National Institute of
Advanced Industrial Science
and Technology
AIST

RISEC
Research Institute
for Secure Systems

Bitvisorをベースとした
既存Windowsのドライバメモリ保護

須崎 有康、八木 豊志樹、古原 和邦、
石山 智祥、村上 純一、鵜飼 裕司

（独）産業技術総合研究所　セキュアシステム研究部門
フォティーンフォティ技術研究所

概要

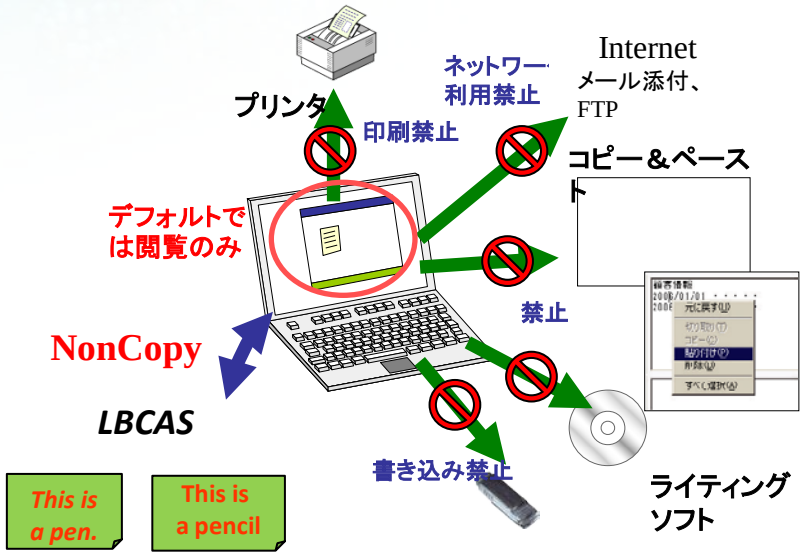
- 背景
  - ドライバメモリを守る理由
- ドライバメモリ保護に必要な技術
  - インストール済みのOSの外部から挿入可能なハイパーバイザー
  - VM Introspection
  - 関連研究
- 現状
  - メモリの保護はStealth breakpointsを活用
- 改善計画
- まとめ

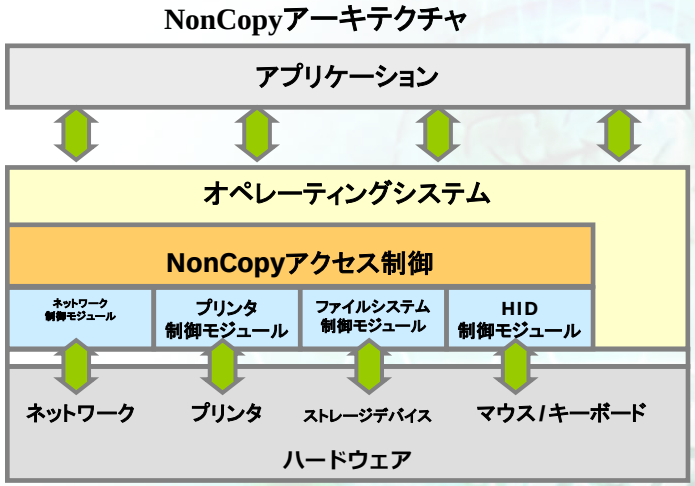
National Institute of Advanced Industrial Science and Technology
AIST

背景

RISEC
Research Institute for Secure Systems

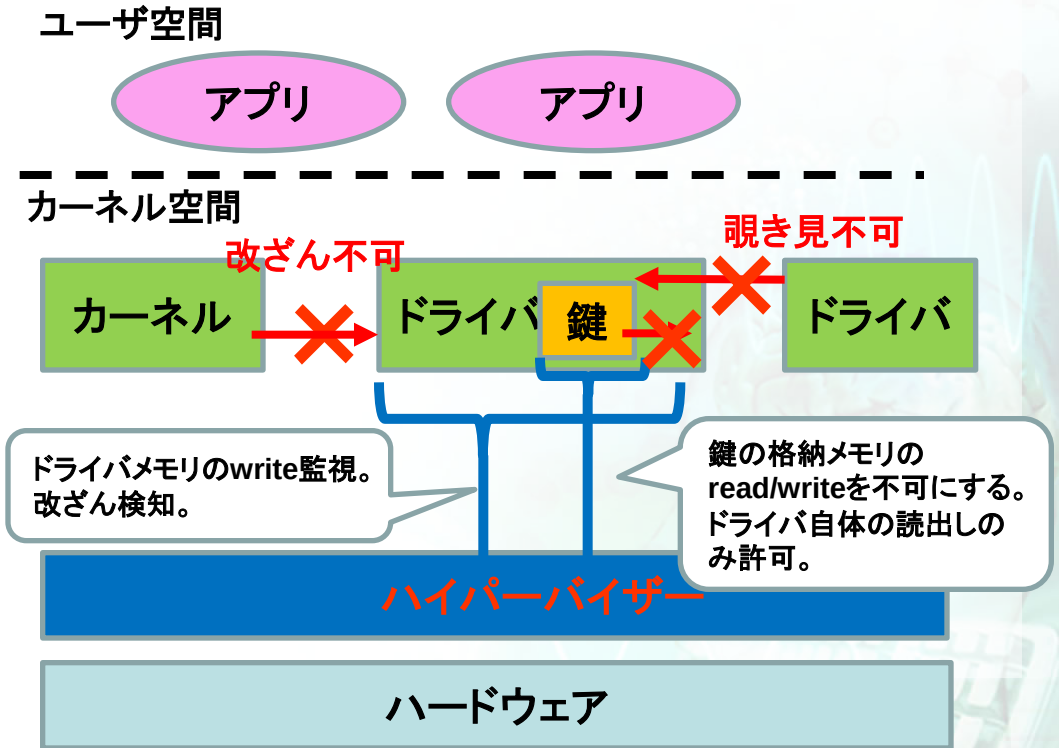
- クライアントおよびサーバ双方からの情報漏えいを防止するアクセス制御技術の研究開発
  - 総務省による戦略的情報通信研究開発推進制度(SCOPE) H23-25
    - 産総研、サイエンスパーク、(FFRI)
- Windows上でアクセス制御を行うNonCopy (サイエンスパーク社製品)を活用
  - NonCopyはドライバによりファイルコピー、Screen Cut&Paste、印刷などを行うWindows APIを抑制する。





ドライバへの攻撃

- ドライバの中に暗号鍵がある。ドライバは長時間メモリに滞在するため、漏えいの危険性がある。
  - メモリに長期滞在するデータの危険性は[Chow, USENIX Sec 04]より指摘されている。
- ハイパーバイザーでOSの下位よりドライバを監視して不正アクセスを防御する。
  - ハイパーバイザーで守ることで、OSをだます攻撃にも対応する。



National Institute of Advanced Industrial Science and Technology
AIST

ドライバメモリ保護に必要な技術

RISEC
Research Institute for Secure Systems

- ハイパーバイザーに必要な技術
  1. インストール済みのOSに対して外部から挿入可能なハイパーバイザー
  2. VM Introspection
- 関連研究

National Institute of Advanced Industrial Science and Technology AIST

Windowsドライバ保護に必要な技術 1(デバイス管理問題)

RISEC Research Institute for Secure Systems

- 多くのハイパーバイザーは複数OSを1台のマシンで起動するために使われている
  - 例：IaaSクラウド
- 問題点
  - ホストOSが必要
    - 管理ソフトウェアの大規模化
      - VMware ESXiは5.xよりホストOS無しになったが、管理サーバが必要
    - 基本的にデバイスがホストOSの管理下になる
  - VMデバイスモデルが固定
    - デバイスはVM中の仮想デバイスで共有される
    - プレインストールのOSとは異なる
    - Xen, KVM, QEMUはQEMU Deviceモデルが標準

National Institute of Advanced Industrial Science and Technology AIST

既存のOSへのハイパーバイザー挿入

RISEC Research Institute for Secure Systems

- 既存のOSが想定している物理デバイスをVM環境内で提供する必要がある。
  - VMデバイスモデルが同じにできればよい
    - 各PC毎にデバイスをモデルを用意しなければならない
  - VMからデバイスを直接アクセス
    - I/O PassThrough。ハードウェアが限定。ハイパーバイザーでも対応が必要
    - Para-PassThrough。Bitvisorで提供される

National Institute of Advanced Industrial Science and Technology AIST

Windowsドライバ保護に必要な技術 2(VM Introspection)

RISEC Research Institute for Secure Systems

- ハイパーバイザーがドライバの動作、メモリ空間を認識する必要がある。
  - 守るべきドライバが組み込まれたことの認識
    - Linux: create_module, init_module システムコールなど
    - Windows: NtLoadDriver システムコールなど。
  - 守るべきメモリ空間の認識
    - 各種の実装あり。

VM Introspection

- Ether [GITech Wenke Leeグループ, CCS'08]
  - XenベースのVM Introspection。オープンソース。対象OSはWindows XP SP2
    - http://ether.gtisc.gatech.edu/source.html
- GreenKiller [FFRI Junichi Murakami, BlackHat'08]
  - BitvisorベースのVM Introspection。対象OSはWindows XP SP3
- Alkanet [立命館大学 大月さん、毛利先生, CSS'11]
  - Bitvisorベースのシステムコールトレース。対象OSはWindows XP SP3

National Institute of Advanced Industrial Science and Technology AIST

ドライバ保護の類似研究

RISEC Research Institute for Secure Systems

- Nooks [M.Swift, SOSP'03]
  - 障害時にリカバーがメイン。
- HUKO[NDSS'11]
  - 信頼できないドライバの保護。ドライバ用のメモリ空間とカーネル用のメモリ空間の分離。CR3の値を変える。
- Windows7の Driver Isolation
  - プリンタがメイン。
- OS2
  - IA32アーキテクチャの4つのリング構造でカーネルとデバイスドライバを分ける。

National Institute of Advanced Industrial Science and Technology AIST

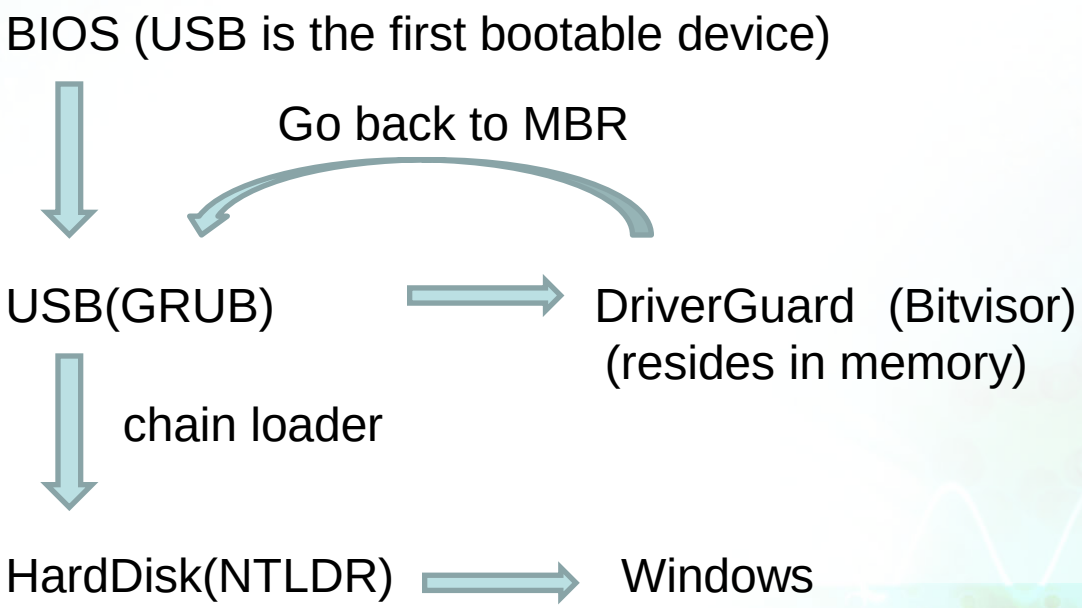
ドライバメモリを保護するハイパーバイザー

RISEC Research Institute for Secure Systems

- 既存OSへの挿入可能性、VM Introspectionを考慮してGreen Killerベースで開発。
  - DriverGurad と命名。
- 動作手順
  - USBからハイパーバイザーを起動後、ハードディスクOSの起動
  - ハイパーバイザーのドライバ認識
  - 保護するメモリ領域の認識
  - 攻撃に対する防御
    - Stealth Breakpoints技術

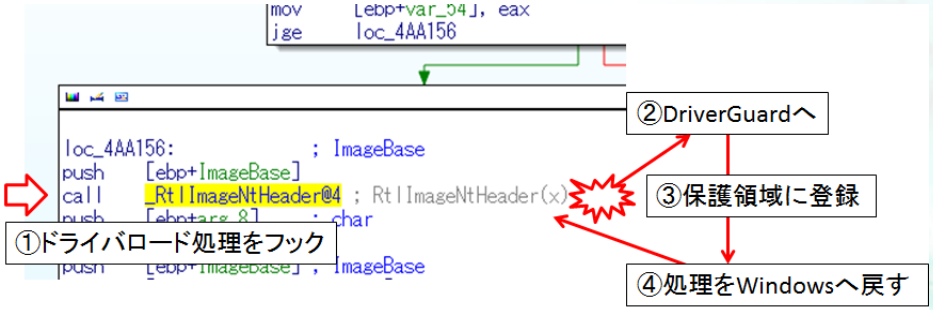
起動手順

- ハードディスクにインストール済みのOSに対して、起動時にDriverGuardを挿入する
  - Chain load技術
  - 類似の技術にはLinuxのkexcシステムコールを使ったkbootがある。PS3Linuxで使われていた。



現在の実装　1/2

- DriverGuardではGreenKillerによるWindowsの解析を利用
- 起動時に守るべきドライバの認識
  - ドライバのIDとしてはタイムスタンプをIDとする
  - ドライバはntkrnlpa.exeのLoadDriver関数を通るため、この関数をフック。
    - フックはint 03H (0xCC)命令例に置き換えることで例外をDriverGuardに通知。



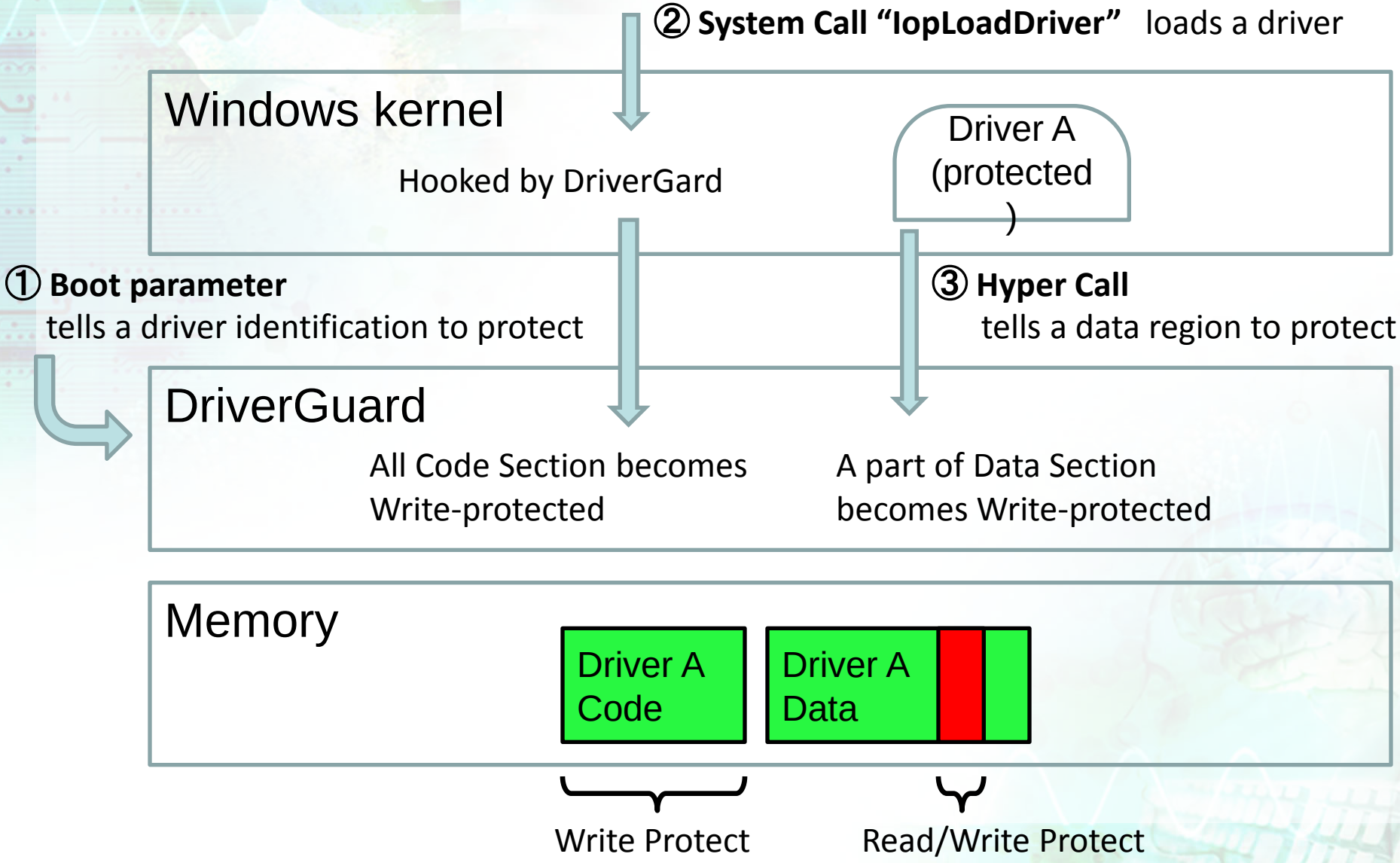
現在の実装 2/2

- 守るべきメモリ空間はドライバからのハイパーコールで通知
  - ドライバの改変が必要
- 現在対象のNonCopyではExAllocatePoolWithTagで保護対象のメモリを確保していたが、他の領域で保護対象にできる

National Institute of Advanced Industrial Science and Technology
AIST

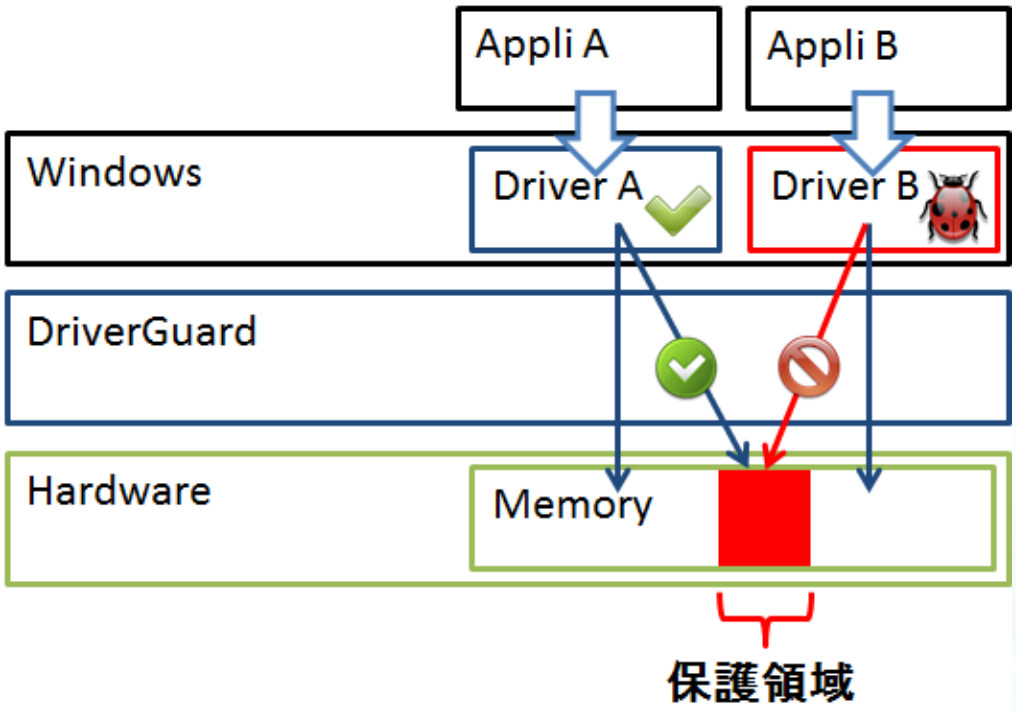
保護手順

RISEC
Research Institute for Secure Systems



攻撃モデル

- 本来のドライバ以外から攻撃を検出
  - カーネル内で同じメモリ空間のドライバからの攻撃



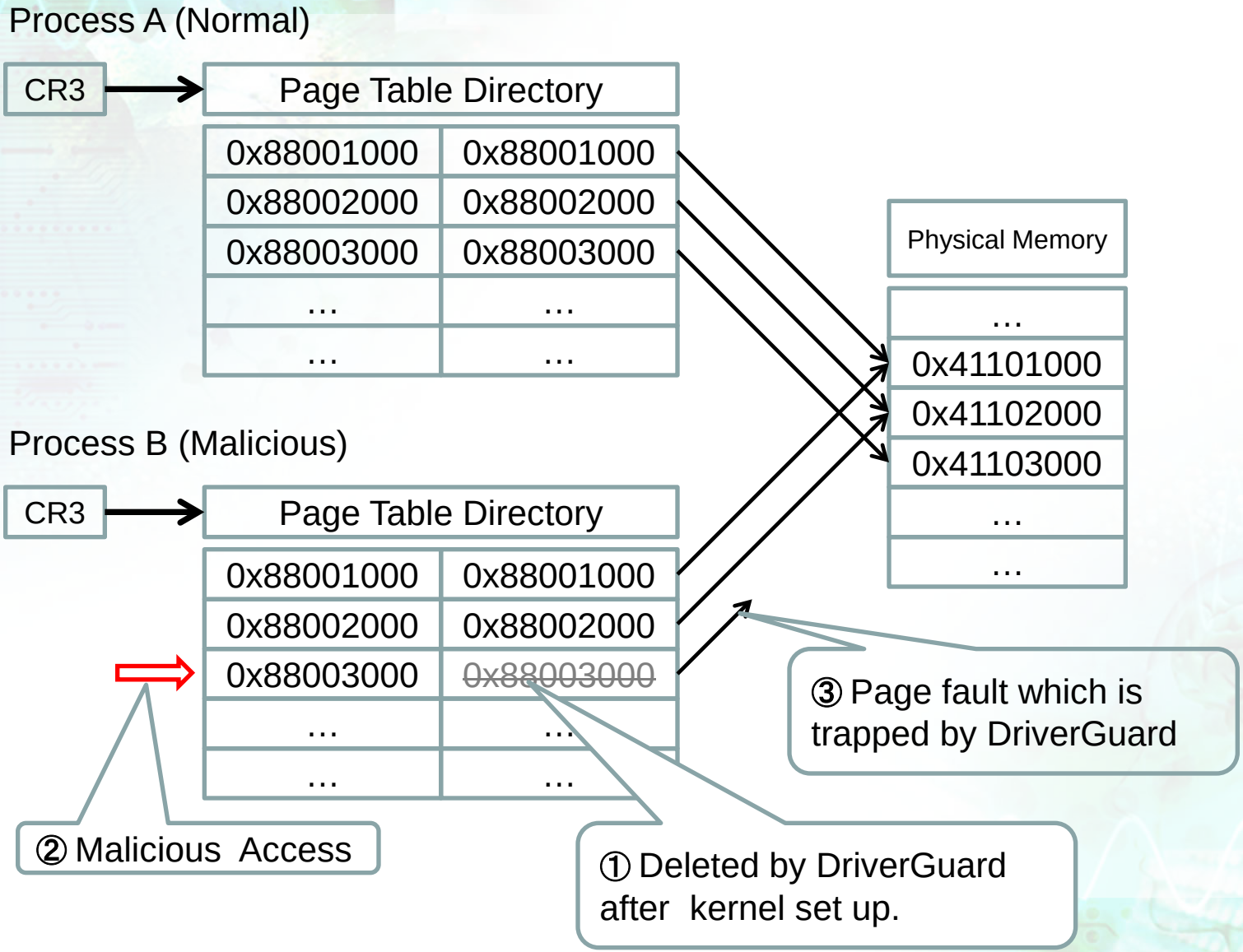
メモリの保護

- Stealth Breakpoints [ACSAC'05]の技術の活用
  - INT 03H (0xCC)に置き換えずにBreakpointの機能を実現。メモリ内容を変えないため、攻撃者から検出され難い。
  - ハイパーバイザーが守るべきメモリを管理するページテーブルを空にし、アクセスが起こるたびにページフォルトを起こす。
  - ページフォルトをハイパーバイザーが捉え、正しいアクセスか検証。問題が無ければアクセスを許す。
  - 問題点
    - 保護対象は4KBページ単位。
    - 保護対象外でも4KBページ内のアクセスは全て検証されてしまう。

National Institute of Advanced Industrial Science and Technology
AIST

Stealth Breakpoints

RISEC
Research Institute for Secure Systems



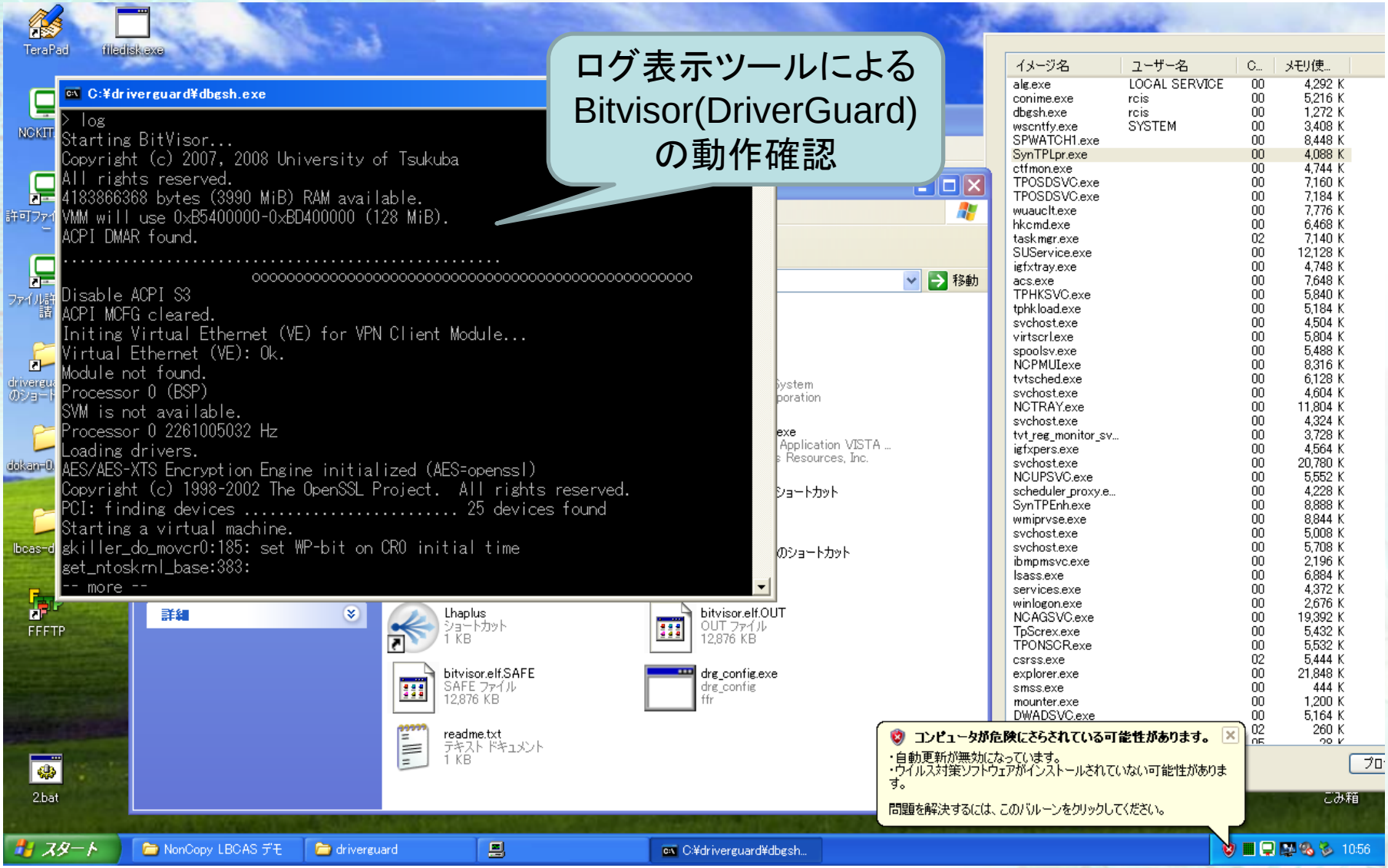
攻撃検出後の動作

- 問題を起こしたプロセスをとらえ、無限ループにする
  - Low IRQL(Interrupt ReQuest Level)で無限ループのすることにより、他のプロセスは実行可能。但し、現状ではCPUをかなり食うので他のプロセスは遅くなる。

National Institute of Advanced Industrial Science and Technology
AIST

動作例1

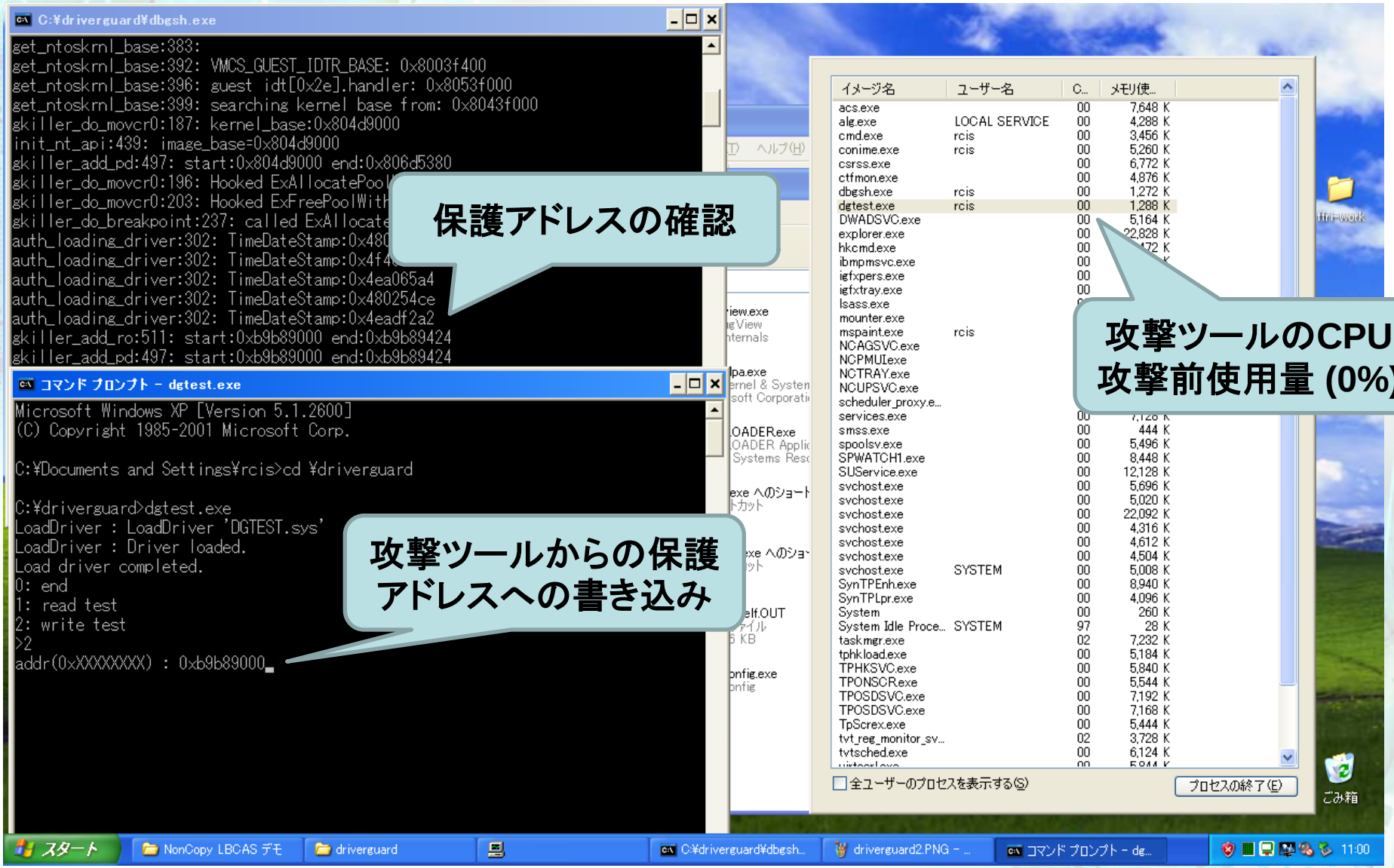
RISEC
Research Institute for Secure Systems



National Institute of Advanced Industrial Science and Technology
AIST

動作例2

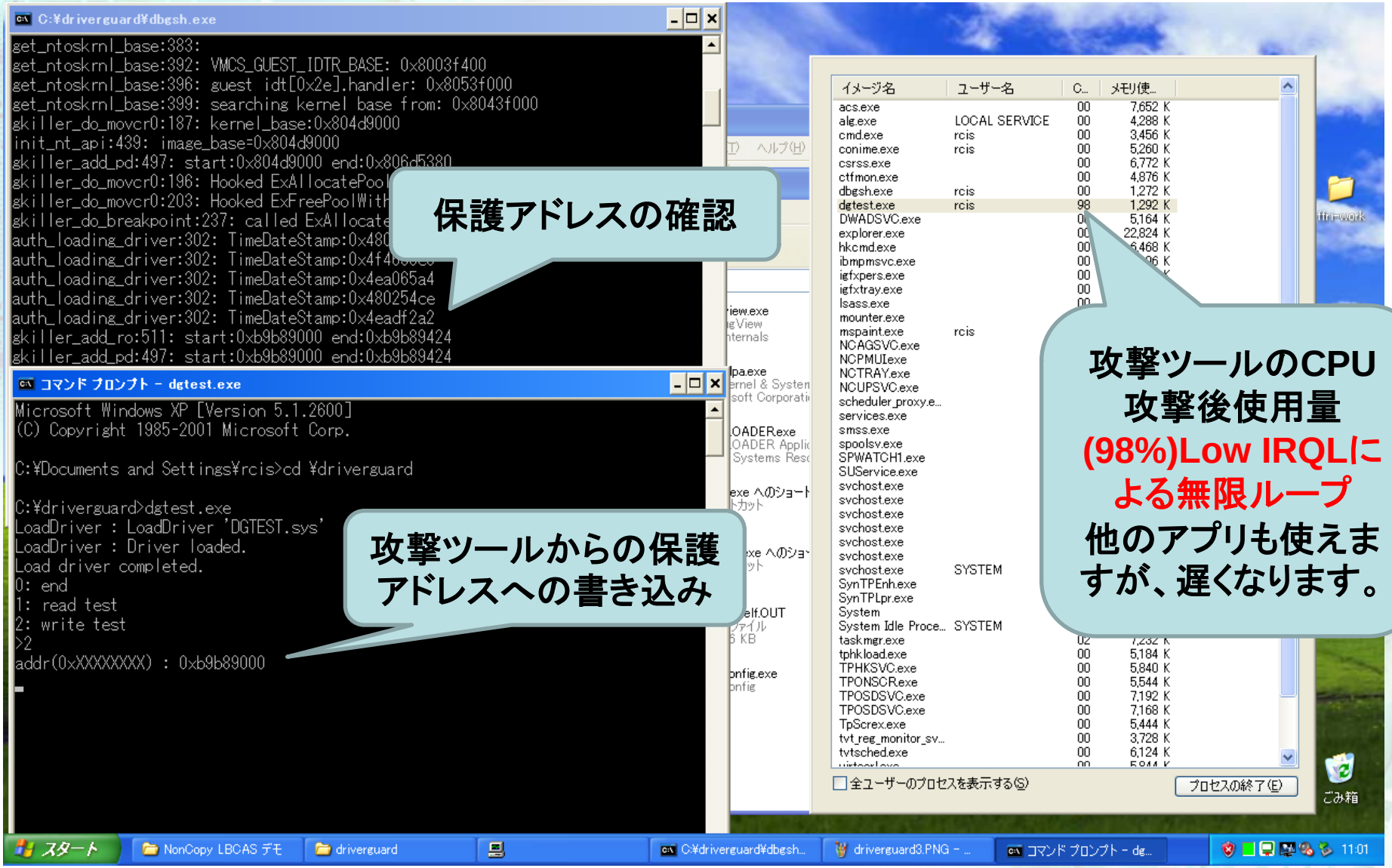
RISEC
Research Institute for Secure Systems



National Institute of Advanced Industrial Science and Technology
AIST

動作例3

RISEC
Research Institute for Secure Systems



National Institute of Advanced Industrial Science and Technology AIST

現在DriverGuardの制約事項

RISEC Research Institute for Secure Systems

- OSはWindowsXP SP3のみ
- BIOSの設定によりマルチプロセッサをDisabled（無効）とする
- OSの設定により仮想メモリ機能を無効の状態とする
- OSの設定によりPSE（Page Size Extension）を無効の状態とする
- 保護領域のアドレス通知をすべて受け取るまでは保護領域は守れない
- OSであるWindowsXP SP3をアップデートした場合は動作を保証できない

現状の課題

- ドライバのIDとしてタイムスタンプは不適切
  - 実装が容易だが、改ざんされる恐れあり。
- ドライバでHyperCallする必要がある
  - ドライバの改良が必要
- 警告確認がCPUの負荷の向上で判り難い
  - ユーザ空間にハイパーバイザーの状態をポーリングする手法も考慮したが、これが改ざんされる(検出漏れ)恐れがある

National Institute of Advanced Industrial Science and Technology AIST

改良予定

RISEC Research Institute for Secure Systems

- ドライバの識別子としてドライバコードのSHA1とする
  - ハイパーバイザーでSHA1の計算が必要
- ドライバの守るべき領域をヒープメモリとする
  - ヒープメモリをハイパーバイザーが実行時に検出する必要あり
  - ヒープは頻繁にアクセスされないと想定
- OSと非依存の警告
  - 悪意のあるアクセスを検出した場合、OSと非依存の警告
  - ハイパーバイザーがビープ音を鳴らすことやビデオドライバへの直接警告する
    - Bitvisor 1.3で入った機能。(電通大の大山先生のADvisor)
      - 現状では対象ビデオカードが限定。Intel 945GMはちょっと古い？

National Institute of Advanced Industrial Science and Technology AIST

まとめ

RISEC Research Institute for Secure Systems

- Windowsを対象としたドライバメモリの情報漏えいを防ぐ、DriverGuardの提案
  - Bitvisorベースとすることで、任意のインストール済みOSに対して、起動時にハイパーバイザーを挿入する
  - Driverを理解するためにVM Introspection機能のあるGreenKillerの活用
  - Stealth Breakpointsによるアクセス検出