RBAC and ABAC in Linux

Linux Sistemlerde kullanıcıların sistem üzerindeki erişimleri kontrol edebilmek için Attribure Base Access Control ve Role Base Access kontrol olak üzere iki farklı erişim kontrol yaklaşımı kullanılmaktadır. Bu yaklaşımlara bakıldığında;

- Attribute Base Access Control, sistem üzerindeki kullanıcı ve kaynakların nitelikleri göz önünde bulundurularak bunların çeşitli kurallar doğrultusunda yönetilmesini sağlayan yaklaşımdır. Bu yaklaşım için DAC mekanizması kullanılmaktadır.
- Role Base Access Control, kullanıcılara verilen sorumluluklar ve bu doğrultuda ihtiyaç duyacağı yetkiler kapsamında kullanıcı bazlı erişimlerin tanımlandığı yaklaşımdır. Bu yaklaşım için MAC mekanizması kullanılmaktadır.

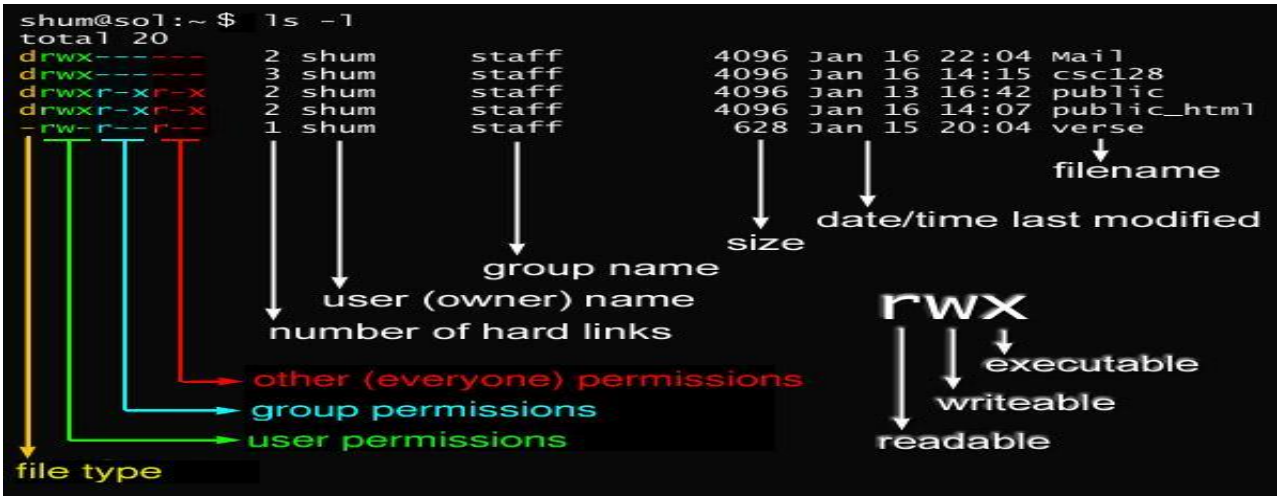
DAC (Discretionary Access Control)

DAC, sistem üzerinde bulunan dosya ve dizinler için tanımlanan erişim kontrol mekanizmasıdır. DAC mekanizması sayesinde dosya veya dizinlere hangi kullanıcıların erişebileceği kontrol edilebiliyor. DAC mekanizması bu kontrolü sağlayabilmek için sistem üzerinde oluşturulan her nesneye (dosya veya dizin) nesneyi oluşturan kullanıcı, nesneyi oluşturan kullanıcının dahil olduğu grupta bulunan kullanıcılar ve diğer kullanıcılar olmak üzere toplamda üç farklı kapsam tanımlıyor. Bu kapsamlarda bulunan kullanıcıların dosya üzerinde yapabilecekleri işlemler için de her kapsama özel işlem kısıtları/haklar tanımlanıyor (r->okuma, w->yazma, x->çalıştırma) (Nesneler üzerinde tanımlı DAC hakları CLI üzerinde "**ls -l**" komutu kullanarak görüntülenebilir).

DAC mekanizması için bir kullanıcı sistem üzerindeki bir dosyaya erişmek istediğinde öncelikle kullanıcının dosya için hangi kapsama dahil olduğu tespit ediliyor (Dosya sahibi mi? Dosya sahibiyle aynı grupta mı? Diğer kullanıcı mı?). Kullanıcının dahil olduğu kapsam tespit edildikten sonra dahil olduğu kapsamda hangi erişim yetkilerine sahip olduğu kontrol ediliyor. Bu doğrultuda kullanıcının nesneye erişip erişemeyeceğine karar veriliyor.

Bir dosya veya dizin üzerinde tanımlanan kullanıcı yetkilerini/haklarını değiştirmek için (bu işlem için kullanıcının kendi oluşturduğu dosya olması veya dosya üzerinde değişiklik yapabilecek yetkiye sahip olması gerekiyor) "**chmod <<User> <Gruop> <Other>> <File Name>**" komutu kullanılıyor. Bu komut kullanılırken kullanıcılar için yetkiler iki farklı şekilde temsil edilebiliyor.

- Yetkiler için belirlenmiş sayısal değerler bulunmaktadır (r->okuma(4), w->yazma(2), x->çalıştırma(1). Kullanıcı kapsamı için tanımlanmak istenen yetkilerin sayısal değerleri toplamı kullanılarak tanımlanacak işlem kısıtları ifade edilebiliyor. Örnek olarak;
    - Sadece dosya sahibinin okuma,yazma ve çalıştırma yetkisi için -> "**chmod 700 file.txt**" (rwx-> 7, 0, 0).
    - Dosya sahibine rwx, dosya sahibinin bulunduğu gruptaki kullanıcılar için rw yetkisi için -> "**chmod 760 file.txt**" (rwx->7, rw->6, 0).
- Tanımlanacak kapsamın baş harfi (u-> user, g-> group, o-> other) ve tanımlanmak istenen yetkilerin sembolik ifadeleri kullanılarak tanımlamalar yapılabiliyor. Örnek olarak;
    - Sadece dosya sahibine okuma yetkisi için -> "**chmod u=r file.txt**".
    - Diğer kullanıcılar için rwx yetkileri tanımlamak için -> "chmod o-rwx file.txt".



Nesnelerde tanımlanan işlem kısıtları konusunda bilinmesi gereken üç özel bit daha bulunuyor. Bu bitler SUID, SGID ve STICKY bitleridir. Bu biterin özellikleri;

- STICKY (1), nesneyi sadece sahibinin silebileceğini gösterir.
- SGID (2), herhangi bir kullanıcının nesneyi nesne sahibinin dahil olduğu grup için tanımlanan işlem kısıtlarıyla kullanabileceğini gösterir.
- SUID (4), herhangi bir kullanıcının nesneyi nesne sahibi için tanımlanan işlem kısıtlarıyla kullanabileceğini gösterir.

Daha açıklayıcı bir örnek vermek gerekirse aşağıdaki görselde göründüğü gibi sadece dosya sahibi için rwx yetkileri tanımlanmış bir dosyada SGID biti set edilmiştir. Bu durumda dosya sahibi ile aynı grupra bulunan bir kullanıcı bu dosyayı dosya sahibinin sahip olduğu (bu görselde rwx yetkileri tanımlanmış) yetkilerde çalıştırabileceği anlamına geliyor.

```
0 -rwx--S--- 1 vm-user vm-user    0 Ara 12 15:50 file.txt
```

Benzer şekilde sadece dosya sahibi için rwx yetkisi tanımlı bir dosya üzerinde SUID biti set edilmiş. Bu durumda herhangi bir kullanıcı bu dosyayı dosya sahibinin sahip olduğu haklarda çalıştırabileceğini gösteriyor. STİCKY biti için de benzer durum geçerlidir.

```
0 -rws------ 1 vm-user vm-user    0 Ara 12 15:50 file.txt
```

SUID, SGID ve STICKY biti ayrı ayrı tanımlanabileceği gibi kombine şekilde kullanılarak da tanımlanabiliyor. Bu bitler dosya hakları tanımlarken kullanılan sayısal ifadelerin en başında temsil ediliyor. Örnek olarak;

- 755 dosya haklarına sahip “file.txt” isimli bir dosya üzerinde sadece SUID biti set edilmek isteniyorsa “chmod **4**755 file.txt” komutu kullanılıyor.
- 755 dosya haklarına sahip “file.txt” isimli bir dosya üzerinde sadece SUID, SGID ve STICKY bitleri set edilmek isteniyorsa “chmod **7**755 file.txt” komutu kullanılıyor.

ACL (Access Control List)

Normal durumlarda bir nesneye erişebilecek kullanıcı profillerini DAC mekanizmasıyla kullanarak kontrol edilebiliyordu. DAC mekanizmasında fark edilebileceği gibi sadece bir kullanıcı veya bir grup (sadece dosya sahibinin dahil olduğu grup) için izinler tanımlanabiliyordu. ACL, nesnelere erişim konusunda daha esnek kontroller oluşturabilmek için kullanılan mekanizmadır. Örnek olarak bir dizin üzerinde birçok grubun erişebilmesi gerekebilir. Bu erişim kısıtı için ACL kullanılmaktadır.

Bir dizin veya bir dosya için geçerli ACL'ler "**getfacl**" komutuyla görüntülenebilir. Nesne üzerinde daha önce bir ACL tanımı yapılmamışsa çıktı olarka sadece dosya sahibi, grup ve diğer kullanıcılar için tanımlanan kısıtlamalar görünecektir.

```
vm-user@Ubuntu-VM:~/Desktop$ getfacl file.txt
# file: file.txt
# owner: vm-user
# group: vm-user
user::rw-
group::rw-
other::r--
```

Bir nesne üzerinde ACL tanımı "**setfacl <Operation> <Action> <File Name>**" fornatında gerçekleştiriliyor (Detaylı kullanımı için "**setfacl -h**" komutu kullanabilirsin). Burada;

- Operation; –m parametresi yeni bir kural ekleneceği zaman kullanılıyor. –x parametresi tanımlı bir kural kaldırılacağı zaman kullanılıyor.
- Action;" <User/Group>:<Name>:<Permission>" formatında kuralın uygulanacağı kapsam ve yetkileri ifade ediliyor. Tanımlara örnek olarak;
    - setfacl –m u:user1name:rwx file.txt
    - setfacl –m g:group1name:rwx file.txt
    - setfacl –x file.txt

```
vm-user@Ubuntu-VM:~/Desktop$ setfacl -m u:furkan:rwx file.txt
vm-user@Ubuntu-VM:~/Desktop$ getfacl file.txt
# file: file.txt
# owner: vm-user
# group: vm-user
user::rw-
user:furkan:rwx
group::rw-
mask::rwx
other::r--

vm-user@Ubuntu-VM:~/Desktop$ setfacl -x u:furkan file.txt
```

MAC (Mandatory Access Control)

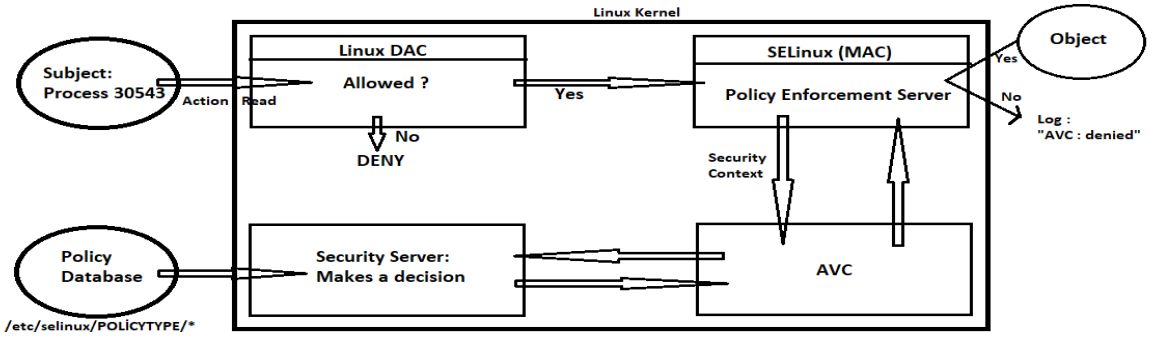
MAC, sistem üzerindeki nesnelere erişim kontrollerinin düzenlenmesini sağlayan kontrol mekanizmasıdır (Erişim kontrolünün en üst seviyisi olarak da tanımlanabilir). MAC mekanizmasında sistem üzerindeki nesnelere ve kullanıcılara güvenlik düzeyine göre etiketler tanımlanır ve erişim kontrolleri bu etiketler üzerinden gerçekleştirilir (Bir nesnenin güvenlik düzeyi Confidential, Secret ve Top-Secret olmak üzere üç farklı şekilde tanımlanabiliyor). Kullanıcılar, yalnızca güvenlik etiketlerinin kendilerine yetki verdiği bir kaynaktaki bilgilere erişebilir. Kullanıcının güvenlik etiketi yeterli yetkiye sahip değilse, kullanıcı kaynaktaki bilgilere erişemez.

MAC politikaları sadece sistem yöneticisi tarafından tanımlanır ve kernel tarafından uygulanır. Sistem yöneticisi dışında herhangi bir kullanıcı tarafından bu tanımlamalar değiştirilemez. Bu sayede kullanıcıların farkında veya farkında olmadan sisteme zarar verebilmelerinin veya yetkisi dışında bulunan nesnelere erişim sağlayabilmesi engellenir.

MAC mekanizması sistem üzerinde SELinux veya AppArmor gibi uygulamalar kullanılarak uygulanmaktadır. Bu uygulamalar Linux dağıtımına göre değişiklik göstermektedir. Örnek olarak Fedora, Centos gibi dağıtımlarda SELinux kullanırken , Ubuntu, Suse, Mandriva gibi dağıtımlarda AppArmor kullanılmaktadır (Detaylı bilgi için SELinux ve AppArmor yazılımlarımı inceleyebilirsin).

MAC ve DAC Mekanizmalarının Çalışma Düzeni

Bir kullanıcı veya proses sistem üzerindeki bir nesneye erişim sağlamak istediğinde öncelikle isteğin DAC mekanizmasına uygun olup olmadığı kontrol edilir. Erişim isteği DAC mekanizmasına uygun değilse erişim doğrudan reddedilir. Erişim isteğinin DAC mekanizmasına uygun olduğu görülürse MAC mekanizması (örnek olarak SELinux) ile sistem yöneticisi tarafından belirlenen politikalara uygun olup olmadığı kontrol edilir. Erişim isteğinin tanımlı politikalara uygun olduğu görülürse nesneye erişim sağlanır, aksi durumda erişim engellenir ve engellenen erişim isteğinin detayları loglanır (AVC üzerine kaydedilir).



NOT

- Subject/Konular, sistem kaynakları üzerinde çalışan kullanıcılar, süreçler, programlar veya iş parçacıklarını temsil etmek için kullanılmaktadır.
- Object/Nesneler, dosyalar, dizinler, hizmetler, giriş/çıkış aygıtları gibi sistemin kaynaklarını temsil etmek için kullanılmaktadır.
- SELinux modunun Enforcing olduğu görüldükten sonra SELinux modundaki değişimlere önlem olarak belirli sıklıklarda “/var/log/audit/audit.log” (audit.log dosyası sistem üzerinde gerçekleştirilen değişimlerin kaydını tutmak için kullanılıyor) dosyası kullanırak “enforcing” değeri üzerinde veya “/etc/sysconfig/selinux” dosyası üzerinde değişim yapılıp yapılmadığının kontrol edilmesi gerekiyor (Örnek olarak “sudo cat /var/log/audit/audit.log |grep “enforcing=0”).

Terminolojiler

- AVC (Access Vector Cache), MAC mekanizmasında istem kaynaklarına yönelik yapılan erişim isteklerinin tutulduğu bir tür log dosyasıdır.

KAYNAKLAR

- https[:]//www.redhat.com/en/topics/linux/what-is-selinux
- https[:]//web.mit.edu/rhel-doc/5/RHEL-5-manual/Deployment_Guide-en-US/ch-selinux.html
- https[:]//linuxhint.com/a-beginners-guide-to-selinux-on-centos/
- https[:]//www.incibe-cert.es/en/blog/basic-access-control
- https[:]//docs.fedoraproject.org/en-US/quick-docs/troubleshooting_selinux/#:~:text=SELinux%20decisions%2C%20such%20as%20allowing,usr%2Fsbin%2Fhttpd%22%20.
- https[:]//docs.freebsd.org/en/books/handbook/mac/
- https[:]//selinuxproject.org/page/Main_Page
- https[:]//www.maketecheasier.com/understanding-apparmor-in-ubuntu-linux/
- https[:]//manpages.ubuntu.com/manpages/trusty/man4/mac.4freebsd.html#:~:text=Introduction%20The%20Mandatory%20Access%20Control,a%20loadable%20security%20policy%20architecture.
- https[:]//www.howtogeek.com/118222/htg-explains-what-apparmor-is-and-how-it-secures-your-ubuntu-system/
- https[:]//ubuntu.com/server/docs/security-apparmor
- https[:]//www.imperva.com/learn/data-security/role-based-access-control-rbac/
- https[:]//www.1kosmos.com/identity-management/mandatory-access-control/
- https[:]//www.redhat.com/sysadmin/linux-access-control-lists