SYNACKTIV

TRAININGS

2024.03

Presentation

Synacktiv is committed to sharing of its experience in cybersecurity, acquired over the years, by providing intercompany training. Combining theoretical teaching and practical work, our training courses have been designed to offer a unique and enriching learning experience, and are mainly aimed at **information security professionals**: pentesters, reverse-engineering experts, SOC analysts, CSIRT analysts, system administrators, security architects, developers, etc.

Each session is led by **two experienced trainers** who will ensure optimal understanding while providing concrete feedback. All the material necessary for the realization of the practical work will be provided to the students and each will have an individual environment in order to ensure an **immersive learning experience**. The course materials will be transmitted in PDF format, allowing participants to consult them at any time and use them as a reference.

The trainings take place in our Parisian offices, in a professional and comfortable environment which will promote the concentration of learners. **Lunches and drinks are included**, as well as a restaurant meal on the last day of training.

- 2 experienced trainers
- 7 to 12 participants
- Minimum 50% practice
- Practical work in individual labs
- Equipment provided (laptops)
- Course materials provided
- In our offices 5 bd Montmartre, Paris 75002
- Meals and drinks included

Pentest

Pentest Discovery

Obtain the skills needed to understand the main phases of an intrusion. Reconnaissance, web applications, Linux and Windows systems, post-exploitation steps, this training provides an essential base for any security professional.

5 days | Junior

Pentest Active Directory 1

Discover the fundamentals of security in Active Directory environments through this offensive training. From anonymous access to the complete compromise of infrastructures, become autonomous in intruding corporate networks.

5 days | Intermediate

Pentest Linux

Master intrusion techniques on Linux infrastructures through this offensive training. From anonymous access to complete compromise of the environment, become autonomous in intruding corporate networks.

5 days | Intermediate

Pentest Active Directory 2

Deepen your intrusion skills in Active Directory environments with this advanced level training. Learn advanced exploitation techniques and master the compromise of complex corporate networks.

5 days | Advanced

Pentest Cloud

Learn about modern network compromise with this cloud infrastructure training. GCP, AWS, Azure and Kubernetes, discover the characteristic mechanisms of these recent technologies, with the posture of an attacker.

5 days | Intermediate

Pentest Web Black Box

Learn about modern web application security mechanisms and advanced exploit methods to circumvent them. PHP, Java, Python and Perl, master the compromise of complex web applications.

5 days | Intermediate

Pentest

Pentest Web White Box

Obtain the skills needed to search for Java and PHP web vulnerabilities. Study of static and dynamic analysis frameworks and tools, this training allows pentesters and developers to optimize their search for vulnerabilities in white box.

5 days | Intermediate

Pentest Android Applications

Discover methodologies and techniques for analyzing Android applications. Architecture, entry points, static and dynamic analyses, master the pentesting methodology of the Android environment.

2 days | Junior

Password Cracking

Study methods for optimizing password cracking with John and Hashcat tools. Mutation rules, masks, prince and siga attacks, become a real password expert.

1 day | Junior

Reverse Engineering

Offensive Windows Development

Understand the basics of the Windows operating system in order to know how to implement, via low-level C APIs, offensive security mechanisms.

5 days | Intermediate

Offensive Linux Development

Understand the basics of the Linux operating system in order to know how to implement, via low-level C APIs, offensive security mechanisms.

5 days | Intermediate

Android for Security Engineers

Discover the internals of the Android operating system and its security mechanisms with the help of practical exercises.

5 days | Intermediate

iOS for Security Engineers

Discover the internals of the iOS operating system and its security mechanisms with the help of practical exercises.

5 days | Intermediate

IDA Advanced

Familiarize yourself with the advanced features of IDA, its API and its ecosystem. Learn how to develop scripts and plugins to extend its functionalities.

5 days | Intermediate

Hardware Intrusion

Learn to tame a PCB: recognize components, identify testpads, infer and then interact with protocols (UART, JTAG/SWD, SDIO, SPI). Use the active/passive tools and materials (analyzer logic, FT2232H, JTAGulator, OpenOCD).

5 days | Intermediate

Forensic

Windows Forensic

Master the digital investigation of Windows 10 and 11 systems by learning to identify and characterize the associated malicious actions, both in the context of a security incident and a search for compromise (removal of doubt, hunting).

5 days | Junior

Linux Forensic

Master the digital investigation of Linux systems by learning to identify and characterize the associated malicious actions, both in the context of a security incident and a search for compromise (removal of doubt, hunting).

5 days | Junior

Mobile Forensic

Discover the digital investigation of Android and iOS mobile operating systems by studying data acquisition techniques, the discovery of malicious applications and phone artifacts.

5 days | Junior

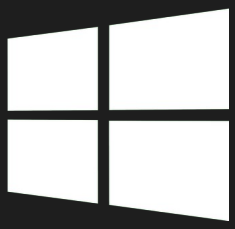
Windows Malware Analysis

Discover the analysis of malicious code in the context of a security incident through various situations and real cases of attackers' operating methods.

5 days | Intermediate

Pentest Discovery

5 days | Junior level



Description

Performing intrusion tests allows a realistic simulation of defense mechanisms and therefore represents a key step in securing information systems. This introductory pentest training aims to provide an in-depth understanding of security auditing by addressing the different stages of an intrusion.

During these five days of training, participants will be exposed to four course modules covering reconnaissance, web applications, Linux and Windows systems, and post-exploitation techniques. Each module will be illustrated by guided practical work to apply the theoretical notions taught. Finally, the training will conclude with a realistic scenario on a corporate network.

- 5 days (35 hours)
- 4 course modules covering the main steps of a penetration test
- Reconnaissance, web applications, Linux, Windows, post-exploitation
- 20 exercises
- 1 guided intrusion on a complete corporate environment (10 machines)

Audience and prerequisites

This training has been designed for people with no prior experience in penetration testing. It is mainly aimed at beginner pentesters, system administrators, security architects and developers, but also at any technical profile wishing to enrich their professional career with a security component.

- Beginner pentesters
- System administrators
- Security architects
- Developers

Basic knowledge of the Unix environment and web languages is recommended.

Content

Day 1

Introduction to discovery methods: DNS and HTTP enumeration, service scans. Overview of the main intrusion tools: Metasploit, Burp Suite. Web application vulnerabilities: SQL injections, XSS (Cross-Site Scripting), XXE (XML eXternal Entities), SSRF (Service-Side Request Forgery), file upload, deserialization, with various practice exercises.

Day 2

Practice on complex web applications: reconnaissance, exploitation and elevations of privileges until obtaining access to servers. Privilege escalation on Linux systems: fundamentals (identity and access management), reconnaissance and exploitation (permissions, sudo configurations, scheduled tasks, systemd units, kernel), and containerization technologies (Docker, LXC/LXD).

Day 3

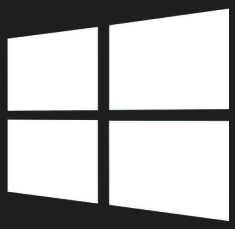
Privilege elevation on Windows systems: fundamentals (identity and access management, secrets management), reconnaissance and exploitation (permissions, service configurations, scheduled tasks, public vulnerabilities). Hands-on practice on servers from non-privileged access.

Days 4 and 5

Post-exploitation steps: extracting secrets (disk, memory), installing backdoors and lateral moves (network bounce, SOCKS proxy, port forward). Hands-on intrusion of a corporate network.

Pentest Active Directory 1

5 days | Intermediate level



Description

For many companies, Active Directory is the heart of identity and access management. Its ubiquity within information systems makes it a prime target for computer attacks, and penetration testing is a key component of its defense against threats.

During this five-day training, you will learn the skills necessary to perform an in-depth Active Directory penetration test. By following the five course modules, students will learn the methodology and techniques used by our experts during an intrusion, from anonymous access to the complete compromise of the environment and the persistence of access within it. To illustrate new concepts, learners will be guided through two comprehensive corporate environments.

- 5 days (35 hours)
- 5 course modules covering all intrusion steps + 1 Azure module
- 2 corporate environments with more than 40 machines and an Azure environment

Audience and prerequisites

This training is suitable for people with notions of offensive security but no prior experience in Active Directory environments. It is aimed primarily at pentesters, system administrators and security architects, but also at any technical profile wishing to enrich their professional career with a security component.

- Pentesters
- System administrators
- Security architects

Notions of offensive security and good network and Unix knowledge are recommended.

Content

Day 1

Theoretical foundations of security mechanisms: administration mechanisms (RPC, SMB, WMI, RDP, WinRM), identity and access management, storage of secrets, network authentication protocols (NTLM, Kerberos), hierarchy and Active Directory trusts. Reconnaissance and exploitation techniques from anonymous access: enumeration, network protocol poisoning, relaying.

Day 2

Reconnaissance on the domain from non-privileged access: objects extraction (users, groups, machines, GPOs) and mapping with BloodHound. Local privilege escalation: enumeration and exploitation (local services, scheduled tasks, ACLs, public vulnerabilities), UAC bypass techniques.

Day 3

Escalation of privileges within a domain: secrets extraction (registry, LSASS, DPAPI), replay of authentication, kerberoasting, abuse of control paths. Bypassing software restrictions: AppLocker, evading restricted desktops (Citrix, RDP Kiosk).

Day 4

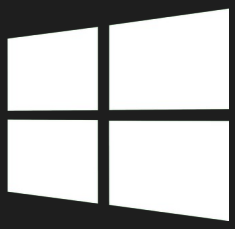
Post-exploitation steps from privileged access on the domain: secrets extraction (NTDS, DPAPI), ticket forgery (silver and golden tickets), manipulating ACLs, persisting within the environment, and erasing traces. Extending the compromise: cross-domain and cross-forest trust relationships, Kerberos delegation abuse.

Day 5

Introduction to Azure: fundamental concepts (terminology, identity and access management), integration with Active Directory (identity synchronization, Single Sign-On mechanisms), recognition and compromise steps from the on-premise environment.

Pentest Active Directory 2

5 days | Advanced level



Description

For many companies, Active Directory is the heart of identity and access management. Its ubiquity within information systems makes it a prime target for computer attacks, and penetration testing is a key component of its defense against threats.

During this five-day training, you will deepen your intrusion skills in an Active Directory environment, as well as on hybrid Azure environments. Guided by our experts, study advanced techniques of reconnaissance, lateral movements, elevation of privileges, extraction of secrets and persistence. To illustrate new concepts, the learners will be put in situation on two complete company environments.

- 5 days (35 hours)
- 5 course modules covering all intrusion steps
- 2 corporate environments with more than 40 machines and an Azure environment

Audience and prerequisites

This training is intended for people who already have a good knowledge of Active Directory environments. It is mainly intended for pentesters, system administrators and security architects.

- Pentesters
- System administrators
- Security architects

Good networking and Unix knowledge is also recommended.

Content

Day 1

Reminder of the fundamentals: Active Directory mechanisms, general and specific intrusion principles for these environments. Recognition and first actions from authenticated access: information retrieval methods (ADIDNS, service detection via LDAP and GPO scans) advanced use of BloodHound (Cypher queries).

Day 2

Lateral movements: ADIDNS, WinRM and JEA poisoning, LAPS, gMSA/sMSA secrets extraction, MS-SQL trust abuse, NTLM relaying (dissection, cross-protocol relaying, WebDAV), authentication coercing, Kerberos relaying, cross-forest pivots, pivoting to Azure (PHS, PTA, ADFS), pivoting from Azure (Intune).

Day 3

Local privilege elevation: access token and impersonation, study of potatoes vulnerabilities. Escalation of privileges on the domain: study and abuse of ACLs, advanced exploitation of Kerberos delegation, ADCS ESC1 to 11, abuse of privileged groups, analysis of public vulnerabilities.

Day 4

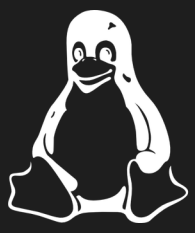
Secrets extraction: LSASS dump methods and tools, token spoofing, registry secrets analysis, DPAPI implementation, KeePass database.

Day 5

Persistence: ADCS (certificates), Kerberos tickets (golden, diamond, sapphire), DSRM, golden gMSA, AdminSDHolder abuse, skeleton key creation, Kerberos delegation, GPO poisoning.

Pentest Linux

5 days | Intermediate level



Description

Linux is a very widely used operating system, especially for servers but also for office workstations and embedded systems, such as network equipment. Managing a Linux infrastructure relies on administration mechanisms and methods that are essential for attackers to understand.

Throughout these five days of training, participants will be exposed to four course modules detailing the methodology of an intrusion from anonymous access to the compromise of the infrastructure, with a particular interest in the limitation of the footprint. An additional module will also be dedicated to hardened systems (AppArmor, SELinux). These notions will be applied throughout the week on two complex corporate networks, resulting from intrusions actually carried out by our experts.

- 5 days (35 hours)
- 4 course modules following realistic intrusion steps + 1 module on hardened systems
- 2 corporate environments with more than 30 machines

Audience and prerequisites

This training is suitable for people with notions of offensive security but no prior experience in the intrusion of corporate Linux environments. It is aimed primarily at pentesters, system administrators and security architects, but also at any technical profile wishing to enrich their professional career with a security component.

- Pentesters
- System administrators
- Security architects

Notions of offensive security and good network and Unix knowledge are recommended.

Content

Day 1

Fundamental concepts: identity and access management, security mechanisms (extended ACLs, standard and extended attributes, capabilities), containerization (namespaces, cgroups, seccomp, Docker and LXC/LXD implementations), administration methods. Reconnaissance and exploitation techniques from anonymous access: network mapping, name resolution protocols (mDNS / DNS), interceptions (ARP spoofing).

Day 2

Discovery from non-privileged access: system and network enumeration (services, sessions, configurations, LDAP, NFS / Samba shares), containerization detection. Local privilege escalation: advanced sudo configurations, scheduled tasks, capabilities, kernel exploitation (analysis of public vulnerabilities, adaptation of exploit code, implementation of protections).

Day 3

Post-exploitation steps: secrets extraction from disks, memory dissection and caching components abuses (SSH / GPG agents, DBUS Secret Service API), authentication poisoning (OpenSSH, PAM, sudo), lateral movements (network bounce, SOCKS proxy, port forwarding).

Day 4

Deep compromise: installation of advanced persistence mechanisms (userland and kernel rootkits), system footprint management (anti-forensic introduction).

Day 5

Compromising hardened systems: Implementing and configuring AppArmor and SELinux LSMs, analyzing and circumventing hardening.

Pentest Cloud

5 days | Intermediate level

Description

Cloud technologies are gradually being integrated into the information system of companies. They provide many security mechanisms that are sometimes difficult to understand and force attackers to rethink their methods of intrusion.

During this five-day course, participants will be exposed to the concepts of the three major cloud providers: GCP (Google), AWS (Amazon), and Azure (Microsoft). After having studied the fundamentals they share, their implementation specificities will be detailed and illustrated through complete environments allowing to learn about cloud intrusion techniques. An additional module will also be dedicated to Kubernetes infrastructures.

- 5 days (35 hours) customizable
- 3 course modules on GCP, AWS and Azure + 1 module dedicated to Kubernetes
- 4 complete and individual environments

Audience and prerequisites

This training is suitable for people with notions of offensive security but no prior experience in cloud environments. It is aimed primarily at pentesters, system administrators, security architects and developers, but also at any technical profile wishing to enrich their professional career with a security component.

- Pentesters
- System administrators
- Security architects
- Developers

Good network and Unix knowledge and notions of web intrusion are recommended.

Content

Day 1

Fundamentals: cloud terminology, infrastructure services, network topology, identity and access management, authentication mechanisms (OAuth), reminders of Linux security mechanisms (namespaces, cgroups, seccomp, LSM), OSINT.

Day 2

Google Cloud Platform: architecture (organization, folders, projects, resources, regions, and zones), IAM (permissions, roles, principals, and policies), authentication (OAuth 2.0, JWT), using the gcloud CLI, service discovery methods, abuse of rights on buckets, App Engine and instance implementations (metadata abuse), elevation of IAM privileges, network reconnaissance (VPC, firewall, VPN, peerings), post-exploitation (delegation on the domain, bounce on Workspace), analysis events.

Day 3

Amazon Web Services: architecture (organization, accounts), IAM (identity types, role assumption, policies), aws CLI usage, service discovery methods, unauthenticated identity enumeration, S3 bucket rights abuse, EC2 (metadata, lateral movements and poisoning of SSM agents), Lambdas (runtime API, persistence, data exfiltration), Cognito (user and identity pools) IAM privilege escalation, network reconnaissance (VPC, network ACL, security groups) , persistence (modification of IAM policies, role chain juggling).

Day 4

Azure: architecture (tenants, management groups, subscriptions), Azure AD (identity types, access management, Azure AD and RBAC roles), synchronization in hybrid environment (PHS, PTA, ADFS), unauthenticated discovery, use of azure CLI and Az module, authenticated discovery (ROADrecon, AzureHound), blob storage implementation, key vault, virtual machines, lateral movements (Vnet, bastions).

Day 5

Kubernetes: architecture (containers, pods, nodes, internal services), recognition, authentication (password, certificates, tokens) and authorizations (node, ABAC, RBAC, WebHook), kubectl CLI usage, pod templates and controllers, escapes (namespaces, PSP, PSA), network concepts (ingress, pod to pod, CNI, policies).

Pentest Web Black Box

5 days | Intermediate level

Description

Web applications represent a large part of the attack surface exposed on the Internet. As technology evolves, new vulnerabilities and exploitation methods continue to emerge, making the intrusion steps more complex.

During this five-day training, participants will study the functioning of the security mechanisms implemented in recent web applications. The various exercises resulting from the feedback of our experts will allow them to refine their intrusion methods for the exploitation of complex vulnerabilities. Finally, learners will be able to understand the specificities of Java, PHP, Python and ASP.NET languages and frameworks, using dedicated modules.

- 5 days (35 hours) customizable
- 9 course modules for Java, PHP, Python and ASP.NET
- Over 30 hands-on exercises

Audience and prerequisites

This training is suitable for people with prior experience in web intrusion techniques. It is mainly intended for pentesters and developers.

- Pentesters
- Developers

Good networking and Unix knowledge is also recommended.

Content

Day 1

BurpSuite: advanced usage, limitations, shortcuts and automation mechanisms, extensions (AuthMatrix, Hackvertor, ActiveScan++). Reconnaissance: DNS enumeration, vhosts, fuzzing, web component identification.

Day 2

Fundamental security mechanisms: authentication (OAuth, JWT, SAML), session management (cookies, tokens, viewstates), password reset, access control, user input management. Advanced exploitation: XXE, SSRF, injections, SSTI, prototype pollution, cryptographic attacks, GraphQL, specifics of cloud environments.

Day 3

Java: recognition and identification of frameworks (extensions, endpoints, headers, administrative interfaces), exploitation of specific vulnerabilities (XXE, HQL injections, deserialization, expression languages, JNDI, path traversals).

Day 4

PHP: recognition and identification of frameworks (endpoints, errors, headers), security functions (session management, sanitization), exploitation of specific vulnerabilities (type juggling, stream wrappers and filters, deserialization and design of complex POP chains, XXE), post-exploitation (fileless execution, disable_functions bypasses).

Day 5

Python Django: attack surface exposure (debug mode, cookie signing, DTL and Jinja2 template injection). ASP.NET: fundamentals, recognition, exploiting specific behaviors (deserialization, ViewState, Web.config, SSTI (Razor), XXE).

Pentest Web White Box

5 days | Intermediate level

Description

The complexity of modern web applications requires a strong understanding of the native mechanisms of the languages used. Source code analysis methods make it possible to optimize the search for vulnerabilities during an intrusion.

During this five-day course, you will acquire the skills necessary to identify complex vulnerabilities within the source code of Java and PHP applications. Based on many practical cases on popular frameworks such as Spring or Symfony, participants will learn how to optimize their research using static and dynamic analysis tools.

- 5 days (35 hours) customizable
- 7 course modules covering the specifics of Java and PHP
- Case studies on Spring, Struts, Hibernate, Zend, Symfony and Laravel frameworks

Audience and prerequisites

This training is suitable for people with good knowledge of web technologies and associated vulnerabilities. It is mainly intended for pentesters and developers wishing to improve their research method.

- Pentesters
- Developers

Good networking and Unix knowledge is recommended.

Content

Day 1

Methodology: top-down, bottom-up and hybrid approaches, static and dynamic analysis, tooling. Classic Java applications: structure of an application (Class components, JAR, JSP, configurations), formats (WAR, EAR), web.xml configuration (URI mapping, filters, hooks, security constraints), application of top-down and bottom-up approaches, tooling.

Day 2

Framework-based Java applications: identification, analysis of architectures and implementations of Spring (JavaBean, MVC pattern, SpEL, AOP, Security), Struts2 (actions, interceptors, views, OGNL, configuration, SMI/DMI, devMode) and Hibernate (definition models, configuration of connectors, ORM, HQL and SQL transformation), overview of other common frameworks (JavaServer Faces, VAADIN, SEAM, Play).

Day 3

Java instrumentation: Byteman, AspectJ and JDWP. Closed-source Java applications: methods and tooling for decompilation.

Day 4

Framework-based PHP applications: setting up the analysis environment (IDE, Xdebug, PHP configuration), analysis of architectures and implementations of Symfony (ORM, routing, constraints, authentication and access control), Zend (routing , authentication and access control) and Laravel (structure, configuration). POP chains: concepts, research and development.

Day 5

Closed-source PHP applications: mechanisms (scrambling, encryption), analysis of Blenc and IonCube implementations, use of protected code extraction and analysis tools (VLD, Xdebug, Dtrace, AOP, APD, RunKit).

Pentest Android Applications

2 days | Junior level

Description

Android is one of the most popular mobile operating systems on the market and on which many applications are developed. This ecosystem defines standards for implementation, communication, storage and security mechanisms that are specific to it and that developers must respect.

During this two-day training, participants will discover the specificities of implementing Android applications and will study the methodologies and techniques used to analyze them.

- 2 days (14 hours)
- 2 course modules
- 9 Android applications with hands-on exercises

Audience and prerequisites

This training is suitable for people with notions of offensive security but no prior experience in auditing Android applications. It is mainly aimed at pentesters and Android developers.

- Pentesters
- Android developers

Notions of offensive security and network and Unix knowledge are recommended.

Content

Day 1

Fundamentals: operation of an application and the Android ecosystem (services, intents, keystore, APK format, cache file, shared prefs, backup mechanism). Static analysis: analysis of permissions and interactions with the system and other applications, presentation of analysis tools and explanation of common artifacts giving information about the activities of an application.

Day 2

Dynamic analysis: architecture of an application at runtime, mechanism for interception and instrumentation of Java code, presentation of Frida and Objection to automate classic workarounds or obtain information. Practical cases: hands-on exercises on Android applications.

Password Cracking

1 days | Junior level

Description

Passwords still constitute an essential component of information system security today. During intrusions, different types of password hashes are recovered and being able to break them in a short time can prove decisive.

This training aims at presenting the techniques and tools for breaking password hashes as quickly as possible. A history of password storage developments will also be presented, to highlight bad examples and mistakes made in popular projects.

- 1 day (6 hours)
- Password cracking optimization techniques
- Datasets provided

Audience and prerequisites

This training is suitable for people having no prior knowledge of password cracking. It is mainly aimed at pentesters, system administrators, and developers.

- Pentesters
- System administrators
- Developers

Content

Password storage and generation theory: storage type, hash functions, function attacks, candidate generation, computational technologies. History of algorithms. Series of practical exercises: identification of algorithms in source code, getting started with John the Ripper (candidate generation modes, development of derivation rules and candidate filters based on a password policy, dynamic formats, implementation or modification of a native format), getting started with Hashcat (advanced candidate generation with prince combination, siga genetic mutations and rule generation).

Offensive Windows Development

5 days | Intermediate level

Description

Nowadays, AVs and EDRs aggressively scan created processes for intrusions, and Windows attempts to protect itself through a significant number of recently introduced countermeasures (AppContainer, ProtectedProcess, AMSI). This is why it is becoming more and more necessary for a pentester to be able to build personalized intrusion tools under Windows in order to go under the radar of security solutions during their red team engagements.

During this training, the students will learn to use low-level Windows APIs in order to perform stealthy operations considered hostile on the targeted system. They will also learn to use traditional system diagnostic tools such as an application debugger in order to resolve the problems inherent to development of intrusion tools. Finally, they will be exposed to the Windows security model and how the operating system is architected on the user-space side.

- 5 days (35 hours)
- 8h theoretical courses / 27h practical labs

Audience and prerequisites

This training is an intermediate level course designed for pentesters, Windows developers, and security teams.

- Pentesters
- Windows developers
- Security teams

Good knowledge of C development and a good understanding of the associated memory model is recommended.

Content

Day 1

Presentation of the work environment. Introduction to the PE format and diagnostic tools under Windows, basic use of a debugger (x64dbg and WinDBG).

Days 2 and 3

Visual Studio toolchain, native Windows development (win32), code injection, persistence and hooking.

Day 4

Practical exercises based on a RAT prototype, implementation of injection and persistence techniques.

Day 5

Presentation of the Windows security model (integrity levels, tokens, security descriptors, SIDs) and understanding of the associated limits.

Offensive Linux Development

5 days | Intermediate level

Description

This training aims at understanding the foundations of the Linux operating system in order to implement offensive security mechanisms through low-level C APIs.

After a first day of refresher on the basics of the Linux operating system, participants will learn to handle low-level APIs linked to processes (creation, communication, injection, debugging). They will also discover the ELF format and its representation in memory. Finally, security mechanisms (LSM), isolation (cgroup, namespaces) and system auditing mechanisms will also be discussed.

During this training, the participants will be required to implement a scenario in which an attacker will inject a library into the sshd service in order to intercept and then exfiltrate user passwords, while ensuring persistence on the system by installing a backdoor in a shared library.

- 5 days (35 hours)
- 13h theoretical courses / 22h practical labs

Audience and prerequisites

This training is an advanced level course designed for pentesters, Linux developers, and security teams.

- Pentesters
- Linux developers
- Security teams

Good knowledge of C development as well as a good general knowledge of security are recommended.

Content

Day 1

Linux OS basics: setup of the work environment, Linux distributions, shells, file system, security model, compilation chain, Systemd, D-Bus and PAM.

Day 2

ELF format, memory representation and hooking techniques.

Day 3

Processes, threads and injection: process creation, termination, monitoring, API, debugging and injection.

Day 4

Inter-Process Communication : IPC System V, signals, sockets.

Day 5

Security and isolation mechanisms: LSM (Apparmor & SELinux), Cgroups, Namespaces. User / kernel interface.

Android for Security Engineers

5 days | Intermediate level

Description

Android is one of the most popular mobile operating systems on the market. While originally based on Linux, it stands out with specific components making it unique and significantly different from the traditional OS. During this training, the participants will discover the architecture of Android and the interactions between its different internal components. The system allows third-party applications to run while protecting end-user data.

Key components of the system will be analyzed, including the boot process and security mechanisms. The trainers will detail the evolutions of the versions starting from Android 10 and will discuss certain particularities of the manufacturers. The concepts presented will be put into practice through concrete exercises.

At the end of this training, participants will have an in-depth understanding of Android and will be able to be autonomous in any research work on this ecosystem.

- 5 days (35 hours)
- 15h theoretical courses / 20h practical labs

Audience and prerequisites

This training is an advanced level course designed for security engineers wishing to conduct research on this system.

- Pentesters
- Android developers
- Security engineers

Good knowledge of C development as well as basic knowledge of Linux systems is recommended.

Content

Day 1

Overall architecture of Android, boot chain, update system, security model and rooting a smartphone.

Day 2

Format of applications (APK) and presentation of compilation and debugging tools (exercises with Frida).

Day 3

Android Runtime, IPC mechanism (Binder) and presentation of the Bionic library (Android libc).

Day 4

Application life cycle: installation, startup, execution and shutdown. Exploration of traces/logs that may be present on a device. Encryption of user data.

Day 5

Final exercise: modification of an Android environment via Magisk modules and putting into practice the concepts learned during the week. Analysis of the specifics of the Linux kernel for Android.

iOS for Security Engineers

5 days | Intermediate level

Description

iOS is one of the most popular operating systems on the market, offering a state-of-the-art security model. During this training, participants will discuss the ecosystem and the fundamental building blocks of the iOS operating system. They will discover how to use the macOS compilation chain to deploy a program, then debugging and diagnostic tools.

The fundamentals of reverse-engineering applications and system services will be covered in a second step: the internals of Objective-C, the IPC mechanisms (mach, XPC, NSXPC) and the kernel APIs. Practical examples and exercises will guide participants throughout the training. Finally, software and hardware security measures specific to iOS will be covered, both in the kernel and user space.

- 5 days (35 hours)
- 18h theoretical courses / 17h practical labs

Audience and prerequisites

iOS for Security Engineers is an intermediate level training course designed for security engineers wishing to carry out research on this system.

- Pentesters
- iOS developers
- Security engineer

Good knowledge of C development and basics in reverse engineering are recommended. An IDA Pro license with the Hex-Rays decompiler for ARM64 is a plus.

Content

Day 1

Introduction: presentation of the working environment, development on Apple platforms (iOS and macOS), use of diagnostic tools, introduction to the Apple ecosystem.

Day 2

Introduction to reverse engineering on Apple platforms: update extraction, important file formats and tools, discovery and experimentation with the inner workings of Objective-C, introduction to the XNU kernel.

Day 3

Mach mechanisms: explanations and exercises around the XNU IPC API, presentation and exercises on the implementation of the Mach API for interaction with kernel objects, use of Frida to instrument services.

Day 4

Reverse-engineering of Mach services: theory and practical exercises around XPC and NSXPC, the abstractions used for inter-process communications. Overview of the use of signed pointers on Apple platforms.

Day 5

XNU security: presentation of the MACF framework, explanations of how AMFI works and isolation policies (sandbox), description of XNU defense-in-depth mechanisms, hardware security countermeasures in the kernel, mitigations of kernel vulnerabilities. Case study on diagnostic data upload.

IDA Advanced

5 days | Intermediate level

Description

Hex-Rays is one of the major players in the development of reverse engineering tools. Their IDA product has established itself over the years as the benchmark in this area. However, the lack of documentation and resources sometimes makes it difficult to use.

The objective of this training is to familiarize yourself with IDA (its interface, its functionalities, its API and its ecosystem) through several theoretical and practical modules. Participants will also learn how to develop scripts and plugins to extend the functionality of IDA and its decompiler.

- 5 days (35 hours)
- 8h theoretical courses / 27h practical labs

Audience and prerequisites

This advanced level training is designed for security researchers and reverse engineering experts wishing to change environments or improve their use of IDA.

- Security researchers
- Reverse-engineering experts

Good knowledge of assembler (x86-x64, ARM) as well as Python programming is strongly recommended. An IDA Pro license (not supplied) is mandatory.

Content

Day 1

Introduction to IDA: terminology, architecture and presentation of the tool.
Understanding the Python SDK and API: basics.

Day 2

Getting started with the features available via different exercises. Static analysis: disassembler, FLIRT, IDS, Type Info Library. Dynamic analysis: debugger, tracer and binary instrumentation.

Day 3

Advanced programming (part 1): detailed presentation of the SDK and practice through scripting to automate complex tasks.

Day 4

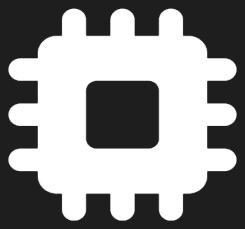
Advanced programming (part 2): development of plugins, loaders and processor extensions to practice on the previous notions.

Day 5

Extension of the decompiler: presentation of the Hex-Rays API, manipulation of microcode and AST, extension and improvement of the tools created during the session.

Hardware Intrusion

5 days | Intermediate level



Description

The objective of this training is to increase skills in hardware security analysis. It is aimed at both novices and those with an intermediate level.

At the end of this training, students must know the basic principles of electronics and soldering. They will be able to recognize the various components of a PCB and search for relevant information in component datasheets such as System on Chip (SoC) or external Flash to take advantage of them (RST implementation, debug functionality).

Finally, they will be able to identify possible test points, infer and then interact with the most common protocols (UART, JTAG/SWD, SDIO, SPI).

During the training, students will also learn to use equipment and tools useful for analysis (logic analyzers & Logic2, probes based on FT2232H & OpenOCD/flashrom)

- 5 days (35 hours)
- 17h theoretical courses / 18h practical labs

Audience and prerequisites

Hardware Intrusion Primer is a beginner to intermediate level training course designed for pentesters, security researchers and security teams.

- Pentesters
- Security researchers
- Security teams

Basic knowledge of electricity and electronics (how to use a multimeter, Ohm's law) is recommended.

Content

Day 1

Component Fundamentals: PCB, SoC, Flash, resistors, capacitors, transistors, crystal oscillators and PMIC.

Day 2

Theoretical reminders: electricity, security, analog and digital electronics.

Days 3 to 5

Common protocols: theory (characteristics, variation, usefulness in security analysis, signal shape) and practice (identify ports of interest, know how to use the hardware and tools to connect to them). Welding: principle, equipment and good practices.

Windows Forensic

5 days | Junior level

Description

Digital investigation makes it possible to reconstruct and understand in detail the chronology of a system's present and past activities. In the case of this training, we are interested in the Windows 10 and 11 operating system. Whether it is a security incident or a search for computer malware, the first responses aim to establish the perimeter of compromise and the attacker's methods. The technical approach presented is intended to be as exhaustive and reproducible as possible.

During these five days of training, the participants will be exposed to the forensic fundamentals in order to carry out a digital investigation for Windows and thus identify the traces of malicious intent. Each module will be illustrated by guided practical work allowing to apply the theoretical concepts previously taught. Finally, the training will conclude with a simulation of several traces (disk, memory, pcap).

This training is focused on the workstation and does not integrate the business dimension like Azure/AD (another training course will address this aspect soon).

- 5 days (35 hours)
- 11 course modules covering the fundamentals of Windows forensic investigation
- Cold or hot approach to cover several intervention situations
- Practical work on example artifacts

Audience and prerequisites

This training was designed for people with initial experience understanding Windows environments (administration, troubleshooting, advanced usage) and wishing to go further in the field of digital investigation. It requires basic knowledge of the Linux environment because this system is used to carry out some investigations.

- Advanced users (developers)
- System administrators
- Level 2 SOC analysts or from a cybersecurity team
- Beginner forensic analysts

Concepts of offensive security and good Windows & Unix knowledge are recommended to follow this training.

Content

Day 1

Getting started: training environment (virtual machine, Linux system). Reminders of the Linux command line. Windows: description of how Windows works (Windows history, processes, services, drivers, files, security model, network stack, main attacks). Windows events: description of the Windows logging model and the events to be aware of per use case. Scenario on event files.

Day 2

NTFS: study of the privileged file system of the Windows environment. MFT, USN Log and other special files. Decoding deleted dates and files. Reconstruct the chronology of events and pivot on an element (date, IOC). Registry: Registry contents. Use cases and configuration of the Windows system. Persistence Mechanisms: the means of persistence favored by an attacker are reviewed and thus identify the malicious programs executed by an attacker.

Day 3

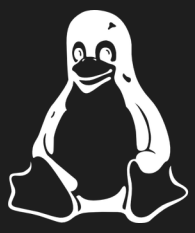
Execution of commands: traces linked to the execution of remote commands on the workstation through the different Windows protocols (WinRM, PsExec, WMI, RPC). Malicious code and files: analysis tools and methods allowing an initial study to be carried out on malicious code and thus extract the information of interest (behavior, IOC). By extension, files that can carry a malicious payload are also studied. Network protocols: particular attention is proposed in order to identify unusual network communications of a Windows system as well as the characterization of certain attacks (DNS tunnel, TOR).

Days 4 and 5

Artifacts: the study of most important forensic artifacts (prefetch, srum, amcache, navigation) in order to complete the timeline of the malware. Collection and acquisition methods are also presented in order to make the files available to be studied by the analyst (DFIR ORC). Memory analysis: techniques for acquiring and identifying suspicious elements are discussed to complement the analysis of offline elements. Running processes, network connections, cached files, memory injections and API hooking. Case study: several images are provided to the participants to put into practice all the techniques studied during the training. These images include various data such as a disk image, a memory capture and network captures.

Linux Forensic

5 days | Junior level



Description

Digital investigation makes it possible to reconstruct and understand in detail the chronology of a system's present and past activities. In this case, we are interested in the Linux kernel and two types of Linux distribution. While the examples and illustrations will focus apt and rpm-based distributions, most of the elements presented can be generalized to others.

During a security incident or a search for computer malware, the first questions deal with establishing the perimeter of compromise and the attacker's methods. The technical approach to such an investigation is intended to be as exhaustive as possible and, above all, reproducible.

During these five days of training, the participants will be exposed to the fundamentals in order to carry out a digital investigation for a Linux distribution and thus identify the traces of malicious intent. Each module will be illustrated by guided practical work allowing to apply the theoretical concepts previously taught. The training includes a role-play on several artifacts (disk, memory, pcap).

- 5 days (35 hours)
- 11 course modules covering the fundamentals of Linux forensic investigation
- Cold or hot approach to cover several intervention situations
- Practical work on example artifacts

Audience and prerequisites

This training was designed for people with initial experience understanding Linux environments (administration, troubleshooting, advanced usage) and wishing to go further in the field of digital investigation.

- Advanced users (developers)
- System administrators
- Level 2 SOC analysts or from a cybersecurity team
- Beginner forensic analysts

Concepts of offensive security and good Windows & Unix knowledge are recommended to follow this training.

Day 1

Getting started / the command line: training environment (virtual machine, Linux system). Reminder of the main commands for Linux. Linux and distribution: description of how Linux works including processes, file descriptors, security model (user/group, ACL, cgroup), named pipes, signals, terminal and command interpreter, X11. Filesystem: main filesystem types found in Linux systems (ext4, LVM, XFS). Specifics and special features for forensics: management of dates, deleted files, metadata, etc. Case of LUKS and virtual disks (qcow, vmdk).

Day 2

Boot sequence: identify the boot sequence in order to verify the integrity of the launch chain (grub, initramfs, UEFI case). Backdoor search on Systemd. Case of SecureBoot and kernel module signing. Program management: control of programs installed on the system (integrity, permissions). ELF format: program and library. Using apt and rpm package managers. Logging: type of logs (/var/log) and associated processes (syslog, auditd). Traces of compromise.

Day 3

Persistence mechanism: means of system and user persistence, device manager, Systemd. Process Analysis: process diagnostic tools, top Linux processes (ssh, X11), remote execution, procfs. Network Analysis: network configuration, network diagnostic tools, network socket, commonly encountered protocol and tunnel, network capture.

Day 4

Malicious code: analysis tools and methods allowing an initial study to be carried out on malicious code and thus extract the information of interest (behavior, IOC). Artifact: other artifacts (coredump, viminfo). Memory analysis: techniques for acquiring and identifying suspicious elements are discussed in order to complete the analysis of offline elements. Running processes, network connections, cached files, memory injections and API hooking.

Day 5

Container: Trace finding in containers and system reconstruction. Data collection: file extraction (disk copy) and selective (velociraptor). Case study: several images are provided to the participants to put into practice all the techniques studied during the training. These images include various data such as a disk image, a memory capture and network captures.

Mobile Forensic

5 days | Junior level

Description

The mobile phone has been evolving for several years as an extension of the workstation and is becoming a privileged target, because it is as close as possible to data. The digital investigation of this type of device aims to identify traces linked to criminal activities, to detect traces of malicious actions and compromise of the mobile phone.

This training aims at presenting the main artifacts present in the Android and iOS environments and to use an open source toolkit in order to analyze them. Adapted analysis methodologies will be presented in order to overcome the black box approach of certain systems and their pre-installed applications which complicate the audit of the phone.

This training exclusively addresses the case where the unlocking secrets of the phone are known.

- 4 days (28 hours)
- 2 mobile exploitation systems: Android & iOS

Audience and prerequisites

This training is suitable for people with knowledge of security or Linux system administration. It is mainly aimed at IT teams wishing to have first-level methods for investigating phones and who do not have software dedicated to this activity. More generally, anyone wishing to enrich their professional career with a security component in the mobile field.

- IT teams
- System administrators
- Security teams

Concepts of offensive security and good Unix knowledge are recommended to follow this training.

iPhone and Android phones are provided during the training for the hands-on exercises.

Day 1

Introduction: description of the mobile investigation ecosystem, its main services and players. Presentation of the main threats, infection vectors and the latest known campaigns. Fundamentals: description of the main sources of information linked to a mobile device (SIM card, Warrant Return), the specificities and problems of acquisition methods compared to classic forensic. Data formats used to store information and analysis methodology common to iOS and Android environments. iOS P1 fundamentals: representation of the architecture and main services.

Day 2

iOS P2 fundamentals: description of the file system and locations of interest, security model and its impacts. Acquisition methods and specific data formats. iOS System artifacts: review the activity of the entire phone looking for various traces of execution or presence of applications.

Day 3

iOS application artifacts: presentation of native applications and third-party applications (activity analysis, specific data). Analysis of encrypted backups: methods of acquisition and analysis in the absence of a complete copy of the phone. Other artifacts: Alternative sources of system information. New artifacts introduced in the latest versions of iOS. Live analysis: acquisition of live system data and network activities.

Day 4

Android fundamentals: overview of the architecture, main services and communication mechanisms. File system, locations of interest and security model. Acquisition methods specific to manufacturers. Android system artifacts: review of the activity of the entire phone looking for various traces of execution or presence of applications.

Day 5

Application artifacts: presentation of native applications (Android and manufacturer) and third-party applications (activity analysis, specific data). Live scan: scan with ADB in the absence of a full phone copy. Malicious APK analysis: static and dynamic analysis methodology and tools.

Windows Malware Analysis

5 days | Intermediate level

Description

When dealing with security incidents, it is common to discover malicious code. This training aims to provide the keys to understanding Windows malware and extract the elements of interest.

During the training, several types of malicious code are illustrated depending on the language used or the phase of the attack (exploitation, persistence). The different static and dynamic analysis methods are explained in order to provide complementary approaches to the analysis. A fairly significant part of the training deals hands-on exercises in the context of security incidents or operating procedures regularly observed in incidents. This course only addresses the case of userland malicious code.

- 5 days (35 hours)
- Malicious code analyzed on different languages
- Study of malicious files (Office, LNK)

Audience and prerequisites

This training is suitable for people who have already done programming under Windows or who have already undertaken program analysis (debugging or malicious code). It is aimed at all people involved in handling malware, particularly security teams (SOC, CSIRT) or wishing to improve their skills on this subject.

- SOC analysts
- CSIRT analysts

Good Windows knowledge is recommended to better understand how a malware works.

Content

Day 1

Qualification of a first level code: OSINT, automatic sandbox. Working environment: installation of an analysis environment (isolated/open) to process malicious code. PE structure: understand the format and aspects used by malware. Static and dynamic analysis of code: concepts and simple examples.

Day 2

x86(-64) assembler: first steps, control of execution flow and important instructions. Windows: Windows API, library to be known and used by malicious codes. Disassembler 101: getting started, case of decompilers. Debugger 101: getting started, step-by-step study & breakpoint.

Day 3

Initial intrusion: type of code used and exploitation. Malicious scripts: website analysis, Javascript deobfuscation. Analysis of malicious files: PDF, Office (OLE, VBA/XLM macros, pcode), VBScript deobfuscation, RTF. PowerShell code analysis: PowerShell code and shellcode emulation. Analysis of concealment techniques: LNK, ISO, HTA.

Day 4

Reverse engineering of complex code: go further on the types of code encountered, unpacking. Anti-reverse design methods: debug, sandbox, static reverse design. Automating analysis: scripting to automate reverse engineering of obfuscated code.

Day 5

.NET code reverse engineering: introduction to .NET and CIL, .NET malware analysis. Go code reverse engineering: introduction to Go, golang malware analysis. Modular code case study.

SYNACKTIV