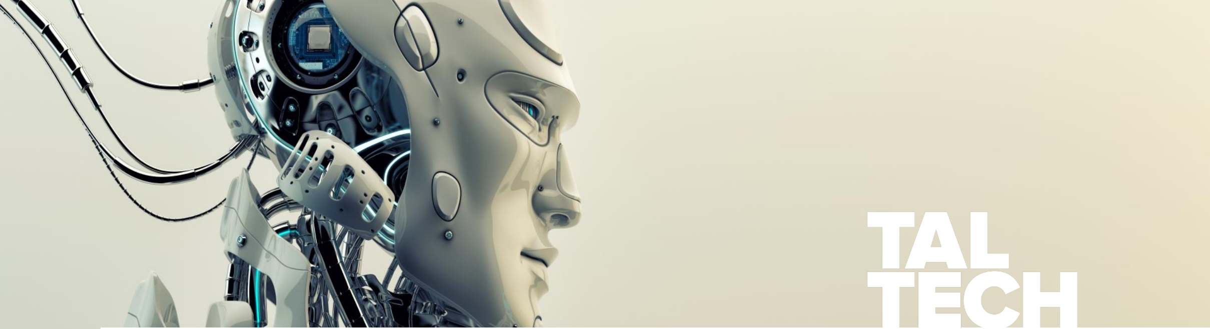

Educating Cyber Defenders

Technical Challenges in Designing Scalable Hands-on Learning Systems

Margus Ernits
margus.ernits@taltech.ee

IT College
School of Information Technology
Tallinn University of Technology

September 11, 2024

About presenter — Margus Ernits

- Visiting Lecturer at Estonian IT College
- Co-founder of RangeForce
- alumni of Barclays TechStar New York 2015 class
- Nominated three times as a Lecturer of the Year in Estonian IT College
- in-depth experience in GNU Linux and in IT Security and in Robotics fields
- Margus holds a Master of Science in Engineering Degree in Cyber Security (Cum Laude) a joint curriculum from Tallinn University of Techology and University of Tartu.

Introduction

Problem

Solution

Results

Conclusion

Introduction

- Cyber Defenders have to be trained to restpond to live-fire events as a team.
- The training of the cyber specialists is a challenging task because we need to train:
  - Individual skills with different categories of tools and techniques.
  - Team skills to work together in a team and be ready to respond to live-fire events.
- The training should be scalable and realistic.
- Cyber Defender: SOC, CERT, CIRT, CSIRT, Windows Security, Network Security, Linux Security, specialist.

Types of Cybersecurity Training and Exercises

- Capture the Flag (CTF) competitions
- Cyber Defense Exercises (CDX)
- Large-scale live-fire exercises
- Forensic challenges
- Online hands-on learning sessions
- Tabletop exercises

Technical Differences in Cybersecurity Training and Exercises

- Offensive techniques
  - Capture the Flag (CTF) competitions
  - Red Team exercises
  - We need a set of vulnerable systems to attack
- Defensive techniques
  - Cyber Defense Exercises (CDX)
  - Blue Team exercises
  - We need a set of systems to defend
  - Indicators of Compromise (IOC)
  - Forensic findings
- Team response
- Online hands-on learning sessions

Scalability and Realism

- Scalability - the ability to scale up or down the number of participants and teams.
- Scalability - time and resources to create the environment and content.
- Realism - the ability to create a realistic environment for the participants.
- Learning experience vs realism. Hands-on, problem-based learning has proven to be an effective method for teaching cybersecurity due to its realism and engaging activities.

```
kernel: [ 4929.449123] BUG: unable to handle kernel NULL pointer dereference at 0000000000000018
kernel: [ 4929.449236] IP: gen8_ppgtt_alloc_page_directories.isra.38+0x115/0x250 [i915]
kernel: [ 4929.449276] PGD 0
kernel: [ 4929.449276]
kernel: [ 4929.449301] Oops: 0002 [#1] SMP
kernel: [ 4929.449321] Modules linked in: ccm uas usb_storage rfcomm cmac bnep nls_iso8859_1 arc4 dell_wmi sparse_keymap dell_laptop iwlmvm dell_smbios dcdbas intel_rapl x86_pkg_temp_thermal
leo intel_powerclamp coretemp mac80211 videobuf2_vmalloc kvm_intel videobuf2_memops iwlwifi kvm btusb videobuf2_v4l2 btrtl btbcm videobuf2_core irqbypass videodev intel_cstate intel_rapl_perf
oth snd_hda_codec_realtek snd_hda_codec_generic snd_hda_codec_hdmi input_leds joydev serio_raw cfg80211 snd_hda_intel snd_soc_rt5640 lpc_ich mei_me snd_hda_codec snd_soc_ssm4567 shpchp snd_so
core snd_hwdep snd_soc_core snd_compress ac97_bus snd_pcm_dmaengine snd_pcm snd_seq_midi snd_seq_midi_event snd_rawmidi int3403_thermal snd_seq snd_seq_device snd_timer dell_smo8800 dw_dmac
kernel: [ 4929.449697]  snd_soc_sst_acpi dw_dmac_core snd_soc_sst_match snd elan_i2c soundcore acpi_als kfifo_buf int3402_thermal industrialio acpi_pad 8250_dw i2c_designware_platform spi_px
r_thermal_device i2c_designware_core int340x_thermal_zone mac_hid int3400_thermal int3406_thermal intel_soc_dts_iosf dell_rbtn acpi_thermal_rel intel_smartconnect parport_pc ppdev lp parport
utofs4 algif_skcipher af_alg dm_crypt hid_generic usbhid crct10dif_pclmul crc32_pclmul ghash_clmulni_intel pcbc aesni_intel i915 aes_x86_64 crypto_simd glue_helper cryptd psmouse i2c_algo_bit
syscopyarea sysfillrect libahci e1000e sysimgblt fb_sys_fops drm sdhci_pci ptp pps_core wmi sdhci_acpi video sdhci fjes i2c_hid hid
kernel: [ 4929.450052] CPU: 2 PID: 1467 Comm: Xorg Not tainted 4.10.0-19-generic #21-Ubuntu
kernel: [ 4929.450090] Hardware name: Dell Inc. Latitude E7450/0D8H72, BIOS A04 05/13/2015
kernel: [ 4929.450129] task: ffff898d4e5b4380 task.stack: ffffa30b01cf8000
kernel: [ 4929.450193] RIP: 0010:gen8_ppgtt_alloc_page_directories.isra.38+0x115/0x250 [i915]
kernel: [ 4929.450233] RSP: 0018:ffffa30b01cfb898 EFLAGS: 00010246
kernel: [ 4929.450260] RAX: ffff898bd0537040 RBX: 0000000000000003 RCX: 0000000000000003
kernel: [ 4929.450294] RDX: 0000000000000000 RSI: ffff898cd0d1a000 RDI: ffff898d4b670000
kernel: [ 4929.450331] RBP: ffffa30b01cfb8f0 R08: 0000000000000000 R09: 0000000000000000
kernel: [ 4929.450367] R10: 0000000000000000 R11: ffff898d5e7d3dc0 R12: ffff898d4b9ce000
kernel: [ 4929.450405] R13: ffff898d4fe077d0 R14: 00000000fedb9000 R15: 0000000000010000
kernel: [ 4929.450442] FS:  00007f825e5c3a40(0000) GS:ffff898d5e500000(0000) knlGS:0000000000000000
kernel: [ 4929.450484] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
kernel: [ 4929.450517] CR2: 0000000000000018 CR3: 0000000210083000 CR4: 00000000003406e0
kernel: [ 4929.450554] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
kernel: [ 4929.450591] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
kernel: [ 4929.450628] Call Trace:
kernel: [ 4929.450666]  gen8_alloc_va_range_3lvl+0xfb/0x9e0 [i915]
kernel: [ 4929.450698]  ? swiotlb_map_sg_attrs+0x49/0x110
kernel: [ 4929.450742]  gen8_alloc_va_range+0x23d/0x470 [i915]
kernel: [ 4929.450789]  i915_vma_bind+0x7e/0x170 [i915]
kernel: [ 4929.450832]  __i915_vma_do_pin+0x2a5/0x450 [i915]
kernel: [ 4929.450877]  i915_gem_execbuffer_reserve_vma.isra.31+0x144/0x1b0 [i915]
kernel: [ 4929.450930]  i915_gem_execbuffer_reserve.isra.32+0x39e/0x3d0 [i915]
kernel: [ 4929.450981]  i915_gem_do_execbuffer.isra.38+0x4a2/0x1750 [i915]
kernel: [ 4929.451015]  ? ___slab_alloc+0x212/0x4b0
kernel: [ 4929.451055]  i915_gem_execbuffer2+0xa1/0x1e0 [i915]
kernel: [ 4929.451102]  drm_ioctl+0x21b/0x4c0 [drm]
kernel: [ 4929.451142]  ? i915_gem_execbuffer+0x310/0x310 [i915]
kernel: [ 4929.451171]  do_vfs_ioctl+0xa3/0x610
kernel: [ 4929.451195]  ? __do_page_fault+0x266/0x4e0
kernel: [ 4929.451220]  SyS_ioctl+0x79/0x90
kernel: [ 4929.451241]  entry_SYSCALL_64_fastpath+0x1e/0xad
kernel: [ 4929.451267] RIP: 0033:0x7f825bfc1987
kernel: [ 4929.451288] RSP: 002b:00007fff93acac48 EFLAGS: 00000246 ORIG_RAX: 0000000000000010
kernel: [ 4929.451327] RAX: ffffffffffffffda RBX: 0000000000000005 RCX: 00007f825bfc1987
kernel: [ 4929.451363] RDX: 00007fff93acac90 RSI: 00000000c0406469 RDI: 000000000000000c
kernel: [ 4929.451401] RBP: 000000000000002a R08: 0000000000000000 R09: 0000000000000000
kernel: [ 4929.451438] R10: 00000000000047f8 R11: 0000000000000246 R12: 0000000000000040
kernel: [ 4929.451475] R13: 0000000000000008 R14: 000055629645a4f0 R15: 0000000000000000
kernel: [ 4929.451512] Code: e6 48 8b 90 20 03 00 00 48 8b b8 d8 02 00 00 48 8b 52 08 48 83 ca 03 e8 ca cd ff ff 48 8b 45 b0 48 8b 4d c8 48 8b 10 48 8b 45 d0 <4c> 89 24 ca 48 0f ab 08 0f 1f
ff 65 8b 05
kernel: [ 4929.451654] RIP: gen8_ppgtt_alloc_page_directories.isra.38+0x115/0x250 [i915] RSP: ffffa30b01cfb898
kernel: [ 4929.451700] CR2: 0000000000000018
kernel: [ 4929.464655] ---[ end trace 6e810281cb9cbfea ]---
```

Problem

- Stability — in cyber security training and exercises many moving things can break.
- Scalability — the ability to scale up or down the number of participants and teams and number of different exercises/lab-modules.
- Why do most existing defense-oriented systems focus primarily on identifying Indicators of Compromise (IOC) or forensic findings?
- How can a cyber defense-oriented learning system be designed where defenders not only identify threats based on log files or forensic evidence but also respond to live-fire events as a team?
- Large scale technical exercises and it is difficult to downscale for massive online learning cases.

Why this problem is not affecting offensive trainings?

- In case offensive trainings, we need a set of vulnerable systems to attack.
- We deplpoy a vulnerable system and don't update it.
- It is easy to create a vulnerable system and keep it vulnerable when we don't change it.

Why this problem is not affecting defensive IOC type trainings?

- Indicators of Compromise (IOC) type trainings and exercises need a set of vulnerable systems.
- We deploy a set of systems and don't update them. Defenders have to identify threats based on system events.
- Defenders don't need to modify the system and fix the vulnerabilities.
- Defenders don't need to remove the malware, backdoors, and other threats from the systems.

Why this problem is affecting defensive trainings?

- For defensive trainings, we need a set of systems to defend.
- We measure the success of the defenders by the number of successful attacks and user emulation.
- To demostrate team readyness to respond to live-fire events they have to:
  - Identify the threats.
  - Remove the malware, backdoors, and other threats from the systems.
  - Fix the vulnerabilities.
  - Respond to the live-fire events as a team.
  - Revoke attackers access to the systems.
- Why it is difficult?
- System updates and changes. Have you tried to pach AD that is not updated for 6 months?

Solution

- Technical architecture and methods for a scalable defense-oriented online learning system.
- Rebuilding lab environments dayly.
- Implementing auto-testing for the lab environments.
  - Automate lab environemt and depencencies testing on lab provisioning.
  - Automate a learner testing - auto-test after each lab-module build.

Automatic rebuilding of lab environments

- We have to rebuild the lab environments regurarly.
- We have to update the lab environment operating systems and software.
- When regurar rebuild fails we create module maintenance tickets for content developers.
- Why rebuilds fail?
  - Software and tool updates and changes.
  - Some vulnerable demand certan software versions.
  - Software conflicts.
- How to fix the rebuilds?
  - Automate the rebuilds and depencencies resolvings.

Automatic learner simulation and lab environment testing

- Automate lab environemt and depencencies testing on lab provisioning.
- Tooling: Packer, cloud-init, Ansible, Terraform, GitLab CI/CD, GitLab Runner, Azure Resource Manager
- Learner simulation: RDP, SSH replays. Expect, Cypress, Selenium, PhantomJS, Puppeteer.
- Learner simulation is executed after each lab-module build and regurarly on the live system.

Results

- In paper we give an Overview, what are the technical challenges in creating a scalable, defense-oriented cybersecurity learning system?
- The give set of tools, an architecture and methods to ensure stability and scalability.
- Our method has been tested in over 100 live-fire team exercises.
- We have more than 700,000 online learning sessions.
- Each month we have 100 module maintenance tickets created by automatic lab building learner's simulation.

Introduction

Problem

Solution

Results

Conclusion

Conclusion and Future Work

- Learner's simulation (autotest) is a key to the scalable defense-oriented online learning system.
- Defense oriented modules and exercises are more difficult to desing and maintain than offensive and IOC type modules.
- We do have 1000 online modules and exercises that are rebuilt and auto-tested daily or weekly.
- In the future we are planning more fully automated and tested team exercises and more hands-on modules.