

# Bölüm 9: Güvenlik

## İşletim Sistemleri

# Güvenlik Hedefleri Ve Tehditler

- .

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service
Exclusion of outsiders	System takeover by viruses

# İzinsiz Kullanıcı

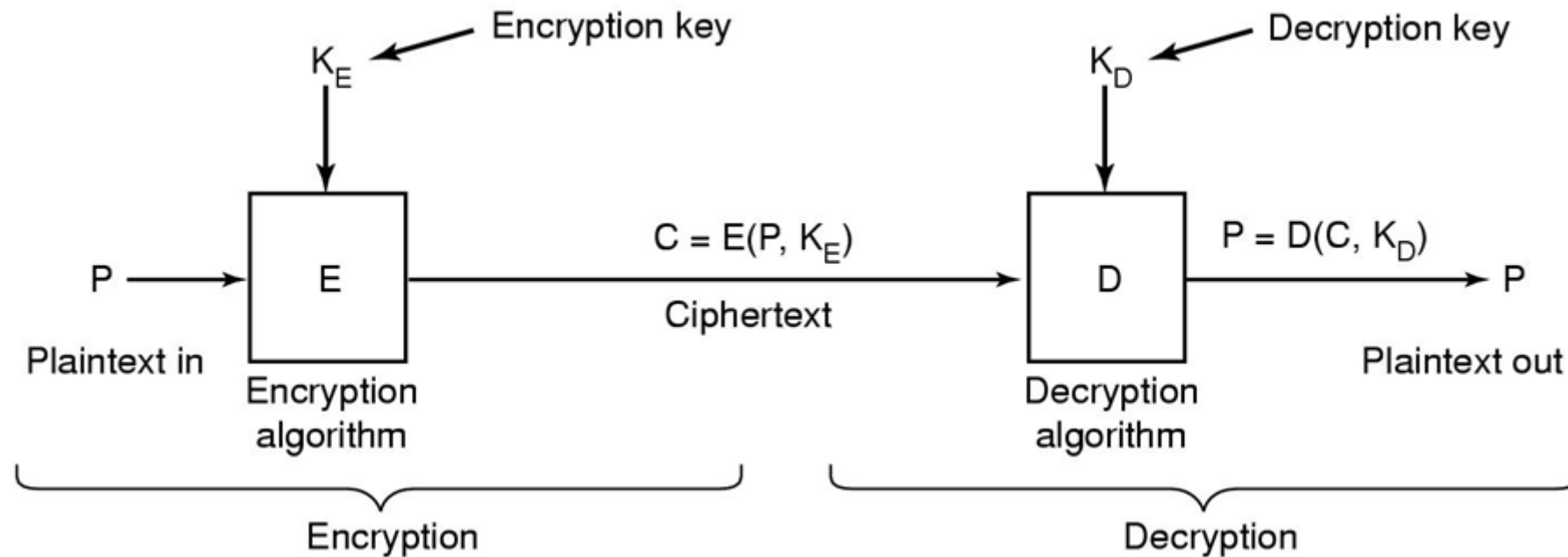
- Teknik olmayan kullanıcılar tarafından rastgele gözetleme.
- İçeridekiler tarafından gözetleme.
- Para kazanmak için kararlı girişimler.
- Ticari veya askeri casusluk.

# Kazayla Veri Kaybı

- Yanlışlıkla veri kaybının yaygın nedenleri:
- Kader: yangınlar, seller, depremler, savaşlar, isyanlar veya yedek bantları kemiren fareler.
- Donanım veya yazılım hataları: CPU arızaları, okunamayan diskler veya teypler, telekomünikasyon hataları, program hataları.
- İnsan hataları: yanlış veri girişi, yanlış teyp veya CD-ROM takma, yanlış program çalıştırma, kayıp disk veya teyp veya başka bir hata.

# Kriptografinin Temelleri

- Düz metin ve şifreli metin arasındaki ilişki.



# Gizli Anahtarlı Kriptografi (secret key)

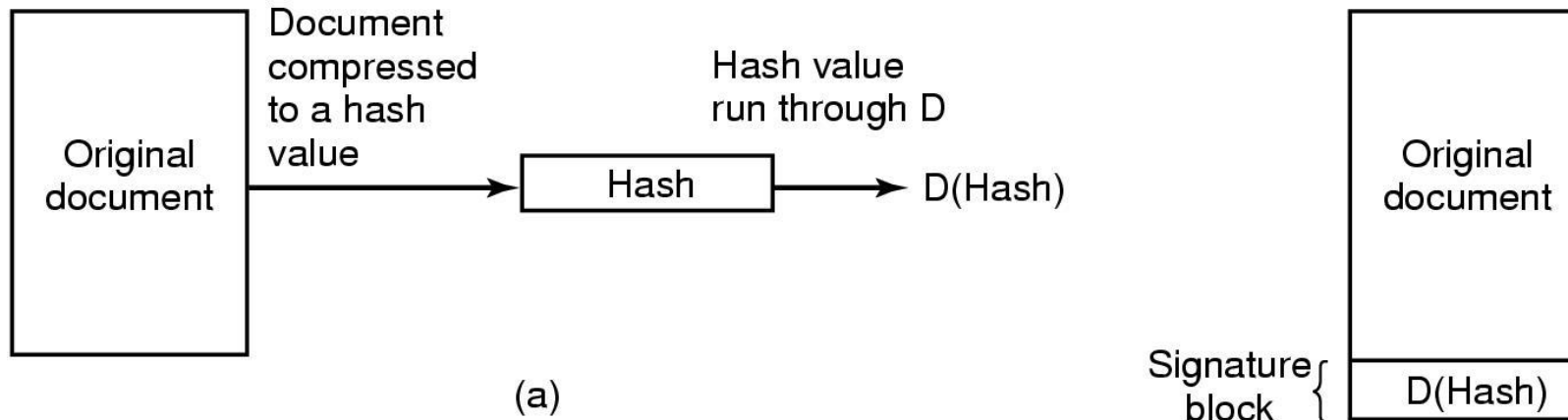
- Monoalfabetik ikame:
- Plaintext:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Ciphertext: QWERTYUIOPASDFGHJKLZXCVBNM

# Açık Anahtarlı Kriptografi

- Şifreleme, "  $314159265358979 \times 314159265358979$  ne kadar" gibi "kolay" bir işlemden yararlanır?
- Anahtar olmadan şifre çözme,  $3912571506419387090594828508241$ 'nin karekökü nedir gibi zor bir işlem yapmanızı gerektirir.

# Dijital İmzalar

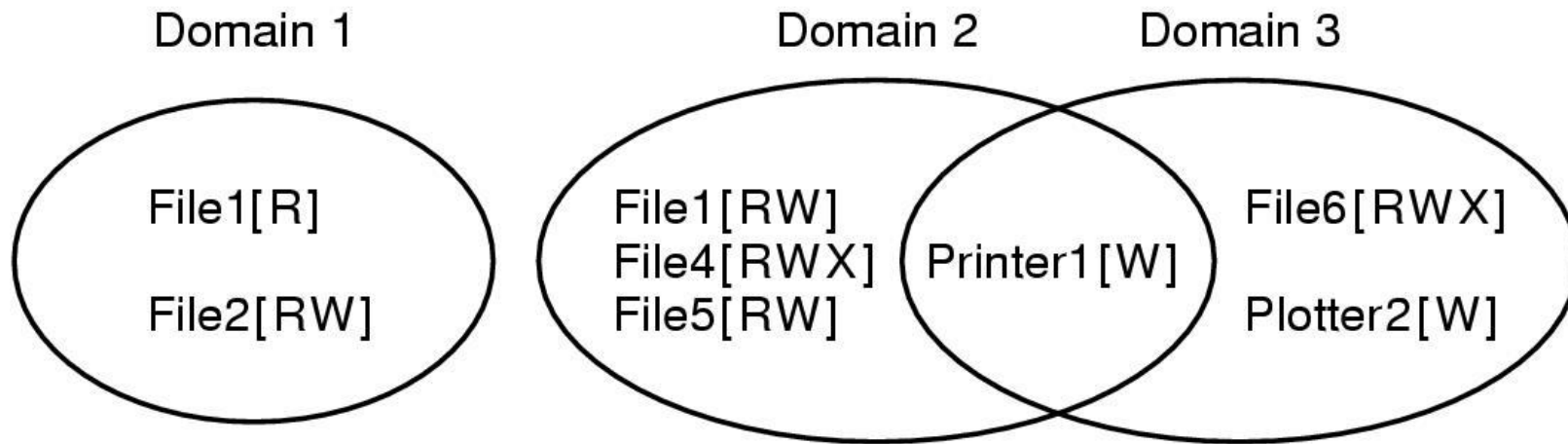
- (a) Bir imza bloğunun hesaplanması. (b) Alıcıya gelen şey.





# Koruma Etki Alanları

- Üç koruma alanı.



# Koruma Etki Alanları

- Bir koruma matrisi.

		Object							
		File1	File2	File3	File4	File5	File6	Printer1	Plotter2
Domain	1	Read	Read Write						
	2			Read	Read Write Execute	Read Write		Write	
	3						Read Write Execute	Write	Write

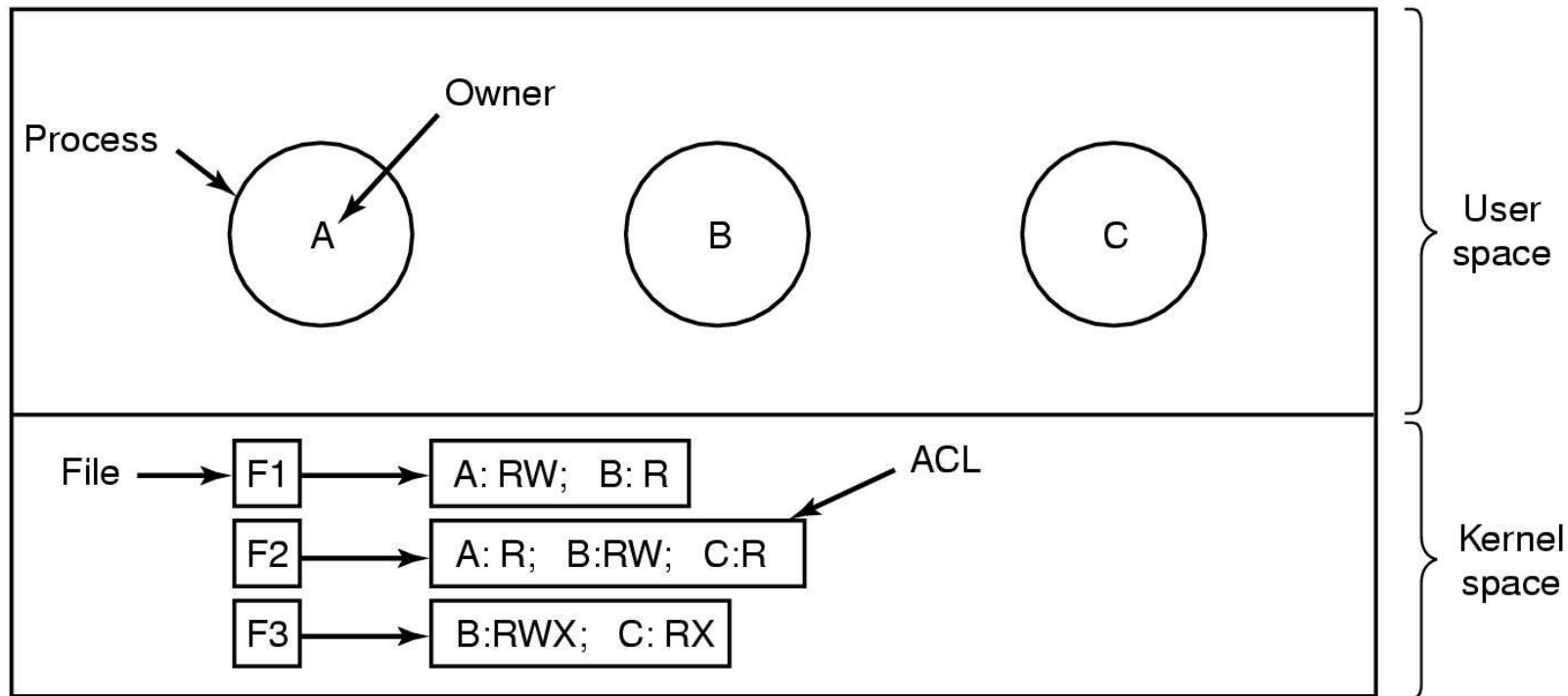
# Koruma Etki Alanları

- Etki alanlarını nesne olarak içeren bir koruma matrisi.

		Object										
		File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
Domain	1	Read	Read Write								Enter	
	2			Read	Read Write Execute	Read Write		Write				
	3						Read Write Execute	Write	Write			

# Erişim Kontrol Listeleri

- Dosya erişimini yönetmek için erişim kontrol listelerinin kullanımı.



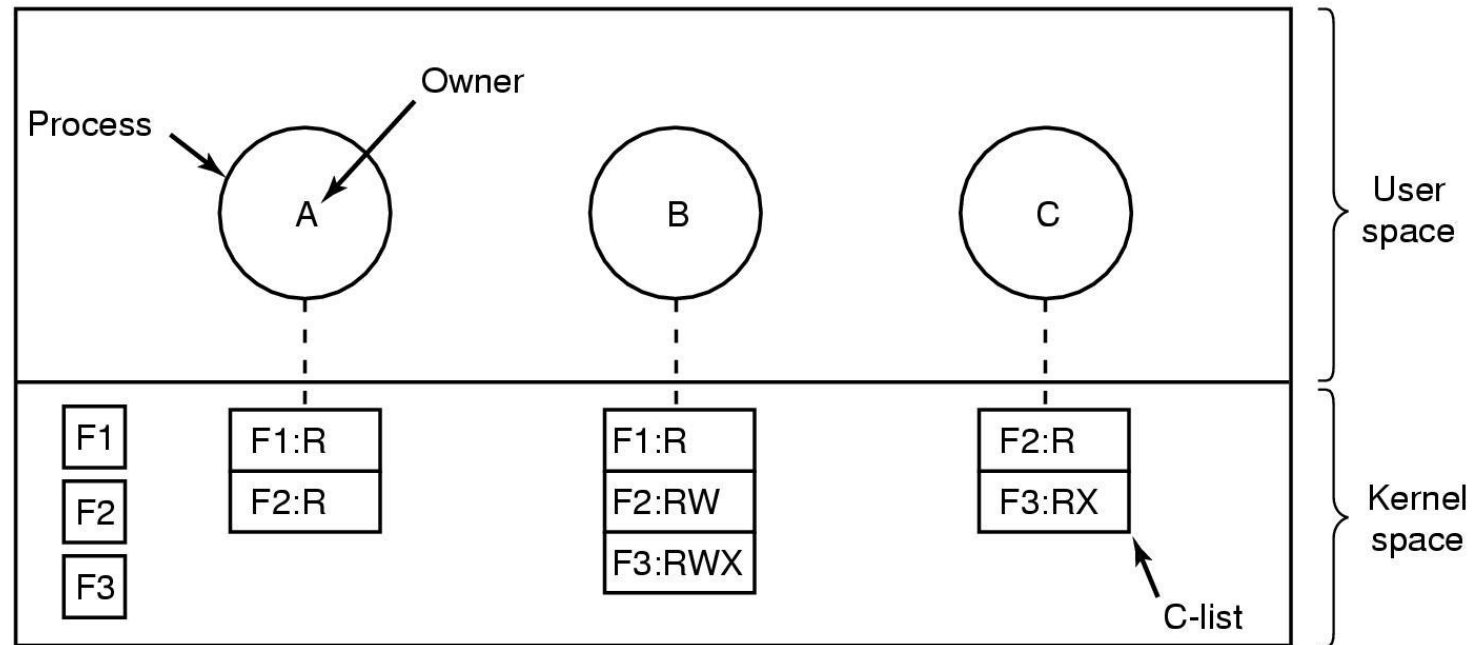
# Erişim Kontrol Listeleri

- İki erişim kontrol listesi.

File	Access control list
Password	tana, sysadm: RW
Pigeon_data	bill, pigfan: RW; tana, pigfan: RW; ...

# Yetenekler

- her sürecin bir yetenek listesi vardır.



# Yetenekler

- Kriptografik olarak korunan bir yetenek.

Server	Object	Rights	f(Objects,Rights,Check)
--------	--------	--------	-------------------------

# Yetenekler

- Genel haklara örnekler:
- Kopyalama yeteneği: aynı nesne için yeni bir yetenek yaratır.
- Nesneyi kopyala: yeni bir yeteneğe sahip yinelenen (duplicate) bir nesne oluşturur.
- Kaldırma yeteneği: yetenek listesinden bir girdiyi siler; nesne etkilenmez.
- Nesneyi yok et: bir nesneyi ve bir yeteneği kalıcı olarak kaldırır.

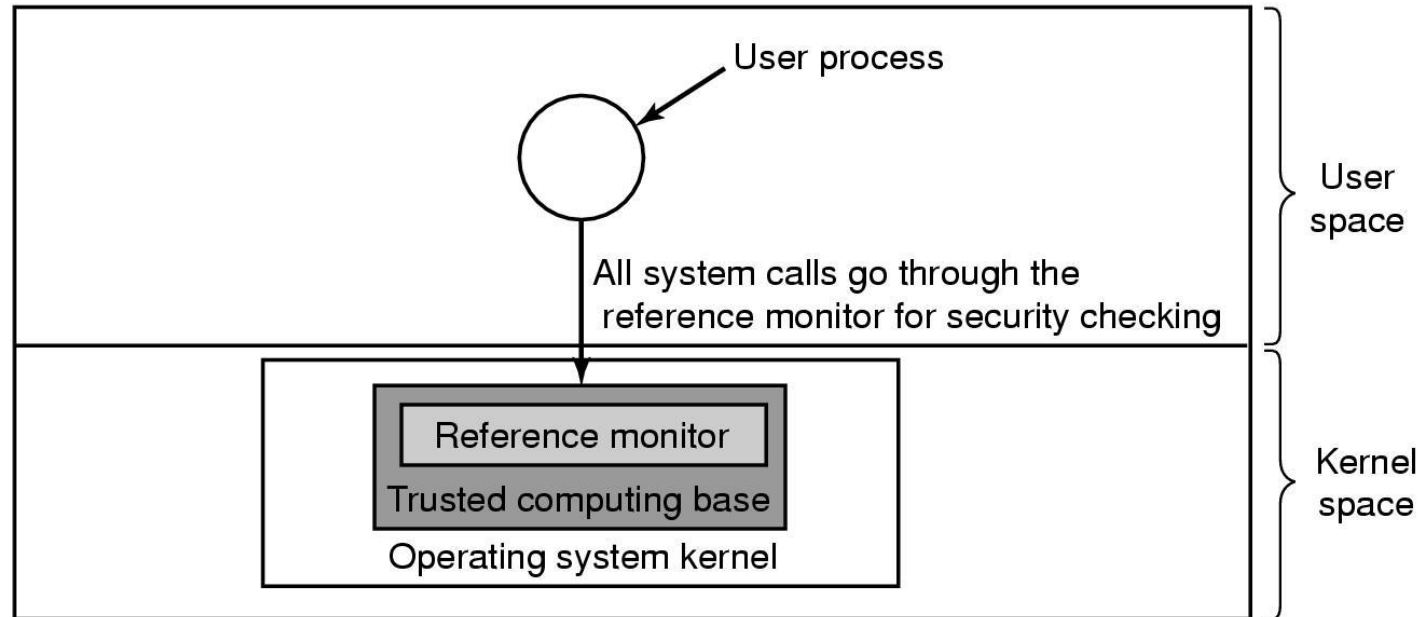


# Güvenilir Sistemler

- Virüs, solucan vb. raporları dikkate alın.
- İki saf (ama mantıklı) soru:
  - Güvenli bir bilgisayar sistemi kurmak mümkün mü?
  - Varsa neden yapılmıyor?

# Güvenilir Bilgi İşlem Tabanı

- Bir referans gözleyici.



# Güvenli Sistemlerin Biçimsel Modelleri

- (a) Yetkili bir durum. (b) Yetkisiz bir durum.

	Objects		
	Compiler	Mailbox 7	Secret
Eric	Read Execute		
Henry	Read Execute	Read Write	
Robert	Read Execute		Read Write

(a)

	Objects		
	Compiler	Mailbox 7	Secret
Eric	Read Execute		
Henry	Read Execute	Read Write	
Robert	Read Execute	Read	Read Write

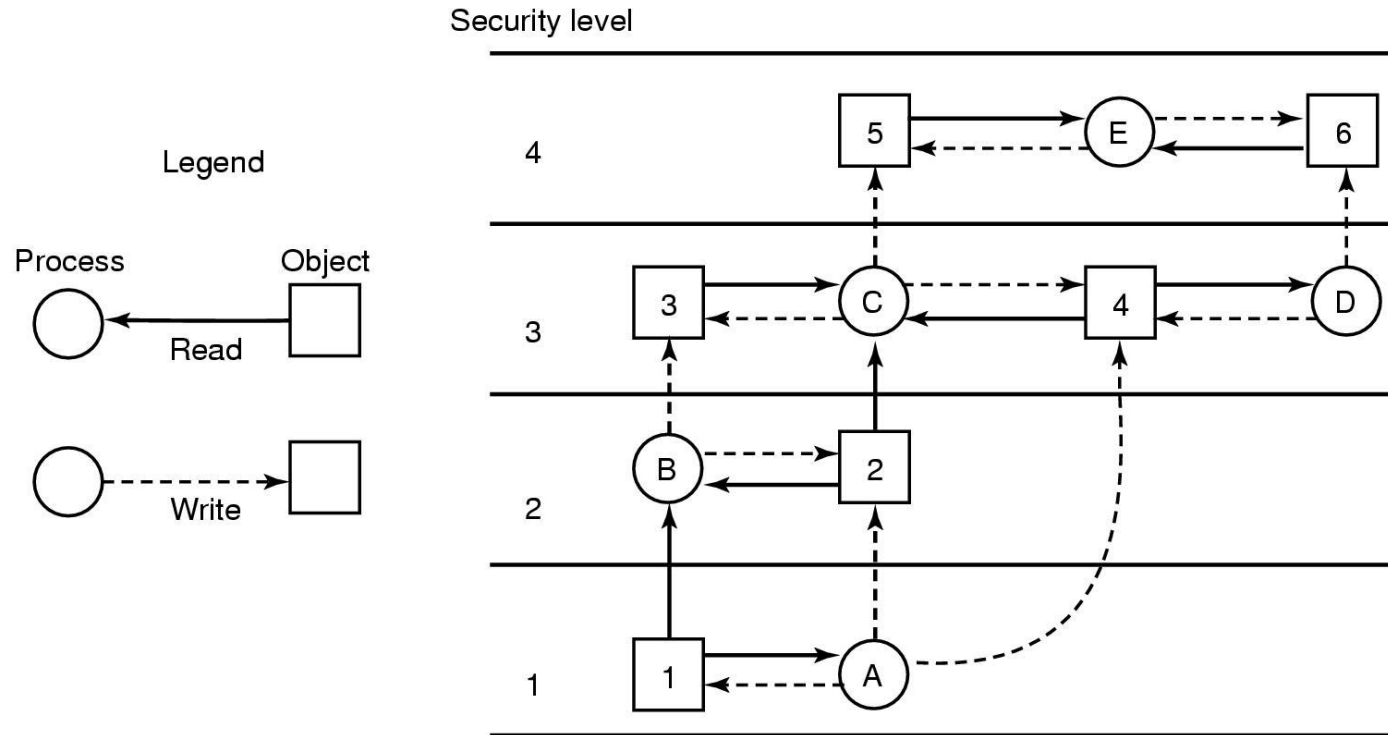
(b)

# Bell-La Padula Modeli

- Bell-La Padula modeli için kurallar:
- Basit güvenlik özelliği:  $k$  güvenlik düzeyinde çalışan bir işlem, yalnızca kendi düzeyindeki veya altındaki nesneleri okuyabilir.
- \* özelliği:  $k$  güvenlik düzeyinde çalışan bir işlem, yalnızca kendi düzeyinde veya daha yüksek olan nesneleri yazabilir.

# Bell-La Padula Modeli

- Bell-La Padula çok düzeyli güvenlik modeli.

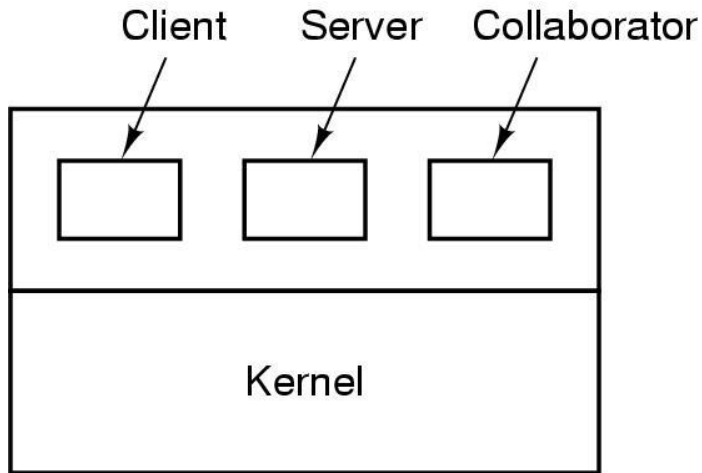


# Biba Modeli

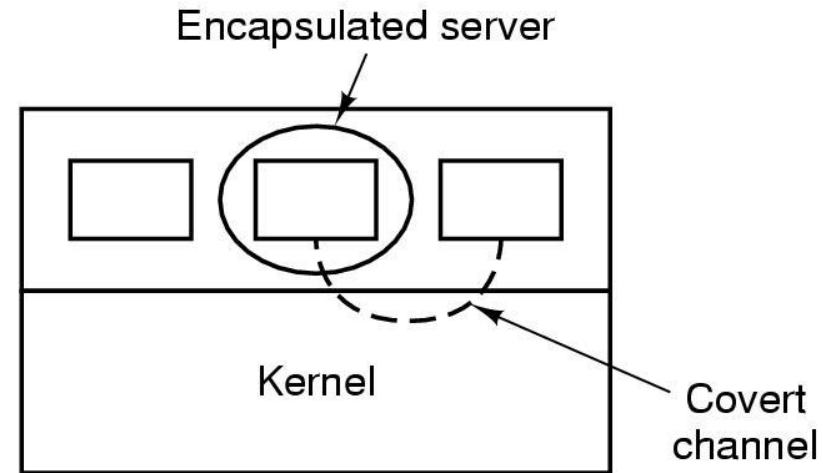
- Biba modeli için kurallar:
- Basit bütünlük ilkesi:  $k$  güvenlik düzeyinde çalışan bir işlem, yalnızca kendi düzeyindeki veya altındaki nesneleri yazabilir.
- Bütünlük \* özelliği:  $k$  güvenlik seviyesinde çalışan bir işlem, yalnızca kendi seviyesindeki veya daha yüksek seviyedeki nesneleri okuyabilir.

# Gizli Kanallar

- (a) İstemci, sunucu ve ortak çalışan süreçleri. (b) Kapsüllenmiş sunucu, gizli kanallar aracılığıyla ortak çalışana yine de sızabilir.



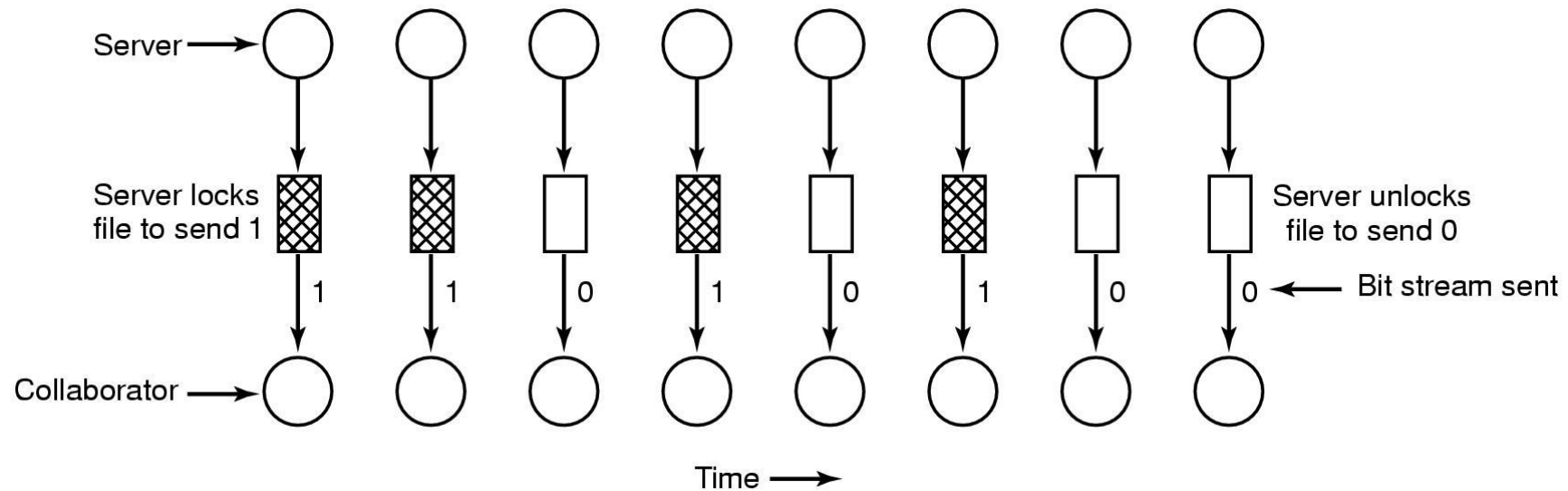
(a)



(b)

# Gizli Kanallar

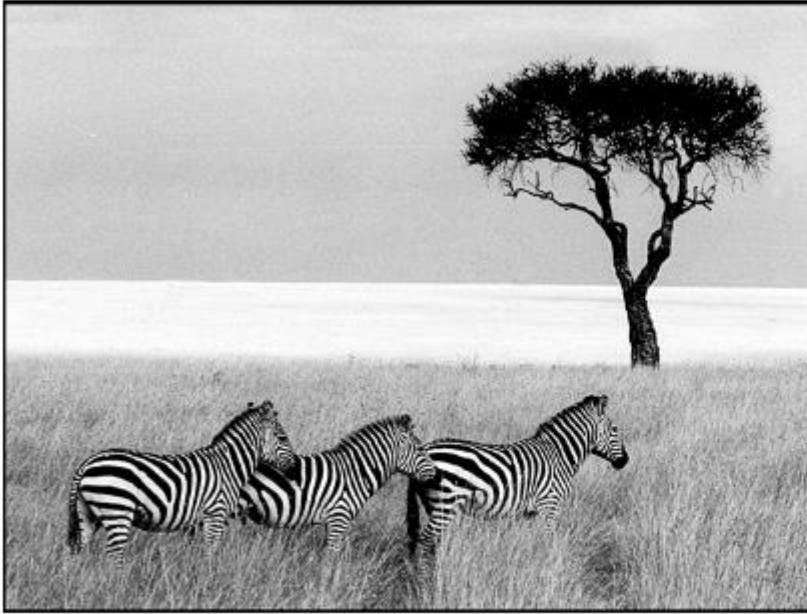
- Dosya kilitleme kullanan gizli bir kanal.



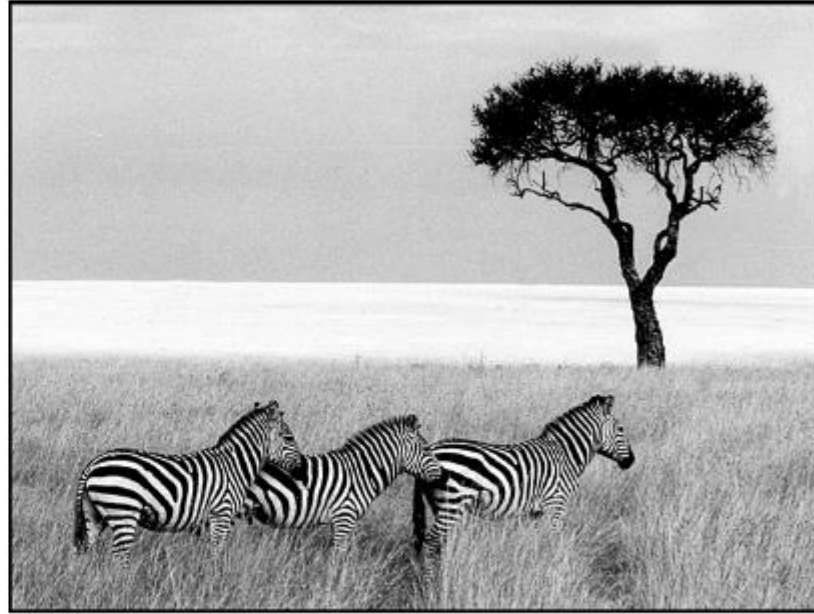


# Gizli Kanallar

- (a) Üç zebra ve bir ağaç. (b) Üç zebra, bir ağaç ve William Shakespeare'in beş oyununun tam metni.



(a)



(b)

# Kimlik Doğrulama

- Kullanıcıların kimliğini doğrulamanın genel ilkeleri:
- Kullanıcının bildiği bir şey. (parola)
- Kullanıcının sahip olduğu bir şey. (parmak izi, göz retina)
- Kullanıcının olduğu bir şey. (kimlik kartı)

# Parola Kullanarak Kimlik Doğrulama

- (a) Başarılı bir oturum açma. (b) Ad girildikten sonra oturum açma reddedildi. (c) Ad ve parola yazıldıktan sonra oturum açma reddedildi.

LOGIN: mitch  
PASSWORD: FooBar!-7  
SUCCESSFUL LOGIN

(a)

LOGIN: carol  
INVALID LOGIN NAME  
LOGIN:

(b)

LOGIN: carol  
PASSWORD: Idunno  
INVALID LOGIN  
LOGIN:

(c)

# Bilgisayar Korsanları Nasıl İçeri Girer?

- .  
LBL> telnet elxsi  
ELXSI AT LBL  
LOGIN: root  
PASSWORD: root  
INCORRECT PASSWORD, TRY AGAIN  
LOGIN: guest  
PASSWORD: guest  
INCORRECT PASSWORD, TRY AGAIN  
LOGIN: uucp  
PASSWORD: uucp  
WELCOME TO THE ELXSI COMPUTER AT LBL

# UNIX Parola Güvenliği

- Şifreli parolaların ön hesaplamasını (precomputation) önlemek (defeat) için tuz (salt) kullanımı.

Bobbie, 4238, e(Dog, 4238)
Tony, 2918, e(6%%TaeFF, 2918)
Laura, 6902, e(Shakespeare, 6902)
Mark, 1694, e(XaB#Bwcz, 1694)
Deborah, 1092, e(LordByron,1092)

# Sorgu-Yanıt Kimlik Doğrulaması

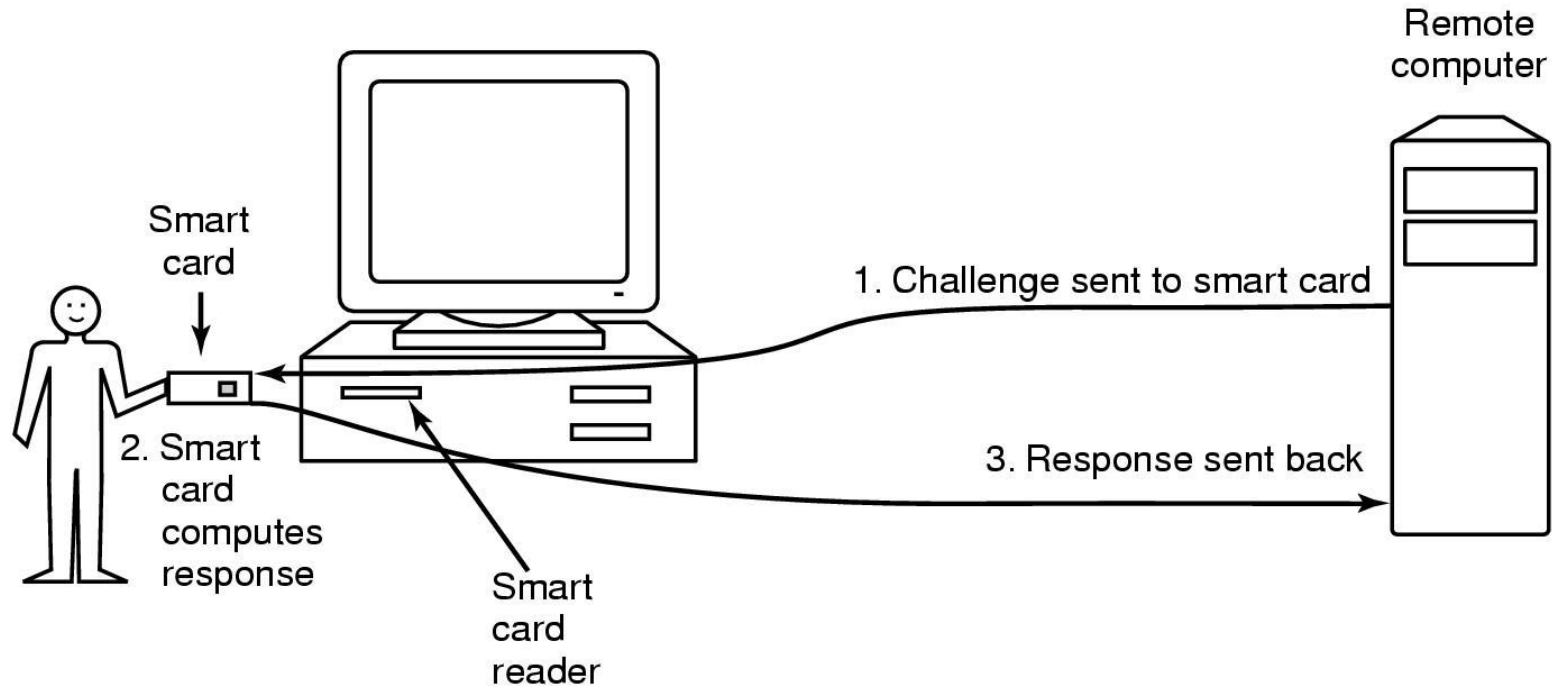
- Sorular, kullanıcının yazmasını gerektirmeyecek şekilde seçilmelidir.

Örnekler:

- Marjolein'in kız kardeşi kimdir?
- İlkokulunuz hangi sokaktaydı?
- Bayan Woroboff ne öğretti?

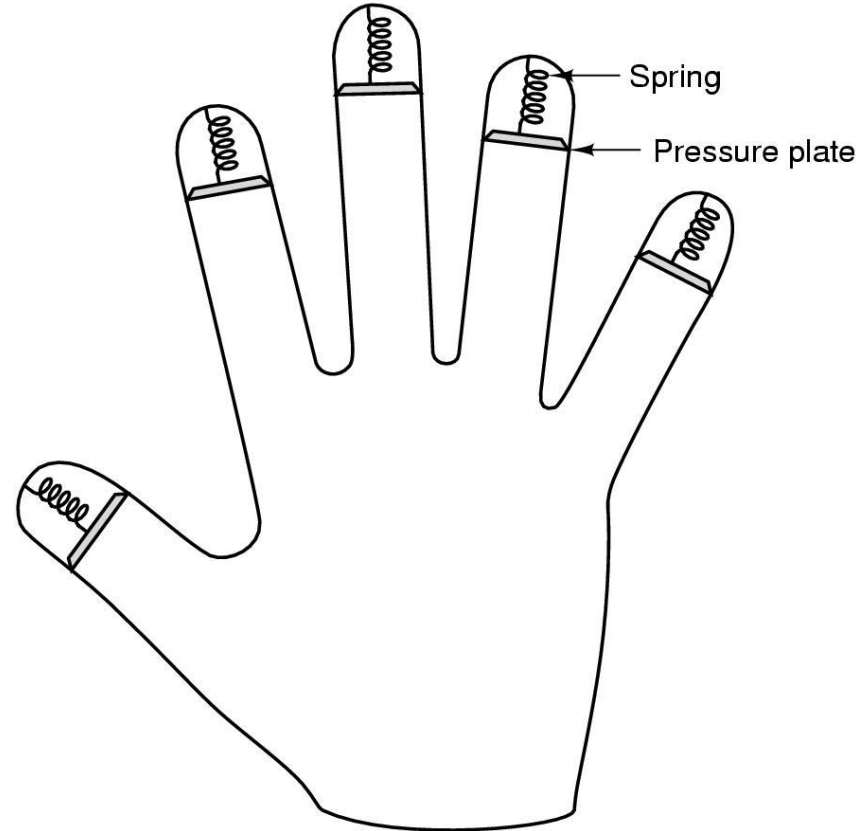
# Fiziksel Nesne Kullanarak Kimlik Doğrulama

- Kimlik doğrulama için akıllı kart kullanımı.



# Biyometri Kullanarak Kimlik Doğrulama

- Parmak uzunluğunu ölçmek için bir cihaz.





# Tuzak Kapısı

- (a) Normal kod. (b) Tuzak kapılı kod.

```
while (TRUE) {  
    printf("login: ");  
    get_string(name);  
    disable_echoing( );  
    printf("password: ");  
    get_string(password);  
    enable_echoing( );  
    v = check_validity(name, password);  
    if (v) break;  
}  
execute_shell(name);
```

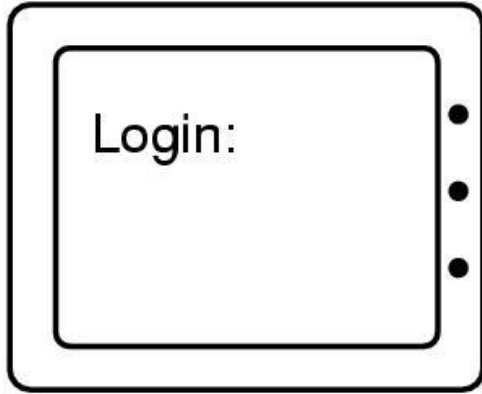
(a)

```
while (TRUE) {  
    printf("login: ");  
    get_string(name);  
    disable_echoing( );  
    printf("password: ");  
    get_string(password);  
    enable_echoing( );  
    v = check_validity(name, password);  
    if (v || strcmp(name, "zzzzz") == 0) break;  
}  
execute_shell(name);
```

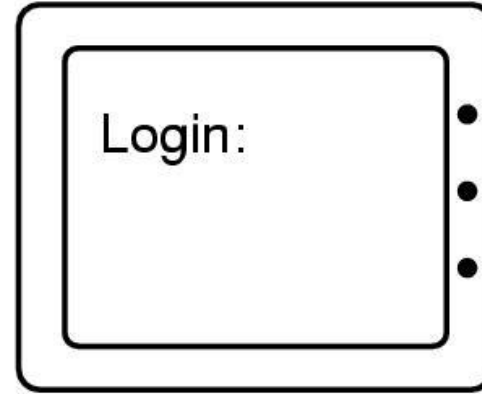
(b)

# Giriş Sayfası Kandırma Saldırısı

- (a) Doğru oturum açma ekranı. (b) Sahte oturum açma ekranı.



(a)



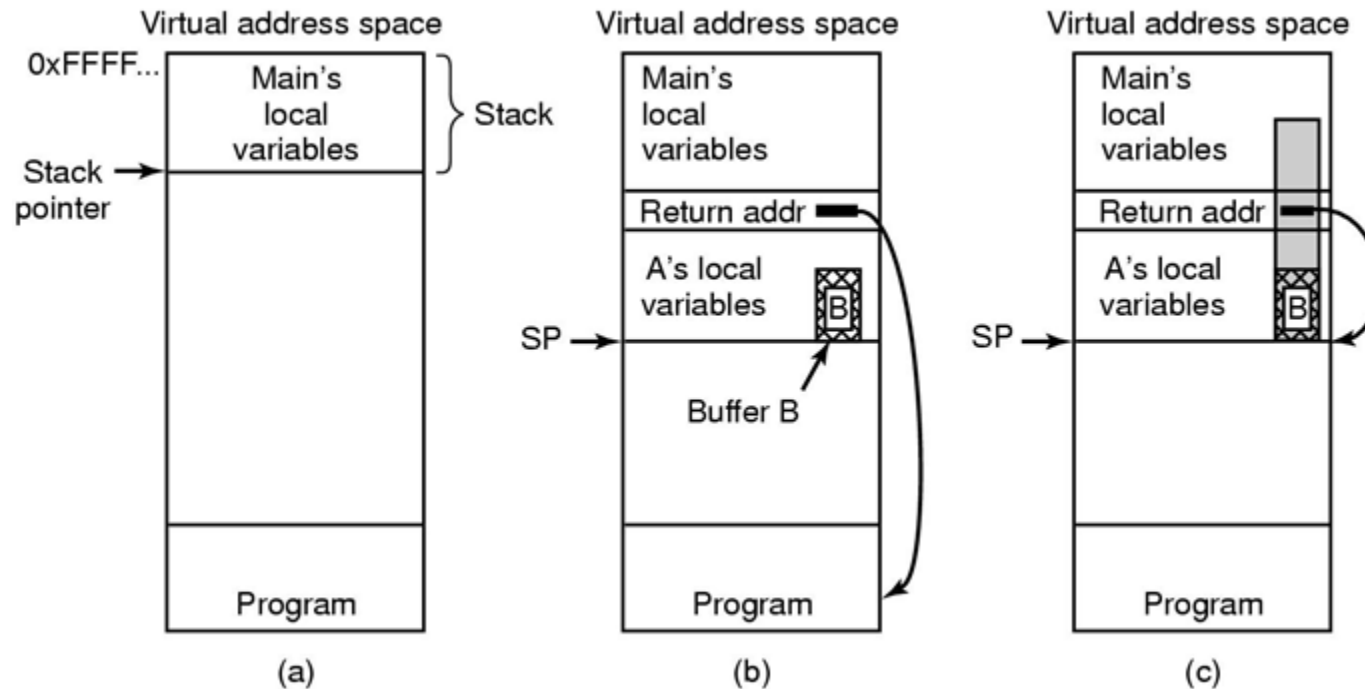
(b)

# Kod Hatalarından Yararlanma

- Bir hatadan yararlanmak için örnek adımlar:
- Telnet bağlantılarını kabul eden makineleri bulmak için bağlantı noktası taramasını (scan port) çalıştır.
- Kullanıcı adı, şifre kombinasyonları tahmin ederek giriş yapmayı dene.
- Girişten sonra, hatayı tetikleyen girdiyle hatalı programı çalıştır.
- Hatalı program SETUID kökü ise, bir SETUID kök kabuğu (root shell) oluştur.
- Cmds için IP port dinleyen bir zombi programı getir (fetch) ve başlat.
- Sistem yeniden başlatıldığında zombi programının başlatılmasını sağla.

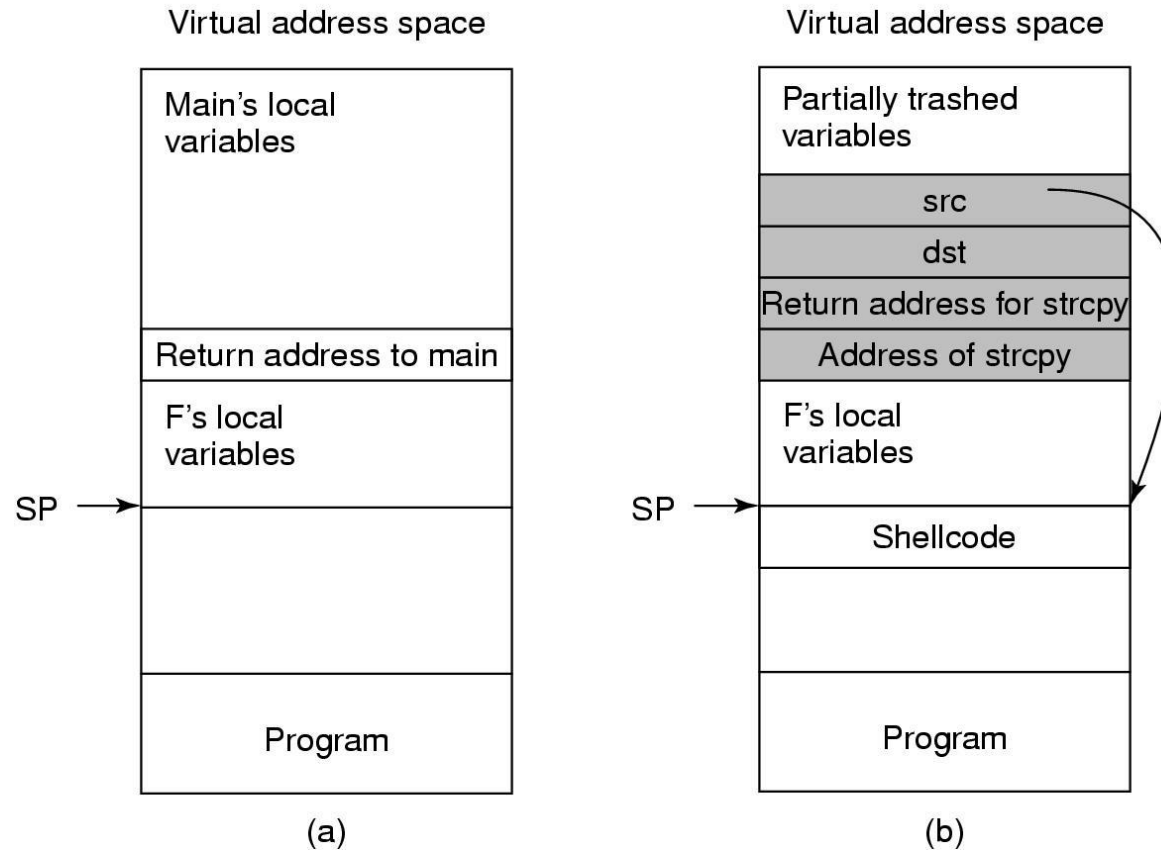
# Tampon Taşma Saldırıları

- (a) Ana programın çalıştığı durum. (b) Prosedür A çağrıldıktan sonra. (c) Gri renkle gösterilen arabellek taşması.



# libc Saldırıları

- (a) Saldırıdan önceki yığın. (b) Yığın üzerine yazıldıktan sonra.



# Kod Enjeksiyon Saldırıları

- Kod enjeksiyon saldırısına yol açabilecek kod.

```
int main(int argc, char *argv[])
{
    char src[100], dst[100], cmd[205] = "cp ";
    printf("Please enter name of source file: ");
    gets(src);
    strcat(cmd, src);
    strcat(cmd, " ");
    printf("Please enter name of destination file: ");
    gets(dst);
    strcat(cmd, dst);
    system(cmd);
}
```

/\* declare 3 strings \*/  
/\* ask for source file \*/  
/\* get input from the keyboard \*/  
/\* concatenate src after cp \*/  
/\* add a space to the end of cmd \*/  
/\* ask for output file name \*/  
/\* get input from the keyboard \*/  
/\* complete the commands string \*/  
/\* execute the cp command \*/

# Kötü Amaçlı Yazılım (malware)

- Bir tür şantaj için kullanılabilir.
- Örnek: Kurban diskindeki dosyaları şifreler, ardından şu mesajı görüntüler...

Greetings from General Encryption

To purchase a decryption key for your hard disk, please send \$100 in small unmarked bills to Box 2154, Panama City, Panama.  
Thank you. We appreciate your business.

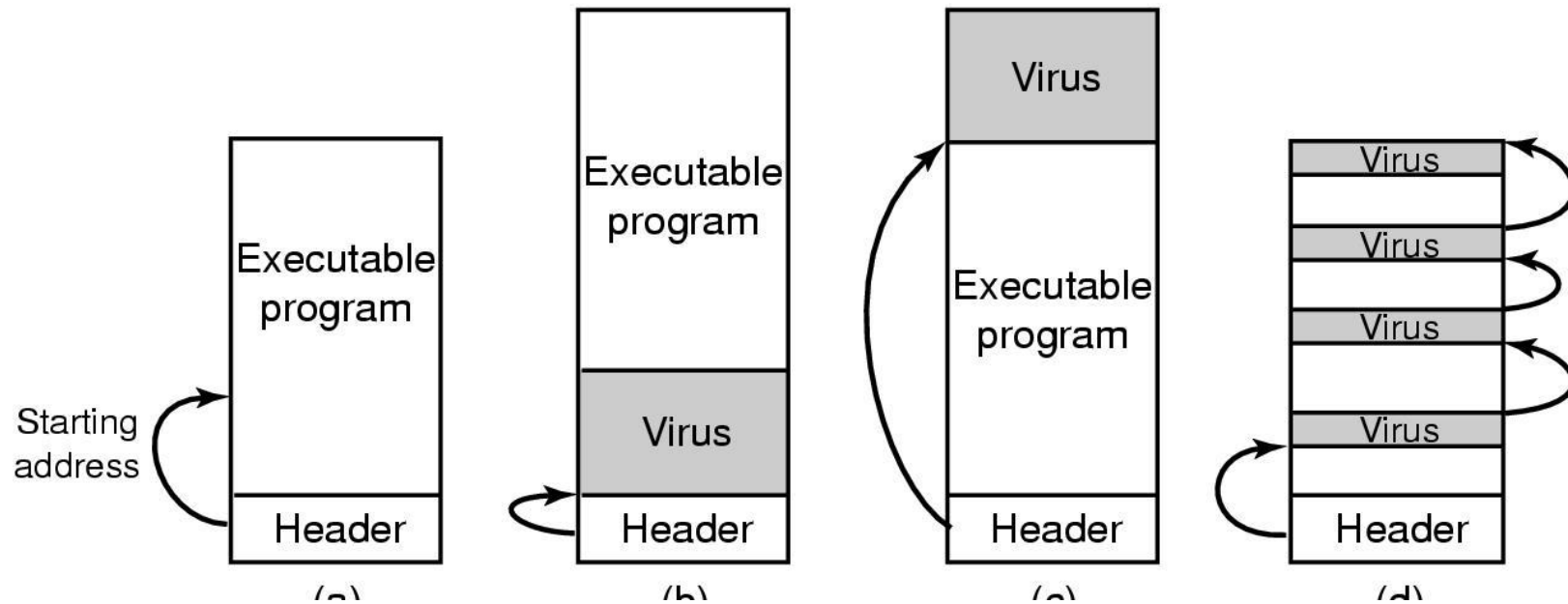
# Virüs Çeşitleri

- Eşlik eden virüs (companion)
- Yürütülebilir program virüsü (executable)
- Parazitik virüs (parasitic)
- Bellekte yerleşik virüs (memory resident)
- Önyükleme sektörü virüsü (boot sector)
- Aygıt sürücüsü virüsü (device driver)
- Makro virüs (macro)
- Kaynak kodu virüsü (source code)



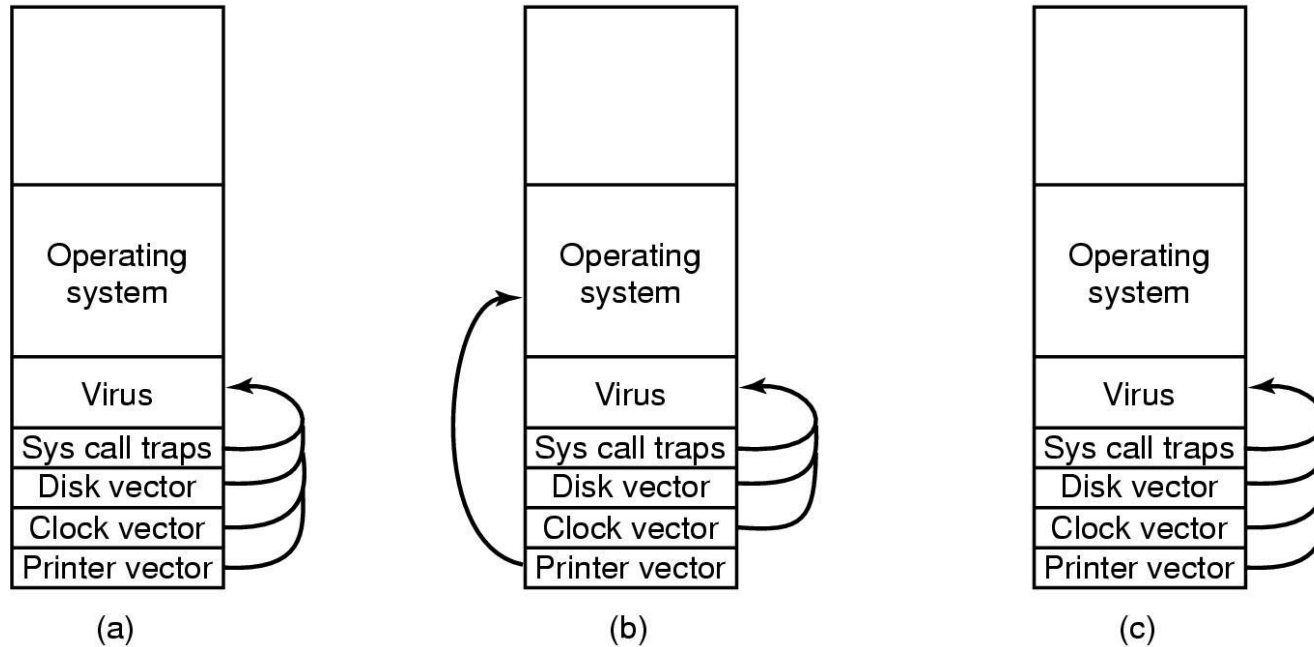
# Parazitik Virüsler

- (a) Yürütülebilir bir program. (b) Önde bir virüs var. (c) Sonda bir virüs var. (d) Program içindeki boş alana yayılmış bir virüs ile.



# Önyükleme Sektörü Virüsleri

- (a) Virüs tüm kesme ve tuzak vektörlerini yakaladıktan sonra. (b) İşletim sistemi yazıcı kesme vektörünü yeniden aldıktan sonra. (c) Virüs, yazıcı kesme vektörünü yeniden yakaladıktan sonra.



# Casus Yazılım

- Sahibinin bilgisi dışında gizlice bir PC'ye yüklenir
- Arka planda çalışır
- Gizlenir, kurban kolayca bulamaz
- Kullanıcı hakkında veri toplar
- Toplanan bilgileri uzakta bir bilgisayara iletir
- Onu ortadan kaldırmak için kararlı girişimlerde hayatta kalmaya çalışır

# Casus Yazılım Nasıl Yayılır

- Olası yollar:
- Kötü amaçlı yazılımla aynı, Truva atı
- İndirme, virüslü bir web sitesini ziyaret etme
- Web sayfaları bir .exe dosyası çalıştırmayı dener
- Şüphelenilmeyen kullanıcı virüslü bir araç çubuğu yükler
- Kötü amaçlı activeX denetimleri yüklenir

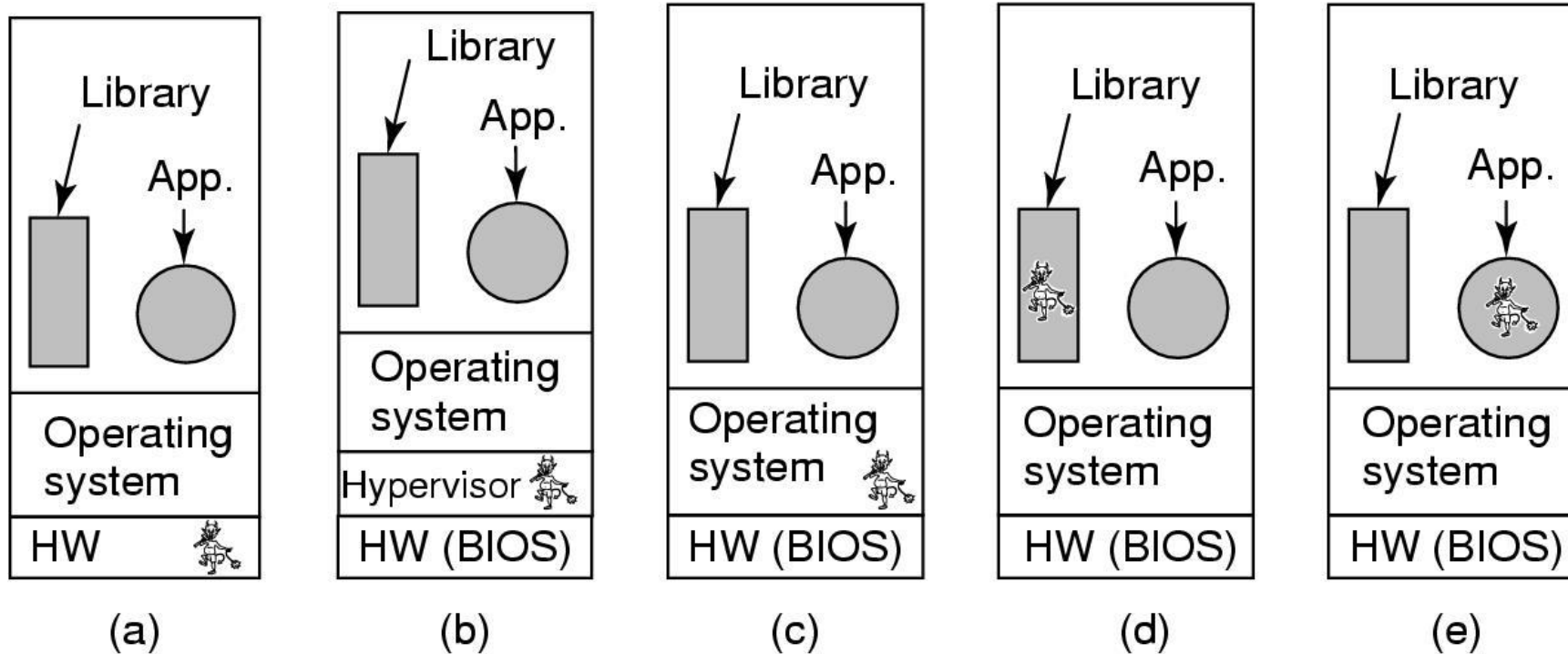
# Casus Yazılım Gerçekleştirdiği Eylemler

- Tarayıcının ana sayfasını değiştirme.
- Tarayıcının favori (yer imi eklenmiş) sayfalar listesini değiştirme.
- Tarayıcıya yeni araç çubukları ekleme.
- Varsayılan medya yürütücüsünü değiştirme.
- Varsayılan arama motorunu değiştirme.
- Windows masaüstüne yeni simgeler ekleme.
- Web sayfalarındaki banner reklamları, casus yazılımın seçtikleriyle değiştirme.
- Reklamları standart Windows iletişim kutularına yerleştirme
- Sürekli ve durdurulamaz bir pop-up reklam akışı oluşturma.

# Rootkit Türleri

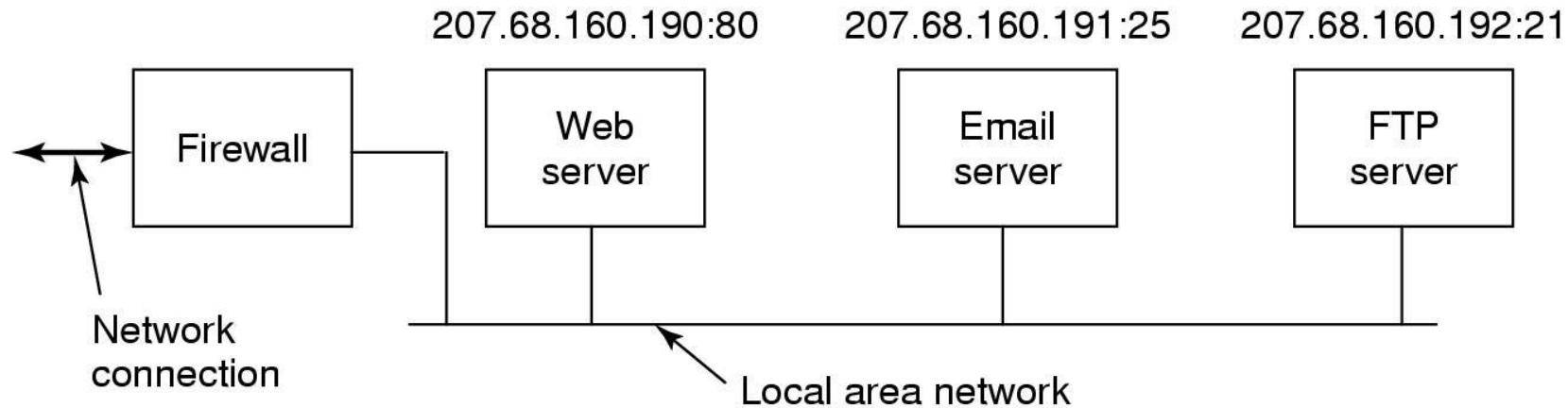
- Bellenim rootkit'leri (firmware)
- Hipervizör rootkit'leri
- Çekirdek rootkit'leri (kernel)
- Kütüphane rootkit'leri
- Uygulama rootkit'leri

# Bir Rootkit'in Saklanabileceği Beş Yer



# Güvenlik Duvarı

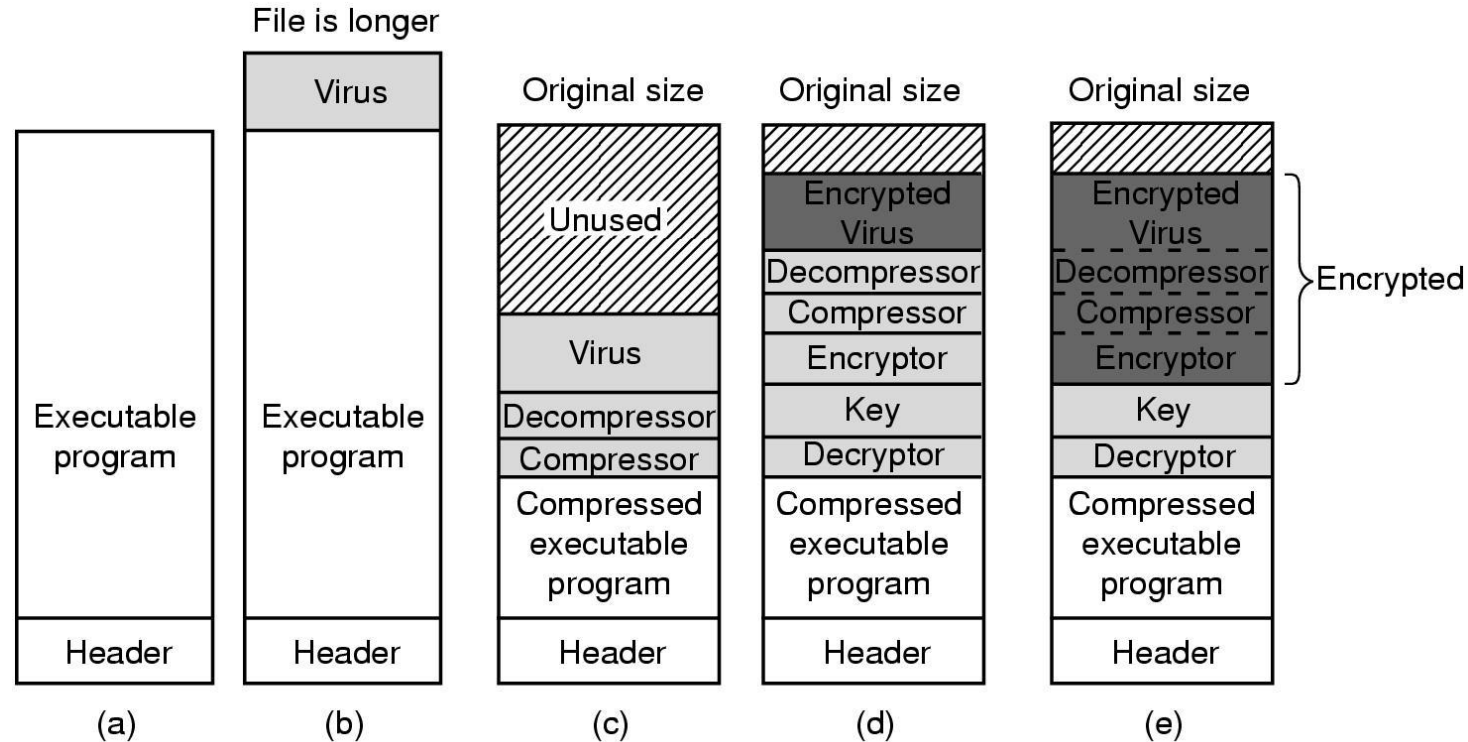
- Üç bilgisayarlı bir LAN'ı koruyan donanım güvenlik duvarı





# Virüs Tarayıcıları

- (a) Bir program. (b) Virüslü program. (c) Sıkıştırılmış virüslü program. (d) Şifrelenmiş bir virüs. (e) Şifrelenmiş sıkıştırma ile sıkıştırılmış virüs.



# Virüs Tarayıcıları

- Polimorfik virüs örnekleri.

```
MOV A,R1
ADD B,R1
ADD C,R1
SUB #4,R1
MOV R1,X
```

(a)

```
MOV A,R1
NOP
ADD B,R1
NOP
ADD C,R1
NOP
SUB #4,R1
NOP
MOV R1,X
```

(b)

```
MOV A,R1
ADD #0,R1
ADD B,R1
OR R1,R1
ADD C,R1
SHL #0,R1
SUB #4,R1
JMP .+1
MOV R1,X
```

(c)

```
MOV A,R1
OR R1,R1
ADD B,R1
MOV R1,R5
ADD C,R1
SHL R1,0
SUB #4,R1
ADD R5,R5
MOV R1,X
MOV R5,Y
```

(d)

```
MOV A,R1
TST R1
ADD C,R1
MOV R1,R5
ADD B,R1
CMP R2,R5
SUB #4,R1
JMP .+1
MOV R1,X
MOV R5,Y
```

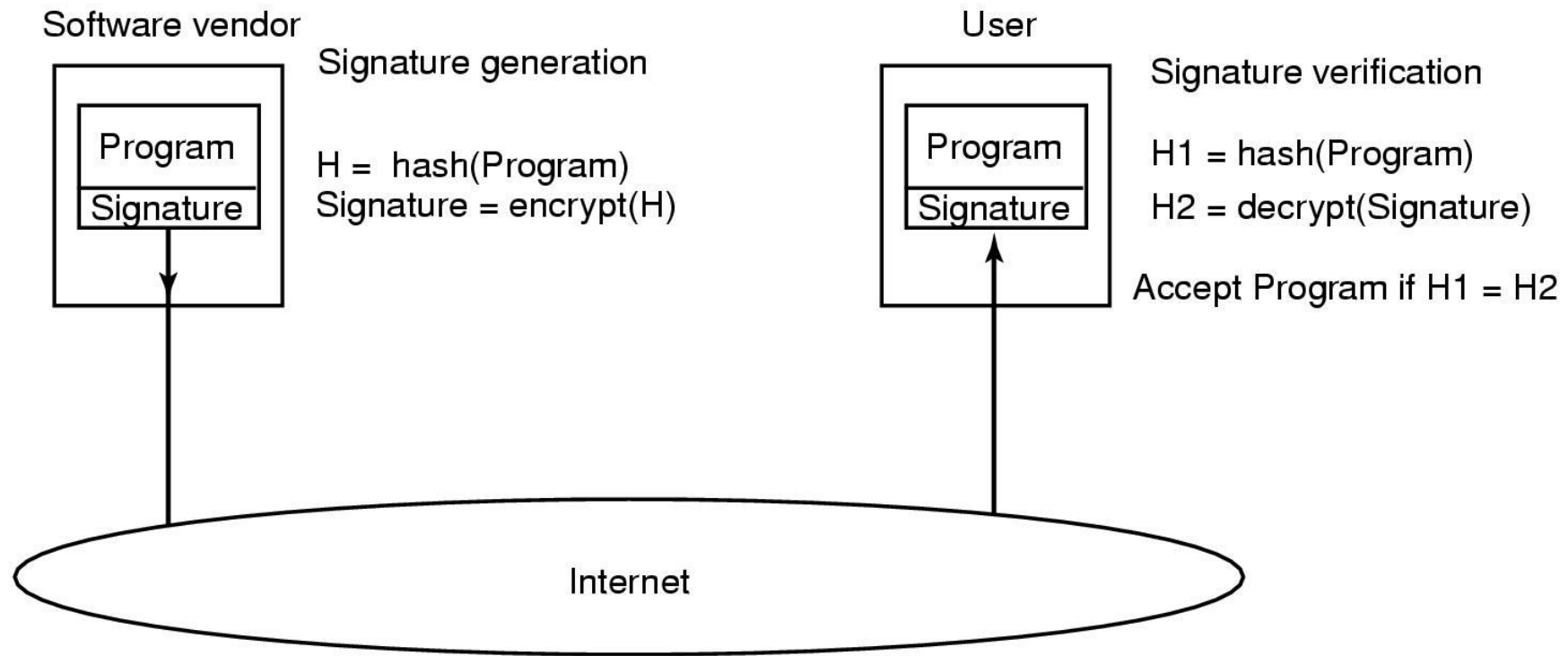
(e)

# Antivirüs ve Anti-Antivirüs Teknikleri

- Virüs tarayıcıları
- Bütünlük denetleyicileri (integrity)
- Davranışsal denetleyiciler
- Virüsten kaçınma

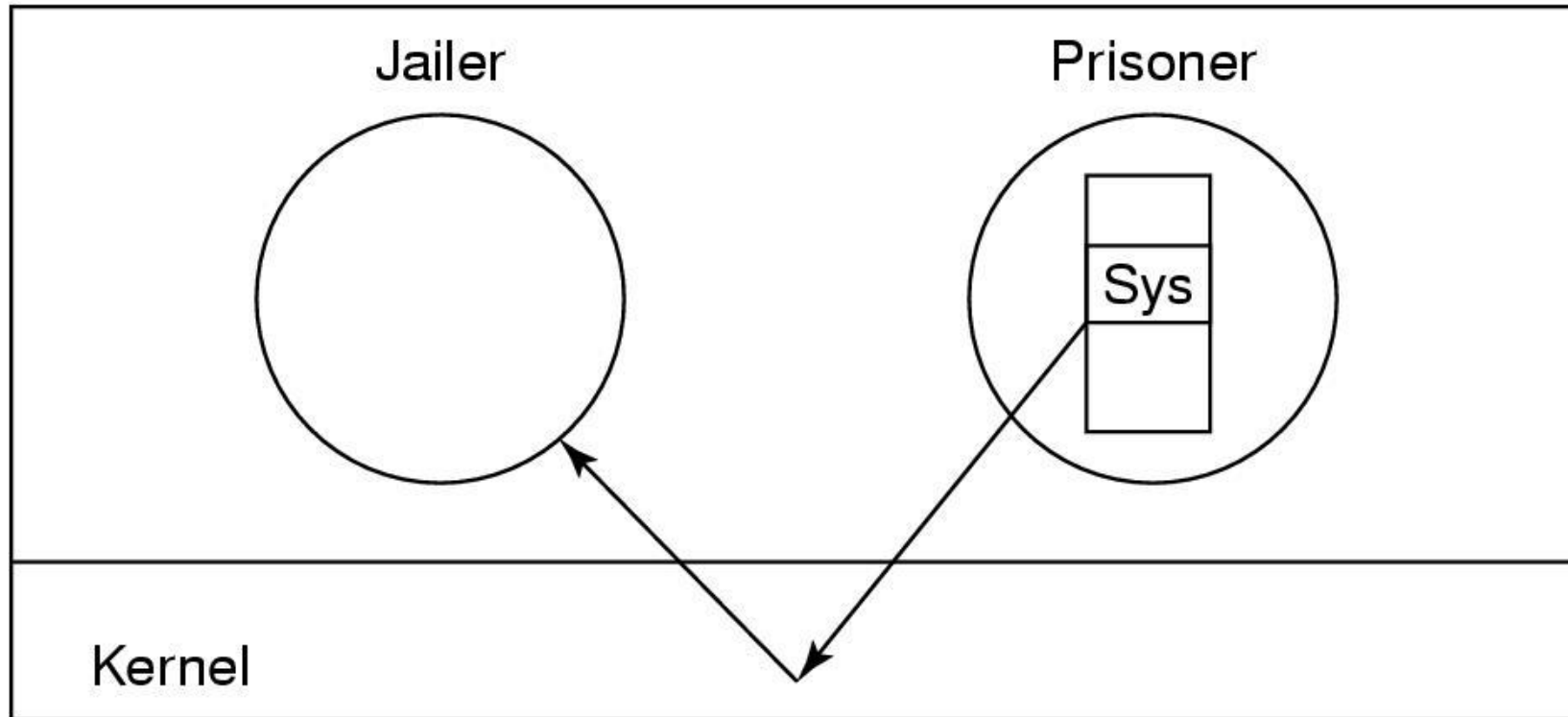
# Kod imzalama Nasıl Çalışır?

• .



# Hapse Atmak (jailing)

- .



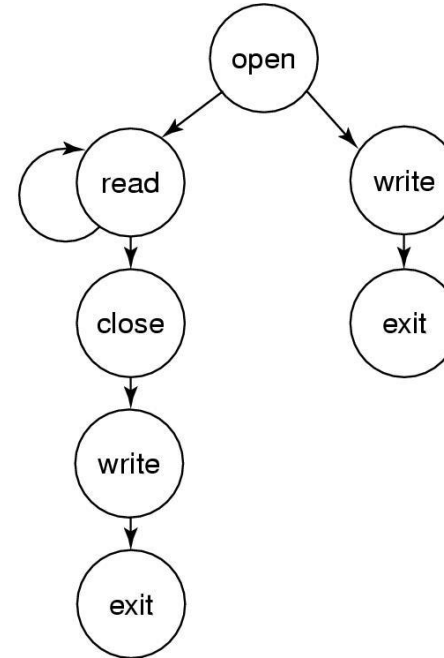
# Model Tabanlı Saldırı Tespiti

- (a) Bir program. (b) (a) için sistem çağrı çizge.

```
int main(int argc *char argv[])
{
    int fd, n = 0;
    char buf[1];

    fd = open("data", 0);
    if (fd < 0) {
        printf("Bad data file\n");
        exit(1);
    } else {
        while (1) {
            read(fd, buf, 1);
            if (buf[0] == 0) {
                close(fd);
                printf("n = %d\n", n);
                exit(0);
            }
            n = n + 1;
        }
    }
}
```

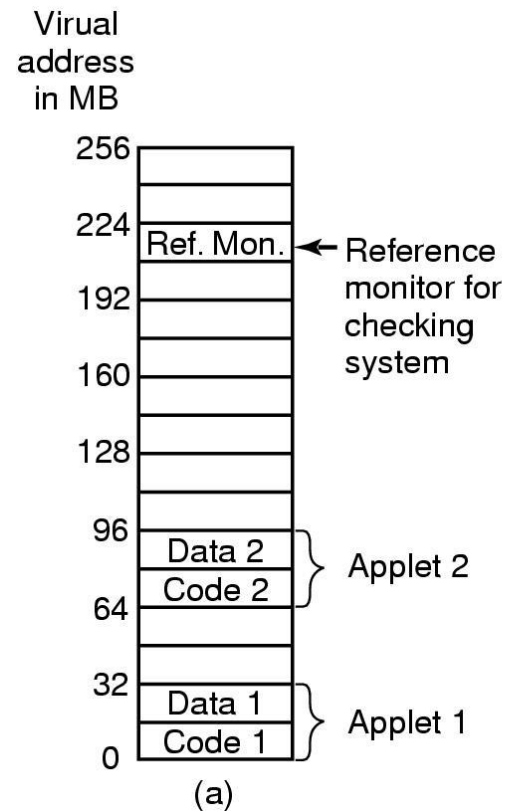
(a)



(b)

# Korumalı Alan (sandboxing)

- (a) 16 MB sanal alanlara bölünmüş bellek. (b) Bir talimatın geçerliliğini kontrol etmenin yolu.

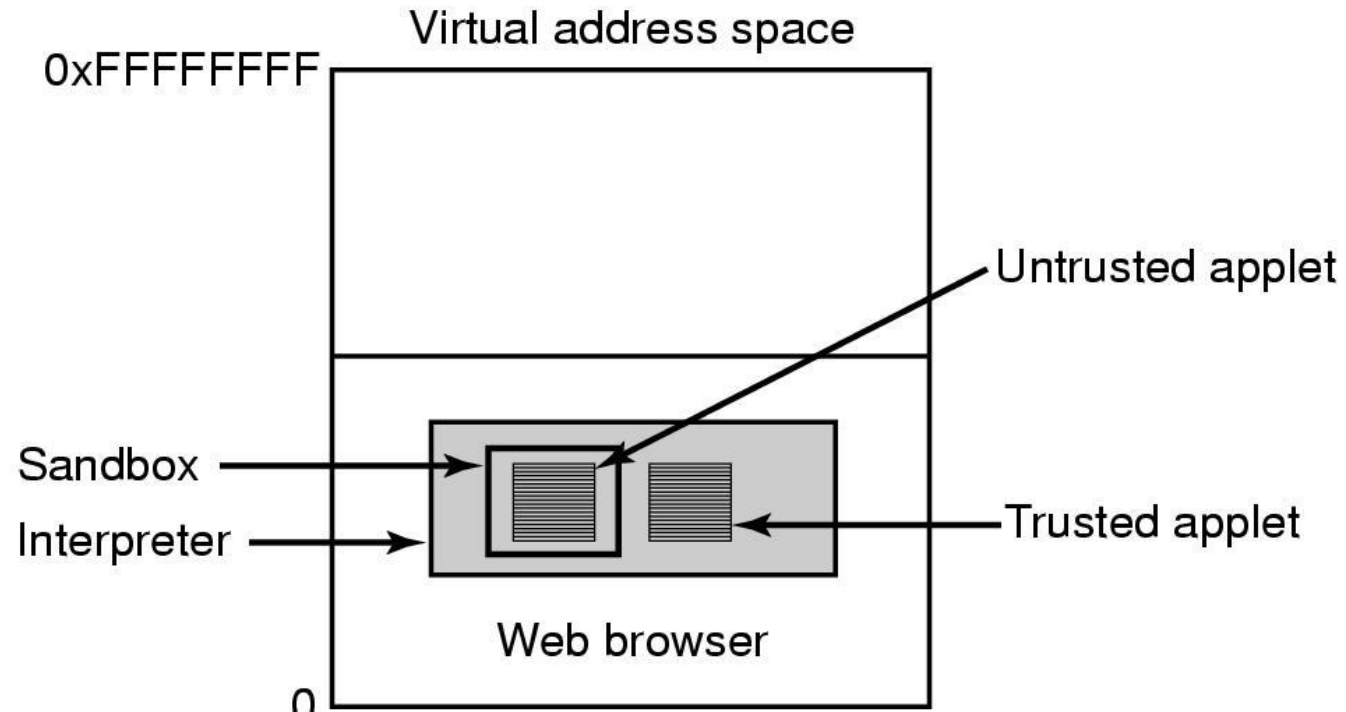


```
MOV R1, S1  
SHR #24, S1  
CMP S1, S2  
TRAPNE  
JMP (R1)
```

(b)

# Yorumlamalı Dil (interpreter)

- Applet'ler bir Web tarayıcısı tarafından yorumlanabilir.





# Java Güvenlik

- JVM bayt kodu doğrulayıcı, uygulamanın belirli kurallara uyup uymadığını kontrol eder:
- Uygulama, işaretçiler oluşturmaya çalışıyor mu?
- Özel sınıf üyeleri üzerindeki erişim kısıtlamalarını ihlal ediyor mu?
- Bir tür değişkeni başka bir tür olarak kullanmaya çalışıyor mu?
- Yığın taşmaları oluşturuyor mu? (stack overflows, underflows)
- Bir türdeki değişkenleri yasa dışı bir şekilde diğerine dönüştürüyor mu?

SON