Non-trusted environment issues

Compromise causes (I)

- **node infection**
  - legitimate software containing malicious code (trojan horses), social engineering, physical access, bug/configuration error exploitation (OS syscall, device driver, application, firmware and BIOS, browser ...)
  - backdoors creation, data stealing, hidden (or not so much) processes disruption, …
  - persistent unauthorized access to a system (as root - i.e. rootkits)
  - spyware (sensitive information collection)
  - Ransomware (encryption of sensitive data)

Compromise causes (II)

- **network injection**
  - nodes capable to read and write data while in transit, actors capable to "poison" routing mechanisms
  - access and modification of network data flow, redirection versus illegitimate destination
  - Sniffers and (growing) family of Man-in the-*

Men-at-work (I)

- **man-in-the-middle**
  - attacker secretly intercepts/alters communication between two unaware parties
    - HTTP session hijacking (interception of session cookies to impersonate a user)
    - ARP table poisoning (alteration ARP tables for traffic redirection)
- **man-in-the-browser**
  - infection in the browser to alter web pages/transactions
    - banking trojans like ZEUS that modify online transaction
- **man-in-the-cloud**
  - stealing of credentials/token to access user cloud environment
    - Interception of **Google Drive** OAuth token to access google's victim files

Men-at-work (II)

- **man-in-the-mobile (MitMo)**
  - mobile infection to intercept communication or 2FA
    - ZitMo intercept SMS and forward to C&C
- **man-in-the-disk**
  - vulnerabilities in handling external storage
    - modification of temporary files stored on external device
- **man-in-the-memory (MitMem – guest star)**
  - interception/modification of data while in RAM
    - fileless (stealth) malware
- **man-on-the-side**
  - observe and inject (but not modify) communication
    - China's great cannon
- **man-at-the-end**
  - end-point communication compromise
    - keylogger infection to capture sensitive information

Compromise causes (III)

- **supply chain attacks**
  - compromise of service, hardware, software of a third-party vendor or partner used (and trusted) by the target organization
  - gain access to the target organization, inject unauthorized behavior
  - infrastructure for update management
    - e.g. SolarWind Orion Attack
      - malicious code into software updates of Orion network monitoring platform.
      - distributed to over 18,000 customers, including government agencies and large corporations.
  - libraries and dependencies
  - hardware during manufacturing
  - IT infrastructure management service
  - ...

Advanced Persistence Threats (APT)

- **advanced**
  - use of sophisticated techniques
    - customised malware, zero day vulnerabilities, evasion stategies
  - targeted to specific victim
    - high budget and expertise, careful preparation
- **persistent**
  - compromise maintained for extended period
    - possible escalation and infection diffusion
  - low-profile operation (during infection)
    - stealth techniques, limited bandwidth usage, mimicking legitimate traffic
- **threat**
  - highly skilled individual aiming strategic goals (espionage, foreign country intelligence, …)

APT attack process

- **initial intrusion**
  - access gain through weak access point
    - zero-day vulnerabilities, spear phishing
- **foothold establishment**
  - persistent access set-up
    - backdoors installation, (stealth) malware infection
- **privilege escalation**
  - empower control on the target system
    - credential stealing, vulnerability exploitation, ...
- **lateral movement**
  - expand infection on the target organization
    - credential stealing, vulnerability exploitation, …
- **goal achievement**
  - data exfiltration, sabotage of critical systems

APTxx

- **APTxx used to indicate organised hacker groups**
- **e.g. APT28 (a.k.a. Fancy Bear)**
  - Russian state sponsored group
    - Russian settings, operating in Russian business hours, closely mirroring Russian government strategic interests (e.g. Caucasus)
  - active from mid-2000s (at least 2008)
  - **attacks aerospace, defense, energy, government, media, dissidents, ...**
  - espionage, political influence, cyberwarfare
    - 2016 DNC Hack
      - breach of the Democratic National Committee during U.S. presidential election
      - sensitive information leakage to influence election outcome
    - NotPetya (2017)
      - ransomware attack,
      - designed to target Ukrainian institutions
      - spread globally (billions in damage)

APT28 typical behavior (I)

- **targets desktop, laptop and mobile**
- **employs (spear-)phishing messages**
  - directing to realistic web site for credential harvesting
    - registering domains that closely resemble domains of legitimate organizations
      - e.g. *qov.hu.com* for *gov.hu* (Hungarian government)
    - using URL-shortener services
  - delivering malware in highly-realistic and targeted emails
    - "weaponised" .docx or .pdf
- **implant custom malware**
  - **e.g. X-Agent**
    - multi-functional malware implant
    - data exfiltration, keystroke logging
    - multiplatform (Windows, Linux, Android, and iOS)

APT28 typical behavior (II)

- **after initial access, actively seeks to harvest credentials**
  - keyloggers, central memory dumping
- **adopt evasion techniques**
  - malware code obfuscation
  - signatures of compromised certificates
  - timestomping (timestamps modification)
  - encrypted communication
- **"lateral movement" inside organization (exploiting harvested credentials)**
  - Remote Desktop Protocols
  - Windows Management Instrumentation Command-line (WMIC) and PsExec
    - to execute commands on remote Windows
  - SSH
    - to connect on remote Linux box
- **privilege escalation**
  - exploiting harvested credentials/vulnerabilities

APT28 typical behavior (III)

- **data exfiltration**
  - custom C2 (Command-and-Control) communication
    - e.g. Zebra C2
  - optionally compressed (for large data)
  - through encrypted HTTPs, FTPs or even custom protocols
- Wiper actions
  - typically, APT28 adopts espionage techniques, but...
  - ...has been involved in destructive attacks
    - KillDisk, designed to destroy the master boot record
    - Disk wiping tools (particularly in energy sector)

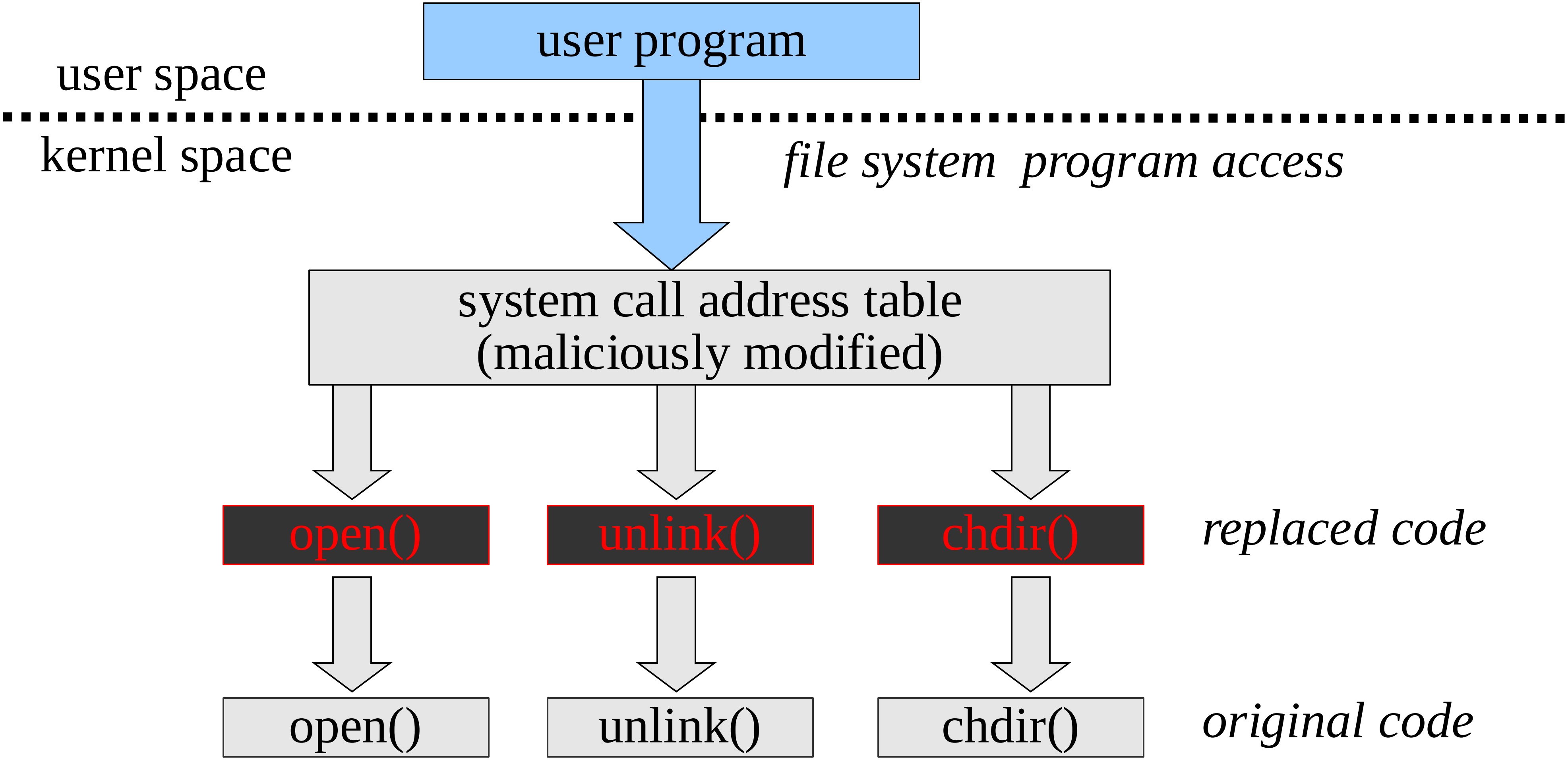
Compromise causes (IV)

- **manipulation from the system owner**
  - If technical-savy, he/she can modify the system in many ways
    - install modified application
    - install different drivers
    - modify system calls

Trusted Environment

- **the analysis must be performed in a trusted environment**
  - *rootkits* can change usual Operanting System behavior
  - changes of usual file system utilities
    - ls, cp, mv, ... commands
  - changes of usual file system calls
    - e.g. intercept of open(), chdir(), unlink(), ... to not show or act on specific files

(example of) System Call Interception



Examples of Linux system modification

- **loadable kernel module (LKM)**
  - same concept exists in many OSes (e.g. *kernel extensions* in macOS, *kernel-mode driver* in windows)
  - LKM can override the original syscall function
    - example steps:
      - develop a different version of the function
      - modify the system call table (an array of function pointers)
      - If you want to modify behavior, re-implement with modified behavior
      - if you want to add functionalities, enrich and call the original one

Examples of Linux system modification

```
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/syscalls.h>
#include <linux/uaccess.h>
asmlinkage int (*original_open)(const char __user *filename, int flags,
mode_t mode);
asmlinkage int custom_open(const char __user *filename, int flags, mode_t
mode) {
  printk(KERN_INFO "Intercepted file open: %s\n", filename);
  return original_open(filename, flags, mode);
 }
 static int __init syscall_init(void) {
   original_open = (void *)sys_call_table[__NR_open];
   sys_call_table[__NR_open] = custom_open;
   return 0;}
 static void __exit syscall_cleanup(void) {
     sys_call_table[__NR_open] = original_open;}
 module_init(syscall_init); module_exit(syscall_cleanup);
 MODULE_LICENSE("GPL");
```