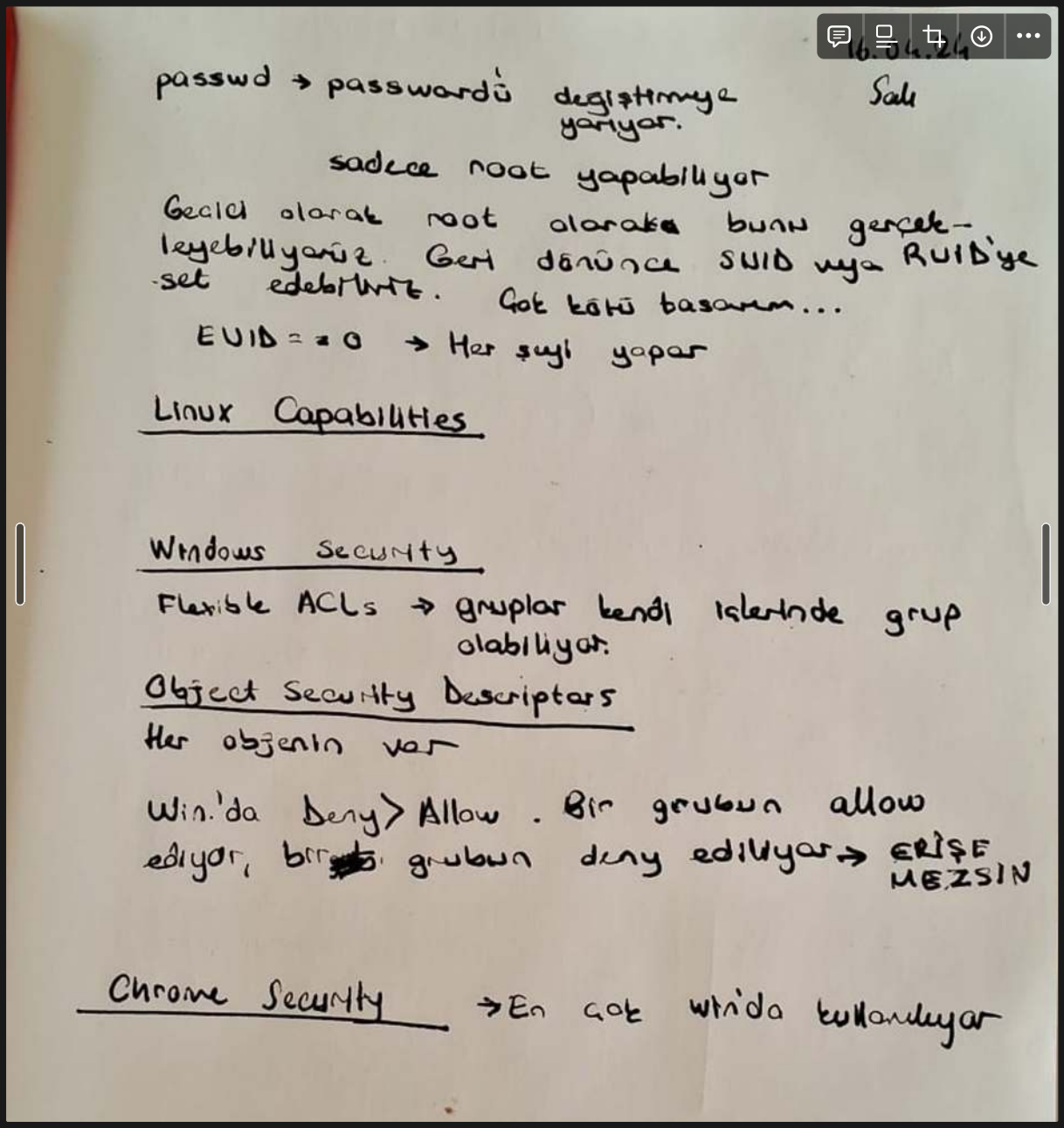
ağ güvenliği - OS



passwd → passwordü değiştirmeye yarıyor.

Salı

sadece root yapabiliyor

Gecici olarak root olarak bunu gerçekleyebiliyoruz. Geri dönünce SUID veya RUID'ye set edebiliyoruz. Çok kötü tasarım...

EUID == 0 → Her şeyi yapar

Linux Capabilities

Windows Security

Flexible ACLs → gruplar kendi içlerinde grup olabiliyor.

Object Security Descriptors

Her objenin var

Win.'da Deny > Allow . Bir grubun allow ediyor, bir grubun deny ediliyor → ERİŞEMEZSİN

Chrome Security → En çok win'da kullanılıyor

UNIX SECURITY MODEL

16.04.24
Salı

UID=0 → root
↳unique

GID → Group ID

Access Control List (ACL)

Objeye access verilenlerin listesi

Bunun group based daima Role Based Access Control (RBAC) denir.

UNIX PROCESSES

Prosesler izoledir, birbirlerinin memoryisine erisemez

Process'i calistirinca, calistiranin UID'si ile calisir. UID'nin yetkisi kadar yetkili dir.

Process'i root baslattiysa process daha düşük bir yetkiye kendini ayarlayabilir.

Process forklandiginda parent'in UID'sini tutar.

Effective UID → izinleri belirler
Real UID → Process'i baslatan
Saved UID → EUID'i saklıyoruz. Çünkü degisirse ori-ginaline ihtiyaç duyarız

} Normalde 3'ü de esit

root UID'lerin tümünü istedigine set edebilir.
fakat EUID'yi RUID veya SUID'ye set edebilir only.

setuid(x) → hepsi x olur
Çalışabilir
seteuid → tehlikeli
↳safe

16.04.24
Salı

Güvenlik Prensipleri

Defense in Depth

Safe dillerde buglar bulunabilir.
Tüm layerlarda koruma olmalıdır.
Chrome → Tablar arasında malicious geçişini engellemeli
OS → Processes arası erişimi engellemeli

Prensipler

- Defense in Depth
- Principle of Least Privilege
- Privilege separation
- Open design
- KISS

Least Privilege

Kullanıcının işlerini yapmak için ihtiyaç duyduğu yetkinin minimumunun verilmesi.

Privilege Separation

Çok kritik durumlar için birden fazla yetkilinin işlem yapması gerekir. Böylece saldırgan bir yetkilinin bilgilerini ele etse bile diğerini edemez. Sadece user değil olarak düşünme, app, process, domain vs. de olabilir.
subjects