Skrývání virtuálního prostředí před malwarem

Fakulta informačních technologií, ČVUT v Praze
Ing. Tomáš Hradský
vedoucí práce: Ing. Petr Kurtin
odborný poradce: Luboš Hnaníček

Úvod

Antivirové společnosti musí být schopné analyzovat a co nejpřesněji klasifikovat desítky až stovky tisíc dennodenně vznikajících – potenciálně škodlivých – programů. Jakákoliv chyba, ať už nesprávné označení programu za neškodný (*false negative*) nebo naopak detekce legitimního programu (*false positive*), může firmě způsobit újmu na dobrém jméně a v konečném důsledku i finanční ztrátu.

Z důvodů ohromného množství neznámých programů se antivirové společnosti spoléhají na automatickou analýzu, která probíhá na virtuálních počítačích. Ovšem „zevnitř" virtualizačních nástrojů lze s jistotou určit, že se vskutku jedná o virtuální počítač, ba dokonce lze přesně určit výrobce – například *VirtualBox*, *VMWare*, . . .

Tvůrci malwaru si tohoto jsou samozřejmě vědomi, a proto detekují virtuálního prostředí a případně se zdrží škodlivé činnosti. Tím se úspěšně vyhne detekci a bude prohlášen za neškodný – *false negative*.

Práce vznikla ve spolupráci s firmou *Avast Software s.r.o.*, která využívá virtualizační nástroj *VirtualBox*.

Přínos

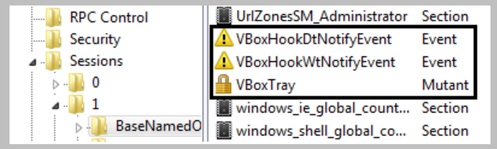
Práce obsahuje detailní popis metod, které malware může použít pro detekci virtuálního prostředí a zároveň i postup, jak jednotlivé metody implementovat. Především však představuje protiopatření, jimiž lze uvedeným metodám používanými malwarem zabránit.

Výsledkem práce je upravený *VirtualBox*, který lze použít pro analýzu malwaru. Jednou z hlavních výhod skrytí virtuálního prostředí je snížení počtu *false negative* bez často protichůdného zvýšení *false positive*.

Z rešeršního pohledu se práce zabývá vznikem a historií virtualizace samotné. Převážná většina teoretické části je věnována jednomu ze způsobů, jak virtuální prostředí vůbec vytvořit a to takzvané hardwarové podpoře virtualizace, jejíž principy jsou v praktické části hojně využívány. V práci lze nalézt i popis některých interních mechanizmů operačního systému *Windows*.

Projevy virtuálního prostředí ve *Windows*

Virtualizační nástroje integrují do virtualizovaných *Windows* několik součástí pro podporu své činnosti. Mezi ně patří například **procesy**, **služby**, **ovladače** a k nim příslušné **spustitelné soubory** nebo **knihovny**.



Obrázek: `WinObj` (součást `Sysinternals`)

Například zde je mimojiné vidět pojmenovaný mutex `VBoxTray`. Jakýkoliv, byť neprivilegovaný proces se může dotázat na existenci tohoto mutexu a tím se 100% jistotou určit, zda se nachází ve *VirtualBoxu*.

Postupem skrytí bylo dynamické generování jmen problematických objektů při každém jejich vytvoření, případně při restartu virtuálního počítače. Pro zachování funkcionality bylo potřeba distribuovat nová jména mezi součástí *VirtualBoxu*. K tomu byl využit správce virtuálních počítačů **hypervisor** a uměle vytvořený komunikační kanál.

Virtuální hardware

Komplexní virtualizační nástroj virtualizuje i hardware. Tím však zanáší další jednoznačné identifikátory virtuálního prostředí. Sem patří například **výrobce**, **model** a **seriové číslo** jednotlivých HW komponent nebo také **MAC adresa**.

Zde byla snaha ztotožnit virtuální hardware s fyzickým – přepsání pevně zakódovaných řetězců na jiné.

Obrázek: `Device Manager`

Identifikace použité virtualizační metody

Dalším způsobem, jak rozpoznat virtuální prostředí, je pokusit se detekovat konkrétní virtualizační metodu. Práce se zabývá pouze hardwarovou podporou virtualizace, která je implementována přímo v procesorech. Její použití je prozrazeno některými **procesorovými hodnotami**, odlišným **chováním některých instrukcí** či specifickými **časovými závislostmi**.

```
int HypervisorPresentBit()
{
    __asm
    {
        mov eax, 1
        cpuid
        shr ecx, 31
        mov eax, ecx
    }
}
```

Obrázek: Vykonání instrukce `CPUID`

Uvedenou funkci může opět vykonat jakýkoliv program, aniž by potřeboval zvýšené oprávnění. Pokud funkce navrátí 1 je jasné, že v procesoru je spuštěna hardwarové podpora virtualizace.

Další odlišnosti (stejný program spuštěn na fyzickém a virtuálním počítači):

```
HypervisorPresentBit: 0
HypervisorVendor:
Prumerny pocet cyklu na CPUID instrukci: 186
```

Obrázek: Fyzický počítač

```
HypervisorPresentBit: 1
HypervisorVendor: VBoxVBoxVBox
Prumerny pocet cyklu na CPUID instrukci: 2240
```

Obrázek: Virtuální počítač (*VirtualBox*)

Jelikož hypervisor má absolutní kontrolu na činností virtuálního počítače, může jakkoliv změnit hodnoty, které lze tímto způsobem získat. Cílem bylo upravit hypervisor a sjednotit chování s fyzickými počítači.

Závěr

V práci je popsáno celkem **15** obecných projevů virtuálního prostředí, kde pro každý je uveden konkrétní postup, jak jej programově objevit. Stěžejní částí jsou však teoretické návrhy, jak jednotlivé projevy efektivně skrýt, samozřejmě se zachováním veškeré funkcionality virtualizačního nástroje.

Ačkoliv je práce zaměřena výhradně na *VirtualBox*, byly veškeré návrhy konstruovány dostatečně obecně, aby mohly být aplikovány na jakýkoliv virtualizační nástroj využívající hardwarovou podporu virtualizace.

Prakticky bylo **11** uvedených návrhů implementováno do zdrojových kódu *VirtualBoxu*, čímž došlo k ověření jejich funkčnosti. Samotné zdrojové kódy jsou přílohou práce.

hradsky@avast.com