# Bölüm 9: Güvenlik

**İşletim Sistemleri**

# Güvenlik Hedefleri

- Gizlilik: Hassas verilere yetkisiz erişimi önleyin
- Bütünlük: Verilerde veya sistemlerde yetkisiz değişikliklere karşı koruma sağlayın
- Erişilebilirlik: Yetkili kullanıcıların gerektiğinde verilere ve sistemlere erişebilmesini sağlayın

# Tehditler

- Kötü Amaçlı Yazılım: Virüs, Truva atı, solucan, casus yazılım, fidye yazılımı vb.
- Yetkisiz Erişim: Uygun yetkilendirme olmadan sistemlere veya verilere erişme girişimleri
- Hizmet Reddi (DoS): Sistemlerin veya verilerin kullanılabilirliğini kesintiye uğratmak
- Arabellek Taşmaları: Bellekte bir arabellek taşması ve kötü amaçlı kod yürütme
- Yarış Koşulları: Beklenmeyen sonuçlara yol açabilecek kodun eşzamanlı yürütülmesi

# Savunma Mekanizmaları

- Erişim Kontrolü: Kimin hangi verilere veya sistemlere erişebileceğini tanımlayın
- Güvenlik duvarları: Yetkisiz erişimi önlemek için ağ trafiğini filtreleyin
- Bellek Koruması: Arabellek taşmalarını ve diğer bellek tabanlı saldırıları önleyin
- Sandboxing: Birbirini engellemelerini önlemek için süreçleri izole edin
- Şifreleme: Aktarım halindeki veya atıl durumdaki hassas verileri koruyun

# En İyi Uygulamalar (best practices)

- Güvenlik yamalarıyla yazılımı güncel tutun
- Güçlü kimlik doğrulama ve erişim kontrol mekanizmaları kullanın
- Güvenlik izleme ve günlük (log) kaydı tutma
- Düzenli güvenlik değerlendirmeleri ve sızma testleri gerçekleştirin
- Kullanıcıları en iyi güvenlik uygulamaları ve güvenlik tehditlerine ilişkin farkındalık konusunda eğitin.

# İzinsiz Kullanıcı

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

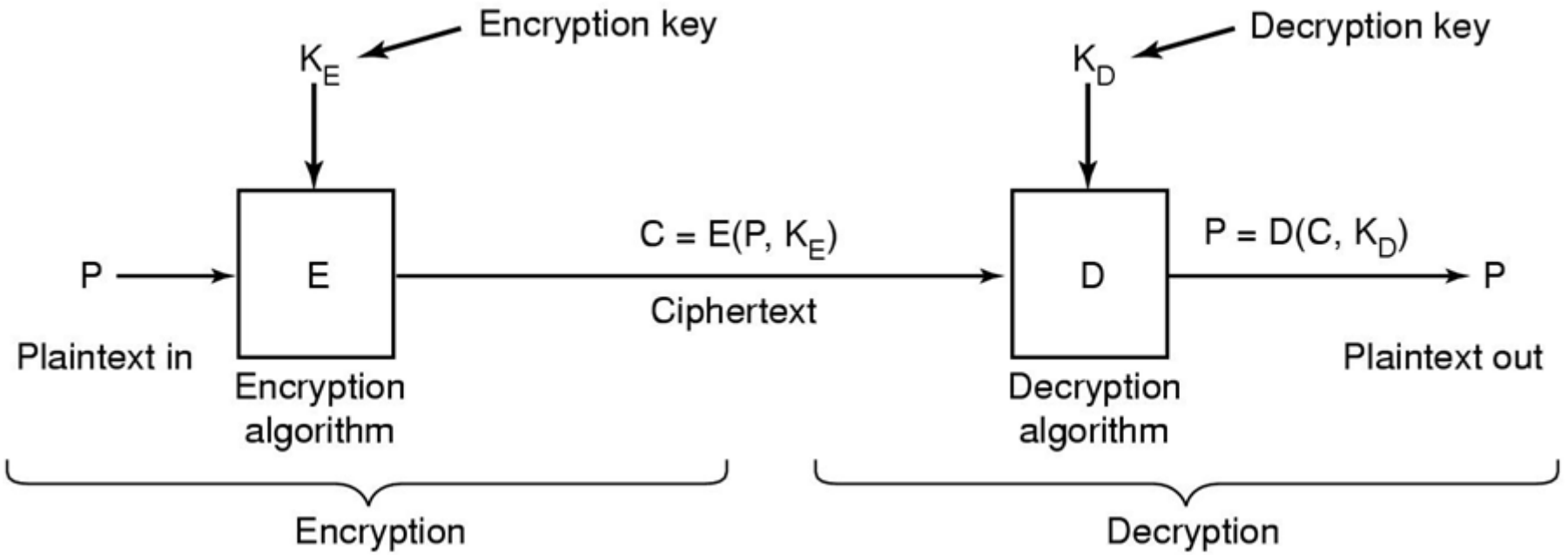
# Kazayla Veri Kaybı

- 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.

# Şifrelemenin Temelleri

- Düz metin ve şifreli metin arasındaki ilişki.



# Gizli Anahtarlı Şifreleme (secret key)

- Monoalfabetik ikame:

- Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Ciphertext: QWERTYUIOPASDFGHJKLZXCVBNM

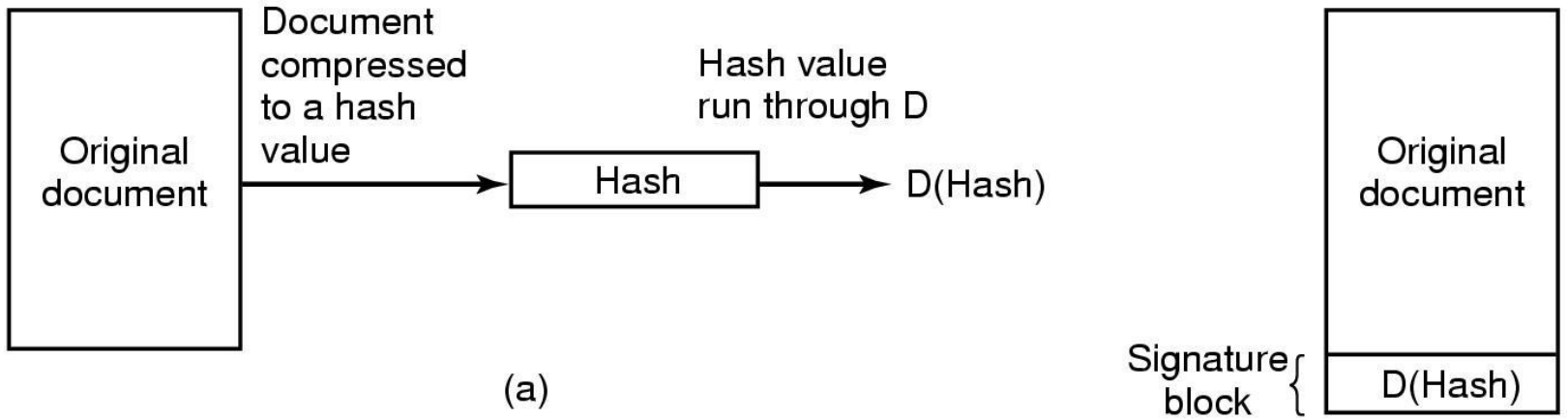
# Açık Anahtarlı Şifreleme

- Şifreleme, " 314159265358979 × 314159265358979 ne kadar" gibi "kolay" bir işlemden yararlanır?

- Anahtar olmadan şifre çözme, 3912571506419387090594828508241'nin karekökü nedir gibi zor bir işlem yapmanızı gerektirir.

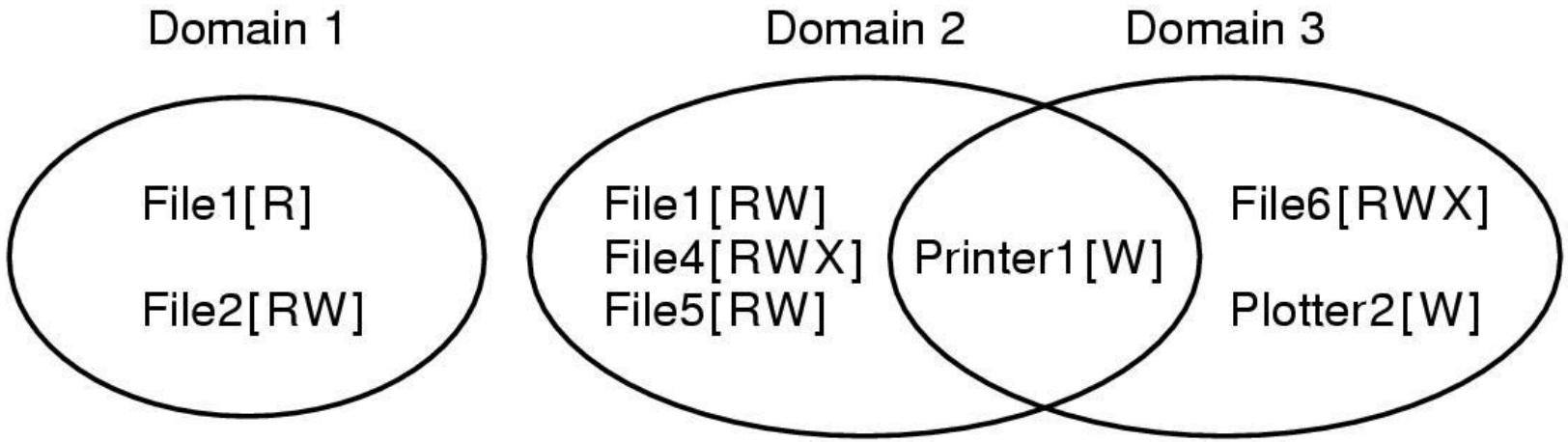
# Dijital İmzalar

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



# Koruma Etki Alanları

- Üç koruma alanı.



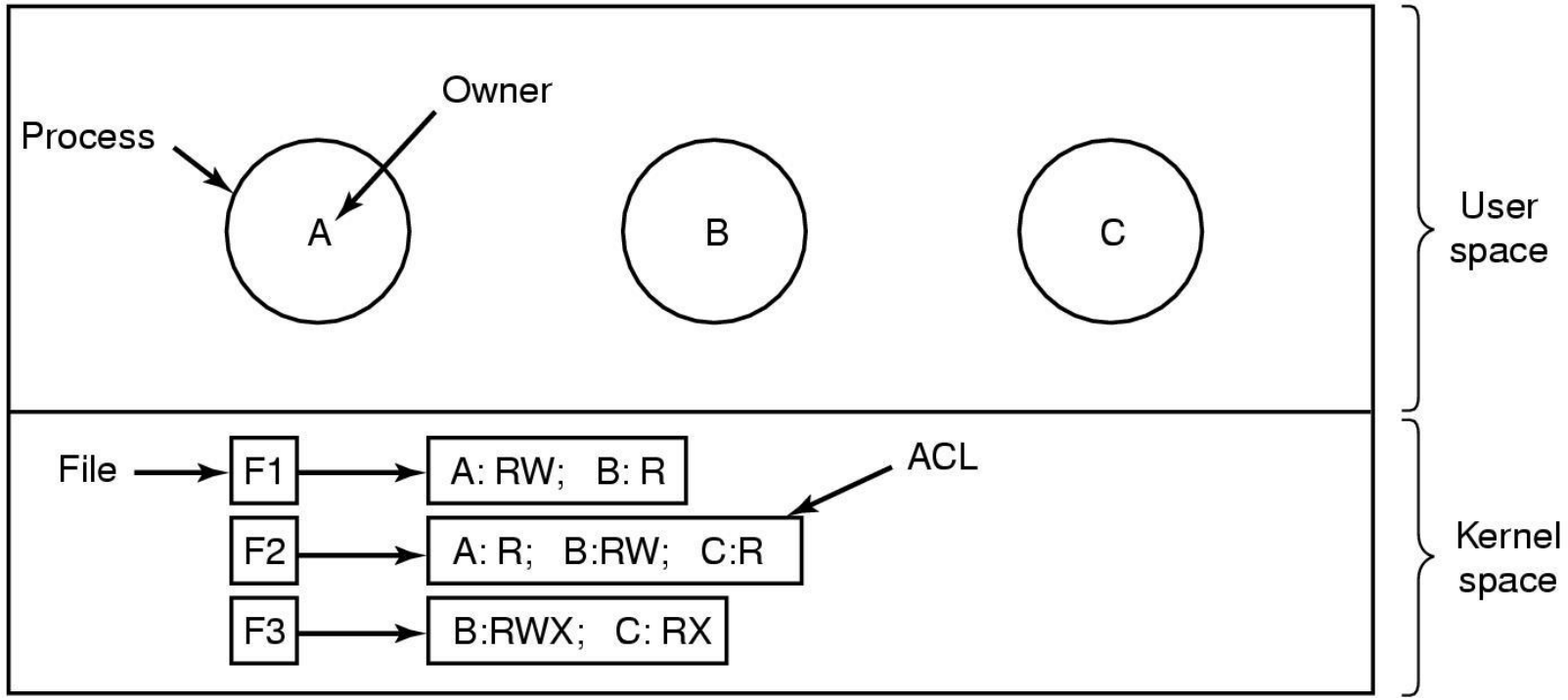
# Koruma Etki Alanları

- Bir koruma matrisi.

| | Object | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | File1 | File2 | File3 | File4 | File5 | File6 | Printer1 | Plotter2 |
| Domain 1 | Read | Read<br>Write | | | | | | |
| 2 | | | Read | Read<br>Write<br>Execute | Read<br>Write | | Write | |
| 3 | | | | | | Read<br>Write<br>Execute | Write | Write |

# Erişim Kontrol Listeleri

- Dosya erişimini yönetmek için erişim kontrol listelerinin kullanımı.



# Erişim Kontrol Listeleri

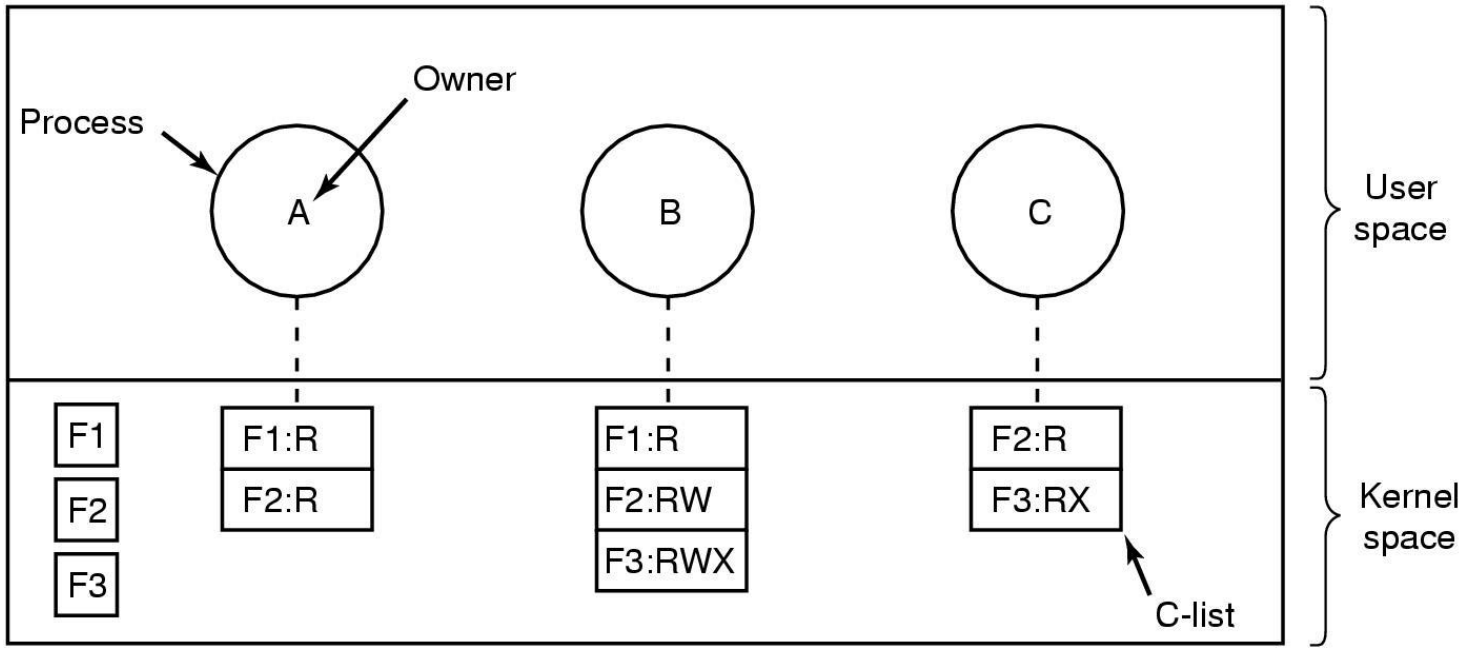
- Erişim kontrol listesi (ACL), bir işletim sistemindeki (OS) dosya ve dizinler gibi sistem kaynaklarına erişim kontrolü sağlayan bir güvenlik mekanizmasıdır.
- Nesneye kimin erişmesine izin verildiğini ve hangi eylemleri gerçekleştirmesine izin verildiğini belirlemek için izinlerin listesi
- Kaynakları kısıtlı sistemlerde performansı etkileyebilecek ek işlem gücü ve depolama gerektirir.
- Yönetmek ve güncellemek, büyük ve karmaşık sistemlerde zor.
- Farklı işletim sistemleri ve uygulamalar arasında farklılık gösterebilir ve bu da uyumluluk ve birlikte çalışabilirlik sorunlarına yol açar.

# Erişim Kontrol Listeleri

- User Group 1: Read-only access to file A, full access to file B.
- User Group 2: Write access to file A, no access to file B.
- User Group 3: Execute access to file C, read access to file D.
- User Group 4: No access to files A, B, C, and D.
- Admin Group: Full access to all files.

# 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.

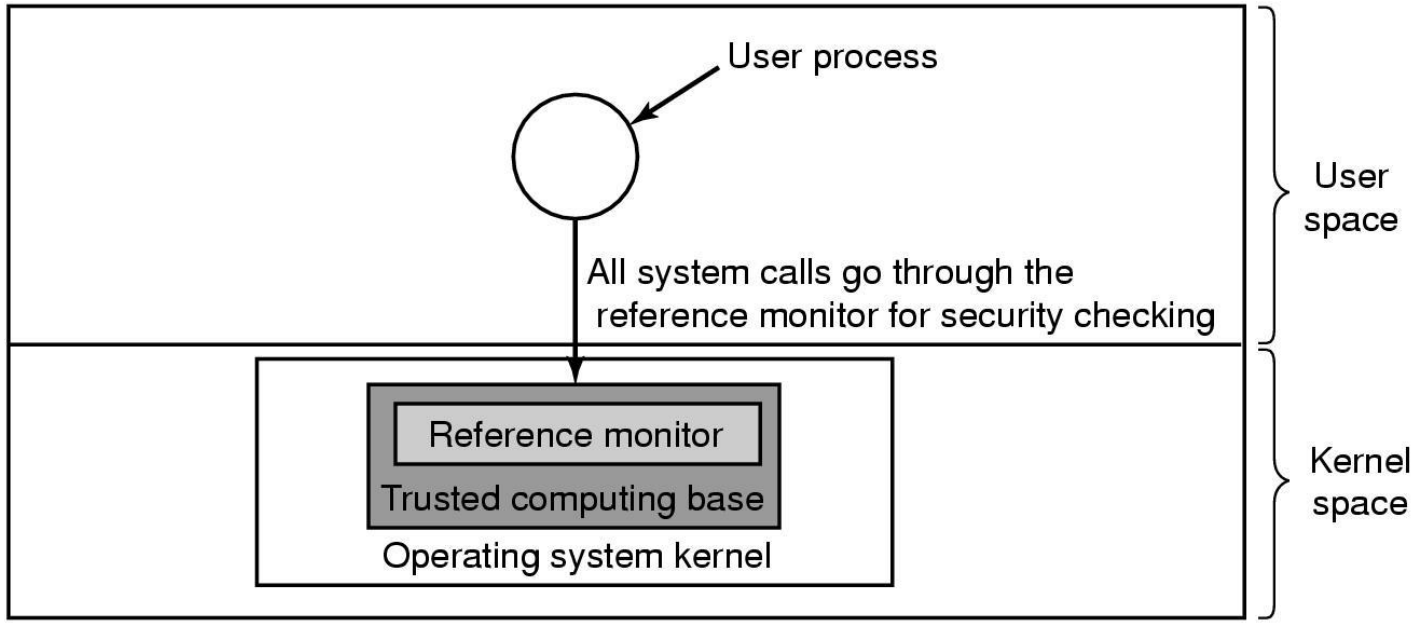
# Güvenilir Sistemler

- Virüs, solucan vb. raporları dikkate alın.

- İki saf (ama mantıklı) soru:
  - Güvenli bir bilgisayar sistemi kurmak mümkün mü?
  - Varsa neden yapılmıyor?

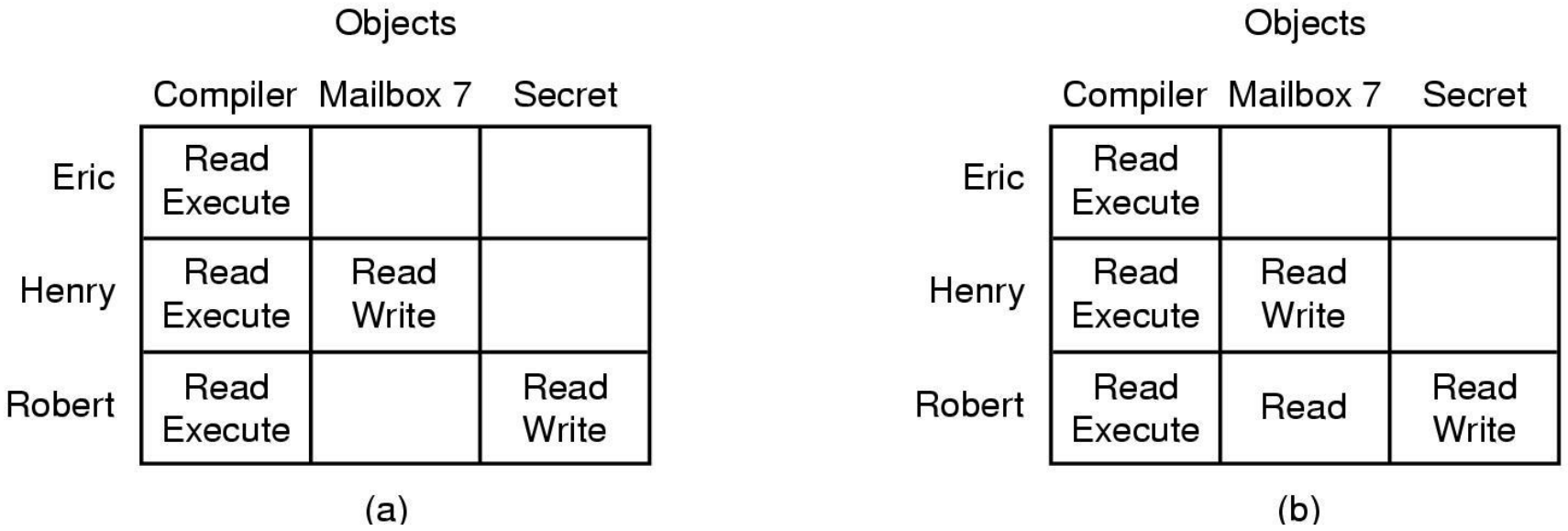
# Güvenilir Bilgi İşlem Tabanı

- 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)

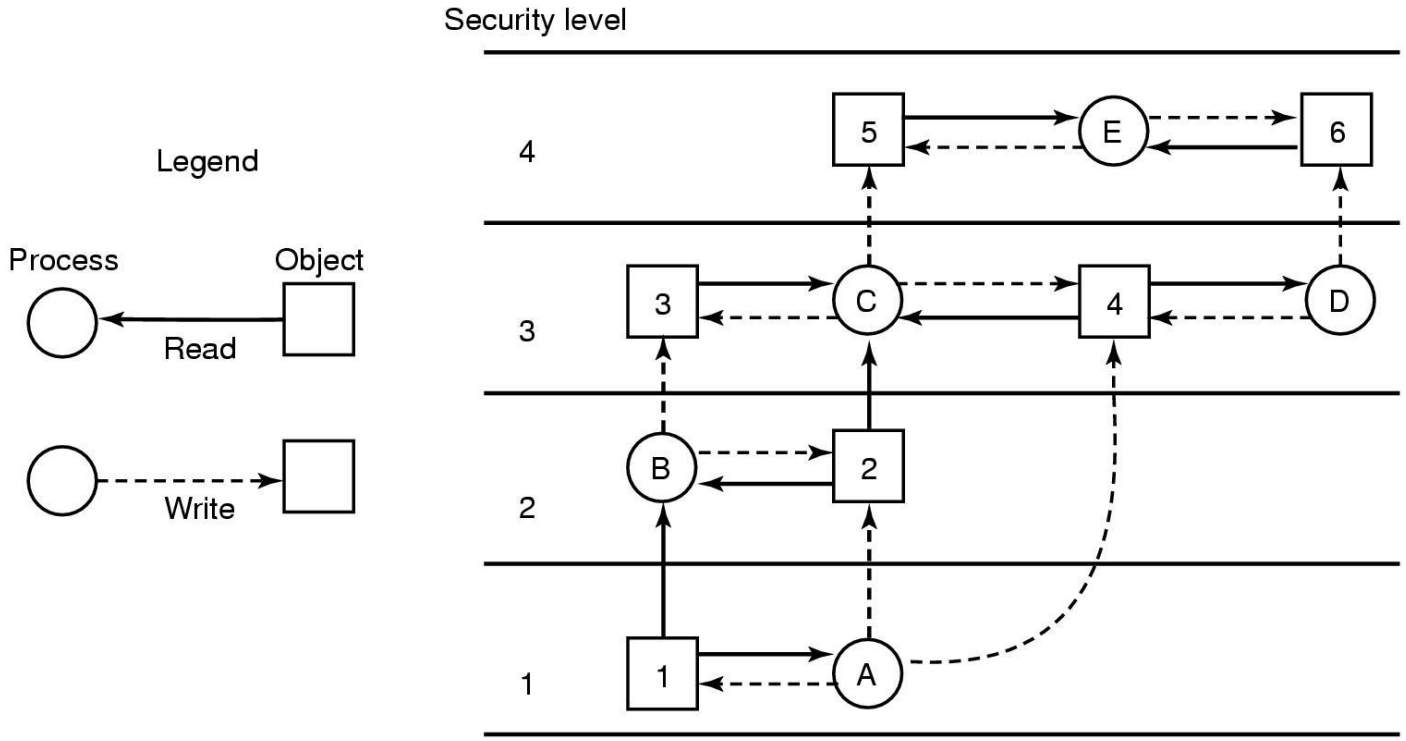
# Bell-La Padula Modeli

- 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.

# Bell-La Padula Modeli

- Gizli bilgilerin güvenliğini sağlamak amacıyla askeri kullanım için geliştirildi
- Gizliliğe odaklanır ve verilere yetkisiz erişime karşı koruma sağlar
- Öznelerin ve nesnelerin sınıflandırma seviyelerine göre erişim kontrolünü tanımlar
- Güvenliği sağlamak için "okuma yok" ve "yazma yok" ilkesini kullanır

# Biba Modeli

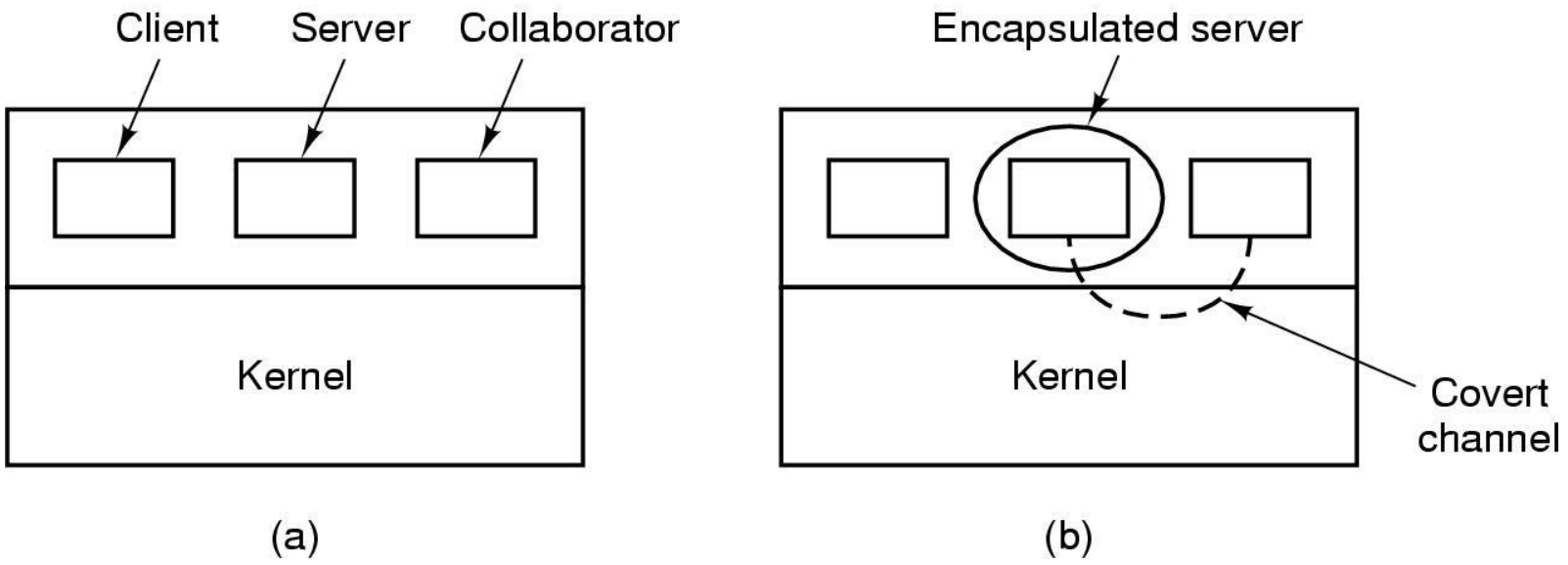
- Verilerde veya sistemlerde yetkisiz değişikliklere karşı koruma sağlamak için geliştirildi
- Bütünlüğe odaklanır ve verilerde veya sistemlerde yetkisiz değişiklikleri önler
- Öznelerin ve nesnelerin bütünlük düzeylerine göre erişim kontrolünü tanımlar
- Güvenliği sağlamak için "okuma yok" ve "yazma yok" ilkesini kullanır

# Covert Channel

- normal güvenlik mekanizmalarını ve ilkelerini atlayarak bir bilgisayar sistemindeki süreçler arasında bilgi ileten bir mekanizmadır.
- Gizli depolama kanalı: Veriler, bilgilerin depolanmasındaki değişiklikler yoluyla iletilir.
- Zamanlama gizli kanalı: Veriler, olayların zamanlamasındaki değişiklikler yoluyla iletilir.
- Gizli kaynak kanalı: Veriler, sistem kaynaklarının kullanımı değiştirilerek iletilir.
- Gizli kanallar, verilerin gizliliğini ve/veya bütünlüğünü tehlikeye atarak güvenlik mekanizmalarından kaçmak için kullanılabilir.

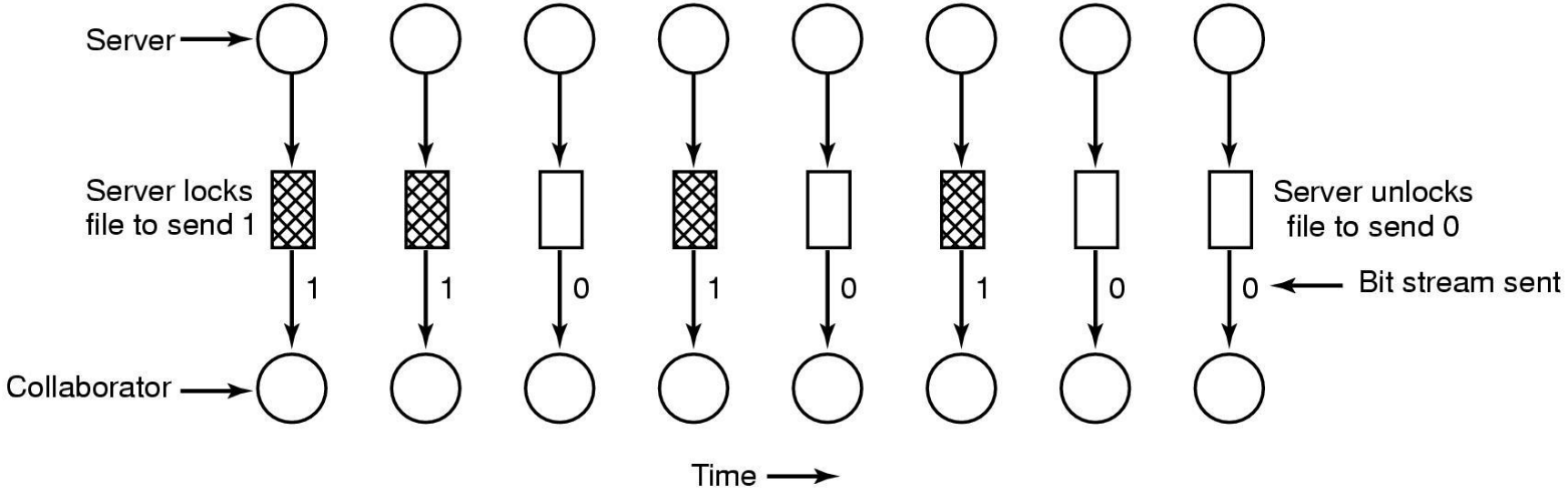
# Gizli Kanallar

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



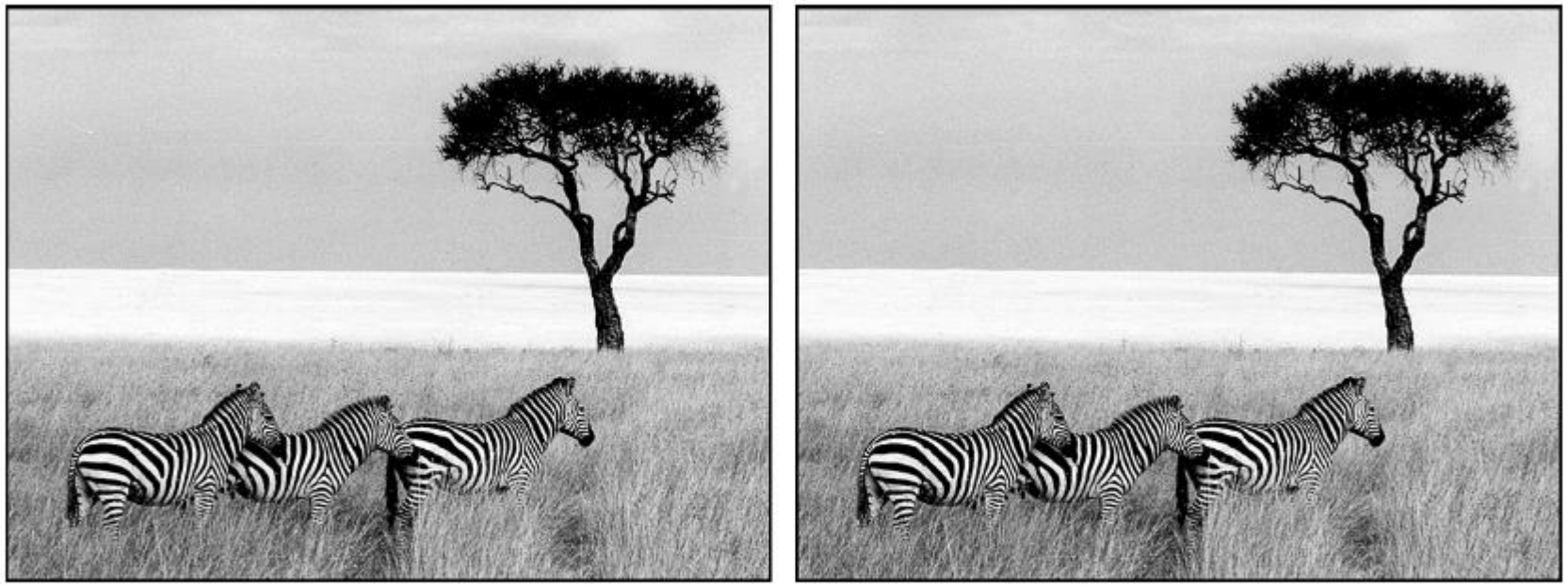
# Gizli Kanallar

- 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)

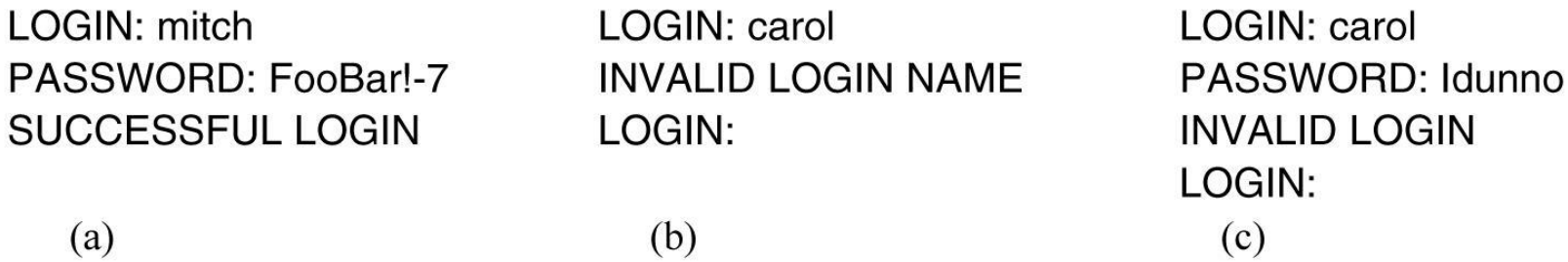
# Kimlik Doğrulama

- 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.



# 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

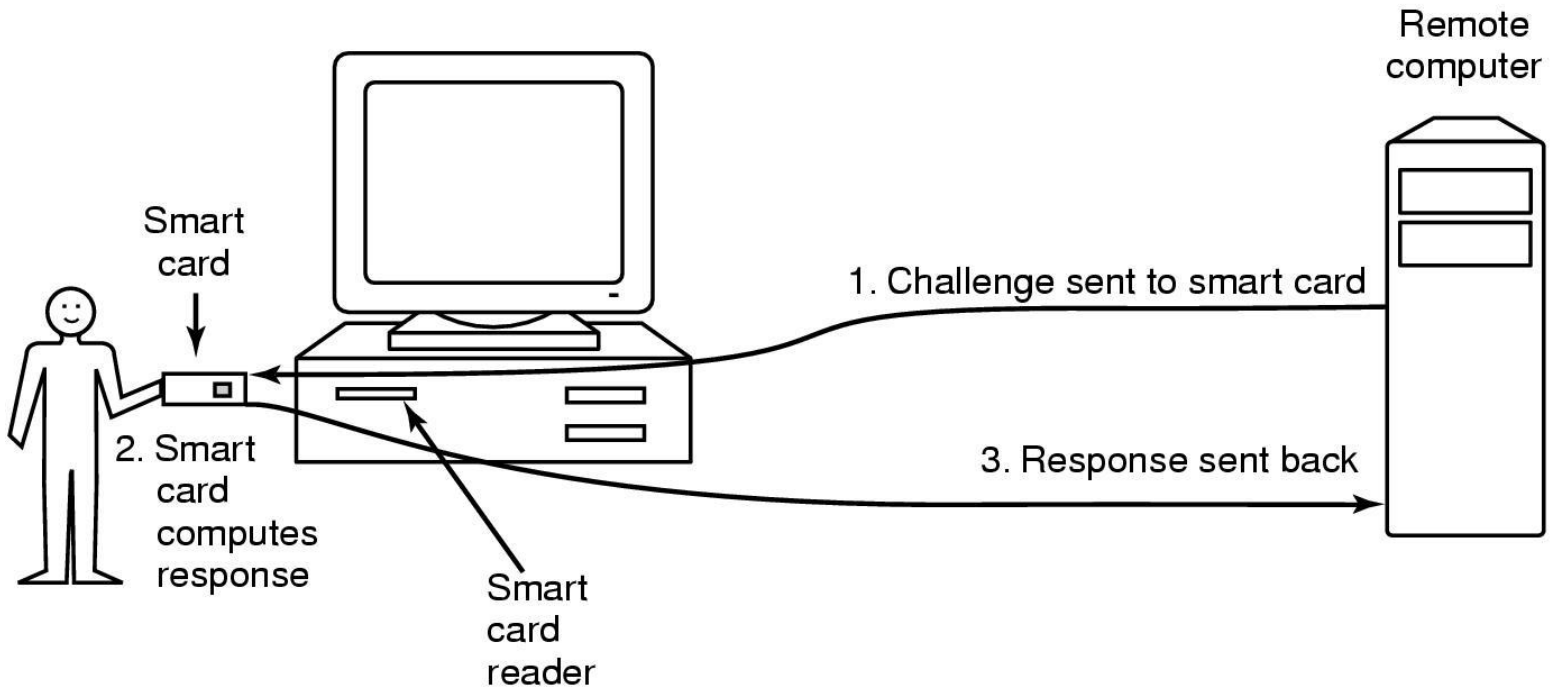
- Sorular, kullanıcının yazmasını gerektirmeyecek şekilde seçilmelidir.

Örnekler:

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

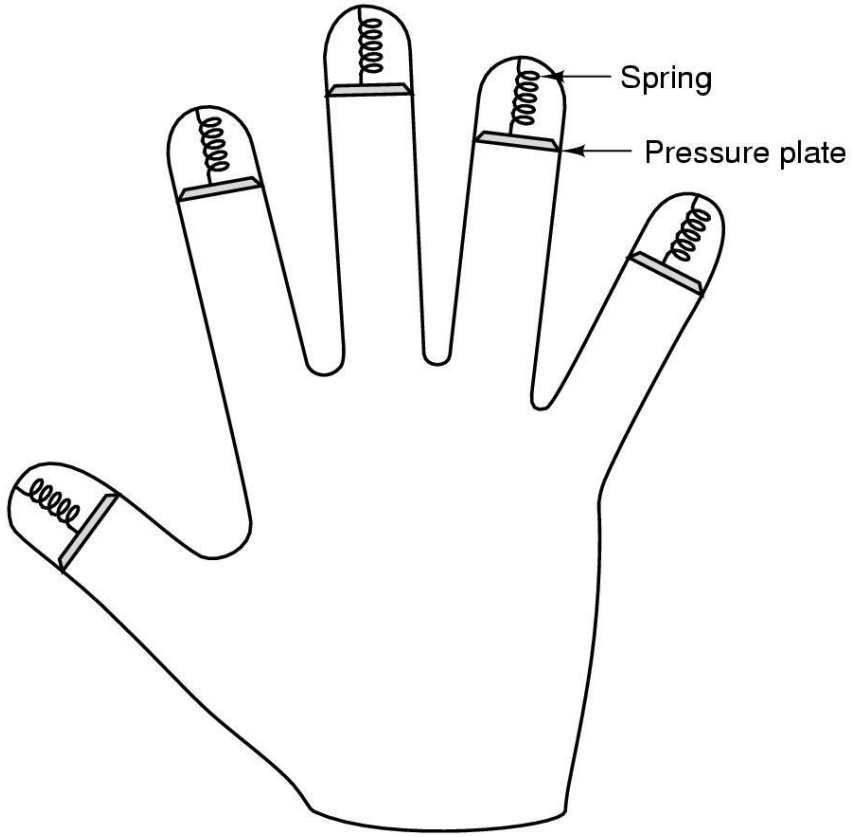
# Fiziksel Nesne Kullanarak Kimlik Doğrulama

- 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);
```

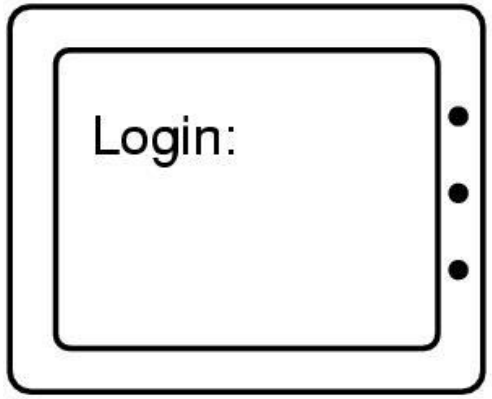
(a)

```
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);
```

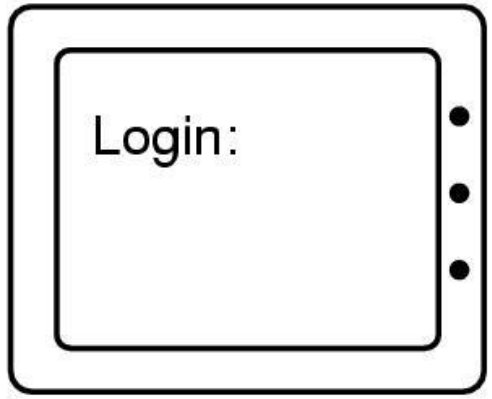
(b)

# Giriş Sayfası Kandırma Saldırısı

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




(a)




(b)

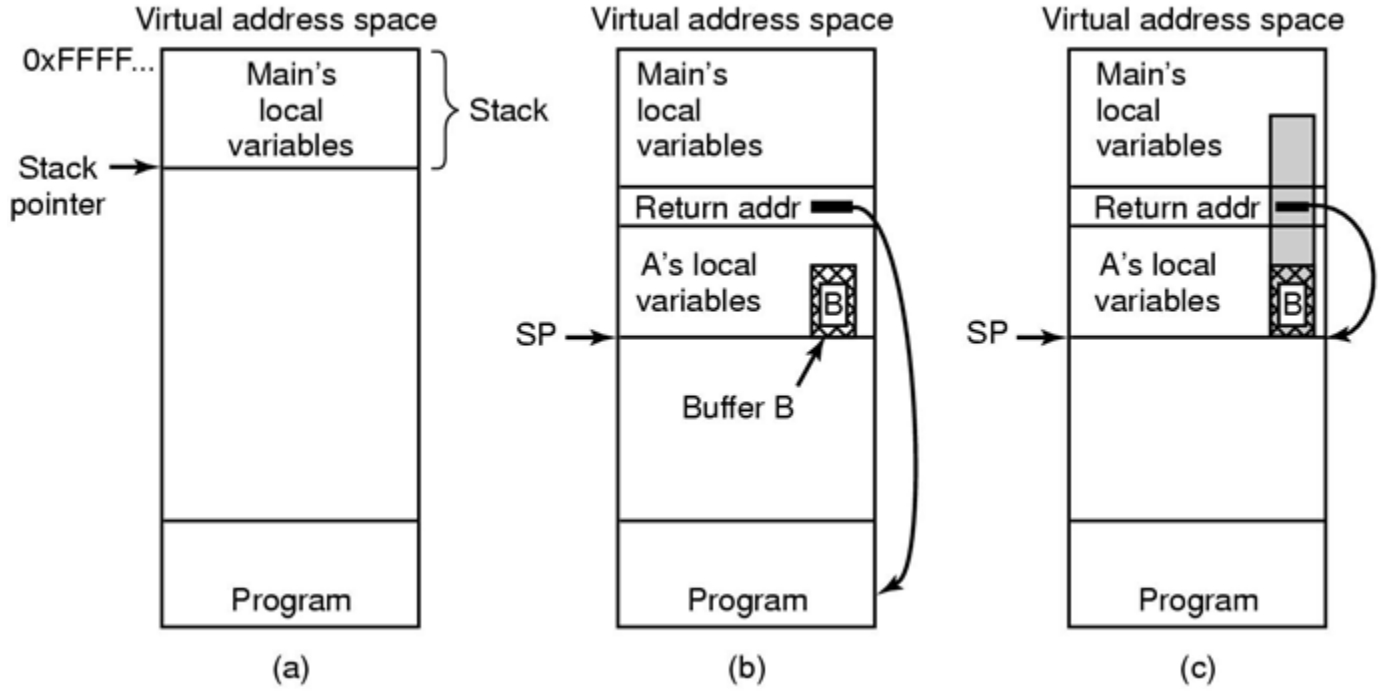
# Kod Hatalarından Yararlanma

- 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.

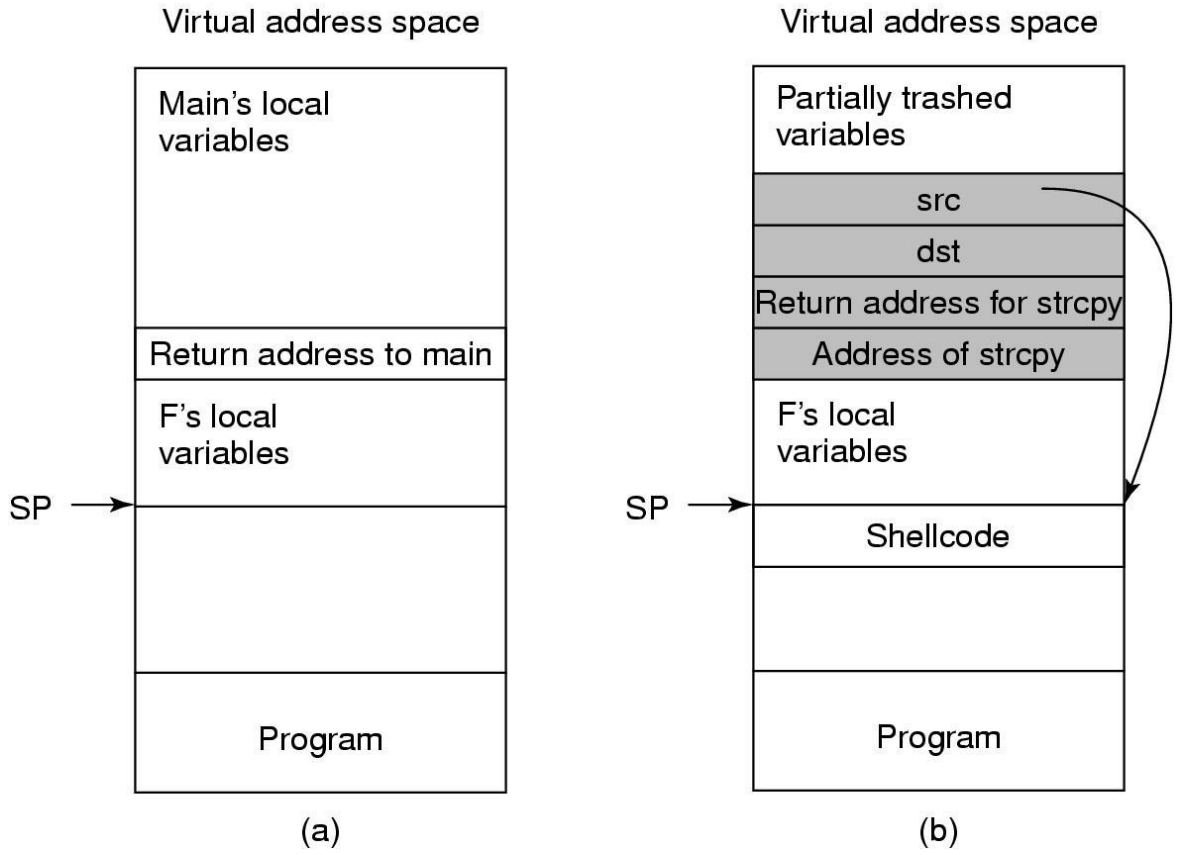
# Tampon Taşma Saldırıları

- (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 ";          /* declare 3 strings */
  printf("Please enter name of source file: ");       /* ask for source file */
  gets(src);                                          /* get input from the keyboard */
  strcat(cmd, src);                                   /* concatenate src after cp */
  strcat(cmd, " ");                                   /* add a space to the end of cmd */
  printf("Please enter name of destination file: ");  /* ask for output file name */
  gets(dst);                                          /* get input from the keyboard */
  strcat(cmd, dst);                                   /* complete the commands string */
  system(cmd);                                        /* 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

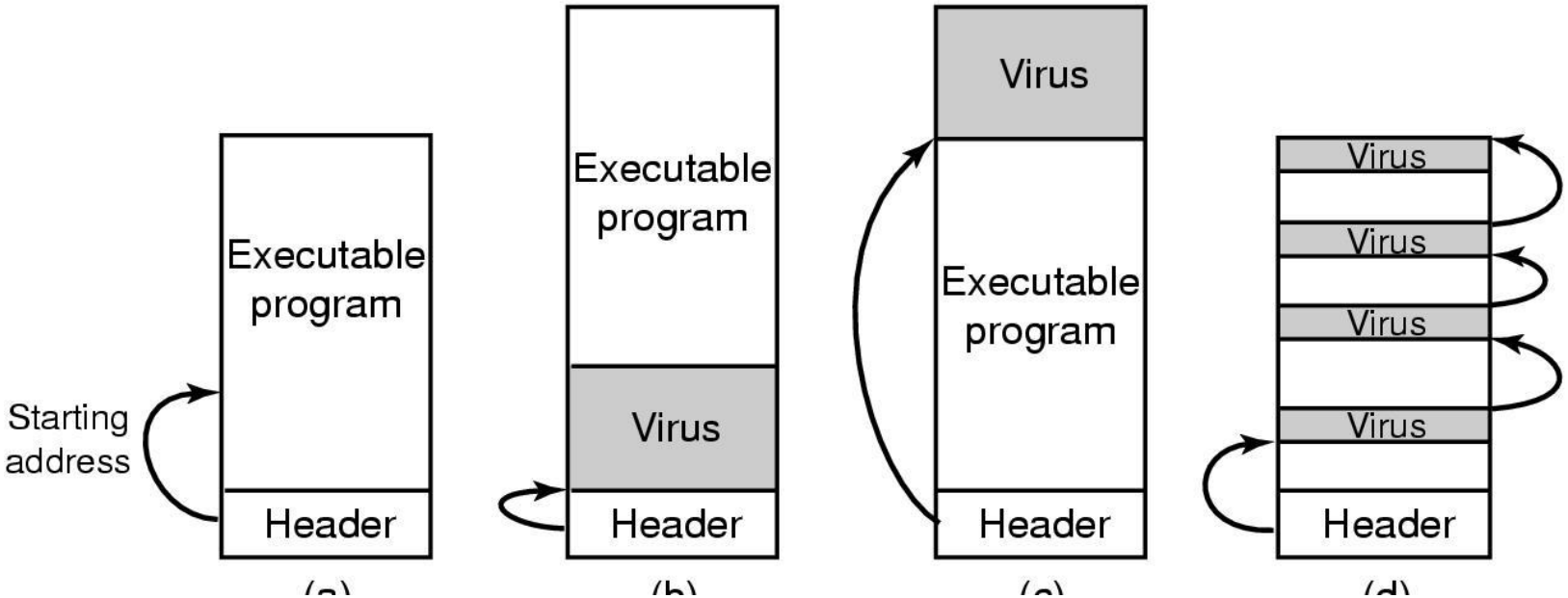
To purchase a decryption key for your hard disk, please send $100 in small unmarked bills to Box 2154, Panama City, Panama.
Thank you. We appreciate your business.

# 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)

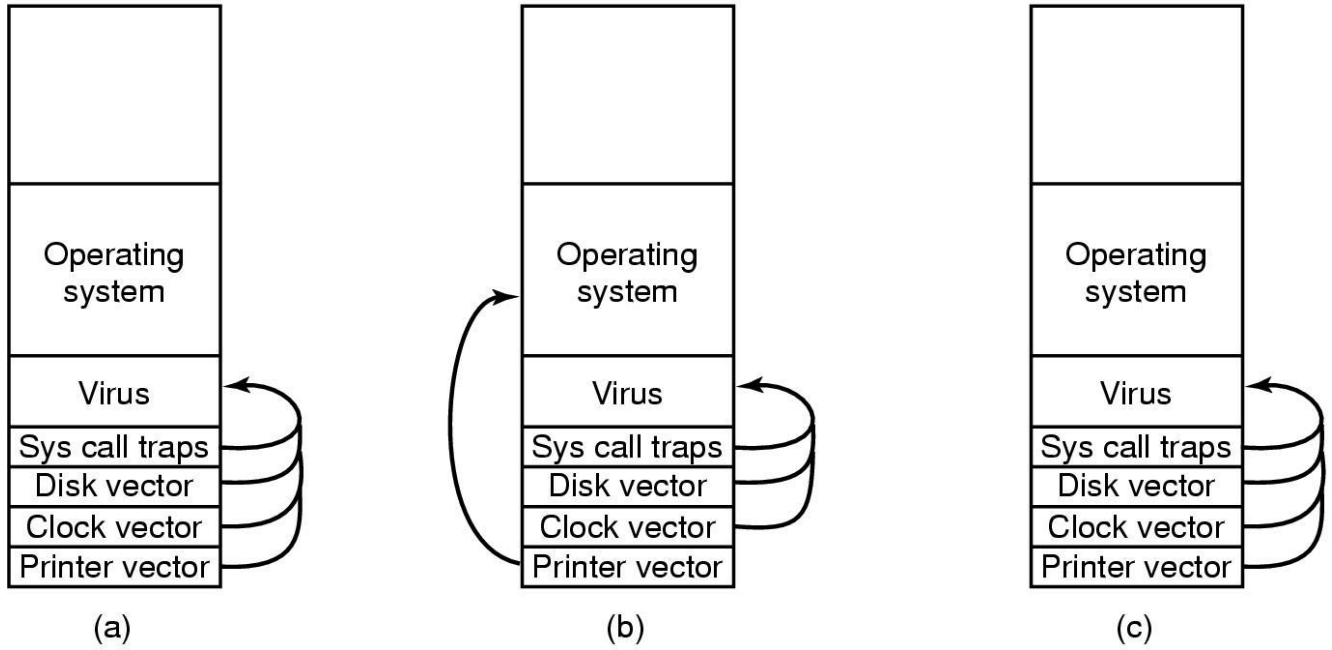
# Parazitik Virüsler

- (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

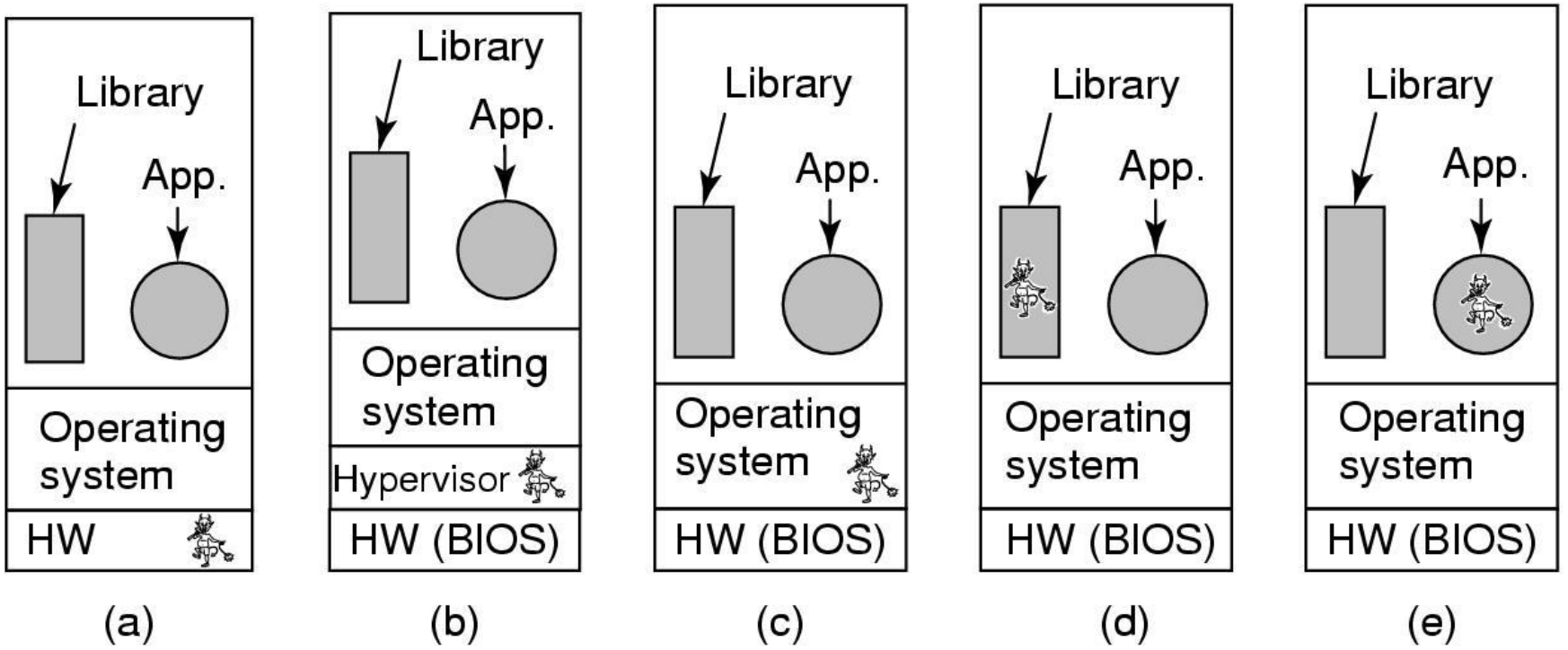
- Tarayıcının ana sayfasını değiştirme.
- Tarayıcının favori (yer imi eklenmiş) sayfalar listesini değiştirme.
- Tarayıcıya yeni araç çubukları ekleme.
- Varsayılan medya yürütücüsünü değiştirme.
- Varsayılan arama motorunu değiştirme.
- Windows masaüstüne yeni simgeler ekleme.
- Web sayfalarındaki banner 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

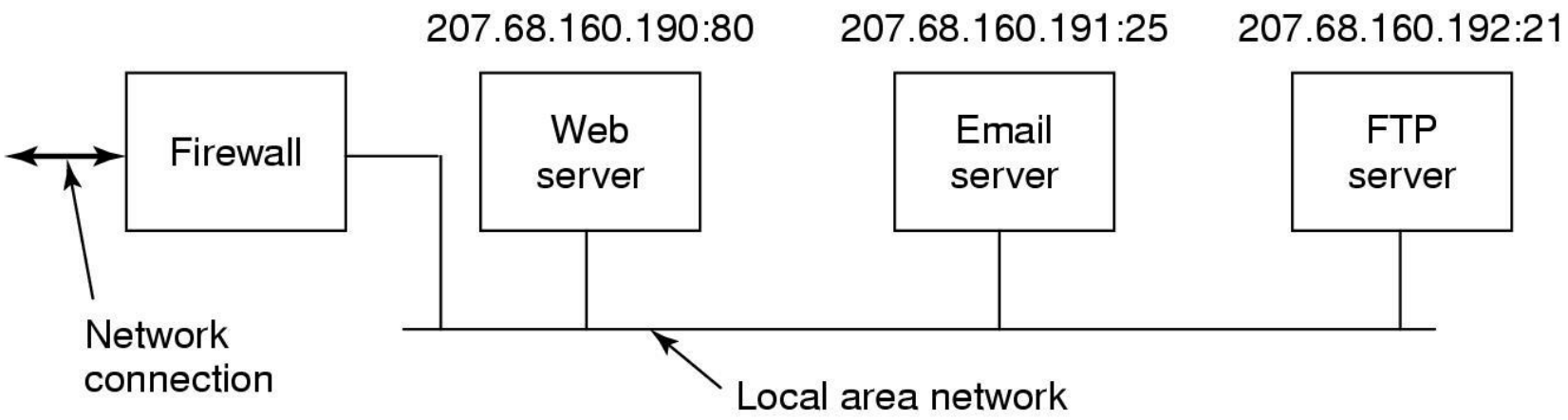
# Bir Rootkit'in Saklanabileceği Beş Yer

- .



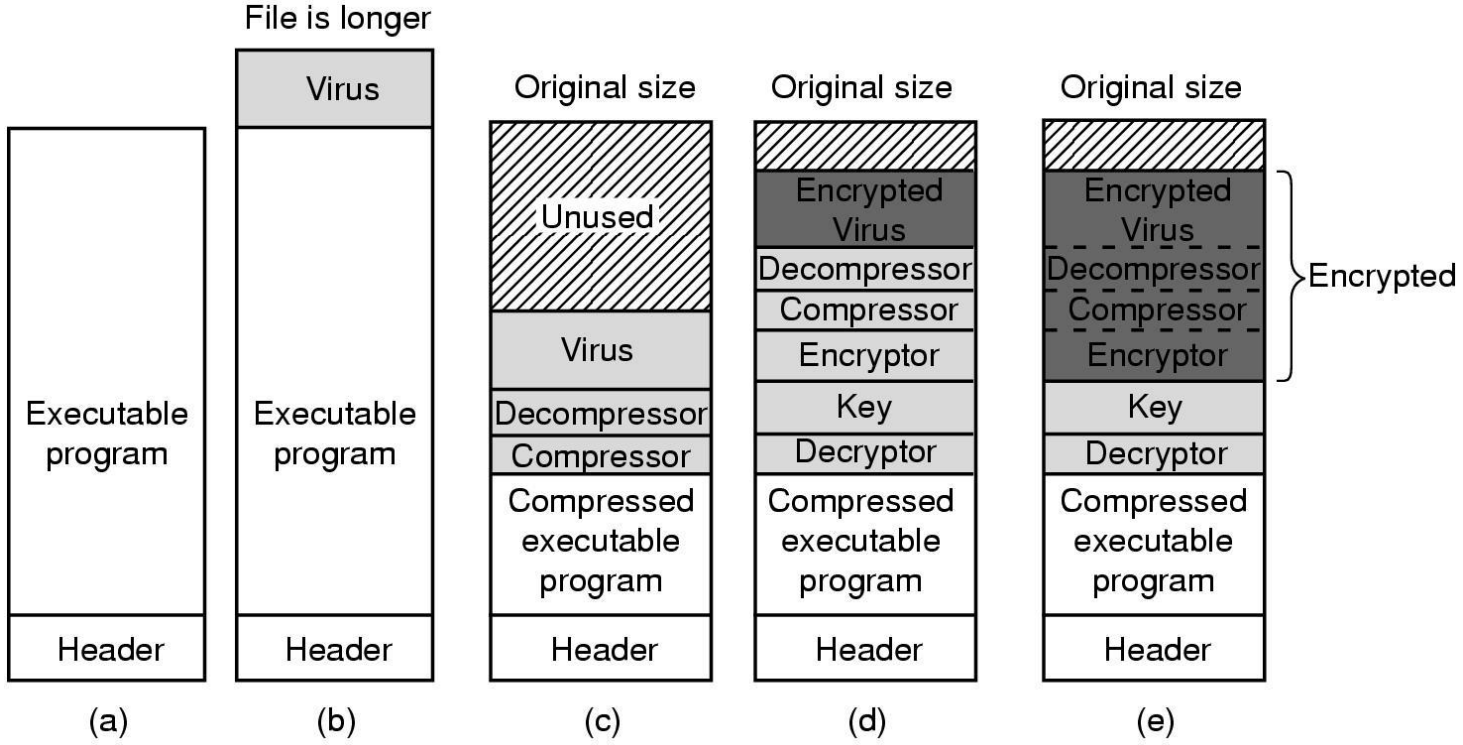
# Güvenlik Duvarı

- 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
```

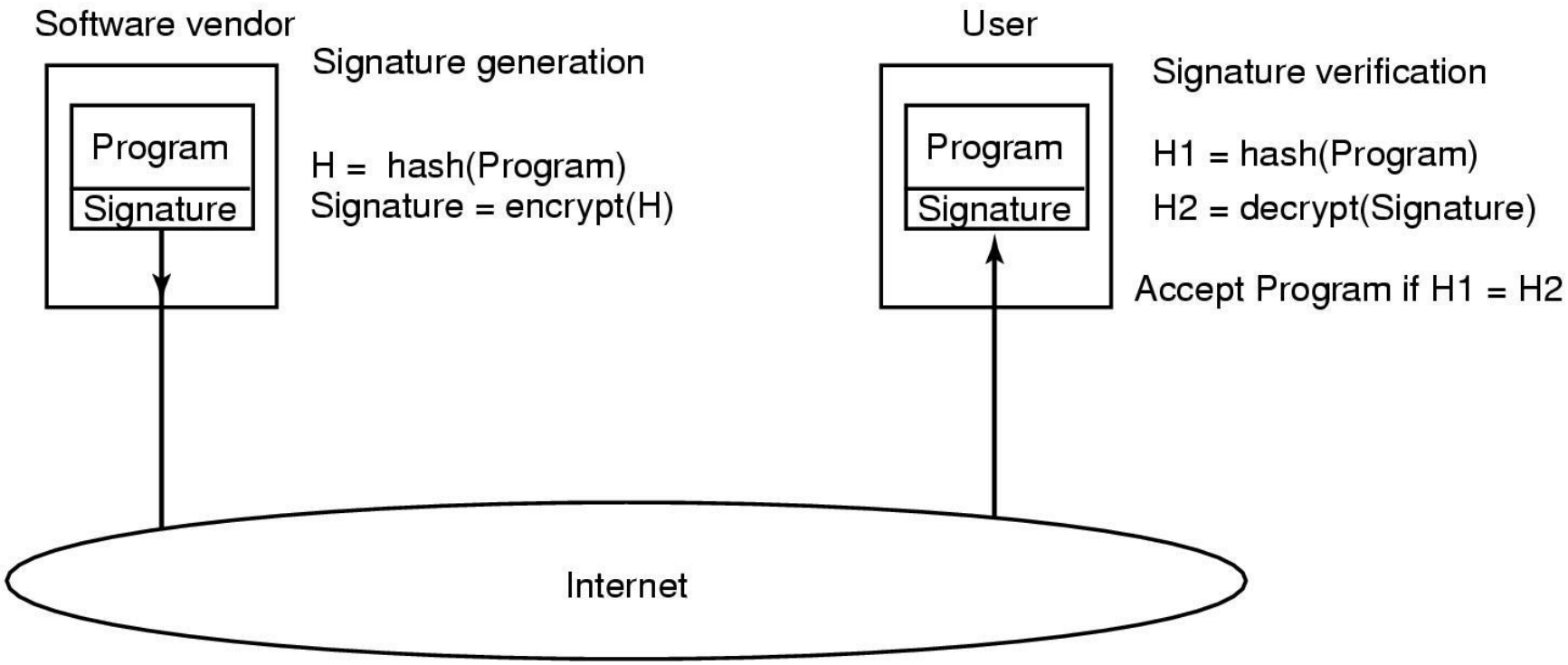
(e)

# Antivirüs ve Anti-Antivirüs Teknikleri

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

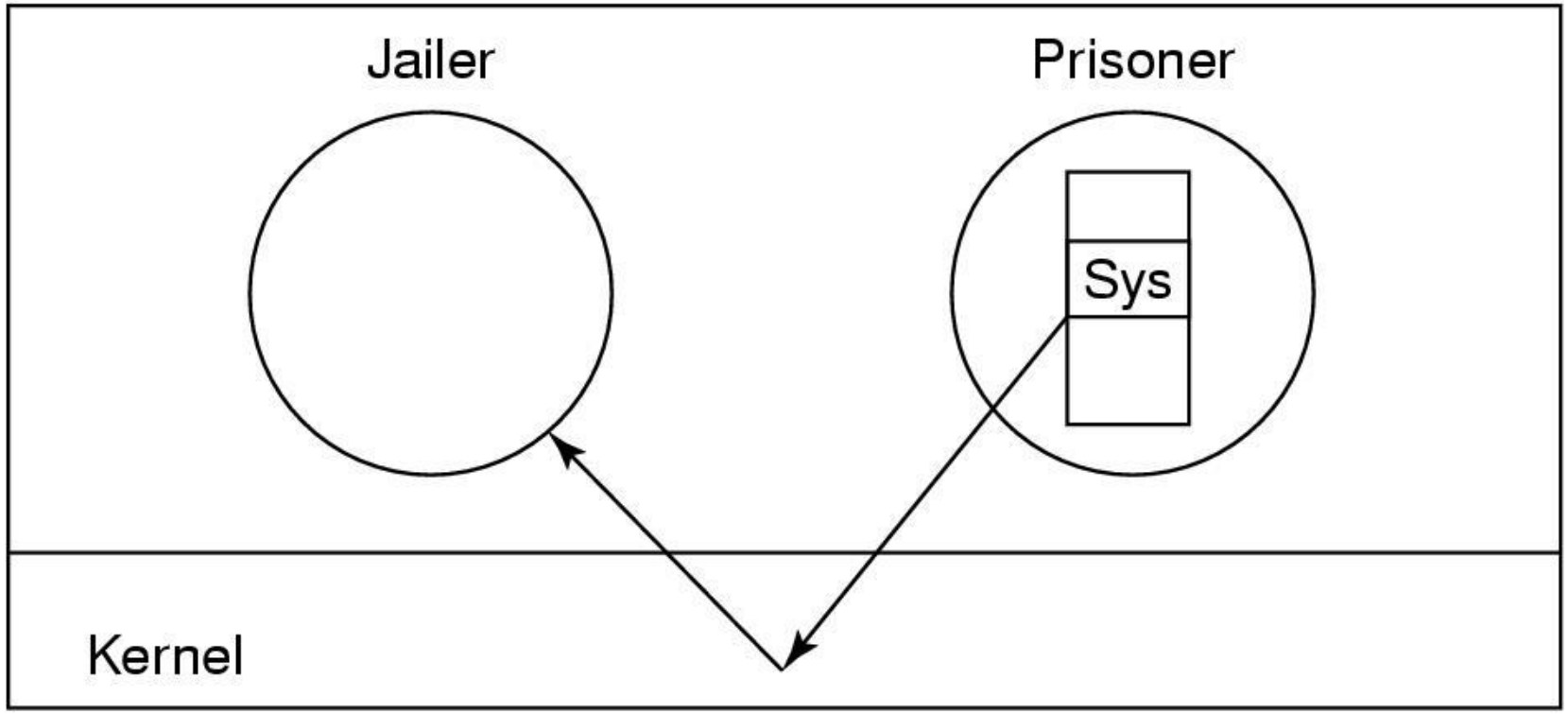
# Kod İmzalama Nasıl Çalışır?

- .



# Hapse Atmak (jailing)

- .



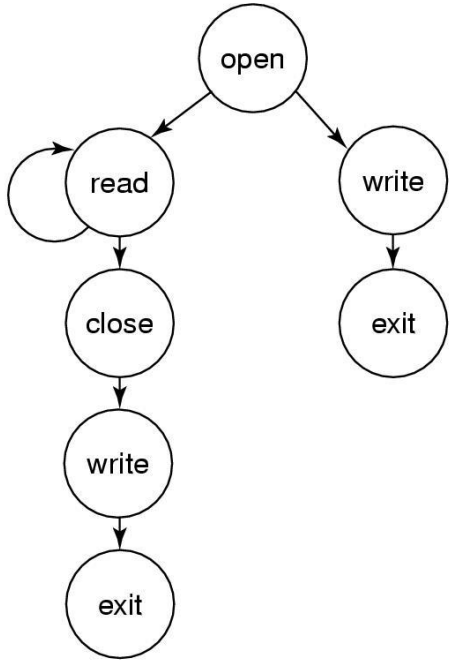
# Model Tabanlı Saldırı Tespiti

- (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;
      }
  }
}
```

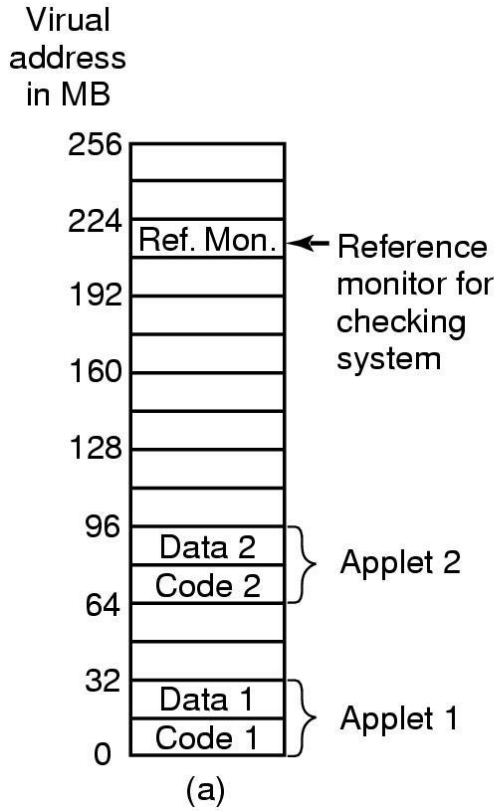
(a)




(b)

# Korumalı Alan (sandboxing)

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

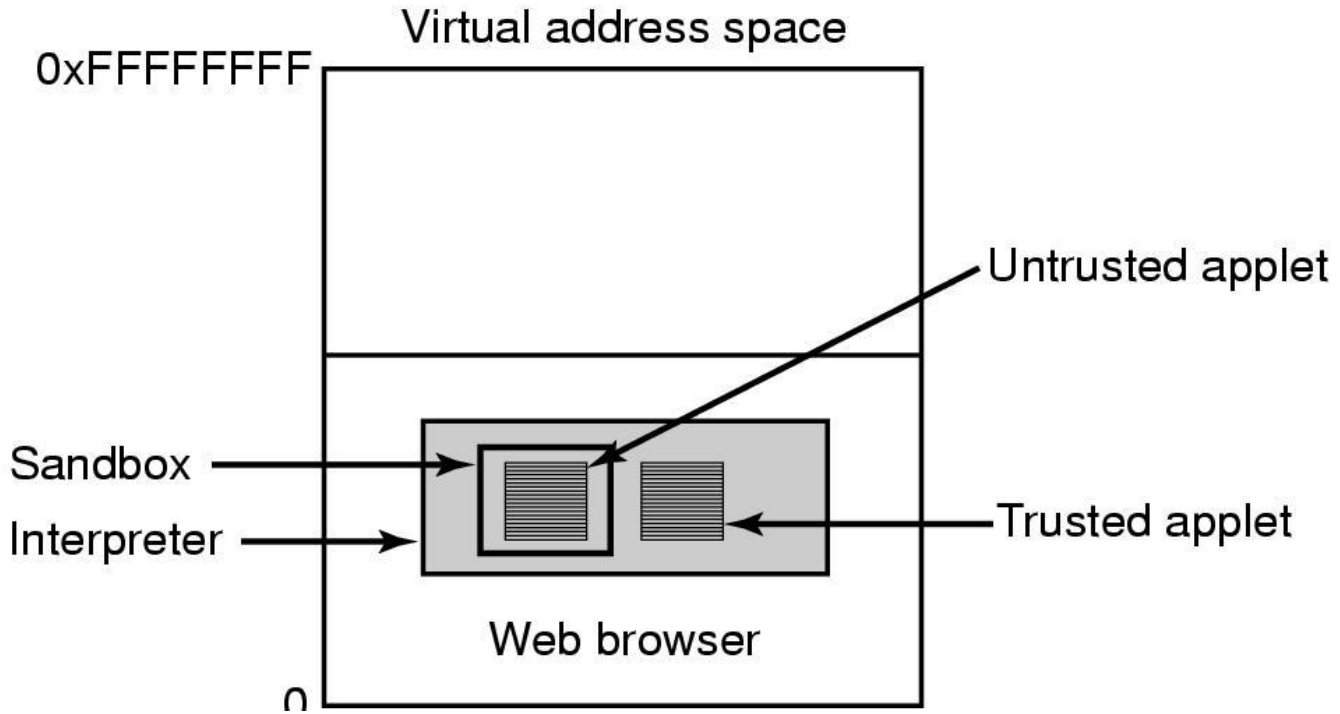



```
MOV R1, S1
SHR #24, S1
CMP S1, S2
TRAPNE
JMP (R1)
```

(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