Virtualny adresovací priestor na linuxoch

Vložil/a socket [1], 12 Květen, 2008 - 15:07

- GNU/Linux a BSD [2]
- Security [3]

Rád by som v článku podrobnejšie predviedol používanie segmentovanej a stránkovanej pamäti na operačnom systéme linux, vzhľadom na bezpečnosť.

Linuxové jadro zásadne nevyužíva možnosti segmentovanej pamäti. Pretože je ale na x86 segmentovanie povinnou časťou adresácie , musí sa s tým nejak jadro linuxu vysporiadať.

Definované sú celkom 4 segmenty. Kódový a dátový segment pre režím jadra , kódový a dátový segment pre užívateľský režím.

V zdrojových kódoch linuxového jadra najdeme pre nastavenie deskriptoru takéto alebo podobné hodnoty (závisí od verzie jadra).

```
ENTRY(gdt_table)
.quad 0x0000000000000000 /* NULL decriptor */
.quad 0x0000000000000000 /* not used */
.quad 0x00cf9a000000ffff  /* 0x10 kernel 4GB code at 0x00000000 */
.quad 0x00cf92000000ffff  /* 0x18 kernel 4GB data at 0x00000000 */
.quad 0x00cffa000000ffff  /* 0x23 user 4GB code at 0x00000000 */
.quad 0x00cff2000000ffff  /* 0x18 user 4GB data at 0x00000000 */
```

Vidíme, že všetky segmenty začínajú na adrese 0x00000000 a sú veľké 4GB.
Sú kódové či dátové , nikdy nie sú systémové. Dva sú určené pre režim jadra a majú DPL na najvyšej úrovni oprávnení (Ring0). Dva sú pre užívateľský režim a majú najnižšiu úroven oprávnení takz. Ring3.
Kódové segmenty majú v type nastavené príznaky pre čítanie a spustiteľnosť. Dátové segmenty majú nastavené príznaky pre čítanie a zápis.

Pretože sa kódový i dátový segment naplno prekrývajú, sú dáta adresované offsetom vzhľadom k ľubovoľnému segmentu uložené na rovnakej virtuálnej adrese. Predstavme si napríklad hodnotu 0x90 uloženú na adrese 0x10. Keď prevedieme prístup na adresu 0x10 s cieľom zapísať na ňu nejaké dáta, prístup sa prevedie cez dátový segment a bude povolený. Ak skúsime data na adrese 0x10 spustiť, prístup sa prevedie cez kódový segment a bude taktiež povolený. To znamená, že v celom virtuálnom adresovom priestore ide previesť čítanie, zápis (pokiaľ nie je stanované stránkovanie inak) aj **spúštanie**.
Jednotlivé segmenty a ich rozdielne príznaky sú prekryté. Zostáva jedine rozdiel úrovní oprávnení DPL.

K ochrane častí virtuálneho adresovacieho priestoru proti zápisu linuxové jadro využíva jedine stránkovanie.

Pri požiadavke procesu o pridelenie stránky sa hodnota U/S bitu nastavuje podľa toho, či o ňu žiada jadro alebo užívateľský proces (pre jadro je nastavená na 0). Spustiteľný binárny formát je vnútorne rozdelený na sekcie, v ktorých sú príznaky na povolenie zápisu. Pri zavedení programu do pamäti stránky preberajú príznaky z týchto sekcií. Stránky, ktoré sa za behu programu alokujú pre potreby zásobníka či haldy, majú zo známych dôvodov povolenie zápisu nastavené.

Z horeuvedených poznatkov môžeme učiniť dva dôležité závery. Celý virtuálny adresovací priestor procesu je prístupný pre čítanie a spúšťanie. Stránky, ktoré obsahujú špecifické sekcie binárneho spustiteľného formátu, môžu byť chránené proti zápisu. Zbytok stránok má príznak zápisu povolený.

Dúfam, že ste si z článku urobili aký-taký záver o virtuálnom adresovacom priestore na linuxoch.

==========
good links:

x86 memory segmentation [4]
Linux kernel Memory Management [5]
Virtual memory [6]
Memory management [7]

btw: URL se při přidání článku vytratily. v případě nesrovnalostí ať autor napíše redakci SP. Díky za článek

URL článku: https://security-portal.cz/clanky/virtualny-adresovac%C3%AD-priestor-na-linuxoch

Odkazy:
[1] https://security-portal.cz/users/socket
[2] https://security-portal.cz/category/tagy/gnu/linux-bsd
[3] https://security-portal.cz/category/tagy/security
[4] http://en.wikipedia.org/wiki/Segment_register
[5] http://linux-mm.org/
[6] http://en.wikipedia.org/wiki/Virtual_memory
[7] http://en.wikipedia.org/wiki/Memory_management