Exploiting and Defense

Dobin Rutishauser

2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024

About Me

Dobin Rutishauser

- Penetration Tester at Compass
- SOC Analyst at Infoguard
- RedTeam at Raiffeisen

Interested in ~~Hacking~~ Security since a young age (1998+)

I got a bit overboard when I was young

Content

Content

Exploiting & Defense

We will write **exploits** to **exploit buffer-overflows**

We will analyze what **defenses** exist to make writing exploits harder

Lecture

Lecture - Online

https://exploit.courses

- Online exploit development website
- Access to your own Linux via JavaScript terminal
- Solve challenges online
  - Write exploits
  - Debug them
- Slides

Lecture - Online

If you wanna try it by yourself on your own machine (not recommended):

The writeup of the challenges: https://github.com/dobin/yookiterm-challenges

Source code of challenges: https://github.com/dobin/yookiterm-challenges-files

Lecture - Online

Important slides are marked with  in top right corner

Sometimes slides have helpful comments in "notes" section

Recap slides at end of chapters point you to which things are important, and should be understood


Siiiii abr ähhhhhh
EBP isch doch 32 bit?
shutterstock
IMAGE ID: 120482521
www.shutterstock.com

Motivation

Motivation for Exploiting & Defense

Motivation

For the hacker:

- Developing exploits
- Debugging of C/C++ code
- Disassembly & reversing of assembler code
- Being 31337
- Understand Windows RedTeaming

For the Sysadmin

- Judge security level of operating systems, and applications
- Harden and protect servers, clients

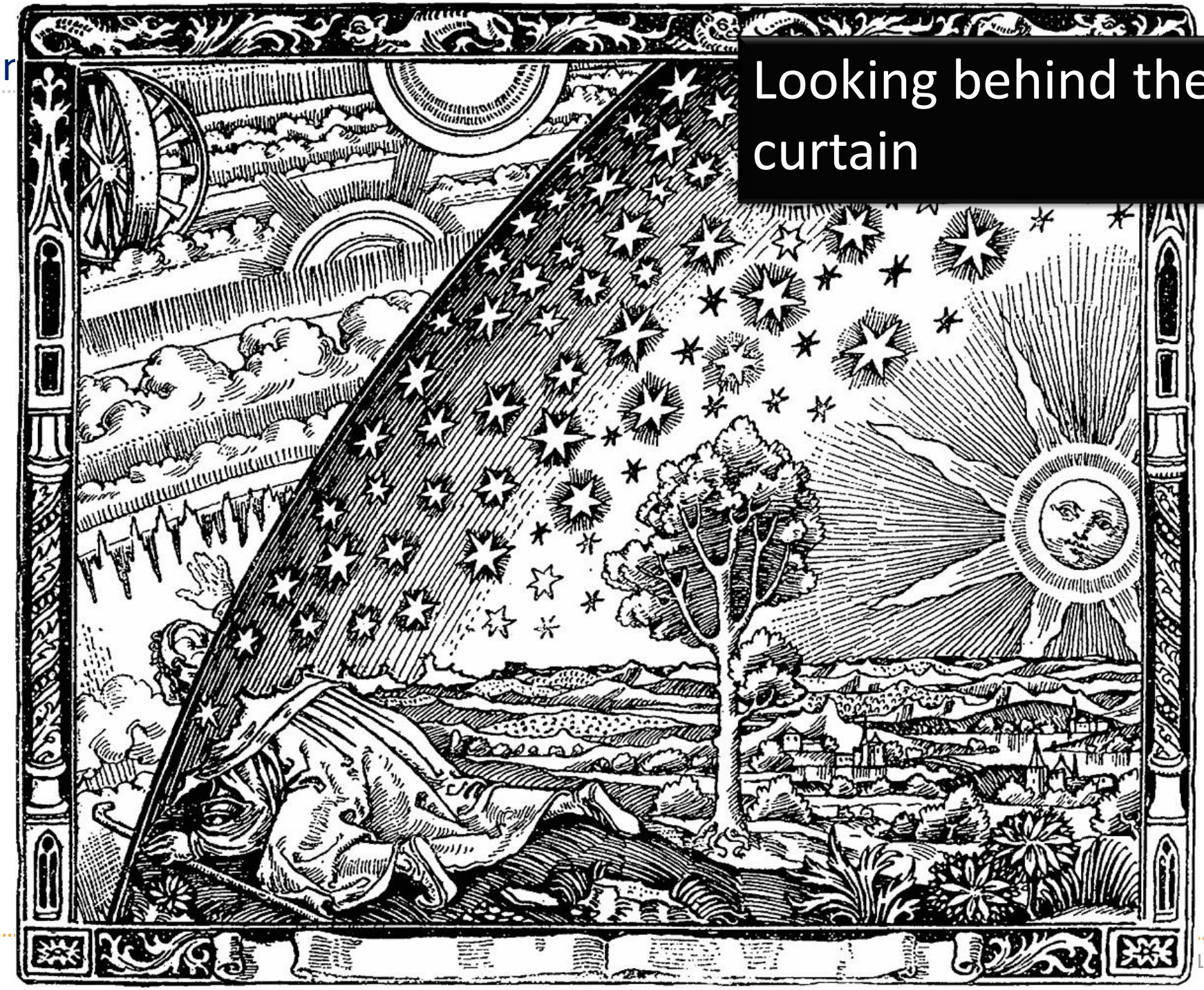
For the CISO / CTI:

- Assess CVSS scores
- Assess (new) security mitigations
- Better risk analysis

Motivation

For everyone:

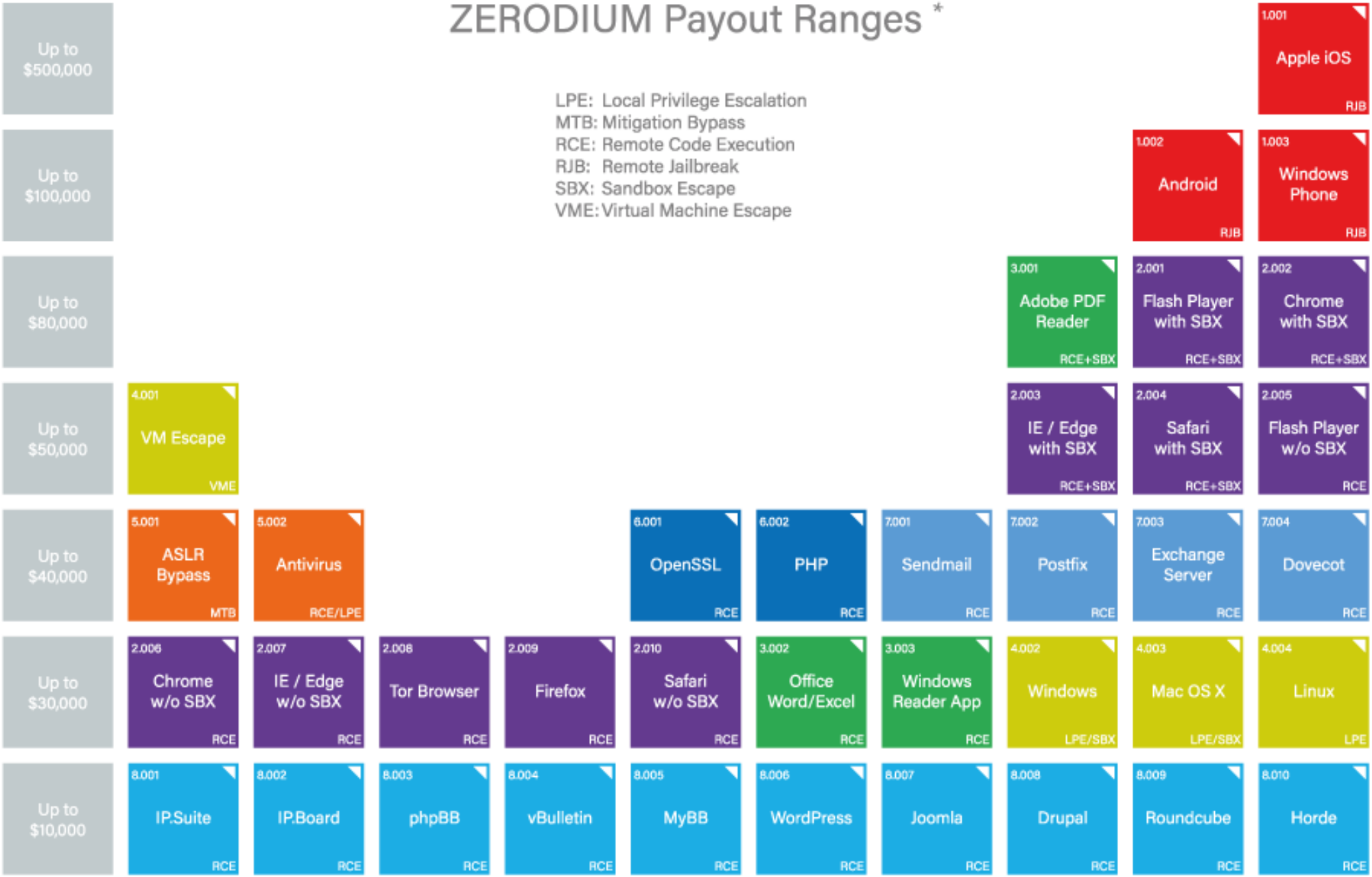
- How do functions work?
- How does computer work?!
- Dance the exploit / mitigation tango
- Breach abstraction layers to get what you want
- Get a sense of long term security development



Looking behind the curtain

ZERODIUM Payout Ranges *

LPE: Local Privilege Escalation
MTB: Mitigation Bypass
RCE: Remote Code Execution
RJB: Remote Jailbreak
SBX: Sandbox Escape
VME: Virtual Machine Escape

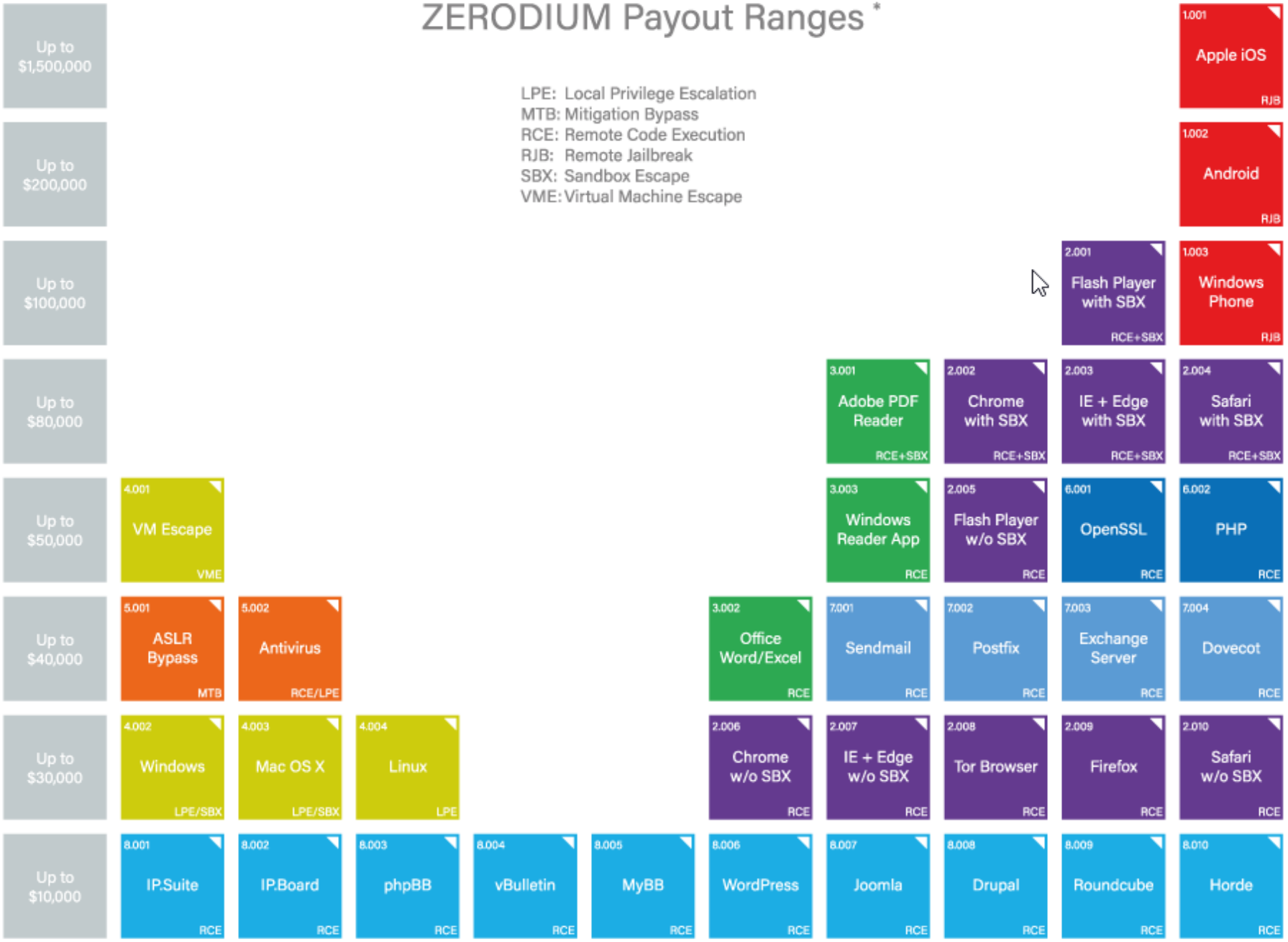


* All payout amounts are chosen at the discretion of ZERODIUM and are subject to change or cancellation without notice.

Mo

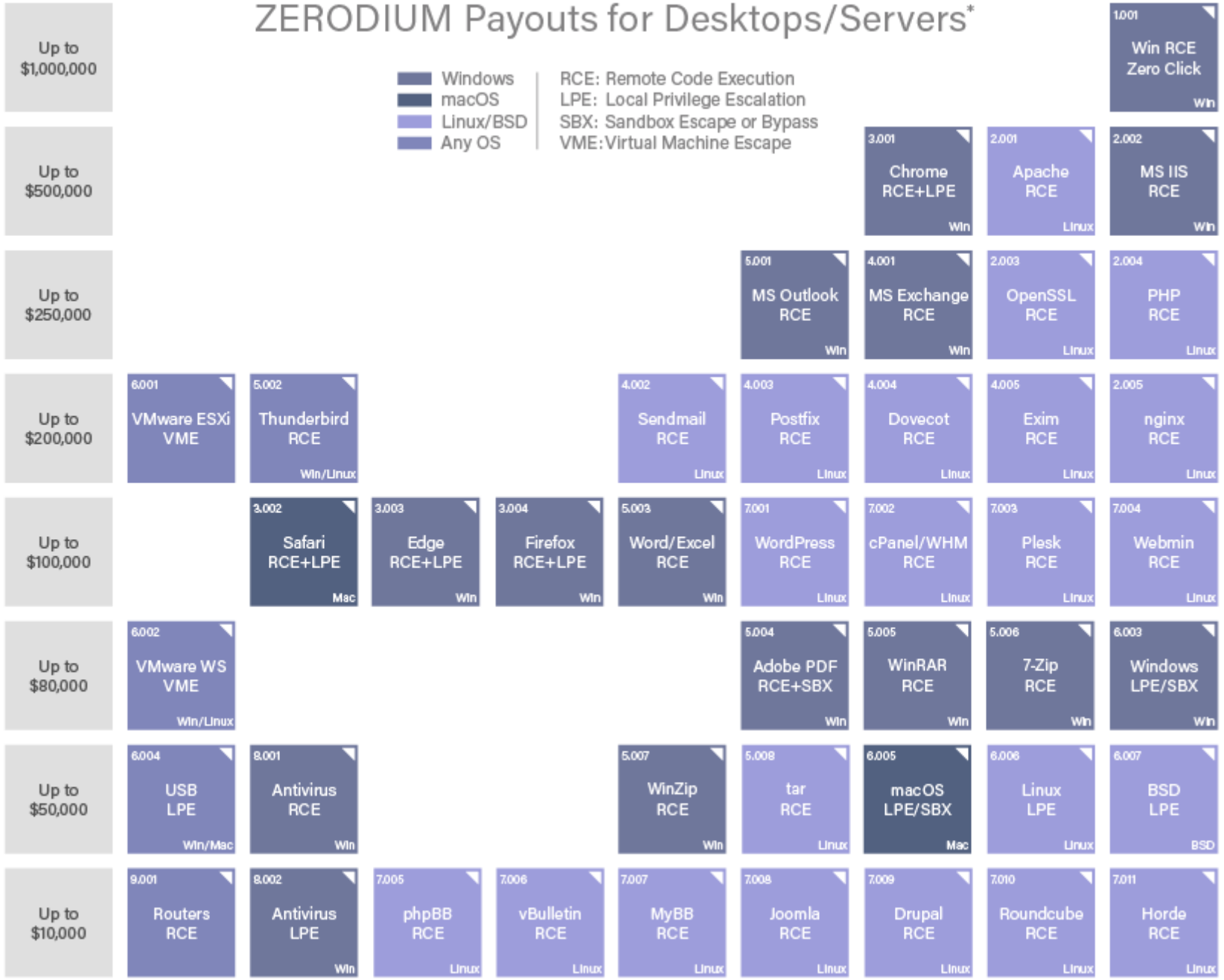
ZERODIUM Payout Ranges *

LPE: Local Privilege Escalation
MTB: Mitigation Bypass
RCE: Remote Code Execution
RJB: Remote Jailbreak
SBX: Sandbox Escape
VME: Virtual Machine Escape



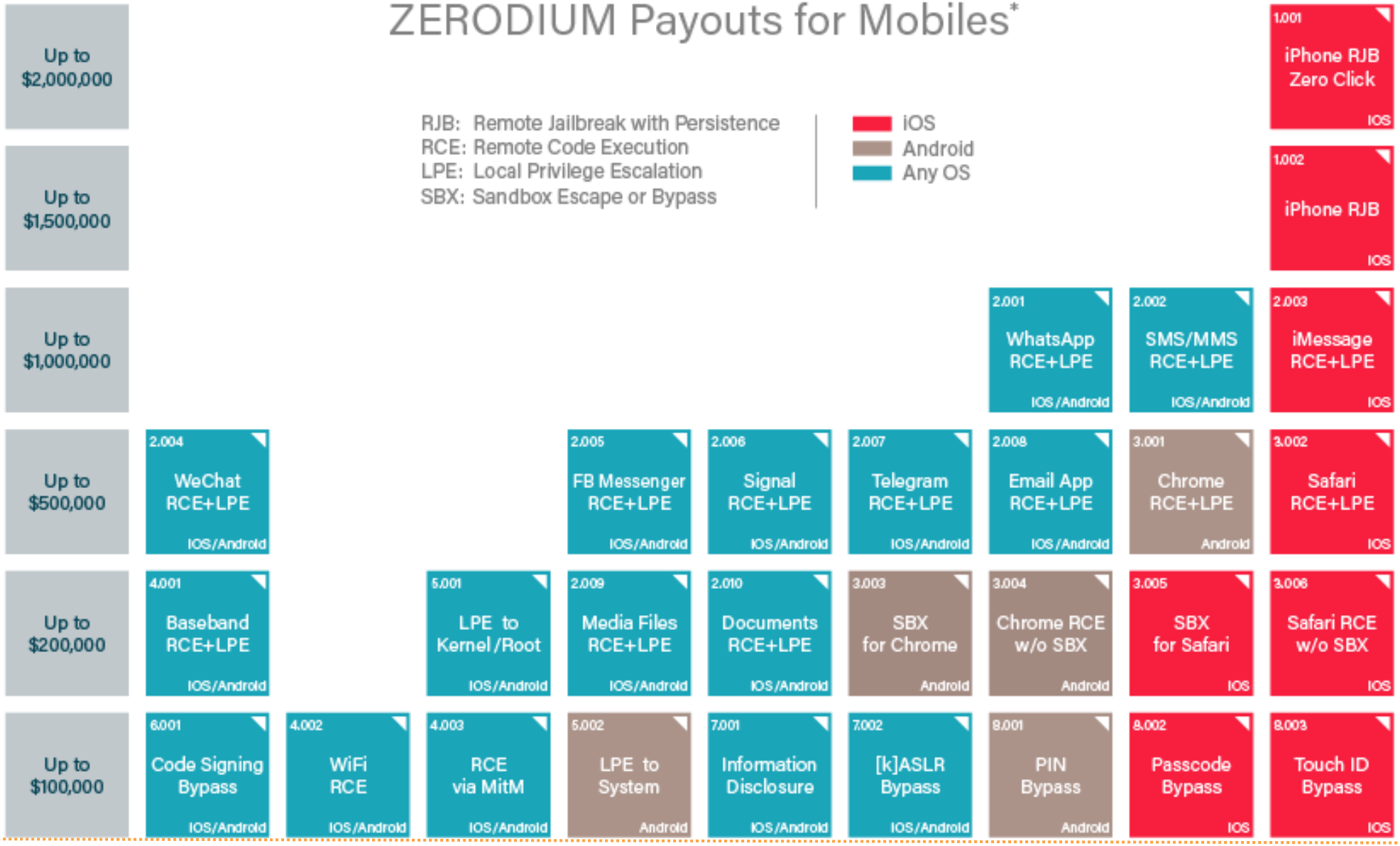
** All payout amounts are chosen at the discretion of ZERODIUM and are subject to change or cancellation without notice.*

ZERODIUM Payouts for Desktops/Servers*



** All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.*

ZERODIUM Payouts for Mobiles*



** All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.*

Linux Vulnerabilities in 2022

CVE-2021-4034: Linux Polkit

- https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034

CVE-2021-44142: Samba

- https://blog.malwarebytes.com/exploits-and-vulnerabilities/2022/02/samba-patches-critical-vulnerability-that-allows-remote-code-execution-as-root/

CVE-2022-0185: Linux Kernel

- https://jfrog.com/blog/the-impact-of-cve-2022-0185-linux-kernel-vulnerability-on-popular-kubernetes-engines/

What is the impact of these vulns? Risk? Whats the vulnerability? Whats the mitigation?

Content of the next 7 Friday afternoons

You want to learn:

- What memory corruptions are
- What buffer overflows are
- What exploits are
- How exploits are being created
- To exploit a local application
- To exploit a remote application
- Learn about anti-exploiting technologies
- To circumvent all common anti-exploiting technologies for Linux
- See how Windows does it
- Use Use-After-Free Heap overflows
- See next generation attacks and defenses
- ~~Hack facebook “for a friend”~~

What you first learn:

- Intel x86
  - Architecture
  - CPU
  - Registers
- Linux
  - Userspace memory layout, stacks, heap
  - Syscalls
  - Sockets
- Programming Languages
  - Assembler
  - C
  - Python
  - Bash

Plan

Plan

19.04.2023

Theory:

- 0x01 Intro (this)
- 0x02 Intro Technical
- 0x10 Intel Architecture
- 0x11 Memory Layout

Challenges:

- 0: Introduction to memory layout - basic
- 1: Introduction to memory layout - advanced

Plan

26.04.2023

Theory:

- 0x12 C Array and Data Structures
- 0x30 Assembler Intro
- 0x31 Shellcode
- 0x32 Function Call Convention
- 0x33 Debugging

Challenges:

- 2: C buffer analysis - simple
- 3: Introduction to shellcode development
- 7: Function Call Convention in x86 (32bit)
- 8: C buffer analysis - with debugging
- 9: Simple Buffer overflow - variable overwrite

Plan

03.05.2023

Theory:

- 0x40 Arrays
- 0x41 Buffer Overflow
- 0x42 Exploit
- 0x44 Remote Exploit

Challenges:

- 11: Development of a buffer overflow exploit - 32 bit
- 12: Development of a buffer overflow exploit - 64 bit
- 13: Development of a remote buffer overflow exploit - 64 bit

Plan

10.05.2023

- Nüt

Plan

17.05.2023

Theory:

- 0x51 Exploit Mitigation
- 0x52 Defeat Exploit Mitigation
- 0x70 Secure Coding

Challenges:

- 14: Stack canary brute force
- 15: Simple remote buffer overflow exploit - ASLR/DEP/64bit
- 16: Remote buffer overflow with ROP - DEP/64bit
- 17: Remote buffer overflow with ROP - DEP/ASLR/64bit

Plan

24.05.2023

Theory:

- 0x52: Defeat Exploit Mitigations
- 0x56: Defeat Exploit Mitigations – PIE
- stuff

Challenges:

- 31: Heap use-after-free analysis

Plan

31.05.2023

Theory:

- 0x54: Defeat Exploit Mitigations ROP
- 0x60: Windows Exploiting
- 0x74: Hardware Hacking
- 0xA0: Browser Security
- stuff

Plan

07.06.2023

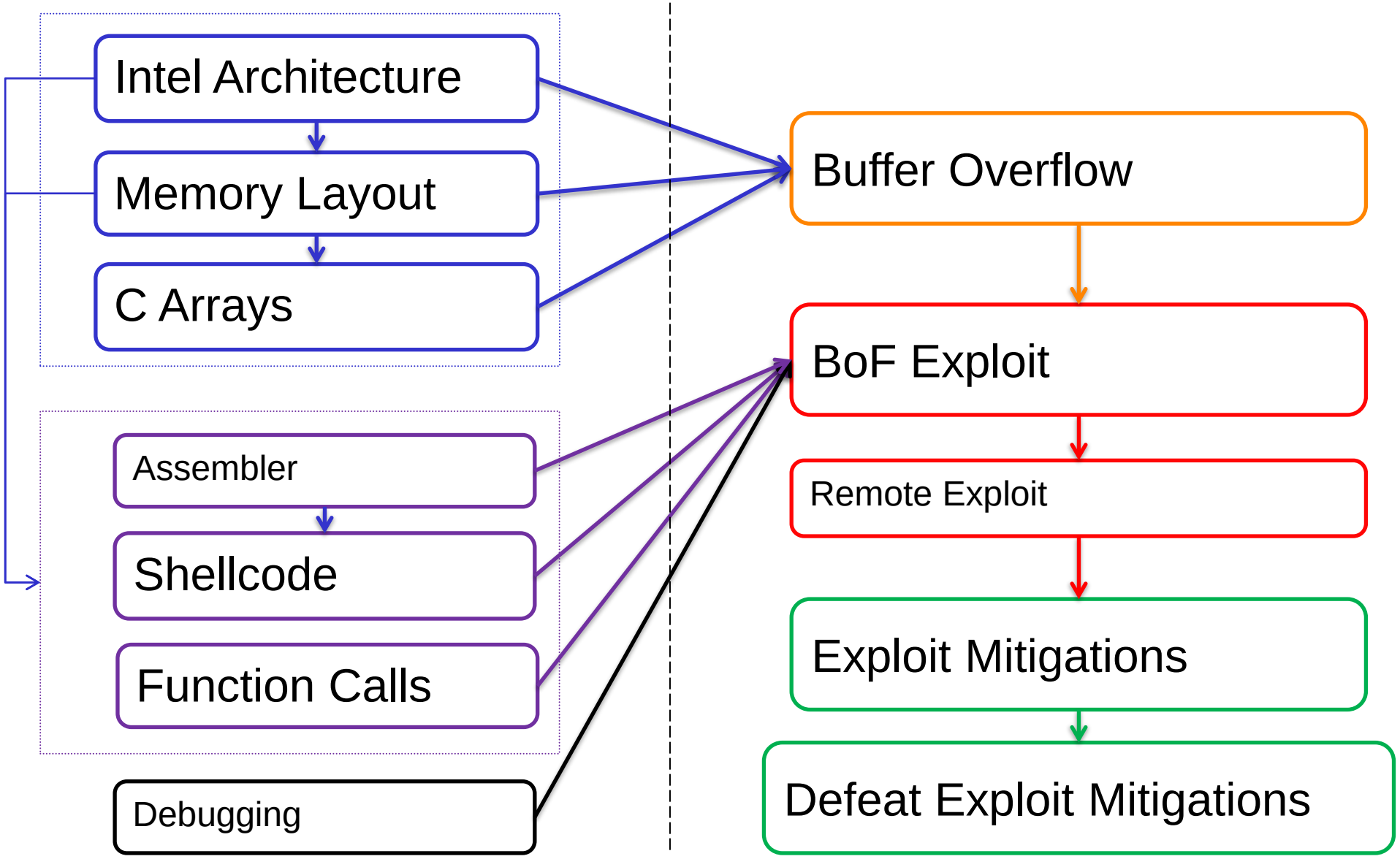
Theory:

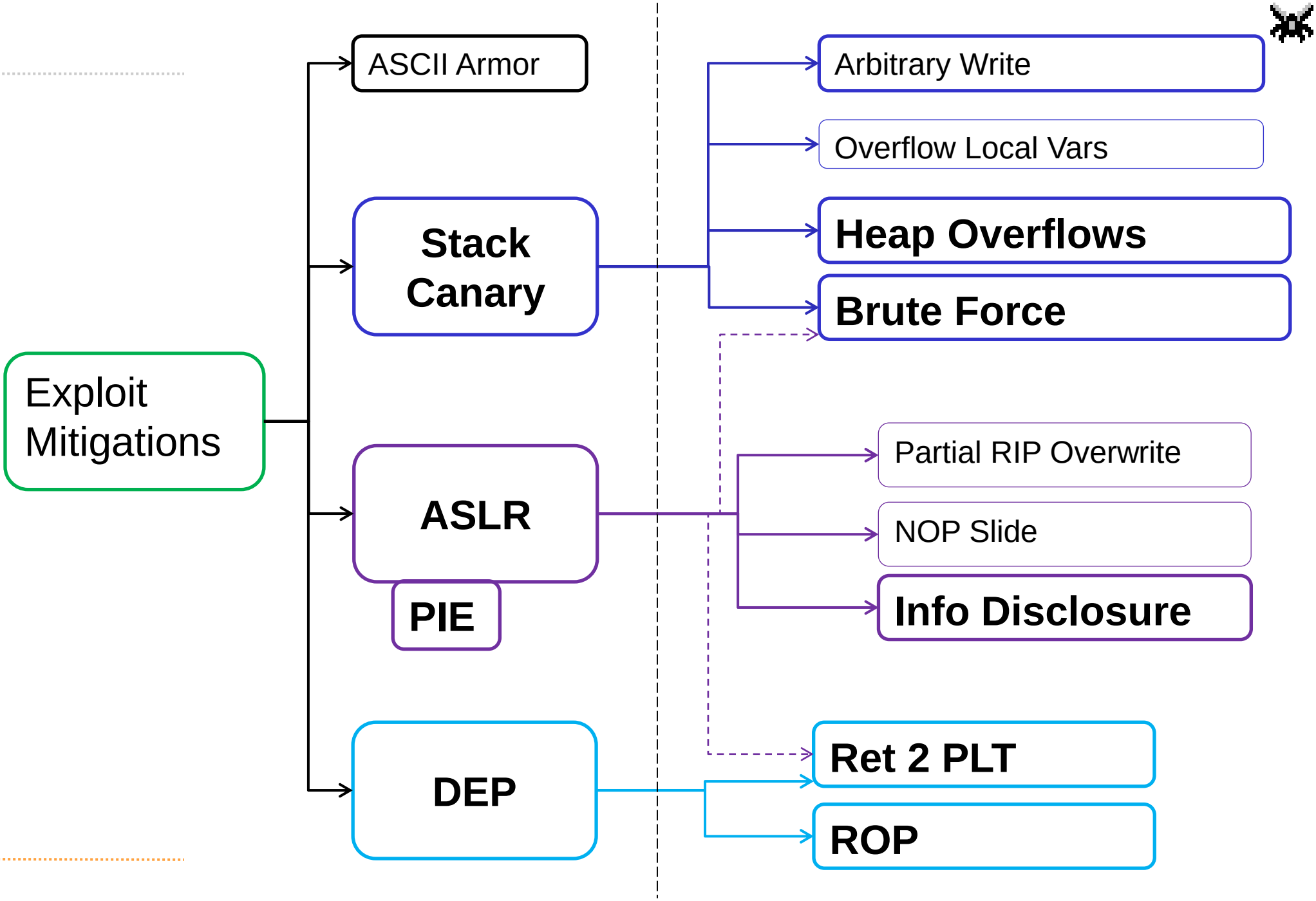
- 0x71: Fuzzing
- 0x75: CFI

Challenges:

- 60: Linux Hardening

Content


Intel Architecture
Memory Layout
C Arrays
Assembler
Shellcode
Function Calls
Debugging
Buffer Overflow
BoF Exploit
Remote Exploit
Exploit Mitigations
Defeat Exploit Mitigations


Exploit Mitigations
ASCII Armor
Stack Canary
ASLR
PIE
DEP
Arbitrary Write
Overflow Local Vars
Heap Overflows
Brute Force
Partial RIP Overwrite
NOP Slide
Info Disclosure
Ret 2 PLT
ROP

And some…

- Windows Exploiting
- Fuzzing
- Browser Security
- Kernel Exploits
- Secure Coding
- Linux Hardening
- Case Studies

Oral Exam

What is (mainly) relevant for the oral exam:

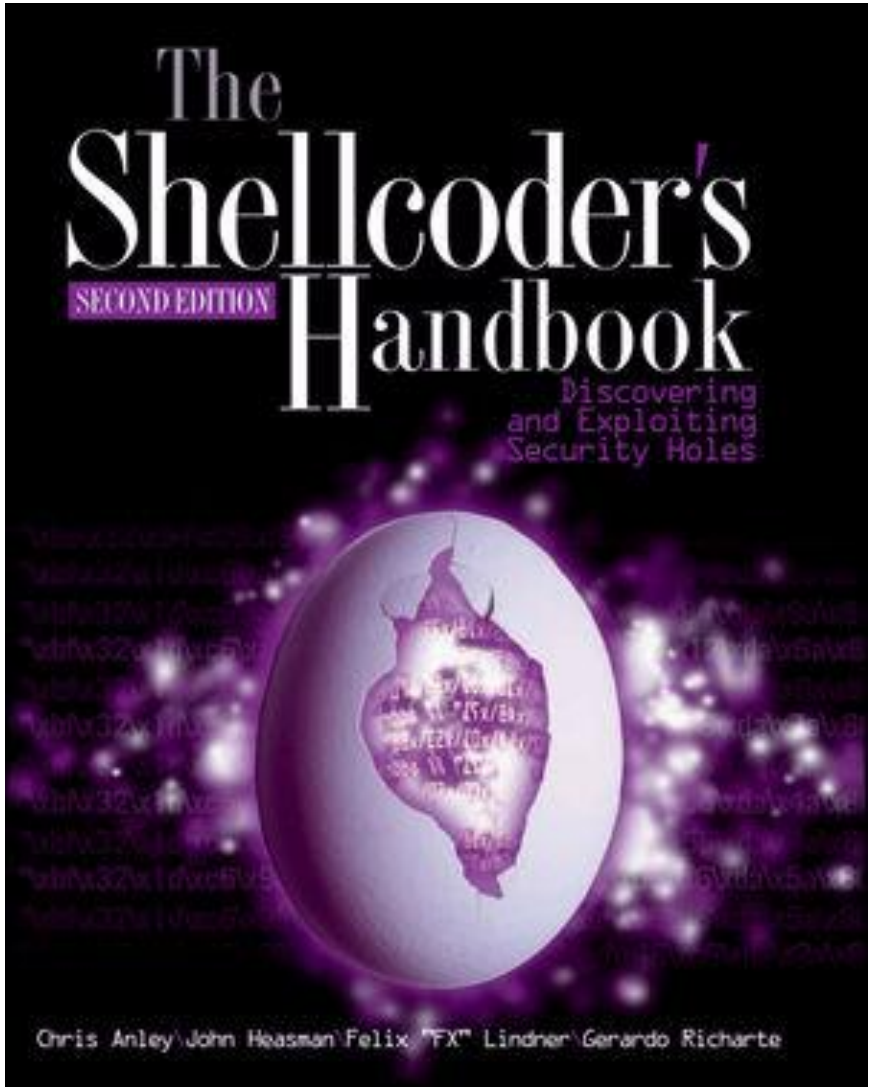
- How does memory corruption work?
- How does an exploit work?
- What exploit mitigations exist?
- How can these exploit mitigations be circumvented?

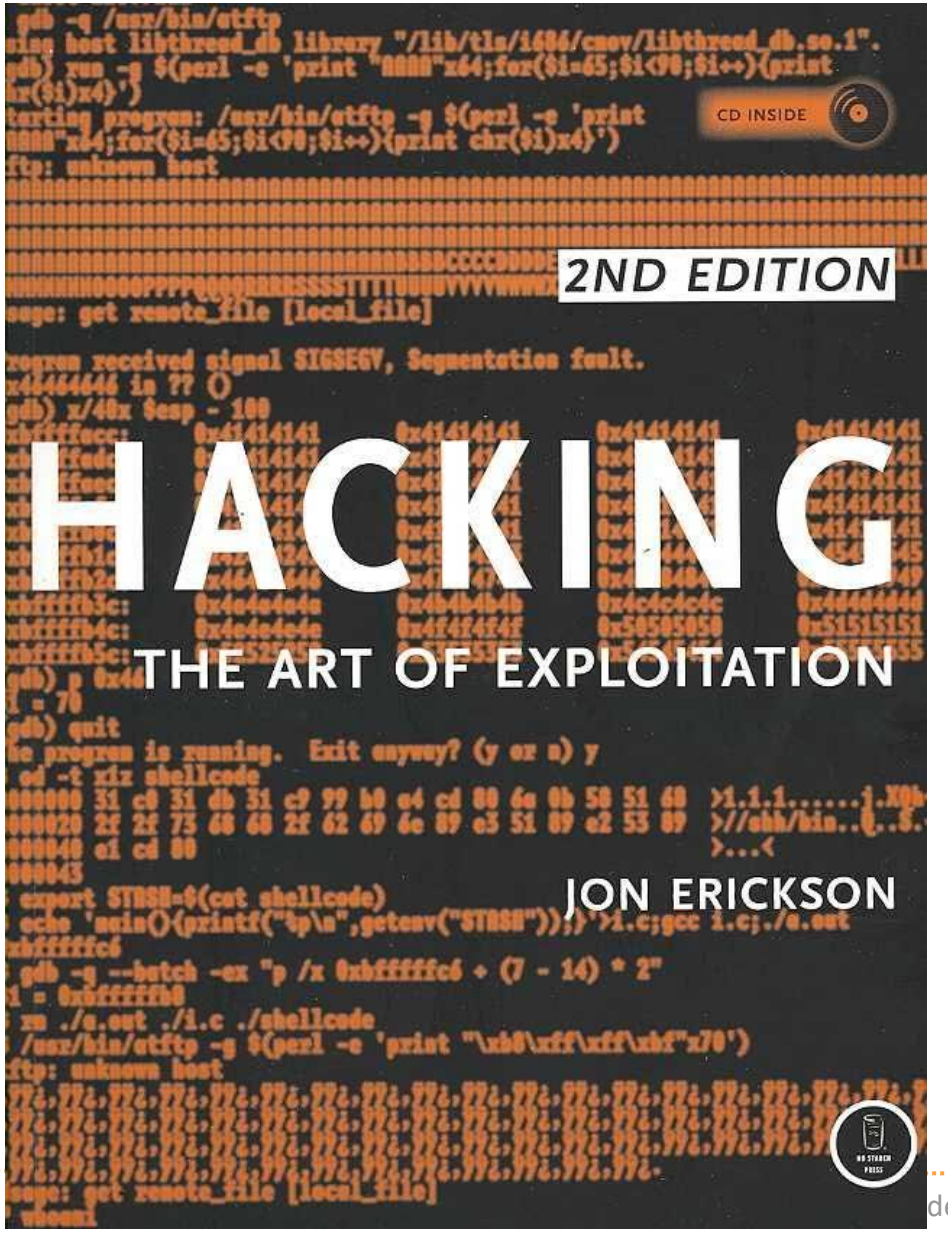
More **theoretical**, not so much the nitty gritty details

Typical question:

- Explain me how a buffer overflow exploit works
- Now we introduce ASLR. What do you need to change?

Books


The
Shellcoder's
SECOND EDITION
Handbook
Discovering
and Exploiting
Security Holes
Chris Anley\John Heasman\Felix "FX" Lindner\Gerardo Richarte


CD INSIDE
2ND EDITION
HACKING
THE ART OF EXPLOITATION
JON ERICKSON

Legal Issues

Legal CH

Don’t hack other people’s systems

«Damit der Tatbestand des **strafbaren Hackens** erfüllt ist, müssen **folgende Voraussetzungen kumulativ** erfüllt sein:

- **Eindringen** in das **Datenverarbeitungssystem**;
- **fremdes Datenverarbeitungssystem**;
- Eindringen auf dem Weg der von **Datenübertragungseinrichtungen;**
- **besondere Sicherung** gegen Zugriff.

https://www.lexwiki.ch/hacken/

Legal International

Wassenaar

- Arms Control Treaty
  - Anti-proliferation of Nukes and stuff
- Includes now (?):
  - Intrusion malware
  - Intrusion exploits
  - IP surveillance
- -> Exploits are now weapons…
  - Not allowed to transport over the border
  - Exception: If they are open source
  - (stop selling 0-days to Chinese gov!)

http://blog.erratasec.com/2015/05/some-notes-about-wassenaar.html