# Exploiting and Defense

Dobin Rutishauser

2016, 2017, 2018

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# Intro

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# About Me

Dobin Rutishauser

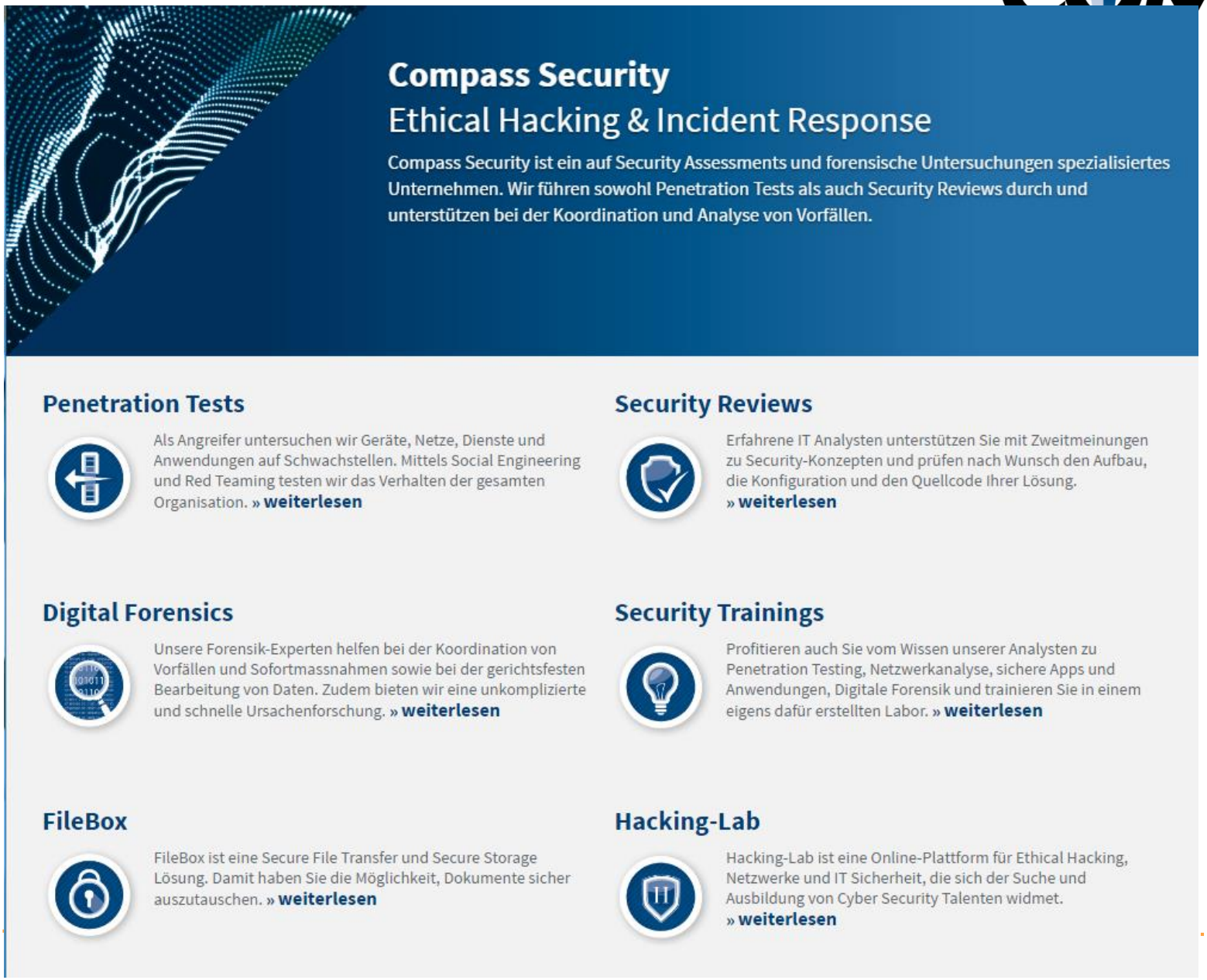
Working as Security Analyst @ Compass Security

- Penetration Tests
- Webapp Checks
- Architecture Reviews
- & lots more

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

# I got a bit overboard when I was little

# Compass


Compass Security
Ethical Hacking & Incident Response
Compass Security ist ein auf Security Assessments und forensische Untersuchungen spezialisiertes Unternehmen. Wir führen sowohl Penetration Tests als auch Security Reviews durch und unterstützen bei der Koordination und Analyse von Vorfällen.
Penetration Tests
Als Angreifer untersuchen wir Geräte, Netze, Dienste und Anwendungen auf Schwachstellen. Mittels Social Engineering und Red Teaming testen wir das Verhalten der gesamten Organisation. » weiterlesen
Security Reviews
Erfahrene IT Analysten unterstützen Sie mit Zweitmeinungen zu Security-Konzepten und prüfen nach Wunsch den Aufbau, die Konfiguration und den Quellcode Ihrer Lösung. » weiterlesen
Digital Forensics
Unsere Forensik-Experten helfen bei der Koordination von Vorfällen und Sofortmassnahmen sowie bei der gerichtsfesten Bearbeitung von Daten. Zudem bieten wir eine unkomplizierte und schnelle Ursachenforschung. » weiterlesen
Security Trainings
Profitieren auch Sie vom Wissen unserer Analysten zu Penetration Testing, Netzwerkanalyse, sichere Apps und Anwendungen, Digitale Forensik und trainieren Sie in einem eigens dafür erstellten Labor. » weiterlesen
FileBox
FileBox ist eine Secure File Transfer und Secure Storage Lösung. Damit haben Sie die Möglichkeit, Dokumente sicher auszutauschen. » weiterlesen
Hacking-Lab
Hacking-Lab ist eine Online-Plattform für Ethical Hacking, Netzwerke und IT Sicherheit, die sich der Suche und Ausbildung von Cyber Security Talenten widmet. » weiterlesen

# Compass is hiring (always)

Wir haben verschiedene Stellen als **Penetration Tester** aber auch als **Software Entwickler** offen und würden uns sehr über Deine **Bewerbung** freuen.

Bist Du grundsätzlich vom Typ "Grübler" und "Tüftler"? Hast Du Freude daran, Dich in neue Themen und Techniken einzuarbeiten? Dann bist Du bei Compass genau richtig!

Bitte schicke Deine Fragen an ivan.buetler@compass-security.com und Deine offizielle Bewerbung an hr@compass-security.com

Gruss Ivan Bütler, E1

# Content

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# Content

Exploit & Defense

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

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

# Lecture

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Website:

https://exploit.courses

- Online exploit development website
- Access to your own Linux via JavaScript terminal
- Uses Hacking-Lab accounts
- Solve challenges online
  - Write exploits
  - Debug stuff

https://www.hacking-lab.com

- Half-online challenges website
- Uses HLCD (Kali-based Linux Distribution)
- VPN-Based
- Use this if you don’t like exploit.courses


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

# Motivation

Motivation for Exploiting & Defense

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# Motivation

## For the hacker:

- Develop exploits
- Debugging of C/C++ code
- Disassembly & reversing of assembler code
- Being 31337

## For the Sysadmin

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

## For the CISO:

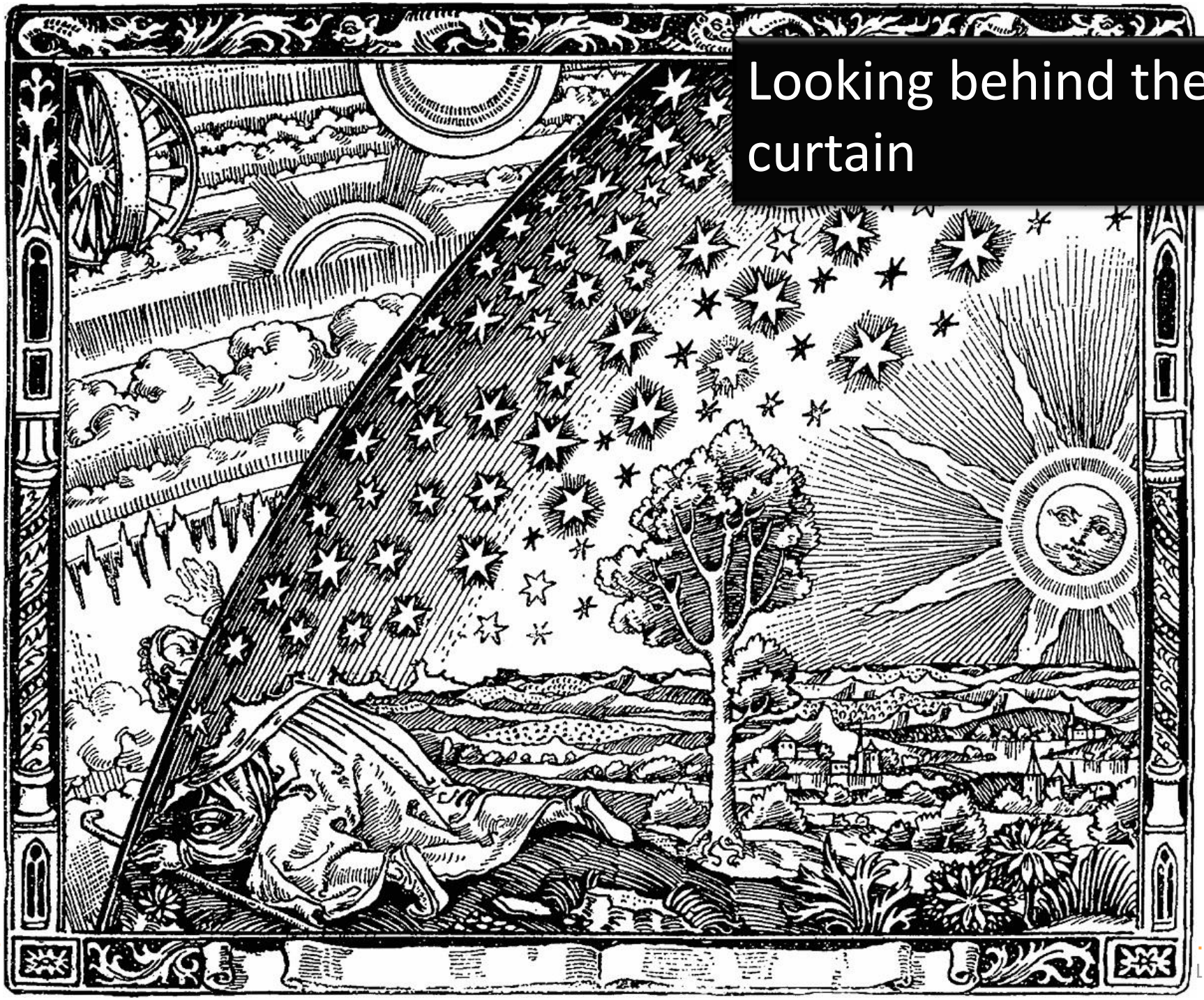
- Assess CVSS scores
- Assess (new) security mitigations
- Better risk analysis

# Motivation

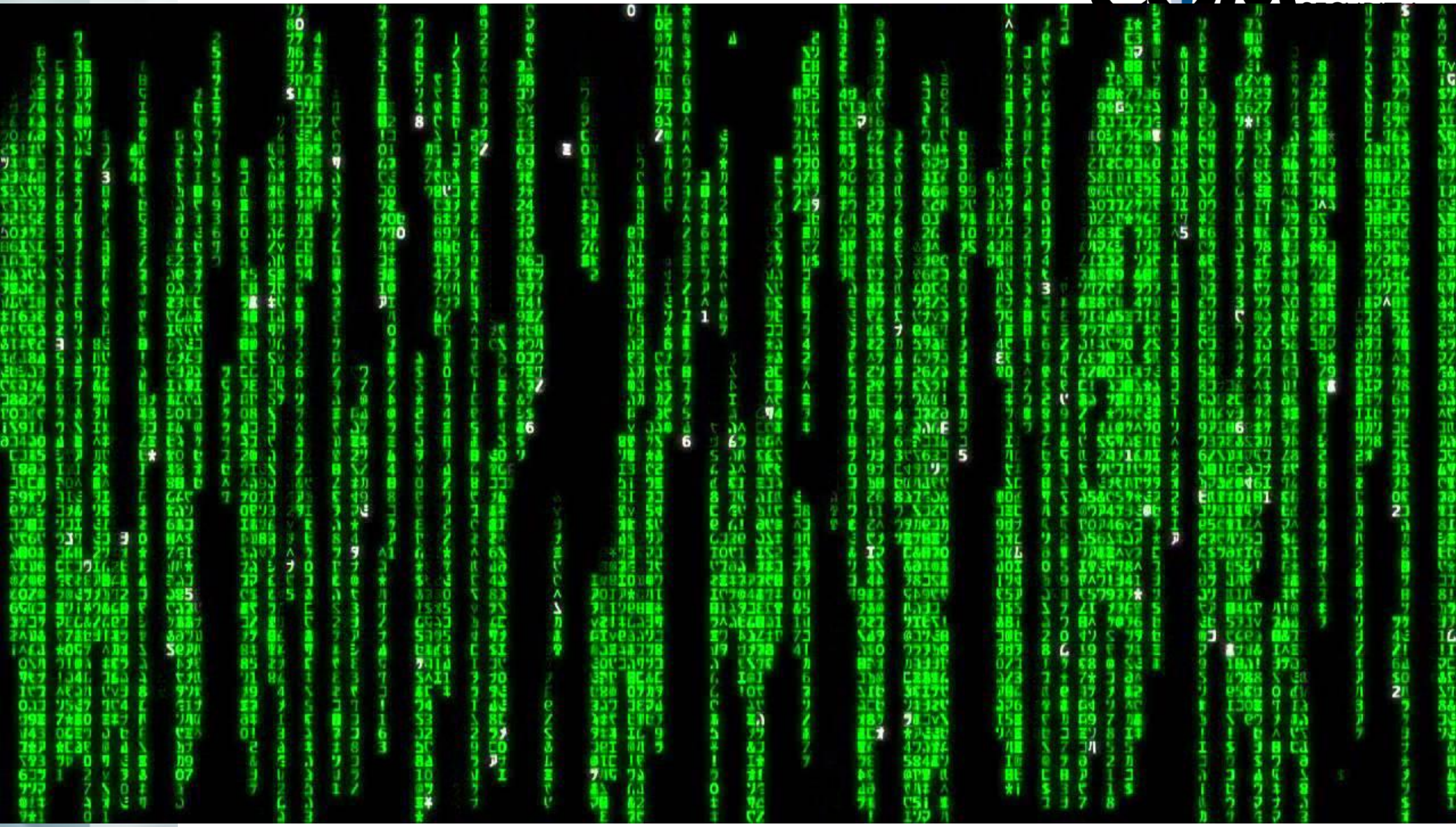
## For everyone:

- How do functions work?
- How does the memory allocator work?
- What's the difference between userspace and kernelspace?
- How does computer work?!

# Looking behind the curtain

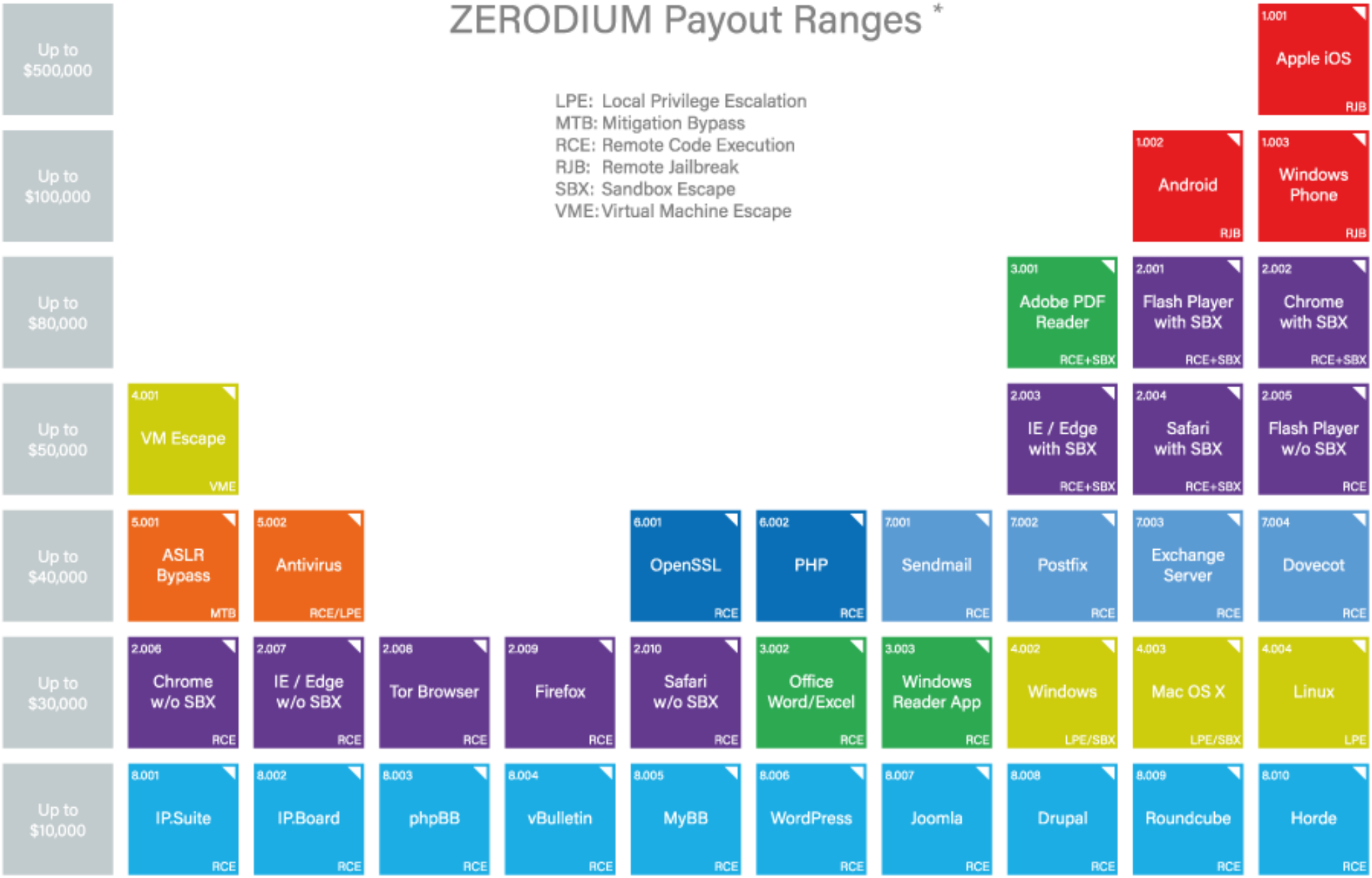


# Motivation



# ZERODIUM Payout Ranges *

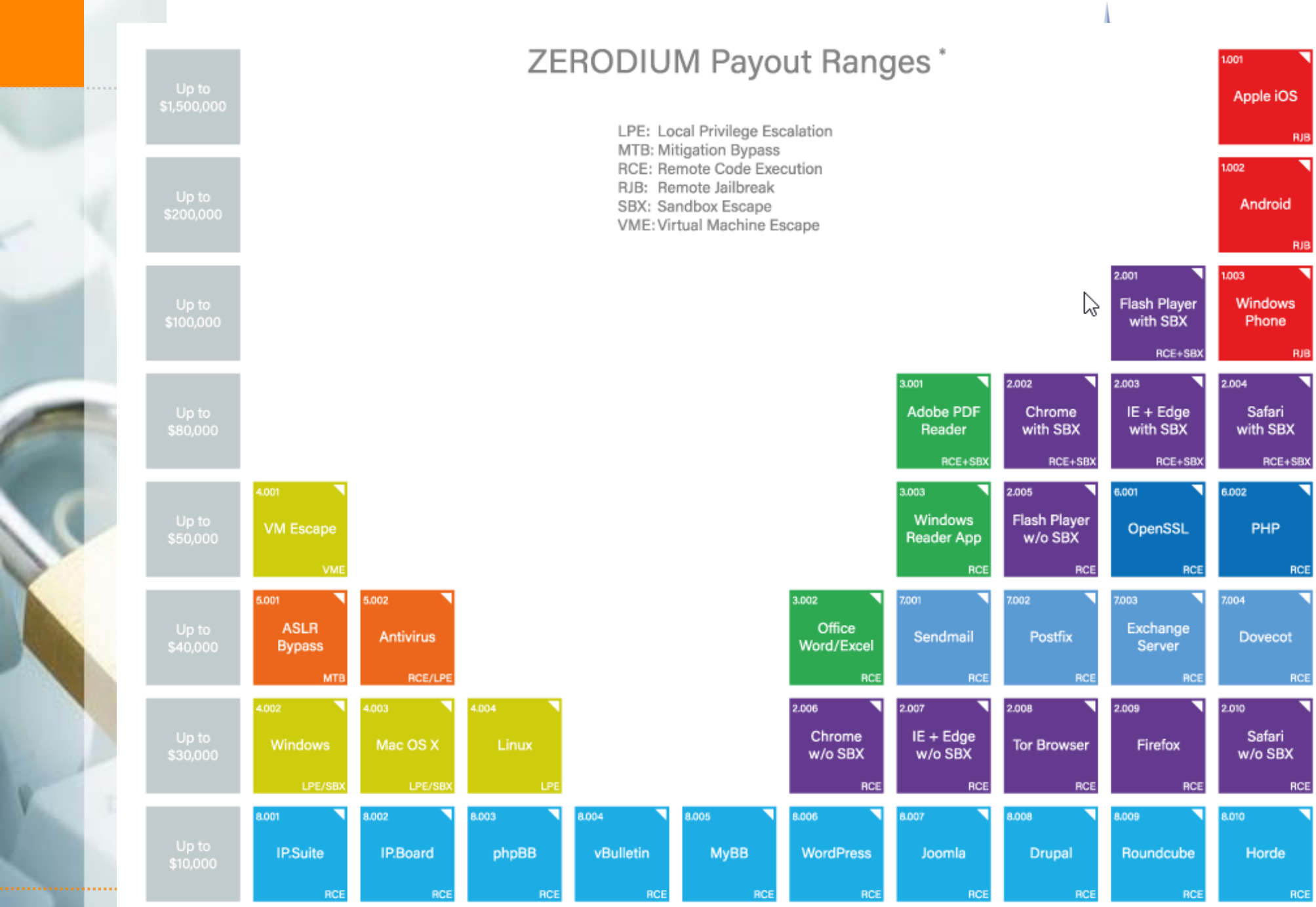
LPE: Local Privilege Escalation
MTB: Mitigation Bypass
RCE: Remote Code Execution
RJB: Remote Jailbreak
SBX: Sandbox Escape
VME: Virtual Machine Escape



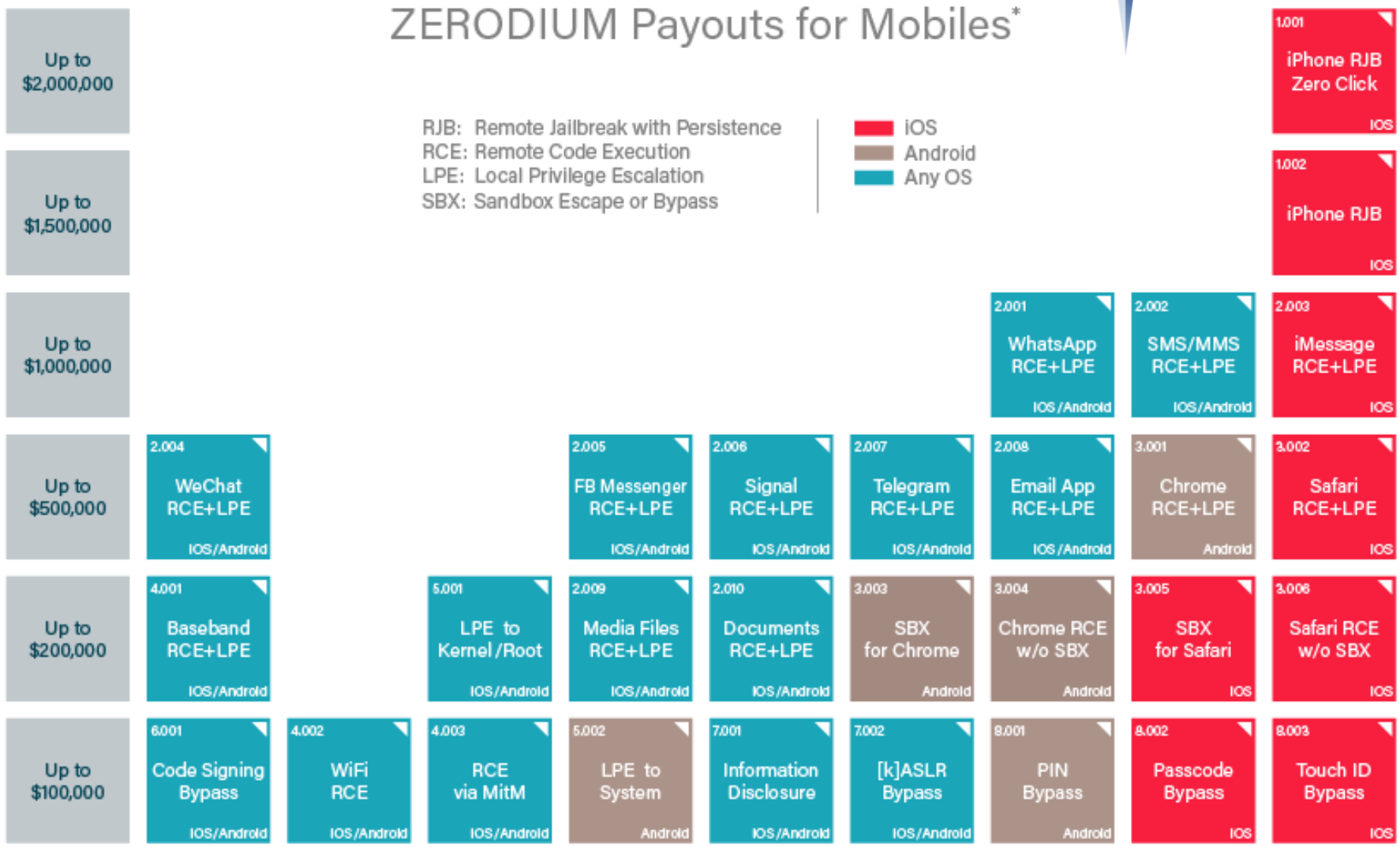

** All payout amounts are chosen at the discretion of ZERODIUM and are subject to change or cancellation without notice.*

# ZERODIUM 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 Mobiles*
RJB: Remote Jailbreak with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass
iOS
Android
Any OS
Up to $2,000,000
1.001 iPhone RJB Zero Click iOS
Up to $1,500,000
1.002 iPhone RJB iOS
Up to $1,000,000
2.001 WhatsApp RCE+LPE iOS/Android
2.002 SMS/MMS RCE+LPE iOS/Android
2.003 iMessage RCE+LPE iOS
Up to $500,000
2.004 WeChat RCE+LPE iOS/Android
2.005 FB Messenger RCE+LPE iOS/Android
2.006 Signal RCE+LPE iOS/Android
2.007 Telegram RCE+LPE iOS/Android
2.008 Email App RCE+LPE iOS/Android
3.001 Chrome RCE+LPE Android
3.002 Safari RCE+LPE iOS
Up to $200,000
4.001 Baseband RCE+LPE iOS/Android
5.001 LPE to Kernel/Root iOS/Android
2.009 Media Files RCE+LPE iOS/Android
2.010 Documents RCE+LPE iOS/Android
3.003 SBX for Chrome Android
3.004 Chrome RCE w/o SBX Android
3.005 SBX for Safari iOS
3.006 Safari RCE w/o SBX iOS
Up to $100,000
6.001 Code Signing Bypass iOS/Android
4.002 WiFi RCE iOS/Android
4.003 RCE via MitM iOS/Android
5.002 LPE to System Android
7.001 Information Disclosure iOS/Android
7.002 [k]ASLR Bypass iOS/Android
8.001 PIN Bypass Android
8.002 Passcode Bypass iOS
8.003 Touch ID Bypass iOS
* All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.
2019/01 © zerodium.com

# Content of the next 8 Friday afternoons

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# 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
- Hack browsers
- ~~Hack facebook “for a friend”~~

# You will actually learn:

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

# Plan

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# Plan

29.03.2019

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

05.04.2019

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

12.04.2019

Theory:

- 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

26.04.2019

Theory:

- 0x51 Exploit Mitigation
- 0x52 Defeat Exploit Mitigation
- 0x53 Exploit Mitigation – PIE
- 0x54 Defeat Exploit Mitigation ROP

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

03.05.2019

Theory:

- 0x55: Defeat Exploit Mitigation – Heap Intro
- 0x56: Defeat Exploit Mitigation – Heap Attacks

Challenges:

- 31: Heap use-after-free analysis

# Plan

10.05.2019

Theory:

- 0x60: Windows Exploiting
- 0x70: Secure Coding
- 0x71: Fuzzing

Challenges:

- 60: Linux Hardening

# Plan

17.05.2017

Theory:

- 0x72: Linux Hardening
- 0x73: Kernel Exploitation
- 0x74: Hardware Hacking

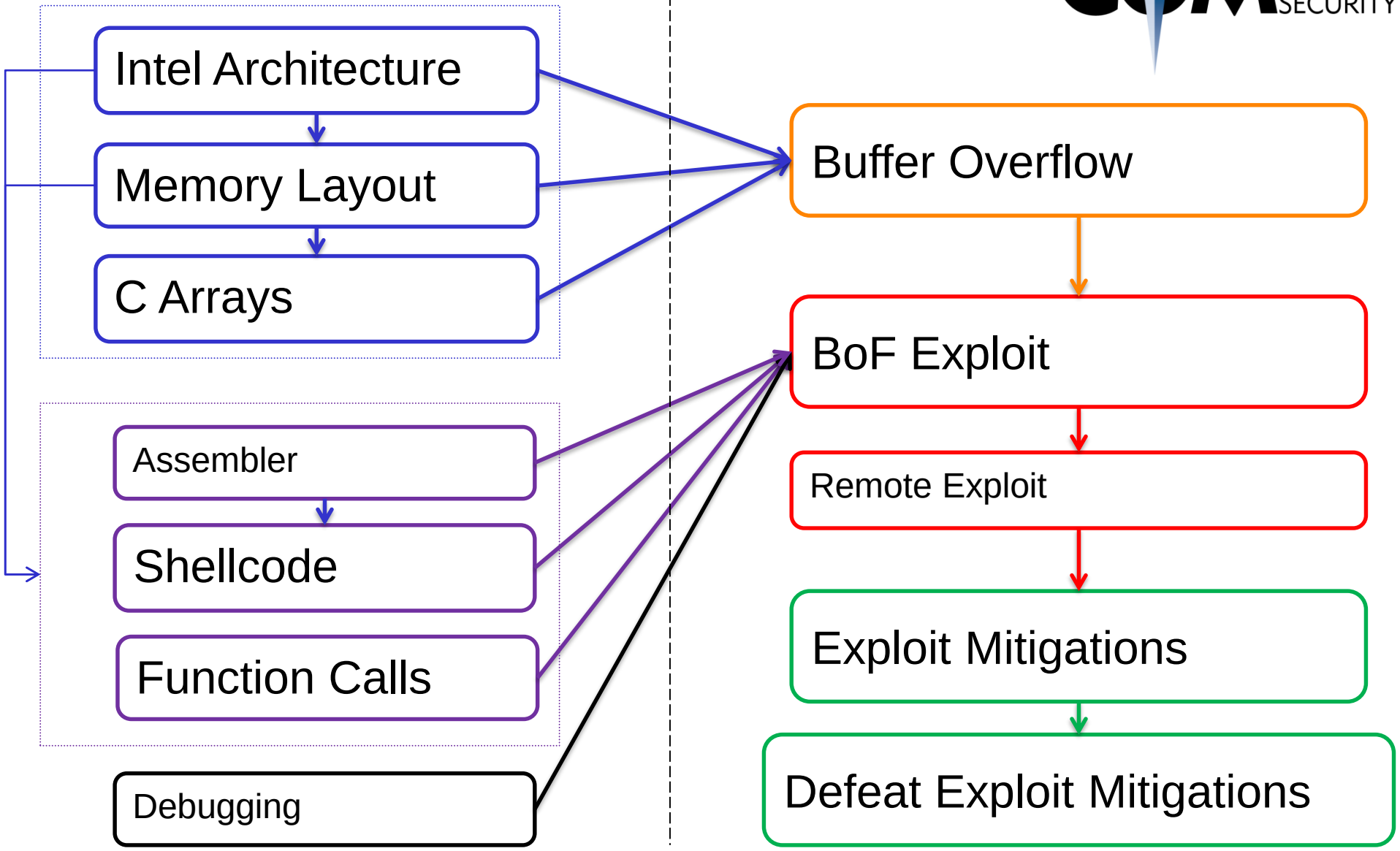
# Plan

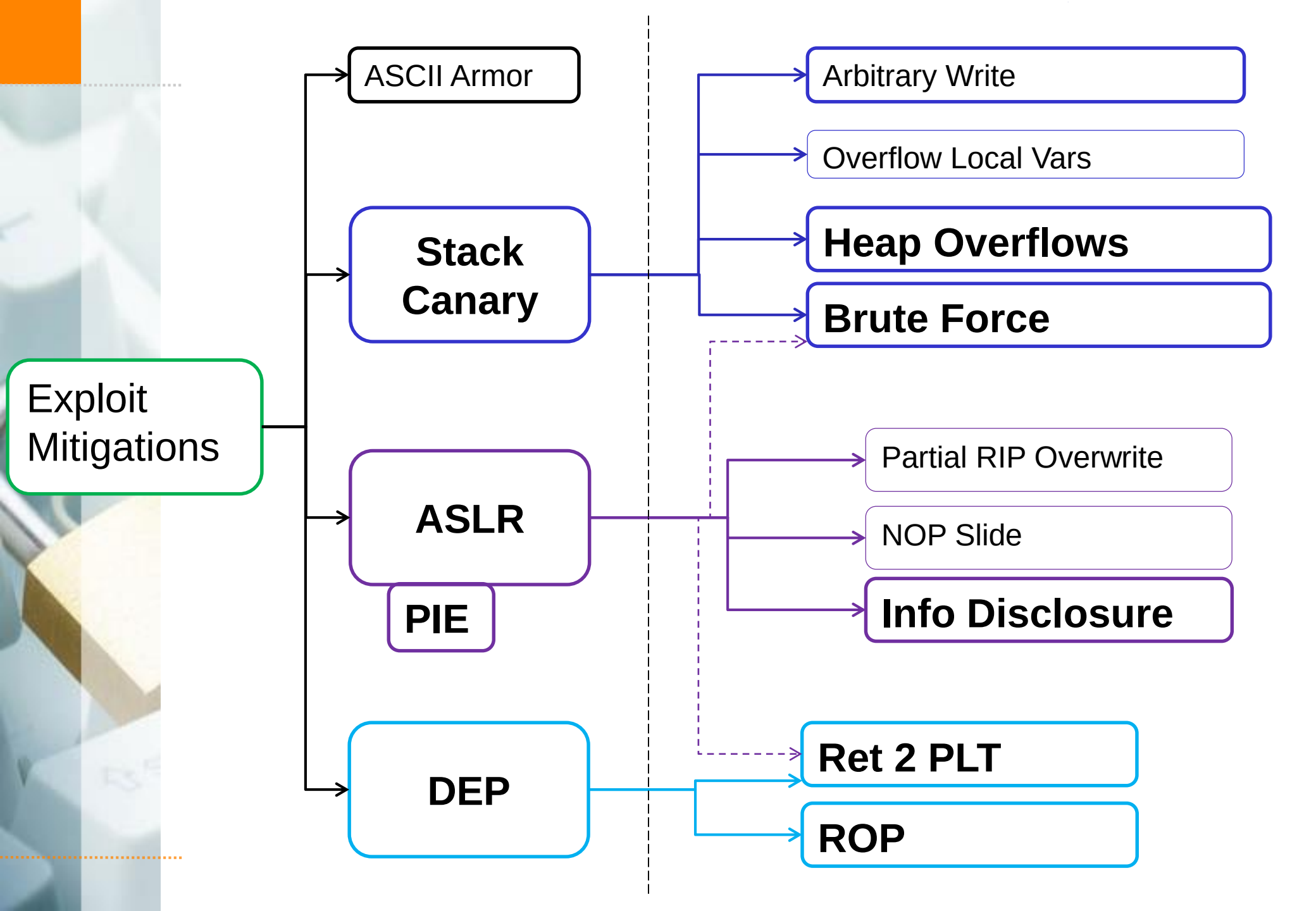
24.05.2017

Theory:

- Puffer
- Case Studies
- Questions

# 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:

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

# Exam Oral

What is (mainly) relevant for the oral exam?

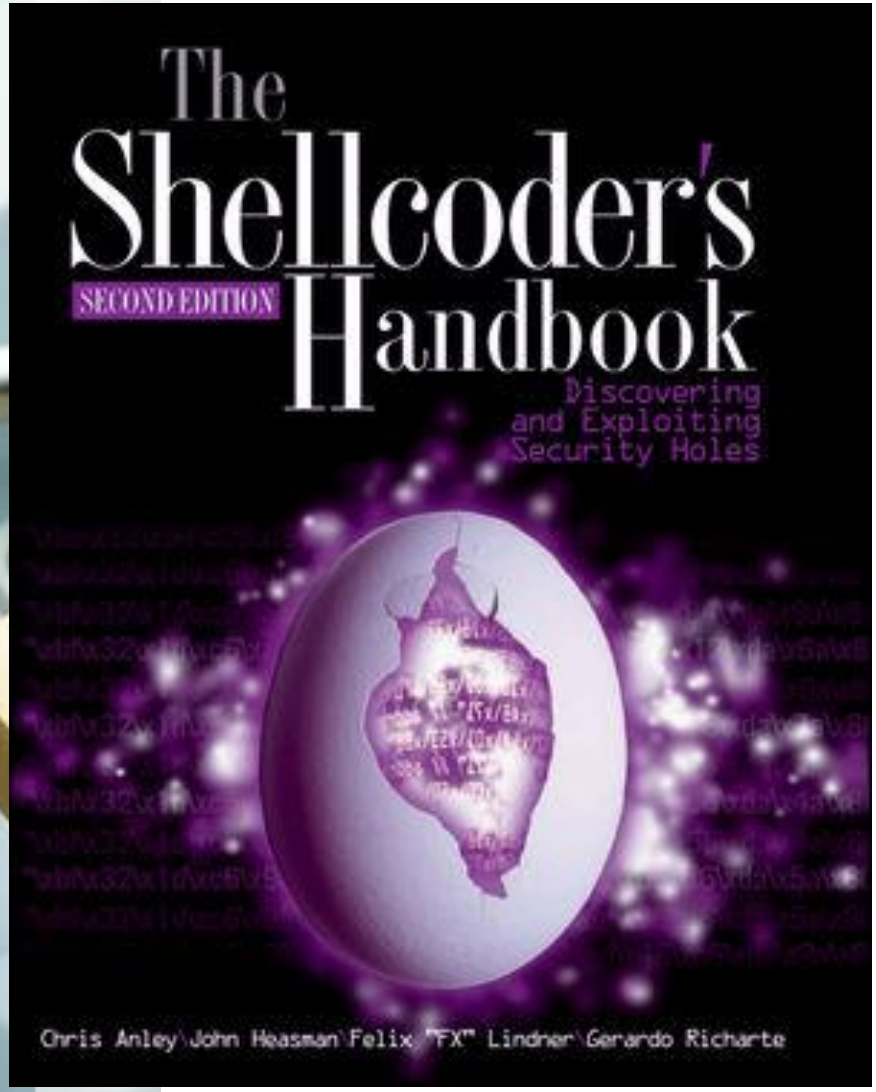
- How does memory corruption work?
- How does an exploit work?
- What exploit mitigations exist?
- How can these exploit mitigations be circumvented?

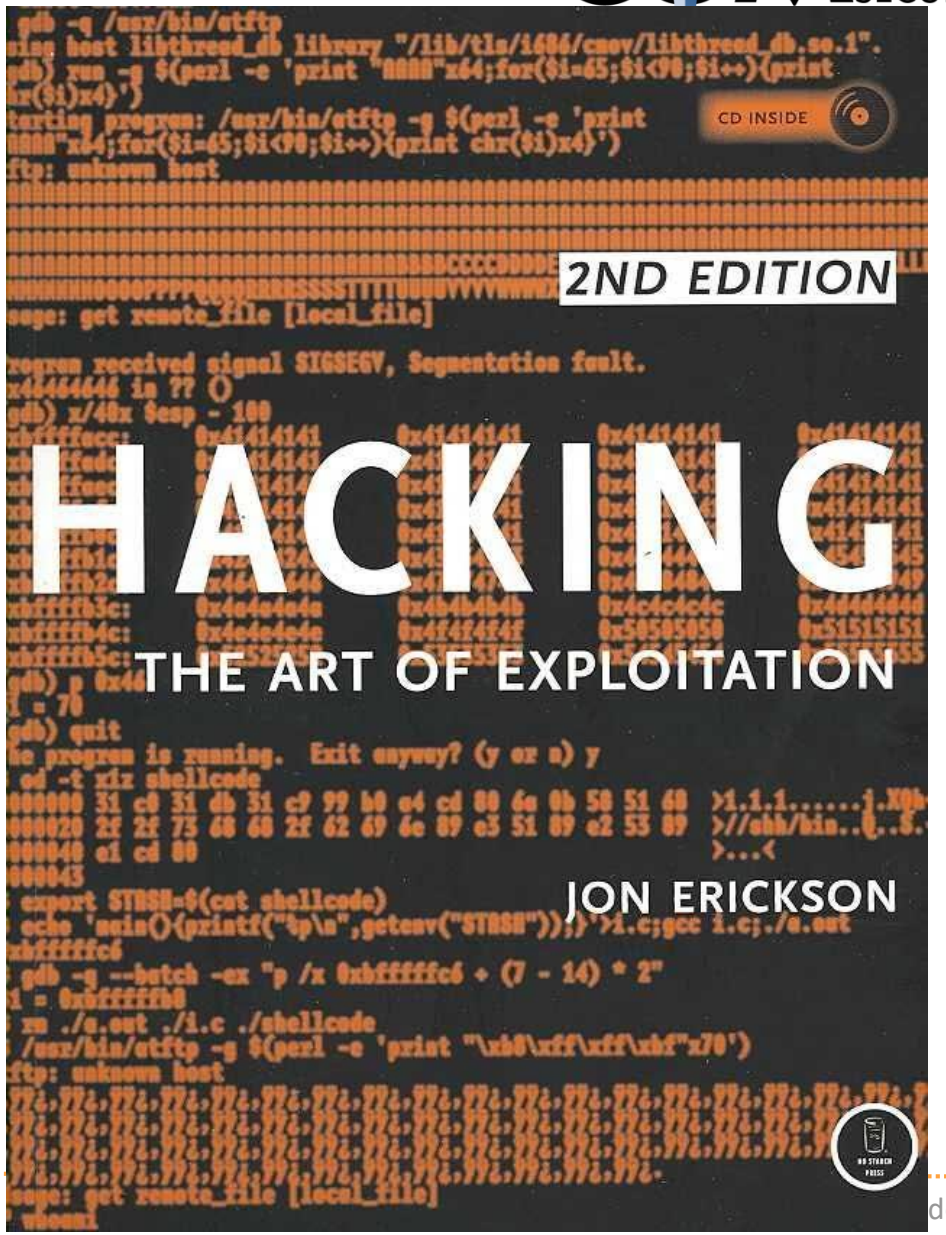
More theoretical, not so much the nitty gritty details

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 Issue

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

# 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