Rootkit Programming

Fabian Franzen, Sergej Proskurin

Technische Universität München
Chair for IT-Security
Munich, Germany

26.06.2018

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 VMI topics or 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 9 and a 64-bit vanilla 4.9 LTS kernel

- system call hooking
- file hiding
- process hiding
- module hiding
- socket hiding
- privilege escalation
- networking from the kernel

- 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

Tuesday 14:00 - 16:00 in **MI 01.05.013**

Registration

- Matching System, but:
- Please solve this small qualification tasks
  - Set up a VM for this course with Debian 9
  - Download the 4.9 kernel from kernel.org and compile it
  - Write a kernel module that prints a `ps aux` including these process properties:
    - UID, GID
    - Command Line
    - PID
    - if the process is ptraced or not
  - Constraint: Do not use `for_each_process`.
- Latest, until Wed, 04 July 2018 23:59 to franzen@sec.in.tum.de!