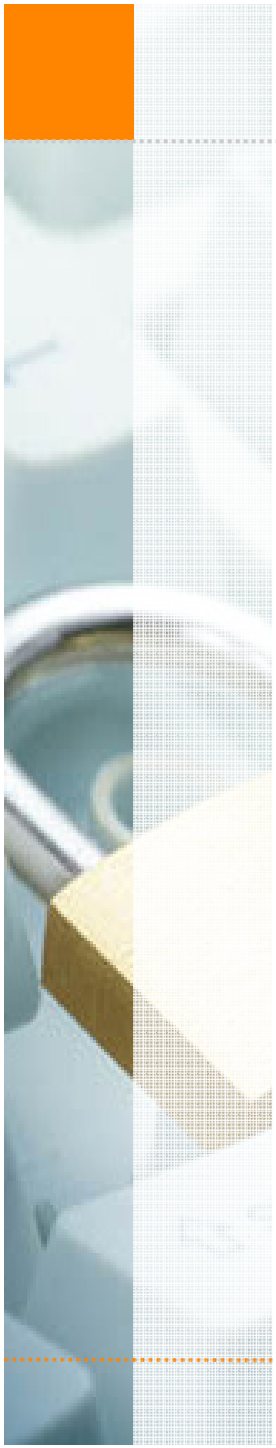
Device Driver Vulnerabilities

Exploiting VMware Tools HGFS.sys

Daniel Roethlisberger
daniel.roethlisberger@csnc.ch

Compass Security AG
Glärnischstrasse 7
Postfach 1628
CH-8640 Rapperswil

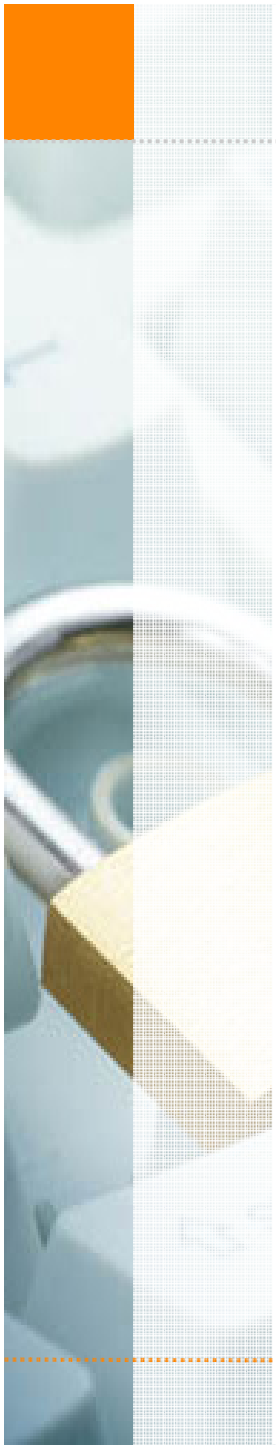
Tel.+41 55-214 41 60
Fax+41 55-214 41 61
team@csnc.ch
www.csnc.ch



Is Windows Update sufficient?

Agenda

- Introduction
- Driver Vulnerabilities Explained
- Exploiting VMware Tools HGFS.sys
- Demo 1: Local Workstation
- Quick Intro to Hash Injection
- Demo 2: Terminal Server
- Mitigation
- Discussion



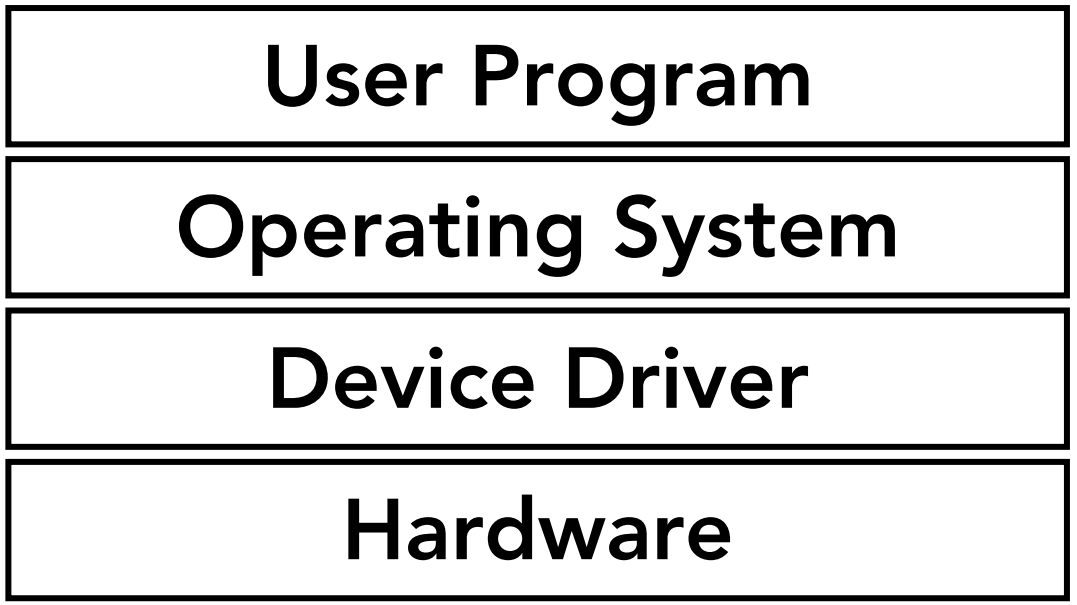
Introduction

Why Attacking Device Drivers?

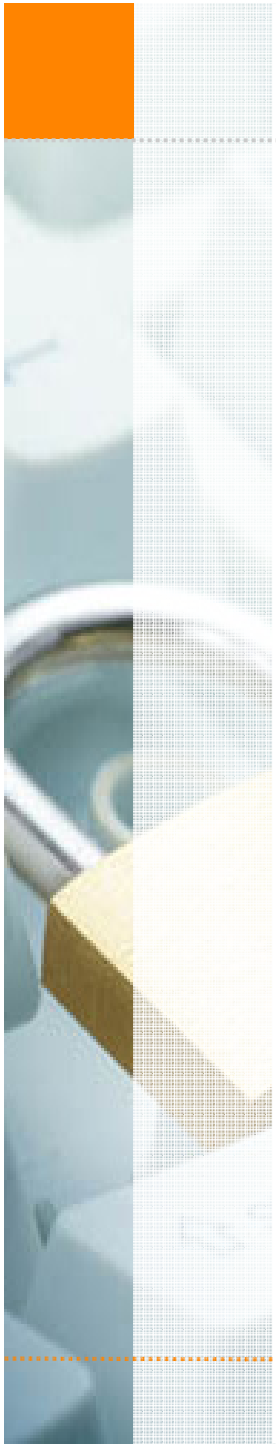
Introduction: Device Drivers

Device drivers:

- Are part of the OS, running with kernel privileges
- Usually talk to hardware



Introduction: Device Drivers

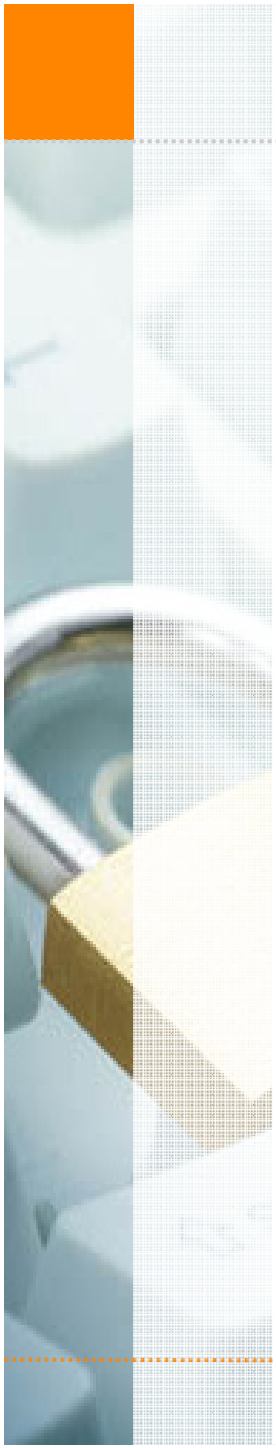


Vulnerabilities lead to:

- **Remote code execution**
  - Remote attacker over Ethernet, WiFi, Bluetooth, IrDA
- **Local privilege escalation**
  - Local workstation users
  - Malware
  - Terminal Server users

→ **Focus of this presentation: local driver vulnerabilities**

Introduction: Device Drivers



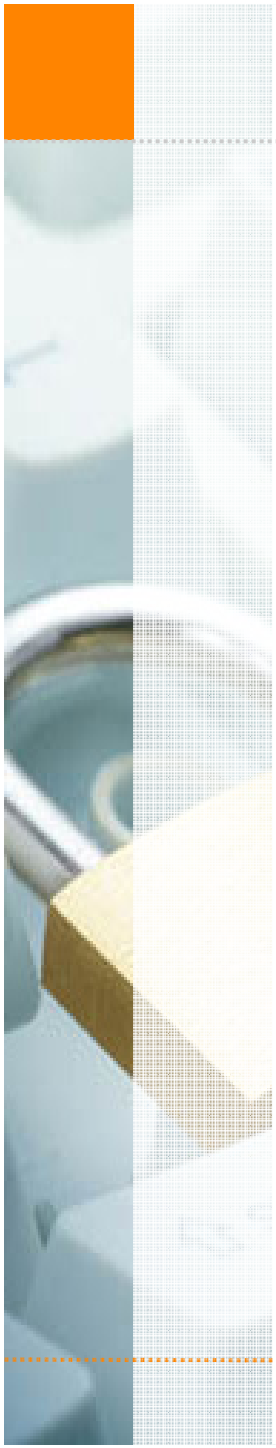
Drivers are frequently:

- Shipped by 3rd-party vendors
- Written by hardware folks, not kernel developers
- Of bad quality
- Neglected in software patch management

→ Many driver vulnerabilities discovered recently!

Introduction: Device Drivers

- **Device driver vulnerabilities in security products:**
  - Kaspersky Internet Security (2008, 2006)
  - Panda Internet Security / Firewall / Anti-Virus (2008)
  - Trend Micro Anti-Virus (2007)
  - Norton/Symantec Anti-Virus / Internet Security (2007, 2006)
  - ZoneAlarm Firewall (2007, 2003)
  - Computer Associates HIPS, Anti-Virus (2006)
  - avast! Anti-Virus (2005)
- **Windows default driver vulnerabilities:**
  - Windows XP/2003 i2omgmt.sys (2008)
  - Windows XP/2003 ndistapi.sys (2007)
  - Windows XP/2003 mrxsmb.sys (2006)
- **Other operating systems affected as well:**
  - Linux driver for Omnikey SmartCard reader (2007)

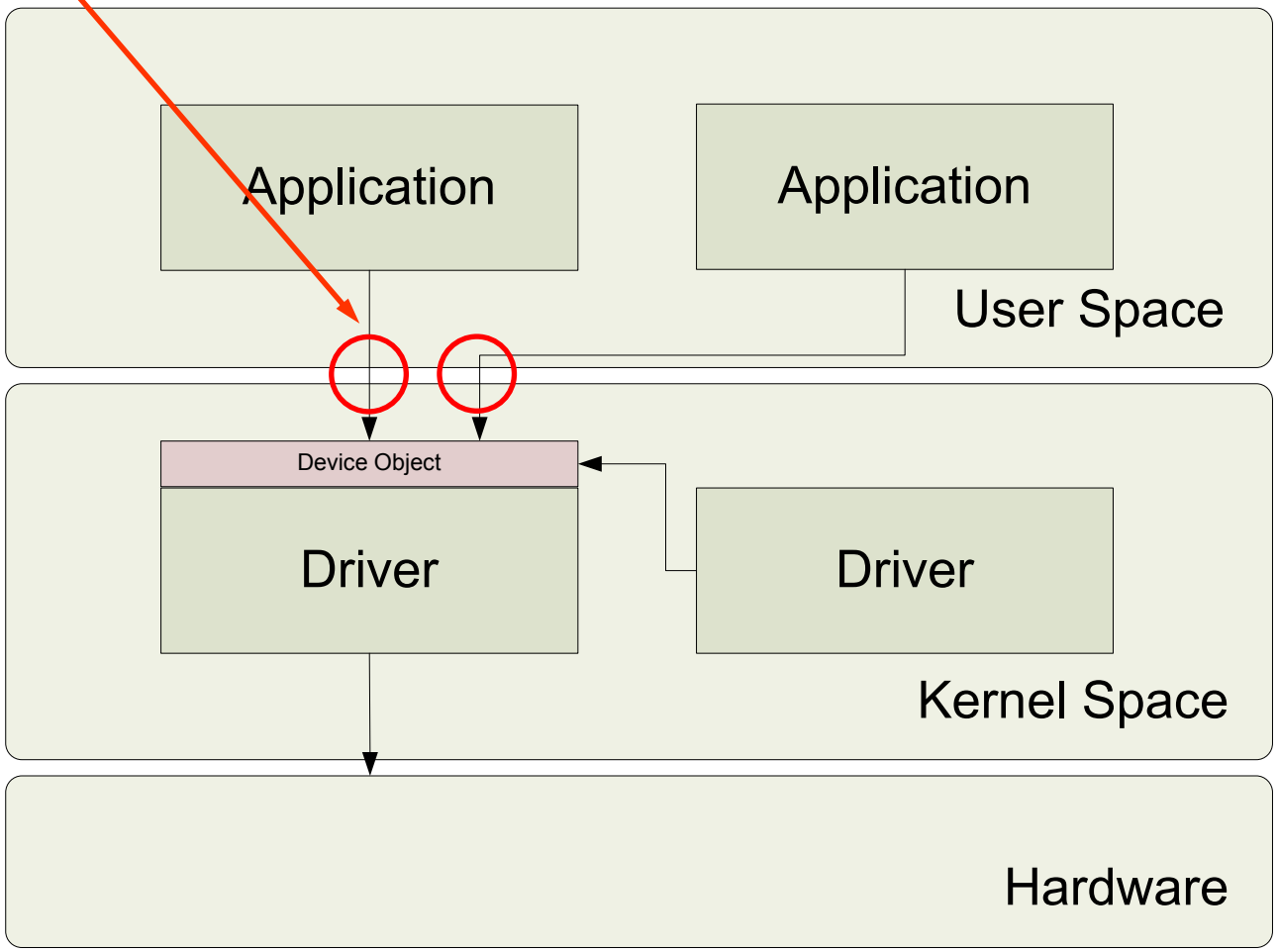


Driver Vulnerabilities Explained

How to Attack Device Drivers

Driver Vulnerabilities Explained

DeviceIoControl



Driver Vulnerabilities Explained

- Device object = virtual file

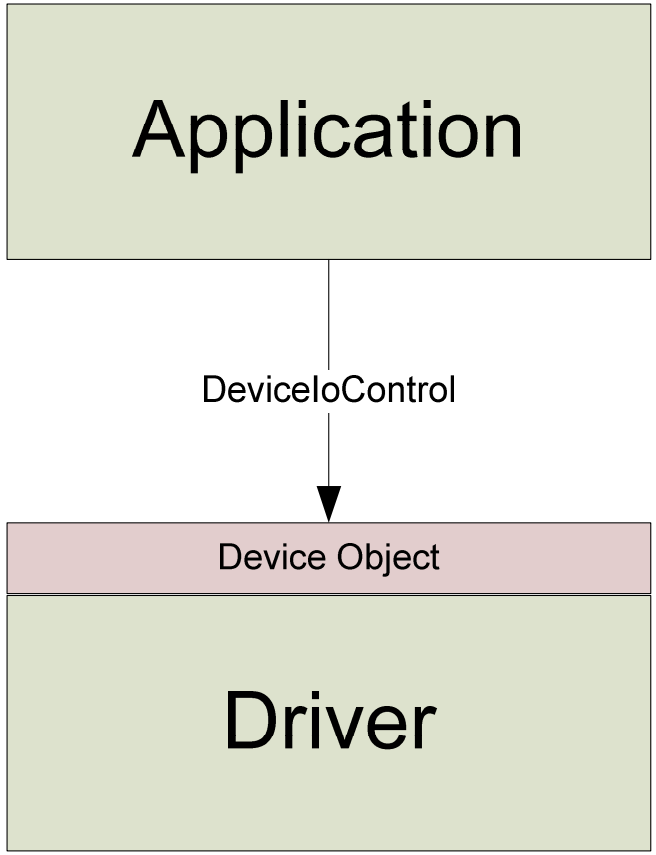
```
\Device\MyDriver
\\.\MyDriver
```

- Communication with driver
- Created by driver:

```
IoCreateDevice("\Device\MyDriver");
IoCreateSymbolicLink("\\.\MyDriver");
```

- Accessed by application:

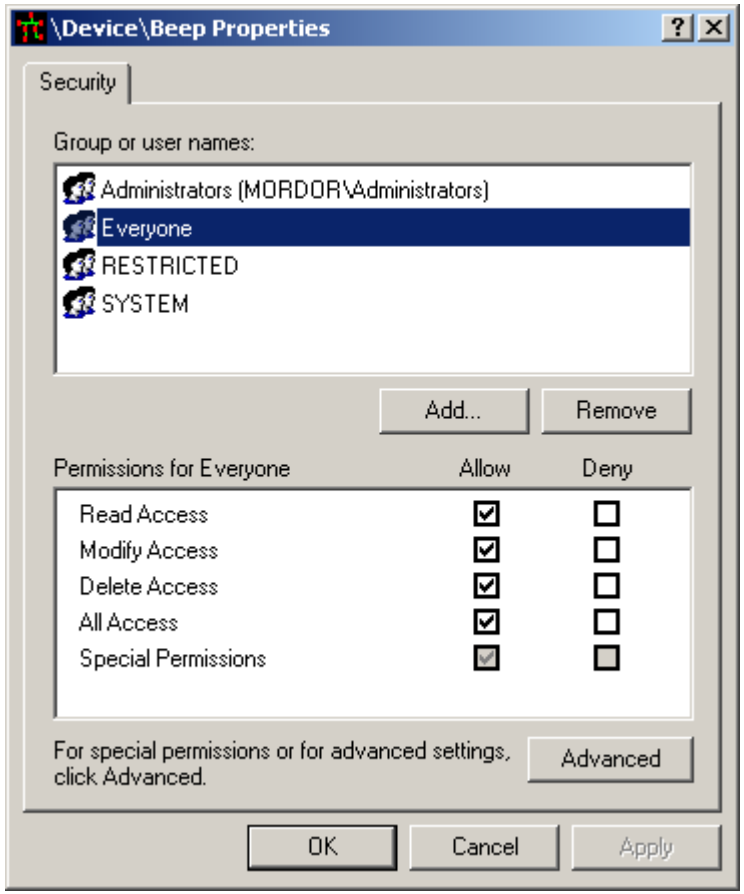
```
handle = CreateFile("\\.\MyDriver");
DeviceIoControl(handle, COMMAND, ...);
CloseHandle(handle);
```



Driver Vulnerabilities Explained

Driver access control:

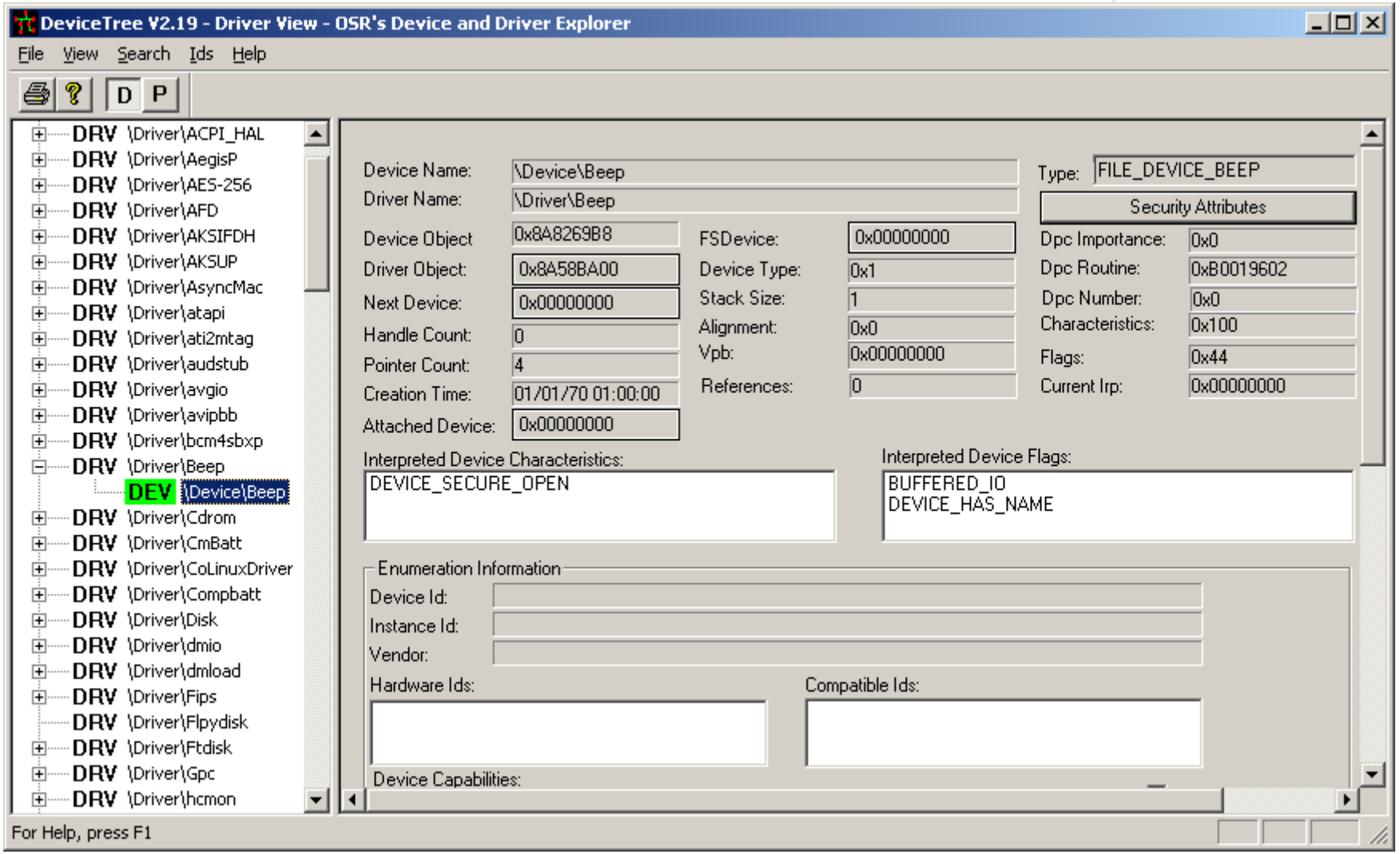
- Device objects have Access Control Lists (ACLs)
- Only harmless device objects should be accessible to users



To inspect driver access control:

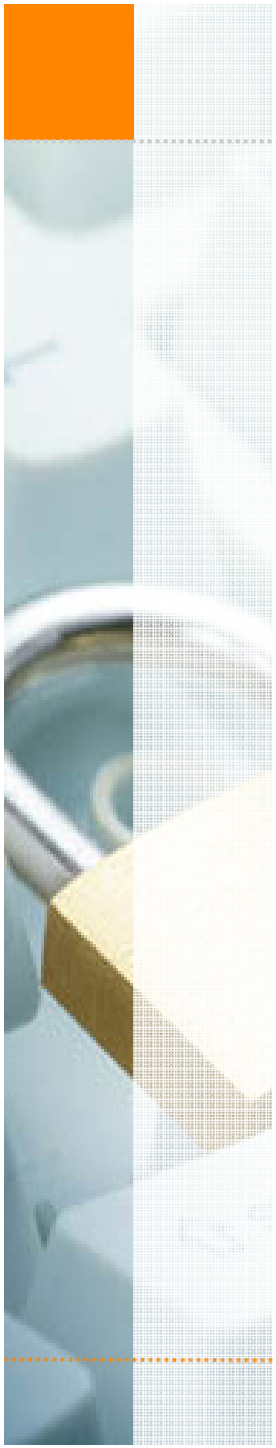
- DeviceTree utility
  - Drivers
  - Device objects
  - Device ACLs

Driver Vulnerabilities Explained



Driver Vulnerabilities Explained

- **Each driver is responsible for security checks**
  - Sanity checks on data from user space apps
  - ACL setup / enforcement
- **We have a vulnerability, if:**
  - Device ACL allows access to "Everyone"
  - Privileged operations can be executed
    - by design
    - by accident: buffer overflows, writing to user supplied memory addresses, ...
- **Exploit (local privilege escalation):**
  - Open vulnerable device object as normal user
  - Send malicious data to device object
  - Malicious code executed with kernel privileges
  - → Gain SYSTEM privileges



Exploiting VMware Tools HGFS.sys

Gaining SYSTEM Privileges

VMware Tools HGFS.sys

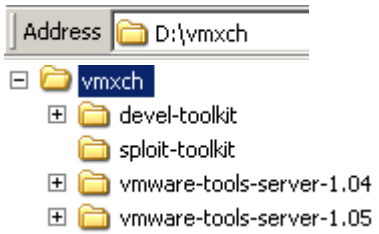
VMware Tools

- Drivers and utilities installed within guest
- Host to guest time sync, performance, GUI usability
- Deployed on all VMs
- Not patched using standard update mechanisms

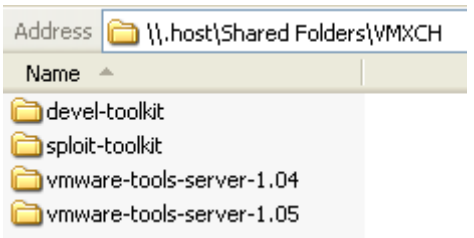
HGFS: Host-Guest File System

- Access files on host from guest
- Supported only on VMware Workstation
- Driver present on VMware Server and ESX as well

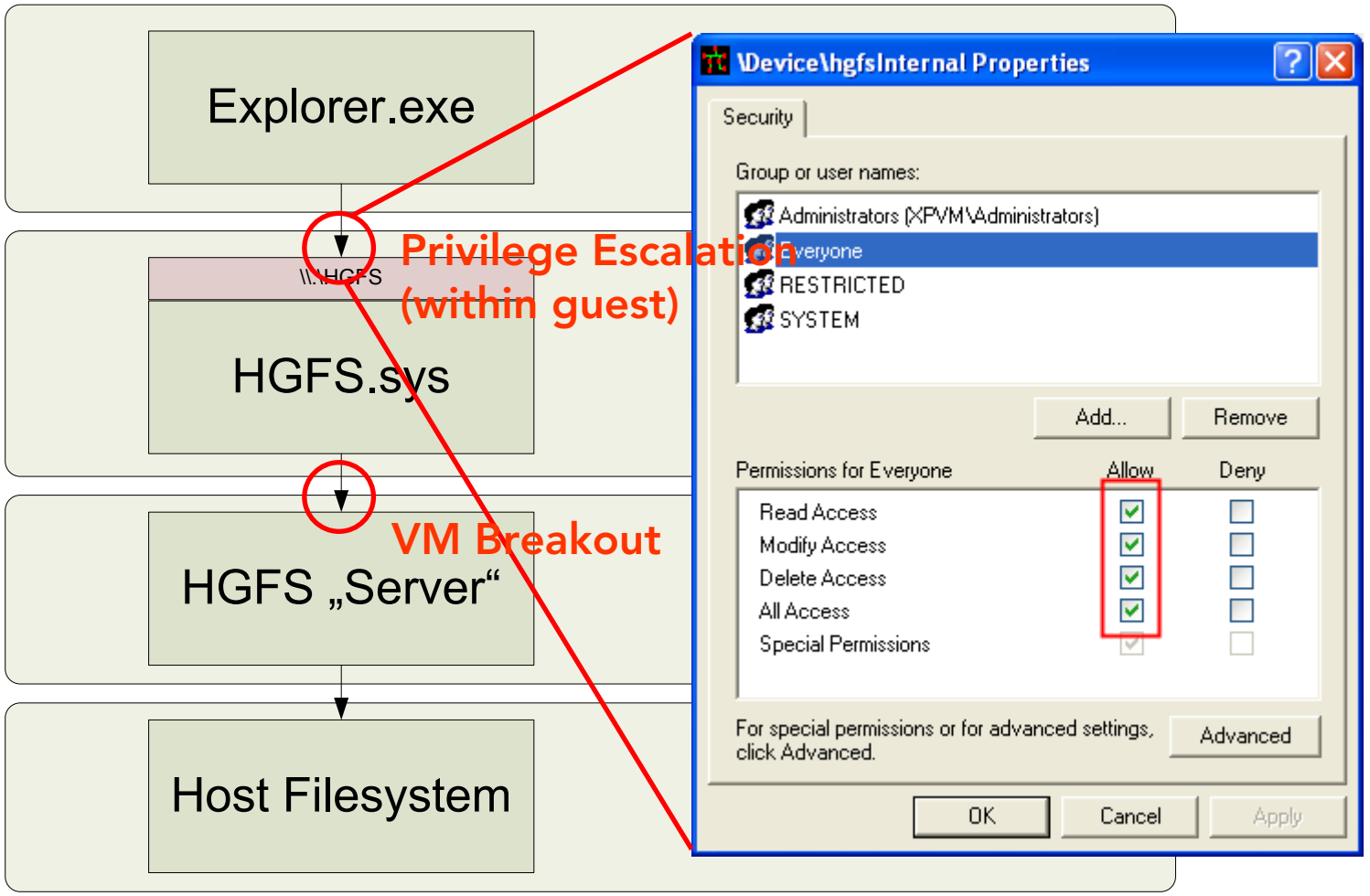
- on host:



within guest:



VMware Tools HGFS.sys



VMware Tools HGFS.sys

- **Vulnerability published without all technical details / no exploit**

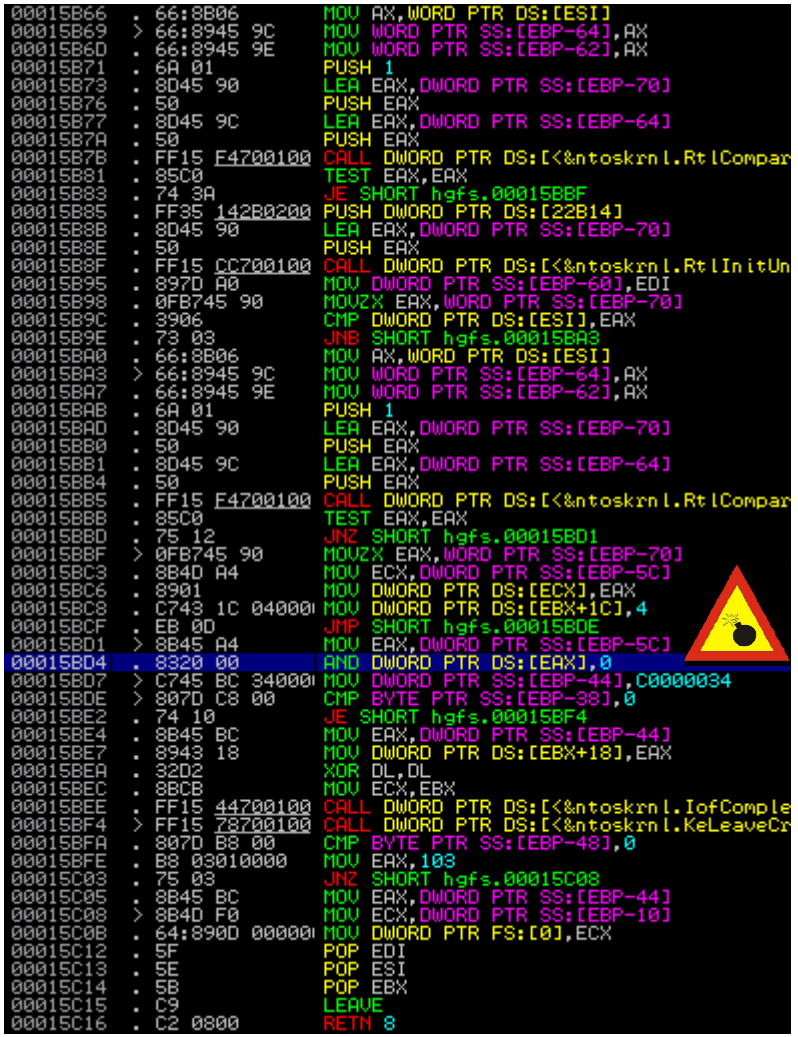
> **II. DESCRIPTION**
>
> Local exploitation of an input validation vulnerability within VMware's Hgfs.sys driver could allow an unprivileged attacker to execute arbitrary code within the kernel of a Windows guest operating system.
>
> When a VMware guest operating system has the VMware Tools package installed, the hgfs.sys driver is loaded on the machine. This driver allows any user to open the device "\\.\hgfs" and issue IOCTLs with a buffering mode of METHOD_NEITHER. This allows untrusted user mode code to pass kernel addresses as arguments to the driver.
>
> With specially constructed input, a malicious user can use functionality within the driver to patch kernel addresses and execute arbitrary code in kernel mode.

- **Steps to exploit the vulnerability:**
  - Setup lab environment
  - Locate the vulnerability
  - Analyze the vulnerability
  - Write an exploit

VMware Tools HGFS.sys

- ACL: "Everybody" may open device object and send requests to driver
- Driver fails to validate input from user space for certain IoControl commands
- Driver writes zero to a memory address
- Memory address under full control of caller

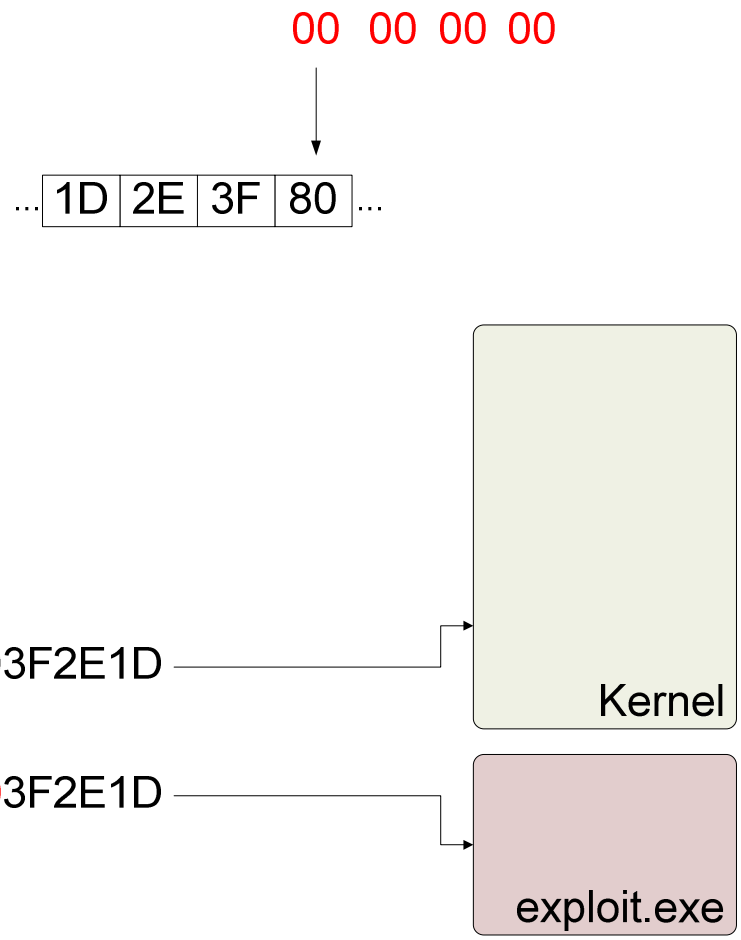
→ Write NULL to arbitrary kernel memory address

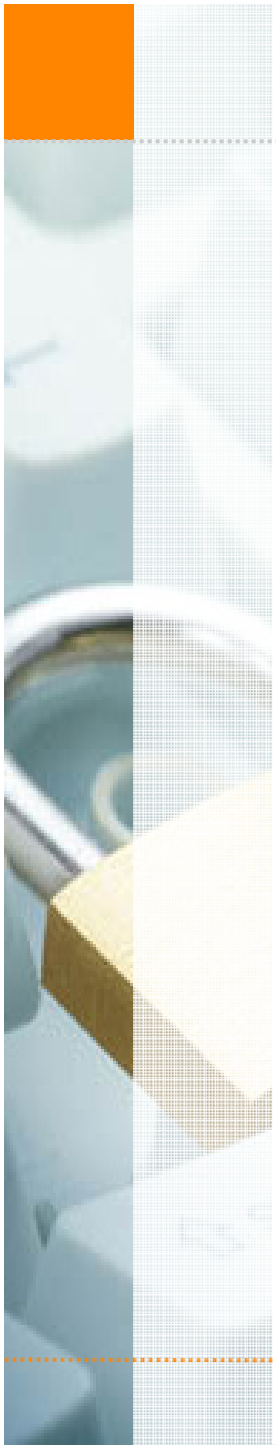


VMware Tools HGFS.sys

- Find address used as jump target by kernel
- Partially overwrite with NULL
- Place shellcode at modified address
- Wait until kernel jumps to modified address

→ Exploit shellcode run by kernel

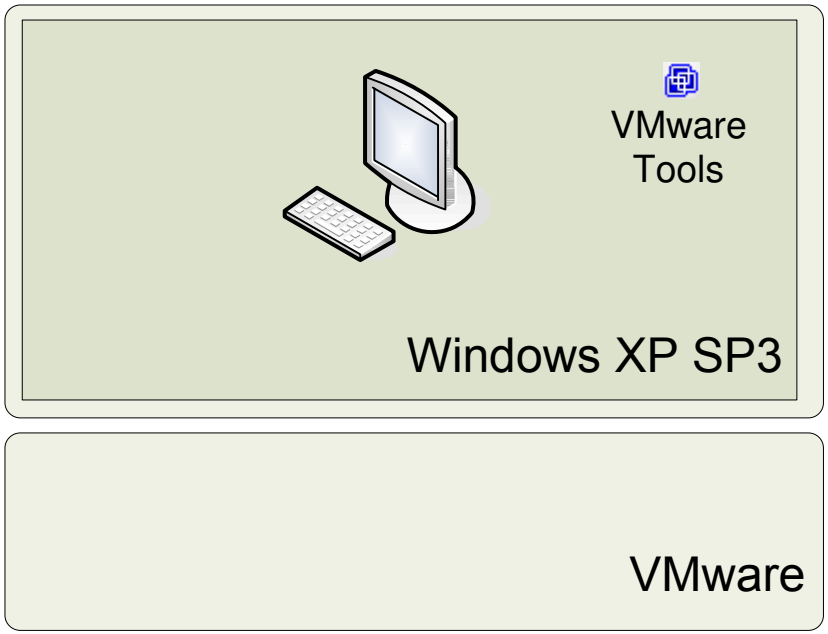


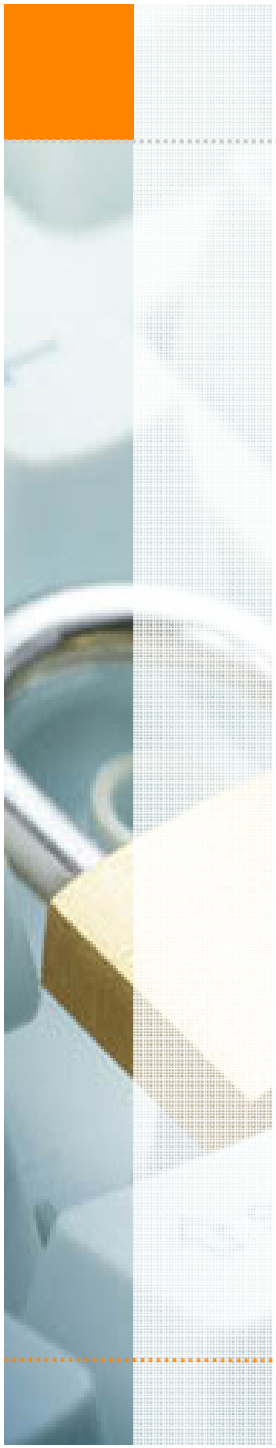


Demo I

Local Workstation

Demo I: Local Workstation

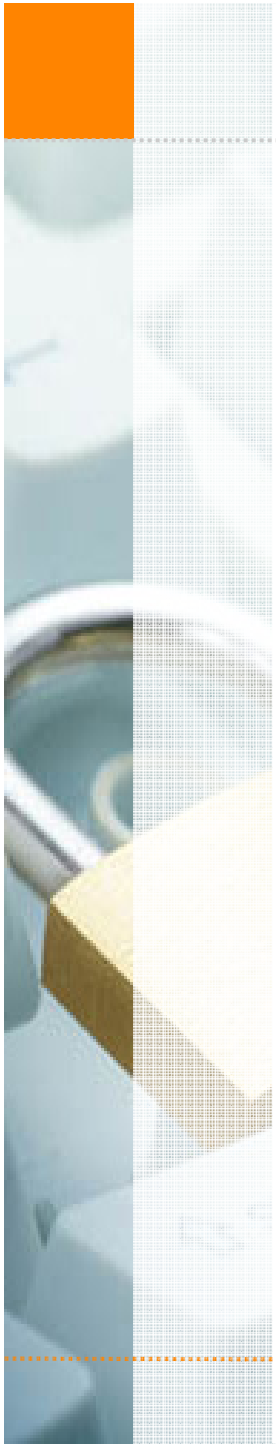




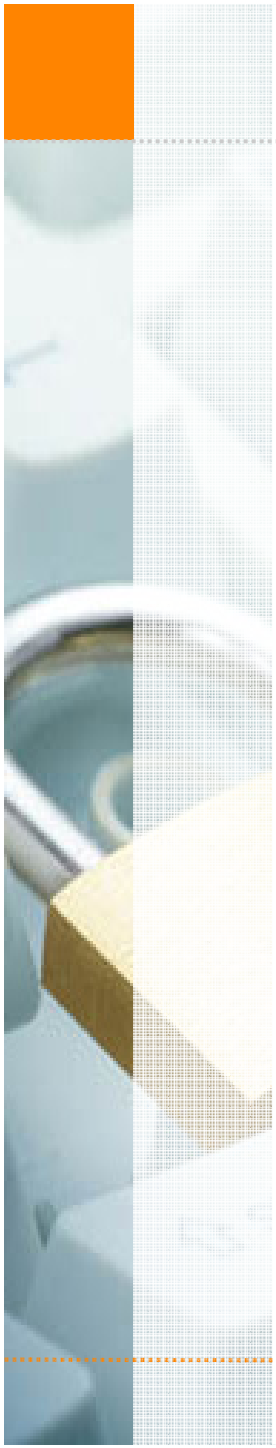
Hash Injection

A Quick Introduction

Hash Injection



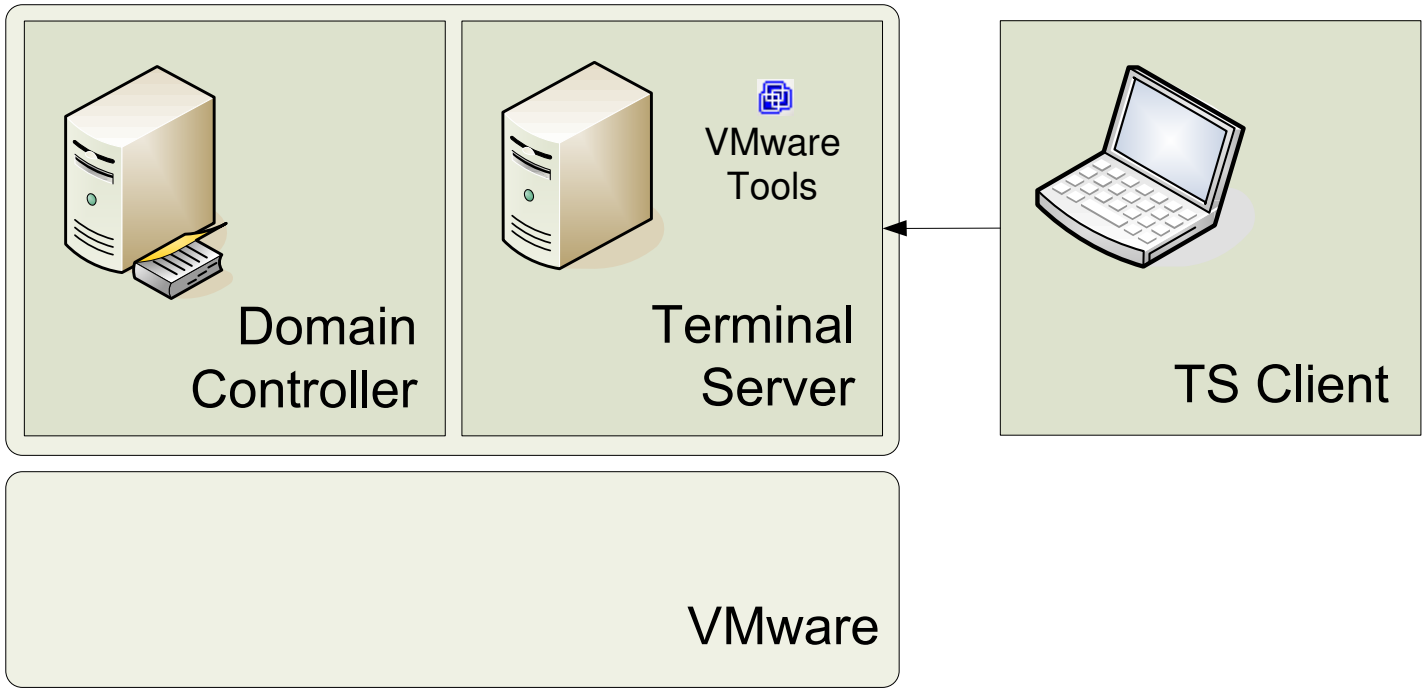
- **Windows uses cryptographical hashes of user passwords: LM/NTLM hashes**
- **Windows stores LM/NTLM hashes locally, even for AD users**
  - AD user hashes are cached until reboot
- **Stored hashes can be dumped**
  - Requires local admin or SYSTEM privileges
- **Attack 1: Offline dictionary attack to find password**
- **Attack 2: Inject hashes into running process; use hashes to authenticate in the domain without knowing the password**

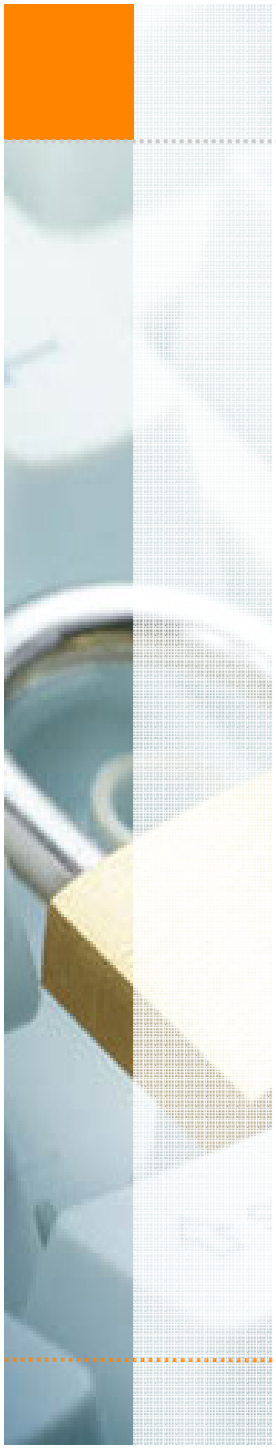


Demo II

Terminal Server

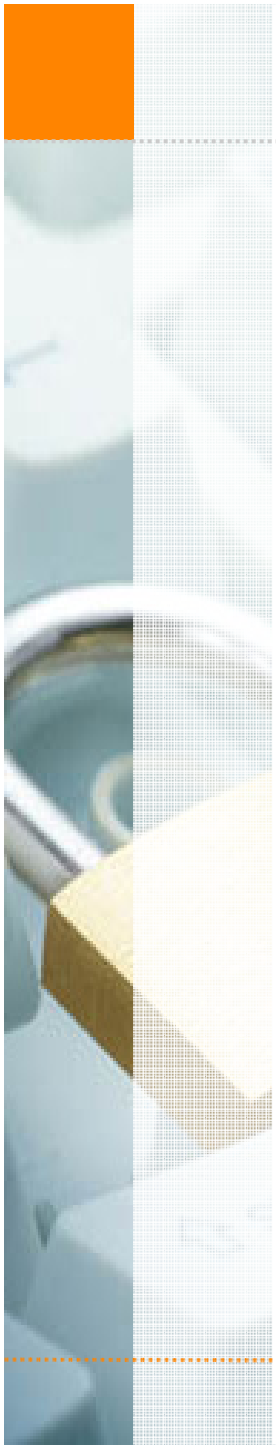
Demo II: Terminal Server





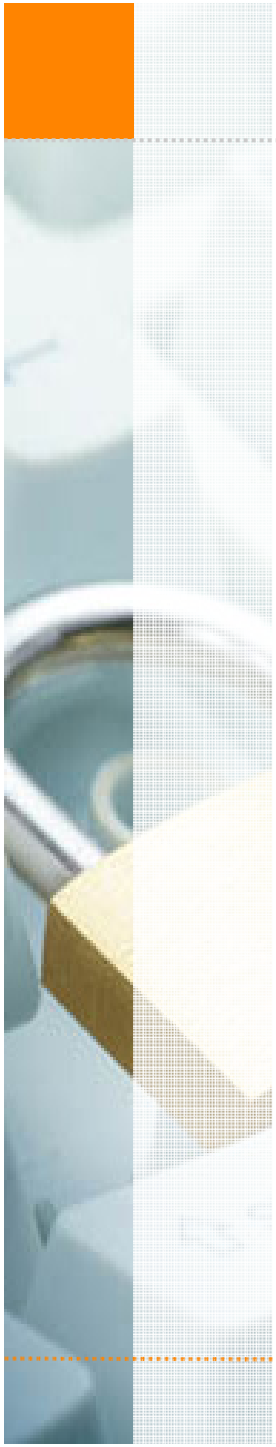
Mitigation

Lessons Learned



Is Windows Update sufficient?

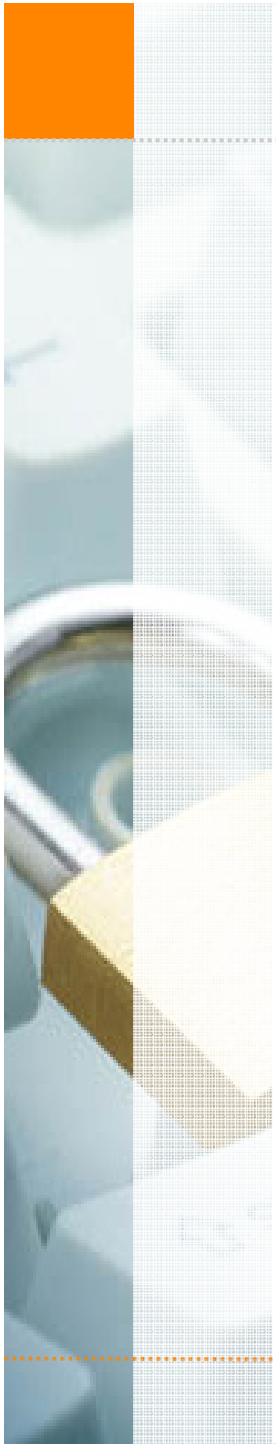
Mitigation: Driver Vulnerabilities



Windows Update is not sufficient!

- **3rd-party drivers require patch management too**
- **Prefer hardware natively supported by Windows**
  - No 3rd-party drivers
  - Updated as part of OS patch management
- **Audit remaining 3rd-party drivers for vulnerabilities**
- **Driver security as hardware evaluation criterion**
  - Quality of drivers also means security
  - Vendor's security history
  - Vendor's responsiveness to driver bugs

Mitigation: Terminal Servers



- **Patching (of course)**
- **Restrict which software terminal server users may run**
  - White list, not black list
  - No development utilities
  - No scripting languages (e.g. Python)
- **Secure web access / perimeter security**

Mitigation: Hash Injection

- **No local admin rights for regular employees**
  - Developers can be local admin on separate systems not in the domain
- **Only use domain admin accounts to logon to domain controllers; never logon to member servers or workstations**
  - Use dedicated admin accounts for member server administration
- **Monitor audit event ID 552**
  - Triggered by hash injection
  - Triggered by legitimate services as well

References

- VMware Tools HGFS Local Privilege Escalation Vulnerability
  http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=712
- OSR DeviceTree by Mark Cariddi:
  http://www.osronline.com/article.cfm?article=97
- Kartoffel – Secure Your Driver
  http://kartoffel.reversemode.com/
- Remote and Local Exploitation of Network Drivers
  https://www.blackhat.com/presentations/bh-usa-07/Bulygin/Presentation/bh-usa-07-bulygin.pdf
- Hash Injection Attack
  http://www.csnc.ch/static/download/Hash_Injection_Attack_E.pdf