DIGITAL FORENSICS · INCIDENT RESPONSE

CBRFIR 300-215

Complete Learning Guide

Conducting Forensic Analysis and Incident Response

UNOFFICIAL STUDY GUIDE

All 5 domains · 18 original diagrams · 159 practice questions

JOZEF BAROŠ

CBRFIR 300-215

Complete Learning Guide

Unofficial Study Guide · Free Sample

Exam version v1.2

About This Free Sample

Thank you for previewing the **CBRFIR 300-215 Complete Learning Guide** — an independent, unofficial study guide for the **300-215** exam (*Conducting Forensic Analysis and Incident Response*), mapped to the **v1.2** version of the exam topics.

This sample gives you a genuine taste of the full book: the same layout, the same original diagrams, the same style of explanation, and a complete knowledge check with balanced, explained answers. What you see here is exactly how all five domains are taught in the complete edition.

The full guide includes	Detail
All five exam domains	In proportion to their exam weight (20/20/30/15/15)
Original diagrams	18, drawn from scratch — no third-party artwork
Practice questions	159, with explanations and balanced A/B/C/D answers
Runnable examples	YARA, Volatility, Wireshark, objdump, Python/PS/Bash
Extras	Exam-day strategy, hands-on lab appendix, 56-term glossary

> ✅ **Exam Tip — What to look for in this sample**
>
> Notice how each concept is built from first principles, reinforced with an original diagram, and locked in with a five-question knowledge check. That pattern repeats through every sub-section of the full book.

Important Notice

Unofficial & Independent Publication

This book is an independent, unofficial study guide. It is **not** authorized, sponsored, endorsed by, or otherwise affiliated with Cisco Systems, Inc.

Cisco® , CCNP® , CCNA® , and all related product, technology, and certification names are trademarks or registered trademarks of Cisco Systems, Inc. in the United States and other countries. They are used solely for identification and educational purposes, under nominative fair use, to indicate the examination for which this guide prepares the reader.

The exam identifiers “CBRFIR” and “300-215” are used only to identify the specific certification exam this guide covers. This guide maps to the **v1.2** version of the exam topics current at the time of writing; exam versions and topics may change, so verify the current version with the official certification provider.

All diagrams, tables, code, and explanations are the original work of the author. No official courseware, figures, or copyrighted material is reproduced.

This guide is provided for educational purposes on an “as is” basis, without warranty of any kind. Passing any certification exam depends on the candidate's own study, practice, and hands-on experience.

The 300-215 Exam at a Glance

This guide maps to the **v1.2** version of the exam topics. Always confirm current details — including the exam version — with the official certification provider, as specifics can change.

Attribute	Detail
Exam code	300-215 (CBRFIR)
Exam version	v1.2 (this guide maps to the v1.2 topic list)
Exam focus	Digital forensics and incident response (DFIR)
Duration	90 minutes
Domains	Five, weighted 20/20/30/15/15

The chapters below are a direct excerpt from the full book — Chapter 1's opening and its first sub-section, presented exactly as they appear in the complete edition.

CHAPTER ONE · 20% OF EXAM

Fundamentals

Every forensic investigation and every incident response effort rests on a shared foundation: a common vocabulary, a repeatable process, and a toolkit you can reach for under pressure. This chapter builds that foundation. Before we ever parse a log or carve a memory image in later chapters, we need to know **what a good investigation looks like**, how attackers try to erase their tracks, how data hides in plain sight, and which tools open which doors.

The exam devotes 20% of its questions to these fundamentals — a deliberately large share, because weak fundamentals produce wrong conclusions no matter how sophisticated your tooling. We open with the shape of the whole discipline: the DFIR lifecycle.

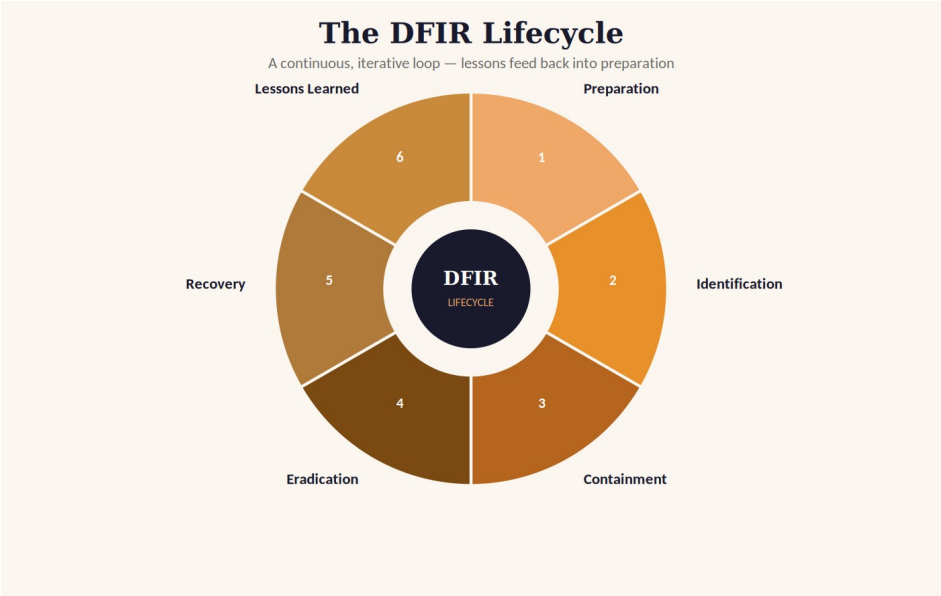

Figure 1.1 — The DFIR lifecycle: a continuous loop where every incident teaches the next.

Digital Forensics and Incident Response (**DFIR**) is not a single act but a cycle. **Digital forensics** is the disciplined recovery, preservation, and analysis of digital evidence; **incident response** is the coordinated effort to detect, contain, and recover from a security event. They interlock: forensics answers *what happened and how*, while incident response acts on those answers to stop the bleeding and prevent recurrence.

> ✅ **Exam Tip — Order matters**
>
> When a question asks for the *next* step, anchor yourself on this lifecycle. A very common trap is jumping to eradication before containment — you must stop the spread and preserve volatile evidence before you start deleting things.

1.1 Components of a Root Cause Analysis Report

A **root cause analysis** (RCA) report is the deliverable that closes the loop of an investigation. It is not a blow-by-blow timeline for its own sake — it is a structured argument that answers three questions for a decision-maker: *what actually went wrong at the deepest level, how do we know, and what must change so it cannot happen again?* An incident that ends without an RCA is an incident you are doomed to repeat.

The word **root** is the key. The proximate cause of a breach might be "an employee clicked a link," but the root cause is usually structural: no MFA, no email filtering, a flat network that let one click become domain-wide compromise. RCA drills past symptoms to the systemic conditions that made the incident possible.

The fishbone: organizing causes before you conclude

Analysts borrow the **Ishikawa (fishbone) diagram** from quality engineering to avoid tunnel vision. You place the effect — the incident — at the head, then branch the contributing causes into categories so no whole dimension of failure gets overlooked.

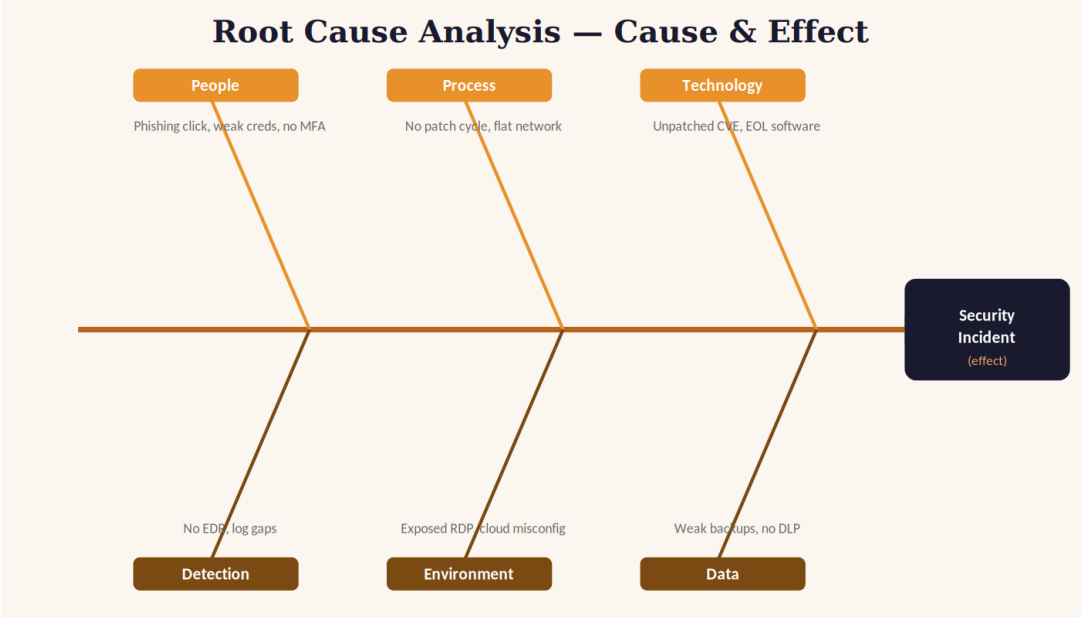

Figure 1.2 — A fishbone diagram sorts contributing causes into categories, forcing breadth before you commit to a root cause.

Each bone is a category of contributing cause; each twig is a specific finding backed by evidence. Only after the whole skeleton is populated do you reason toward the **root** — the cause whose removal would have prevented the chain.

Required components

A defensible RCA report contains, at minimum, the following sections. Missing any one weakens the report's ability to drive change or survive scrutiny.

Component	Purpose
Executive summary	One page a manager can act on: impact, root cause, top recommendations

Component	Purpose
Incident timeline	Evidence-anchored sequence with timestamps and time zone (UTC)
Scope & impact	Systems, data, users, and business functions affected
Root cause statement	The systemic cause, distinguished from proximate triggers
Evidence & methodology	Artifacts examined, tools used, chain of custody
Remediation & recommendations	Prioritized, assigned, time-bound corrective actions

💡 Did you know? — The 5 Whys

A quick technique that pairs with the fishbone: ask "why?" repeatedly until you reach something systemic. "Host was encrypted → why? Ransomware ran → why? User ran an attachment → why? It bypassed the filter → why? No attachment sandboxing → why? Never budgeted." The last answer is the root cause — and it is a policy gap, not a person.

⚠️ Common Pitfall — Blaming the human

Stopping at "the user clicked" is the classic RCA failure. Humans will always click eventually; a mature RCA treats that as a given and asks why one click was catastrophic. Reports that blame individuals get filed and forgotten; reports that name systemic gaps drive funding and change.

🌍 From the Field — Timestamps in UTC, always

In the field, the single most common error that wrecks a timeline is mixing local time zones across artifacts collected from servers in different regions. Normalize every timestamp to UTC in the report and note the original offset.

Summary

A root cause analysis report converts an investigation into organizational change. It drills from proximate triggers to systemic root causes, using tools like the fishbone diagram and the 5 Whys to guarantee breadth before conclusion. A complete report carries an executive summary, an evidence-anchored UTC timeline, scope and impact, an explicit root cause statement, methodology and chain of custody, and prioritized remediation.

Key Takeaways

Root ≠ proximate: name the systemic cause, not just the trigger. **Fishbone + 5 Whys** force breadth and depth. **Executive summary** must be independently actionable. **Normalize timestamps to UTC. Never stop at "the user clicked."**

Knowledge Check — 1.1

Q1. What distinguishes a *root* cause from a *proximate* cause in an RCA report?

A. The root cause is always human error while the proximate cause is technical

B. The root cause is whatever happened first chronologically

C. The root cause is the systemic condition that made the incident possible; the proximate cause is the immediate trigger

D. There is no meaningful difference; the terms are interchangeable

Correct answer: C. RCA drills past the immediate trigger (proximate cause) to the systemic condition whose removal would have broken the chain (root cause).

Q2. Which technique borrows from quality engineering to sort contributing causes into categories?

- **A.** The Ishikawa (fishbone) diagram
- **B.** The MITRE ATT&CK matrix
- **C.** A YARA condition block
- **D.** A packet capture funnel

Correct answer: A. The fishbone/Ishikawa diagram places the effect at the head and branches causes into categories, forcing breadth before conclusion.

Q3. Why should every timestamp in an RCA timeline be normalized to UTC?

- **A.** Because UTC is legally required in all jurisdictions
- **B.** Because forensic tools cannot read local time
- **C.** To make the report shorter
- **D.** To prevent time-zone ambiguity from corrupting the causal sequence

Correct answer: D. Artifacts from servers in different regions carry different offsets; normalizing to UTC removes the ambiguity that can misorder events.

Q4. An RCA report concludes only that "an employee clicked a phishing link." What is the primary weakness?

- **A.** It uses the wrong diagram type
- **B.** It stops at the proximate trigger and ignores the systemic gaps that made one click catastrophic
- **C.** It is too technical for management
- **D.** It omits the malware hash

Correct answer: B. Blaming the human is the classic RCA failure; a mature report asks why a single click could cause domain-wide impact.

Q5. Which component makes an RCA report independently actionable for a manager?

- **A.** The executive summary
- **B.** The full packet capture
- **C.** The YARA rule set
- **D.** The raw disk image

Correct answer: A. The executive summary condenses impact, root cause, and top recommendations into a page a decision-maker can act on.

1.4 Encoding & Obfuscation

Malicious content rarely travels in the clear. It is **encoded** to survive transport, **encrypted** to stay secret, and **obfuscated** to resist analysis — three distinct ideas that exam questions love to blur together. Confusing them leads to wasted effort: you cannot "decrypt" Base64 (there is no key), and you cannot simply "decode" AES (you need one).

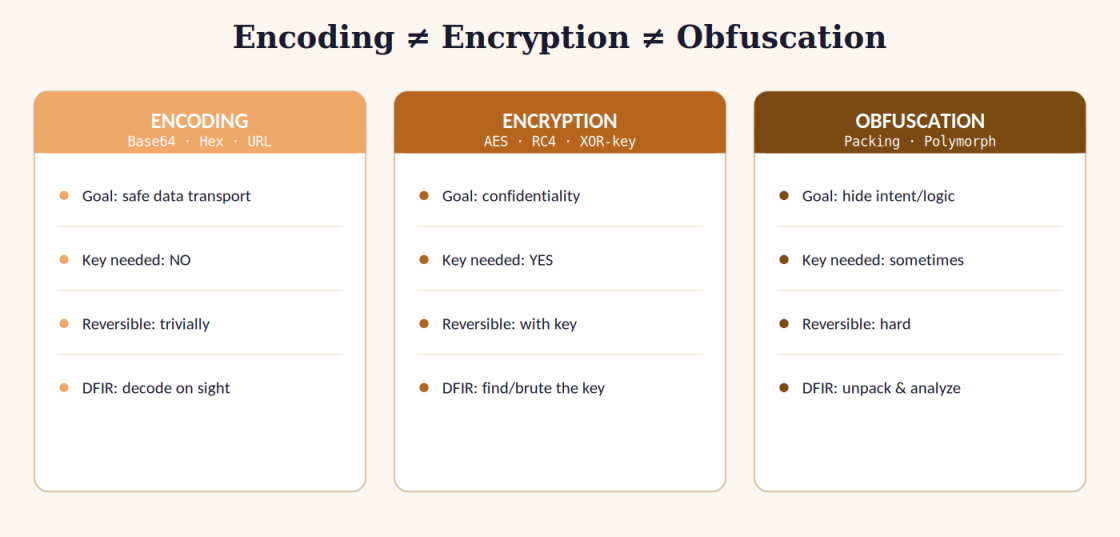

Figure 1.3 — Encoding, encryption, and obfuscation solve different problems and demand different responses.

Polymorphic vs. metamorphic

Obfuscation hides intent while preserving behavior. Two advanced forms defeat signature detection differently, and the exam tests the distinction directly:

- **Polymorphic** malware keeps a constant underlying payload but wraps it in a changing encryptor/packer. Each specimen looks different on disk, yet a **decrypted core is identical** across variants.
- **Metamorphic** malware rewrites its **own code** each generation — reordering instructions, inserting junk, substituting equivalent operations — so there is no constant decrypted core at all.

> ✅ **Exam Tip — The one-line discriminator**
>
> **Polymorphic = same body, different wrapper. Metamorphic = the body itself is rewritten.** If a question hinges on whether a constant decrypted payload exists, "yes" points to polymorphic and "no" to metamorphic.

> 💡 **Did you know? — Entropy as a tripwire**
>
> Packed, encrypted, or compressed regions have high **Shannon entropy** (near 8 bits/byte), while normal code and text sit lower. Entropy analysis flags a likely packed/obfuscated binary before you ever unpack it.

> ⚠️ **Common Pitfall — Calling Base64 "encryption"**
>
> Base64 offers zero confidentiality — it is a public, keyless transformation. Treating it as encryption (or

trying to "crack" it) wastes time. Decode it and move on.

Summary

Encoding, encryption, and obfuscation are distinct: encoding changes representation with no key, encryption provides secrecy with a key, and obfuscation hides intent while preserving behavior. Base64 and hex are keyless and should be decoded on sight. Among obfuscation methods, polymorphic malware varies its wrapper over a constant decrypted core, while metamorphic malware rewrites its own body so no constant core exists — the single most tested distinction here.

Key Takeaways

Encoding ≠ encryption ≠ obfuscation. Base64/hex are keyless — decode, don't "crack." **Polymorphic:** changing wrapper, **constant** core. **Metamorphic:** the code body itself is rewritten. **High Shannon entropy ≈ 8** flags packing/encryption.

Knowledge Check — 1.4

Q1. What fundamentally separates encoding from encryption?

A. Encoding is a keyless, public transformation for representation; encryption requires a key and provides confidentiality

B. Encoding is stronger than encryption

C. Encoding is only used by attackers

D. There is no difference

Correct answer: A. Encoding (Base64, hex) changes representation with no secret and is reversible by anyone; encryption needs a key and hides content.

Q2. Which statement correctly distinguishes polymorphic from metamorphic malware?

A. Polymorphic rewrites its code; metamorphic only changes the wrapper

B. Both keep an identical decrypted core

C. Polymorphic keeps a constant decrypted core under a changing wrapper; metamorphic rewrites its own code body

D. Neither can evade signatures

Correct answer: C. Polymorphic varies the encryptor over a constant payload; metamorphic rewrites the actual instructions so no constant core exists.

Q3. You find the string 4d5a9000 at the start of a decoded blob. What does it most likely indicate?

A. A valid AES key

B. A Windows PE executable (the MZ header)

C. A JPEG image

D. A YARA condition

Correct answer: B. 4D 5A is the ASCII "MZ" DOS/PE header; seeing it after decoding signals an embedded Windows executable.

Q4. Why is trying to "crack" a Base64 string a mistake?

A. Base64 uses AES internally

B. Base64 is unbreakable

C. Base64 is a hashing algorithm

D. Base64 has no key — it is a public, reversible encoding you simply decode

Correct answer: D. Base64 provides no confidentiality and no key exists; you decode it directly.

Q5. A binary shows a section with Shannon entropy near 8 bits per byte. What is the most reasonable inference?

A. The file is definitely benign

B. The file is plain ASCII text

C. The section is likely packed, compressed, or encrypted

D. The section is empty

Correct answer: C. Near-maximal entropy indicates high randomness typical of packed/encrypted/compressed data.

Keep Reading — Get the Full Guide

This sample covered Chapter 1's opening and two of its sub-sections. The complete **CBRFIR 300-215 Complete Learning Guide** continues through all five exam domains with the same clarity, the same original diagrams, and a knowledge check after every sub-section.

In the full edition you get:

- **All five domains** — Fundamentals, Forensics Techniques, Incident Response Techniques, Forensics Processes, and Incident Response Processes — sized to their exam weight.
- **18 original diagrams** covering the DFIR lifecycle, ATT&CK mapping, alert triage, attack surface, the threat-intelligence cycle, STIX/TAXII, and more.
- **159 practice questions** with explanations and balanced A/B/C/D answers.
- **Runnable examples** — YARA, Volatility, Wireshark filters, objdump, and Python/PowerShell/Bash log-parsing scripts.
- **Exam-day strategy, a hands-on lab appendix, and a 56-term glossary.**

> 🌍 **From the Field — Get the complete guide**
>
> Read the full CBRFIR 300-215 Complete Learning Guide (v1.2) on LeanPub at leanpub.com/cbrfir — all five domains, 18 original diagrams, and 159 practice questions.

leanpub.com/cbrfir

Complete edition · 129 pages · all 5 domains · 159 practice questions

Good luck on the exam — and in the investigations that follow it.

— Jozef Baroš