıştırıldığında sistemle ilgili bilgiler alınabilir.

**/mnt:** Geçici mount edilmiş dosya sistemlerinin tutulduğu dizindir.

**/media:** Çıkarılabilir medya cihazlarının mount edildiği dizindir.

**/home:** Kullanıcıların ev dizinlerinin tutulduğu dizindir. İçindeki bir dizinin silinmesi durumunda silinen dizinin sahibi olan kullanıcı dosyalarını kaybetmiş olur.

**/srv:** srv services kelimesinin kısaltmasından gelir. Sistem tarafından sunulan servis dosyalarının bulunduğu dizindir.

**/lost+found:** Sistemde kaybolan dosyaların tutulduğu dizindir. Herhangi bir sebepten dosyanız kaybolursa bu dizine bakmakta fayda var.

Sistem hakkında bilgi toplamak için birçok alt dizin vardır. İşe ilk önce sistem hakkında bilgi toplamak ile başlanılabilir. Bunun **/etc/os-release** adresinde bulunan dosyayı cat ile okumak yardımcı olabilir. Ayrıca kernel ve sistem bilgilerini görüntülemek için **uname –a** komutu da kullanılabilir.

```
┌──(kali㉿kali)-[~]
└─$ cat /etc/os-release
PRETTY_NAME="Kali GNU/Linux Rolling"
NAME="Kali GNU/Linux"
VERSION_ID="2024.2"
VERSION="2024.2"
VERSION_CODENAME=kali-rolling
ID=kali
ID_LIKE=debian
HOME_URL="https://www.kali.org/"
SUPPORT_URL="https://forums.kali.org/"
BUG_REPORT_URL="https://bugs.kali.org/"
ANSI_COLOR="1;31"

┌──(kali㉿kali)-[~]
└─$ uname -a
Linux kali 6.6.15-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-05-17) x86_64 GNU/Linux
```

Sistem mimarisi, çekirdek sürümü ve işletim sistemi gibi bilgiler, doğru yazılımların seçilmesi, güvenlik yamalarının uygulanması ve gelecekteki ihtiyaçlara yönelik planlamalar yapılmasına olanak tanır. Sürümlerine göre sistemlere uygulanacak güvenlik önlemleri farklılık gösterebilir.

**/etc/passwd** dosyası sistemdeki kullanıcılar hakkında bilgiler verir. Cat ile bu dosya okunabilir. Çıktıda kullanıcı adı, şifre bilgileri, kullanıcı kimliği(uid), grup kimliği(gid), kullanıcı oturum açtığında çalıştırılan kabuk gibi bilgileri içerir. Windows'ta olduğu gibi, kullanıcı tarafından oluşturulan kullanıcı hesaplarının kullanıcı kimlikleri 1000 veya üzeridir. Grep, awk gibi komutlarla filtreleme yapılarak istenilen bilgilere daha çabuk ulaşılabilir.

```
┌──(kali㉿kali)-[~]
└─$ cat /etc/passwd| column -t -s :
root                x  0      0      root                          /root                  /usr/bin/zsh
daemon              x  1      1      daemon                        /usr/sbin              /usr/sbin/nologin
bin                 x  2      2      bin                           /bin                   /usr/sbin/nologin
sys                 x  3      3      sys                           /dev                   /usr/sbin/nologin
sync                x  4      65534  sync                          /bin                   /bin/sync
games               x  5      60     games                         /usr/games             /usr/sbin/nologin
man                 x  6      12     man                           /var/cache/man         /usr/sbin/nologin
lp                  x  7      7      lp                            /var/spool/lpd         /usr/sbin/nologin
mail                x  8      8      mail                          /var/mail              /usr/sbin/nologin
news                x  9      9      news                          /var/spool/news        /usr/sbin/nologin
uucp                x  10     10     uucp                          /var/spool/uucp        /usr/sbin/nologin
proxy               x  13     13     proxy                         /bin                   /usr/sbin/nologin
www-data            x  33     33     www-data                      /var/www               /usr/sbin/nologin
backup              x  34     34     backup                        /var/backups           /usr/sbin/nologin
list                x  38     38     Mailing List Manager          /var/list              /usr/sbin/nologin
irc                 x  39     39     ircd                          /run/ircd              /usr/sbin/nologin
_apt                x  42     65534                                /nonexistent           /usr/sbin/nologin
nobody              x  65534  65534  nobody                        /nonexistent           /usr/sbin/nologin
systemd-network     x  998    998    systemd Network Management    /                      /usr/sbin/nologin
systemd-timesync    x  992    992    systemd Time Synchronization  /                      /usr/sbin/nologin
messagebus          x  100    102                                  /nonexistent           /usr/sbin/nologin
tss                 x  101    104    TPM software stack,,,         /var/lib/tpm           /bin/false
strongswan          x  102    65534                                /var/lib/strongswan    /usr/sbin/nologin
tcpdump             x  103    105                                  /nonexistent           /usr/sbin/nologin
sshd                x  104    65534                                /run/sshd              /usr/sbin/nologin
usbmux              x  105    46     usbmux daemon,,,              /var/lib/usbmux        /usr/sbin/nologin
```

Linux sistemlerde **group (grup),** kullanıcıların ortak erişim izinlerini yönetmek için oluşturulan birimdir. Kullanıcılar gruplara dahil edilerek dosya ve kaynaklara erişimleri kolayca düzenlenir. Gruplar sayesinde, kullanıcılar arasında belirli bir kaynak paylaşımı, grup düzeyinde tanımlanan izinlerle kolayca sağlanır. Cat ile **/etc/group** dizini okunarak grup bilgilerine ulaşılabilir.

```
┌──(kali㉿kali)-[~]
└─$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kali
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:kali
fax:x:21:
voice:x:22:
cdrom:x:24:kali
floppy:x:25:kali
```

Örneğin fotoğrafta görüldüğü gibi adm kullanıcı grubunun gid'si 4 ve bu grupta kali kullanıcı bulunmaktadır yorumu yapılabilir.

**Journalctl** komutu, sistem ve servislerle ilgili tüm logları izlemek için kullanılabilir. Loglar tek bir yerde toplandığında hata teşhisi, performans ve güvenlik analizi gibi işlemler belirli bir tarih, servis veya öneme göre filtrelenerek kolaylıkla yapılabilir. Örneğin sistem boot kayıtları, servis hataları veya kernel mesajlarını incelemek fayda sağlayabilir. Journalctl ile kullanılabilecek komutlar şunlardır:

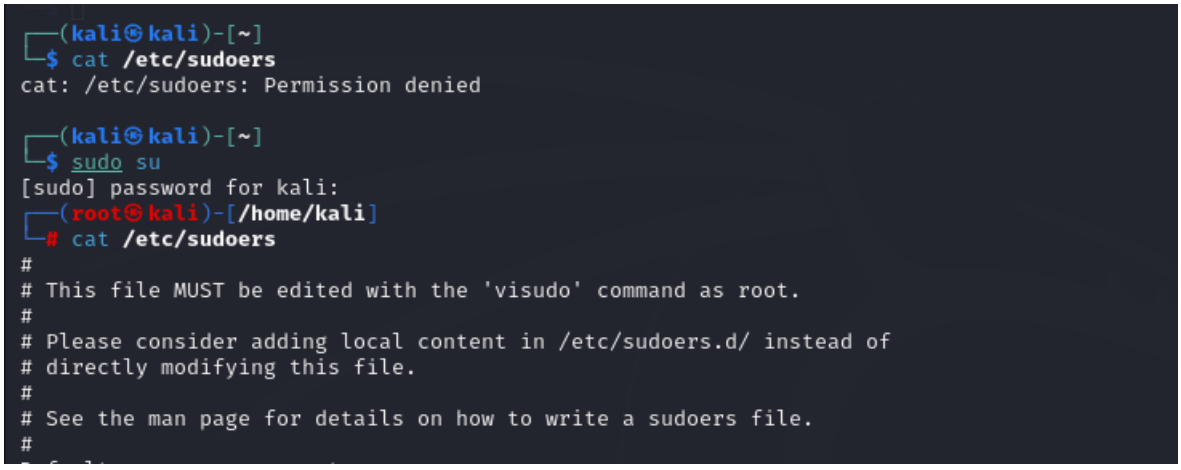
**journalctl –b:** Sistemin her açılışında oluşan logları görmek için kullanılır.

**journalctl -u servis_adi:** Belirtilen servisin loglarını incelemek için kullanılır**. --priority=3** parametresiyle sadece ilgili serviste oluşan hatalar görünebilir.

**journalctl –k:** Sadece kernel ile ilgili mesajları görüntüler. **–f** parametresiyle gerçek zamanlı izleme yapılabilir.

**journalctl –f:** Sistemde anlık olarak oluşan logları incelemek için kullanılır.


**`sudo` komutu,** Linux sistemlerinde bir kullanıcıya yönetici **root yetkileriyle** komut çalıştırma izni verir. Bu, sistemi değiştirme, yazılım yükleme veya kritik ayarları değiştirme gibi işlemleri güvenli bir şekilde yapmayı sağlar. Yetkili kullanıcılar, `sudo` kullanarak yalnızca belirli komutlar için geçici olarak root yetkisi alabilir. Sudo yetkisine sahip kullanıcıları incelemek için **/etc/sudoers** dosyası incelenebilir. Fakat bu dosyayı incelemek için de root yetkilerine sahip olmak gereklidir. Aksi halde permission denied hatası alınır. Sudo yetkisi almış kullanıcılar sistemde tam yetkiye sahip olduğu için istedikleri işlemleri hemen halledebilir.



Linux sistemlerde kimlik doğrulaması yapmış her kullanıcı için bir log tutulur. Bu loglar authentication log olarak tutulur. Bu loglar **/var/log/auth.log** altına kaydedilir. Dosya boyutundan dolayı inceleme yapmak için head, tail, more, less gibi komutları kullanmak yardımcı olabilir. Sistem yapılandırmasına göre auth.log dosyası yerine journalctl de authentication loglarını tutabilir. **journalctl | grep authentication** komutuyla authentication logları incelenebilir. Özellikle bir servis için inceleme yapılmak isteniyorsa ssh gibi **–u** parametresiyle servis adı belirtilerek loglar incelenebilir.

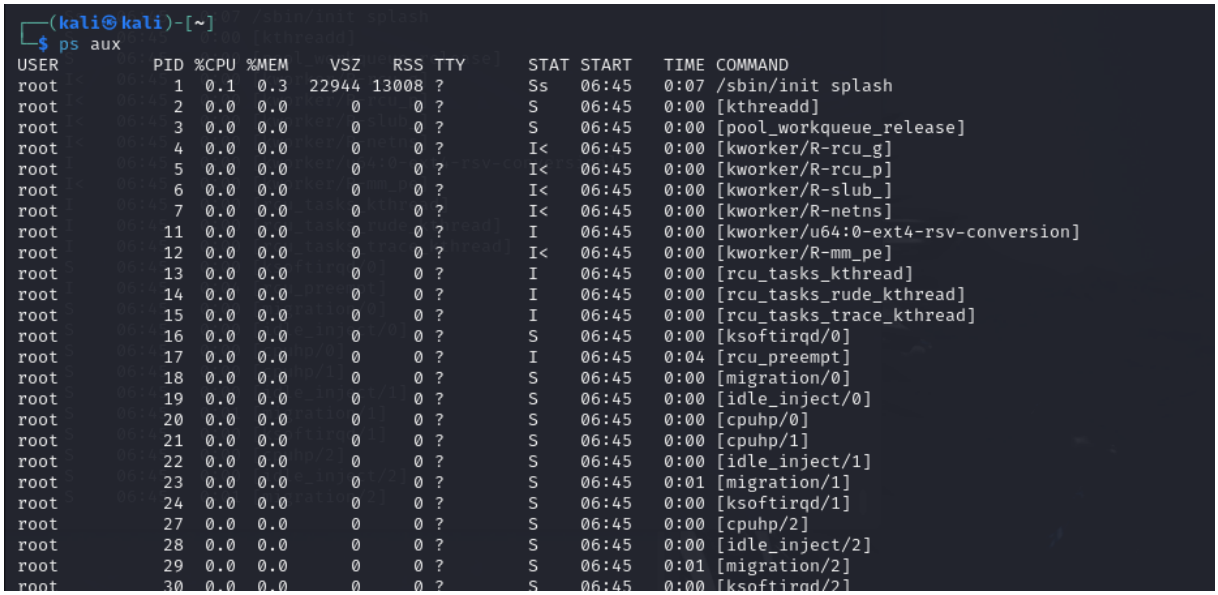

Örneğin ilk log da sudo kullanıcısıyla bir işlem yapılmaya çalışılmış fakat kimlik doğrulama başarısız olmuştur. Altındaki lightdm logunda ise bir kullanıcının giriş yapmaya çalıştığını ancak kimlik doğrulamanın başarısız olduğunu gösteriyor. LightDM, sistemin giriş ekranını sağlayan bir oturum yöneticisidir. Bu loglar incelenerek sisteme yapılan oturum girişimleri tespit edilebilir.

Sistemin saat dilimini öğrenmek için windows'ta olduğu gibi bir timezone alanı bulunur. Bu alanı görmek için **/etc/timezone** dizinin cat ile okumak yeterli olacaktır.

Network interfaceleri hakkında bilgi sahibi olmak için **/etc/network/interfaces** altına bakılabilir. Farklı arayüzlerin mac ve ip adresleri bilgilerine ulaşmak için **ip address show** komutu kullanılabilir.


Ayrıca canlı bir sistemdeki aktif bağlantıları görmek için netstat komutu kullanılabilir. netstat –natp komutuyla sistemdeki bağlantıları görünür. –n, port numaralarını ve IP adreslerini numerik formatta gösterir; -a, tüm bağlantıları gösterir, dinleme (listen) bağlantıları da dahil olmak üzere; -t, TCP bağlantılarını gösterir; -p, bağlantıyı oluşturan sürecin adını/pid'sini gösterir.

Sistemde anlık olarak çalışan tüm processleri görmek için **ps –aux** komutu kullanılır. Sistemde çalışan processleri incelemek ve bilgi dışında olan processleri tespit etmek çok önemlidir. Tespit edilen processler pid'si kullanılarak durdurulabilir.



```
┌──(kali㉿kali)-[~]
└─$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.3  22944 13008 ?        Ss   06:45   0:07 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    06:45   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    06:45   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   06:45   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   06:45   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   06:45   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   06:45   0:00 [kworker/R-netns]
root          11  0.0  0.0      0     0 ?        I    06:45   0:00 [kworker/u64:0-ext4-rsv-conversion]
root          12  0.0  0.0      0     0 ?        I<   06:45   0:00 [kworker/R-mm_pe]
root          13  0.0  0.0      0     0 ?        I    06:45   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    06:45   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    06:45   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    06:45   0:00 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    06:45   0:04 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    06:45   0:00 [migration/0]
root          19  0.0  0.0      0     0 ?        S    06:45   0:00 [idle_inject/0]
root          20  0.0  0.0      0     0 ?        S    06:45   0:00 [cpuhp/0]
root          21  0.0  0.0      0     0 ?        S    06:45   0:00 [cpuhp/1]
root          22  0.0  0.0      0     0 ?        S    06:45   0:00 [idle_inject/1]
root          23  0.0  0.0      0     0 ?        S    06:45   0:01 [migration/1]
root          24  0.0  0.0      0     0 ?        S    06:45   0:00 [ksoftirqd/1]
root          27  0.0  0.0      0     0 ?        S    06:45   0:00 [cpuhp/2]
root          28  0.0  0.0      0     0 ?        S    06:45   0:00 [idle_inject/2]
root          29  0.0  0.0      0     0 ?        S    06:45   0:01 [migration/2]
root          30  0.0  0.0      0     0 ?        S    06:45   0:00 [ksoftirqd/2]
```

Çıktıdaki her bir sütün farklı bir anlama gelir.

**USER:** İşlemi başlatan kullanıcı adı

**PID:** İşlem Kimliği (Process ID), yani işlemin benzersiz numarası

**%CPU:** CPU kullanım yüzdesi

**%MEM:** Bellek kullanım yüzdesi

**VSZ:** Toplam sanal bellek boyutu (kilobyte cinsinden)

**RSS:** Resident Set Size, yani işlemin fiziksel bellek boyutu (kilobyte cinsinden)

**TTY:** İşlemin bağlı olduğu terminal

**STAT:** İşlem durumu (örneğin, çalışıyor, beklemede, durmuş)

**START:** İşlemin başlatılma zamanı

**TIME:** İşlem tarafından kullanılan toplam CPU zamanı

**COMMAND:** İşlemi başlatan komut veya program

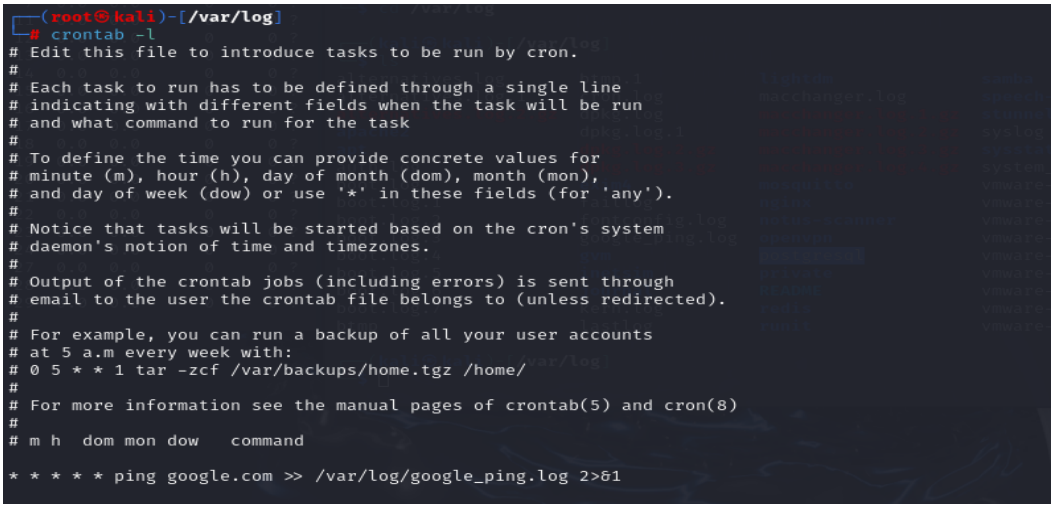
CPU ve bellek kullanım yüzdelerini incelemek anomali tespitinde büyük önem taşır. Ayrıca sistemde çalıştırılan komutları da incelemek gereklidir. Anormal bir komut çalıştıran programlar tespit edilip aksiyon alınabilir.

Linux sistemde kullanıcıların terminalde girdiği komutlar sistem kabuğuna göre bir history dosyasına kaydedilir. Örneğin kali Linux eski versiyonları varsayılan olarak bash kabuğunu kullanırken yeni versiyonlar zsh kabuğunu kullanır. **.zsh_history** dosyası incelendiğinde terminalde çalıştırılan komutlar incelenip anomali tespiti yapılır.


Şekilde networkmanager'ı yeniden başlattıktan sonra bir Shell dosyası zipten çıkarılmış ve daha sonra crontab servisi yeniden başlatılmıştır.

Sistemde syslog servisi aktifse bu loglar incelenebilir. Syslog, sistem başlangıç mesajları, kernel aktiviteleri, ağ bağlantıları ve sistem hataları gibi geniş bir yelpazede olayları kaydeden kapsamlı bir günlük dosyasıdır. Syslog dosyasını incelemek, sistem olaylarını izlemek, belirli hizmetlerin ne zaman başlatıldığını veya durdurulduğunu tespit etmek ve yetkisiz girişler veya sistem hataları gibi olağandışı davranışları belirlemek için kritik bir öneme sahiptir. Eğer sistemde syslog aktif değilse journalctl üzerinden inceleme yapılabilir.

Cron jobs, belirli aralıklarla otomatik olarak çalışan zamanlanmış görevlerdir. Bu görevler genellikle sistem yöneticileri tarafından bakım, yedekleme ve diğer otomatik işlemler için yapılandırılır. Cron jobları analiz etmek, belirli aralıklarla çalışacak şekilde ayarlanmış yetkisiz veya kötü amaçlı görevlerin varlığını ortaya çıkarabilir. **/var/spool/cron/crontabs/** dizininde sistemde var olan cron joblar incelenir. **Crontab –l** komutuyla da o anki kullanıcı için oluşturulmuş cronjoblar listelenir.



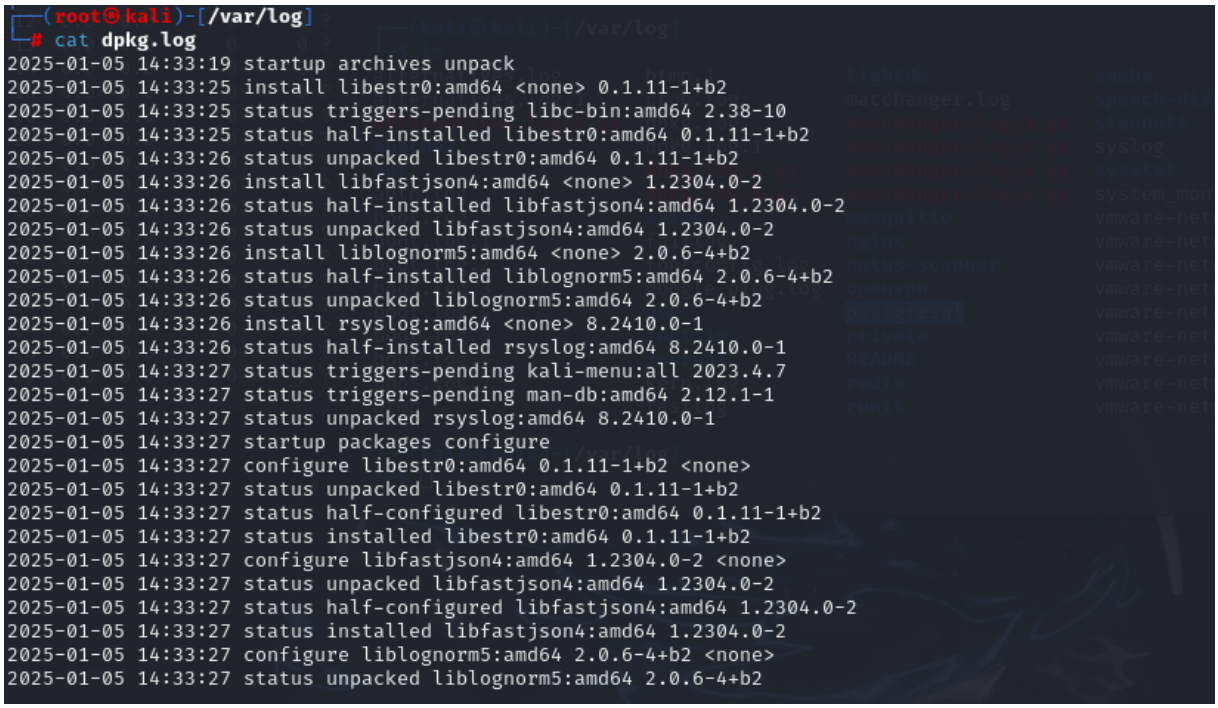
Bu sistemde her dakikada bir google'a ping atacak ve loglarını /var/log/Google_ping.log adlı bir dosyaya kaydedecek bir crontab oluşturulmuş. İncelenilen sistemde herhangi bir cronjob yoksa no crontab for user uyarısı verir. Bu sistemde bu kullanıcı için herhangi bir zamanlanmış görev olmadığını gösterir.

SSH (Secure Shell), güvenli uzaktan giriş ve komut yürütme için kullanılan bir protokoldür. SSH ile ilişkili loglar ve yapılandırma dosyaları, uzaktan erişim girişimlerini, zayıf veya ele geçirilmiş anahtarların kullanımını ve SSH hizmetinin güvenlik durumunu anlamak için kritik öneme sahiptir. Bu incelemeler, saldırganların SSH üzerinden bir sisteme kalıcı erişim sağlamış olabilecekleri durumları tespit etmek için önemlidir**. /etc/ssh** dizini altında ssh sunucu yapılandırma dosyaları bulunur. **/etc/ssh/sshd_config;** SSH bağlantıları, şifreleme yöntemleri, portlar, kimlik doğrulama seçenekleri gibi ayarlar yapılan SSH sunucusunun yapılandırma dosyasıdır. **/etc/ssh/ssh_config**; SSH istemcilerinin sunucularla nasıl bağlantı kuracağına dair ayarları içeren SSH istemci yapılandırma dosyasıdır ve genellikle istemci tarafında bulunur.

Ssh üzerinden giriş denemesi loglarını ise auth.log dosyasından incelenir. **sudo cat /var/log/auth.log | grep sshd** komutu auth.log dosyasını yazdırırken sadece sshd loğlarını görmek için grep ile bir filtreleme yapar.

Paket yönetimi loglarını analiz etmek, yetkisiz yazılımların kurulumunu tespit etmeye, olası zararlı yazılımlar veya backdoorları belirlemeye ve sistemdeki değişikliklerin zaman içindeki izini sürmeye yardımcı olur. Çünkü saldırgan sistemi ele geçirmek ve kontrol altında tutmak amacıyla kötü amaçlı paketler yüklemiş olabilir. Bu loğları incelemek için sistemin paket yönetim aracını bilmek gerekir. Debian tabanlı sistemlerde bu dpkg'dir ya da red hat tabanlı sistemlerde yum'dur.

Paket yönetim logları debian sistemlerde **/var/log/dpkg.log** olarak tutulur. Buradan sisteme kurulan paketler hakkında incelemeler yapılıp anomali tespiti yapılabilir.




Üstteki fotoğrafa bakıldığında librestro, libfastjson4, rsyslog, liblognorm5 gibi paketlerin yüklendiği ve yapılandırıldığı görülüyor. Tarih ve zaman damgaları ise yanlarında görülmektedir. Analiz sırasında belirli bir tarih aralığında filtreleme yapmak için kullanılabilir.

## HAZIRLAYAN

Ayşe BALCI Linkedin