AxxB36SOJ

Anotace

V předmětu posluchači získají znalosti potřebné k tvorbě assemblerových programů pro nejrozšířenější platformu PC. Důraz je kladen na optimální využívání vlastností mikroprocesoru a efektivní řešení spolupráce HW a SW. Dále budou probrána x86 specifika majoritních OS z pohledu jádra, kódu aplikace i návaznosti k vyšším jazykům. Tyto znalosti budou dále využity při reverzní analýze, optimalizacích a posuzování bezpečnosti kódu.

Osnova přednášek

1. Úvod, zaměření a požadavky předmětu. Historie x86, vývoj uspořádání PC.
2. Procesor i8086. Instrukční soubor, registry a reálný režim.
3. Softwarová přerušení. Moduly periferií, služby BIOSu, funkce DOSu.
4. Přímý přístup k HW. Hardwarová přerušení, obsluha základních periferií PC.
5. Procesor i80386. Instrukční soubor, chráněný režim, ochrana paměti, stránkování.
6. Optimalizace kódu. Ladění na velikost, rychlost, techniky refaktorizace.
7. Assembly gems a demoscéna. Rutiny pro generování zvuku a grafiky.
8. Specifika x86 linuxového jádra. Zavádění, virtualizace paměti, správa procesů.
9. Specifika x86 jádra Windows. Virtualizace paměti, HAL, správa procesů.
10. Anatomie x86 aplikace. Uživatelský prostor, spustitelné soubory, knihovny, relokace.
11. Vazba na vyšší jazyky a reverzní analýza. Volací konvence, externí moduly.
12. Bezpečnost aplikací a jádra. Havárie kódu, druhy útoků, zapouzdření kódu.
13. Nestandardní režimy procesoru, x86-64. Nepublikované instrukce, podpora virtualizace.

Osnovy cvičení

1. Základní instrukce, použití registrů, vztah strojového kódu k syntaxi assembleru.
2. Větvení, podprogramy, instrukce IN a OUT, interakce s uživatelem.
3. Použití vvbraných služeb BIOSu, soubory COM a EXE, práce s pamětí.
4. Obsluha přerušení, manipulace s vektory, kontext a reentrance, práce s řadičem.
5. Mechanismy chráněného režimu, vytvoření a spuštění 32 bitového kódu.
6. Optimalizační úlohy pro 16/32 bitový kód, pipeline, základy optimalizačních triků.
7. Pseudonáhodné generátory, konvolutorní filtry, fázový akumulátor, vázané oscilátory.
8. Analýza setupu, fáze zavedení kernelu, rozbor klíčových části jádra, (ne)známé chyby.
9. Rozbor klíčových částí jádra Windows, proces zavádění, anatomie aplikace.
10. Konvence volání jádra, ELF, vytváření jednoduché int 80h aplikace, signály, minimalizace.
11. In-line asembler, direktivy, symboly a constraints, vlivy HLL optimalizace.
12. Konstrukce root-shellcode, eskalace práv, využití bezpečnostních chyb, skrytí.
13. Použití nepublikovaných instrukcí, utajování kódu, mechanismy polymorfismu.

Poznámky

1) Uvod a historie (zamereni a pozadavky predmetu, historie architektury x86, vyvoj usporadani PC)
   - zakladni instrukce a pouziti registru, strojovy kod a syntaxe assembleru, nasm, prvni rutina na 0000:7c00 komunikujici skrz videoram

2) Realny rezim, i8086 (instrukcni soubor, strojovy kod a assembler i8086, segmentace, mapa pametoveho prostoru 0..1MB)
   - vetveni, slozitejsi programy, podprogramy, instrukce in/out, jednuducha interakce s uzivatelem, realizace cekani

3) Pouziti BIOSu a DOSu (soft. preruseni, proces bootovani, sluzby DOSu, BIOSu, formaty COM a EXE, alokace pameti)
   - pouziti vybranych sluzeb BIOSu a DOSu, pametova mapa DOSu, vytvareni COM a EXE souboru, vyuziti pro ladeni realmode rutin

4) Primy pristup k hardwaru (IO prostor, VGA, hard. preruseni, obsluha klavesnice a portu, casovac, radic preruseni)
   - vytvoreni obsluhy preruseni, tabulka vektoru, nalezitosti ochrany kontextu a reentrance, prace s radicem preruseni

5) Chraneny rezim, i80386 (registry, instrukce a prefixy, PM, selektor/deskriptor, strankovani, IDT/GDT/LDT, A20, FPU)
   - vkladani 32-bit instrukci z realneho modu, prepnuti z/do chraneneho modu, vytvoreni 32-bit segmentu, manipulace s limity, vlastni 32-bit rutina

6) Optimalizace kodu (opt. velikost/rychlost, techniky refaktorizace, tabulky, rozkladani cyklu, opt. skoku, align, lokalita dat)
   - optimalizacni ulohy pro 16-bit a 32-bit kod, principy zakladnich optimalizacnich triku, vlastnosti pipeline modernich procesoru, metodika optimalizace na rychlost

7) Assembly gems a demoscena (fix. P. aritmetika, celociselne algoritmy, prand generatory, fazovy akumulátor, oscilatory)
   - generovani palet, textur, subsampling, mixovani zvuku, LCG generatory periody, konvolutorni filtry, gen. gon. fci

8) Koncepce linuxoveho jadra (boot/dekomprese, virtualizace, segmentace, strankovani, sprava procesu, mmap, fork, COW)
   - analyza setupu, stylu dekomprese a mechanismu postupneho zavedeni kernelu, rozbor klicovych casti linuxoveho jadra, zname i nezname chyby, zajimavosti

9) Koncepce jadra windows (HAL, virtualizace pameti, segmentace, strankovani, sprava procesu, uziv./syst. knihovny)
   - rozbor klicovych casti jadra windows, anatomie PE-aplikace, proces zavadeni

10) Prostredi aplikace (userspace, inicializace, spustitelne soubory, sdilene knihovny, vyjimky, signaly, thready, sluzby jadra)
   - konvence volani jadra, ELF, PE, vytvareni jednoduche int 80h aplikace, chovani signalu, vyuziti multiplexu, minimalizace velikosti, debugging asm-aplikace

11) Vazba na vyssi jazyky (volacich konvence, in-line assembler, externich moduly, reverzni analyza kodu a jeji nastroje)
   - C-programy s inline assemblerem, nalezitosti interface, direktivy, symboly a constraints, rizika vyssich stupnu optimalizace, reverzni analyza aplikace

12) Bezpecnost aplikaci a jadra (havarie kodu, vycerpani prostredku, DOS, exploit, SMC, injektaz, root-shellcode, eskalace)
   - konstrukce root-shellcode, vyuziti bezpecnostnich chyb aplikaci k aktivaci kodu, vyuziti chyb jadra k eskalaci prav, modifikace beziciho jadra, skryti

13) Dalsi rezimy procesoru (nepub. Instr., VM86 rezim, SMM, FLAT rezim, x86-64, Vanderpool, Pacifica, PAE/PSE, DEP)
   - pouziti nepublikovanych instrukci, rozdilna chovani soucasnych procesoru, metody utajovani kodu, zaklady polymorfismu, prokladani kodu)