BEZPEČNOST IS P7

2008-04-03

ARCHITEKTURA MICROSOFT WINDOWS:

Základní struktura systému:

✓ **Kernel Mode:**
- Jádro, režim jádra (x86 ring 0)
- Neprocesový kód poskytující služby procesům
- Může cokoliv – nebezpečné ovladače
- Veškeré požadavky na hardware musí jít tudy
- Správa virtuální paměti, bezpečnost, souborové systémy, plug'n'play, plánování procesů, okna, …

✓ **User Mode:**
- Uživatelský režim (x86 ring 3)
- Všechno běží jako proces
- Veškeré akce podléhají rozhodnutí jádra

Kernel Mode:

✓ **Správa procesů:**
- Jádro startuje a ukončuje procesy, vytváří thready a umožňuje je zabezpečovat

✓ **Správa virtuální paměti:**
- Přidělování (alokace) paměti pro procesy
- Odkládání nepotřebných stránek do PAGEFILE.SYS
- Stránkování aktuálně potřebných stránek z odkládacího souboru
- Zavádění spustitelných souborů do paměti

✓ **Plug and play (PnP):**
- Detekce zařízení, inicializace příslušných ovladačů
- Dynamické nahrávání ovladačů za běhu

✓ **Ovladače zařízení – .SYS:**
- Každé zařízení musí mít svůj ovladač
- Možnost jejich filtrování (více driverů „na sobě")
- Automatické instalace driver.cab
- Nebezpečí zhroucení systému (WHCL podpisy)

✓ **Instalovatelné souborové systémy (IFS):**
- Ovladače jádra které převádějí požadavky na práci se soubory přímo na fyzické čtení/zápis z diskového ovladače
- Realizují formát dat na disku NTFS.SYS, FASTFAT.SYS, CDFS.SYS, …
- Umožňujjí instalaci filtrů – antiviry

User Mode:

✓ **Uživatelské a systémové procesy:**
- Bezpečnost na základě uživatelů
- Privátní paměťové prostory
- Odstíněný provoz více instancí stejného .EXE
- Prostředky (paměť, hardware, …) přidělovány jádrem
- Veškeré požadavky na hardware musí projít jádrem
- Jádro má absolutní kontrolu nad systémem

✓ **Paměťové prostory (2GB)** – procesy mohou mít na stejných adresách různá data

BEZPEČNOST OPERAČNÍCH SYSTÉMŮ:

Seznam oprávnění (Access Control List):

✓ Tentokrát je s každým objektem udržován seznam informací, které subjekty k němu mají jaká oprávnění

- ✓ Metoda umožňuje snadno přidělovat implicitní práva subjektům případně skupinám subjektů
- ✓ **Přístupová matice (Access Control Matrix):**
  - Řádky matice odpovídají jednotlivým subjektům, sloupce objektům
  - V políčku daném řádkem a sloupcem je záznam o úrovni oprávnění odpovídajícího subjektu k příslušnému objektu
  - Přístupová matice je zpravidla velmi velká záležitost, zhusta řídká

Způsobilost (Capability):

- ✓ **Způsobilost** budeme chápat jako nefalšovatelný token, jehož vlastnictví dává vlastníkovi specifická práva k danému objektu. Lze chápat jako lístek do kina.
- ✓ Jednou z metod zajštění nefalšovatelnosti je, že tokeny se nepředávají přímo subjektům, ale jsou udržovány v chráněné oblasti paměti, přístupné pouze systému
- ✓ Při přístupu k objektu tak systém zkontroluje existenci příslušného tokenu, tento postup lze urychlit tím, že zvlášť udržujeme seznam Způsobilostí právě běžícího procesu
- ✓ Výhodou metody je, že dovoluje definovat nové dosu neznámé způsoby používání objektů a přidělovat odpovídající oprávnění

Linux:

- ✓ **Jádro** Linuxu je 32bitový program ovládající hardware a poskytující aplikacím prostředky počítače (od sériových portů přes disky až po přidělování procesorového času a zabíjení neposedných programů). Jádro běží v chráněném režimu procesoru 386 a vyšším s prioritou 0, tj. může zapisovat a číst kamkoli do/z paměti a posílat cokoli na porty. Rozdíl proti aplikacím (platí i obecně v UNIXu) je ten, že aplikace tuto možnost nemají (zjednodušeně řečeno pokud nejsou nastavené/vykonávané v root režimu). Pokud chce aplikační program manipulovat s porty, musí být setuid root a vždy nejprve požádat jádro o povoleni přístupu k těm kterým portům.
- ✓ Normálně se totiž aplikacím porty počítače nemapuji. Setuid root znamená že při vykonávání aplikace se ji propůjčí efektivní práva vlastníka souboru - roota (např. jako uživatel bubo spustím program vgareset který je vlastněn rootem a má nastaven setuid bit). SETUID - něco jako SET User ID on execution.
- ✓ Výsledek je ten, že jádro si může dělat cokoli (předpokládá se smysluplná činnost), ale aplikace se musí podřídit nadvládě jádra. Pokud by si třeba program (proces) xxx usmyslel ze chce zapisovat přímo do registru videokarty a neměl na to právo, tak se nic nestane nebo ho jádro s klidem odstřelí.
- ✓ Z tohoto důvodu také nenajdete na UNIXových stanicích viry. Zde totiž něco jako přesměrování přerušení aplikací vůbec není. všechna hardwarová přerušení (IRQ 0-15) jsou zpracovávána jádrem, respektive vektory přerušení jsou pověšeny na vstupní body funkci obsluhujících hardware (TIMER, DMA, SERIAC, PARALEL, FPU, FDD, HDD etc.).
- ✓ Toliko teorie, jak to však vypadá **prakticky**:
  - Jádro se musí **spustit**. Tomu odpovídá:
    - Zavedení jádra z diskety (většinou používáno pro instalační program-BOOT disk)
    - Zavedení jádra z disku počítače (někdy přes lilo - jednoduchý bootmanager)
    - Zavedení jádra z DOSu (programem loadlin.exe)
  - Všechny varianty, ať nahrávají jádro odkudkoli, mají za **následek** to, že se jádro v paměti rozbalí (zabalené se musí vejít do dolního 1M paměti (resp. 640k) - omezení 8086), přepne do protected módu a spálí za sebou všechny mosty - zapomene na BIOS i DOS. Potom obvykle následuje část identifikace hardwaru (jaký CPU máte, velikost paměti, disky, sériové porty, paralelní porty, síťové karty, zvukové karty atd).

Windows 95:

- ✓ Systém Windows 95 (dále jen W95) není rozhodně tak bezpečný, jak se většina uživatelů domnívá. To platí jak v oblasti virových infekcí, tak i v oblasti neautorizovaného přístupu k datům či programům. Řada autorů zastává názor, celé W95 na tom nejsou o mnoho lépe než starý známý a nevyhovující MS DOS. Proto je zde poměrně široká oblast pro produkty, které mohou bezpečnost uvedeného systému zvýšit.
- ✓ Většina uživatelů a mnohdy bohužel i programátorů pokládá za W95 pouze to, co je skutečně "vidět", tj. "knihovny" jako Kernel, GDI a User. K tomu si někdy ještě uvědomí sem tam nějakou knihovnu navíc, ale to je vše. Takto pojatý systém W95 je vlastně jenom určitým „Nortonem Commanderem
- ✓ **VxD moduly:**
  - V prvním přiblížení, dostatečném pro naše účely, jsme si zanalyzovali architekturu operačního systému W95 a nyní můžeme pokročit dále v oblasti bezpečnosti OS. Zkratka VxD může někoho svou složitostí

zaskočit, ale vytváření modulů s tímto názvem není zas tak složité. Ve své podstatě jen potřebujeme mít k dispozici vhodný assembler, či překladač jazyka C/C++ a linker schopný linkovat do formátu LE (Linear Executable). S těmito nástroji a microsofským DDK (Device Development Kit) je pro zkušenější systémové programátory tvorba VxD hračkou.

- co vlastně VxD umí. Bylo už řečeno že VxD může kromě jiného nabízet i služby ostatním VxD modulům. Jde přitom vysloveně o obdobu exportovaných funkcí, tak jak jsou známy z knihoven DLL. Dále je každý VxD ve velice úzkém kontaktu se správcem VMM, který mu pomocí jeho kontrolní funkce pravidelně zasílá správy o všech důležitých změnách v systému. Za další je pak ještě náš modul schopen nabízet služby přímo aplikacím DOS, WIN16 API a WIN32 API.
- činnost VxD není multitaskována. Z hlediska operačního systému je to sice samozřejmé, ale odvíjejí se od toho další souvislosti. Z hlediska bezpečnosti je totiž velmi důležité, že pokud běží naše VxD, nemají ostatní komponenty systému nárok.
- To samé však platí i o VxD případných virů, trojských koní apod.
- Mechanismus hookovacích funkcí je velmi mocným mechanismem ve zkoumaném systému, program v MS DOSu, když chtěl "vidět" příslušný příkaz dříve než například jádro MS DOSu, použil tabulku přerušení. Ve W95 (na úrovni VxD) nehrají sice přerušení takovou roli, neboť zde máme služby volané jako API. Správce VMM poskytuje totiž místo modifikace vektorů přerušení právě takzvané "hookovacví" funkce. Pomocí nich můžeme instalovat některou z funkcí našeho VxD přes službu jiného modulu. Srovnáme-li MS DOS s W95 zas tak mnoho se nezměnilo - dříve se přepisovaly tabulky vektorů přerušení, dnes se zas budou "hookovat" služby.

Windows 2000:

- ✓ Windows NT Lan Manager (NTLM) – historický protokol, který nevyhovuje současným požadavkům a je ve Windows 2000 zabudován z důvodu kompatibility se staršími klienty.
- ✓ Protokol Kerberos byl navržen na začátku 80-tých let na MIT. Nyní je ve verzi 5. Jedná se o základní protokol pro síťovou autentizaci na počítačích s Windows 2000. Podle mytologie je Kerberos jméno tříhlavého psa střežícího vchod do podsvětí
- ✓ KERBEROS je prostředek pro správu distribuované databáze hesel uživatelů. V rámci několika serverů internetové domény tak umožnuje synchronizaci hesel pro přihlašování kdykoliv si uživatel změní své heslo, projeví se tato změna na všech serverech, na nichž je Kerberos nainstalován Základní výhodou je tedy jednotné přihlašování k serverům v rámci jedné domény, pokud na nich příslušný uživatel má založen účet.
- ✓ Kromě výše popsané vlastnosti podporuje systém Kerberos ještě celou řadu služeb a má vlastnosti, které ho staví ve vývojové řadě produktů tohoto typu mnohem výš než např. NIS a jemu podobné systémy. Mj. zavádí tzv. **lístky (tickets)**, což je metoda, která se blíží dnes velmi populárním přístupovým metodám typu "login once, use anytimes", tedy jednorázové přihlášení bez nutnosti znovu uvádět své heslo při každém dalším použití autorizačních služeb během soustavné práce.
- ✓ **Tickets (tzv. lístky)** – každý uživatel po přihlášení do systému může požádat o tzv. **ticket**. Tím se autorizuje vůči systému Kerberos a ten mu pak dovolí více než uživateli, který ticket nepoužil. První lístek získá uživatel automaticky během procesu přihlášení k systému. Platnost lístku je standardně 5 minut, pokud uživatel hodlá jeho platnost obnovit (přesněji řečeno: vygenerovat nový), může tak učinit prostřednictvím programu **kinit**.
- ✓ **Integrita dat a jejich utajení** – klient i server mají k dispozici společný symetrický klíč Kc,s. Tento klíč může být přímo použit pro šifrování a dešifrování dat, o které se stará modul CSP (Crypto Service Provider). Proti jakýmkoliv změnám během přenosu jsou přenášená data chráněna digitálním podpisem, který je tvořen zašifrovaným kontrolním součtem těchto dat. Pro kontrolní součet se používá algoritmus HMAC (Hash-based Message Authentication Code) a pro jeho zašifrování algoritmus RC4.
- ✓ **Kerberos a veřejný klíč** – Windows 2000 obsahují i rozšíření protokolu Kerberos o možnost autentizace založené na využití šifrování pomocí veřejného klíče. KDC je možné zaslat zašifrovanou žádost o průkaz privátním klíčem, který je certifikován některou z certifikačních autorit, kterým KDC důvěřuje.

VIZUALIZACE:

Paravirtualizace:

- ✓ **Paravirtualizace** je tak možná jen díky tomu, že konkrétní procesory podporují více úrovní ochrany. Procesory Intel mají definovány 4 úrovně ochrany, tzv. okruhy (rings). Na nejvyšším stupni ochrany (ring 0) běží operační systém, uživatelské programy běží s nejnižším stupněm ochrany (ring 3). Ostatní stupně se běžně nevyužívají. Pokud použijeme paravirtualizaci, pak virtuální monitor pracuje na nejvyšším stupni ochrany, tj.

v okruhu 0. Operační systém virtuálního počítače se posune o jeden stupeň (do okruhu 1), aplikační programy běží stále s nejmenší ochranou. Operační systém má tak stále vyšší úroveň ochrany než aplikační programy, na druhé straně už nemůže provádět operace, které vyžadují plně privilegovaný přístup.

- ✓ Úrovně ochrany však můžeme využít i místo výše zvýšené modifikace privilegovaných instrukcí - necháme operační systém ve virtuálním počítači provádět všechny instrukce, pokud však bude chtít provést "zakázanou" operaci (tj. takovou, na kterou teď nemá dostatečná oprávnění), pak dojde k přerušení a řízení převezme virtuální monitor. Ten operaci zkontroluje a provede ji tak, aby správně změnila stav virtuálního počítače. Není v principu třeba měnit operační systém, většina instrukcí běží přímo, pouze privilegované instrukce jsou výrazně pomalejší, protože je musí provést virtuální monitor. Operační systém však může zjistit, že běží ve virtuálním prostředí, protože může mít i na úrovni 1 možnost číst některé části paměti, které jsou ve virtuálním počítači jiné něž ve fyzickém. Pro paravirtualizaci je proto třeba modifikovat některé součásti operačního systému, změny jsou však malé a dobře lokalizovatelné (zvlášť dobře je pak možné provést tyto změny u operačních systémů, k nimž jsou k dispozici zdrojové kódy; i proto začala být tak oblíbená (para)virtualizace v prostředí Linuxu).