ağ güvenliği- isolation

Güvenilir olmayan kodları günlük hayatta sık sık kullanıyoruz:

-internet aracılığıyla

-saldırıya uğramış uygulamalar

-legacy daemons

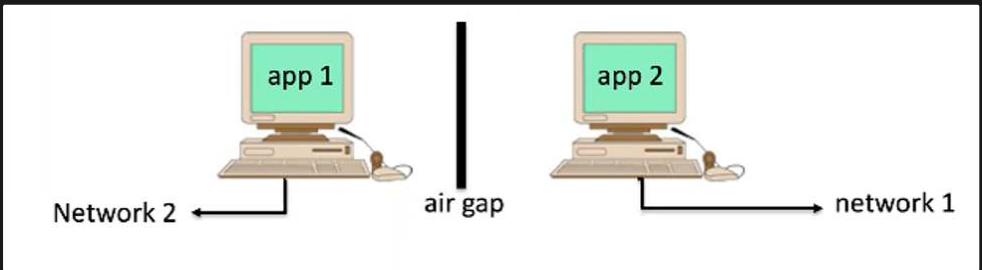
-honeypots

Başarısız sonuçlanan bir saldırı DoS'a sebep oluyor. Programı kapatmamızdan başka çaremiz yok.

Isolation'da amaç zararlı yazılımın belirli bir alanda kalmasıdır. Yaklaşımlar:

1-) Confinement(Kapatılma)

a-) Donanımın ayrılması(Air Gap)



donanım seviyesinde izolasyon. Her uygulama farklı pc sisteminde ve fiziksel olarak ayrı. Bir taraftaki hata diğerini etkilememeli.

Zorlukları: ancak bunu yapabilmek için aynı zamanda networkleri de ayırmak lazım, çok para. internette yer almak isteyen her yapının name servisi olması gerekiyor, frontendi olması lazım vs. vs. o zaman paran kadar işlemcisiyle memorysiyle ayrı bilgisayarlar kurduk diyelim, bu pc sayısı kadar network kurmamız mümkün değil bu sefer de. yönetimsel olarak da sıkıntı, hepsini aynı anda güncellicez falan.

b-)Virtual machine

Bir bilgisayar sisteminin tüm birimleri izole olacak şekilde parçalara bölünmesi. Üst taraflar farklı donanımlarmış gibi gösteriliyor.

c-)System Call Interposition(müdehale)

Process işletim sistemiyle sistem çağrısında etkileşir. Processlerin zaten memoryleri ayrıdır. Sistem çaprılarını kullanarak processlerin birbirlerini etkileyebilecekleri aktiviteleri engelleriz.

d-)Software Fault Isolation (SFI)

Process içerisinde birden fazla akış içerebilir ve bunlar thread içerir. Bu processleri ve threadleri de izole etmek gerekir. Ama bunu yapmak daha zordur çünkü adresleri aynıdır ve erişimleri yüksektir.

e-)Reference monitor

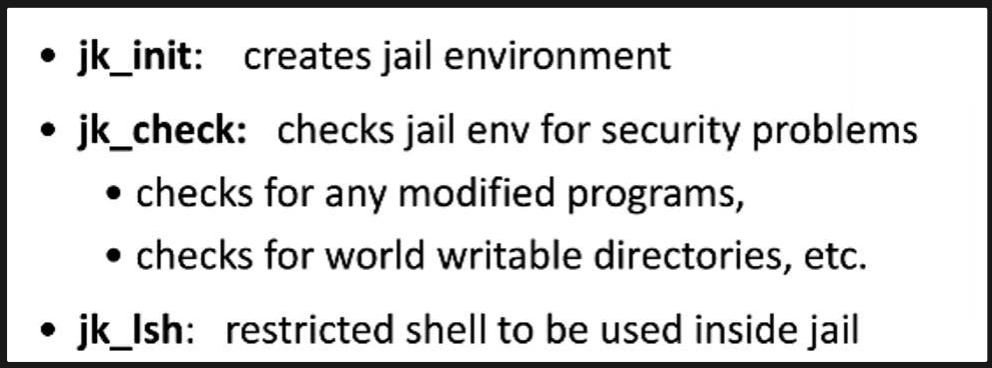
Uygulamalar işletim sist ile etkileşime geçeceği zaman kontrol mekanizması olur. İşletim sistemine erişecek bütün istekler reference monitor'den geçer. Reference monitor devredışı bırakılamamalı!

f-)Chroot

Change root. Yalnız root kullanıcısı değiştirilebilmeli. Sadece dosyalarla alakalı. Çıkarılan ilk confinement. Network erişimini kısıtlamaz o açıdan zaafiyeti var.

g-)Jailkit

Tüm yardımcı programlar jail'de yaşar. Jail'deki dosyaları, kütüphaneleri vs. jailkit otomatik olarak oluşturur.



Jail'den kaçmak için chroot kullanılmış.

Jail ve chroot'un sıkıntılı tarafı hassas ayar olmuyor oluşu, ya hep ya hiç mantığı var.

h-)Freebsd

Bir uygulamayı sadece dosya değil, bağlanabileceği ip adresleri vs açısından da kısıtlıyor.

2-) System Call Interposition (SCI)

Uygulamanın sistem çağrılarını gözlemleyip uygun bulmadıklarımızı engelleriz.

Uygulamanın etrafına yerleştireceğimiz izolasyona sandbox diyoruz. Chroot ve jail de sandbox sayılır.

Zararlı sistem çağrıları:

a-)Kernel'da

b-)User space'de

c-)Hybrid(Kernel ve user tarafında)

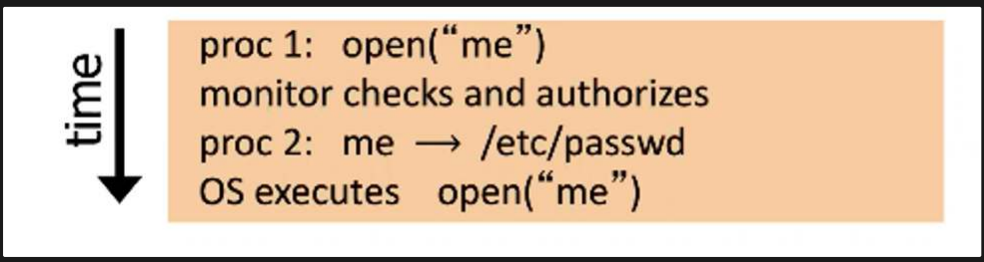
Bu teknolojinin ilk uygulamaları Linux'ta ptrace(process trace). Tüm sistem çağrılarını takip eden veya hiçbir sistem çağrısını takip etmeyen uygulamalar veya sistem çağrısını iptal ederken sistemi kill eden uygulamalar için uygun değil.

Komplikasyonlar:

-Bir uygulama kendinden kopya oluşturma ihtiyacı duyabiliyor(fork). Bir uygulamayı takip eden başka bir uygulama da fork oluşturmak zorunda kalacak, sistemdeki işler iki katına çıkacak.

-Monitor'un başına bir şey geldiğinde tüm uygulamanın sonlanması lazım.

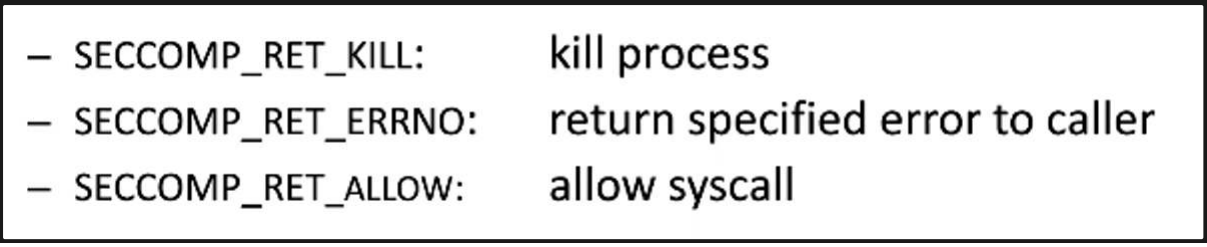
-Monitor uygulamasının uygulamanın yaptığı değişiklikleri takip etmesi gerekiyor.



TOCTOU: Time of check time of use

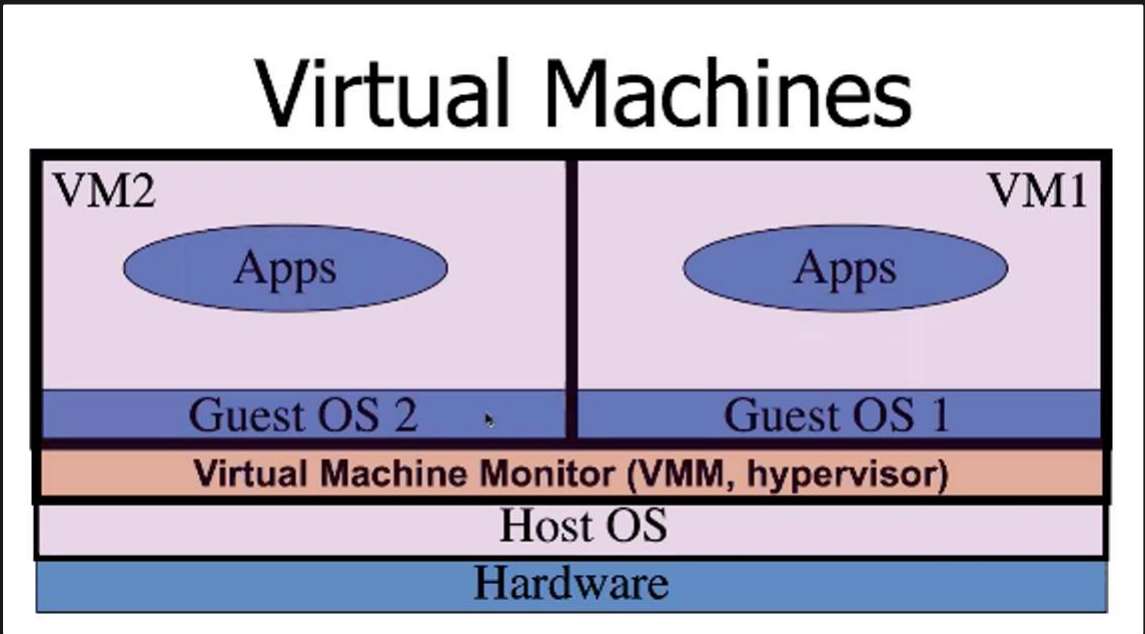
BPF Filters()

Yüklendi mi silinemezler restart, forklanırsa child aynı filtrelere sahip oluyor. Üç input alıyor: yscall num, syscal args, arch. Üç dönütten biri olabiliyor:



3-)Virtual Machine

Aynı VM farklı veya aynı kişilere ait olabilir.



Özelliğe sahip olan ve olmayan veriyi aynı anda kullanıyoruz.

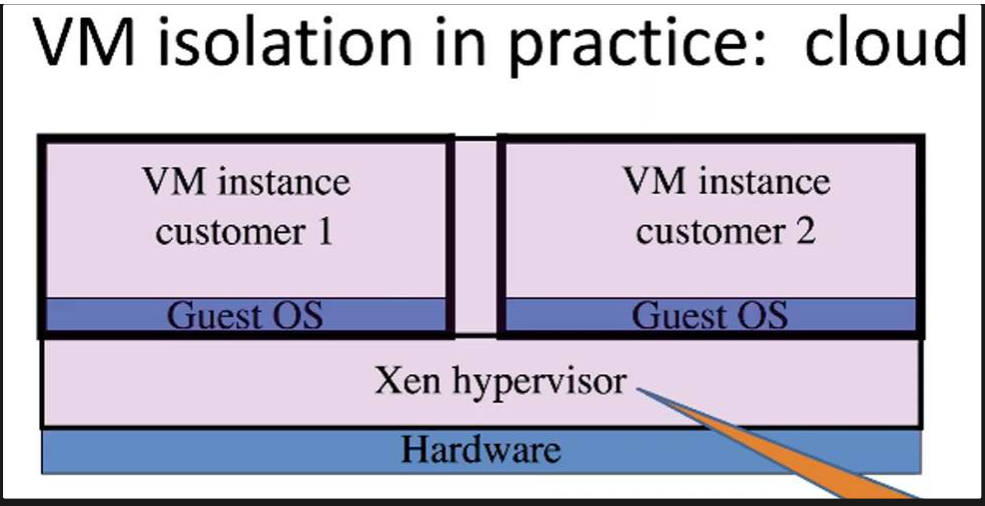
Misafir işletim sistemi ele geçirilebilir ancak host os ve diğer sistemleri ele geçiremez olması gerek. Tam işletim sistemi gibi olmasa da benzer çalışabilmesi gerek. Sorunlar:

Covert channels

Componentler arası kasıt dışı haberleşmeye denir. İşlemci gücünün paylaşılması bunu tetikleyebilir.

Hypervisor

Hypervisor, tek bir fiziksel makinede birden fazla sanal makineyi çalıştırmak için kullanabileceğiniz bir yazılımdır.

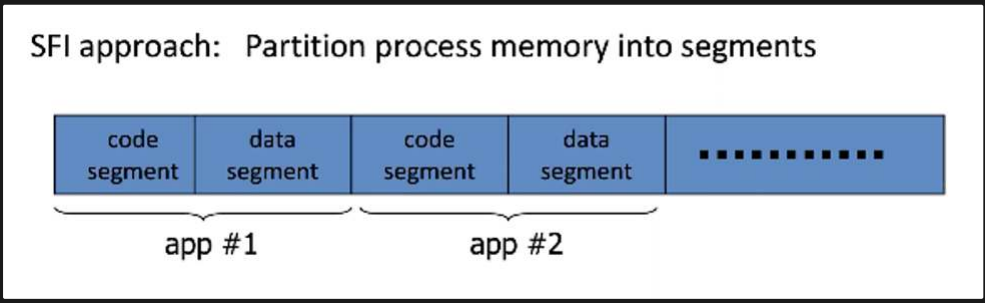


Host OS olmadığından daha iyi yönetiliyor.

Bir işletim sisteminin hypervisor üzerinde çalıştığını anlayabiliyoruz. Bazı malwareler veya normal uygulamalar vm'de çalışmayabilir. Malware fark edilmemiş olur böylece.

Web crawl ile malware detection yapabiliriz

Software Fault Isolation



Aynı adresteki treadleri bile ayırıyoruz, performansı kötü etkiliyor ama yapacak bir şey yok.

SFI Summary

- Performance
  - Usually good: mpeg_play, 4% slowdown
- Limitations of SFI: harder to implement on x86 :
  - variable length instructions: unclear where to put guards
  - few registers: can't dedicate three to SFI
  - many instructions affect memory: more guards needed

Isolation: summary

- Many sandboxing techniques: