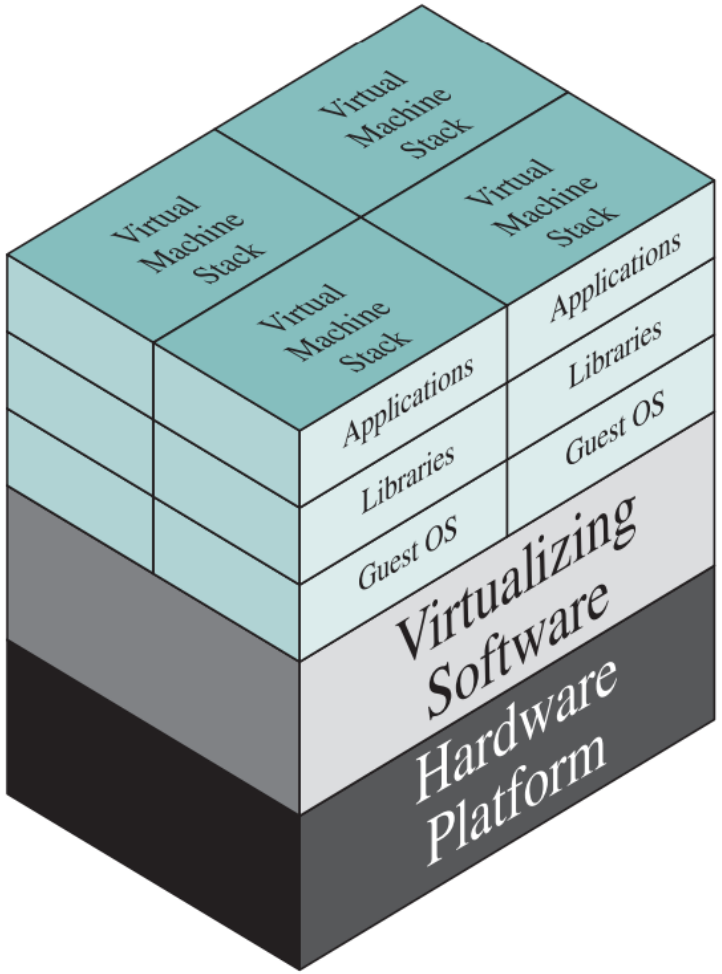
Virtualizacija i sigurnost

Virtualni strojevi (VM)

- Virtualizacija omogućuje jednom računalu ili poslužitelju istovremeno pokretanje više operacijskih sustava ili više sesija jednog OS-a
- Računalo sa softverom za virtualizaciju može ugostiti brojne aplikacije, uključujući one koje rade na različitim operativnim sustavima
- Operativni sustav domaćina može podržati brojna virtualna računala, od kojih svaki ima karakteristike određenog OS-a i, u nekim verzijama virtualizacije, karakteristike određene hardverske platforme
- Rješenje koje omogućuje virtualizaciju je virtual machine monitor (VMM) ili hipervizor
- Ovaj se softver nalazi između hardvera i VM-ova koji djeluje kao broker resursa
-

Koncept virtualnog računala



Ključni razlozi za korištenje virtualizacije

- Ključne razloge zbog kojih organizacije koriste virtualizaciju možemo sažeti na sljedeći način:
- Zastarjeli hardver
  - Aplikacije izgrađene za zastarjeli hardver i dalje se mogu pokretati virtualizacijom, omogućujući njegovo gašenje
- Brza implementacija
  - Novo virtualno računalo može se napraviti za par minuta
- Svestranost
  - Korištenje hardvera može se optimizirati maksimiziranjem broja vrsta aplikacija koje jedno računalo može izvršavati
- Konsolidacija
  - Resurs velikog kapaciteta ili velike brzine možgu se učinkovitije koristiti dijeljenjem među više aplikacija istovremeno
- Agregiranje
  - Virtualizacija olakšava kombiniranje više resursa u jedan virtualni resurs, primjerice u slučaju virtualizacije prostora za pohranu
- Dinamika
  - Hardverski resursi mogu se lako dodijeliti na dinamičan način, povećavajući uravnoteženje opterećenja i toleranciju na kvarove
- Jednostavnost upravljanja
  - Virtualni strojevi olakšavaju implementaciju i testiranje softvera
- Povećana dostupnost
  - Virtualni poslužitelji mogu se grupirati zajedno kako bi stvorili skupove računalnih resursa

Hipervizori

Virtualno računalo softverski je konstrukt koji oponaša karakteristike fizičkog poslužitelja

Konfiguriran je s određenim brojem procesora, određenom količinom RAM-a, resursima za pohranu i povezivanjem putem mrežnih priključaka

Nakon stvaranja VM-a može se uključiti poput fizičkog poslužitelja, podići operaciski sustav i softverska rješenja i koristiti u maniri fizičkog poslužitelja

Za razliku od fizičkog poslužitelja, ovaj virtualni poslužitelj vidi samo resurse s kojima je konfiguriran, a ne sve resurse samog fizičkog računala

Hipervizor olakšava prijevod U/I s virtualnog stroja na uređaje fizičkog poslužitelja i natrag

Hipervizori

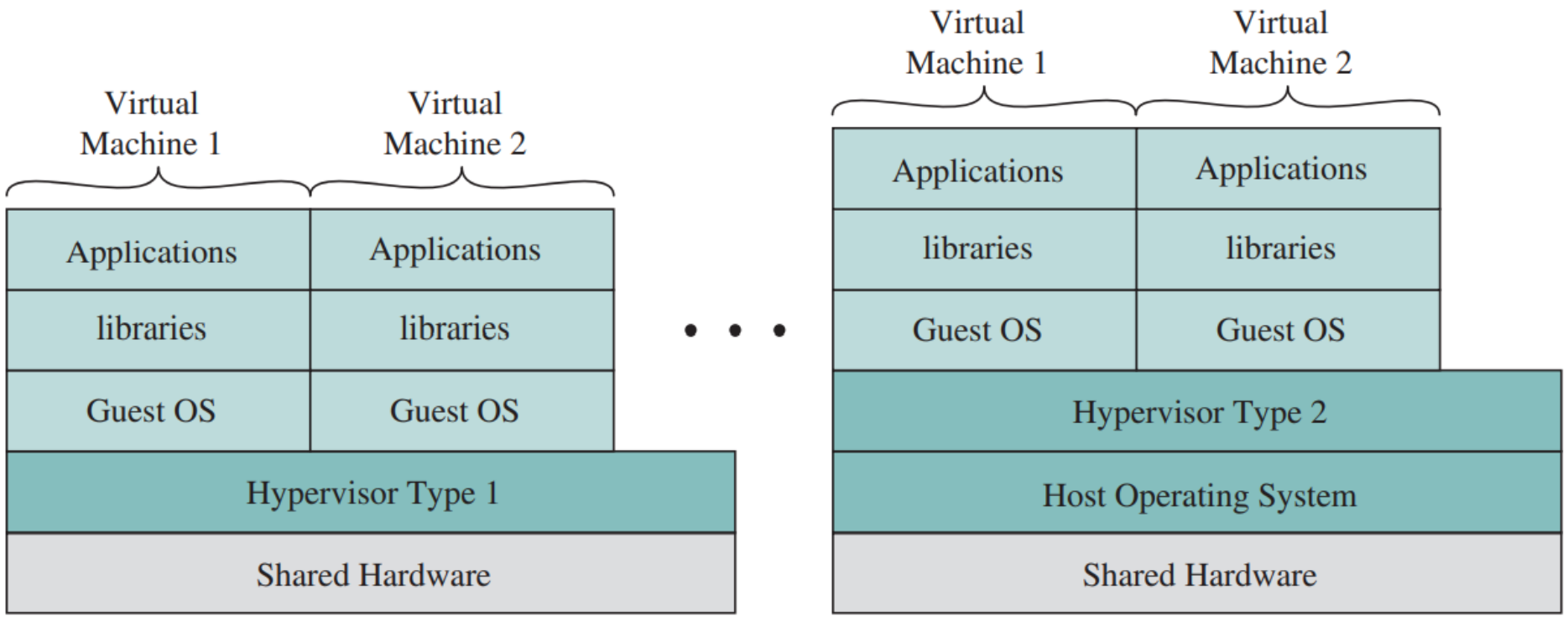
VM instanca definirana je u datotekama:

- Konfiguracijska datoteka opisuje atribute virtualnog računala
- Sadrži definiciju poslužitelja, koliko je virtualnih procesora (vCPU) dodijeljeno ovom virtualnom stroju, koliko je RAM-a dodijeljeno, kojim ulazno-izlaznim uređajima VM ima pristup, koliko kartica mrežnog sučelja (NIC) ima na virtualnom poslužitelju i još mnogo toga
- Također opisuje pohranu kojoj VM može pristupiti
- Kada je virtualno računalo uključeno ili instancirano, stvaraju se dodatne datoteke za zapisivanje, za stranice memorije i druge funkcije
- Budući da su virtualna računala već datoteke, njihovim kopiranjem dobijamo sigurnosnu kopiju podataka i kopiju cijelog poslužitelja, uključujući operativni sustav, aplikacije i samu konfiguraciju hardvera

Funkcije hipervizora

- Glavne funkcije koje obavlja hipervizor su:
  - Upravljanje izvršavanjem VM-ova
  - Emulacija uređaja i kontrola pristupa
  - Izvršavanje povlaštenih operacija od strane hipervizora za gostujuće VM-ove
  - Upravljanje VM-ovima
  - Administracija hipervizorske platforme i hipervizorskog softvera

Hipervizori tipa 1 i tipa 2



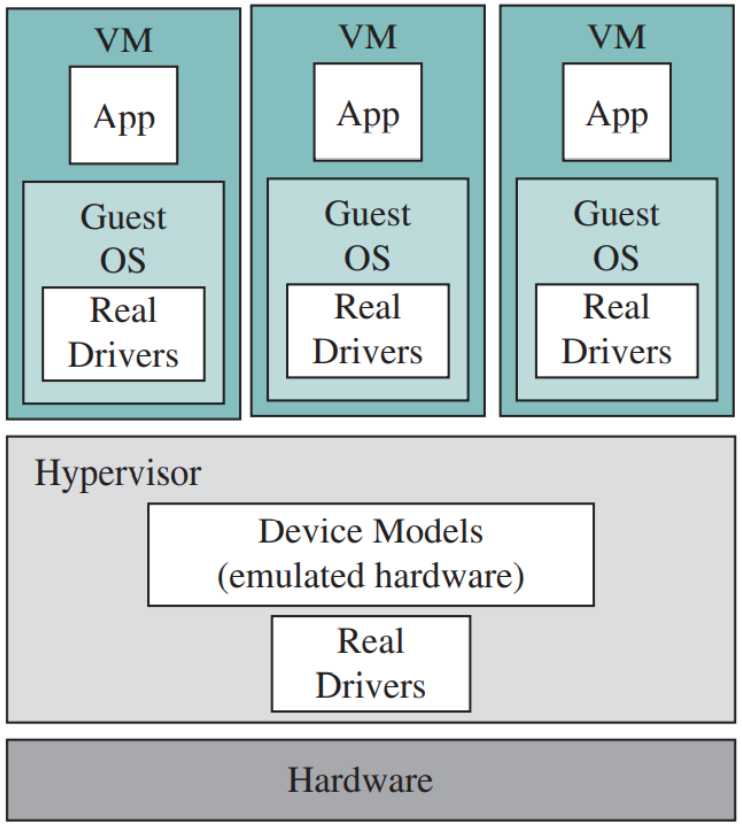
(a) Type 1 Hypervisor

(b) Type 2 Hypervisor

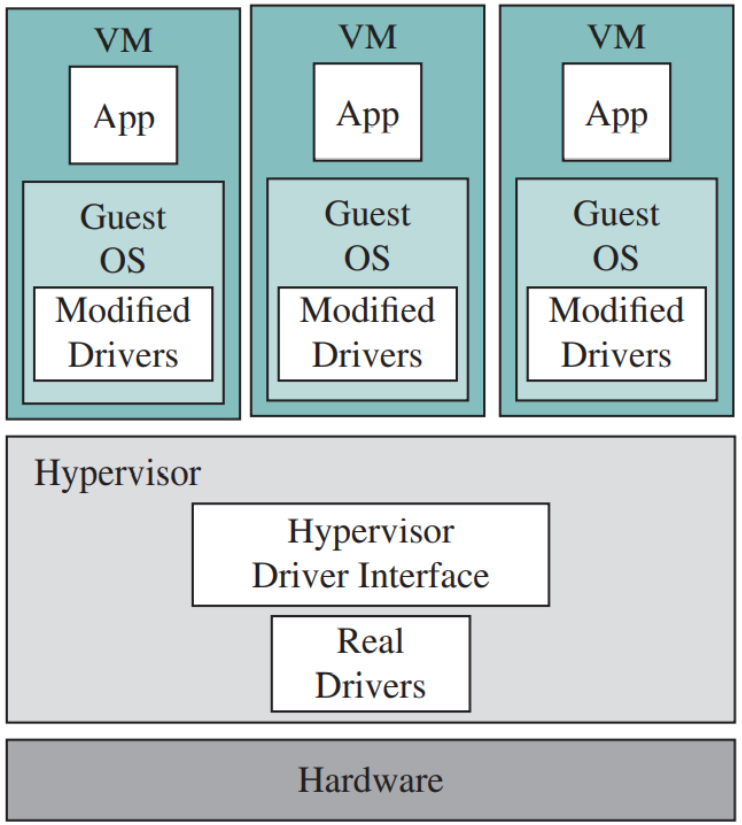
Paravirtualizacija

- Tehnika virtualizacije potpomognuta softverom koja koristi specijalizirane API-je za povezivanje virtualnih strojeva s hipervizorom kako bi optimizirala njihove performance
- Operativni sustav u virtualnom stroju, Linux ili Microsoft Windows, ima specijaliziranu podršku za paravirtualizaciju kao dio jezgre, kao i specifične upravljačke programe paravirtualizacije koji omogućuju OS-u i hipervizoru učinkovitiji suradnju
- Podrška se nudi kao dio mnogih općih Linux distribucija od 2008. godine

Paravirtualizacija



(a) Type 1 Hypervisor



(b) Paravirtualized Type 1 Hypervisor with Paravirtualized Guest OSs

Hardverski potpomognuta virtualizacija

- Proizvođači procesora AMD i Intel dodali su funkcionalnost svojim procesorima kako bi poboljšali performanse s hipervizorima
- AMD-V i Intelov VT-x označavaju proširenja za virtualizaciju potpomognuta hardverom koja hipervizori mogu iskoristiti tijekom obrade
- Intelovi procesori nude dodatni skup uputa pod nazivom Virtual Machine Extensions (VMX)
- Uz neke od ovih dadataka kao dio procesora, hipervizori više ne moraju izvršavati te funkcije

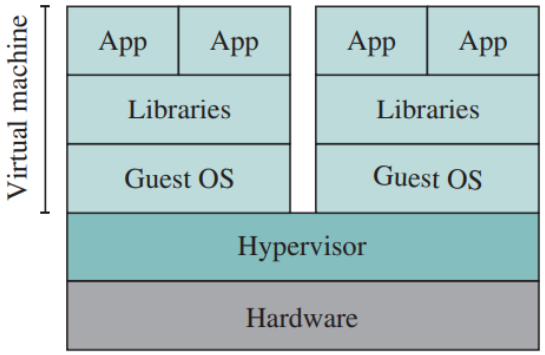
Virtualni uređaj

- Virtualni uređaj je samostalni softver koji se može distribuirati kao slika virtualnog stroja
- Sastoji se od zapakiranog skupa aplikacija i OS-a
- Neovisan je o hipervizoru ili arhitekturi procesora, a može se izvoditi na hipervizoru tipa 1 ili tipa 2
- Virtualni uređaji postaju de facto sredstvo distribucije softvera
- Nedavni i važan razvoj je sigurnosni virtualni uređaj (SVA)
  - SVA je sigurnosni alat koji obavlja funkciju praćenja i zaštite ostalih VM-ova i izvodi se izvan tih VM-ova
  - SVA dobiva svoju vidljivost u stanju VM-a, kao i mrežnog prometa između VM-ova, te između VM-ova i hipervizora putem API-ja
  - Prednosti SVA
    - Nije osjetljiv na propust u gostujućem OS-u
    - Neovisnan je o konfiguraciji virtualne mreže i ne mora se rekonfigurirati svaki put kada se konfiguracija virtualne mreže promijeni zbog migracije VM-ova

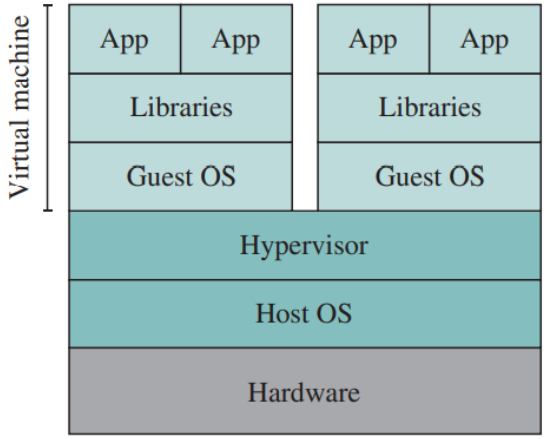
Virtualizacija kontejnera

- Virtualizacija kontejnera relativno je nedavni pristup virtualizaciji
  - U ovom pristupu softver, poznat kao virtualizacijski kontejner, radi na vrhu jezgre OS-a i pruža izolirano okruženje za izvršavanje aplikacija
  - Za razliku od VM-ova temeljenih na hipervizoru, kontejneri nemaju za cilj oponašanje fizičkih poslužitelja; umjesto toga, sve aplikacije u kontejnerima na glavnom računalu dijele zajedničku OS jezgru
  - Time se eliminiraju resursi potrebni za pokretanje zasebnog OS-a za svaku aplikaciju i mogu se uvelike smanjiti troškovi

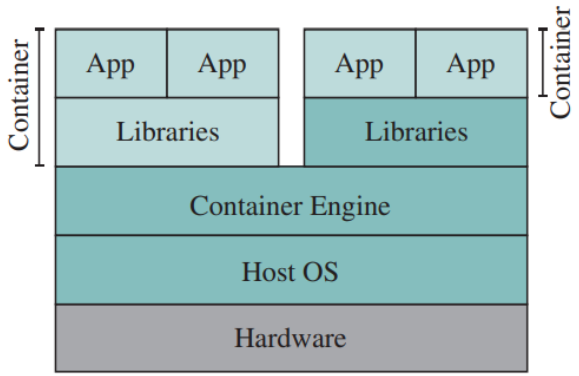
Usporedba virtualnih računala i kontejnera



(a) Type 1 Hypervisor

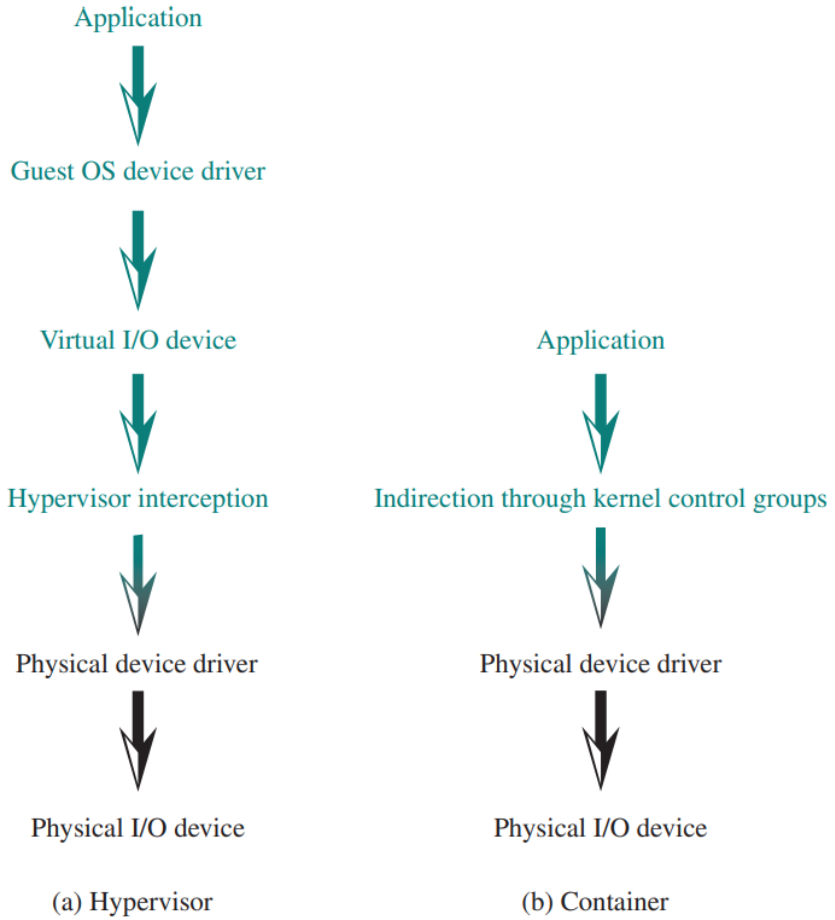


(b) Type 2 Hypervisor



(c) Container

Ulazno-izlaznu protok podataka putem hipervizora i kontejnera



Mikroservisi

- NIST definira mikroservis kao:

"osnovni element koji proizlazi iz arhitektonske razgradnje komponenti aplikacije
u labavo povezane obrasce koji se sastoje od samostalnih usluga
koje međusobno komuniciraju pomoću standardnog komunikacijskog protokola
i skupa dobro definiranih API-ja,
neovisno o bilo kojem dobavljaču, proizvodu ili tehnologiji"

Mikroservisi

Docker

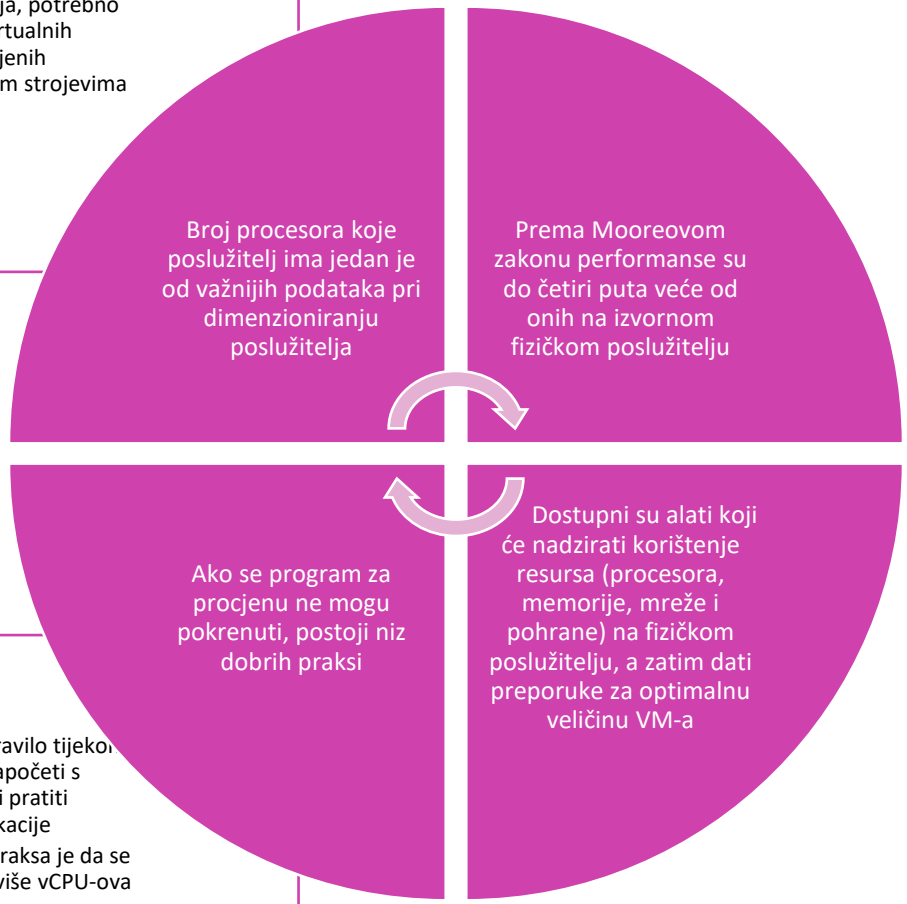
- Pruža jednostavniji i standardiziraniji način pokretanja kontejnera
- Jedan od razloga zašto je Docker popularniji u usporedbi s konkurentskim spremnicima je njegova sposobnost učitavanja slike spremnika na operativni sustav domaćina na jednostavan i brz način
- Docker kontejneri pohranjuju se u oblaku slike i korisnici ih pozivaju na izvršenje kada je to potrebno na jednostavan način
- Glavne komponente Dockera su:
  - Docker image
  - Docker client
  - Docker host
  - Docker engine
  - Docker machine
  - Docker registry
  - Docker hub

Izazovi s procesorom

- U virtualnom okruženju postoje dvije glavne strategije za pružanje procesorskih resursa:
  - Programsko oponašajte procesora
    - Primjeri ove metode su QEMU i Android Emulator u Android SDK-u
  - Dodjeljivanje segmenata vremena obrade na fizičkim procesorima (pCPU) virtualizacijskog poslužitelja virtualnim procesorima (vCPU) virtualnih strojeva smještenih na fizičkom poslužitelju
    - Tako većina hipervizora dodjeljuje procesorske resurse

Dodjela procesora

- Kada se aplikacije migriraju u virtualna okruženja, potrebno je odrediti broj virtualnih procesora dodijeljenih njihovim virtualnim strojevima



- Jedno osnovno pravilo tijekom kreiranja VM je započeti s jednim vCPU-om i pratiti performanse aplikacije
- Još jedna dobra praksa je da se ne dodjeljuje previše vCPU-ova

Prsten O

- Izvorni operativni sustavi upravljaju hardverom djelujući kao posrednik između zahtjeva za aplikacijskim kodom i hardvera
- Jedna od ključnih funkcija operativnog sustava je spriječiti zlonamjerne ili slučajne pozive sustava da ometaju aplikacije ili sam operativni sustav
- Zaštitni prstenovi opisuju razinu pristupa ili privilegiju unutar računalnog sustava, a mnogi operativni sustavi i arhitekture procesora iskorištavaju ovaj sigurnosni model
- Najpouzdaniji sloj često se naziva Ring 0 (nula) i mjesto je gdje jezgra operativnog sustava radi i može izravno komunicirati s hardverom
- Hipervizori rade u Ringu 0 upravljajući hardverskim pristupom za virtualna računala koje se na njemu izvršavaju

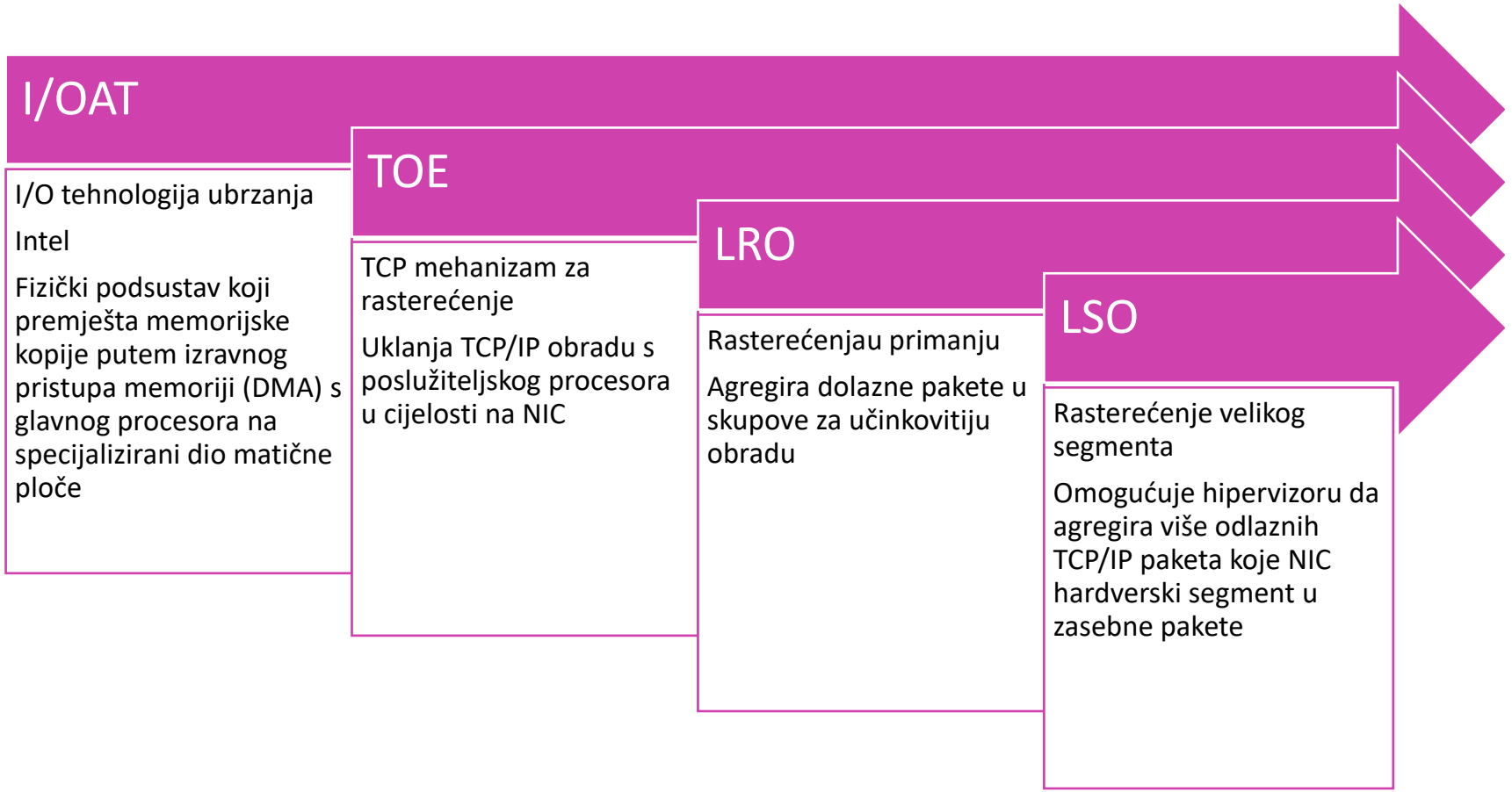
Upravljanje memorijom

- Budući da hypervisor upravlja dijeljenjem stranica, operativni sustavi virtualnog stroja nisu svjesni što se događa u fizičkom sustavu
- Baloni
  - Hipervizor aktivira upravljački program koji (virtualno) napuhuje i pritišće operacijski sustav gostiju za pohranu stranica na disk
  - Nakon što se stranice očiste, upravljački program balona se ispuhuje, a hipervizor može koristiti fizičku memoriju za druge VM-ove
  - Nadziranje memorije
  - Mogućnost dodjele više memorije nego što fizički postoji na hostu

Upravljanje ulazno-izlaznim operacijama

- An advantage of virtualizing the workload's I/O path enables hardware independence by abstracting vendor-specific drivers to more generalized versions that run on the hypervisor
- This abstraction enables:
  - Live migration, which is one of virtualization's greatest availability strengths
  - The sharing of aggregate resources, such as network paths
  - The memory overcommit capability is another benefit of virtualizing the I/O of a VM
  - The trade-off for this is that the hypervisor is managing all the traffic and requires processor overhead
    - This was an issue in the early days of virtualization, but now faster multicore processors and sophisticated hypervisors have addressed this concern

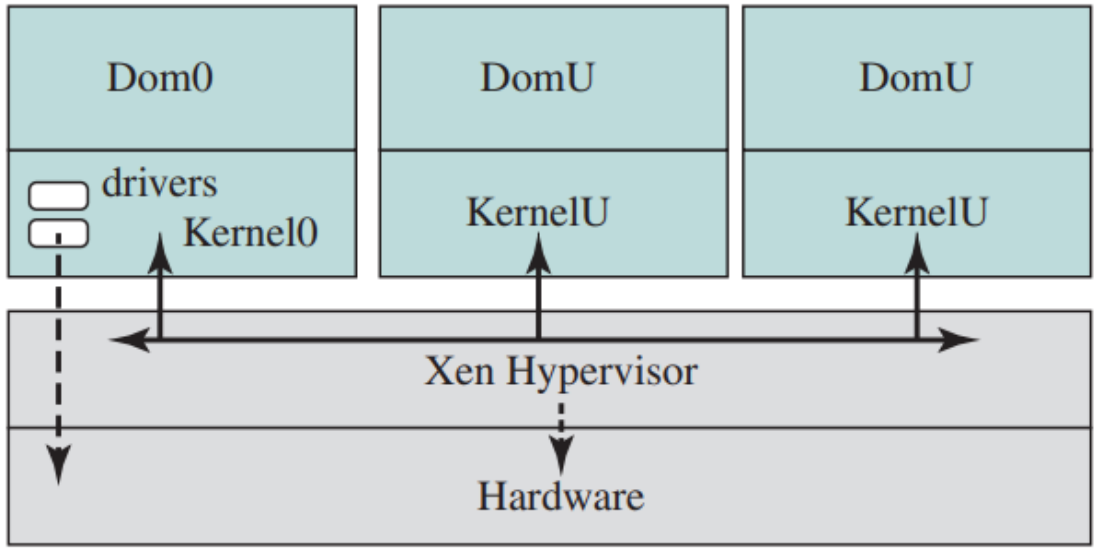
Tehnologije performansi

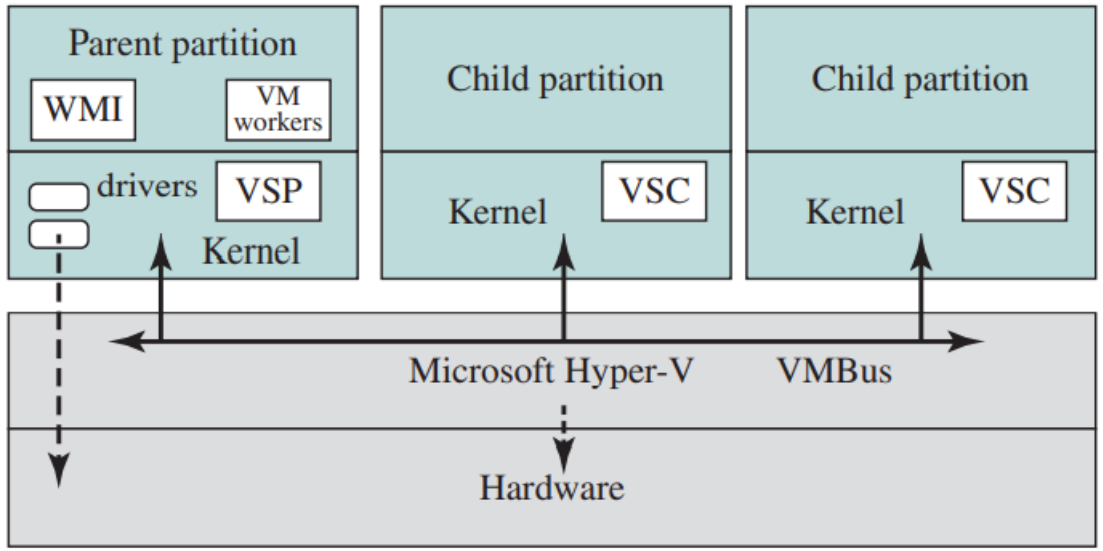


VMware ESXi

- Komercijalno dostupan hipervizor tvrtke VMware koji korisnicima pruža hipervizor tipa 1 za smještaj virtualnih strojeva na svojim poslužiteljima
- VMware je razvio svoja početna rješenja temeljena na x86 krajem 1990-ih i prvi su isporučili komercijalni proizvod na tržište
- Kontinuiranim inovacijama, zadržao se čvrsto na vrhu po tržišnom udjelu

Xen & Hyper-V


Dom0
DomU
DomU
drivers
Kernel0
KernelU
KernelU
Xen Hypervisor
Hardware


Parent partition
WMI
VM workers
drivers
VSP
Kernel
Child partition
Kernel
VSC
Child partition
Kernel
VSC
Microsoft Hyper-V
VMBus
Hardware

Java VM

- Cilj Java virtualnog stroja (JVM) je pružiti prostor za vrijeme izvođenja skupa Java koda za izvođenje na bilo kojem operativnom sustavu postavljenom na bilo kojoj hardverskoj platformi bez potrebe za promjenama koda kako bi se prilagodili različitim operativnim sustavima ili hardveru
- JVM može podržavati više dretvi
- „Napiši jedno, izvršavaj bilo gdje"

- JVM je opisan kao apstraktni računalni stroj koji se sastoji od:
  - Skup naredbi/uputa
  - Registar brojača programa
  - Stog za držanje varijabli i rezultata
  - Heep za pohranu podataka kod izvođenja i prikupljanju smeća
  - Područje metode za kod i konstante

Linux VServer

- Linux VServer je open-source, brz i lagan pristup implementaciji virtualnih strojeva na Linux poslužitelju
- Uključena je samo jedna kopija Linux jezgre
- VServer se sastoji od relativno skromne modifikacije jezgre plus mali skup alata OS
- VServer Linux jezgra podržava niz zasebnih virtualnih poslužitelja
- Jezgra upravlja svim sistemskim resursima i zadacima, uključujući zakazivanje procesa, memoriju, prostor na disku i vrijeme procesora

Arhitektura

- Svaki virtualni poslužitelj izoliran je od ostalih pomoću mogućnosti Linux jezgre
- Izolacija uključuje četiri elementa:
  - chroot
    - UNIX ili Linux naredba kako bi korijenski direktorij (/) postao nešto drugo osim njegovog zadanog za vijek trajanja trenutnog procesa
    - Ova naredba pruža izolaciju datotečnog sustava
  - chcontext
    - Linux uslužni program koji dodjeljuje novi sigurnosni kontekst i izvršava naredbe u tom kontekstu
    - Svaki virtualni poslužitelj ima vlastiti kontekst izvršavanja koji omogućuje izolaciju procesa
  - chbind
    - Izvršava naredbu i zaključava dobiveni proces i njegovu djecu u korištenje određene IP adrese
    - Sistemski poziv omogućuje izolaciju mreže
  - capablities
    - Odnosi se na partcioniranje privilegije dostupne korijenskom korisniku
    - Svakom virtualnom poslužitelju može se dodijeliti ograničeni podskup privilegija korijenskog korisnika koji osigurava izolaciju korijena

Prijetnje pristupa sustavu

Uljezi

Maskirani

Pojedinac koji nije ovlašten koristiti računalo i koji zaobilazi kontrole pristupa sustavu kako bi iskoristio legitimni korisnički račun

Interni

Legitimni korisnik koji pristupa podacima, programima ili resursima za koje takav pristup nije odobren ili koji je ovlašten za takav pristup, ali zloupotrebljava svoje privilegije

Tajni korisnik

Pojedinac koji preuzima nadzornu kontrolu nad sustavom i koristi tu kontrolu za izbjegavanje nadzora i kontrola pristupa

Zlonamjerni softver

- Programi koji iskorištavaju ranjivosti u računalnim sustavima
- Može se podijeliti u dvije kategorije:
  - Parazitski
    - Fragmenti programa koji ne mogu postojati neovisno o nekom stvarnom aplikacijskom programu, uslužnom programu ili sistemskom program
    - Primjeri su virusi i logičke bombe
  - Nezavisan
    - Samostalni programi koje operacijski sustav može pokrenuti
    - Primjeri su crvi i bot programi

Protumjere

- RFC 4949 (Internet Security Glossary) definira otkrivanje upada kao sigurnosnu uslugu koja prati i analizira sistemske događaje u svrhu pronalaženja i pružanja upozorenja u stvarnom vremenu na pokušaje neovlaštenog pristupa sistemskim resursima
- Sustavi za otkrivanje upada (IDS) mogu se klasificirati kao:
  - Host-based IDS
    - Prati karakteristike jednog računala i događaje koji se događaju unutar njega otkrivanja sumnjivih aktivnosti
  - Network-based IDS
    - Nadzire mrežni promet za određene mrežne segmente i analizira mrežne, transportne i aplikacijske protokole kako bi identificirao sumnjive aktivnosti

Komponente IDS/IPS sustava

Autentikacija

- U većini konteksta računalne sigurnosti autentikacija korisnika je primarna linija obrane
- RFC 4949 definira autentikaciju korisnika kao postupak provjere identiteta
- Postupak provjere autentičnosti sastoji se od dva koraka:
  - Korak identifikacije
    - Predstavljanje identifikatora sigurnosnom sustavu
  - Korak provjere
    - Predstavljanje ili generiranje informacija o provjeri autentičnosti koje potvrđuju povezivanje između entiteta i identifikatora

Sredstva provjere autentičnosti

- Nešto što pojedinac zna
  - Primjeri uključuju lozinku, osobni identifikacijski broj (PIN) ili odgovore na unaprijed dogovoreni skup pitanja
- Nešto što pojedinac posjeduje
  - Primjeri uključuju elektroničke kartice s ključevima i pametne kartice
  - Naziva se token
- Nešto što pojedinac jest (statička biometrija)
  - Primjeri uključuju prepoznavanje otiskom prsta, mrežnicom i licem
- Nešto što pojedinac radi (dinamička biometrija)
  - Primjeri uključuju prepoznavanje po uzorku glasa, karakteristikama rukopisa i ritmu tipkanja

Kontrola pristupa

- Provodi sigurnosna pravila koja određuju tko ili što može imati pristup svakom određenom resursu sustava i vrsti pristupa koja je dopuštena u svakoj instanci
- Posredovanje između resursa korisnika i sustava, kao što su aplikacije, operacijski sustavi, vatrozidi, usmjerivači, datoteke i baze podataka
- Administrator sigurnosti održava bazu podataka autorizacije koja određuje koja vrsta pristupa kojim je resursima dopuštena za ovog korisnika
  - Funkcija kontrole pristupa pregledava ovu bazu podataka kako bi odredila hoće li odobriti pristup
- Funkcija nadzora nadzire i vodi evidenciju o pristupu korisnika sistemskim resursima

Vatrozidi

Ciljevi dizajna:

1. Sav promet iznutra prema van i obrnuto, mora proći kroz vatrozid. To se postiže fizičkim blokiranjem svih pristupa lokalnoj mreži, osim putem vatrozida
2. Bit će dopušten prolazak samo ovlaštenog prometa, kako je definirano lokalnom sigurnosnom politikom. Koriste se različite vrste vatrozida koji provode različite vrste sigurnosnih pravila
3. Sam vatrozid je imun na upade. To podrazumijeva uporabu pouzdanog sustava sa osiguranim operacijskim sustavom.

Napadi prelijevanja međuspremnika - *Buffer Overflow*

- Poznat i kao prekoračenje međuspremnika
- Definirano u NIST kao:

> “Uvjet na sučelju pod kojim se u međuspremnik ili područje držanja podataka može smjestiti više ulaza od dodijeljenog kapaciteta, prebrisivanjem drugih informacija. Napadači iskorištavaju takvo stanje kako bi srušili sustav ili umetnuli posebno izrađeni kod koji im omogućuje da steknu kontrolu nad sustavom”

Iskorištavanje prelijevanja međuspremnika

- Da bi se iskoristila bilo koja vrsta prelijevanja međuspremnika, napadaču je potrebno:
  - Prepoznavanje ranjivosti prelijevanja međuspremnika u nekom programu koja se može pokrenuti pomoću vanjskih podataka pod kontrolom napadača
  - Razumjeti kako će se taj međuspremnik pohraniti u memoriju procesa, a time i potencijal za oštećenje susjednih memorijskih mjesta i potencijalno mijenjanje tijeka izvršavanja programa

Obrana

- Protumjere se mogu široko svrstati u dvije kategorije:
  1. Obrana kod kompajliranja, čiji je cilj očvrsnuti programe kako bi se oduprli napadima
  2. Obrana kod izvođenja, čiji je cilj otkrivanje i prekid napada u izvršavanju programa

Tehnike obrana kod kompajliranja

- Izbor programskog jezika
  - Jedna od mogućnosti je pisanje programa u modernim programskim jezicima visoke razine
- Jezična proširenja i korištenje sigurnih biblioteka
  - Libsafe je primjer koji implementira standardnu semantiku, ali uključuje dodatne provjere kako bi se osiguralo da se operacije kopiranja ne protežu izvan lokalnog memoriskog prostora u okviru stoga
- Tehnike sigurnog kodiranja
  - Programeri moraju pregledati kod i maknuti sve nesigurne dijelove koda
- Mehanizmi zaštite stoga
  - Stackguard, jedan od najpoznatijih zaštitnih mehanizama, je proširenje kompajlera zbirke GNU Compile Collection (GCC) koje umeće dodatni kod za funkcije unosa

Tehnike obrana kod izvođenja

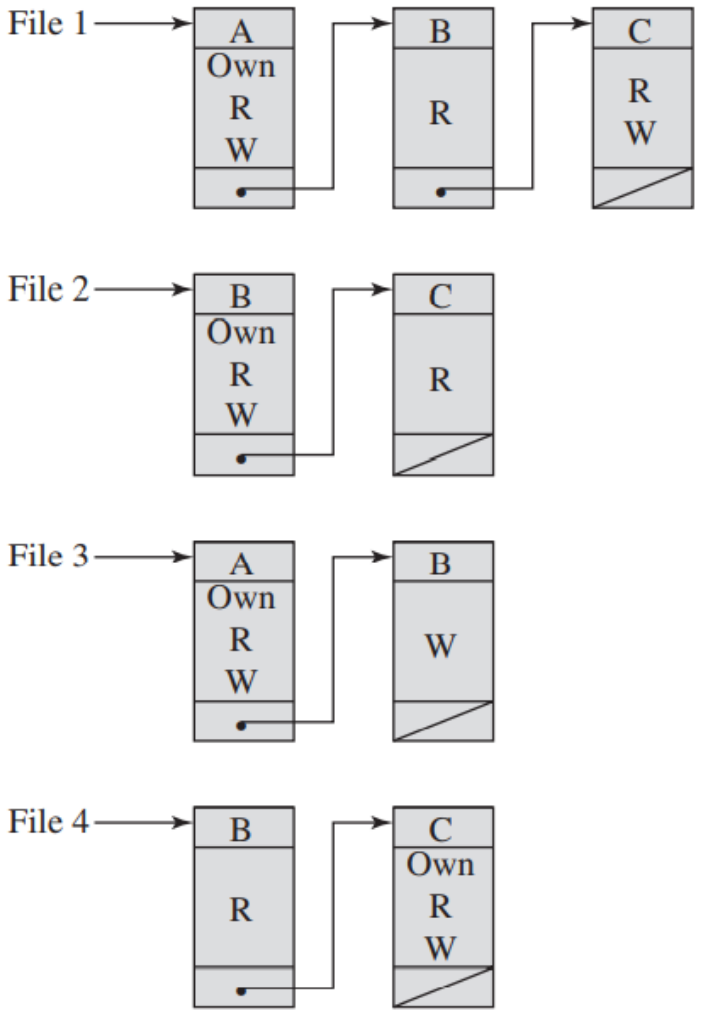
- Izvršna zaštita adresnog prostora
  - Moguća obrana je blokiranje izvršavanja koda na stogu i heap-u
- Randomizacija adresnog prostora
  - Premještanje područja memorije stoga za megabajt ili slično ima minimalan utjecaj na većinu programa, ali predviđanje ciljane adrese međuspremnika čini gotovo nemogućim
  - Druga tehnika je korištenje sigurnosnog proširenja koje randomizira redoslijed učitavanja standardnih biblioteka od strane programa i njihovih adresa
- Zaštičene stranice
  - Postavljaju se granice između raspona adresa koje se koriste za svaku komponentu adresnog prostora
  - Te su praznine ili zaštitne stranice označene u MMU-u kao neispravne adrese, a svaki pokušaj pristupa rezultira prekidom procesa
  - Daljnje proširenje postavlja zaštitne stranice između okvira stogova ili između različitih alokacija na heepu

Kontrola pristupa datotečnom sustavu

- Identificira korisnika u sustavu
- Povezano sa svakim korisnikom može postojati profil koji određuje dopuštene operacije i pristup datotekama
- Operativni sustav tada može provoditi pravila na temelju korisničkog profila
- Sustavi upravljanja bazom podataka moraju kontrolirati pristup određenim zapisima ili čak dijelovima zapisa
  - Odluka o pristupu ne ovisi samo o identitetu korisnika, već i o određenim dijelovima podataka kojima se pristupa

Matrica pristupa

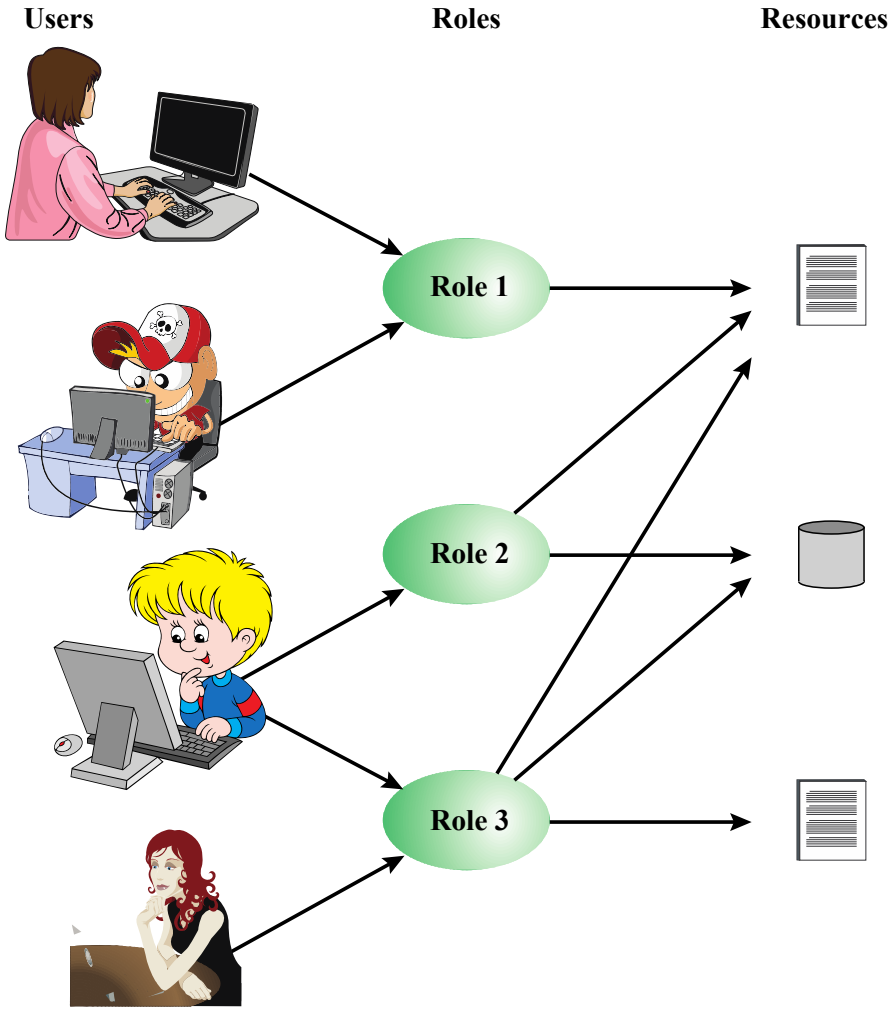
		Objects			
		File 1	File 2	File 3	File 4
Subjects	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write



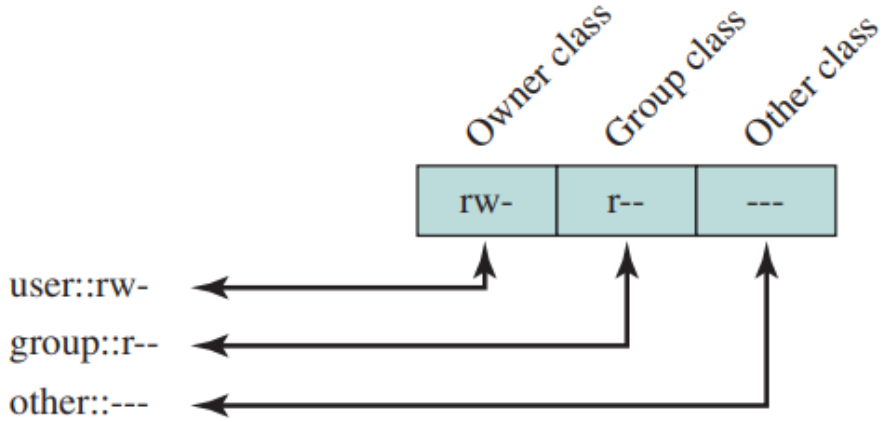
Pravila kontrole pristupa

- Politika kontrole pristupa diktira koje su vrste pristupa dopuštene, pod kojim okolnostima i od koga
- Pravila kontrole pristupa obično su grupirana u sljedeće kategorije:
  - Diskrecijska kontrola pristupa (DAC)
    - Kontrolira pristup na temelju identiteta podnositelja zahtjeva i pravila pristupa u kojima se navodi što podnositelji zahtjeva smiju učiniti
  - Obvezna kontrola pristupa (MAC)
    - Kontrolira pristup na temelju usporedbe sigurnosnih oznaka sa sigurnosnim provjerama
  - Kontrola pristupa temeljena na ulogama (RBAC)
    - Kontrolira pristup na temelju uloga koje korisnici imaju unutar sustava i pravila u kojima se navodi koji su pristupi dopušteni korisnicima u danim ulogama
  - Kontrola pristupa temeljena na atributima (ABAC)
    - Kontrolira pristup na temelju atributa korisnika, resursa kojem treba pristupiti i trenutačnih uvjeta zaštite okoliša

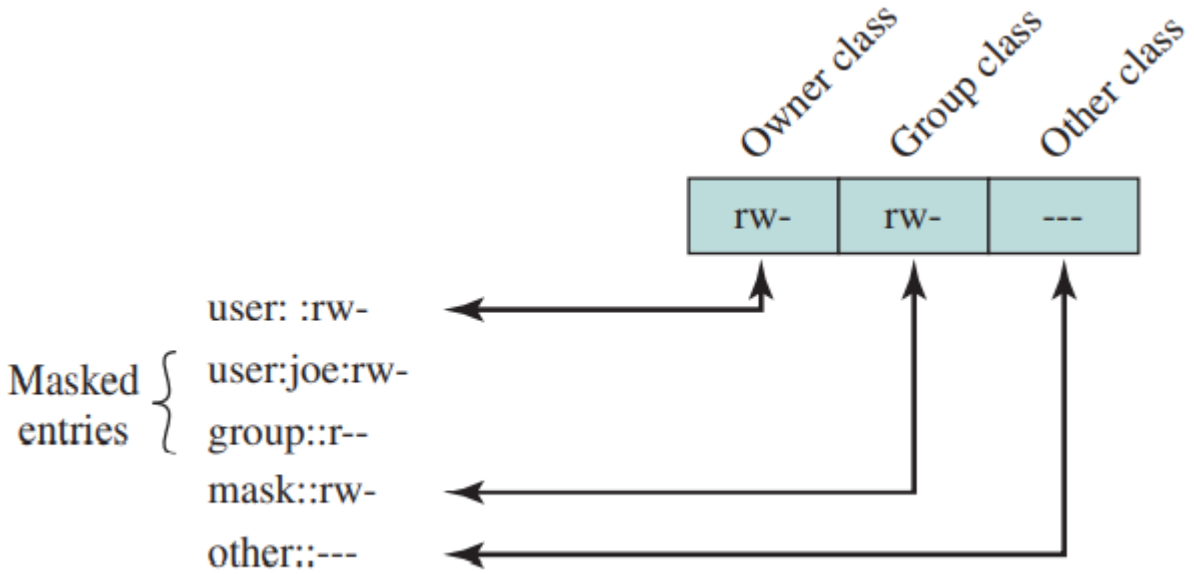
Korisnici, uloge i resursi



UNIX kontrola pristupa datoteci



(a) Traditional UNIX approach (minimal access control list)



(b) Extended access control list

Zaštita operacijskih sustava

- Osnovni koraci za zaštitu operacijskog sustava:
  - Instaliranje i zakrpa operacijskog sustava
  - Očvrsnite i konfigurirajte operativni sustav tako da adekvatno odgovori na utvrđene sigurnosne potrebe sustava na sljedeći način:
    - Uklanjanje nepotrebnih usluga, aplikacija i protokola
    - Konfiguriranje korisnika, grupa i dozvola
    - Konfiguriranje kontrola resursa
  - Instalacija i konfiguracija dodatnih sigurnosne kontrole, kao što su antituvirusni programi, vatrozidi i sustavi za otkrivanje upada (IDS)
  - Testirajte sigurnost operacijskog sustava kako biste osigurali da poduzeti koraci odgovaraju sigurnosnim potrebama

Instalacija operacijskog sustava: Početno instaliranje i nadogradnje

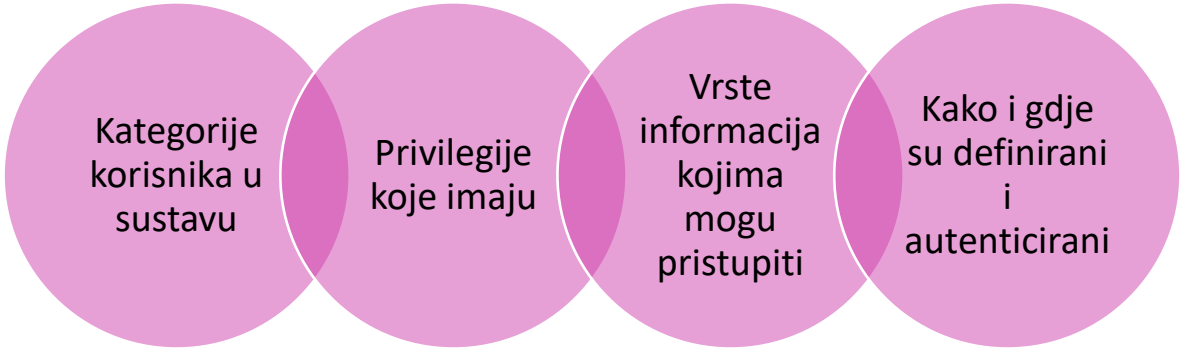
- Sigurnost sustava započinje instalacijom operativnog sustava
- U idealnom slučaju novi sustavi trebali bi biti instaliran na zaštićenoj mreži
- Početna instalacija trebala bi se sastojati od minimuma potrebnog za željeni sustav, s dodatnim softverskim paketima koji su uključeni samo ako su potrebni za funkciju sustava
- Cjelokupni postupak pokretanja također mora biti osiguran
- Također je potrebna njega pri odabiru i instalaciji bilo kojeg dodatnog koda upravljačkog programa uređaja, jer se to izvršava s privilegijama pune razine jezgre, ali ga često isporučuje treća strana

Uklanjanje nepotrebnih servisa, aplikacija i protokola

- Proces planiranja sustava trebao bi identificirati što je zapravo potrebno za određeni sustav kako bi se osigurala odgovarajuća razina funkcionalnosti, uz uklanjanje softvera koji nije potreban
- Prilikom izvođenja početne instalacije ne smiju se koristiti isporučene zadane postavke, već bi instalaciju trebalo prilagoditi tako da se instaliraju samo potrebni paketi
- Mnogi vodiči za povečanje sigurnosti pružaju popise usluga, aplikacija i protokola koji se ne bi trebali instalirati ako to nije potrebno
- Uklanjanje ili onemogućavanje neželjenih i nepotrebnih aplikacija
  - Ako napadač uspije dobiti određeni pristup sustavu, onemogućeni softver mogao bi se ponovno omogućiti i koristiti za daljnje ugrožavanje sustava
  - Bolje je za sigurnost ako neželjeni softver nije instaliran, pa stoga uopće nije dostupan za upotrebu

Konfiguriranje korisnika, grupa i autentikacije

- Proces planiranja sustava trebao bi uzeti u obzir:



- Ograničite veće ovlasti samo na one korisnike koji ih zahtijevaju
- U ovoj fazi trebalo bi osigurati sve zadane račune uključene u instalaciju sustava
- One račune koji nisu potrebni trebalo bi ukloniti ili barem onemogućiti
- Sistemski računi koji upravljaju uslugama u sustavu trebali bi biti postavljeni tako da se ne mogu koristiti za interaktivne prijave
- Sve lozinke instalirane prema zadanim postavkama treba promijeniti u nove vrijednosti s odgovarajućom sigurnošću
- Konfigurirana su sva pravila koja se primjenjuju na vjerodajnice za provjeru autentičnosti i sigurnost lozinke

Konfiguriranje kontrola resursa

- Nakon što se definiraju korisnici i njihove pridružene grupe, na podatke i resurse mogu se postaviti odgovarajuće dozvole koje odgovaraju navedenom pravilu
- To može biti ograničavanje korisnika koji mogu izvršavati neke programe ili ograničavanje korisnika koji mogu čitati ili pisati podatke u određenim mapama
- Mnogi vodiči za povečanje sigurnosti pružaju popise preporučenih promjena zadane konfiguracije pristupa radi poboljšanja sigurnosti

Instalacija dodatnih sigurnosnih kontrola

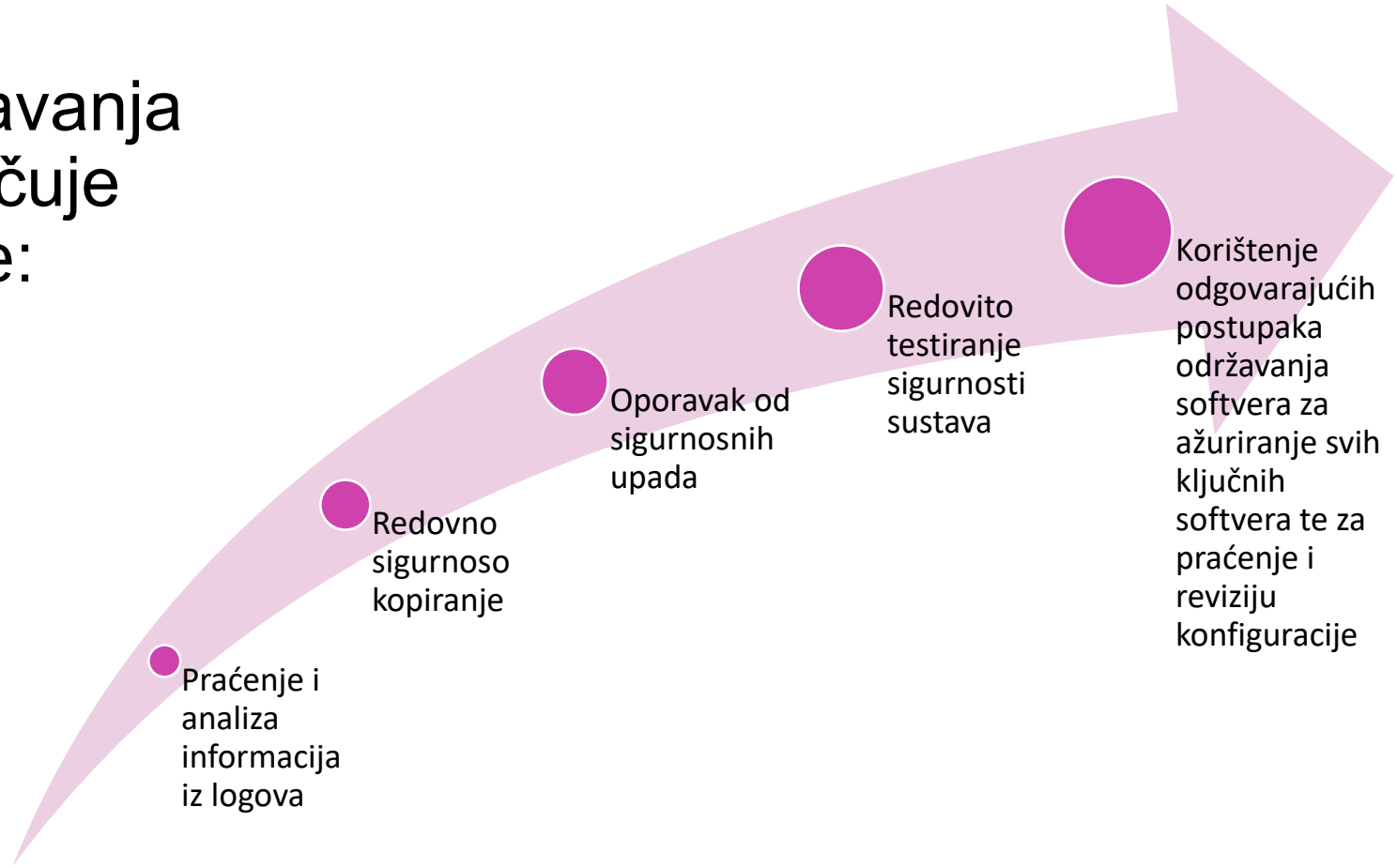
- Daljnje poboljšanje sigurnosti može biti moguće instaliranjem i konfiguriranjem dodatnih sigurnosnih alata kao što su antivirusni softver, vatrozid, IDS/IPS softver ili popis dopuštenih aplikacija
- Neki od njih mogu se isporučiti kao dio instalacije operativnih sustava, ali nisu konfigurirani i omogućeni prema zadanim postavkama
- S obzirom na raširenu učestalost zlonamjernog softvera, odgovarajući antivirusna je ključna sigurnosna komponenta
- IDS i IPS softver mogu uključivati dodatne mehanizme kao što su praćenje prometa ili provjera integriteta datoteka kako bi identificirali i sprječili neke vrste napada
- White-list aplikacije ograničavaju programe koji se mogu izvršavati u sustavu samo na one na eksplicitnom popisu

Testiranje sigurnosti sustava

- Posljednji korak u procesu početnog osiguranja osnovnog operativnog sustava je sigurnosno testiranje
- Cilj je osigurati ispravnu provedbu prethodnih koraka sigurnosne konfiguracije i identificirati sve moguće ranjivosti
- Prikladni kontrolni popisi uključeni su u mnoge vodiče za povečanje sigurnosit
- Postoje i programi posebno dizajnirani za pregled sustava kako bi se osiguralo da sustav zadovoljava osnovne sigurnosne zahtjeve i za traženje poznatih ranjivosti i loših konfiguracijskih praksi
- To treba učiniti nakon početnog osiguranja sustava i povremeno ponavljati kao dio održavanja sigurnosti sustava

Održavanje sigurnosti

- Postupak održavanja sigurnosti uključuje sljedeće korake:



Dnevnički zapisi (logovi)

- Učinkovito vođenje dnevničkih zapisa pomaže osigurati da u slučaju upada ili kvara sustava administratori sustava mogu brže i točnije identificirati što se dogodilo i učinkovitije usmjeriti svoje napore prema sanaciji i oporavku
- Dnevničke zapise mogu generirati sustav, mreža i aplikacije
- Raspon prikupljenih podataka treba odrediti tijekom faze planiranja sustava
- Dnevnički zapisi mogu generirati značajne količine podataka pa je važno da imamo dovoljno prostora
- Odgovarajući sustav automatskog rotiranja i arhiviranja dnevnika treba biti konfiguriran tako se olakša upravljanje
- Preferira se eki oblik automatizirane analize jer je vjerojatnije da će identificirati abnormalne aktivnosti
  - Ručna analiza je zamorna i nije pouzdano sredstvo za otkrivanje štetnih događaja

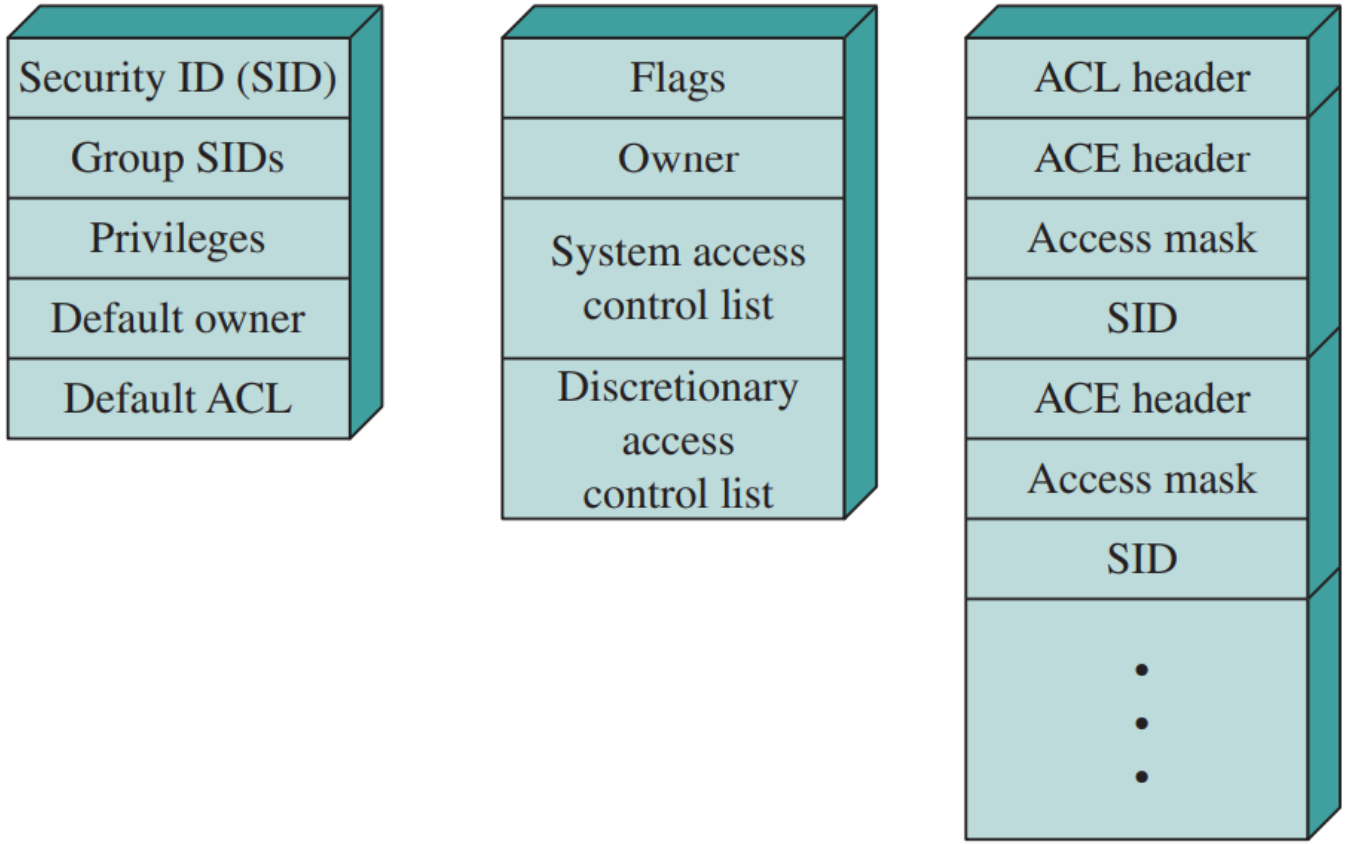
Sigurnosno kopiranje i arhiviranje

- Izvođenje redovitih sigurnosnih kopija podataka na sustavu još je jedna kritična kontrola koja pomaže u održavanju integriteta sustava i korisničkih podataka
- Potrebe i politike koje se odnose na sigurnosno kopiranje i arhiviranje trebalo bi odrediti tijekom faze planiranja sustava
  - Ključne odluke uključuju treba li kopije čuvati na internetu ili izvan njega te treba li kopije pohraniti lokalno ili ih prenijeti na udaljeno mjesto
- Sigurnosna kopija
  - Postupak izrade kopija podataka u redovitim intervalima, omogućujući povrat izgubljenih ili oštećenih podataka u relativno kratkim vremenskim razdobljima od nekoliko sati do nekoliko tjedana
- Arhiviranje
  - Postupak čuvanja kopija podataka tijekom duljeg vremenskog razdoblja, mjeseci ili godina, kako bi se ispunili zakonski i operativni zahtjevi za pristup prošlim podacima

Shema kontrole pristupa

- Kada se korisnik prijavi u sustav Windows, za provjeru autentičnosti korisnika koristi se shema imena/lozinke
- Ako je prijava prihvaćena, stvara se proces za korisnika i s tim procesom povezan je pristupni token
  - Pristupni token uključuje sigurnosni ID (SID) koji je identifikator po kojem je ovaj korisnik poznat sustavu u svrhu sigurnosti
  - Token sadrži i SID-ove za sigurnosne grupe kojima korisnik pripada
- Pristupni token služi u dvije svrhe:
  - Drži sve potrebne sigurnosne informacije zajedno kako bi se ubrzala validacija pristupa
  - Omogućuje svakom procesu izmjenu svojih sigurnosnih karakteristika na ograničene načine bez utjecaja na druge procese koji se izvode u ime korisnika

Sigurnosne strukture Windows OS-a



Sažetak

- Koncepti virtualnih računala
- Hipervizori
- Izazovi s procesorom
- Upravljanje memorijom
- Upravljanje ulaznim/izlaznim pozivima
- VMware ESXi
- Microsoft Hyper-V i Xen varijante
- Java VM
- Linux VServer arhitektura virtualnog računala
- Uljezi i zlonamjerni softver
- Prelijevanje međuspremnika
- Kontrola pristupa
- UNIX kontrola pristupa
- Zaštita operacijskih sustava
- Održavanjesigurnosti
- Sigurnost Windows OS-a
-

Thank you for your attention!