

Web Exploit Finder

**Detecting Drive-By-Downloads
in a virtualized environment**

Benjamin Mack
xnos Internet Services

About Benjamin Mack

- Student of „computer science and media“ at the Hochschule der Medien, Stuttgart since 2003
- Started xnos Internet Services for Hosting, Security and Web Development in 2006
- Finishing my diploma thesis in late 2007
- Also involved in TYPO3 core development

Agenda

- The Problem
- What are malicious websites?
- Our Approach
 - Features
 - Architecture
 - Rootkit
 - Fast reproduction of virtual clients
 - Inspecting a website
- State & Future Plans

What is the Web Exploit Finder?

- Developed by Thomas Müller, Mehmet Arziman and Benjamin Mack in Summer 06
- Student project from the Hochschule der Medien, Stuttgart
- Now hosted, developed and supported by xnos Internet Services

Introduction

- A lot of software connects to the internet
- Security threats occur through remote code execution after buffer overflows
- Can happen to every piece of software

The Problem

- Focus on internet browsers
- Both Microsoft Internet Explorer and Mozilla Firefox still include several vulnerabilities
- Primary user interfaces to the WWW
- Browsers are used most frequently
- Many non-technical users

The Problem

- Many users don't install security updates
- Even fully patched systems are vulnerable to zero-day exploits
- Unknown amount of malicious sites on the web

How can we find these malicious websites?

The Problem

- What is „malicious“?
- How can we detect malicious web content?
- How can we design an adequate system?

What is malicious?

- A website that downloads and installs a malicious software (virus, trojan horse) on the local system without any user interaction.
- so-called „Drive-By-Downloads“
- No phishing attacks

How does a hacker achieve this?

- Attacker executes his code in the browser through a buffer overflow

- Execution code is limited

- Only a small „Dropper“ or „Downloader“ is run which retrieves the malicious software
 - Starts new processes
 - Modifies the registry
 - Writes files to the hard drive

Worst Case

- Windows XP Professional w/o Service Packs
- No security updates installed
- Windows running as an Administrator
- Using Microsoft Internet Explorer 6
- Scripting and Java both activated

How can we detect malicious software?

- Two techniques
 - **Intrusion Detection**
Compare the state of the system before and after a visit to a website
 - **Rootkit**
Monitor suspicious actions in real-time modifying the operating system

How can we design an adequate system?

- The system should be
 - automatic, require little user interaction
 - controlled remotely, with a web interface
 - scalable and extensible
 - secure, ensuring that the system itself cannot be infected by malicious websites

System Architecture

- Virtualization layer
 - protect the system
 - check multiple websites simultaneously

VMware Server

- Client OS component
 - modify the operating system
 - monitor system calls

Hand-made Rootkit

System Architecture

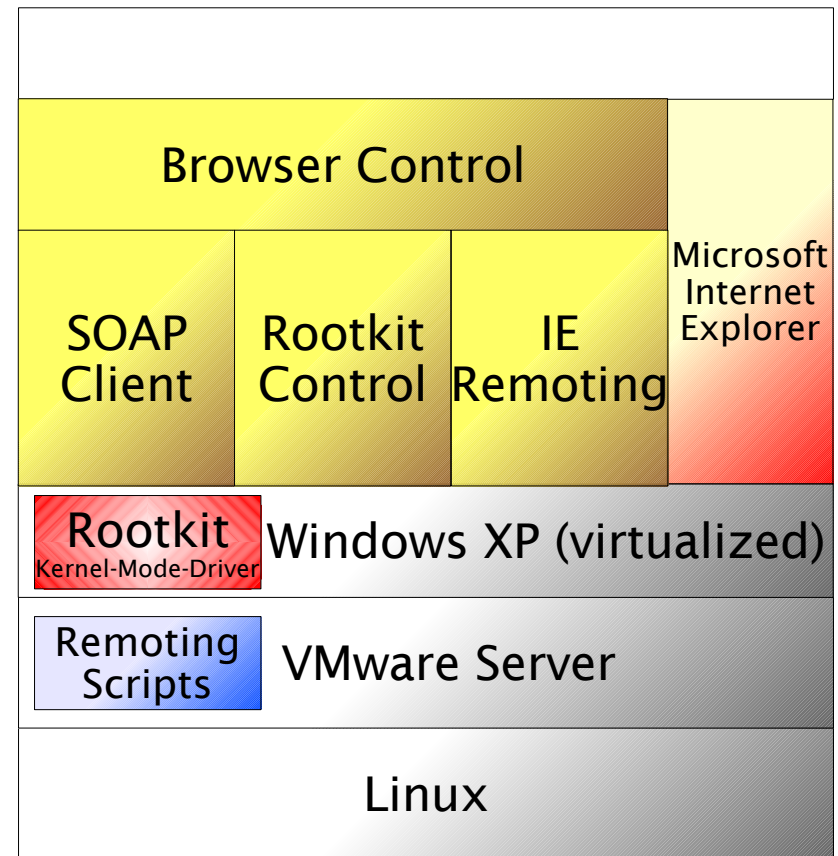
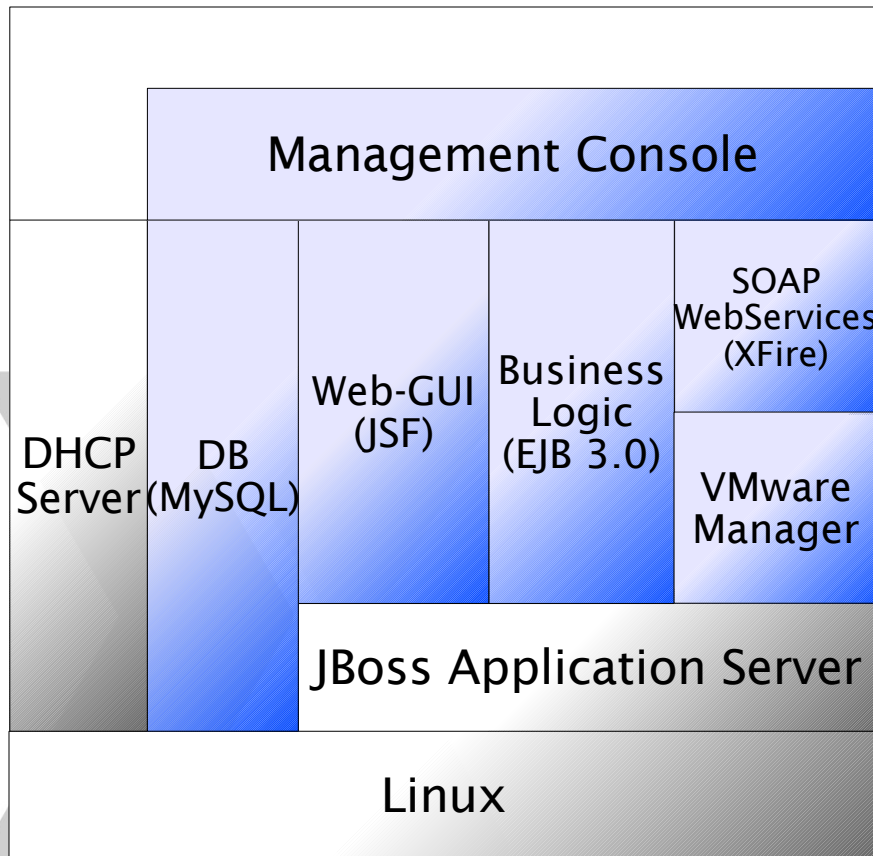
- Browser Control
 - manage the rootkit
 - control the browser
 - communicate with the management console

Windows MFC Application

- Management Console
 - configure and control the system
 - monitor system calls

JBoss Application Server

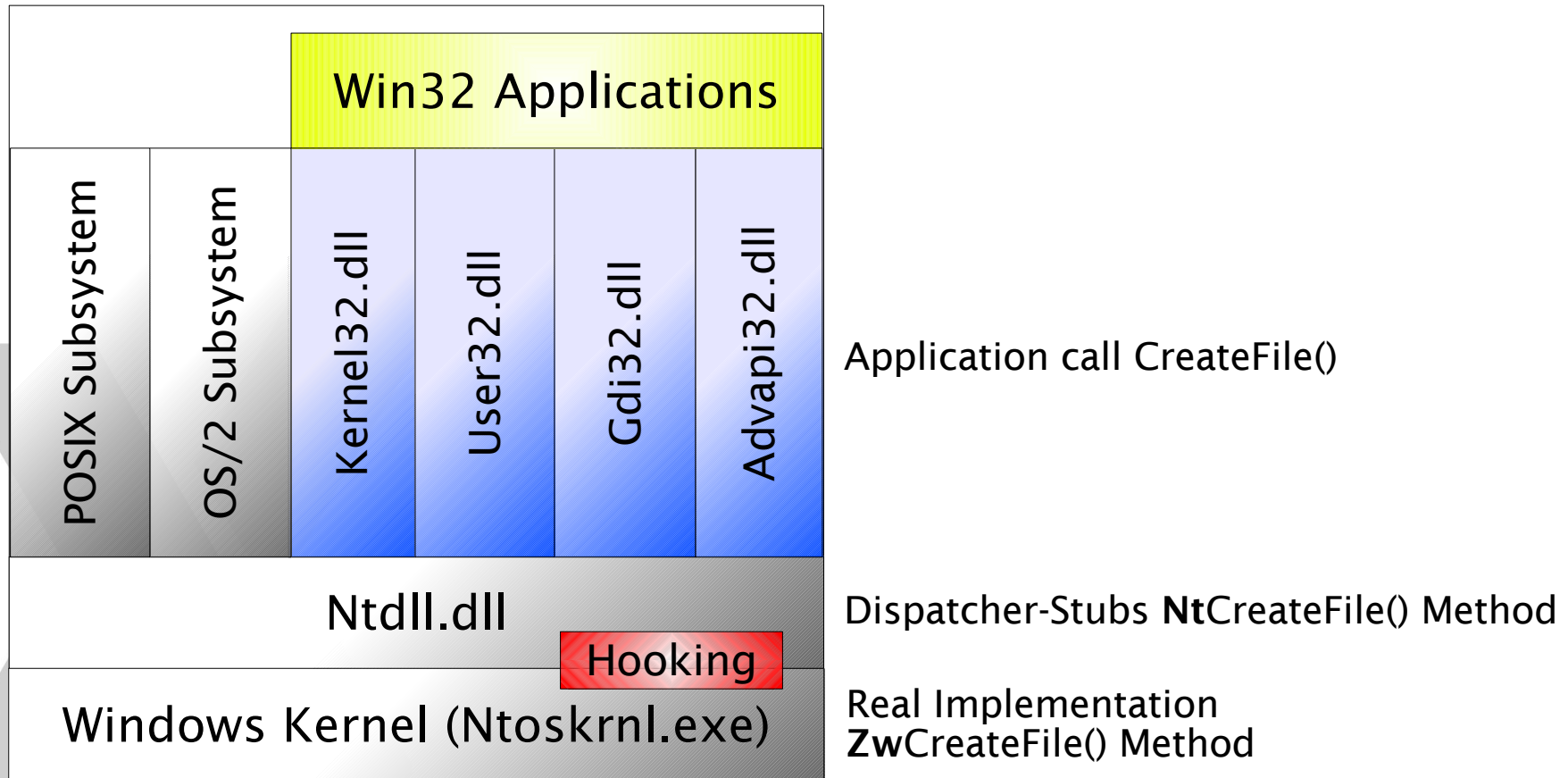
System Architecture



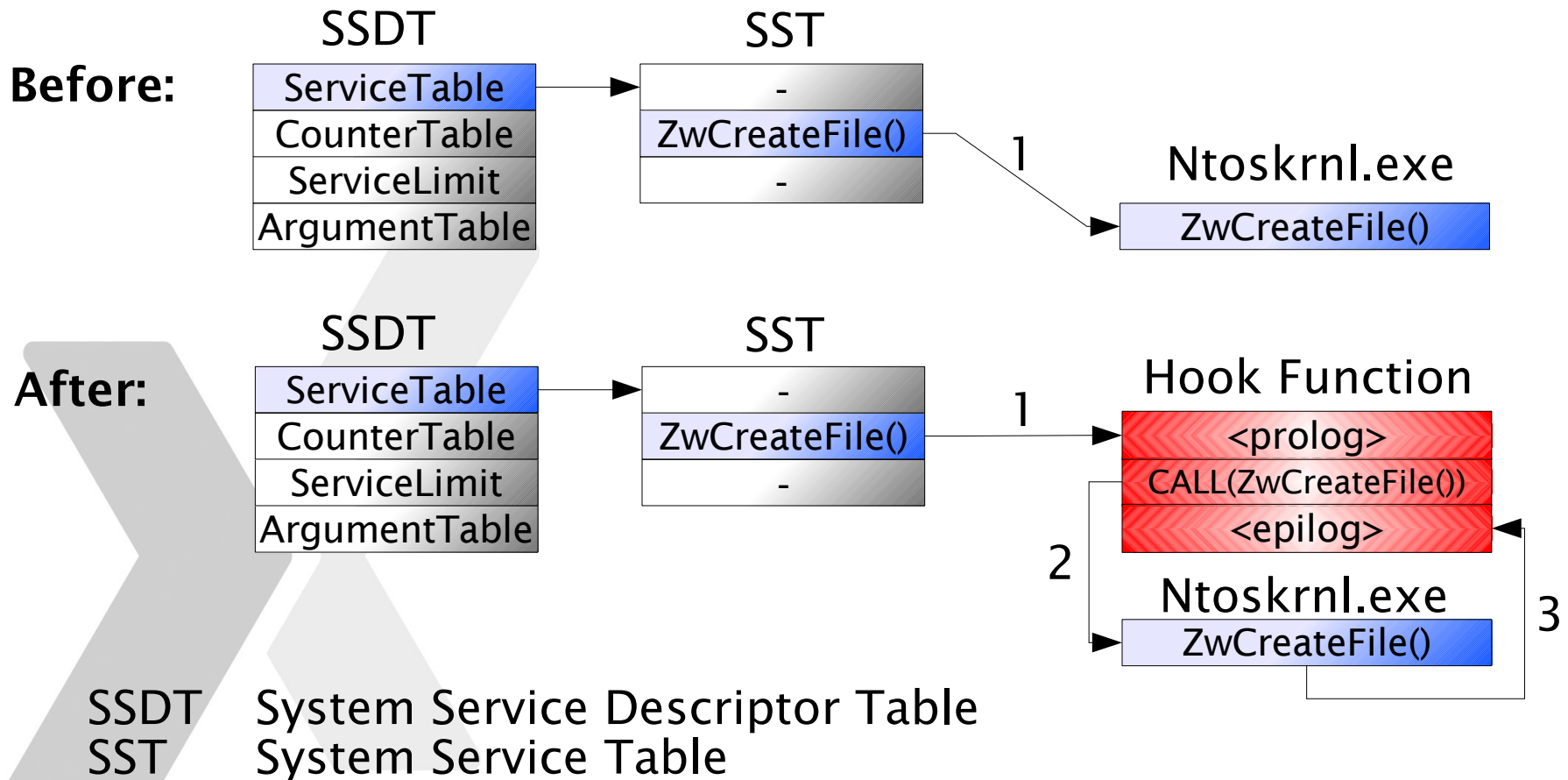
Rootkit

- SSDT-Hooking
 - Redirects the system call
 - Access to the protected memory of the kernel
- Implemented as a system driver in C

The Windows API



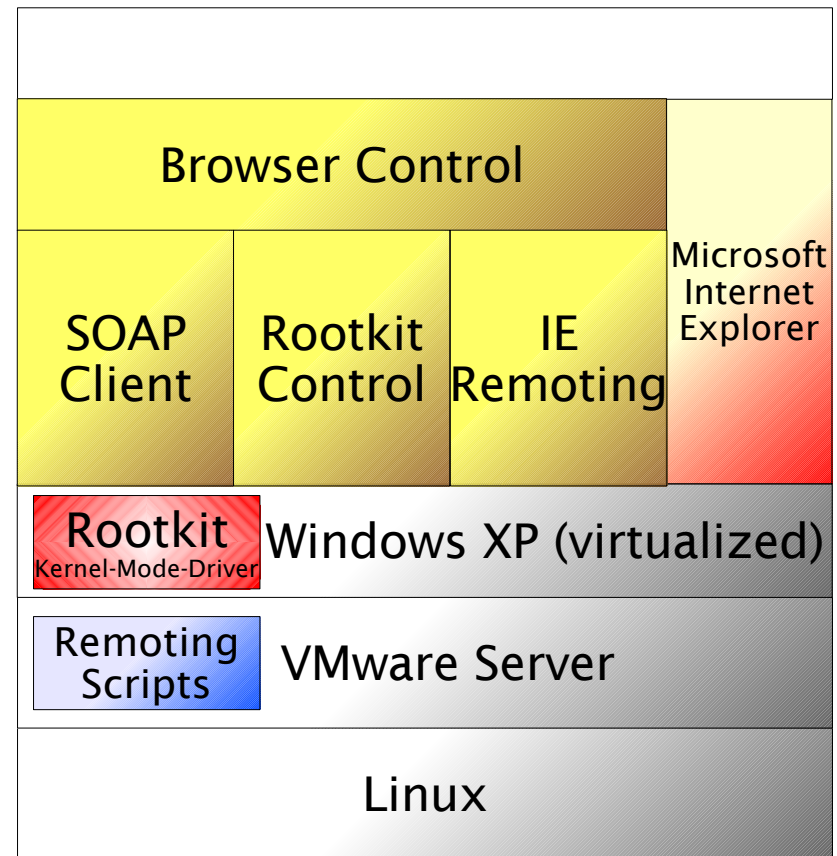
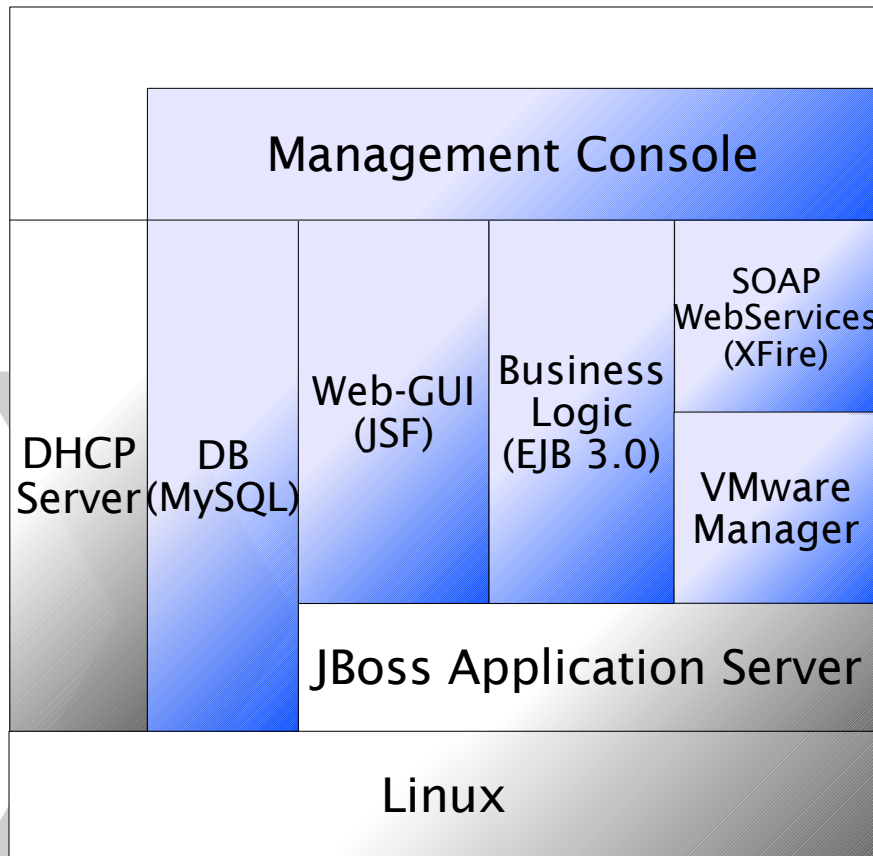
Kernel Rootkit: SSDT Hooking



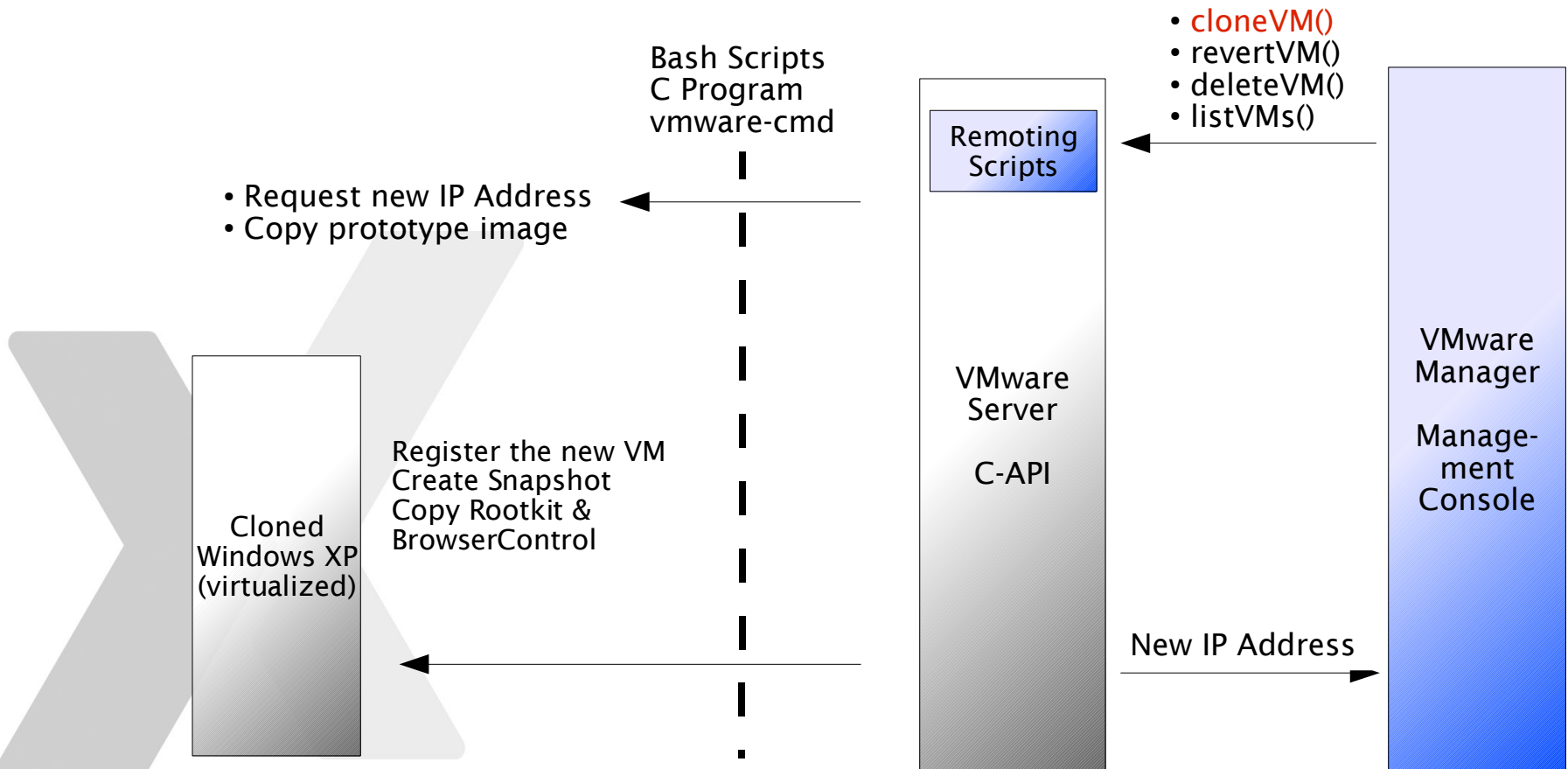
VMware Manager

- Our virtualized environment needs to...
 - Create a new virtual machine
 - Clone from a clean template
 - Copy the most recent version of the rootkit
 - Take a snapshot to revert fast
 - Revert to a clean state
 - Delete a virtual machine

Creating a new Virtual Machine



VMware Control



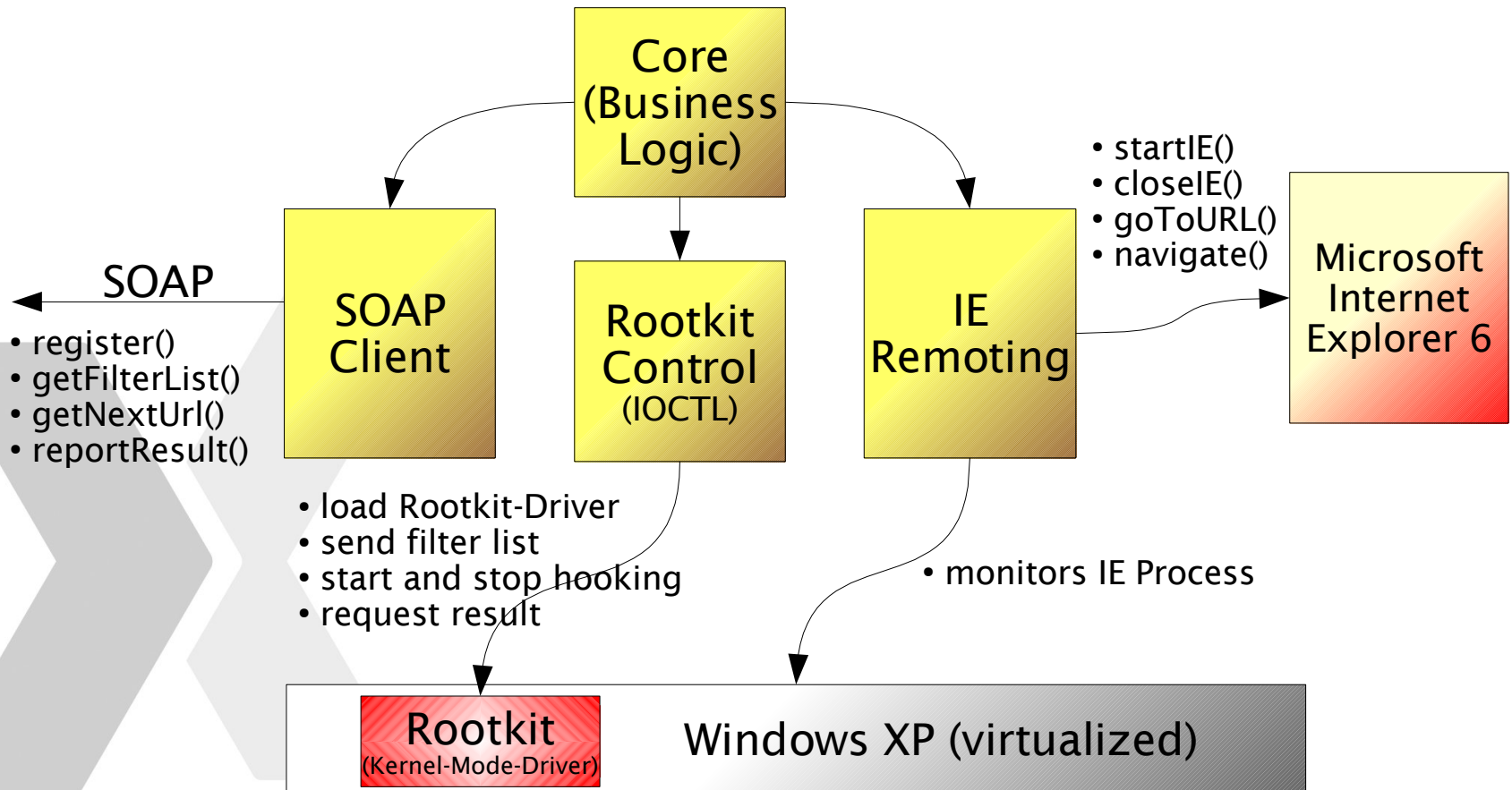
Browser Control

- Communicate with the Management Console
 - Get URL to check
 - Tell if website was malicious
 - use SOAP calls (gSOAP)

- Communicate with the Rootkit
 - Start & Stop Hooking
 - Configure Rootkit
 - Request Results After Delay

- Run the Browser

Browser Control



Management Console

- Web Interface
 - Display running VMs
 - Manually add URLs
 - Create more virtual machines
 - Manage filters
- Web Crawler
 - Automatically add more URLs to check all of them
 - Store in database
- Database holding all URLs and running VMs
- SOAP interface to the VMs

State of the system

- Beta phase
 - The system works
 - Rootkit needs some small adjustments

- Implementing the crawler

- Web interface rewrite

- Hook more Windows system calls

Future Plans

- First open-source release in the next weeks
 - including a complete manual to set the software up
 - will be available on www.xnos.org
- New Features
 - Try different IE versions
 - Use Firefox and Opera

Future Plans contd.

- Different virtualization technologies
 - Xen (for Windows with HVM)

- Different operating systems as clients
 - Windows Vista (32 bit)
 - Linux

- Cooperating with other client honeypot projects for evaluating the malicious software on the websites

Support wanted

- System has a lot of potential
 - Dutch government and a couple of big companies want to use WEF already

- We need developers once the software is released as open-source

- We need support
 - Either by testing the package or
 - by sponsoring the developers

Questions

Any Questions?

xnos Internet Services
Benjamin Mack

Gartenstraße 29
70563 Stuttgart

Phone +49 711 508 85 44 22
Fax +49 711 508 85 44 29

mack@xnos.de
www.xnos.de