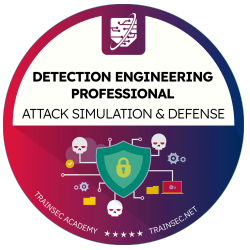
trainsec
where pros train pros.



Detection Engineering Professional: Attack Simulation & Defense

About TrainSec Academy

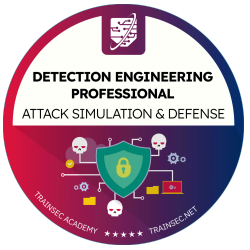
TrainSec is an online learning hub for current and aspiring cybersecurity pros who understand the need for excellence. Level up or change career using skill-expanding modular learning way. We turn beginners into experts and experts into masters. TrainSec is designed for those who are serious about being at the vanguard of the latest knowledge, skills and trends. We're already working with the best.

Course Overview

This is a hands-on detection engineering course built for practitioners who already work in security. Instead of teaching you to recognize an attack, it teaches you to build the detection that catches it. Through guided attack simulations, log analysis and detection engineering, you will develop the skills to simulate real attacker techniques, investigate them across enterprise telemetry, and respond to them exactly as you would on the job, whether that job is red team, blue team, or a general security role. Every technique is first executed against a Windows Active Directory lab, then followed through the telemetry it leaves behind in Windows Event Logs, Sysmon and your EDR, and finally closed out with a custom detection rule that you write, tune and validate yourself. At the end of the course you will not only recognize how modern intrusions unfold, you will own the detections that catch them.

Course Objectives

This course turns security theory into working detections. You will build a full detection lab, simulate the attacker techniques that matter most in enterprise Windows environments, investigate them across every relevant telemetry source, and finish each cycle by authoring, tuning and validating your own detection rules, the exact loop a detection engineer runs in production. Gain job-ready skills, earn certification, and move confidently into detection engineering, one of the highest-demand specializations in security today.



The Course Methodology

Every attack and defense module in this course follows the same repeatable cycle. It is the single most important habit the course teaches, because it is how real detection engineering work is done:

1. **Simulate** - execute the attacker technique yourself in a controlled lab so you know exactly what happened and when.
2. **Investigate** - follow the technique through the telemetry it produced, and discover which data sources saw it and which were blind.
3. **Detect** - write the detection rule against the durable artifact the attack leaves behind rather than against the tool that created it.
4. **Tune** - measure the alert volume, exclude the legitimate activity, and accept that every exclusion you add is a new blind spot.
5. **Validate** - re-simulate the attack to prove the rule fires, then ask how you would bypass your own rule and close that gap.

Course Format

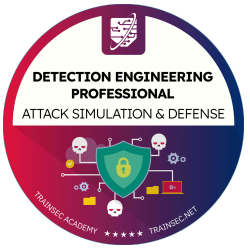
- **URL**: https://trainsec.net/courses/detection-engineering-attack-simulation-defense/
- **Scope and Mode**: 148 Learning materials, spanning on around 14 hours and 37 minutes of online videos, live demonstrations and hands-on detection labs.
- **Format**: Pre-recorded Videos and Presentations, discussion group and personal trainer support.
- **Pace**: Self-paced
- **Platform**: TrainSec.net Learning Platform, Discord server.

Course Content

Module 1: Welcome & Introduction

- What a detection engineer, does day to day across SIEM, EDR, ticketing and threat intelligence feeds
- How a SIEM correlates cloud, Windows Event Log, Sysmon, syslog and NetFlow sources into a single attack picture
- Reading a correlation rule, and how repeated failed logons become a brute force alert
- The course methodology: simulate a TTP, collect telemetry, investigate, tune false positives, write the rule
- Why detection logic, once proven, can be promoted into prevention
- The full course map, and why the Windows Active Directory enterprise is the primary focus

Video Content:



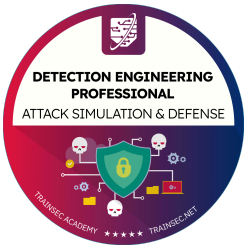
- Welcome
- What you will find here
- Blue Team / Purple Team Fundamentals
- The Essence of this Course

Module 2: The Attack & Defense Mindset

- What a hacker really is, and telling black hat, white hat, gray hat and script kiddie apart by skill, intent and authorization
- The core offensive vocabulary: vulnerability, exploit, payload, zero-day versus one-day, buffer overflow, remote code execution
- The three reconnaissance pillars - email harvesting, credential stuffing from breach dumps, and IP, port and service mapping
- Why banner grabbing and service version disclosure matter to a defender, and how suppressing them raises the attacker's cost
- Applying the CIA triad as a design and triage lens, including the practical confidentiality versus availability trade-off
- Using the DAD triad - disclosure, alteration, denial - to describe attacker objectives as the inverse of your own goals
- Walking an intrusion through all seven stages of the Cyber Kill Chain
- Why the installation stage decomposes into persistence and privilege escalation, and why command and control is the hinge of an intrusion
- The threat actor taxonomy: ransomware gangs, hacktivists, terrorists, thrill seekers, nation-state actors and insiders
- Defensive lessons from real incidents including Kaseya VSA, Colonial Pipeline, the Lockheed Martin F-35 data theft and WannaCry
- How VPNs, proxy servers, proxy chaining and Tor are repurposed to hide attacker infrastructure, and where attribution is still possible
- What the darknet is, how it is reached, and the role of criminal forums and marketplaces in enabling attacks

Video Content:

- Introduction
- Who is a Hacker
- Basic Concepts - Part 1,2
- The CIA Triad
- The DAD Triad
- The Cyber Kill-Chain - Part 1,2
- Threat Actor Types
- Infamous Cyber Attacks
- Attacker's Goals & Desires
- How attackers hide their traces
- The Darknet



Module 3: The Attacker's Martial Art

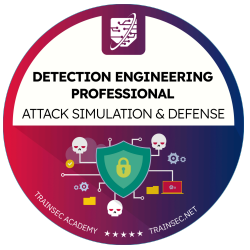
- Passive versus active reconnaissance, and which of the two your security controls can actually observe
- Auditing what your own organisation exposes publicly - staff names, email address formats, domain registration data, internet-reachable devices
- How ordinary search engines become an exposure discovery tool, and auditing your estate for the same indexed artifacts first
- The network footprint that host and service enumeration leaves in firewall, IDS, flow and endpoint telemetry
- Reading TCP handshake behaviour to tell an open service, a closed port and a filtered port apart in packet captures and firewall logs
- Why signature-based and behaviour-based endpoint controls can be circumvented, and why a clean scan result is not a clean system
- The case for defense in depth, layered telemetry and independent validation of your own detection coverage
- Identifying credential-harvesting phishing infrastructure from domain, hosting, certificate and mail gateway indicators rather than page appearance
- How adversary-in-the-middle positioning works, what it does to protocol downgrade risk, and which modern controls neutralise it
- Tracing post-compromise objectives to the specific endpoint artifacts each one creates
- Recognising remote authentication brute forcing in logs, and setting defensible alerting thresholds and lockout policies
- Prioritising remediation on internet-facing services, using the reality that decade-old vulnerable versions remain exposed at scale

Video Content:

- Reconnaissance
- Google Hacking & Dorks
- Port Mapping with Nmap
- AV-EDR Bypass - Part 1,2
- Social Engineering Phishing Web Page
- SSL Stripping - Part 1,2
- Infrastructure Exploitation
- Post Exploitation - Part 1,2
- Remote Brute Force

Module 4: Introduction to Malware Attacks

- Classifying malware by category - virus, worm, trojan, rootkit, backdoor, spyware, infostealer, ransomware - and what each implies for your response



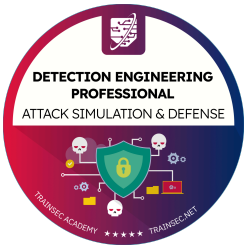
- Dropping versus downloading, and why that distinction changes which telemetry you examine
- Choosing the right analysis method: automated sandbox, basic static, dynamic, or full reverse engineering
- Why sandbox verdicts are not proof, and how anti-sandbox and anti-VM checks produce empty or misleading reports
- Reading the MITRE ATT&CK matrix properly - tactics, techniques, sub-techniques and procedures, plus mitigations and detection guidance
- What LOLBins, LOLBAS and LOLScripts are, and why signed built-in binaries are the attacker's preferred execution path
- Turning an ATT&CK technique into an executable test with Atomic Red Team, including Invoke-AtomicTest and the cleanup commands
- Deciding realistically what your EDR can and cannot detect, including API-level process injection that most vendors never expose
- Reconstructing the Emotet infection chain stage by stage, from the macro-weaponized document to the base64 PowerShell stage
- How the WMI Win32_Process Create method breaks the winword.exe to powershell.exe parent-child chain and defeats rules built on it
- Which telemetry classes you must have in place before any detection can be written
- Contributing techniques back to ATT&CK, and why unlisted techniques are your blind spots

Video Content:

- Intro
- Types of Malware
- Malware Analysis
- The ATT&CK MITRE Framework
- Atomic Red Team
- The Emotet Usecase - Part 1,2

Module 5: Proactive Defense - Hardening & Mitigation

- Separating patch, update and upgrade correctly, and putting AAA - authentication, authorization, accounting - into daily vocabulary
- The principle of least privilege as a two-sided constraint: too little breaks the business, too much gets weaponized
- The CPU ring model, and why local admin is usually enough for an attacker without ever touching the kernel
- The most common real-world misconfiguration - SMB shares granting Everyone Full Control or Write - and what it lets an unprivileged user do
- Designing segregation at every layer: users and groups, file system permissions, per-department network IDs, VLANs and a DMZ



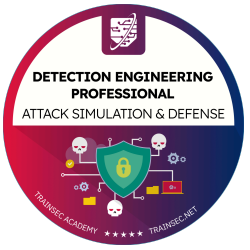
- Building a hardening baseline: mapping assets and zones, patching fast, shutting unused switch interfaces, removing stale accounts, validating firewall rulebases
- Executing and then mitigating VLAN hopping by double tagging and switch spoofing, using the exact Cisco commands
- DHCP spoofing, DHCP starvation and ARP poisoning, and shutting them down with DHCP snooping, Dynamic ARP Inspection and IP Source Guard
- Running a Slowloris application-layer denial of service against a web server and applying realistic mitigations
- Performing a DNS zone transfer with nslookup, then restricting transfers to the secondary server only
- Credential dumping from LSASS, the difference between LSASS and the SAM, pass-the-hash and pass-the-ticket, and stopping it with Credential Guard
- Concrete Windows and Linux hardening - LAPS, delegation, GPO restrictions, SELinux, sudo, iptables and file permissions
- Standing up the monitoring layer: Windows and Linux logs, syslog, SNMP, NetFlow and IDS/IPS, all orchestrated into a SIEM

Video Content:

- Intro
- Basic Concepts
- The Principle of Least Privilege - Part 1,2
- General Security Best Practices
- VLAN Hopping
- More Attack Techniques
- Network Attacks Mitigation Practices
- DDoS Attacks & Mitigations
- Brute Force & Physical Access Attacks
- DNS Zone Transfer Attack & Mitigations
- Credential Dumping & Mitigations
- OS & Network Security

Module 6: Proactive Defense - Threat Hunting & Incident Response

- Reactive incident response versus proactive threat hunting, and knowing which one you are doing on any given shift
- Building a preparation baseline: backups and snapshots, VSS shadow copies, offline versus online trade-offs, and log retention
- Justifying a data retention policy, and what a responder simply cannot reconstruct without it



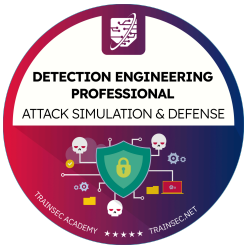
- The endpoint and network telemetry detection engineer and SOC centers needs for full visibility: Windows Event Logs, Sysmon, syslog, NetFlow, ETW and EDR telemetry
- Early identification signals worth escalating - unexplained CPU load, repeated crashes across hosts, new privileged accounts, services, scheduled tasks and startup items
- Auditing your EDR for lack of telemetry versus insufficient telemetry, using WMI activity events as the worked example
- Containing by segment rather than by pulling cables: firewall rules, VLANs, routing changes and targeted disconnection
- Acquiring volatile evidence in the correct order - memory first, then disk - and knowing exactly what a shutdown destroys
- Proving a victim-to-C2 connection from a memory image using the netstat table and Volatility netscan
- Eradicating and recovering safely: verifying backups are clean, scanning with a second engine, and never trusting a prevented verdict without validation
- Running the aftermath phase properly: lessons learned, timeline optimization, measured handling time per incident type, and tool updates
- Writing a hypothesis-driven hunt across all seven steps - scope, telemetry, hypothesis, hunt, validate, detect, improve
- Preferring indicators of attack and TTPs over static IOCs, and why polymorphic malware makes hashes worthless
- Promoting a validated detection from alert-only into prevention or block mode after a proper tuning period

Video Content:

- IR Intro - Part 1,2
- Preparation
- Identification & Analysis - Part 1,2
- Incident Containment - Part 1,2
- Incident Eradication & Recovery
- The Aftermath - What happens after the Incident
- Threat Hunting - Part 1,2,3

Module 7: Working with Windows Event Logs

- Locating the raw event logs in C:\Windows\System32\winevt\Logs, and why Security.evtx is the most valuable channel for DFIR
- The Security, System and Application channels, and what class of attacker activity lands in each
- Driving Event Viewer as an analyst tool: custom views, saved logs, Find, filtering by ID lists and ranges, and Friendly versus XML view



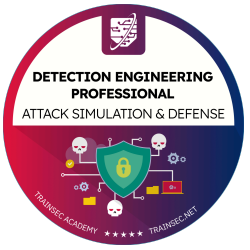
- Reading Event ID 4624 and 4625 field by field - Logon Type, Account Name, Failure Reason, Network Information and Process Information
- Using Logon Type as the primary context field: type 2 interactive, type 3 network and SMB, type 10 RDP, and service logons running as SYSTEM
- Recognising a password spray or brute force from a burst of 4625 events followed by a single 4624, and what that sequence proves
- Why a domain controller's Security log holds the authentication record for the entire domain, and why LSASS sits behind those events
- Simulating failed and successful network logons with net use, then locating and analyzing both events on the domain controller
- The impact of an attacker who can map a DC's C$ share, including exposure of C:\Windows\NTDS\ntds.dit
- Interpreting a NULL SID in a logon event as an indicator of a local rather than a domain account
- Detecting anti-forensic log clearing, and why an unexplained log-clear event is itself a high-confidence incident indicator
- Performing attacker-style domain discovery with whoami /priv, net user /domain and nltest, and confirming first-hand how little it leaves behind
- The concept of insufficient versus absent telemetry, and the remediation options: EDR, Sysmon, audit policy, registry and GPO settings
- Correlating Windows Security events with Sysmon Event ID 1 to prove which binary produced an observed authentication

Video Content:

- Intro & Log Sources
- Exploring Windows Event Viewer - Part 1,2
- Common Windows Event IDs - Part 1,2
- Analyzing Event ID 4625 Failed Logon - Part 1,2
- Analyzing Event ID 4624 Successful Logon
- Domain Discovery & Lack of Telemetry Data - Part 1,2

Module 8: Sysmon Detection Engineering

- Evaluating an EDR against a concrete checklist: cost, performance, support, out-of-the-box coverage, custom rule authoring, and which telemetry classes are exposed at all
- What Sysmon is - a Windows service plus driver writing deep telemetry to the event log - and what it is not: it detects, but never prevents
- Installing, configuring and validating Sysmon, including hot-reloading a configuration and confirming the service start type
- Navigating the Microsoft-Windows-Sysmon/Operational channel, and why it does not live under Windows Logs



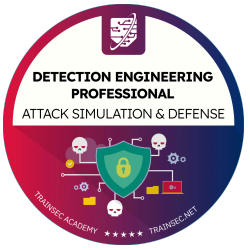
- The core Sysmon event IDs, and for each one the fields that carry real detection value
- Building a Sysmon configuration from scratch: schemaversion, EventFiltering, RuleGroup and groupRelation, onmatch include and exclude blocks, and field conditions
- Choosing the right match condition - is, contains, contains any, contains all, begins with, ends with, excludes - and why the wrong one silently kills a rule
- Writing ProcessCreate rules for nltest.exe, net.exe and net1.exe, then extending them with CommandLine conditions targeting specific arguments
- Writing a DnsQuery rule keyed on QueryName rather than the calling image, and understanding where it still misses
- Writing a FileCreate rule that catches any drop into any user's Startup folder
- Writing a RegistryEvent rule for Run key persistence, and interpreting Event ID 12 versus Event ID 13
- Writing an ImageLoad rule for vaultcli.dll, researching the legitimate loaders and excluding them with a contains any list
- Writing a CreateRemoteThread rule that fires on everything, then adding SourceImage and TargetImage exclusions as false positives appear
- Troubleshooting configuration failures from the command-line output: schema versions, unclosed tags, reported line numbers and duplicate event tags
- Reasoning about your own bypasses - why parent-image, single-name and file-extension rules fail, and why renames must be covered

Video Content:

- Sysmon Intro - Part 1,2
- Core Sysmon Event IDs & Blind Spots
- Deploying Sysmon
- Detecting Nltest using ProcessCreate Event - Part 1,2
- Detecting the Net command using ProcessCreate Event - Part 1,2
- Detecting DNS Queries - Part 1,2
- Detecting Persistence using the FileCreate event - Part 1,2
- Detecting Persistence using RegistryEvent - Part 1,2
- Detecting Credential Access using ImageLoad - Part 1,2
- Sysmon Events Exclusion
- Detecting the CreateRemoteThread Process Injection

Module 9: EDR/XDR Detection Fundamentals & Deployment

- Distinguishing AV/EPP, EDR, XDR and SIEM by the telemetry each collects and the response actions each enables, rather than by vendor marketing
- Walking the full pipeline from attacker action to sensor observation, telemetry ingestion, detection logic, correlation and incident decision



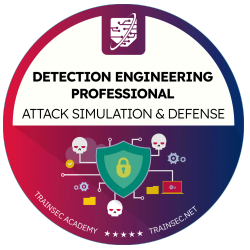
- Reading a process tree as a parent-child chain with command lines, user and timing, and telling a benign chain from an Emotet-style chain
- Correlating independent low-value events - RDP logon, discovery commands, LSASS access, archive creation, outbound connection - into one verdict
- Applying the behaviour plus context plus time window model instead of alerting on the presence of a single binary such as PowerShell
- Judging severity and triage order from device role and from the rarity of the observed behaviour in your estate
- Running the hunt, detect, investigate, scope, respond and improve loop, and feeding findings back into better rules and playbooks
- Deploying the Elastic Stack in the cloud and enrolling a Windows endpoint through Fleet with the Elastic Defend integration
- Optimizing an Elastic Defend policy: malware, ransomware, memory threat and malicious behaviour protection in Prevent mode, plus the full Windows event collection set
- Enabling Prevent Agent Tampering, and why the uninstall token is a high-value secret an attacker will hunt for
- Performing a telemetry sanity check across process, file, registry, network, DLL load, event log and Windows API sources before writing a single rule

Video Content:

- Intro
- AV vs. EDR vs. XDR vs. SIEM
- Detection Kill-Chain & Process Tree
- Detection Correlation
- Incidents & Alerts Chain of Events
- Elastic Security EDR Overview
- Deploying Elastic Defend - Part 1,2
- Optimizing the Security Policy and Enabling Anti-Tampering
- EDR Telemetry Collection Sanity Check

Module 10: Attack and Defense - Initial Access and Lateral Movement

- Building a detection from behaviour rather than tool names, because process and file names can be changed by the operator in seconds
- Enumerating a domain from a foothold workstation with hostname, whoami /priv, whoami /all, ipconfig /all and nltest to locate the domain controller
- Simulating an RDP intrusion end to end: enabling RDP, verifying the 3389 listener, and authenticating with domain versus local credentials
- Reading event 4624 and interpreting logon types 10, 7 and 3, and why an RDP session does not always produce logon type 10



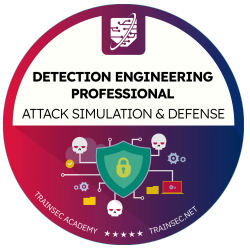
- Correlating Windows Security events with Elastic Defend endpoint telemetry to place a source IP address and a user account behind a remote logon
- Translating raw Windows event IDs into Elastic normalized fields: event.code, event.outcome, winlog.event_data.LogonType, host.name and process.name
- The full PsExec execution chain - ADMIN$ share check, named pipe, PSEXESVC.exe drop, service creation and service execution
- Why file events carry no network fields, so a source IP has to come from a correlated authentication event
- Recognising wsmprovhost.exe as the WinRM host process, and why PowerShell over WinRM never spawns a visible powershell.exe on the target
- Discovering a secondary WinRM indicator by DLL load, and validating the uniqueness of an indicator before building a rule on it
- Authoring KQL detection rules in Elastic Security: data view selection, query, severity, MITRE tagging, schedule interval and response actions
- Choosing between rule types - KQL, threshold, event correlation (EQL) and machine learning - and knowing when a threshold rule is the right fit
- Tuning and staging rules detect-only first, excluding false positives, and only then considering prevention such as host isolation or process kill
- Applying protocol hardening: Network Level Authentication and restricted Remote Desktop Users via GPO, and WinRM TrustedHosts allowlisting

Video Content:

- Attack & Defense Intro
- RDP Attack Simulation
- RDP Attack Investigation
- RDP Attack Detection - Part 1,2,3
- PSexec Attack Simulation
- PSexec Attack Investigation
- PSexec Attack Detection
- WinRM Attack Simulation
- WinRM Attack Investigation
- WinRM Attack Detection

Module 11: Attack and Defense - Discovery

- Mapping Active Directory domain trusts from a compromised host with nltest
- Enumerating domain controllers three different ways - dclist, dsgetdc and dnsgetdc
- Why nltest is high-value to block or allowlist rather than merely alert on
- Separating the network (DNS) event from the process event that a single command generates



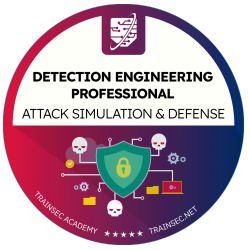
- Reading the Elastic fields that matter: process.command_line, user.name, host.name and destination.port
- Enumerating local versus domain accounts with net user and net user /domain
- Recognising the krbtgt account and why attackers target it for golden tickets
- Detecting both net.exe and net1.exe, and catching the /do /dom /domain switch family
- Writing case-insensitive, wildcard-based Elastic rules and escaping special characters correctly
- Avoiding rules over-fitted to a single parent process that an attacker can simply swap out
- Creating Elastic detection rules with sensible severity, interval and rule exceptions
- Baselining built-in binaries against historical usage to set alert severity

Video Content:

- Nltest Discovery Simulation
- Nltest Discovery Investigation
- Nltest Discovery Detection
- Net Discovery Simulation
- Net Discovery Investigation
- Net Discovery Detection

Module 12: Attack and Defense - Living-off-the-Land (LOLBin & Friends)

- Recognising encoded PowerShell execution - the -e and -enc pattern - as a high-signal indicator of malware such as Emotet
- Deobfuscating a base64-encoded command safely with CyberChef to reconstruct what an attacker's script actually did
- Validating vendor built-in detections instead of trusting them blindly, and writing your own rules where coverage is missing
- How WMI process creation breaks the parent/child process chain, and why that blinds many EDRs
- Pivoting on parent process lineage using WmiPrvSE.exe as the tell that a process was spawned via WMI
- Mapping Elastic field names to their equivalents in other products so your detection logic ports across tools
- Building a behavioural rule keyed on process ancestry rather than a single child-process string, and testing it against variations
- Proving why a rule scoped to one binary is bypassed the moment the attacker swaps in another, then generalizing it properly



- Reasoning about detection gaps caused by missing telemetry rather than by weak rule logic
- Knowing when to supplement built-in EDR telemetry with Sysmon, and why native coverage is preferable when it exists
- Treating rule robustness, coverage testing and gap analysis as an ongoing engineering discipline rather than a one-time task

Video Content:

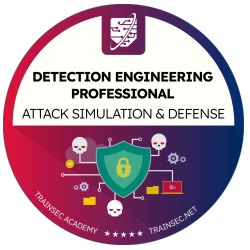
- The Emotet Usecase - Detecting PowerShell Attacks
- WMI Attack Simulation
- WMI Attack Investigation
- WMI Attack Detection
- KeyMgr Detection Bypass Red Team Edition - Part 1,2

Module 13: Attack and Defense - Persistence & Credential Access

- Planting Registry Run key persistence with reg add, and why reg.exe is a standalone binary rather than a shell built-in
- The full set of autostart registry locations - Run, RunOnce and RunOnceEx - across both the HKCU and HKLM hives
- Writing Elastic queries against registry.path with wildcards and escaped backslashes so one rule catches every hive and key
- Detecting suspicious payload strings inside registry.data.strings rather than relying on a process name
- Creating a Windows service for persistence with sc create, and locating its definition under HKLM\SYSTEM\CurrentControlSet\Services
- Detecting service persistence at the registry level so the rule fires whether sc.exe or the CreateService API was used
- Planting Startup folder persistence with a .lnk shortcut, and recognising that its payload runs as a child of explorer.exe
- Building a file-based detection on file.path and file.extension covering executable file types dropped into Startup
- Converting each saved query into a scheduled Elastic detection rule, and validating it by re-simulating the attack and watching alerts fire
- Dumping LSASS with the comsvcs.dll MiniDump technique and reading the rundll32 command line back out of EDR telemetry
- Testing detection of LOLBAS credential-dumping tools and gauging EDR resilience against rename, path and hash changes
- Separating legitimate autostart and service entries from attacker persistence, and why built-in rules must be supplemented with your own

Video Content:

- Registry Persistence Attack Simulation



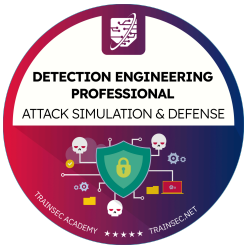
- Registry Persistence Attack Investigation
- Registry Persistence Attack Detection
- Service Persistence Attack Simulation
- Service Persistence Attack Investigation
- Service Persistence Attack Detection
- Startup Folder Persistence Attack Simulation
- Startup Folder Persistence Attack Investigation
- Startup Folder Persistence Attack Detection
- Credential Access Techniques

Module 14: Attack and Defense - AI & Process Injection

- The telemetry footprint of the main process injection families, including CreateRemoteThread-style injection, thread execution hijacking, APC injection and process hollowing
- Separating detection coverage from prevention coverage in your EDR, and proving which one you have by simulation rather than by datasheet
- Identifying the failure mode where an EDR reports a blocked attack while the injected process survives, and what that means for triage
- Querying Elastic Defend's Windows API telemetry with process.Ext.api.name and process.Ext.api.behaviors instead of relying on process names
- Building a hunting query around API-level signals such as VirtualAlloc, VirtualAllocEx and WriteProcessMemory, and widening or narrowing it deliberately
- Pivoting from an injector process to its injected target and reconstructing the cross-process relationship from events
- Using the Elastic Security AI assistant with skills and a chosen model to draft detection rule ideas and translate queries between ES|QL and KQL
- Structuring a prompt that carries real context: the environment, the observed alert, the tool behaviour and the exact question you want answered
- Validating every AI answer against the product itself, including checking the advanced policy setting it cites before you believe the explanation
- Reviewing an AI-generated rule field by field, deleting conditions you do not trust, and keeping the one condition that carries the detection
- Creating, scheduling and enabling a custom detection rule manually when the assistant cannot complete the action, and attaching a response action
- Re-simulating the attack after enabling your rule to confirm the detection fires and to discover whether the response action really executes

Video Content:

- Process Injection Techniques Attack Simulation
- Process Injection Techniques Attack Investigation & Detection
- AI Assisted Detection Engineering - Part 1,2,3



Module 15: To Know Your Enemy, You Must Become Your Enemy - Final Notes & Labs

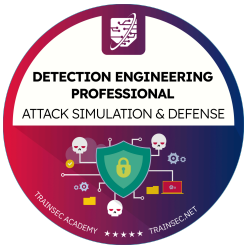
- Running a repeatable alert triage workflow: confirm the entity, validate the activity, expand the timeline, decide the verdict, then document and improve
- Distinguishing a true positive from a false positive and from a benign true positive - the case that matters most in real environments
- Tuning detections with allow-lists scoped by account, device and parent process rather than by tool name alone
- Recognising that every allow-list entry becomes a blind spot, and scheduling exception reviews accordingly
- Applying rarity and first-seen logic instead of writing rules that are either too generic or too specific
- Deciding deliberately when not to alert, and converting noisy rules into hunting dashboards or weekly review queries instead of deleting them
- Writing resilient detections keyed on the attacker's objective and durable result, so they survive renaming, recompilation and LOLBin substitution
- Keeping rule logic readable and documented so any teammate understands what an alert means and what to do about it
- Validating every rule the way the course does: simulate, check the volume, clean the false positives, re-simulate, then ask how you would bypass it
- Porting a detection idea across platforms by learning each product's schema, treating KQL as one implementation of a universal concept
- Learning what normal looks like on Windows before hunting for abnormal
- Working the practice labs independently and building your own detection rules for them without a provided answer key

Video Content:

- Pro Tips
- Attack Simulation and Detection Engineering Practice Labs
- Thank you

Bonus Content: Windows Paths & Registry Keys Reference

- How the core Windows system directories are laid out, and why System32 holds 64-bit binaries while SysWOW64 holds the 32-bit ones
- Which user-profile and AppData locations attackers prefer for dropping and running code, and how their permissions differ
- The difference between the two Temp directories, and why their access control lists change what an attacker can do there
- Why ProgramData is interesting both as a staging location and as a source of environment fingerprinting such as VMware Tools detection



- Which world-writable and template profile folders - Public and Default - deserve attention, and why
- How the Prefetch directory lets you reconstruct what executed, when, and in what sequence
- Where the registry hives and Windows event logs physically live on disk, and why those files are credential and evidence targets
- Which registry keys record user activity: shellbags, RecentDocs, open/save dialog history and Office recent files
- Which registry keys record device history, including USB storage enumeration and mounted device letter assignments
- How to turn each of these locations into a detection rule, a periodic hunt or a baseline exercise

Video Content:

- Interesting Windows Folder Paths - Part 1,2
- Interesting Windows Registry Keys

Required Materials

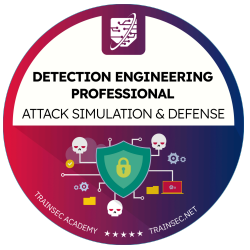
- **Prerequisites**:
  - A couple of years of hands-on cybersecurity experience (red team, blue team, or general security), or completion of the Cybersecurity Fundamentals course plus equivalent hands-on knowledge
  - Basic understanding of networking: TCP/IP, routing, forwarding, VLANs and trunk ports
  - Basic understanding of Windows Server, Active Directory and Linux shell commands
  - Basic understanding of well-known protocols such as HTTP/HTTPS, DNS, SMTP, FTP, SSH, SMB and RDP
  - Comfort reading command lines, scripts and simple XML configuration
  - PC/MAC with Intel i5/i7/i9 CPU, 16GB of RAM (32GB recommended) and an SSD storage
  - VMware Workstation/Fusion installed

Lab Environment:

- A Windows Active Directory lab - one domain controller and at least one Windows workstation
- Sysmon (Sysinternals) for deep endpoint telemetry
- An Elastic Cloud trial with Elastic Security and the Elastic Defend integration

Additional Resources: https://discord.com/invite/qugcNyWdaU

Instructor Information



- **Name**: Uriel Kosayev
- **Email**: uriel@trainsec.net
- **Trainers Biography**:
  Uriel: Security researcher, consultant, and the author of the "*Antivirus Bypass Techniques book"* who lives both on the offensive and defensive fronts. Passionate about malware research, and red teaming while providing real-world security solutions.

Contact & Support

For any course-related or billing-related inquiries, please contact info@TrainSec.net