Operační systémy 2

Bezpečnost

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

6. prosinec, 2011

Viry (1/2)

- různé kategorie
  - boot viry
  - makro viry
  - companion viry
  - parasitické viry
- různé účely
- poškození spustitelného programu (tan. 622)
  - zápis dodatečného kódu (začátek/konec)
  - přemostění jeho spuštění (do přidané části)
  - spuštění viru
  - spuštění původního kódu
- snaha maskovat viry (šifréry)
- snaha měnit vir, snaha zabránit analýze
- komprese dat

Albánský virus

```
Hi,

This is an Albanian virus. As you know we are not so technical
advanced as in the West. We therefore ask you to delete all your
files on your harddisk manually and send this email to all your
friends.

Thanks for helping us,
The Albanian Hackers
```

Změny kódu

- jmp-makra

```
01: push ebp                    1.: push 05
02: mov eax, 123                2.: ret
                                5.: mov eax, 123
                        ==>     6.: jmp 07
                                3.: push ebp
                                4.: jmp 04
```

- zdroj: Matlach V. Nástroj pro reverse engineering
- polymorfní viry $\Longrightarrow$ změna kódu programu
- viz Tan. 630/631

Debugger

- přerušení (jednobytová operace INT3) – EXCEPTION_BREAKPOINT
- debugger přebírá obsluhu vyjímek
- 1. v místě breakpointu je přepsána instrukce na INT3
- 2. vyvolá se vyjímka $\Longrightarrow$ předání řízení dubuggeru (pozastavení programu)
- 3. instrukce je vrácena zpět, a je provedena znovu

Odhalení debuggeru

- hledání instrukce INT3 v kódu
- záměrné vyvolání vyjímky a sledování, jestli byla obsloužena programem
- sledování prodlev mezi operacemi

Matice přístupů & TCB

- oddělení politiky od mechanismů
- reference monitor – kontrola přístupu
- doména ochrany (protection domain): množina dvojic objekt—oprávnění ($\Longrightarrow$ matice)
- každý proces běží v nějaké doméně (Tan. 646)
- přechod mezi doménami opět lze specifikovat pomocí domény ochrany
- implementace pomocí ACL (+ skupiny, wildcards) $\Longrightarrow$ sloupce
- ,,schopnosti" (capabilities) $\Longrightarrow$ řádky (IBM AS/400)

Trusted Computing Base

- bezpečný $\times$ důvěryhodný systém
- TCB
  - část systému, která splňuje bezpečtnostní kritéria (tan. 655)
  - stará se o dodržování a vynucování práv
  - mj. správa procesů, paměti, I/O

Object Manager (1/2)

- soustřeďuje práci s objekty do jednoho místa
- jednotný způsob práce s objekty
- možnost řídit přístup k objektům
- kontrola spotřebovaných zdrojů
- globální pojmenovávání objektů
- typy objektů:
  - jaderný (kernel) objekt (základní struktury)
  - executive objekt (objekt v executive části jádra; může obsahovat jaderné objekty, RuSo. 135)
  - GDI/User objekt (objekty graf. modulu—odlišný přístup)
- object manager spravuje executive objekty
- k jednotlivým objektům se z uživatelského prostředí přistupuje přes handly
- jádro může přes ukazatele
- uvolnění objektu na základě počtu referencí

Object Manager (2/2)

- executive objekty mají sdílenou hlavičku
  - jméno objektu
  - adresář objektu
  - práva přístupu
  - počet itevřených handlů + ukazatelů
  - využití přidělených kvót
  - seznam procesů mající otevřený handle na objekt
  - ukazatel na typ objektu
    - typické oprávnění
    - příznaky pro synchronizaci
    - metody objektu (open, close, delete, . . .)
    - a další
- handle je vlastně ukazatel do tabulky handlů (každý proces má vlastní)
- záznamy v tabulce: pointer na objekt + příznaky (použití, dědičnost) + oprávnění
- trojúrovňová tabulka
- objekty ve vlastním jmeném prostoru

Kategorie bezpečnosti podle Orange Book

- DoD definuje několik úrovní zabezpečení
- D – žádné požadavky
- C – systémy spolupracujících uživatelů
  - chráněný režim; možnost ovládat oprávnění
  - dokumentace; testování
  - úrovně C1, C2
  - v úrovni C2 nestačí kontrola na úrovni unixových rwx
- B – zpřísněná pravidla
  - popsaný bezpečtnostní model (neformálně nebo formálně)
  - požadavky na architekturu (omezení složitosti $\Longrightarrow$ možnost testovat/ověřit)
  - posílené bezpečnostní prvky (např. přihlašování)
  - detekce napadení, atd.
  - tři úrovně B1, B2, B3
- A
  - A1 – formální model ochrany, včetně důkazu správnosti (XTS-400)

Víceúrovňová bezpečnost

Bell-La Padula Model

- proces s oprávněním na úrovni k může číst pouze na své úrovni a nižší
- proces může zapisovat pouze do objektů na své úrovni a vyšší
- $\Longrightarrow$ armáda
- zajištěno utajení, ne integrita dat

Biba model

- proces s oprávněním na úrovni k může číst pouze na své úrovni a vyšší
- proces může zapisovat pouze do objektů na své úrovni a nižší

Singularity

- OS z Microsoft Research $\Longrightarrow$ nové koncepty
- softwarově izolované procesy (SIP); mohou se skládat z vláken
- izolace na bázi softwaru (ostatní OS využívají HW)
- managed prostředí založené na MSIL, každý SIP má vlastní běhové prostředí, GC
- procesy spolu mohou komunikovat pouze zasíláním zpráv
- $\Longrightarrow$,,kanály" s jednoznačně definovaným rozhraním $\Longrightarrow$ jazyk Sign#
- jde automaticky ověřit, že komunikace neskončí v nežadoucím stavu
- jde (je nutné) ověřit, že program neovlivňuje okolí (spolupráce překladače)
- mikrojádro převážně implementované v Sing#
- přepnutí kontextu/systemové volání rychlé (není potřeba měnit nastavení stránek, mazat TLB)
- základ pro projekt Midori (nástupce Windows?)

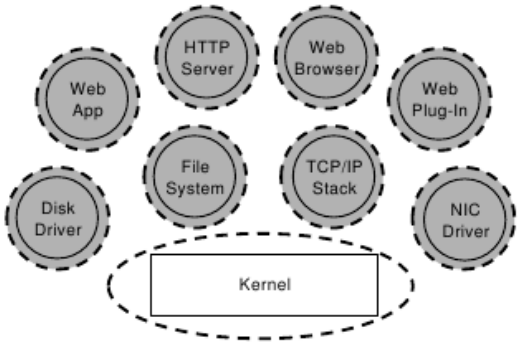

Figure 4a. Micro-kernel configuration (like MINIX 3). Dotted lines mark protection domains; dark domains are user-level, light are kernel-level.

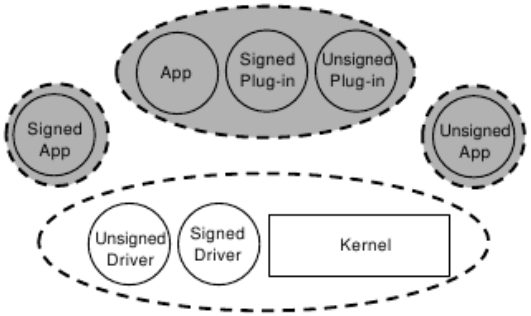

Figure 4b. Monolithic kernel and monolithic application configuration.

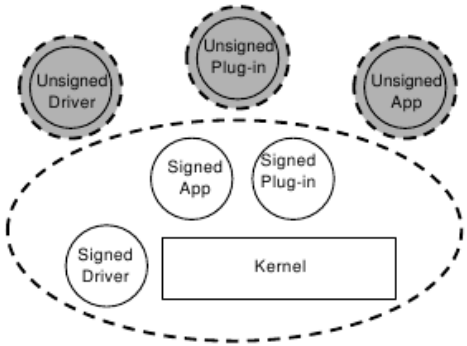

Figure 4c. Configuration with distinct policies for signed and unsigned code.