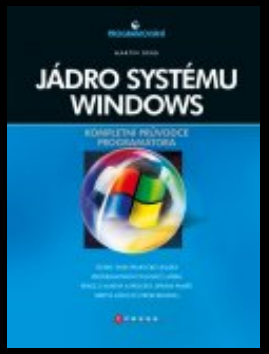
# Implementace systémů HIPS: ve znamení 64bitových platforem




Martin Dráb
martin.drab@email.cz

SECIT.SK

# HIPS: základní definice

- Majoritně používané operační systémy disponují bezpečnostními modely, které dovolují jednotlivým uživatelům určit, co smějí a co nesmějí.
- V případě Windows je většina domácích uživatelů zvyklá pracovat neustále s oprávněním administrátora, což činí bezpečnostní model značně neefektivní.
- Systémy HIPS – programy, které monitorují, oznamují (a blokují) podezřelé aktivity v operačním systému.

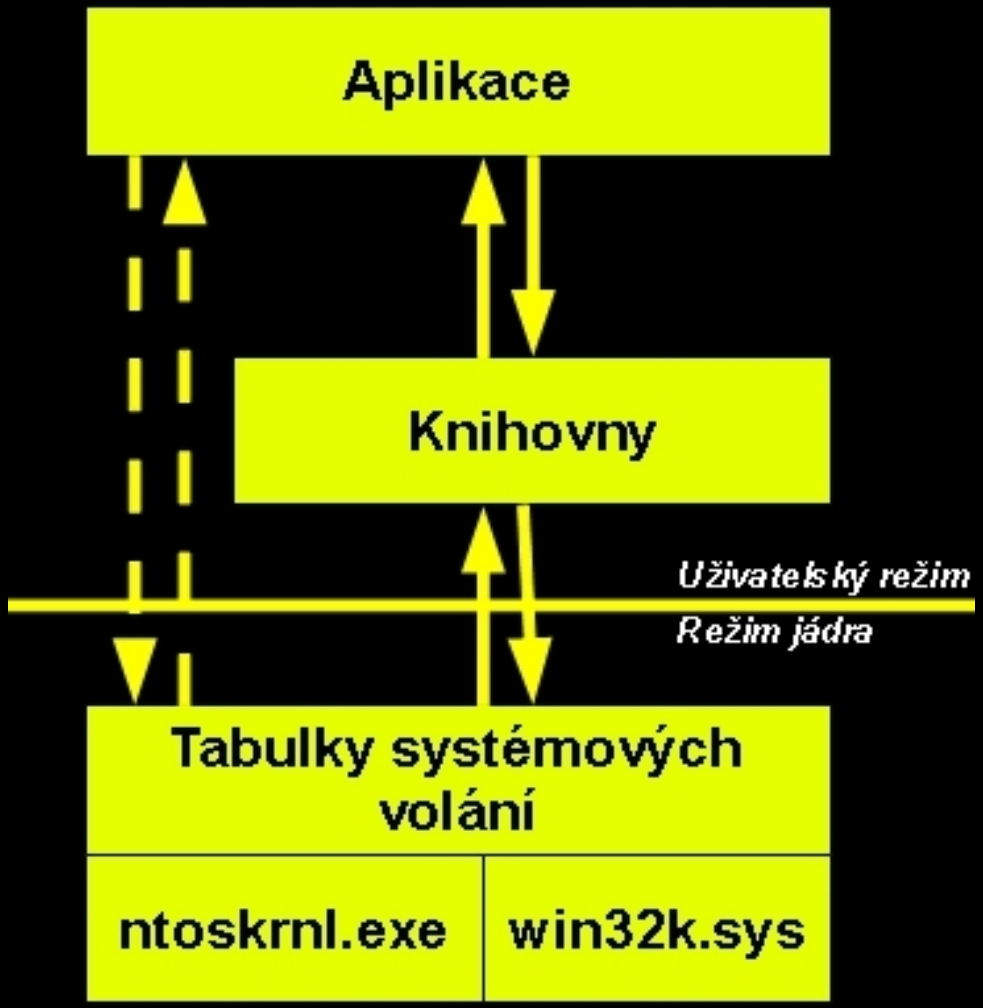
# Čemu se budeme věnovat

- 64bitovým verzím Windows
- Rozhraním, která pro ulehčení (a umožnění) implementace poskytuje Microsoft.
- Zajímavým aspektům jádra, které mohou při implementaci systému HIPS pomoci (ač k tomuto účelu původně zamýšleny nebyly)

# Čemu se budeme věnovat

- Patchguard
- OB Filtering Model
- Chráněné procesy
- GUI a možnosti s ním spojené
- Případně něco dalšího

# Architektura systému



# Patchguard

- Ochrana integrity některých součástí jádra
- Nedeterministický
- Ověřování kontrolních součtů různých oblastí
- V případě zjištění nesrovnalosti je běh systému zastaven modrou obrazovkou smrti
- V režimu ladění není aktivní
- Implementován tak, aby nebylo snadné naň užít reversní inženýrství

# Patchguard

- Není tedy možná „jen tak“ modifikovat kód a důležité datové struktury jádra
  - Souborový systém
  - Registr
  - Procesy a vlákna
  - Síť
- Některá jsou použitelná až od Windows Vista SP1

# Objektový model

- Jádro Windows reprezentuje mnoho entit (procesy, vlákna, klíče registru, otevřené soubory...) jednotným způsobem; entitou zvanou **objekt jádra** (kernel object)
- Jednotná reprezentace dává těmto entitám například možnost pojmenování či nastavení ochrany v rámci bezpečnostního modelu.
- Aplikace s objekty jádra pracují prostřednictvím nepřímých odkazů – **handle**. Každé handle v sobě nese zakódovanou informaci o cílovém objektu a o tom, co skrz něj může s objektem aplikace dělat.

# OB Filtering Model

- Oficiálně dokumentované, a tedy legitimní
- Dovoluje monitorovat (někdy i blokovat či měnit) přístup k objektům jádra
- Zatím podporovány pouze procesy a vlákna
- Dává jistou kontrolu pouze nad vytvářením handle, ostatní operace nejsou podporovány.
- Dostupné od Windows Vista SP1

# Chráněné procesy

- Speciální typ procesu dostupný od Windows Vista
- Systém brání obyčejným procesům získat k těm chráněným některá oprávnění. Mezi chráněnými procesy navzájem žádná bariéra neexistuje.
- Chráněné procesy nemohou mít děti
- Hlavní soubor chráněného procesu a jím používané knihovny musí být podepsány speciálním certifikátem

# Chráněné procesy

- Chráněné procesy lze:
  - Násilně ukončit
  - Čekat na ukončení
  - Zjistit některé vlastnosti
  - Pozastavit
- Chráněná vlákna lze:
  - Pozastavit
  - Čekat na ukončení
  - Číst některé vlastnosti
  - Měnit některé vlastnosti (ale kontext k nim nepatří)

# Chráněné procesy

- Důvodem vzniku není pomoci systémům HIPS, ale lepší podmínky pro implementaci DRM
- System a audodg.exe
- Vytvoření chráněného procesu
  - Změna bitu ve struktuře procesu
  - Okamžitý účinek
  - Patchguard to zřejmě nehlídá

# Grafické uživatelské rozhraní

- Implementováno v rámci ovladače win32k.sys
- Vlastní tabulka systémových volání
- Mnoho ovládacích prvků (okna, tlačítka, textová pole...) reprezentováno objekty zvanými okna (windows)
- Založeno na zasílání zpráv příslušným oknům, případně vláknům
- Příliš nepodléhá bezpečnostnímu modelu (okna nepatří mezi objekty exekutivy)

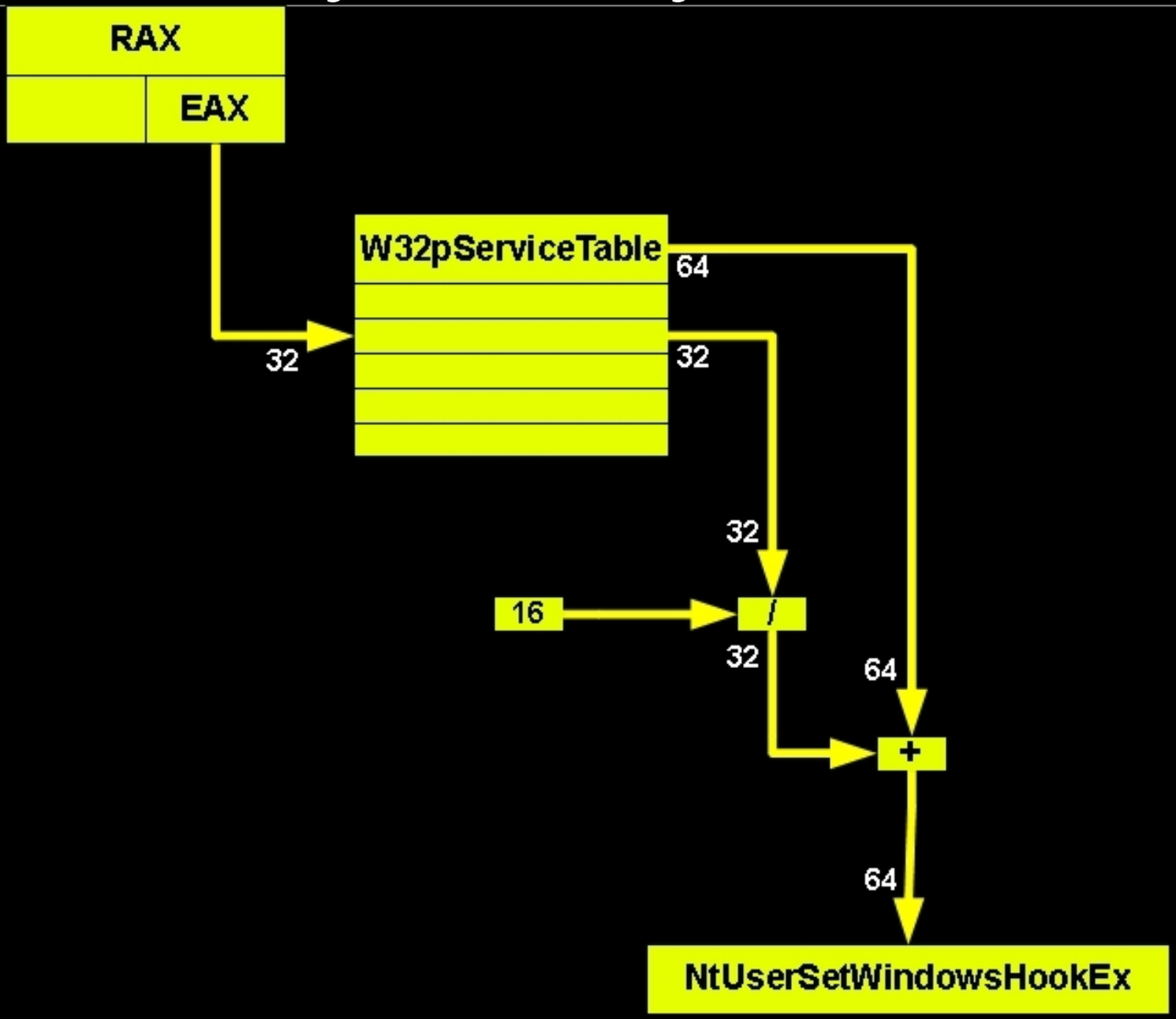
# Grafické uživatelské rozhraní

- Služby ovladače win32k.sys jsou zajímavé pro malware či spyware. Proto by se mělo jednat i o předmět zájmu systémů HIPS
- Škodlivý kód může prostřednictvím win32k.sys:
  - Ukončovat ostatní procesy
  - Injektovat vlastní knihovny do cizích procesů
  - Monitorovat nejen události myši a klávesnice
  - Útočit na GUI ostatních aplikací
  - ...
- Windows zatím nedisponují žádnými speciálními rozhraními.

# Grafické uživatelské rozhraní

- Ovladač win32k.sys však není hlídán technologií Patchguard
- Obranu lze založit na modifikaci jeho kódu
- Také lze modifikovat jeho tabulku systémových volání
- Obtížnější než na 32bitových verzích Windows
- V některých případech lze použít i elegantnější způsoby, které jsou založené na implementaci jednotlivých nebezpečných mechanismů.

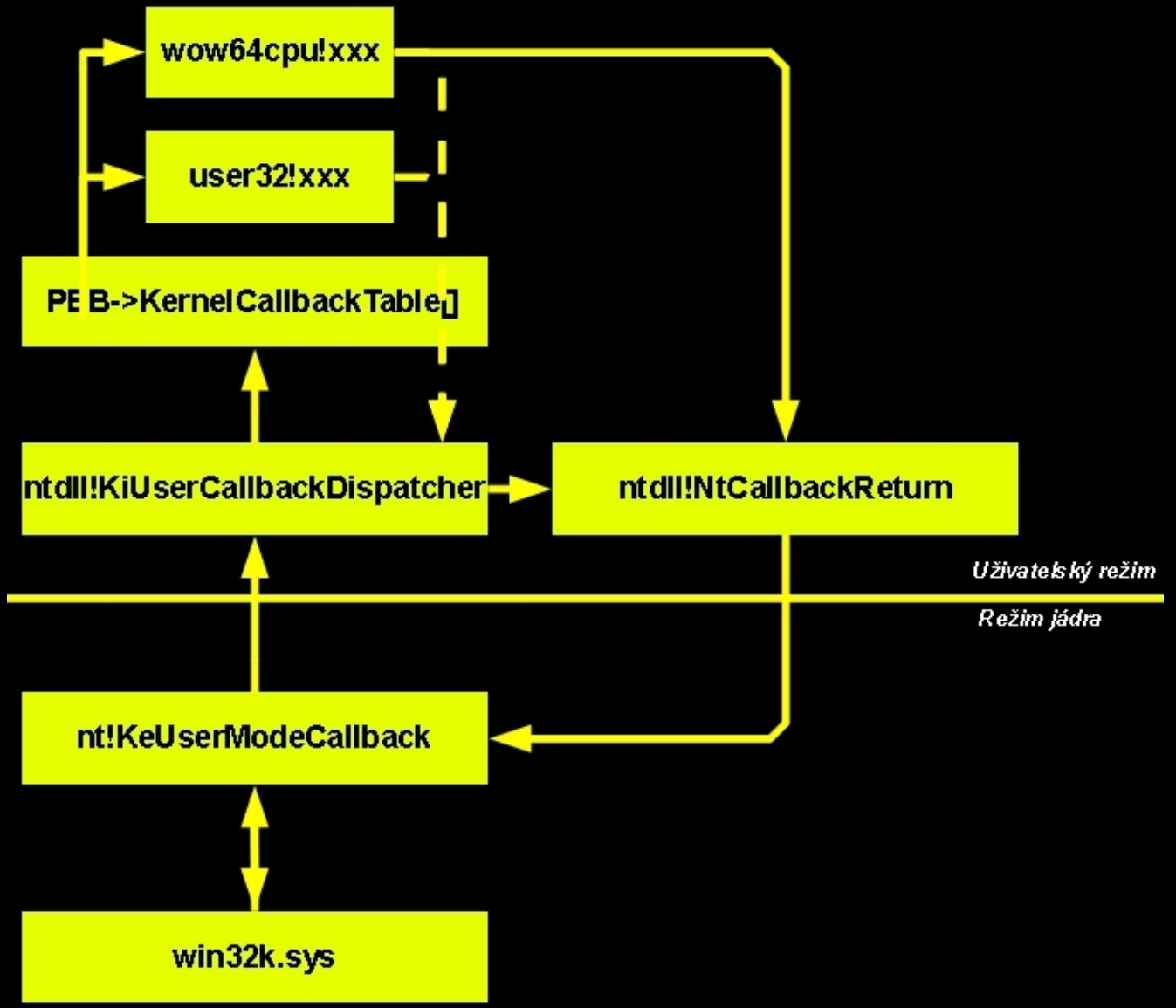
# Tabulka systémových volání na x64



# Windows Hooks

- Velmi staré rozhraní (snad již od Windows 3.x), které umožňuje monitorovat a ovlivňovat zejména transport přijímání zpráv
- Při většině použití dochází k samovolnému vložení knihovny DLL s monitorovacím kódem do adresového prostoru aplikace přijímající zprávy. Prováděno líným algoritmem.
- Rozhraní tedy poskytuje možnost injekce kódu do cizích procesů, naštěstí její rozsah lze i legitimními způsoby omezit

# Implementace rozhraní Windows Hooks



# Důsledky a zajímavosti

- Aplikace se může rozhodnout, zda bude pomocí WH monitorována (a ovlivňována)
- Lze zajistit pouze pomocí modifikací knihoven v uživatelském režimu, pokud si ohlídáme, aby tyto modifikace nemohly být odstraněny
- Funguje i na některé další mechanismy ovladače win32k.sys použitelné pro špatnou věc
- Podobným způsobem lze monitorovat i předávání výjimek a APC

# Další zajímavé koncepty

- Objekty Desktop a WindowStation
  - Omezení dosahu zpráv a rozhraní Windows Hooks na vlákna/procesy běžící na v rámci jednoho objektu Desktop (WindowsStation).
- Objekty Job (omezení na skupinu procesů)
  - Zprávy pouze v rámci procesů v objektu
  - Zákaz změny rozlišení, změny nastavení systému...
  - I standardní limity (paměť, doba běhu, počet...)
- Primárně pro systémy HIPS nepoužitelné, nelze se dotázat uživatele. Sandbox realizovatelný
- Chromium, Avast, ...

# Závěr

- Jsou nějaké otázky?
- Kontakt
  - **Email:** martin.drab@email.cz
  - **ICQ:** 332970040
- Máte-li s sebou moji knížku a chcete-li ji podepsat, přijďte nyní nebo kdykoliv během konference