BI.ZONE
Cybersecurity

St. Uriel
Education

Cybersecurity training course description

Table of content

I. Malware Analysis Fundamentals (2 days / 4-5 hours per session)

Description

The course introduces participants to the basic concepts of malware analysis (such as analysis methods, targets, and goals) and covers the evolution of this type of software. The participants will also learn about modern classification of malware by studying the standard behaviour of a malicious program and the techniques it utilizes. During the course, we will talk about malware analysis labs: how they are set up and what tools have to be used. We will also talk about modern solutions for protection against malware.

Requirements

No background in malware analysis is required. Experience in system programming is recommended.

Tools, concepts, and technologies covered

- Malware Analysis
- Malware Analysis Lab
- Virtual machine
- VMware
- MITRE ATT&CK
- Antivirus
- IDS/IPS
- SIEM
- Sandbox
- OllyDbg
- IDA Pro
- Wireshark
- Process monitor
- PEStudio

Topics covered

- Targets and objectives of malware analysis

- Malware analysis methods
- Malware evolution and classification
- Malware behaviour in system
- MITRE ATT&CK
- Malware analysis lab
- External resource used in analysis
- Introduction to essential tools for malware analysis
- Principle of operation of modern anti-malware solutions

Acquired skills and knowledge

- Students will be able to determine the level of severity of malware files
- Students will be able to set up a malware analysis lab and install the necessary software
- Students will get familiar with malware analysis tools and will be able to determine the course of further research
- Students will be able to perform initial analysis of suspicious files using open platforms

Course layout

- **Day 1**

During day 1, students will be introduced to the fundamentals of malware analysis. Understanding theory is necessary before studying the malware analysis as a practice.

Aside from examining the targets and objectives of malware analysis, we will cover various methods of analysis: static, behavioural, and code analysis. Students will also learn about the main types of malware, history of malware evolution, their distinct characteristics and purpose.

We will also examine typical attack scenarios and the impact of a malicious program on a target system. This part of the course will be concluded by an extended look at the MITRE ATT&CK matrix.

- **Day 2**

Analysis of malicious software is performed using a specific set of tools. This kind of software can offer full automation or semi-automation of the analysis. During day 2 of the course, we will examine the most essential of such tools. They are used for various methods of analysis.

Specifically, students will learn about web resources that are useful for early detection of malware type and its general functionality (e.g., VirusTotal, HybridAnalysis). Special attention will be paid to the setup and configuration of one’s own malware analysis lab that would contain all the tools necessary for research. The purpose of such laboratory is to isolate the research environment from the potential impact of the malicious program.

Students will also be familiarised with the features of modern protection solutions, specifically antivirus software: what it consists of, how it interacts with the system, and how a researcher can use it to track suspicious activity.

II. Introduction to Malware Analysis (5 days / 4-5 hours per session)

Description

The course covers the concepts and methods of static and behavioural malware analysis. Participants will familiarise themselves with the tools designed to analyse malware without running it (specifically, to obtain the files' metadata). Special attention will be paid to behavioural analysis tools: SysInternals utilities, sniffers, sandboxes, both proprietary and publicly available. In addition, architecture and some important components of modern Operating Systems (Windows, Linux) will also be covered.

Requirements

Experience in system programming is required. Knowledge of a modern OS internals and network protocols is recommended. Students will be required to take a mandatory test to determine their knowledge level.

Tools and technologies covered

- System API
- WinAPI
- Android API
- PEStudio
- file
- strings
- SysInternals
- Process Monitor
- APIMonitor
- Wireshark
- Fiddler
- VMray
- VirusTotal
- HybridAnalysis

Topics covered

- Modern OS architecture (Windows, Linux)

- Components of modern OSs relevant to an analysis
- The concepts of static and behavioural malware analysis
- Files' metadata acquisition and processing
- SysInternals tools
- Malware traffic research
- Sandboxes
- Static and behavioural analysis results

Acquired skills and knowledge

Students will be introduced to modern OS architectures and will know about the most important components which are the most interesting for analysis

Students will be able to extract and analyse malware metadata

Student will learn how to use behavioural analysis software

Student will be able to analyse reports generated by an automated behaviour analysis tool

Students will be able to perform behavioural analysis of suspicious files and generate a list of IoCs for the purpose of signatures creation

Course layout

- **Day 1**

Malware analysis requires extensive knowledge of architecture of the OS that the malicious program is launched on.

During day 1, students will learn about the high-level architecture of modern operational systems, i.e. Windows, Linux, and Android. Understanding high-level architecture is the first step in understanding the technologies used in the operation of malware.

We will also examine how processes, streams, and memory work in OS.

- **Day 2**

This part of the course includes a more in-depth look at the architectural characteristics of modern OSs.

Students will examine internal APIs of Windows, Linux: how various components of a system interact through such APIs, which calls are invoked in case of the file system and the network resources, how processes and streams are created, how processes manage memory resources.

- **Day 3**

On day 3, students are introduced to the concepts of static and behavioural malware analysis.

Students will work with utilities for collecting and processing metadata of executable files for different OSs (e.g. PEstudio, file, strings, CFF explorer). A set of such tools allows to perform early stages of malware analysis which entails identifying, the purpose and functionality of a malicious program, and, possibly, its C&C servers, all based on the APIs involved.

Static analysis allows a researcher to determine what programming language the malicious program was written in and whether the file was compressed. The results of a static analysis establish the direction of further examination of the program.

- **Day 4**

This session is focused on the basics of behavioural analysis without the use of complex tools for automated analysis (i.e., sandboxes).

Lectures on day 4 will cover tools for monitoring of system events and function calls (Process monitor, APIMonitor). Such utilities allow users to observe activity of malware that was hidden from them during static analysis.

Also, we will cover SysInternals utilities that can be used to examine in-depth the operation of internal components of the Windows OS during the operation of an analysed malicious program.

Another topic of the behavioural analysis session is the examination of network communication. Specifically, we will look at the Wireshark tool which allows us to capture and analyse every communication packet that the device sends and receives. However, if a malicious program is controlled via HTTPS, which encrypts all of its communication, the analysis gets more challenging. This is where the Fiddler utility proves itself useful, and it will also be covered during day 4.

- **Day 5**

During the final session of the course we will look at the tools for behavioural analysis automation. Specifically, VirusTotal, HybridAnalysis, VMRay, Cuckoo Sandbox.

Students will learn about the structure of reports generated by these tools and will also learn how to collect and process additional data. We will focus on VMRay and Cuckoo Sandbox since these systems can be activated in an analysis lab inside its network. These sandboxes can be used to open and analyse both executable (PE, ELF) and non-executable files (Microsoft Office documents, PDF, JavaScript, etc.).

Students will learn about the nuances of configuring and using a sandbox.

III. Reverse Engineering (5 days / 4-5 hours per session)

Description

Reverse software engineering is the most advanced and difficult method of malware analysis. Reverse engineering helps identify the hidden capabilities of malicious programs which are impossible or difficult to obtain by means of static and behavioural analysis. As a part of the course, students will learn about reverse engineering tools and their application. The course contains information on the principles of disassembly, assembler instructions, as well as the basic concepts of software operation on the processor level.

Requirements

Experience in system programming and knowledge of a modern OS internals are required. Experience with an assembler and Python programming language is recommended. Students will be required to take a mandatory test to determine their knowledge level.

Tools and technologies covered

- OllyDbg
- IDA Pro
- x64dbg

Topics covered

- Targets and objectives of reverse engineering
- PE and ELF file structure
- Processor primitives (differences between x86 and x64)
- Assembler instructions
- Assembly programming concepts
- Unpacking and deobfuscation
- Malware functionality identification
- Anti-debugging and anti-disassembly techniques
- Using IDAPython for automated malware reverse engineering

Acquired skills and knowledge

- Students will be able to analyse code using IDA Pro, OllyDbg, x64dbg
- Students will learn how to unpack a malicious program and identify its functionality
- Students will learn what anti-debugging and anti-disassembly techniques are used and how to bypass them
- Students will be able to reverse engineer malware

Course layout

- **Day 1**

During the first session, students will be introduced to the objectives and targets of reverse malware engineering (code analysis). They will also learn about the structure of executable files of Windows and Linux Operating Systems.

The lecture will also cover tools used to create executable files, specifically, the purpose of using various compiler settings.

Students will also learn about the components of an executable file: headers, code sections, import and export sections. They will also be familiarised with tools for disassembly and examination of an executable file (OllyDbg, IDA Pro).

- **Day 2**

Reverse engineering of a malicious program requires a solid grasp on base processor primitives and assembler instructions.

During this session, students will be introduced to processor registries and their purpose, as well as the FLAGS registry.

Students will learn the difference between address spaces of x86 and x64 architectures. We will also cover the structure of an address space.

The session will also pay a lot of attention to assembler instructions: arithmetic, logical, conditional, memory. Students will learn when each type of instructions is applicable.

- **Day 3**

Third session of the course is dedicated to the programming constructs which can be identified during disassembly: conditional statements, loops, and function calls. Explanation of the latter entails examination of calling conventions and usage of local and global variables.

Students will also be introduced to tools for disassembly and debugging: IDA Pro, OllyDbg, x64dbg.

Another topic of the session is compression and unpacking of a malicious program. Students will learn what tricks malware developers use to hide the functionality of their programs from analysts, as well as methods of obtaining the original code of a malicious program.

- **Day 4**

During this session, students will learn about malware obfuscation techniques.

However, the focus will be on identification of malware functionality. Since static and behavioural analyses do not provide a full picture of a malicious program operation, a researcher often needs to perform code analysis. During this stage, a researcher may obtain every section of the code, even hidden ones and those that are called under certain conditions.

Functionality identification includes the search of persistence methods and C&C communications, and documentation of executable commands.

- **Day 5**

Final day of the course will have students examine the anti-debugging and anti-disassembly techniques. They serve the purpose of preventing a researcher from analysing a malicious program with the previously discussed tools.

Anti-debugging techniques are used to monitor analysis attempts and most often involve shutting down or self-destruction of the program. Anti-disassembly techniques try to disrupt the work of a disassembler which poses obstacles for the research.

We will also look at the mechanism of making extensions for the IDA Pro debugging tool. This involves usage of IDAPython, an extension of IDA for writing scripts in Python programming language. Students will use this extension to create simple plug-ins that make sample inspection easier.

IV. Non-PE Malware Analysis (5 days / 4-5 hours per session)

Description

Malware analysis does not boil down to analysis of executable files. Malware can be presented in the form of user documents, such as Microsoft Office documents or Adobe PDF. These documents can contain both exploits for vulnerabilities and components executed by the users themselves. The course covers the methods of malicious document analysis, the tools used for this, and the challenges of such analysis. In addition, the course will examine the malware analysis for the Android OS.

Requirements

Special knowledge about document structure is not required. Software development experience for Android OS or knowledge of Android OS internals is recommended. Students will be required to take a mandatory test to determine their knowledge level.

Tools and technologies covered

- OfficeMalScanner
- Shellcode
- OllyDbg
- pdfid
- pdf-parser.py
- AndroidStudio
- jd-gui
- jeb

Topics covered

- Microsoft Office files analysis
- OfficeMalScanner usage
- Shellcode analysis
- RTF files analysis
- PDF files structure
- Analysis of parts of PDF files
- Extraction and examination of JavaScript and VBScript
- APK files structure
- APK files creation and modification

- APK files reverse engineering

Acquired skills and knowledge

- Students will learn how to analyse Microsoft Office files
- Students will be able to parse PDF files and analyse the extracted structures
- Students will get familiar with the structure and the process of analysing an APK file
- Students will be able to analyse suspicious documents and APK files

Course layout

- **Day 1 (MS Office and Shellcode)**

Aside from executable files, user documents can also pose a threat, specifically, various Microsoft Office files (doc/docx, xls/xlsx).

Students will work with OfficeMalScanner suite of tools which is used to parse Microsoft Office files, locate anomalies, and provide the researcher with separate parts for analysis.

What sets OfficeMalScanner apart is its ability to search a Microsoft Office file body for shellcode. This is why the second part of the session will be dedicated to the concept of shellcodes. Students will learn about building shellcodes, their main components and ways to examine those components.

- **Day 2 (PDF and JavaScript+VBScript (powershell))**

PDF is another common threat because of its popularity in the enterprise. For this reason, this session will be focused on understanding the structure of a PDF file. Also, we will cover tools for parsing files of this type, such as pdfid, pdf-parser.py, Origami Framework.

Parsing a PDF leads to the extraction of the data relevant to the analysis. We will look at JavaScript code sample as an example of such data. Students will be introduced to the methods of JavaScript obfuscation and debugging tools.

Following up on the topic of the use of scripting languages in malware, we will look in similar fashion at the VBScript language which is commonly used in malicious Microsoft Office documents.

- **Day 3 (Android OS and APK structure)**

As part of non-executable malicious files analysis, we will look at APK files which are essentially application files for the Android OS.

Understanding the principles of APK file analysis requires understanding of the basic concepts and architecture of Android OS, the structure of APK file: its components and their interaction, as well as the APK file's interaction with the OS.

These topics will be the focus of the third day of the course.

- **Day 4 (Creation and modification of an APK file)**

During this session students will learn how an APK file is created and modified.

We will take a close look at the AndroidStudio development environment: how it is set up, how the APK components are presented in such environment, and how to debug an application.

We will talk about essential characteristics of programming in Java and Kotlin languages.

Students will learn how to decompile an application, modify it and compile a working application.

- **Day 5 (Reverse engineering of an APK)**

Having gained knowledge in the aforementioned sessions, students will be able to perform reverse engineering of an APK file.

Along with tools such as jd-gui and jeb and their functionality, students will learn how to debug an application on a virtual device using AndroidStudio and examine an APK on an actual mobile device.

V. Basic Digital Forensics and Incident Response (5 days / 4-5 hours per session)

Description

The course covers incident response in case of a successful attack detection and the infection of an enterprise's system. The course also covers the sequence of actions during response, as well as the basics of computer forensics, which includes searching for traces of the system compromise, attack vector identification, and the extent of malware infection.

Requirements

Basic knowledge of Windows architecture and Linux commands is required. Basic knowledge of computer forensics is recommended. Students will take a mandatory test to determine their knowledge level.

Tools and technologies covered

- FTK Imager
- Volatility
- Log2timeline
- Autoruns
- Registry Explorer
- The Sleuth Kit
- WinPrefetchView
- DB browser SQLite
- PhotoRec

Topics covered

- CERT activity
- Memory dump creation and analysis
- Volatility framework
- Disk dump creation
- Timeline creation and analysis
- NTFS file system
- Windows OS file artefacts
- Windows registry analysis
- Browser artefacts analysis

Acquired skills and knowledge

- Students will learn about a typical incident response process
- Students will learn how to acquire memory dumps and disk images and will be able to perform basic forensic analysis
- Students will learn about artefacts usually left by malware on computer systems during an attack
- Students will be able to respond to a typical incident

Course laylout

- **Day 1**

During the first session students will be introduced to the general principles of a CERT (computer emergency response team), as well as standard procedures during incident response and incident investigation.

Students will learn how to acquire a disk image and a memory dump (using FTK Imager Lite software) and get familiar with tools to process and analyse artefacts from this evidence. The session also covers the necessity of preserving the disk data integrity and the importance of a memory dump.

- **Day 2**

RAM contains the current state of a system and the context of every running process and used files. This is why memory dump is an essential piece of evidence for a digital investigation.

During the second session students will be introduced to the Volatility framework which is used to analyse RAM dumps. The framework has a modular architecture, where every plugin serves a specific analysis task. Students will learn about essential plugins required for examination of a dump: process information, registry information, file plugins and virtual process memory plugins.

The session also covers anomalies in RAM that point to malware execution in a system. Students will learn how to extract executable files and malicious code sections from RAM.

- **Day 3**

The main artefact of a disk image is the file system. To understand the principles of a file system structure, we will take a look at NTFS. Students will learn about the NTFS structure and its distinct features relevant to the analysis. We will also extensively cover the concept of file as a structural part of the file system and attributes at file system level.

Along with NTFS characteristics, students will also examine some interesting artefacts of Windows OS which is one of the most popular systems in the corporate environment. This topic will cover malware launch monitoring, persistence methods, and analysis of Windows event log.

- **Day 4**

Following up on the topic of the previous day, we will look at a variety of important aspects of investigation of an incident involving Windows-based machines.

Specifically, we will look at the structure of the Windows registry and the data that can be extracted from it. For this purpose, students will be introduced to the Registry Explorer utility which is used for learning system settings of the OS, previously launched software, identifying opened files and directories.

Along with registry analysis, students will get familiar with browser history examination which may help identify the infection vector.

Also, we will look at the tools for processing users' mail history. This is important because email phishing is a common way to attack a corporate network.

- **Day 5**

During the final day of the course students will learn to create a timeline, a list of system events sorted by their time. Timelines are used for understanding the attack scenario and identifying system anomalies.

Students will learn how to create and analyse file system timeline using The Sleuth Kit (TSK) utilities. The course will be concluded with the discussion of the process of searching and proving malware operation on a system.

VI. Advanced Digital Forensics and Incident Response (5 days / 4-5 hours per session)

Description

This training course covers in-depth forensic analysis, including inner workings of Windows operating system, and how to successfully investigate and respond to complex incidents in a corporate environment.

This course also covers evidence acquisition and basic memory analysis of Linux systems.

Requirements

Basic knowledge of Windows architecture and Linux commands is required. Basic knowledge of computer forensics is recommended. Students will take a mandatory test to determine their knowledge level.

Tools and technologies covered

- The Sleuth Kit (TSK) utilities and tools
- Volatility and Rekall memory analysis frameworks
- GRR Rapid Response
- Plaso (log2timeline)
- AnalyzeMFT
- Dfir_ntfs
- yarp
- grep
- strings
- dd

Topics covered

- Detailed file system (NTFS) analysis with TSK
- In-depth memory dump analysis and recovery of malicious executables and code from memory using Volatility and Rekall frameworks
- Linux memory dump acquisition and analysis
- Super-timeline creation and analysis
- Detecting anti-forensics techniques and timeline anomalies
- Windows Registry parsing, timeline creation and recovery of deleted records
- Windows Event Log analysis and recovery
- WMI, PowerShell and JScript activity on compromised systems

- Common malware persistence locations on Windows systems
- MITRE ATT&CK™ matrix and how to use it
- Using and developing Indicators of Compromise (writing and using YARA signatures)
- Detection and basic analysis of suspicious files (common packers, entropy checks, timestamp anomalies, malware identification)

Acquired skills and knowledge

- Students will be able to perform in-depth forensics analysis of Windows systems, detect anomalies and identify suspicious activity
- Students will learn how perform detailed memory dump analysis with Volatility framework and locate malicious code
- Student will know how to promptly develop indicators of compromise based on their findings during forensics analysis of compromised systems, and use them to search for malware on larger network
- Students will understand common TTPs (Tactics, Techniques and Procedures) used by modern cybercriminal groups

Course layout

- **Day 1**

During the first session, students will learn about essential disk structures – MBR, partition tables, Volume Boot Records (VBRs), various types of file systems and tools that could be used to analyse them.

Students will learn about inner workings of NTFS file system, differences between NTFS and FAT32, exFAT and EXT3/4, system files on NTFS partitions ($MFT, $UsnJrnl and $LogFile) and how information from these structures could be used in digital forensic investigations and incident response (DFIR) process.

Students will also learn about file system timestamps, their formats and how to detect timestamp tampering and anomalies, about Volume Shadow Copy mechanism, NTFS links and directory junctions.

Students will practice using test disk image from compromised system with tools from The Sleuth Kit (TSK) and other open-source utilities, including analyzeMFT and dfir_ntfs.

- **Day 2**

During the second session, students will learn how Windows manages memory – the difference between virtual memory, physical memory, VADs, various methods of memory allocation, essential Windows kernel structures and process management, and how all these things could be used to identify malicious activity on a Windows system.

Students will also learn about the differences in memory analysis between various Windows versions, starting from Windows XP to Windows 10, and how to correctly acquire memory images on recent Windows releases (10+).

Some of the time will be also dedicated to the acquisition and analysis of Linux memory dumps, and how to build Linux kernel profiles for Volatility Framework.

Students will get hands-on practice with several memory dumps with various malicious artefacts and real-world malware using Volatility memory analysis framework. Students will learn about the whole package of Volatility plugins used for memory dump analysis.

- **Day 3**

During this day, students will learn about all Windows artefacts that could be analysed to identify malicious activity and malware execution on compromised system, including the following:

- Windows Application Compatibility Cache
- AmCache.HVE and RecentFileCache.bcf files
- Prefetch Files
- Windows LNK files and Startup folders
- WScript.exe and CScript.exe
- Windows Task Scheduler
- Windows Explorer Shell Extensions

Along with these, students will learn how to analyse WMIC and PowerShell activity.

Special attention will be given to the analysis of Windows Event Log files and their recovery in case of their deletion by the attackers.

- **Day 4**

During this day of the training, students will acquire an understanding of the inner workings and structures of Windows Registry – the database that stores all the configuration parameters and system settings of a Windows system. Students will learn how to carve and recover deleted registry entries and detect anomalies.

After that, students will learn how to create a super timeline, a list of all system events sorted by their timestamps using log2timeline/Plaso tools.

Events include these from Windows Event Log, Registry timestamps, browser history events (visited sites etc.), email messages, file activity (MACB), program launch artefacts (Application Compatibility Cache, Prefetch etc.) and so on. Super timelines are used for understanding the attack scenario and identifying system anomalies.

- **Day 5**

In the course of this day, student will get a quick run-down of the various kinds of malicious files and tools which could be used for their express analysis.

- Then, the training will conclude with a hand-on exercise – a real-world investigation concerning the case of a Windows system compromise, during which students will need to identify all elements of an attack – from initial compromise vector, to all the other post-exploitation activity on the infected system. This way students will test all their skills and knowledge acquired during this training course.

VII. Malware Analysis Overview

This training will be conducted once every three months. The duration of each session will last 2 hours.

The training is aimed to provide the audience with a digest of important events and updates on the current cybersecurity landscape.

Participants will learn about key events, essential cybersecurity bulletins by various vendors, and notorious cyberattacks. The audience will also be introduced to recently emerging attack vectors: what they consist of, which tools are involved, and how such attacks can be mitigated.

The information presented during the training is meant to assist the participants in defence against known threats.

VIII. Red Team Operation

Description

Red Team operations is a necessary part of any modern security exercise. This exercise means the simulation of real APT attack on the company. BI.ZONE regularly takes part in investigations of real hacker attacks with different attacker motivation - from hacktivism to state actor. Also, BI.ZONE specialists have successfully accomplished many projects in this field, that gave a lot of experience depending on region, infrastructure design and maturity of the client company.

During this course, our specialists will explicate complex chain of Red Team operations. BI.ZONE specialists will demonstrate and explain different approaches of every stage. As an example, initial access can be performed by vishing, road apple, phishing, using malware payload or fake website and 2nd factor bypass. BI.ZONE specialists have unique techniques and knowledge that will be revealed during this course.

Topics

- Bypassing modern AV software
- Make trustful phishing emails
- Interaction with purple team
- Privilege escalation on windows and linux servers
- Make secured hidden channel
- Hiding from forensic investigation

Requirements

- Good network protocols knowledge
- Good programming skills
- Basic enterprise it architechture knowledge
- Linux and Windows operation system expertise
- Familiarity with Windows Active Directory concept

Acquired knowledge:

- How to perform real life APT attack simulations
- How to prepare phishing emails
- How to escalate your privileges on adversary endpoints
- How to hide from Blue Team

- How to make legal arrangements
- How to find vulnerabilities in different applications

Course layout

- **Day 0. INTRO TESTING**

Entry-level testing is required to attend this course.

- Entry-level testing to check requirements fulfilment

- **Day 1. INTRODUCTION**

This day will set up the baseline in terms and definitions.

The introduction in Red Teaming is really important because cybersecurity is a separate discipline which require quite a long education. Introduction will help to align the knowledge of the group to make them easier to understand the topics.

The students will gain an overview of modern cybersecurity, listen to real-cases and learn about the differences between the various kinds of cybersecurity services, such as penetration test, security assessment and Red Teaming. They will also learn the terms about Red Teaming.

This day will cover the following topics.

- Red Team vs Penetration Test vs Security Assessment vs Vulnerability Scanning
- Overview of modern APT attacks and BI.ZONE real-world cases
- Setting up, Legal basis in different countries
- Philosophy of Red Teaming
- What are Red, Blue and Purple Team
- Bash basics + CLI programs
- Kali / Blackarch / Backbox

- **Day 2. RECONNAISSANCE**

OSINT stands for *Open Source Intelligence*, and it is one of the key aspects in understanding the cybersecurity that rules the Internet these days. **Open source intelligence** (**OSINT**) is information collected from public sources such as those available on the Internet, although the term isn't strictly limited to the Internet, but rather means all publicly available sources. It's really important to know as much as you can about your adversary. This will help to find the weakest spot in company defensive mechanisms. The following topics will be discussed:

- Defining an information needed

- External Reconnaissance
    - Scope Discovery
        - Google dorks
        - Search engines techniques
        - Attack Surface platforms
    - OSINT methods
    - Active interaction
        - Spidering
        - Physical reconnaissance
        - Staff interaction techniques
- Real world examples of reconnaissance into successful attack
- Practical reconnaissance task for specific company

- **Day 3. INITIAL ACCESS**

This day will cover interaction with company networks, scan mechanisms, their types and where hide mechanisms will be explained. This topic is important due to high amount of sensitive resources usually available on the company's external network perimeter. This will give us a chance to find and exploit vulnerabilities and get access to internal network. External perimeter vulnerabilities may help us to extend attack surface and increase our chances of a successful attack.

- Scanning methods and tools in-depth
- Hiding behind proxies, cloud providers, VPNs, TOR, bulletproof service providers
- External Assessment Toolkit
- Vulnerability Exploitation
    - Web Application Assessment
    - Network Services Assessment
    - Practical scanning and version enumeration task

- **Day 4. SOCIAL ENGINEERING**

Social engineering attacks are not only becoming more a common thorn in the side of enterprises, but they're also become increasingly sophisticated. Social engineering attacks typically involve some form of psychological manipulation, fooling otherwise unsuspecting users or employees into handing over confidential or sensitive data. Commonly, social engineering involves email or other communication that invokes urgency, fear, or similar emotions in the victim, leading the victim to promptly reveal sensitive information, click a malicious link, or open a malicious file. At this moment it's the most frequent way to get into the company perimeter, APT groups often use it in their attack.

- Email phishing
- Voice phishing
- Going for an interview
- Real-world cases of successful phishing + whaling
- Real life phishing scenario demonstration with getting attack results

- **Day 5. PAYLOADS - 1 (Advanced)**

Payloads is the most sophisticated thing in getting access to the company resources. The requirements to the payloads are quite wide because they should run on every victim machine without crashing. During this day the students will know how to create a good payload, how to test it, how to check the environment of the victim and detect sandboxes. This course will cover the following topics:

- Toolset observation
- Macros, name juggling, .lnk, .url, .scf etc filesTypes of shell codes

- **Day 6. PAYLOADS CONCEALING - 2 (Advanced)**

This topic will cover ways to bypass company defensive mechanisms such as AV, IDS, corporate proxies, HIDS and other. This includes covert channels and custom traffic encryption mechanisms. The students will study tools and methods of hiding from blue team to make a real-life highly effective Red Team operations. On this day the following topics will be covered:

- Tunneling techniques (DNS tunneling, gist)
- In-memory loading techniques
- Antivirus bypass testing
- Encoders and packers

- **Day 7. ATTACK DEVELOPMENT**

This day will cover possible lateral movement during Red Team operation. Enterprise networks are enormous in size. It does not require the creation of your own attack management tools from scratch. There are different frameworks available for use during the attack. The most popular frameworks will be covered and explained. Also, the students will know how to write additional plugins for the frameworks. The following topics will be covered:

- Engagement management
  - Metasploit Framework
  - Cobalt Strike
  - Powershell Empire

- Searching / patching / writing exploits
  - Python basics for exploitation
  - Exploit-db

- **Day 8. ACTIVE DIRECTORY AND NETWORK SECURITY**

Active directory is the cornerstone of almost every company. That is why it is a prime target in virtually any attack, and attackers know just how crucial it is in their quest to find and steal what they are looking for. If an attacker has enough privileges in AD, they can get access to almost every information in the company. Kerberos is the authentication scheme used in AD. The deep knowledge of this scheme gives an advantage in developing attacks against corporate infrastructure. In addition, during the day, students will learn about attacks on popular network protocols. The following topics will be covered:

- CDP / DTP / STP
- DNS / DHCP
- NBTNS / LLMNR / WPAD + responder
- Active Directory popular attacks
- Attacks on Kerberos in-depth

- **Day 9. PRIVILEGE ESCALATION**

Privilege Escalation consists of techniques that adversaries use to gain higher-level permissions on a system or network. Adversaries can often enter and explore a network with unprivileged access but require elevated permissions to follow through on their objectives. Common approaches are to take advantage of system weaknesses, misconfigurations, and vulnerabilities. The following topics will be covered during a day:

- Linux
- Windows
- Practical tasks on privilege escalation

- **Day 10. POST EXPLOITATION**

The purpose of the Post-Exploitation phase is to determine the value of the machine compromised and to maintain control of the machine for later use. The value of the machine is determined by the sensitivity of the data stored on it and the machines usefulness in further compromising the network. During this day students will learn how to find the easiest ways to gather information in the host and get additional access with user-interaction or without. This part is really important due to the huge size of a corporate network. Successful post-exploitation leads to fulfilment of all the Red Team objectives.

Non-technical post-exploitation

- keyloggers
- screengrabbers
- fakecreds
- Lazagne
- Executables spoof (putty)

Understanding target infrastructure

- Bloodhound
- Crackmapexec
- Pass-the-hash (+psexec)

- **Day 11. PERSISTENCE (Advanced)**

Getting the persistence is a part of playing hide-and-seek against the Blue Team. It's really an art of hiding in the depth of operation system executable files, processes and logs. This day students will be exposed to the different approaches of hiding their payloads and getting really persistent access to victims' machines. The approaches will cover different architectures and operating systems:

- Windows
- Linux
- Network equipment

- **Day 12. LATERAL MOVEMENT**

Lateral movement is a means to an end; a technique used to identify, gain access to and exfiltrate sensitive data. The students will know how to pivot through adversary network, how to hide from detection systems and how to find a right way into the corporate networks. The following topics will help to better understand network topology and its weakest parts:

- Pivoting methods
- Tools
- Understanding of and looking for patterns

- **Day 13. WIRELESS SECURITY**

The usage of wireless network in corporate networks is drastically increased over past few years. The wireless network is one of the targets of potential attackers. During this day the students will study the most popular attacks on wireless network protocols. These attacks can give a member of the Red Team access to the internal corporate network. The following topics will be covered:

- WEP
- WPA2-PSK
- WPA2-Enterprise
- Rogue Access Point
- ARP spoof wireless points

- **Day 14. ORGANISATIONAL QUESTIONS (Advanced)**

Before engaging in Red Team operation it is crucial to discuss all the legal issues related to this topic. This day covers pre-engagement stages of Red Team operation, on-the-day communication with Purple and Blue Teams to maximise the efficiency of the project, followed by reporting after Red Team operation ceases. Also, we will discuss the proper infrastructure that the Red Team should have to perform successful attacks. The following topics will be covered:

- Communicating with the Blue Team
- Proper reporting
- Red Team infrastructure and team assessment
- House cleaning

- **Day 15. Searching vulnerabilities (Advanced)**

Fuzzing is a software testing technique, which basically consists of finding implementation bugs using malformed/semi-malformed data injection in an automated fashion. This is one of the ways to find and exploit new vulnerabilities. During this day students will learn how to fuzz application in a correct way using automated tools and how to identify one-day vulnerability. We will discuss real-life examples finding vulnerabilities in open-source software. The following topics will be covered:

- Fuzz testing
- 1day vulnerabilities
- Searching for bugs in open source software

RELATED COURSES AND CERTIFICATIONS:

- OSCP
- OSCE
- LPT
- GXPN

- GPEN

Courses:

- SANS SEC760
- SANS SEC564
- SPECTREOPS Red Team operation

IX. Web Application Security

Description

Web application security course – a special course developed by BI.ZONE to share its knowledge in finding and exploiting vulnerabilities. Our specialists have daily experience in web application security analysis, which helps us to keep our techniques up-to-date. The course is aimed not only at vulnerability exploitation, but also at searching vulnerabilities and explaining how to protect web application against the risks of being exploited. That is why the course will be interesting to application developers and application security specialists.

Prerequisites

Web application development skills. Basic cybersecurity knowledge

Tools and technologies covered

- Historical approaches to web app security analysis
- Methodologies of web app analysis
- Classification of vulnerabilities
- Modern browser techs and protection measures
- Impact on exploitation of XSS vulnerabilities
- Exploitation of XSS vulnerabilities
- Frameworks for XSS exploitation
- Security measures against XSS
- Practice tasks on exploitation
- Client-Side Vulnerabilities *-
- CSRF exploitation
- Open Redirect exploitation
- CSTI and JS hijacking exploitation
- Real cases of XSS auditor bypass
- Browser technologies, such as CSP, SOP, etc
- Real cases of exploitation
- Practice tasks on exploitation
- Server-side vulnerability classification

- Specific things on programming language and web server configuration using in vulnerability exploitation
- Typical vulnerabilities for programming languages
- SQL injection classification, how to find and exploit
- XXE injection, actual exploitation in 2019
- OS injections, CGI injections and many others
- Security measures to prevent exploitation of “injections”
- Practice on injections
- SSRF - how to find and exploit
- Insecure deserialization in different programming languages
- Authentication vulnerabilities – how to exploit and protect
- Insecure Direct Object Reference – how to find exploit and mitigate
- Business logic vulnerabilities – eternal flaws of web apps

Topics Covered:

- The bundle of tools for web app security analysis;
- Web application security analysis methodologies and approaches;
- Vulnerability classification;
- The nature of web app vulnerabilities;
- The measures should be applied during development to keep web apps secure;
- The real-life cases of penetration;

Acquired skills and knowledge:

- Conduct web application security assessment;
- Detect web application vulnerabilities;
- Exploit web app vulnerabilities using automated tool;
- Exploit web app vulnerabilities manually;
- Fix vulnerabilities and mitigate risk of vulnerabilities being exploited.

Course layout

- **Day 1:**

Understanding the attacker's perspective is key to successful web application penetration testing. The course begins by thoroughly examining web technology, including protocols, languages, clients, and server architectures, from the attacker's perspective. We also examine different authentication systems, including Basic, Digest, Forms, and Windows Integrated authentication, and discuss how servers use them and attackers abuse them. After authentication, we analyse the importance of encryption and HTTPS.

We then turn to the four steps that make up our process for conducting web application penetration tests: reconnaissance, mapping, discovery, and exploitation. On the first day, we review the fundamental principles of each phase and discuss how penetration testers can use them together as a cyclical in-depth attack process. We then cover the types of penetration testing and what pieces need to be part of a thorough, high-value pentest report. To round off the day, we will explore aspects of a vulnerable web application using Burp Suite.

- **Day 2**

The second day begins with the reconnaissance and mapping phases of a web app penetration test. Reconnaissance includes gathering publicly available information regarding the target application and organisation, identifying the machines that support our target application, and building a profile of each server, including the operating system, specific software, and configuration. The discussion is underscored through several practical, hands-on labs in which we conduct reconnaissance against in-class targets.

In the mapping phase, we build a map or diagram of the application's pages and features. This phase involves identifying the components, analysing the relationship between them, and determining how the pieces work together. We often discover configuration flaws in web application infrastructure components during the mapping phase. We then dive deep into spidering/crawling web applications. Spidering represents a vital part of both the mapping phase and the overall penetration test.

- **Day 3**

This section continues to explore our methodology with the discovery phase. We build on the information identified during the mapping phase, exploring methods to find and verify vulnerabilities within the application. Students also begin to explore the interactions between the various vulnerabilities.

This day we dive deep into vital manual testing techniques for vulnerability discovery. To facilitate manual testing, we kick off the day with an introduction to Python and a hands-on lab working with it.

In addition to custom scripts, we focus on developing in-depth knowledge of interception proxies for web application vulnerability discovery. A highlight of the day involves spending significant time working with server-side vulnerabilities such as sqlI, OS command injection, xxe, ssti for different frameworks.

Throughout the discovery phase, we will explore both manual and automated methods of discovering vulnerabilities within applications and discuss the circumstances under which each is appropriate.

- **Day 4**

Students continue exploring the discovery phase of the methodology. We cover methods to discover key vulnerabilities within web applications, like client-side vulnerabilities such as open redirect, different types of xss and others. Manual discovery methods are employed during hands-on labs.

After detailing the various vulnerabilities and manual discovery methods, day four concludes with a review of various automated web application vulnerability scanners, to complement our previous coverage of manual techniques with scripting, ZAP, and the Burp Suite.

- **Day 5**

On the fifth day, we launch actual exploits against real-world applications, building on the previous three steps, expanding our foothold within the application, and extending it to the network on which it resides. As penetration testers, we specifically focus on ways to leverage previously discovered vulnerabilities to gain further access, highlighting the cyclical nature of the four-step attack methodology.

During our exploitation phase, we expand our use of tools such as ZAP and the Burp Suite, and complement them with further use of sqlmap, BeEF, the Browser Exploitation Framework, and Metasploit to help craft exploits against various web applications. We launch SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery attacks, amongst others. In class we exploit these flaws to perform data theft, hijack sessions, steal passwords, obtain shells, pivot against connected networks, and much more. Through various forms of exploitation, the students gain a keen understanding of the potential business impact of these flaws to an organisation.

- **Day 6 and Day 7**

Day 6 and 7 consists of large practical exercises providing students with an opportunity to wield their newly developed or further honed skills to answer questions, complete missions, and exfiltrate data, applying skills gained throughout the course. The style of challenge and integrated hint system allows students of various skill levels to both enjoy a game environment and solidify the skills learned in class.

X. Basic Penetration Testing

Description

Special course developed by BI.ZONE specialists to share knowledge in finding and exploitation of vulnerabilities in different systems, teaching scanning, enumeration, exploitation techniques.

Prerequisites

Attendees are expected to have a working knowledge of TCP/IP, understand the differences between cryptographic routines such as DES, AES, and MD5, and have a basic knowledge of the Windows and Linux command lines before they come to class. Python and BurpSuite basic knowledge.

Topics and Technologies Covered

- Goals of reconnaissance
    - Active/passive reconnaissance
    - Reconnaissance tools
    - Search engines. Dorks
    - IPs and subnets
    - Domains and subdomains
    - Scanning methodology
    - Network technologies
    - OSI model
    - Common network protocols
    - Scanning tools
- Enumeration
    - LDAP
    - SNMP
    - NETBIOS
    - SMTP
    - NTP
    - DNS
- Exploitation frameworks
    - Public and private exploits
    - Shells • Privilege escalation
    - Persistence • Social engineering
    - MitM

- Known attacks on low-level protocols
    - DHCP
    - ARP
    - DNS
    - Traffic interception tools
    - IDS, IPS, Firewalls
    - Wireless networks attacks
- Client-Side Vulnerabilities
    - XSS
    - CSRF
    - Clickjacking
- Server-Side Vulnerabilities
    - SQL injection
    - XXE
    - OS command injections

Acquired skills and knowledge

Upon completion, the students will have the capacity to:

- Utilise the Nmap scanning tool to conduct comprehensive network sweeps, port scans, Operating System fingerprinting, and version scanning to develop a map of target environments along with using NMAP scripting engine
- Configure and launch the vulnerability scanner so that it discovers vulnerabilities through both authenticated and unauthenticated scans in a safe manner, and customize the output from such tools to represent the business risk to the organization
- Perform penetration testing using Netcat and the Scapy packet crafting tools
- Utilise the Windows and Linux command lines to extract valuable data
- Familiarise yourself with Metasploit exploitation tool to scan, exploit, and then pivot through a target environment in-depth
- Learn to perform password attacks including automated password guessing (while avoiding account lockout), traditional password cracking, rainbow table password cracking, and pass-the-hash attacks
- Manually exploit Cross-Site Request Forgery, Cross-Site Scripting, Command Injection, and SQL injection attacks to determine the business risks faced by an organization

Course layout

- **Day 1**

Day one starts with knowledge on how to create penetration testing infrastructure that includes all the hardware, software, network infrastructure, and tools you will need to conduct great penetration tests, with specific low-cost recommendations for your arsenal. There is also a significant recon portion of a penetration test, covering the latest tools and techniques, including hands-on document metadata analysis to pull sensitive data.

- **Day 2**

Day 2 is about scanning tools freely available today and run them in numerous hands-on labs to help hammer home the most effective way to use each tool. We will also conduct a deep dive into some of the most useful tools available to pen testers today for formulating packets: Scapy and Netcat.

- **Day 3**

Day 3 covers different exploits that penetration testers use to compromise target machines, including client-side exploits, service-side exploits, and local privilege escalation. Knowledge about frameworks like Metasploit and Meterpreter is extensively presented to audience. Anti-virus evasion techniques are trained, as well as methods for pivoting in infrastructure.

- **Day 4**

Day 4 covers post-exploitation, gathering information from compromised machines and pivoting to other systems in your scope. Including Windows command line skills in-depth, including PowerShell’s awesome abilities for post-exploitation. We will be looking at how we can leverage malicious services and the incredible WMIC toolset to access and pivot through a target organisation including mimikatz, msfconsile and Metasploit in-depth

- **Day 5**

John the Ripper/Cain tools used for password cracking on Day 5 along with Rainbow-Tables attacks and “pass-the-hash” attacks, leveraging Metasploit, the Meterpreter, and more. Day 5 finishes with web application pen testing, including methods like cross-site scripting (XSS), cross-site request forgery (XSRF), command injection, and SQL injection flaws in applications such as online banking, blog sites, and more.

- **Day 6 and Day 7**

Day 6 and 7 consists of large practical exercise in penetration testing. In the series of progressively harder tasks students will be recovering “flags” signifying sensitive data located inside apps or in surrounding infrastructure.

BI.ZONE Penetration testing team

Anton Prokhorov — Lead penetration testing specialist.

Anton has extensive experience in web and mobile application security analysis.projects
Qualifications: NRNU MEPhI, Automated Information Systems Security (05.10.03).
Work experience:

- BI.ZONE — Lead testing specialist.
- CryptoPro LLC — Engineering analyst.
- GlavNIVTs — Software developer.

Certification:

- Offensive Security Certified Professional (OSCP).
- GIAC Mobile Device Security Analyst (GMOB).

CTF-competitions:

- No cON Name CTF 2014 (BalalaikaCr3w) — 1st place.
- Volga CTF 2014 (BalalaikaCr3w) — 1st place.
- Positive Hack Days CTF 2014 (BalalaikaCr3w) — 3rd place.

Igor Motroni — Senior testing analyst

More than 5-years' experience in the field of offensive information security, taking part in projects on penetration testing, security analysis of systems of varying complexity, as well as advisory and consulting in the field of secure systems building.

Qualifications: NRNU MEPhI, Integrated Automated Information Systems Security, Department of Cryptology and Discrete Mathematics.

Work experience:

PWC (Jul 2016 - May 2017) – Consultant at Risk Assurance Services

Atlas НТЦ (Feb 2014 - Jun 2017) – Head information security specialist

Informzaschita (May 2017 - Jun 2019) – Lead security analysis expert

BI.ZONE (Jun 2019 - present) – Senior application security testing analyst

Certification:

OSCP

CEH

CRT

Pavel Zagumennov — Penetration testing specialist

Pavel has extensive experience working on the side of the customer as an engineer and information security specialist, including experience in implementing, operating and maintaining information security assets. He has competencies in external and internal penetration testing, management of information security assets, and the conduct of sociotechnical testing.

Qualifications: NRNU MEPhI, Automated Information Systems Security.

Work experience:

- BI.ZONE - Penetration tester
- Sberbank CIB - Security specialist, Senior Security specialist
- A-Security - Security engineer
- IBM RCIS - Network Engineering intern

Certification:

- OSCP
- CCNA R&S

Alexey Kuznetsov — Head of Application Security

- **Education:**

NRNU "MEPhI", Department of "Cryptology and Discrete Mathematics", specialty "Integrated support of information security of automated systems."

- **Experience:**

Alexey has more than 6 years of experience in the field of information security, 5 of which involved engaging in penetration testing projects. Before becoming head of the department, he worked as a testing specialist at BI.ZONE, a senior consultant at PwC and a developer of security tools at FSUE NTC Atlas and FSUE GlavNIVTS. Alexey has experience in designing and developing tools using hardware virtualisation technology, source code analysis in large industrial projects. His professional interests include the development of new approaches to the analysis of application security, the security of IoT devices, connected cars, smart city and others.

  - BI.ZONE — Head of Application Security;
  - PwC — Senior Consultant of Cybersecurity department;
  - FSUE NTC Atlas — Head Information Security Specialist;
  - FSUE GlavNIVTS — Software Engineer.
- **Additional Information:**
  - Offensive Security Certified Professional (OSCP).
  - CTF-competitions:
    - PHD2018 Met3rHacker (https://habr.com/company/pt/blog/414557/) — 1 place;
    - VolgaCTF2017 (https://ctftime.org/event/473) team "BI.ZONE" — 1 place;
    - PHD2017 Standoff (http://2017.phdays.com/standoff/leaders/) team "BI.ZONE" — 2 place;
    - PHD2016 (CAN4ALL) — 2 place;
    - CTFZone 2016 — 3 place;
    - Moscow CTF 2016 — 3 place;
    - Volga CTF 2015 — 5 place.

Vladislav Lazarev - Head of Penetration Testing

Experience:

- Conducted penetration testing
- Conducted web app security analysis
- Conducted corporate infrastructure security analysis
- Conducted security assessment of banking infrastructure
- BI.ZONE — Head of Penetration Tester;
- FSUE NTC Atlas — Head Information Security Specialist;

Certifications:

- (ISC)² Certified Information Systems Security Professional (CISSP)
- Offensive Security Certified Professional (OSCP)
- EC-Council Licensed Penetration Tester Master (LPT)
- SWIFT Security Bootcamp.

CTF-competitions:

- PHDays2018 HackBattle — 1 place.

- PHDays2017 Standoff — 2 place.
- VolgaCTF2017 — 1 place.
- Moscow CTF 2015 — 1 place.