Rootkit Programming

Sergej Proskurin, Thomas Kittel, Julian Kirsch

Technische Universität München
Chair for IT-Security
Munich, Germany

25.01.2016
Goal: to learn about rootkits from a security perspective and to learn about the Linux kernel internals

You will learn:

1. what a rootkit is
2. Linux kernel principles and Linux LKM programming
3. how rootkits work from a practical perspective
4. How to detect and analyse rootkits
5. Maybe also Windows Rootkit principles
What is a rootkit?
A *kit* (i.e., group of programs or functions) that allows an attacker to maintain *root* access.

What specific roles does a rootkit have?

1. provides a backdoor or way back into the system
2. hides files, connections, etc that provide this backdoor
3. overtime the term has been perverted and there are often additional elements implemented into a rootkit
We will be working with Debian 8 and a 64-bit vanilla kernel

- system call hooking
- file hiding
- process hiding
- module hiding
- socket hiding
- privilege escalation
- networking from the kernel
You will work in teams of two.
There will be weekly programming assignments.
Plus a final project.

You must have a background programming in C
  - the kernel is written in C and all assignments will be done in C.

Additional “required” background
  - Linux OS principles
  - IT security principles
Rootkit Programming

- Time & Place

Tue 14:00 - 16:00 in MI 01.05.013

Registration

- Matching System, but:
- You will only be granted access if you write a motivational letter.
- Latest, until Fri, Feb 04 2016 to kittel@sec.in.tum.de!