BLUE TEAM FIELD HANDBOOK · EDITION 1

Phishing Email Investigation Guide

From First Alert to Incident Closure

A complete, step-by-step investigation handbook for SOC Analysts (L1–L3), Threat Hunters, DFIR Analysts, Incident Responders, Security Engineers and cybersecurity students — built around Microsoft Defender XDR, Microsoft Sentinel, Exchange Online and the modern open-source analysis toolkit.

- Header & Authentication Analysis
- SPF / DKIM / DMARC
- URL & Attachment Detonation
- 70+ KQL Hunting Queries
- MITRE ATT&CK Mapping
- AiTM & BEC Playbooks
- Decision Trees
- Hands-on Scenarios
- Incident Report Templates
- Interview Q&A

AUTHOR

Ganesha T

Cyber Security Consultant | Threat Hunter

Practical Handbook

Detect · Analyse · Contain · Report

A4 · Print-friendly · Reference Edition

Contents

Every chapter and section, with its page number.

About This Handbook

What it is, who it is for, and how to get the most out of it.

Phishing is still the single most common way that attackers get their first foothold inside an organisation. It is cheap for the attacker, it scales, and it targets the one component that cannot be patched: people. For a Security Operations Centre, phishing is also the highest-volume alert type you will ever handle — which means the difference between a strong SOC and a weak one is very often just this: **how well do your analysts investigate a phishing email?**

This handbook answers that question end to end. It walks through one complete investigation lifecycle, from the moment an alert or a user report arrives, to the moment the incident is closed and lessons are fed back into detection engineering. Every phase is explained the same way, so you always know what you are looking at:

Objective	What this step is trying to achieve.
Why it matters	What you miss, or get wrong, if you skip it.
How to perform it	The practical procedure, with worked examples.
Expected findings	The evidence and indicators you should end up holding.
Techniques	How SOC analysts, incident responders and threat hunters each approach it.
Tools	Microsoft-native and open-source tooling that does the job.
MITRE ATT&CK	The adversary behaviour you are actually observing.
IOC extraction	The artefacts to record for enrichment, blocking and hunting.
KQL	Copy-paste queries for Defender XDR Advanced Hunting and Sentinel.
Decision point	How the findings push you toward malicious, suspicious or legitimate.
Response actions	What to contain, eradicate, remediate and communicate.

Who this handbook is for

READER	HOW TO USE IT
SOC Analyst (L1)	Read Parts I and II in order. Live off the triage checklist (Ch. 24) and the header cheat sheet (Ch. 25). Your goal is a fast, correct verdict with clean evidence.
SOC Analyst (L2 / L3)	Focus on Chapters 10–17: URL and attachment analysis, threat-intel enrichment, blast-radius scoping and post-compromise identity checks.
Threat Hunter	Chapter 18 (M365 hunting methodology) and the KQL library are your core. Use campaign pivots rather than single-email pivots.
DFIR / Incident Responder	Chapters 6 (evidence preservation), 14 (lateral movement), 16 (containment) and 17 (reporting) are written for you.
Security Engineer	Use the detection gaps recorded in each chapter's lessons-learned section to drive policy and rule changes.
Student / Career-changer	Read cover to cover, then work the three hands-on scenarios in Chapter 21 before attempting the interview questions in Chapter 26.

Conventions used in this book

READING THE CALLOUTS

Concept explains an idea before it is used. **Warning** flags a common trap. **Red flag** marks attacker behaviour. **Good practice** is the recommended way. **Pro tip** is field wisdom from experienced responders. `T1566` denotes a MITRE ATT&CK technique.

All email addresses, domains, IP addresses, URLs, hashes and user names used in examples are fictional or deliberately defanged (for example `hxxps://` and `[.]`) and exist only to illustrate investigation technique. Defang every indicator you put into a ticket, an email or a report — a live link in a report is a phishing email you sent yourself.

PART ONE

Foundations: What You Are Actually Looking At

Before you can investigate a phishing email, you need to understand how email travels, where security controls inspect it, what evidence each stage leaves behind, and how attackers design their campaigns to slip through. This part builds that base.

CHAPTER 1

Phishing, Attackers and the Role of the SOC

What phishing really is, why it keeps working, and what a good investigation is actually supposed to produce.

1.1 What phishing is, in plain English

Phishing is **a message that pretends to be from someone you trust, in order to make you do something that helps the attacker**. That "something" is almost always one of four things: type your password into a fake page, open a file that runs code, approve something (an MFA prompt, an OAuth consent, an invoice payment), or simply reply and start a conversation the attacker can steer.

> **ANALOGY**
>
> Think of a physical office. A stranger in a delivery uniform holds a box and asks you to hold the door open. The uniform is the *brand impersonation*. The box is the *pretext*. The open door is the *action* you performed for them. Phishing is exactly this, delivered at the speed of SMTP and at a cost of almost nothing per attempt.

The four outcomes an attacker wants

OUTCOME	WHAT THE EMAIL ASKS FOR	WHAT IT BECOMES
Credential theft	"Sign in to review the document / your password expires today."	Account takeover, mailbox rules, internal phishing, data theft.
Malware execution	"Open the attached invoice / enable content."	Loader → C2 → ransomware or infostealer.
Fraudulent approval	"Approve this MFA prompt / grant this app access."	MFA fatigue bypass, illicit OAuth consent, token theft.
Business fraud (BEC)	"Our bank details have changed — pay to this account."	Direct financial loss, often with no malware at all.

> **WARNING — THE MOST DAMAGING PHISHING HAS NO PAYLOAD**
>
> Beginner analysts look for a link or an attachment and, if there is neither, close the ticket as spam. Pure BEC and payroll-diversion emails contain nothing but text. There is no URL to detonate and no hash to look up. If you only know how to investigate payloads, you will miss the attacks that cost the most money.

1.2 Why phishing keeps working

- **It exploits authority and urgency, not software.** No patch closes "the CEO asked me to do it quickly."
- **Infrastructure is disposable.** A domain costs a few dollars; a phishing kit is free; a compromised mailbox is even better because it is already trusted.

- **Legitimate services are used as cover.** Links hosted on cloud storage, form builders, marketing redirectors and code-hosting platforms inherit those platforms' good reputation.
- **MFA is no longer a wall.** Adversary-in-the-Middle (AiTM) kits proxy the real login page and steal the *session cookie* after the user completes MFA correctly.
- **Volume beats accuracy.** The attacker needs one success out of ten thousand. You need to be right every time.

1.3 What a phishing investigation must produce

An investigation is not "I looked at it and it seemed dodgy." It must produce five concrete outputs. If your ticket does not contain all five, the investigation is not finished.

#	OUTPUT	DEFINITION OF DONE
1	**A defensible verdict**	Malicious, Suspicious, Spam/Unwanted, or Legitimate — with the evidence that supports it.
2	**A complete IOC set**	Sender addresses, sending IPs, sending domains, URLs, redirect chains, landing domains, file names, hashes, message IDs.
3	**The blast radius**	Every recipient who received it, who read it, who clicked, who submitted credentials, who executed a file.
4	**Containment evidence**	Proof that the mail was purged, the URLs/domains blocked, the accounts reset, the sessions revoked.
5	**A detection improvement**	At least one answer to "why did this reach the inbox, and what would stop the next one?"

1.4 Who does what — roles across the investigation

ROLE	PRIMARY QUESTION	TYPICAL SCOPE IN A PHISHING CASE
SOC L1	"Is this real, and who else got it?"	Triage, header check, authentication result, reputation lookups, initial scoping, escalate or close.
SOC L2	"What does it actually do?"	URL detonation, redirect chains, attachment analysis, sandboxing, IOC extraction, purge and block actions.
SOC L3 / IR	"Did anyone fall for it, and what happened next?"	Click and submission analysis, sign-in forensics, token/session theft, inbox rules, lateral movement, containment.
Threat Hunter	"Where else is this campaign hiding?"	Infrastructure pivots, campaign clustering, retro-hunting across 30–90 days, TTP-based hunts.
DFIR Analyst	"Prove what happened."	Evidence preservation, timeline reconstruction, host forensics if malware executed, root-cause and report.
Security Engineer	"Why did it get through?"	Policy tuning, transport rules, DMARC enforcement, new analytic rules, awareness content.

PRO TIP — WRITE THE VERDICT SENTENCE FIRST

Experienced responders draft one sentence before they start: "*This is a [verdict] email that attempted [objective] using [technique], affecting [N] recipients.*" Then they investigate to prove or disprove every bracket. It keeps you from wandering, and the sentence becomes the executive summary of your report.

CHAPTER 2

How Email Really Works — Mail Flow and Inspection Points

You cannot read a header until you understand the journey that wrote it. This chapter maps the path an email takes, where each control inspects it, and which log each stage produces.

2.1 The players (learn these five words)

MUA — Mail User Agent	The client: Outlook, the mobile app, webmail. Where a human writes and reads.
MTA — Mail Transfer Agent	The mail server that relays the message across the internet. Each MTA adds a `Received:` line.
MX record	DNS record that says "mail for this domain goes here." The attacker's sending MTA looks yours up.
Envelope	The SMTP conversation itself: `MAIL FROM` and `RCPT TO`. The postal envelope.
Header/body	The letter inside the envelope: `From:`, `To:`, `Subject:`, content. Fully attacker-controlled.

RED FLAG CONCEPT — TWO DIFFERENT "FROM" VALUES

Email has **two** senders. The **envelope sender** (`MAIL FROM`, also called Return-Path or P1 sender) is what SPF checks. The **header sender** (`From:`, the P2 sender) is what the user *sees*. An attacker can make these different. That single fact is the root of most spoofing, and it is why **DMARC alignment** exists. Memorise it now — the whole of Chapter 8 depends on it.

2.2 The journey of an inbound email

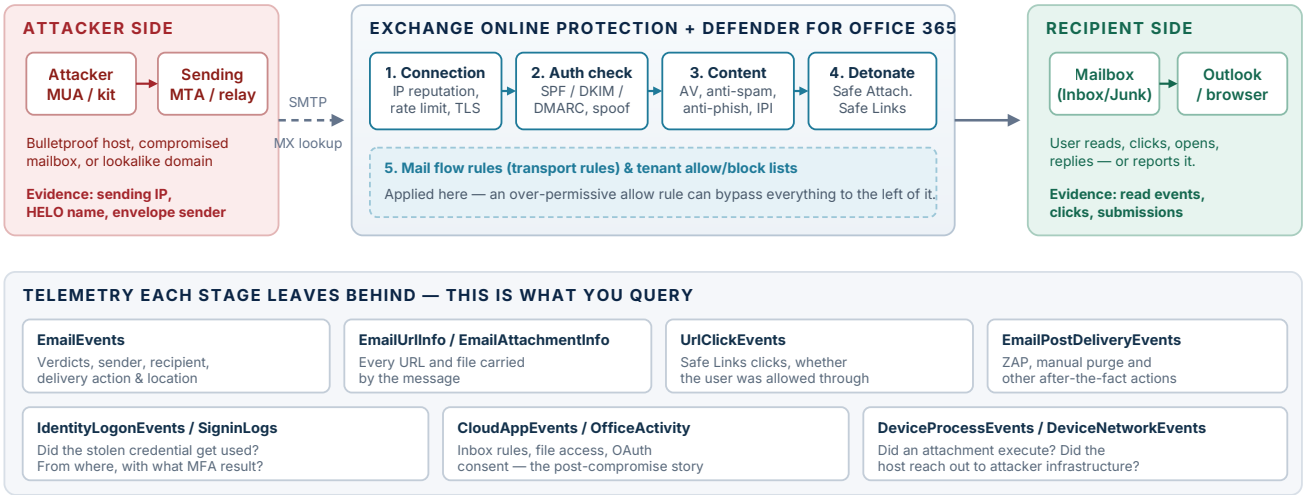

Figure 2.1 — Inbound mail flow and the evidence it generates. Every box on the top rail writes to a table on the bottom rail. When you plan an investigation, you are really deciding which of these tables to open, and in what order.

2.3 Reading the journey backwards

Headers are written **top-down in reverse chronological order**. The `Received:` line at the *top* is the last hop (your own mail system); the one at the *bottom* is the first hop (closest to the attacker). So you read a header stack from the bottom up if you want to follow the mail forwards in time, and from the top down if you want to peel back trust.

GOOD PRACTICE — THE TRUST BOUNDARY

Only the `Received:` lines added by servers *you* control (or by a provider you trust, such as Exchange Online) are trustworthy. Everything below that boundary was written by machines the attacker may own, and can be entirely forged. The **lowest trustworthy Received line** tells you the IP that actually handed the mail to you. That IP is your first hard fact.

2.4 Where email security controls sit — and how each is bypassed

CONTROL	WHAT IT DOES	HOW ATTACKERS GET PAST IT
Connection filtering	Blocks known-bad sending IPs before content is even accepted.	Send from fresh cloud VMs, compromised legitimate mailboxes, or reputable ESPs.
SPF / DKIM / DMARC	Verifies that the sending domain authorised the sender.	Don't spoof at all — register a lookalike domain and pass authentication legitimately.
Anti-spam / anti-phish	Scores content, detects impersonation of your users and domains.	Minimal text, image-only bodies, benign wording, no obvious keywords.
Safe Attachments	Detonates files in a sandbox before delivery.	Password-protected archives (sandbox can't open them), or no attachment at all.
Safe Links	Rewrites URLs and re-checks them at click time.	Benign-at-delivery URLs that only turn malicious later; CAPTCHA and geo-fencing to defeat crawlers; QR codes that never appear as a URL.
Mail flow rules	Custom allow/block/route logic.	Abuse an over-broad allow-list entry, e.g. an entire partner domain or a "skip filtering" rule for an IP range.
ZAP (zero-hour auto purge)	Retro-removes mail found bad after delivery.	Move the user quickly — the click happens within minutes of delivery, long before ZAP fires.

PRO TIP — ALWAYS ASK "WHY WAS THIS DELIVERED?"

In Defender XDR the `EmailEvents` table has `DeliveryAction` and `DeliveryLocation`, and the Threat Explorer view shows an *override* reason. If a clearly malicious email landed in the inbox, there is almost always an override: an allowed sender list, a tenant allow entry, a transport rule, or the user's own safe-sender list. Finding that override is often the single most valuable output of the whole investigation, because it is a hole that every future attacker can walk through.

CHAPTER 3

Phishing Taxonomy, Attacker Techniques and Evasion

Know the shapes an attack can take, so that you recognise one at a glance instead of discovering it three hours in.

3.1 The taxonomy — naming what you are looking at

TYPE	DESCRIPTION	TELL-TALE SIGNS IN THE EVIDENCE
Bulk phishing	Mass, low-effort, brand-themed credential harvesting.	Many recipients, same subject, same URL family, low personalisation.
Spear phishing	Targeted at named individuals with researched context.	Few recipients, correct job titles, real project names, no obvious brand.
Whaling	Spear phishing aimed at executives.	Recipients are all VIPs; pretext is legal, M&A or board-level.
BEC / CEO fraud	Impersonates an executive or supplier to move money or data.	Text-only, reply-to mismatch, urgency, secrecy, banking details.
Vendor / supply-chain	Sent from a genuinely compromised partner mailbox.	Perfect authentication, real thread history, changed bank details.
Thread hijacking	Attacker replies into an existing, real conversation.	Genuine `In-Reply-To` and quoted history; only the newest link is bad.
AiTM phishing	Proxies the real login page to steal the post-MFA session cookie.	Redirect chain ends at a proxy; sign-in succeeds from an odd IP *right after* the click.
Quishing (QR)	URL is embedded in a QR image, often in a PDF.	Image-only body, no clickable URL, victim's phone is the click device (often unmanaged).
Consent phishing	Tricks the user into granting an OAuth app permissions.	Link goes to a genuine Microsoft consent page for an attacker-owned app.
Callback / TOAD	"Call this number to cancel your subscription" — attack continues by phone.	PDF invoice, phone number, no URL and no malware at all.
Malware phishing	Delivers a loader via attachment or link.	Archives, ISO/IMG, OneNote, LNK, macro documents, HTML smuggling.
Smishing / Vishing	SMS and voice variants of the same social engineering.	Reported by the user, not by email controls; often part of a multi-channel campaign.

3.2 The phishing attack chain

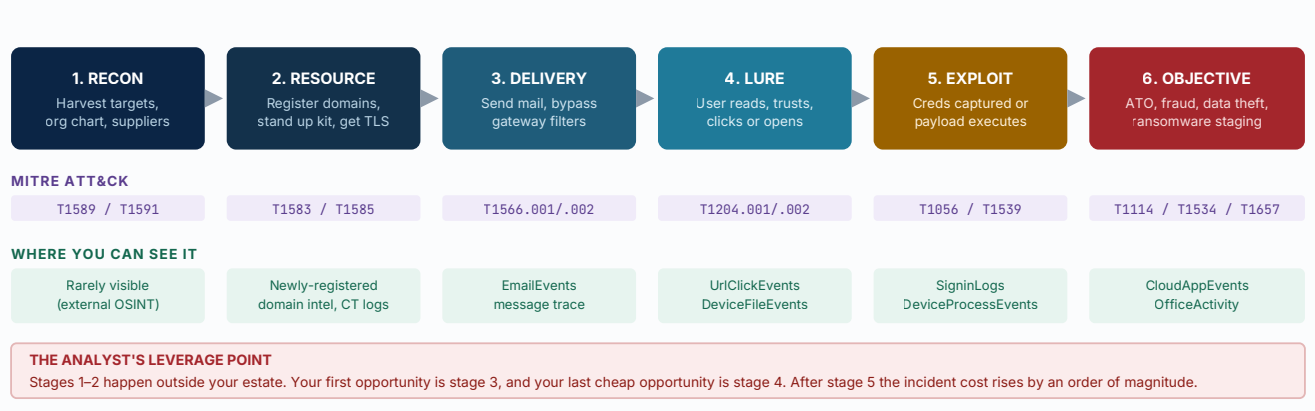

Figure 3.1 — The phishing attack chain, mapped to ATT&CK techniques and to the telemetry where each stage becomes observable. Investigation is the art of pulling the timeline as far left as possible.

3.3 Evasion techniques you must recognise

Against the mail gateway

- **Reputation borrowing.** Send from Gmail, a marketing platform, a survey tool, or a compromised partner — all of which pass SPF/DKIM/DMARC perfectly, because the attacker is not spoofing anyone.
- **Lookalike (cousin) domains.** `rn` for `m`, `l` for `I`, extra hyphens, different TLD: `contoso-support[.]com`, `rnicrosoft[.]com`, `contoso[.]co`.
- **Display-name spoofing.** The `From:` shows *"Sarah Patel (CFO)"* but the address is `sarah.p.cfo@gmail[.]com`. Mobile clients hide the address entirely — which is exactly why BEC works so well on phones.
- **Empty or image-only bodies.** No text for a content filter to score. The whole "email" is one JPEG.
- **HTML smuggling.** The attachment is an innocuous `.html` file; the malicious archive is assembled by JavaScript *inside the browser*, so nothing malicious ever crosses the network.
- **Password-protected archives.** The password is in the email body (or in the image), so the human can open it but the sandbox cannot.

Against URL scanners and sandboxes

- **Time-delayed activation.** The URL serves a harmless page at delivery time and swaps to the phishing kit hours later, after scanning has passed.
- **Cloaking.** The kit inspects User-Agent, ASN, geolocation and headless-browser tells. Security vendors get a 404 or a cat picture; the victim in the right country on a real phone gets the login page.
- **CAPTCHA and "click to continue" gates.** Automated crawlers stop; humans click through.
- **Open redirects.** The link in the mail is on a genuinely trusted domain, which then forwards to the attacker's page. Reputation checks on the first hop come back clean.
- **URL shorteners and multi-hop chains.** Five redirects, each on a different reputable service.
- **Personalised URLs.** The victim's email address is encoded in the URL (in the path, query string or fragment). Strip it and the kit refuses to load — which is why a naive "clean" verdict from a scanner is meaningless.

- **QR codes.** The URL exists only as pixels, and the click happens on a personal phone that has none of your controls or logging.

Against the human

- **Authority** ("the CEO needs this"), **urgency** ("within the hour"), **fear** ("your account will be closed"), **curiosity** ("see the attached bonus letter"), and **helpfulness** ("can you do me a favour, I'm in a meeting").
- **Secrecy clauses** — "don't discuss this with anyone yet" — which exist purely to stop the victim verifying through a second channel.
- **Conversation-first BEC.** The first email asks nothing at all ("Are you at your desk?"). No links, no attachments, nothing to detect. The attack is in message three.

RED FLAG — THE "CLEAN SCAN" TRAP

VirusTotal shows 0/94. URLScan shows a login page that looks legitimate. The sandbox reports no malicious activity. **None of this proves the email is safe.** It may prove only that the attacker's cloaking worked. Reputation tools answer "has anyone reported this yet?" — not "is this malicious?" A brand-new domain with zero detections and zero history is *more* suspicious than an old domain with a couple of detections, not less.

CHAPTER 4

The Investigation Lifecycle — The 13-Phase Workflow

The spine of this handbook: a repeatable order of operations from alert to closure, and the logic that moves you from one phase to the next.

4.1 Why a fixed order matters

Under pressure, people jump straight to the exciting part — detonating the link. That is how evidence gets destroyed, how scope gets missed, and how an analyst spends forty minutes on a URL that was only ever sent to a test mailbox. A fixed workflow guarantees three things: **evidence is preserved before it is touched**, **scope is known before effort is spent**, and **containment is not delayed by analysis**.

> **GOOD PRACTICE — CONTAIN IN PARALLEL, NOT IN SEQUENCE**
>
> Containment does not have to wait for the final verdict. The moment you have *reasonable suspicion* plus *evidence of a click*, you revoke sessions and reset the password. You can always apologise for an unnecessary password reset. You cannot un-exfiltrate a mailbox.

4.2 The workflow

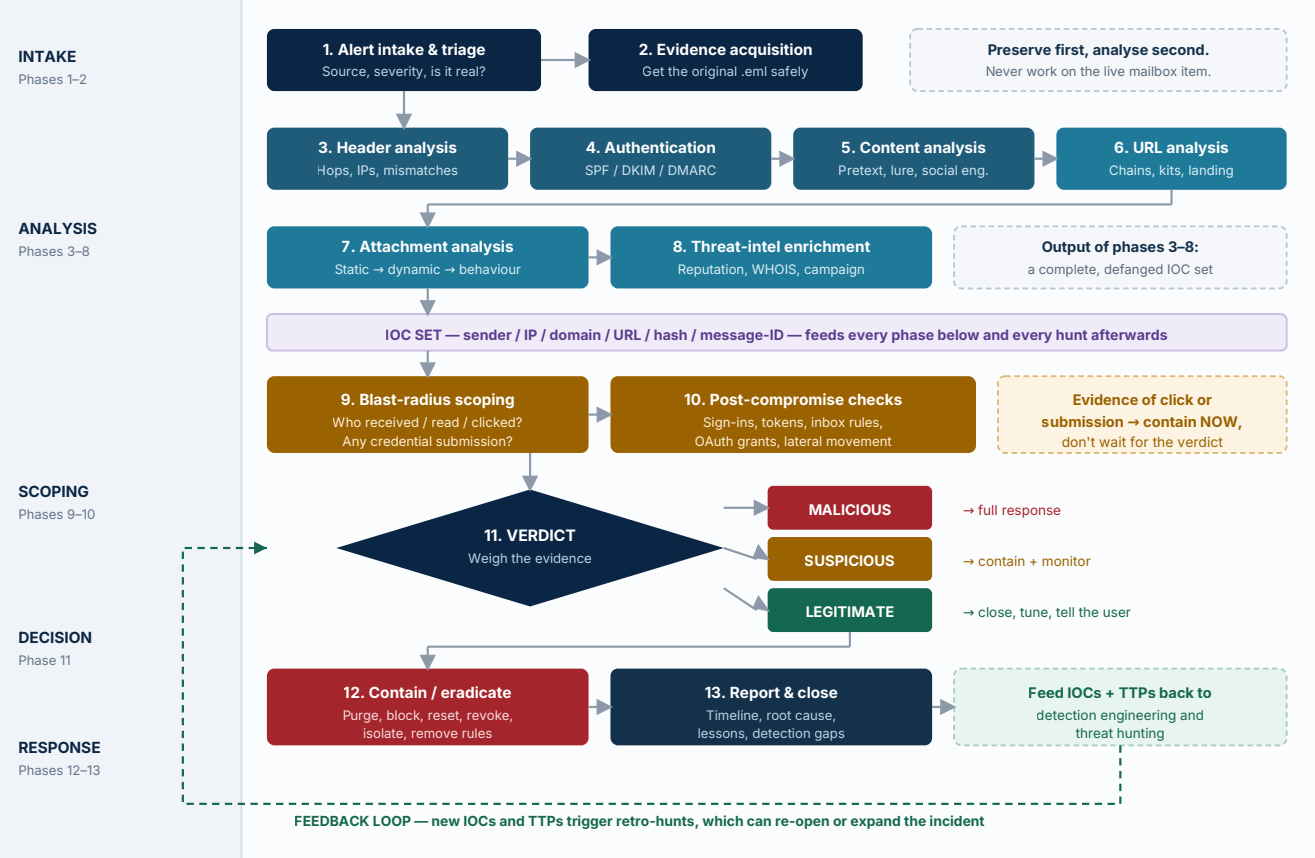

Figure 4.1 — The 13-phase phishing investigation workflow. Chapters 5–17 follow this diagram exactly, one chapter per phase. Note the two shortcuts: containment can start as soon as a click is confirmed (phase 10), and the feedback loop can re-open a closed case.

4.3 Phase-to-chapter map

PHASE	NAME	CHAPTER	KEY QUESTION IT ANSWERS
1	Alert intake & triage	Ch. 5	Is this worth my next hour, and how urgent is it?
2	Evidence acquisition	Ch. 6	Do I have the original message, safely and defensibly?
3	Header analysis	Ch. 7	Where did it really come from?
4	Authentication analysis	Ch. 8	Was the sender authorised to use that domain?
5	Content analysis	Ch. 9	What is it trying to make the user do?
6	URL analysis	Ch. 10	Where does the link actually go, and what waits there?
7	Attachment analysis	Ch. 11	What does the file do when opened?
8	Threat-intel enrichment	Ch. 12	Is this known, and is it part of something bigger?
9	Blast-radius scoping	Ch. 13	Who else got it, and what did they do?

PHASE	NAME	CHAPTER	KEY QUESTION IT ANSWERS
10	Post-compromise / lateral movement	Ch. 14	Did anyone actually get compromised, and how far did it go?
11	Verdict	Ch. 15	Malicious, suspicious, or legitimate — and can I defend it?
12	Containment, eradication, recovery	Ch. 16	How do I remove the threat and restore trust?
13	Reporting & lessons learned	Ch. 17	What happened, why, and what changes because of it?

4.4 Severity model — how urgent is this?

SEVERITY	TRIGGER	TARGET RESPONSE	IMMEDIATE ACTION
Critical	Credential submission confirmed, or malware executed, or a VIP/finance account is involved, or an attacker is already sending internal phishing.	15 min	Contain first, investigate second. Wake people up.
High	Confirmed malicious email delivered to inboxes; clicks recorded but submission unconfirmed.	1 hour	Purge, block, force password reset on clickers.
Medium	Suspicious email delivered; no clicks yet; indicators unclear.	4 hours	Purge from inboxes, continue analysis.
Low	Blocked or quarantined before delivery; user-reported spam; no exposure.	1 business day	Confirm the block held, document, close.

PRO TIP — SEVERITY IS SET BY EXPOSURE, NOT BY HOW SCARY THE EMAIL LOOKS

A beautifully crafted, obviously malicious email that was quarantined before delivery is a *Low*. A crude, typo-ridden email that reached forty inboxes and got three clicks is a *High*. Always let the telemetry — not the aesthetics — drive the severity.

PART TWO

The Investigation Workflow, Phase by Phase

Thirteen chapters, thirteen phases, one order of operations. Each chapter answers the same eleven questions — objective, why, how, findings, techniques, tools, ATT&CK, IOCs, KQL, decision point, response — so that you can drop into any phase and know exactly where you are.

CHAPTER 5 · PHASE 1

Alert Intake and Triage

The first ten minutes. Decide what you are holding, how urgent it is, and whether anyone has already been exposed — before you spend an hour on analysis.

PHASE 1 **Alert Intake and Triage**

OBJECTIVE

Establish, quickly and cheaply, **what arrived, where it came from, who has it, and whether it was delivered or blocked** — and from that, set a severity that decides how fast everything else moves.

WHY THIS STEP IS IMPORTANT

Triage is where you decide how to spend the next hour of a shift that has forty other tickets in it. Get it wrong in one direction and you burn a morning on a marketing newsletter. Get it wrong in the other and a credential-harvesting email sits in twenty inboxes for six hours. Triage also protects the evidence: this is the phase where you stop other people from deleting the mail before you have a copy.

EXPECTED FINDINGS

A one-line summary of the message (sender, subject, timestamp), the delivery outcome, the recipient count, the detection source, and an initial severity.

5.1 Where phishing alerts come from

SOURCE	WHAT IT TELLS YOU	WHAT IT DOES *NOT* TELL YOU
User report (Report Phishing button)	A human found it suspicious. Highest-value source: users catch the payload-free BEC that machines miss.	Nothing about scope. The reporter is rarely the only recipient.
Defender for Office 365 alert	A control fired — e.g. "A potentially malicious URL click was detected" or "Email messages containing malicious URL removed after delivery".	Whether the user actually got compromised.
Defender XDR incident	Multiple alerts correlated across email + identity + endpoint. Treat as high priority by default.	The full timeline — correlation is not investigation.
Sentinel analytic rule	Custom logic, often campaign- or IOC-driven.	Context; you must pull the original message yourself.
Threat intel / ISAC feed	A campaign is hitting your sector. Retro-hunt immediately.	Whether *you* were hit — that is your hunt to run.
Abuse mailbox / helpdesk	Often the first sign of an internally-sent phish from an already-compromised account.	Usually arrives with the message forwarded as text, which destroys the headers — see Chapter 6.

WARNING — A USER REPORT IS A DETECTION FAILURE

Every user-reported phish that reached an inbox is, by definition, an email your controls let through. The ticket is not finished when you purge the mail; it is finished when you have answered *why the control missed it*. Record that answer in the lessons-learned field (Chapter 17). Over a quarter, those answers become your detection-engineering backlog.

5.2 How to perform triage

Work the following six questions *in order*. Stop and escalate immediately if question 5 or 6 is a yes.

1. **What is the message?** Sender address, display name, subject, received time, message ID.
2. **What was the verdict and delivery outcome?** Was it blocked, quarantined, sent to Junk, or delivered to the Inbox? A delivered message is a live exposure; a quarantined one is not.
3. **How many recipients?** One user, one team, or the whole tenant? This is your scope and it drives severity more than anything else.
4. **Does it carry a payload?** URLs, attachments, QR image, or none (pure BEC).
5. **Did anyone click or open?** Check `UrlClickEvents` before you do anything else.
6. **Are the recipients high-value?** Finance, payroll, executives, IT admins, HR.

Worked example

TRIAGE NOTE (WHAT A GOOD L1 WRITES IN THE FIRST 10 MINUTES)

Ticket SOC-14892. User-reported email. Sender `billing@contoso-invoices[.]com` (display name "Contoso Billing"). Subject: "Invoice 44821 – overdue, action required". Received 09:12 UTC. **Delivered to Inbox** (not junked). Message trace shows **14 recipients**, all in Accounts Payable. Carries one URL (`hxxps://secure-contoso-billing[.]com/inv/44821`) and a PDF. `UrlClickEvents` shows **3 clicks, all "ClickAllowed"**. Recipients are finance staff. **Severity: High → escalating to L2 now, and starting containment in parallel.**

5.3 Triage decision tree

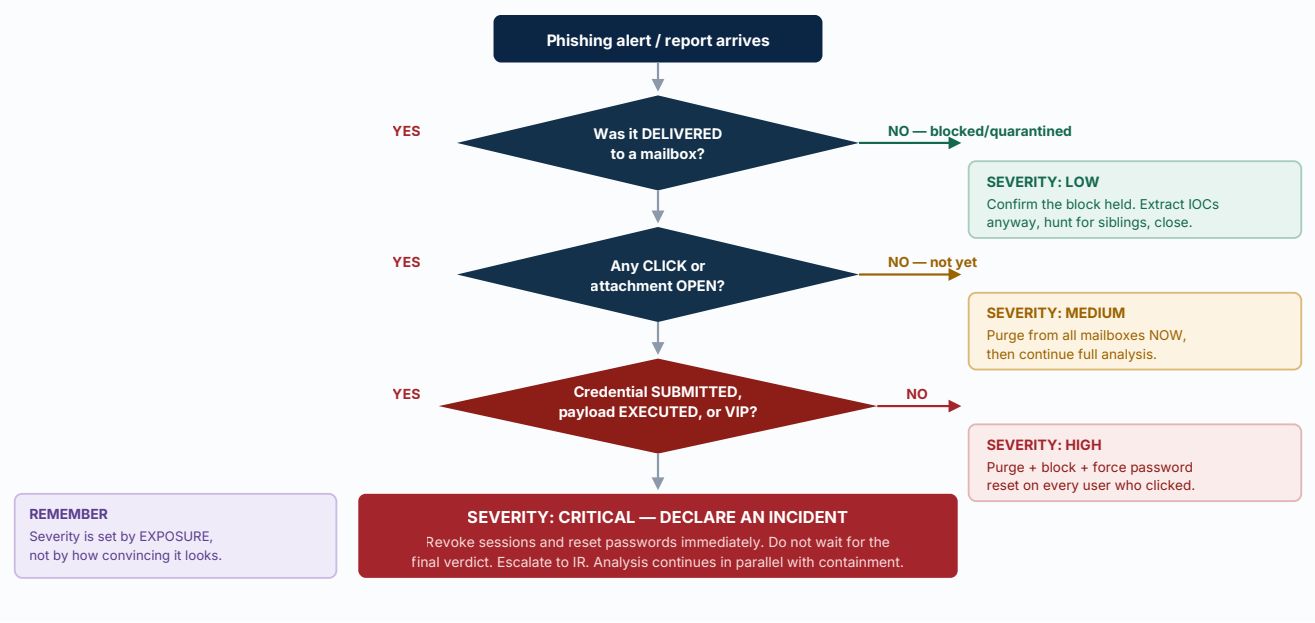

Figure 5.1 — Triage decision tree. Three questions decide severity: was it delivered, was it acted on, and did that action succeed. Everything else is detail you can gather later.

5.4 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE AT THIS PHASE	TOOLS
SOC L1	Run the six triage questions; capture the message identifiers; set severity; preserve evidence.	Defender XDR portal, Threat Explorer, Message Trace, ticketing system
SOC L2/L3	Check for sibling messages from the same infrastructure already in the estate.	Advanced Hunting (KQL)
Threat Hunter	Ask "is this one email, or the first one we noticed?" — pivot on sender domain and URL domain across 30 days.	Advanced Hunting, Sentinel
Incident Responder	Decide whether this becomes an incident with a bridge call, or stays a ticket.	IR playbook, on-call escalation

5.5 MITRE ATT&CK mapping

TECHNIQUE	NAME	RELEVANCE AT THIS PHASE
`T1566`	Phishing	The parent technique for the whole case.
`T1566.001`	Spearphishing Attachment	Assign if the message carries a file.
`T1566.002`	Spearphishing Link	Assign if the message carries a URL or QR code.
`T1566.004`	Spearphishing Voice	Assign for callback/TOAD lures containing a phone number.

5.6 IOC extraction at this phase

You are not doing deep analysis yet — you are recording the identifiers that let you find the message again and pivot on it later. Capture all of them now, because after a purge the message is harder to retrieve.

- **NetworkMessageId** — Microsoft 365's unique ID. This is your primary key for every KQL query.
- **InternetMessageId** — the RFC 5322 `Message-ID` header, e.g. `<a1b2@mail.example.com>`.
- **SenderFromAddress** (header From) and **SenderMailFromAddress** (envelope From).
- **SenderIPv4**, subject line, recipient list, timestamp.

5.7 KQL for triage

Defender XDR — Advanced Hunting | What is this message and where did it land?

```
// Start here for every phishing ticket. Replace the subject or sender.
EmailEvents
| where Timestamp > ago(7d)
| where Subject has "Invoice 44821" or SenderFromAddress =~ "billing@contoso-invoices.com"
| project Timestamp, NetworkMessageId, InternetMessageId, SenderFromAddress,
          SenderMailFromAddress, SenderDisplayName, SenderIPv4, RecipientEmailAddress,
          Subject, DeliveryAction, DeliveryLocation, ThreatTypes, DetectionMethods,
          AuthenticationDetails, EmailDirection, AttachmentCount, UrlCount
| sort by Timestamp desc
```

Defender XDR | How big is this? Recipient count and delivery outcome

```
// Scope in one query: how many people got it, and did it reach the Inbox?
EmailEvents
| where Timestamp > ago(7d)
| where SenderFromAddress =~ "billing@contoso-invoices.com"
| summarize Recipients = dcount(RecipientEmailAddress),
            Messages   = count(),
            Landed     = make_set(DeliveryLocation),
            Actions    = make_set(DeliveryAction),
            FirstSeen  = min(Timestamp),
            LastSeen   = max(Timestamp)
  by Subject
```

Defender XDR | The question that sets severity: did anyone click?

```
// ClickAllowed / ClickBlocked is the difference between Medium and High.
let msgIds = EmailEvents
  | where Timestamp > ago(7d)
  | where SenderFromAddress =~ "billing@contoso-invoices.com"
  | distinct NetworkMessageId;
UrlClickEvents
| where Timestamp > ago(7d)
| where NetworkMessageId in (msgIds)
| project Timestamp, AccountUpn, Url, ActionType, IsClickedThrough, IPAddress, Workload
| sort by Timestamp asc
```

PRO TIP — `ISCLICKEDTHROUGH` IS THE FIELD THAT RUINS YOUR AFTERNOON

`ActionType = "ClickAllowed"` means Safe Links let the user through. `IsClickedThrough = 1` means the user was shown a warning page and **clicked past it anyway**. That user knowingly proceeded to a page Microsoft flagged as risky. Treat them as compromised until proven otherwise.

5.8 Decision point

IF TRIAGE SHOWS...	THEN...	NEXT ACTION
Blocked / quarantined, no delivery	**Low**	Extract IOCs, hunt for siblings, document, close.
Delivered, no clicks, payload present	**Medium**	Purge now; proceed to Chapter 6 (evidence).
Delivered, clicks recorded	**High**	Purge, block, reset clickers; proceed to Chapter 6 and run Chapter 14 in parallel.
Submission, execution, or VIP/ finance target	**Critical**	Declare an incident. Contain first. Everything else runs in parallel.

5.9 How this shapes the next step

Triage hands the next phase three things: the **NetworkMessageId** (so you can find the message), the **severity** (so you know how fast to move), and the **exposure list** (so you know whose mailbox you need to protect). If the message was delivered, your immediate next action is *not* to open it — it is to get a forensically clean copy of it, which is Chapter 6.

CHAPTER 6 · PHASE 2

Evidence Acquisition and Preservation

Get the original message, with its headers intact, without detonating anything and without destroying what you are about to analyse.

PHASE 2 **Evidence Acquisition and Preservation**

OBJECTIVE

Obtain the **original, unmodified message** — ideally as an `.eml` or `.msg` file with full headers — store it safely, and record a hash so that your analysis is defensible later.

WHY THIS STEP IS IMPORTANT

Every later phase depends on this one. Headers cannot be reconstructed once lost. A forwarded copy is *not* evidence: forwarding rewrites the headers, so the `Received:` chain you analyse is the chain from the reporting user to you, not from the attacker to your gateway. Analysts who skip this step end up confidently analysing their own mail server.

EXPECTED FINDINGS

An `.eml` file with complete headers, a SHA-256 hash of it, extracted attachments (defanged and stored in a password-protected archive), and a chain-of-custody note.

6.1 The three ways to get a copy — ranked

RANK	METHOD	NOTES
Best	Download from Defender's Threat Explorer / Email entity page	Preserves the original message exactly as it arrived, headers intact. Requires the "Preview" / download role — get that role assigned *before* you need it.
Good	Ask the user to attach the email to a new mail (drag the message into a blank message as an attachment)	Preserves headers, because the message travels as a `message/rfc822` attachment rather than as forwarded text.
Acceptable	User copies the full internet headers from Outlook (File → Properties → Internet headers) and pastes them into the ticket	Gives you the headers but not the body or attachments. Fine for a quick spoof check, insufficient for a full case.
Never	The user forwards the email normally	Rewrites the header chain, may strip attachments, and renders the message HTML in *your* mailbox. This is the single most common evidence-handling mistake in phishing response.

RED FLAG — NEVER ANALYSE ON A PRODUCTION ENDPOINT

Do not open the `.eml`, its attachments, or its links on your corporate workstation. Use an isolated analysis VM or a cloud sandbox. If your organisation does not have one, do the static analysis only (Chapters 7–9 and static parts of 10–11) and send the dynamic work to a hosted sandbox.

6.2 How to perform it — a safe handling procedure

1. **Record the identifiers first** (from Chapter 5) so you can find the message even if it is purged.
2. **Do not purge yet** if you still need to download the sample — or, if the risk is high, purge first and retrieve the copy from quarantine, where it remains available.
3. **Download the** `.eml` from the Defender email entity page.
4. **Hash it** and record the hash in the ticket. This ties your report to a specific artefact.
5. **Extract attachments without opening them** — use a script, not a double-click.
6. **Defang all indicators** before pasting them anywhere: `hxxps://`, `[.]`, `[@]`.
7. **Store** the sample in the case folder, in a password-protected archive (password: `infected` is the industry convention), and note who took it, when, and from where.

Analysis VM | Safe extraction and hashing — nothing is executed

```
# Hash the sample first - this is your chain-of-custody anchor
Get-FileHash .\sample.eml -Algorithm SHA256

# Extract headers, body and attachments WITHOUT opening the message
python3 -c "
import email, hashlib, pathlib
msg = email.message_from_file(open('sample.eml', errors='ignore'))
print('--- HEADERS ---')
for k in ['Return-Path','Received','Authentication-Results','From','Reply-
To','To','Subject','Date','Message-ID']:
    for v in msg.get_all(k) or []:
        print(f'{k}: {v}')
print('--- ATTACHMENTS ---')
for part in msg.walk():
    fn = part.get_filename()
    if fn:
        data = part.get_payload(decode=True) or b''
        print(fn, len(data), hashlib.sha256(data).hexdigest())
        pathlib.Path('attach_' + fn).write_bytes(data)   # written, never opened
"
```

GOOD PRACTICE — THE CHAIN-OF-CUSTODY NOTE

Four lines in the ticket is enough for most corporate cases: *who* acquired it, *when* (with timezone), *from where* (Threat Explorer / quarantine / user), and the *SHA-256* of the `.eml`. If the case might ever become an HR matter, a legal matter or an insurance claim, this note is what makes your findings usable.

6.3 Expected findings and IOC extraction

ARTEFACT	WHAT YOU DO WITH IT
Full `.eml`	Source for Chapters 7–11. Hash it, store it, never open it in a mail client.
Header block	Feeds header analysis (Ch. 7) and authentication analysis (Ch. 8).
HTML body	Feeds content and URL analysis (Ch. 9–10). Read it as source, not rendered.
Attachments + SHA-256	Feeds attachment analysis (Ch. 11) and threat intel (Ch. 12).
Embedded images	May contain a QR code. Decode it — the URL you need may only exist as pixels.

6.4 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Download the sample from Defender; hash it; attach to the ticket.	Defender XDR → Explorer → Email entity
SOC L2	Parse the message offline into headers, body, URLs and attachments.	Python `email` module, `emldump`, Thunderbird (headers only), 7-Zip
DFIR	Preserve with hashes and a custody note; snapshot the mailbox audit log before purge.	Purview eDiscovery / Content Search, Compliance search, PowerShell
Threat Hunter	Keep the sample in a corpus so that future campaigns can be diffed against it.	Internal sample repository, MISP

6.5 Decision point and next step

There is no verdict at this phase — only a gate. **If you do not have a copy with intact headers, you cannot proceed to Chapter 7.** Go back and get one. If the message has already been purged and the user has deleted it, retrieve it from quarantine or from a Purview content search before continuing. With the `.eml` in hand, the next question is the first hard one: *where did this actually come from?*

CHAPTER 7 · PHASE 3

Email Header Analysis

The headers are the flight recorder of the message. This chapter teaches you to read them line by line, and to spot the three mismatches that expose most phishing.

PHASE 3 **Email Header Analysis**

OBJECTIVE

Establish the message's **true origin**: the IP that handed it to your gateway, the path it took, the real envelope sender, and any inconsistency between what the user sees and what actually happened.

WHY THIS STEP IS IMPORTANT

The body is what the attacker *wants* you to look at. The headers are what the infrastructure *recorded*. Body text can be perfect; headers are much harder to fake completely. Header analysis is also the cheapest phase — no detonation, no sandbox, no risk — and it very often produces the verdict on its own.

EXPECTED FINDINGS

Originating IP and its owner, the hop chain, envelope-vs-header sender mismatch, Reply-To divergence, suspicious mailer strings, and timestamp anomalies.

7.1 The headers that matter (and what each one really means)

HEADER	WHAT IT IS	WHY AN INVESTIGATOR CARES
`Received:`	One line added by each MTA that handled the message. Bottom = earliest.	Reconstructs the route. The lowest *trustworthy* line reveals the true sending IP.
`Return-Path:`	The envelope sender (`MAIL FROM` , P1).	This — not `From:` — is what SPF checks. Bounces go here.
`From:`	The display sender (P2). Attacker-controlled.	What the user sees. Compare against `Return-Path` for spoofing.
`Reply-To:`	Where a reply is actually sent.	A classic BEC tell: `From:` looks internal, `Reply-To:` is a free webmail account.
`Authentication-Results:`	Your gateway's SPF/DKIM/DMARC verdicts.	The single most information-dense line in the header. See Chapter 8.
`Message-ID:`	Unique ID assigned by the originating system.	The domain in it should match the sending infrastructure. Random or mismatched IDs suggest a bulk-mailing script.
`X-Originating-IP:`	Sometimes added by webmail providers.	Can reveal the client IP behind a compromised webmail account.

HEADER	WHAT IT IS	WHY AN INVESTIGATOR CARES
`X-Mailer / User-Agent:`	The sending software.	PHPMailer, Python scripts and unusual mailers on a "corporate" email are a strong signal.
`X-Forefront-Antispam-Report`	Microsoft's scoring detail (SCL, SFV, CAT, CIP).	Tells you what EOP thought and, crucially, whether an override applied.
`X-MS-Exchange-Organization-AuthAs`	Whether the message was treated as internal/authenticated.	"Internal" on an externally-sourced mail is a serious finding.
`Date:`	Sender-claimed send time.	Compare with the Received timestamps. Impossible times = forged header.

7.2 Anatomy of a real (defanged) header

sample.eml | annotated header block — read the Received lines from the bottom up

```
# --- HOP 3 (most recent): your Exchange Online tenant. TRUSTED.
Received: from DB7PR04MB4321.eurprd04.prod.outlook.com (2603:10a6:5:1e::15)
 by DB9PR04MB8123.eurprd04.prod.outlook.com with HTTPS; Mon, 12 May 2025 09:12:41 +0000

# --- HOP 2: Microsoft's edge accepting the mail from the internet. TRUSTED.
#     >>> THIS LINE IS THE TRUST BOUNDARY. Everything below it can be forged. <<<
Received: from mail.contoso-invoices.com (185.220.101.47) by
 DB7PR04MB4321.eurprd04.prod.outlook.com (10.172.55.12) with Microsoft SMTP Server
 (TLS 1.2) id 15.20.7587.24; Mon, 12 May 2025 09:12:39 +0000

# --- HOP 1 (claimed): written by the attacker's own server. UNTRUSTED.
Received: from localhost (unknown [10.0.0.5]) by mail.contoso-invoices.com
 (Postfix) with ESMTP id 4F2A1; Mon, 12 May 2025 09:12:37 +0000

Authentication-Results: spf=pass (sender IP is 185.220.101.47)
 smtp.mailfrom=contoso-invoices.com; dkim=pass (signature was verified)
 header.d=contoso-invoices.com; dmarc=pass action=none
 header.from=contoso-invoices.com; compauth=pass reason=100

Return-Path: bounce@contoso-invoices.com
From: "Contoso Billing" <billing@contoso-invoices.com>
Reply-To: accounts.receivable.contoso@gmail.com
To: ap-team@example.com
Subject: Invoice 44821 - overdue, action required
Date: Mon, 12 May 2025 09:12:35 +0000
Message-ID: <20250512091235.4F2A1@mail.contoso-invoices.com>
X-Mailer: PHPMailer 6.8.0 (https://github.com/PHPMailer/PHPMailer)
X-Forefront-Antispam-Report: CIP:185.220.101.47;CTRY:NL;SFV:NSPM;SCL:1;
 CAT:NONE;SFTY:;DIR:INB
```

What this header actually tells you

True sending IP — `185.220.101[.]47` — taken from the *lowest trusted* Received line, the one written by Microsoft's edge. The `10.0.0.5` below it is attacker-written fiction.

Authentication	**All three pass.** Read that again — SPF, DKIM and DMARC all pass. That is *not* a sign of legitimacy. The attacker owns `contoso-invoices[.]com` and configured it properly. See the warning below.
Reply-To divergence	**Red flag** The message claims to be corporate billing, but replies go to a **Gmail address**. Legitimate billing systems do not do this.
Lookalike domain	**Red flag** The real company is `contoso[.]com`. This is `contoso-invoices[.]com` — a cousin domain. Check its registration date (Chapter 12); a domain registered nine days ago is decisive.
X-Mailer	**Red flag** `PHPMailer`. A real enterprise billing platform sends through a commercial ESP, not a PHP script on a VPS.
Sending country	`CTRY:NL` from the antispam report — a Dutch hosting IP for a supplier that operates only in one region. Weak on its own, corroborating in combination.

RED FLAG — "DMARC PASSED, SO IT'S LEGITIMATE" IS THE #1 BEGINNER ERROR

DMARC answers exactly one question: *"is this sender authorised to send on behalf of the domain in the* `From:` *header?"* If the attacker **owns** the domain in the `From:` header, then yes, they are authorised — and DMARC passes, correctly. Authentication proves **authorisation**, never **trustworthiness**. The question you must ask next is: *is that domain the one it is pretending to be?*

7.3 The three mismatches that expose most phishing

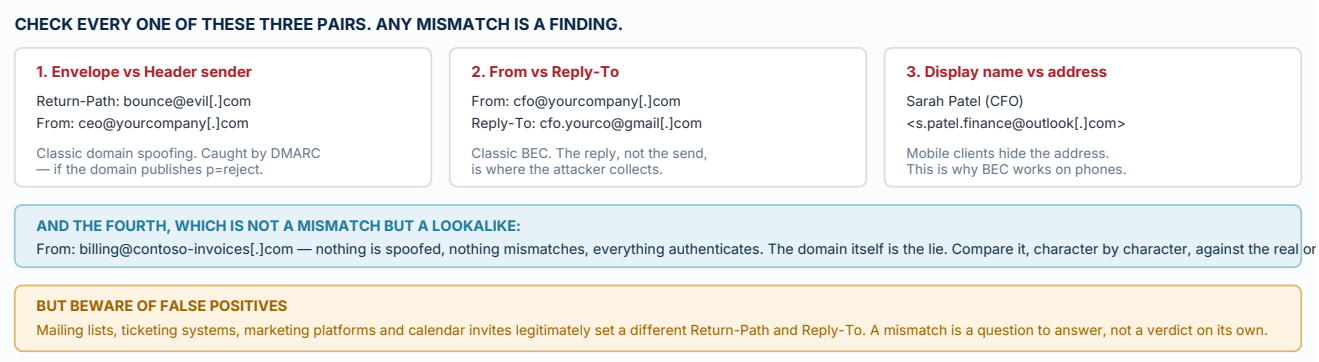

Figure 7.1 — The mismatch triangle. Three comparisons take thirty seconds and resolve a large share of phishing tickets. The fourth case — the lookalike domain — produces no mismatch at all, which is precisely why it is the hardest and most dangerous.

7.4 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Paste headers into an analyser; check the mismatch triangle; look up the sending IP.	MXToolbox Header Analyzer, Microsoft Message Header Analyzer, Google Admin Toolbox
SOC L2	Manually trace the hop chain; identify the trust boundary; check X-headers and mailer strings.	Text editor, `mha`, PowerShell, `dig`
Threat Hunter	Pivot on the sending IP, ASN and `Message-ID` pattern across the estate to cluster the campaign.	Advanced Hunting, Sentinel, WHOIS/ASN lookups
DFIR	Build a precise timeline from the Received timestamps; identify forged or impossible hops.	Timeline tooling, `eml_parser`

7.5 MITRE ATT&CK mapping

TECHNIQUE	NAME	WHAT THE HEADER EVIDENCE SHOWS
`T1585.002`	Establish Accounts: Email Accounts	Attacker registered their own domain and mailbox (the lookalike case).
`T1586.002`	Compromise Accounts: Email Accounts	Mail sent from a genuine but compromised third-party mailbox — authentication passes perfectly.
`T1583.001`	Acquire Infrastructure: Domains	Newly registered sending or landing domain.
`T1656`	Impersonation	Display-name spoofing and cousin domains.
`T1534`	Internal Spearphishing	Header shows an internal, authenticated sender — i.e. the account is already compromised.

7.6 IOC extraction

- **Sending IP** (from the lowest trusted Received line) → block candidate, ASN pivot.
- **Sending domain** and **HELO/EHLO name** → block candidate, WHOIS pivot.
- **Envelope sender, header sender, Reply-To** → all three, separately.
- **Message-ID** and its domain part → campaign clustering (attackers reuse the ID format).
- **X-Mailer string** → a surprisingly durable campaign fingerprint.

7.7 KQL for header analysis

```
Defender XDR  |  Find every message from the same sending IP
```

```
// The IP is a much stronger pivot than the sender address, which changes per-message.
EmailEvents
| where Timestamp > ago(30d)
| where SenderIPv4 == "185.220.101.47"
| summarize Messages = count(),
            Senders = make_set(SenderFromAddress, 20),
            Subjects = make_set(Subject, 20),
            Recipients = dcount(RecipientEmailAddress),
            Delivered = countif(DeliveryLocation == "Inbox/folder")
  by SenderIPv4, bin(Timestamp, 1d)
| sort by Timestamp desc
```

Defender XDR | Envelope-sender vs header-sender mismatch across the tenant

```
// Hunts for spoofing structurally, not by IOC. Expect legitimate hits from
// mailing lists and ESPs - baseline them out with the exclusion list.
EmailEvents
| where Timestamp > ago(7d)
| where EmailDirection == "Inbound"
| extend HeaderDomain   = tostring(split(SenderFromAddress, "@")[1])
| extend EnvelopeDomain = tostring(split(SenderMailFromAddress, "@")[1])
| where isnotempty(EnvelopeDomain) and HeaderDomain =~ EnvelopeDomain
| where HeaderDomain !in ("mailchimp.com", "salesforce.com", "sendgrid.net")   // baseline
| summarize Messages = count(), Recipients = dcount(RecipientEmailAddress),
            Subjects = make_set(Subject, 5)
  by HeaderDomain, EnvelopeDomain, DeliveryAction
| sort by Messages desc
```

Defender XDR | Display-name impersonation of your own executives

```
// Any EXTERNAL mail whose display name matches a VIP name is worth a look.
let vips = dynamic(["Sarah Patel", "James Okoro", "Priya Raman"]);
EmailEvents
| where Timestamp > ago(14d)
| where EmailDirection == "Inbound"
| where SenderDisplayName has_any (vips)
| where not (SenderFromAddress endswith "@example.com")     // your real domain
| project Timestamp, SenderDisplayName, SenderFromAddress, SenderIPv4,
          RecipientEmailAddress, Subject, DeliveryAction, DeliveryLocation
| sort by Timestamp desc
```

7.8 Decision point

HEADER FINDING	WEIGHT	INTERPRETATION
Header `From:` domain is yours, but the mail arrived from an external IP and DMARC failed	**Decisive**	Domain spoofing. Malicious.
Internal-looking sender, authenticated, sent from an internal IP	**Decisive**	Account compromise — go straight to Chapter 14.
Reply-To on free webmail while `From:` claims a company	**Strong**	BEC pattern.
Cousin/lookalike sending domain	**Strong**	Confirm registration age in Chapter 12; usually decisive together.
Script-based X-Mailer on "corporate" mail	**Moderate**	Corroborating, not decisive.
All checks pass, sender is a long-established, correct domain	**Weak-negative**	Not proof of safety. Continue to content and URL analysis — the sender may be a compromised partner.

7.9 How this shapes the next step

Header analysis gives you the *who* and the *where from*. The next question — *was that sender allowed to use that domain?* — is answered by the `Authentication-Results` line, which deserves a chapter of its own. If the header work has already proved the sender is impersonating an internal user, jump straight to Chapter 14 in parallel: you may be dealing with a compromised account, not an inbound phish.

CHAPTER 8 · PHASE 4

Email Authentication: SPF, DKIM and DMARC

The three checks every analyst quotes and many misread. What each one actually proves, what it cannot prove, and how to interpret every combination of results.

PHASE 4 **Email Authentication Analysis**

OBJECTIVE

Determine whether the sending server was **authorised** to send mail using the domain in the `From:` header — and understand precisely what that authorisation does and does not tell you.

WHY THIS STEP IS IMPORTANT

Authentication is the only part of email that is cryptographically or DNS-verifiable. It cleanly separates *domain spoofing* (attacker pretends to be your domain) from *domain impersonation* (attacker uses a lookalike domain they legitimately own). These require completely different responses: the first is a DMARC policy problem, the second is a detection and awareness problem.

EXPECTED FINDINGS

SPF, DKIM and DMARC results with their alignment status; the `compauth` verdict from Microsoft; and a clear classification of the spoofing technique in use.

8.1 The three checks in plain English

SPF Sender Policy Framework	**"Is this IP allowed to send for this domain?"** The domain publishes a DNS TXT record listing its authorised sending IPs. The receiver checks the connecting IP against it. *Checks the envelope sender (`Return-Path`), not the `From:` the user sees.* **Weakness** Breaks on forwarding, and says nothing about the visible sender.
DKIM DomainKeys Identified Mail	**"Was this message signed by the domain, and is it unmodified?"** The sender signs selected headers and the body with a private key; the receiver verifies with the public key in DNS. *Survives forwarding.* **Weakness** A valid signature from `evil[.]com` is still a valid signature — it proves integrity, not intent.
DMARC Domain-based Message Authentication	**"Do SPF or DKIM *align* with the domain the user actually sees?"** This is the one that connects the plumbing to the human. It requires SPF or DKIM to pass *and* the passing domain to match the `From:` domain. It also tells the receiver what to do on failure: `p=none` (monitor), `p=quarantine`, or `p=reject`.

CONCEPT — ALIGNMENT IS THE WHOLE POINT

An attacker can trivially pass SPF: they send from their own server, using their own domain in the envelope, and their own SPF record authorises it. SPF says "pass". Meanwhile the `From:` header says `ceo@yourcompany.com`. **DMARC alignment is what catches this**: it demands that the domain which passed SPF or DKIM be the *same domain* the user sees in `From:`. Without alignment, "SPF: pass" is close to meaningless as a trust signal.

GOOD TO KNOW — THE DMARC SPECIFICATION WAS UPDATED IN 2026

DMARC ran for over a decade on `RFC 7489` (2015, Informational status). In May 2026, the IETF published `RFC 9989` (core protocol), `RFC 9990` (aggregate reporting) and `RFC 9991` (failure reporting) — collectively "DMARCbis" — which obsolete RFC 7489 and move DMARC to formal IETF Standards-Track status for the first time. Existing `v=DMARC1` DNS records remain valid and every concept in this chapter is unchanged; the update mainly replaces the old Public Suffix List lookup with a DNS "tree walk" for finding the organisational domain, and adds a few optional record tags. Nothing here requires you to change how you investigate a message.

8.2 The decision logic, as your gateway runs it

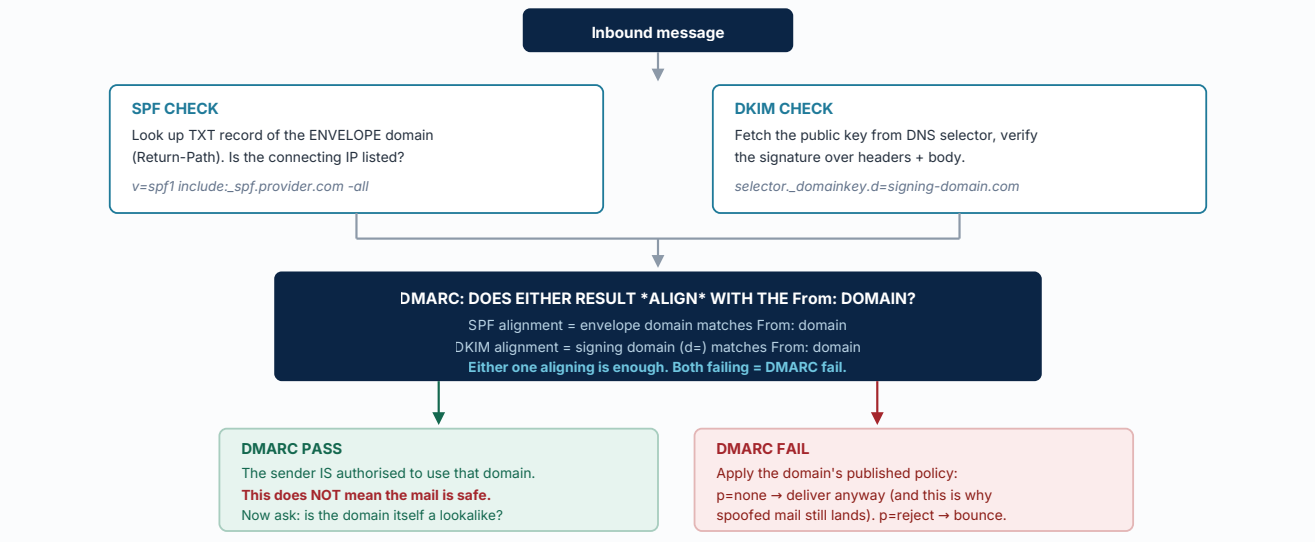

Figure 8.1 — SPF, DKIM and DMARC decision logic. Note the two asymmetric outcomes: a *fail* is strong evidence of spoofing, but a *pass* is not evidence of safety. Analysts who treat pass as "clean" will wave through every lookalike-domain attack that exists.

8.3 Reading the Authentication-Results line

Header excerpt | every field decoded

```
Authentication-Results: spf=fail (sender IP is 203.0.113.88)
 smtp.mailfrom=mailer-x9[.]net; dkim=none (message not signed)
 header.d=none; dmarc=fail action=quarantine
 header.from=example.com; compauth=fail reason=001
```

spf=fail The IP `203.0.113[.]88` is not authorised to send for `mailer-x9[.]net`.

smtp.mailfrom	The **envelope** domain — here `mailer-x9[.]net`, which is *not* the domain the user sees.
dkim=none	Not signed at all. Legitimate corporate senders almost always sign.
header.from	`example.com` — **your own domain**. The attacker is spoofing you, to you.
dmarc=fail	Nothing aligns with `example.com`. This is domain spoofing, full stop.
compauth=fail reason=001	Microsoft's composite authentication also failed. The `reason` codes are worth knowing: `0xx` = explicit failure, `1xx` = pass, `2xx` = implicit pass, `3xx` = not checked, `4xx` / `5xx` = pass due to an override or trusted-sender configuration.

WARNING — `COMPAUTH` CODES IN THE 4XX/5XX RANGE

A `compauth=pass reason=4xx` or `reason=5xx` often means the message passed *because someone configured an override* — an allowed sender, a partner exception, an IP allow-list. If you see this on a malicious message, you have found the hole. Document it and fix it: that override is a standing invitation.

8.4 The interpretation matrix

SPF	DKIM	DMARC	WHAT IT MEANS	YOUR MOVE
pass	**pass**	**pass**	Sender is authorised for the `From:` domain. **Says nothing about safety.**	Check whether the domain is a lookalike, or a genuine but compromised partner. Continue to content/URL analysis.
fail	**fail**	**fail**	Nothing authorises this sender. Classic domain spoofing.	Malicious with high confidence. Purge, block, and check why `p=` did not stop it.
fail	**pass**	**pass**	Very often a **legitimate forwarded message** or a mailing list — SPF breaks on forwarding, DKIM survives.	Do not treat as malicious on this basis alone. Look at content.
pass	**none**	**fail**	SPF passed for the envelope domain, but that domain does not align with the visible `From:`. Textbook alignment failure.	Strong spoofing signal. Investigate fully.
softfail	**none**	**fail**	The domain owner publishes `~all` and is not enforcing. Common on poorly configured legitimate domains *and* on spoofing.	Ambiguous. Weight the content and URL evidence heavily.
none	**none**	**none**	The sending domain publishes no email authentication at all.	Suspicious for any business sender in 2025. Treat as elevated risk.
pass	**pass**	**pass**	*…and the sender is a real, long-standing partner domain*, but the content is a bank-detail change.	**Suspect vendor compromise. Verify by phone, using a number you already hold — never one from the email.**

8.5 Techniques, tools and verification

Do not just read the gateway's verdict — **verify the DNS yourself**. It takes twenty seconds and it tells you things the verdict does not, such as whether your own domain is enforcing DMARC.

Analysis VM | Verify the records yourself

```
# SPF record of the sending domain
dig +short TXT contoso-invoices.com | grep spf1

# DMARC policy - note p= and pct=
dig +short TXT _dmarc.contoso-invoices.com

# DKIM public key for the selector seen in the header (s=selector1)
dig +short TXT selector1._domainkey.contoso-invoices.com

# And check YOUR OWN domain's enforcement - the answer is often uncomfortable
dig +short TXT _dmarc.example.com
```

ROLE	TECHNIQUE	TOOLS
SOC L1	Read `Authentication-Results`; classify with the matrix above.	MXToolbox (SPF/DKIM/DMARC lookups), Defender email entity page
SOC L2	Verify the DNS records directly; check `compauth` reason codes for overrides.	`dig` / `nslookup`, dmarcian, EasyDMARC
Security Engineer	Review your own SPF/DKIM/DMARC posture; move toward `p=reject`; check the SPF 10-lookup limit.	DMARC aggregate (RUA) reports, MXToolbox, Microsoft 365 Defender policies
Threat Hunter	Hunt for inbound mail claiming your domain that fails DMARC — a continuous spoofing detector.	Advanced Hunting, Sentinel

8.6 MITRE ATT&CK mapping

TECHNIQUE	NAME	EVIDENCE
`T1656`	Impersonation	Cousin domain passing its own DMARC.
`T1566.002`	Phishing: Spearphishing Link	The delivery vehicle whose authenticity you are testing.
`T1585.001`	Establish Accounts: Social Media / Email	Attacker-owned domain with correct SPF/DKIM published.
`T1598`	Phishing for Information	Reconnaissance mail that carries no payload but tests whether spoofing lands.

8.7 KQL for authentication analysis

Defender XDR | Inbound mail spoofing YOUR domain and failing DMARC

```
// This is the highest-value standing hunt in this entire chapter.
// Any inbound message claiming to be from you, that fails DMARC, is a spoof attempt.
EmailEvents
| where Timestamp > ago(14d)
| where EmailDirection == "Inbound"
| where SenderFromDomain =~ "example.com" // your own domain
| extend Auth = parse_json(AuthenticationDetails)
| extend SPF = tostring(Auth.SPF), DKIM = tostring(Auth.DKIM),
         DMARC = tostring(Auth.DMARC), CompAuth = tostring(Auth.CompAuth)
| where DMARC in~ ("fail", "none")
| project Timestamp, SenderFromAddress, SenderMailFromAddress, SenderIPv4,
          RecipientEmailAddress, Subject, SPF, DKIM, DMARC, CompAuth,
          DeliveryAction, DeliveryLocation
| sort by Timestamp desc
```

Defender XDR | Malicious mail that was DELIVERED despite failing authentication

```
// If this returns rows, you have a policy or override problem, not just a phishing problem.
EmailEvents
| where Timestamp > ago(30d)
| where EmailDirection == "Inbound"
| where DeliveryLocation == "Inbox/folder"
| extend Auth = parse_json(AuthenticationDetails)
| extend DMARC = tostring(Auth.DMARC), CompAuth = tostring(Auth.CompAuth)
| where DMARC =~ "fail"
| summarize Delivered = count(),
            Recipients = dcount(RecipientEmailAddress),
            Domains = make_set(SenderFromDomain, 25)
  by CompAuth, bin(Timestamp, 1d)
| sort by Delivered desc
```

Sentinel | Newly-seen sending domains that pass DMARC (the lookalike hunt)

```
// Attackers who register their own domain PASS authentication. So hunt for
// domains that are BOTH authenticated AND never seen before in your tenant.
let knownDomains = EmailEvents
    | where Timestamp between (ago(90d) .. ago(7d))
    | where EmailDirection == "Inbound"
    | distinct SenderFromDomain;
EmailEvents
| where Timestamp > ago(7d)
| where EmailDirection == "Inbound"
| where SenderFromDomain !in (knownDomains)
| extend Auth = parse_json(AuthenticationDetails)
| where tostring(Auth.DMARC) =~ "pass"
| summarize Messages = count(), Recipients = dcount(RecipientEmailAddress),
            Subjects = make_set(Subject, 5), FirstSeen = min(Timestamp)
  by SenderFromDomain
| where Messages ⩾ 1
| sort by FirstSeen desc
```

8.8 Decision point

Authentication analysis produces one of exactly three classifications. Everything downstream depends on which one you land on.

CLASSIFICATION	SIGNATURE	WHAT IT MEANS FOR THE RESPONSE
Domain spoofing	DMARC fail on *your* domain, or on a well-known brand's domain.	High-confidence malicious. Also raises a policy question: why is `p=` not enforcing?
Domain impersonation (lookalike)	DMARC passes — for an attacker-owned cousin domain.	Authentication will never catch this. Your controls are content analysis, domain-age intel and user awareness.
Legitimate-sender compromise	Everything passes, the domain is genuinely the partner's, and the mail is real — but the content is fraudulent.	The hardest case. Verify out-of-band by phone, and notify the partner: *they* have an incident.

8.9 How this shapes the next step

You now know whether the sender is who they claim to be. What you still do not know is **what the message is asking the recipient to do**. A perfectly authenticated email from a compromised supplier is more dangerous than a crude spoof, precisely because every technical check passes. That is why the next phase — content and social-engineering analysis — is not optional, even when authentication looks clean.

CHAPTER 9 · PHASE 5

Content and Social Engineering Analysis

Once you know who sent it, read the message the way the victim did — then read it again the way an analyst must: for the ask, the lever, and the tells that survive perfect grammar.

PHASE 5 **Content and Social Engineering Analysis**

OBJECTIVE

Identify the **pretext** (the cover story), the **psychological lever** being pulled, and the exact **action** the message wants the recipient to take — then check the message structure for inconsistencies a genuine sender would not produce.

WHY THIS STEP IS IMPORTANT

Every technical channel can be perfectly clean — a genuinely compromised supplier mailbox, correctly authenticated — while the content is one hundred percent fraudulent. Content is the *only* durable evidence against BEC, callback phishing and other payload-free attacks, because there is no URL or hash to analyse. Content analysis is also where a great deal of it goes wrong: "bad grammar" used to be a reliable tell, and it no longer is.

EXPECTED FINDINGS

The requested action, the lever used, a brand-consistency assessment, any hidden or mismatched links in the HTML source, mail-merge leakage, tracking pixels, and any phone numbers or payment details in the body.

9.1 The anatomy of a lure

Every social-engineering message is built from three layers. Naming them out loud makes an email much faster to assess than reading it as a whole.

Pretext	The story: an invoice, a document share, a password expiry, a delivery failure, a job offer.
Lever	The emotion doing the work: urgency, authority, fear, curiosity, greed, or helpfulness.
Ask	The single action wanted: click, open, reply, call, approve, or pay.

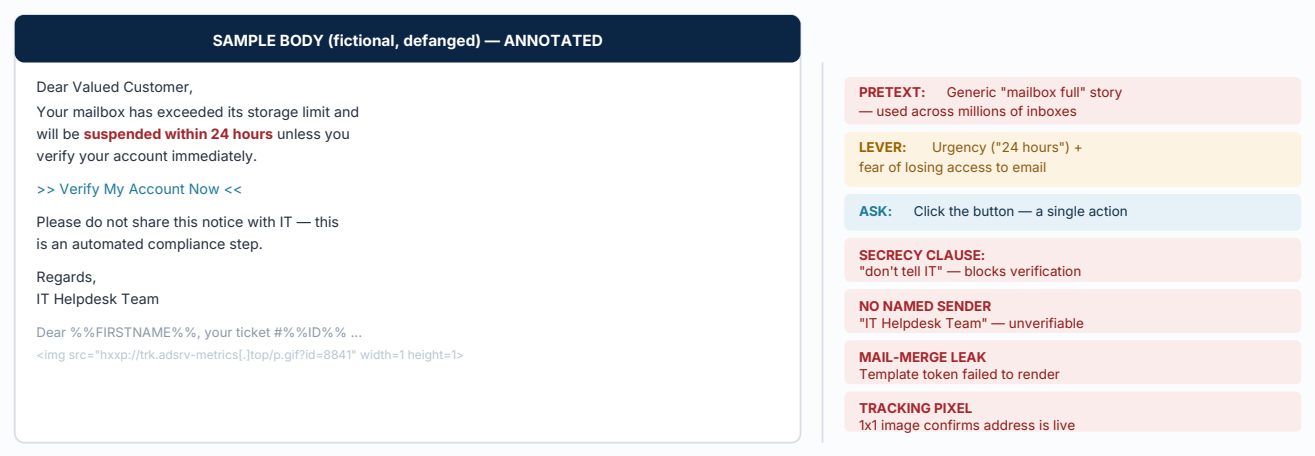

Figure 9.1 — Anatomy of a lure. None of these tells require any technical tooling to spot — they require reading the source, not the rendered preview, and reading it slowly.

Common pretexts, the lever each relies on, and the giveaway

PRETEXT	LEVER	TYPICAL ASK	GIVEAWAY TO CHECK
Invoice / payment overdue	Urgency, authority	Open attachment or click to "view invoice"	Vendor never billed this way before; amount or account is new.
Password / mailbox expiry	Fear	Click to "verify" or "reactivate"	Real IT never asks you to confirm a password via email link.
Shared document	Curiosity, trust in platform	Click to "view" via a cloud storage link	Sharing notification from a platform you don't use with that contact.
Delivery failure	Urgency	Click to "reschedule" or pay a customs fee	You weren't expecting a parcel; fee requests are a strong tell.
Executive request	Authority, urgency, secrecy	Buy gift cards, change bank details, wire funds	Off-channel, unusual for that person, explicit "keep this confidential".
Job offer / bonus letter	Greed, curiosity	Open attachment for "details"	Unsolicited, generic greeting, too good to be true.
Subscription cancellation (TOAD)	Fear of a charge	Call a phone number	No URL at all — the entire attack continues by voice.

9.2 Reading the source, not the render

A rendered email shows you what the attacker wants you to see. The underlying HTML shows you what is actually there. Always inspect the source — in an isolated VM, never on a production endpoint.

- **Anchor text vs. destination.** The visible text may read `www.microsoft.com` while the `href` attribute points somewhere else entirely. This is the single most useful thing source view gives you.
- **Hidden or white-on-white text.** Attackers stuff invisible keywords to confuse content filters, or to defeat "read the visible text" detection logic.

- **Tracking pixels.** A 1×1 image confirms your address is live and monitored — valuable reconnaissance for the attacker even if you never click anything else.
- **Mail-merge leakage.** Failed template tokens such as `%%FIRSTNAME%%` or `{{company}}` prove the message was mass-produced, whatever the personalised-looking parts claim.

Analysis VM | Pull every link and compare anchor text to destination

```
python3 -c "
import re, html
body = open('body.html', encoding='utf-8', errors='ignore').read()
for m in re.finditer(r'<a[^>]+href=[\"\\']([^\"\\']+)[\"\\'][^>]*>(.*?)</a>', body, re.I|re.S):
    href, text = m.group(1), re.sub('<[^>]+>', '', m.group(2)).strip()
    print(f'ANCHOR TEXT: {html.unescape(text)[:60]!r:65} →  HREF: {href}')
"
```

WARNING — GRAMMAR IS NO LONGER A RELIABLE SIGNAL

Attackers now routinely use AI writing tools, which means spelling and grammar can be flawless while the pretext, the lever and the ask remain completely fraudulent. Teach yourself — and teach the users you support — to check the *ask* and the *channel*, not the prose quality.

9.3 Brand and language forensics

CHECK	WHAT A MISMATCH TELLS YOU
Logo resolution and placement	Screenshotted or low-resolution logos suggest the template was copied, not sourced from the real brand kit.
Footer / unsubscribe link domain	Legitimate marketing platforms link unsubscribe to their own ESP domain, consistently. A footer linking to a freshly registered domain is a strong tell.
Currency, date format, and legal boilerplate	A "US" company invoicing in an unexpected currency, or a date format inconsistent with the claimed sender region.
Tone consistency	Real correspondence with a known contact has a tone. Sudden formality, or sudden pressure, breaks the pattern.

9.4 The BEC-specific checklist — when there is no payload at all

When the message carries no URL and no attachment, content analysis *is* the entire investigation. Run through this list explicitly:

- [ ] Is this request **unusual for this process** — a new bank account, a new vendor, an off-cycle payment?
- [ ] Does it demand **secrecy or urgency** that bypasses normal approval?
- [ ] Does it request a channel switch — "let's continue by text" or "call me on this new number"?
- [ ] Is the requester's **usual writing style** different (short, clipped messages from someone normally detailed, or vice versa)?

- [ ] Would a **phone call to a number you already have on file** (never one from the email) resolve this in thirty seconds?

WORKED EXAMPLE — FICTIONAL BEC THREAD, PARAPHRASED

An email arrives appearing to be from the CFO, sent while she is "in back-to-back board meetings," asking Accounts Payable to process an urgent supplier payment to a new bank account and to keep the change confidential until she can brief the board herself. The account is genuinely authenticated — the attacker is using a compromised personal webmail the CFO also uses for travel bookings, not a spoofed corporate domain. Every technical check in Chapters 7–8 comes back clean. Only the ask — a new account, secrecy, urgency, an out-of-band request — identifies this as fraud.

9.5 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Read for pretext, lever and ask; apply the BEC checklist when there is no payload.	Defender email entity preview (safe rendering)
SOC L2	View HTML source; extract anchor/href mismatches, tracking pixels, hidden text.	CyberChef, isolated VM, Python `email` / regex
Threat Hunter	Cluster campaigns by subject-line and body similarity, even across sender-domain changes.	Advanced Hunting, fuzzy hashing (ssdeep)
DFIR	Correlate pretext against known campaign templates and prior incidents.	Internal case repository, MISP

9.6 MITRE ATT&CK mapping

TECHNIQUE	NAME	RELEVANCE
`T1566`	Phishing	The overall delivery technique the content is designed to serve.
`T1598`	Phishing for Information	Reconnaissance messages and tracking pixels that harvest engagement data rather than credentials.
`T1656`	Impersonation	Executive or brand impersonation in the pretext.
`T1204`	User Execution	The ultimate goal of the ask — getting the human to act.

9.7 IOC extraction

- Exact subject line (and any templated / sequential ID pattern within it)
- Tracking pixel URL and its hosting domain
- Any phone numbers (critical for callback / TOAD tracking)
- Any bank account numbers, beneficiary names or payment references quoted in a BEC attempt
- Fuzzy hash (ssdeep) of the body, for clustering near-identical campaigns

9.8 KQL for content analysis

Defender XDR | Cluster near-duplicate subject lines across the tenant

```
// Microsoft groups related messages under EmailClusterId - use it directly.
EmailEvents
| where Timestamp > ago(7d)
| where isnotempty(EmailClusterId)
| summarize Messages = count(), Recipients = dcount(RecipientEmailAddress),
            Senders = make_set(SenderFromDomain, 10), Subjects = make_set(Subject, 5)
  by EmailClusterId
| where Messages ⩾ 5
| sort by Messages desc
```

Defender XDR | Confidence and bulk-complaint signals for payload-free mail

```
// Useful when there is no URL/attachment to analyse - lean on Microsoft's own scoring.
EmailEvents
| where Timestamp > ago(7d)
| where UrlCount == 0 and AttachmentCount == 0
| where EmailDirection == "Inbound"
| where ConfidenceLevel has "High" or ThreatTypes has "Phish"
| project Timestamp, SenderFromAddress, SenderDisplayName, RecipientEmailAddress,
          Subject, ThreatTypes, ConfidenceLevel, DeliveryAction, DeliveryLocation
| sort by Timestamp desc
```

9.9 Decision point

CONTENT FINDING	WEIGHT	INTERPRETATION
Secrecy clause + urgency + novel payment detail	**Decisive**	BEC. Malicious regardless of clean authentication.
Anchor text names a trusted brand, href points elsewhere	**Strong**	Deceptive linking — proceed to URL analysis.
Mail-merge token leaked, generic greeting	**Moderate**	Mass campaign, not targeted — still likely malicious, lower sophistication.
Tone and process match known legitimate correspondence	**Weak-negative**	Supportive of legitimacy, not proof — confirm via authentication and, if money is involved, a phone call.

9.10 How this shapes the next step

Content analysis tells you *what kind* of payload to expect. A "shared document" pretext sends you to Chapter 10 for URL analysis. An "invoice attached" pretext sends you to Chapter 11 for attachment analysis. A pure BEC message with neither sends you directly to Chapter 13 to scope who else received it, and then to the verdict in Chapter 15 — content evidence alone is often sufficient to close a payload-free case.

CHAPTER 10 · PHASE 6

URL and Link Analysis

Where does the link actually go, how many hops does it take to get there, and what is waiting at the end — a credential form, a download, or an invisible proxy stealing the session?

PHASE 6 **URL and Link Analysis**

OBJECTIVE

Trace every URL in the message to its true final destination, classify what that destination does, and determine whether the mechanism is simple credential harvesting or something more dangerous, such as an Adversary-in-the-Middle (AiTM) proxy that steals a live session.

WHY THIS STEP IS IMPORTANT

The URL in the email is frequently **not** the URL that harms the user. Redirect chains, open-redirect abuse, and time-delayed activation all mean the link you see at delivery time may look completely benign. A "clean" scan of the first hop proves nothing about the last one.

EXPECTED FINDINGS

The complete redirect chain, the final landing domain and its hosting, a content classification (credential harvest, malware download, benign decoy, or AiTM proxy), and confirmation of any personalisation or cloaking.

10.1 How to trace a link safely

1. **Defang immediately** when you extract it: `hxxps://evil-domain[.]com/path`.
2. **Never click the live link.** Use a URL-scanning service or an isolated, disposable VM with no corporate identity signed in.
3. **Submit to URLScan.io** for a screenshot, the DOM, the full request/response chain and the final resolved IP — without your browser ever touching attacker infrastructure directly.
4. **Check VirusTotal** for the URL and the final domain — but treat a clean result as *inconclusive*, not reassuring (see Chapter 3.3).
5. **Follow the chain manually** if the scanner shows a gate (CAPTCHA, geo-block): note where it stops and try again from an appropriate vantage point if you have one (e.g. a sandboxed browser configured for the target region).
6. **Decode any QR code** in an image or PDF; the entire URL may exist only as pixels.
7. **Check for personalisation** — strip query parameters and reload; if the kit refuses without them, the victim's email address is doing the personalising, and this is itself an evasion technique.

```
Analysis VM  |  Decode a QR code from an attached image without opening it in a viewer
```

```
pip install pyzbar pillow --break-system-packages
python3 -c "
from pyzbar.pyzbar import decode
from PIL import Image
for r in decode(Image.open('attach_qr.png')):
    print(r.type, r.data.decode(errors='replace'))
"
```

10.2 The redirect chain, and how AiTM changes the picture

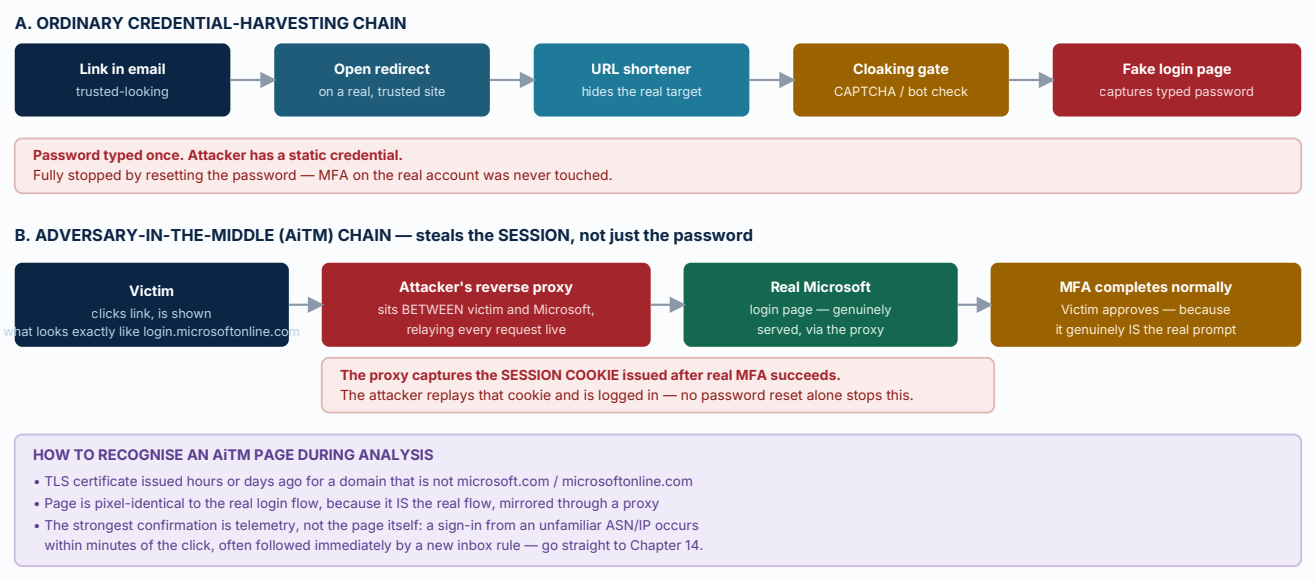

Figure 10.1 — Ordinary credential harvesting vs. AiTM session theft. The visual difference on screen can be zero. The difference that matters is what happens to the session token afterwards — which is why URL analysis and identity analysis (Chapter 14) must always be read together for any confirmed click.

10.3 Classifying the landing page

LANDING PAGE TYPE	WHAT YOU'LL SEE	VERDICT WEIGHT
Credential harvest form	A login-style form on a non-corporate domain, often a copy of a real portal.	**Decisive**
AiTM proxy	Pixel-perfect real login flow through an attacker-controlled reverse proxy domain.	**Decisive — escalate to Ch.14 immediately**
Malware download	Direct file download, or a page that triggers one automatically.	**Decisive — proceed to Ch.11**
Consent phishing page	A genuine Microsoft/Google OAuth consent screen for a suspicious third-party app.	**Decisive**
Parked / dead domain	Registrar parking page, "domain for sale," or a 404.	**Inconclusive — kit may have been taken down or is cloaked**

LANDING PAGE TYPE	WHAT YOU'LL SEE	VERDICT WEIGHT
Genuinely benign redirect	Marketing tracker leading to a real, expected destination.	**Weak-negative**

10.4 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Defang, submit to a URL scanner, read the screenshot and verdict.	URLScan.io, VirusTotal, Microsoft Defender Safe Links reputation
SOC L2	Follow multi-hop chains; test with/without personalisation parameters; capture the full request chain.	URLScan.io (full API), Any.Run interactive browser, CheckPhish
Threat Hunter	Pivot on favicon hash, TLS certificate issuer/ serial, and page template similarity to find sibling kits.	URLScan.io search, Censys/Shodan-style certificate search, VirusTotal Relations
DFIR / IR	Correlate a confirmed click with sign-in telemetry within the same time window to confirm or rule out AiTM.	Entra ID sign-in logs, Advanced Hunting

10.5 MITRE ATT&CK mapping

TECHNIQUE	NAME	EVIDENCE
`T1566.002`	Spearphishing Link	The delivery mechanism.
`T1204.001`	User Execution: Malicious Link	The user clicking through.
`T1539`	Steal Web Session Cookie	Confirmed AiTM proxy behaviour.
`T1656`	Impersonation	Pixel-identical clone of a real login portal.
`T1189`	Drive-by Compromise	Landing page delivers an exploit or auto-download rather than a form.

10.6 IOC extraction

- Original URL exactly as it appeared in the message
- Every hop in the redirect chain, in order
- Final landing domain, its resolved IP, and hosting provider/ASN
- TLS certificate issuer, serial number, and issuance date
- Screenshot hash and, where available, the page's favicon hash
- Whether the URL was personalised (victim address in path/query/fragment)

10.7 KQL for URL analysis

Defender XDR | Every URL carried by this message, and where it sits in the body

```
EmailUrlInfo
| where NetworkMessageId == "<paste-from-EmailEvents>"
| project Url, UrlDomain, UrlLocation
```

Defender XDR | Who clicked, and did Safe Links let them through?

```
UrlClickEvents
| where Timestamp > ago(7d)
| where Url has "secure-contoso-billing"
| project Timestamp, AccountUpn, Url, UrlChain, ActionType, IsClickedThrough,
          IPAddress, Workload
| sort by Timestamp asc
```

Defender XDR | Retro-hunt: has this landing domain been sent to anyone else, ever?

```
let badDomain = "secure-contoso-billing.com";
EmailUrlInfo
| where UrlDomain has badDomain
| join kind=inner EmailEvents on NetworkMessageId
| summarize Recipients = make_set(RecipientEmailAddress, 50), FirstSeen = min(Timestamp),
            LastSeen = max(Timestamp), Messages = count()
  by UrlDomain
```

10.8 Decision point

URL FINDING	WEIGHT	INTERPRETATION
Final page requests Microsoft/Google credentials on a non-Microsoft/Google domain	**Decisive**	Credential phishing. Malicious.
Redirect chain ends at a reverse-proxy mirroring a real IdP login	**Decisive**	AiTM. Malicious — treat any click as a probable session compromise.
URL refuses to render without the victim's email encoded in it	**Strong**	Deliberate evasion of automated scanners. Malicious.
Domain resolves but page is now a 404 or parked	**Inconclusive**	Kit may be cloaking you, or was already taken down. Rely on header/content findings.
Single-hop link to a long-established, correctly-branded corporate domain	**Weak-negative**	Supportive of legitimacy only if authentication and content also check out.

10.9 How this shapes the next step

If the landing page is a credential form, proceed to Chapter 12 (threat intel enrichment on the domain) and Chapter 13 (who else clicked). **If you suspect AiTM, do not wait** — jump to Chapter 14 immediately and in

parallel, because a stolen session cookie is exploitable within minutes, and password resets alone will not contain it.

CHAPTER 11 · PHASE 7

Attachment and Payload Analysis

Move from static inspection to safe detonation, in that order, and never on a production endpoint. This chapter is about understanding what a file does — not about reverse engineering it.

PHASE 7 **Attachment and Payload Analysis**

OBJECTIVE

Determine what a file actually is, what it does when opened or executed, and what infrastructure it talks to — through static analysis first, then safe dynamic detonation.

WHY THIS STEP IS IMPORTANT

File extensions lie. A file named `invoice.pdf.exe`, or a genuine PDF that embeds a launch action, behaves completely differently from what its icon suggests. Understanding the full execution chain — loader, command-and-control, secondary payload — is what lets you scope impact accurately and write a detection that stops the next one.

EXPECTED FINDINGS

True file type, presence of macros or embedded objects and whether they auto-execute, the process tree observed on detonation, any command-and-control domain contacted, and any dropped files with their hashes.

11.1 Static analysis first — nothing is executed

1. **Confirm the true file type** from magic bytes, not the extension.
2. **Hash it** and check VirusTotal / your internal TI store before doing anything else — someone else may have already detonated this exact file.
3. **Inspect Office documents for macros** and check whether they are set to auto-run.
4. **Inspect PDFs for embedded JavaScript or launch actions.**
5. **Inspect archives** (ZIP/RAR/7z, ISO, IMG) for their contents without extracting to a live path; note password protection, which is itself a sandbox-evasion signal.
6. **Check LNK files** for their target command line — this is often where the real payload hides.

```
Analysis VM  |  Static triage toolbelt
```

```
pip install oletools --break-system-packages
# True file type, regardless of extension
file attach_invoice.doc

# Office macro analysis - flags AutoOpen/Document_Open and suspicious API calls
olevba attach_invoice.doc

# PDF structure - look for /JavaScript, /Launch, /OpenAction
python3 -c "
import re
data = open('attach_invoice.pdf','rb').read()
for tag in [b'/JavaScript', b'/JS', b'/Launch', b'/OpenAction', b'/EmbeddedFile']:
    print(tag, data.count(tag))
"

# LNK target - the real command often lives here
python3 -m pylnk3 attach_shortcut.lnk

# Hash everything before you touch it further
sha256sum attach_*
```

RED FLAG — PASSWORD-PROTECTED ARCHIVES

A ZIP that needs a password quoted in the email body cannot be opened by an automated sandbox — that is precisely why attackers use it. The presence of a password, by itself, is a meaningful indicator: legitimate business attachments are very rarely protected this way for internal correspondence.

11.2 Dynamic analysis — safe detonation

Once static analysis is done, detonate the file in an environment built for it — never your own machine, and ideally not even your internal network unless it is properly isolated.

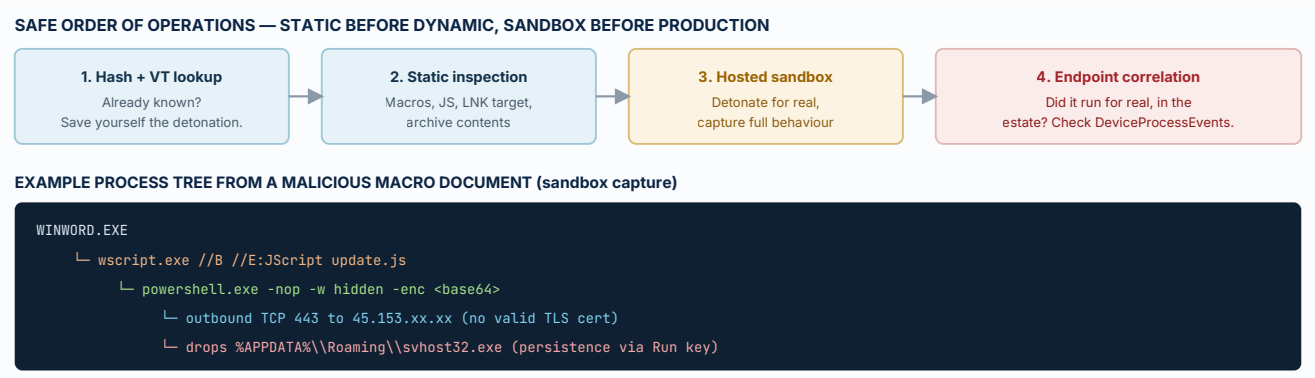

Figure 11.1 — Safe attachment analysis workflow and a representative malicious process tree. `WINWORD.EXE` spawning a scripting host, which spawns PowerShell, which reaches out to an uncategorised external IP, is one of the most reliable "this is malware" signatures in endpoint telemetry.

11.3 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Hash lookup only; escalate anything not already known-clean.	VirusTotal, Defender file reputation
SOC L2	Static triage; submit to a hosted sandbox; read the behaviour report.	oletools, CyberChef, Any.Run, Hybrid Analysis
Malware Analyst	Manual static/dynamic reversing when the sandbox report is inconclusive or evasive.	Ghidra/IDA (for deep RE, outside this guide's scope), x64dbg, PE-bear
DFIR	Correlate sandbox behaviour with real endpoint telemetry to confirm actual execution in the estate.	Defender for Endpoint (DeviceProcessEvents, DeviceNetworkEvents, DeviceFileEvents)

11.4 MITRE ATT&CK mapping

TECHNIQUE	NAME	EVIDENCE
`T1566.001`	Spearphishing Attachment	The delivery mechanism.
`T1204.002`	User Execution: Malicious File	User opens the file and enables content.
`T1059.001/.005/.007`	Command and Scripting Interpreter	PowerShell, VBScript or JScript observed in the process tree.
`T1218`	System Binary Proxy Execution	Abuse of a signed OS binary (e.g. `mshta.exe`, `rundll32.exe`) to launch the payload.
`T1027`	Obfuscated Files or Information	Base64/encoded PowerShell, packed binaries.
`T1105`	Ingress Tool Transfer	Secondary payload downloaded after initial execution.
`T1547.001`	Boot or Logon Autostart: Registry Run Keys	Persistence mechanism observed in sandbox or endpoint telemetry.

11.5 IOC extraction

- SHA-256 (and MD5, for older TI feeds) of the original attachment
- SHA-256 of every dropped or downloaded secondary file
- Command-and-control domains/IPs contacted during detonation
- Mutex names, registry Run-key values, scheduled task names (persistence artefacts)
- Any file paths written outside expected temp/download locations

11.6 KQL for attachment and endpoint correlation

Defender XDR | Has this exact attachment hash been sent to anyone else?

```
EmailAttachmentInfo
| where SHA256 == "<attachment-sha256>"
| join kind=inner EmailEvents on NetworkMessageId
| project Timestamp, SenderFromAddress, RecipientEmailAddress, Subject,
          FileName, FileType, DeliveryAction, DeliveryLocation
| sort by Timestamp desc
```

Defender XDR | Did the attachment actually execute on an endpoint?

```
// Bridges the email side and the endpoint side using the file hash.
DeviceProcessEvents
| where Timestamp > ago(7d)
| where SHA256 == "<attachment-sha256>" or InitiatingProcessSHA256 == "<attachment-sha256>"
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine,
          InitiatingProcessFileName, InitiatingProcessCommandLine
| sort by Timestamp asc
```

Defender XDR | Network beacon from the affected device after execution

```
DeviceNetworkEvents
| where Timestamp > ago(7d)
| where DeviceName == "<device-from-previous-query>"
| where InitiatingProcessFileName in~ ("powershell.exe","wscript.exe","mshta.exe","rundll32.exe")
| project Timestamp, DeviceName, RemoteIP, RemoteUrl, RemotePort,
          InitiatingProcessFileName, InitiatingProcessCommandLine
| sort by Timestamp asc
```

11.7 Decision point

ATTACHMENT FINDING	WEIGHT	INTERPRETATION
Macro auto-executes and downloads a second-stage payload	**Decisive**	Malware. Confirm scope via endpoint telemetry.
Hash matches a known family on VirusTotal / internal TI	**Decisive**	Malicious; use the known family's IOCs to widen the hunt.
Sandbox shows a beacon to an uncategorised external IP	**Strong**	Malicious even without a named family.
Password-protected archive with password only in the email body	**Strong (evasion itself is a signal)**	Treat as malicious pending detonation with the supplied password.
File is a genuine, unmodified document with no macros or embedded objects	**Weak-negative**	Supportive of legitimacy only alongside clean header/content findings.

11.8 How this shapes the next step

A confirmed malicious attachment moves the case immediately toward Chapter 13 (who else received the same file) and, if it executed anywhere in the estate, Chapter 14 (was there follow-on lateral movement?) even though that chapter is framed around identity compromise — the same containment urgency applies to a running implant. IOCs from this phase also feed directly into Chapter 12's enrichment.

CHAPTER 12 · PHASE 8

Threat Intelligence Enrichment

Turn a list of isolated indicators into context: how old is this infrastructure, who else has seen it, and is this part of something bigger than one email?

PHASE 8 **Threat Intelligence Enrichment**

OBJECTIVE

Enrich every IOC gathered so far — domains, IPs, hashes, URLs — with reputation, registration history and any known campaign association, to corroborate or challenge the verdict forming from Chapters 7–11.

WHY THIS STEP IS IMPORTANT

A single source is an opinion; three independent sources agreeing is evidence. Enrichment is also what lets you prioritise: a one-off, unsophisticated attempt and an active, multi-victim campaign both start as "one email," but they deserve very different amounts of response effort.

EXPECTED FINDINGS

Domain age and registrar, historical resolutions, related/typosquat domains, detection ratios and first-seen dates, and — where available — a named campaign or actor cluster.

12.1 The enrichment checklist

CHECK	WHAT IT TELLS YOU	WHERE TO RUN IT
WHOIS / RDAP	Registration date, registrar, privacy-proxy usage.	WHOIS, a domain registrar lookup, or RDAP directly
Passive DNS history	What else has this IP hosted recently? Rapid domain churn on one IP is a hosting-for-abuse pattern.	Passive DNS providers, URLScan.io history
Reputation score	Has anyone already flagged this as malicious?	VirusTotal, AbuseIPDB
URL scan history	Has this exact URL been submitted before, and what did earlier scans show?	URLScan.io search
Certificate transparency	Related subdomains or sibling typosquats sharing the same certificate issuance pattern.	crt.sh or equivalent CT log search
Internal TI / MISP match	Have we, or a sharing community we belong to, seen this before?	MISP, internal TI platform

CHECK	WHAT IT TELLS YOU	WHERE TO RUN IT
Open-source reporting	Is there public reporting describing this campaign or kit?	Vendor blogs, ISAC advisories, general web search

RED FLAG — TREAT "BRAND NEW" AS SUSPICIOUS, NOT AS NEUTRAL

A domain registered nine days ago, with zero VirusTotal detections and no WHOIS history, is not "unproven." It is exactly what freshly-stood-up phishing infrastructure looks like *before* anyone has reported it. Absence of reputation is not evidence of safety — it is often the strongest indicator you will get from this phase for a new campaign.

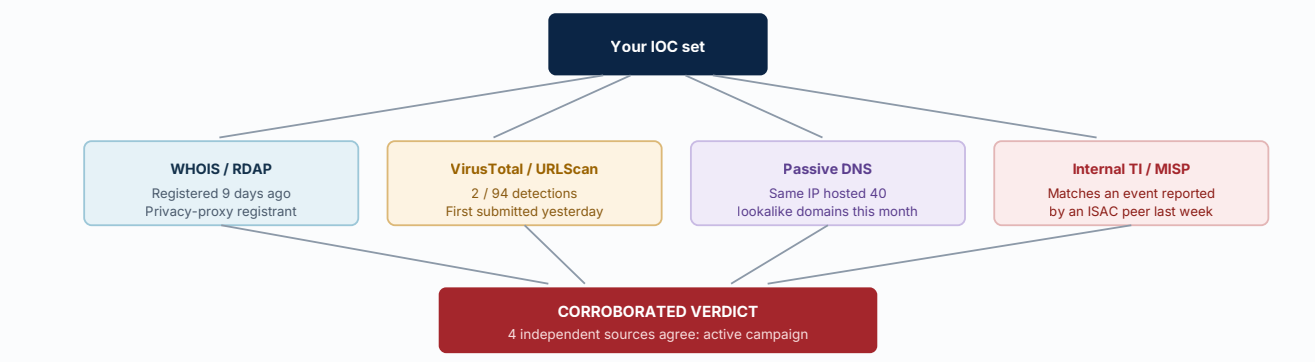

Figure 12.1 — Enrichment as corroboration. Each source alone is weak evidence. Several independent sources pointing the same direction is what turns a suspicion into a defensible verdict.

12.2 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Quick reputation lookups on the domain, IP, URL and hash.	VirusTotal, URLScan.io, MXToolbox
SOC L2	WHOIS/RDAP and passive DNS review; cross-check certificate transparency for sibling domains.	WHOIS, crt.sh, AbuseIPDB
Threat Hunter	Cluster the campaign; add or update an internal TI event; write a working name for tracking.	MISP, internal TI platform, Advanced Hunting
DFIR	Use enrichment to prioritise scope and containment order across multiple concurrent cases.	Case management, MISP

12.3 MITRE ATT&CK mapping

TECHNIQUE	NAME	EVIDENCE
T1583.001	Acquire Infrastructure: Domains	Freshly registered lookalike domain.
T1583.003	Acquire Infrastructure: Virtual Private Server	Hosting pattern seen in passive DNS.
T1584.004	Compromise Infrastructure: Server	Legitimate but compromised site used to host the kit.

12.4 IOC extraction (enriched records)

- Domain registration date, registrar, and name-server pattern
- Historical IP resolutions and their hosting ASN
- Related domains sharing infrastructure, certificate, or template
- Detection ratio and first-seen date on reputation platforms
- Campaign or cluster name (internal working name if nothing public exists)

12.5 KQL for enrichment and retro-hunting

Defender XDR | Has any IOC from this case appeared anywhere in the tenant before?

```
let iocDomains = dynamic(["contoso-invoices.com","secure-contoso-billing.com"]);
let iocIPs     = dynamic(["185.220.101.47"]);
union
  (EmailEvents | where Timestamp > ago(90d) | where SenderFromDomain has_any (iocDomains) or
SenderIPv4 in (iocIPs)
     | project Timestamp, Kind="Email", Detail=strcat(SenderFromAddress," from ",SenderIPv4)),
  (EmailUrlInfo | where UrlDomain has_any (iocDomains)
     | project Timestamp, Kind="URL", Detail=Url),
  (DeviceNetworkEvents | where Timestamp > ago(90d) | where RemoteUrl has_any (iocDomains) or
RemoteIP in (iocIPs)
     | project Timestamp, Kind="Network", Detail=strcat(DeviceName," → ",RemoteUrl))
| sort by Timestamp desc
```

Sentinel | Match tenant telemetry against a curated threat-intel indicator list

```
// Requires ThreatIntelligenceIndicator populated via your TI feed / MISP integration.
ThreatIntelligenceIndicator
| where Active == true
| where TimeGenerated > ago(30d)
| project IndicatorType, NetworkIP, Url, DomainName, ConfidenceScore, ThreatType
| join kind=inner (
    DeviceNetworkEvents | where Timestamp > ago(7d)
  ) on $left.NetworkIP == $right.RemoteIP
| project TimeGenerated, DeviceName, RemoteIP, RemoteUrl, ThreatType, ConfidenceScore
```

12.6 Decision point

ENRICHMENT FINDING	WEIGHT	INTERPRETATION
Multiple independent sources corroborate (WHOIS age + reputation + passive DNS pattern)	**Decisive**	High-confidence malicious; prioritise accordingly.
Domain matches a known, named campaign in TI	**Decisive**	Apply that campaign's known IOC set to widen the hunt immediately.
Brand-new domain, zero history anywhere	**Treat as suspicious, not neutral**	Lean on Chapters 7–11 findings; absence of reputation is not absence of risk.
Long-established domain, clean history, no corroborating hits	**Weak-negative**	Supportive of legitimacy only in combination with clean findings elsewhere.

12.7 How this shapes the next step

Enrichment either confirms the case is isolated or reveals it is one email inside a much larger campaign. Either way, the next step is the same: find out exactly how far this has spread inside your own tenant, which is the subject of Chapter 13.

CHAPTER 13 · PHASE 9

Blast-Radius Scoping in Microsoft 365

The reporting user is almost never the only recipient. This chapter builds the exposure funnel that every response action in Chapter 16 depends on.

PHASE 9 **Blast-Radius Scoping**

OBJECTIVE

Determine, tenant-wide, exactly who received the message, who opened it, who clicked any link, who opened any attachment, and who shows signs of actual compromise.

WHY THIS STEP IS IMPORTANT

Scoping too narrowly leaves compromised accounts unaddressed and lets an attacker continue operating inside your tenant. Scoping too broadly wastes response capacity chasing users who were never at risk. This phase produces the precise list that containment (Chapter 16) acts on — nothing more, nothing less.

EXPECTED FINDINGS

Total recipient count, and counts at each stage of the exposure funnel: delivered, read, clicked, and confirmed or suspected compromised.

13.1 The exposure funnel

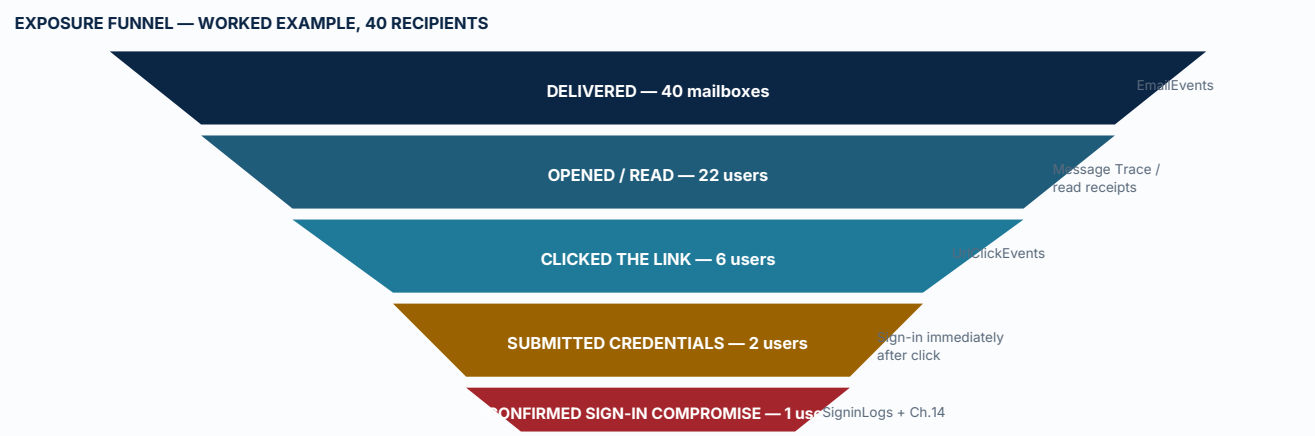

Figure 13.1 — The exposure funnel. Each stage narrows the response: all 40 recipients need the mail purged, but only the 1 confirmed-compromised user needs incident-level identity remediation. Getting this funnel right is what keeps the response proportionate.

13.2 How to build the funnel

1. **Pull the full recipient list** using the `NetworkMessageId` (or the campaign's `EmailClusterId` if multiple message variants exist).
2. **Check delivery location per recipient** — Inbox, Junk, or Quarantine — because only Inbox deliveries are a live exposure.

3. **Check** `UrlClickEvents` for every recipient, joined on `NetworkMessageId`.
4. **Check attachment execution** on endpoints via `DeviceFileEvents` / `DeviceProcessEvents` if a payload was involved.
5. **For any click, check sign-in activity** in the minutes immediately following — this is your practical proxy for "did they actually submit credentials," since Defender does not directly instrument every third-party phishing page's form submission.

13.3 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Export the recipient list and delivery status for the ticket.	Threat Explorer, Message Trace
SOC L2	Join email, click and attachment tables to build the funnel programmatically.	Advanced Hunting (KQL)
Threat Hunter	Expand scoping across the whole campaign cluster, not just the single reported message.	Advanced Hunting, Sentinel workbooks
DFIR / IR	Hand the "confirmed compromise" shortlist to identity remediation without delay.	Case management, Entra ID

13.4 IOC extraction at this phase

- The full list of `NetworkMessageId`s in the campaign cluster
- The list of affected UPNs/mailboxes, tagged by funnel stage
- The list of affected device names, where attachments executed

13.5 KQL for blast-radius scoping

Defender XDR | Full funnel for one campaign, in one query

```
let campaignMsgs = EmailEvents
    | where Timestamp > ago(7d)
    | where SenderFromDomain has "contoso-invoices.com"
    | project NetworkMessageId, RecipientEmailAddress, DeliveryLocation;
let clicked = UrlClickEvents
    | where NetworkMessageId in (campaignMsgs | project NetworkMessageId)
    | distinct NetworkMessageId, AccountUpn;
campaignMsgs
| join kind=leftouter clicked on NetworkMessageId
| summarize
    Delivered = dcount(RecipientEmailAddress),
    DeliveredToInbox = dcountif(RecipientEmailAddress, DeliveryLocation == "Inbox/folder"),
    Clicked = dcountif(RecipientEmailAddress, isnotempty(AccountUpn))
```

Defender XDR | At-risk shortlist: clicked AND signed in from a new location within 15 minutes

```
let clicks = UrlClickEvents
    | where Timestamp > ago(7d)
    | where Url has "secure-contoso-billing"
    | project ClickTime = Timestamp, AccountUpn, ClickIP = IPAddress;
clicks
| join kind=inner (
    AADSignInEventsBeta
    | where Timestamp > ago(7d)
  ) on $left.AccountUpn == $right.AccountUpn
| where (SignInTime := Timestamp) between (ClickTime .. ClickTime + 15m)
| where IPAddress =≠ ClickIP                    // sign-in from a DIFFERENT IP than the click
| project AccountUpn, ClickTime, ClickIP, SignInTime, IPAddress, Country, ErrorCode
| sort by ClickTime asc
```

13.6 Decision point

FUNNEL RESULT	SEVERITY	NEXT ACTION
Delivered only, zero opens/clicks	**Low/Medium**	Purge and close after IOC extraction.
Clicks recorded, no follow-on sign-in anomaly	**Medium/ High**	Reset affected passwords as a precaution; continue monitoring.
Click followed by a sign-in from an unfamiliar location/ASN	**Critical**	Treat as confirmed compromise. Proceed to Chapter 14 immediately.

13.7 How this shapes the next step

If the funnel shows **any** user in the "confirmed or suspected compromise" tier, Chapter 14 is not optional — it runs next, and in parallel with containment. If the funnel stops at "delivered" or "opened" with no clicks, you can move straight to the verdict in Chapter 15.

CHAPTER 14 · PHASE 10

Post-Compromise: Identity and Lateral Movement

A click or a submitted password is only dangerous if the attacker actually uses it. This chapter finds out whether they did, and how far they got.

PHASE 10 **Post-Compromise Identity Checks**

OBJECTIVE

Determine whether a stolen credential or stolen session was actually **used**, and if so, map everything the attacker did with it — sign-ins, persistence, data access, and any onward attack launched from inside your own tenant.

WHY THIS STEP IS IMPORTANT

This is where a phishing ticket becomes a full incident. A password reset alone does nothing against a stolen session cookie. An account disable alone does nothing against an OAuth app the attacker granted themselves, which keeps working independently of the user's password. Missing any one persistence mechanism means the attacker keeps their foothold even after you believe you have closed the case.

EXPECTED FINDINGS

Anomalous sign-ins (new location, ASN, or device), newly created inbox rules, new mail-forwarding configuration, new OAuth application consents, new MFA methods registered, and any phishing sent onward from the compromised mailbox to internal colleagues.

14.1 What attackers do once they are inside a mailbox

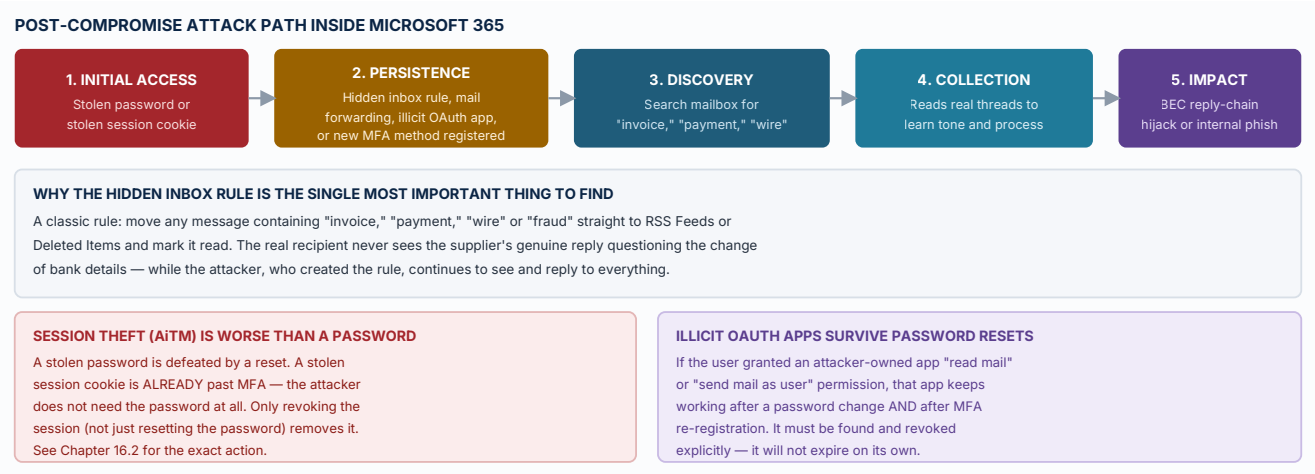

Figure 14.1 — The post-compromise attack path inside Microsoft 365. Every one of these stages leaves telemetry. The job of this phase is to check all five, not just the sign-in log.

14.2 The checklist, in order

1. **Sign-in anomalies.** New country, new ASN, a legacy authentication protocol suddenly in use, or "impossible travel" relative to the user's normal pattern, in the window right after a click.
2. **New or modified inbox rules.** Especially rules that move, hide, or delete mail containing financial keywords.
3. **Mailbox-level forwarding.** A silent SMTP forward is different from an inbox rule and must be checked separately.
4. **New OAuth application consents.** Look for unfamiliar app names, unverified publishers, and broad scopes such as `Mail.Read`, `Mail.Send`, or `offline_access`.
5. **New MFA / security-info registrations.** An attacker who reaches the security-info page can register their own authenticator or phone number, giving themselves durable access.
6. **Onward phishing.** Search for mail sent *from* the compromised account to other internal users — this is now an internal spearphishing case, not just an inbound one.
7. **Data access and download activity.** SharePoint/OneDrive access, Teams messages sent, or admin actions, if the account carries elevated privilege.

14.3 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L2/L3	Pull sign-in history and inbox rules for the affected account; compare against the user's normal baseline.	Entra ID sign-in logs, Exchange Online PowerShell
IR / DFIR	Build a full timeline correlating email, identity and cloud-app telemetry; coordinate token revocation.	Advanced Hunting, Purview audit log search
Threat Hunter	Proactively hunt the whole tenant for the same persistence pattern (not just the one known-affected account).	Advanced Hunting, Sentinel
Security Engineer	Review Conditional Access policy gaps that allowed the anomalous sign-in through.	Entra ID Conditional Access, Identity Protection

14.4 MITRE ATT&CK mapping

TECHNIQUE	NAME	EVIDENCE
`T1078.004`	Valid Accounts: Cloud Accounts	Sign-in using the stolen credential.
`T1550.004`	Use Alternate Authentication Material: Web Session Cookie	AiTM session replay — sign-in without a fresh MFA challenge.
`T1114.003`	Email Collection: Email Forwarding Rule	The hidden inbox rule or SMTP forward.

TECHNIQUE	NAME	EVIDENCE
T1098.001/.003	Account Manipulation: Additional Cloud Credentials / Roles	New MFA method or elevated role granted.
T1528	Steal Application Access Token	Illicit OAuth app consent.
T1534	Internal Spearphishing	Onward phishing sent from the compromised mailbox.
T1213	Data from Information Repositories	SharePoint/Teams search and access post-compromise.

14.5 IOC extraction

- Anomalous sign-in IP, ASN, country and device fingerprint
- Inbox rule name, condition and action (exact configuration)
- OAuth application ID, display name, publisher verification status, and granted scopes
- MFA method added (phone number or authenticator device identifier)
- Message IDs of any onward phishing sent internally

14.6 KQL for post-compromise hunting

Defender XDR / Entra ID | Sign-ins for this user around the click window

```
AADSignInEventsBeta
| where Timestamp between (datetime(2025-05-12T09:10:00Z) .. datetime(2025-05-12T11:00:00Z))
| where AccountUpn =~ "victim@example.com"
| project Timestamp, AccountUpn, IPAddress, Country, ISP, ClientAppUsed,
          AuthenticationRequirement, ConditionalAccessStatus, ErrorCode
| sort by Timestamp asc
```

Defender XDR | New inbox rules created tenant-wide in the last 7 days

```
// Surfaces via Exchange admin audit records ingested into CloudAppEvents.
CloudAppEvents
| where Timestamp > ago(7d)
| where ActionType in ("New-InboxRule", "Set-InboxRule", "UpdateInboxRules")
| extend RuleName = tostring(RawEventData.Parameters), Actor = AccountDisplayName
| project Timestamp, Actor, ActionType, RuleName, IPAddress
| sort by Timestamp desc
```

Defender XDR | New OAuth application consent grants

```
CloudAppEvents
| where Timestamp > ago(14d)
| where ActionType has "Consent to application"
| project Timestamp, AccountDisplayName, Application, ActionType, IPAddress, RawEventData
| sort by Timestamp desc
```

Defender XDR | Did the compromised account send mail onward, internally?

```
EmailEvents
| where Timestamp > ago(3d)
| where SenderFromAddress =~ "victim@example.com"
| where EmailDirection == "Intraorg"
| project Timestamp, SenderFromAddress, RecipientEmailAddress, Subject, UrlCount, AttachmentCount
| sort by Timestamp asc
```

14.7 Decision point

FINDING	WEIGHT	INTERPRETATION
Sign-in from unfamiliar ASN/country within minutes of a click, no matching travel	**Decisive**	Confirmed account takeover.
New inbox rule hiding financial-keyword mail	**Decisive**	Active BEC persistence — likely mid-fraud, treat as critical.
New OAuth app with mail read/send scope, unverified publisher	**Decisive**	Persistent access independent of password — must be revoked explicitly.
Click confirmed, but no sign-in anomaly, no new rules, no new apps found	**Inconclusive-negative**	No confirmed compromise yet — keep the account under heightened monitoring for the containment window (Chapter 16).

14.8 How this shapes the next step

Any decisive finding here elevates the case to Critical (Chapter 5.4) if it was not already, and moves the investigation straight into Chapter 16's containment actions — session revocation, rule removal, and app revocation all happen together, not one at a time. Whatever you find here becomes the centrepiece of both the verdict (Chapter 15) and the incident report (Chapter 17).

CHAPTER 15 · PHASE 11

The Verdict and the Decision Framework

Every phase so far has produced evidence. This chapter turns that evidence into one defensible sentence: malicious, suspicious, or legitimate — and why.

PHASE 11 **The Verdict**

OBJECTIVE

Synthesise every finding from Chapters 7–14 into a single verdict — **Malicious**, **Suspicious**, or **Legitimate** — with a confidence level and a written rationale that another analyst could check and reach the same conclusion from.

WHY THIS STEP IS IMPORTANT

A pile of individually interesting findings is not an answer. Stakeholders, auditors, and your own future self need one clear conclusion, reproducibly reached. A verdict without a stated rationale cannot be reviewed, challenged, or learned from.

EXPECTED FINDINGS

A verdict, a confidence level (High / Medium / Low), and the specific evidence — phase by phase — that supports it.

15.1 Decisive gates vs. weighed evidence

Some findings are **decisive** on their own: any single one of them is sufficient for a Malicious verdict, regardless of what anything else shows. Everything else is **corroborating**: individually weak, but persuasive in combination. Always check the decisive gates first — they save time and remove ambiguity.

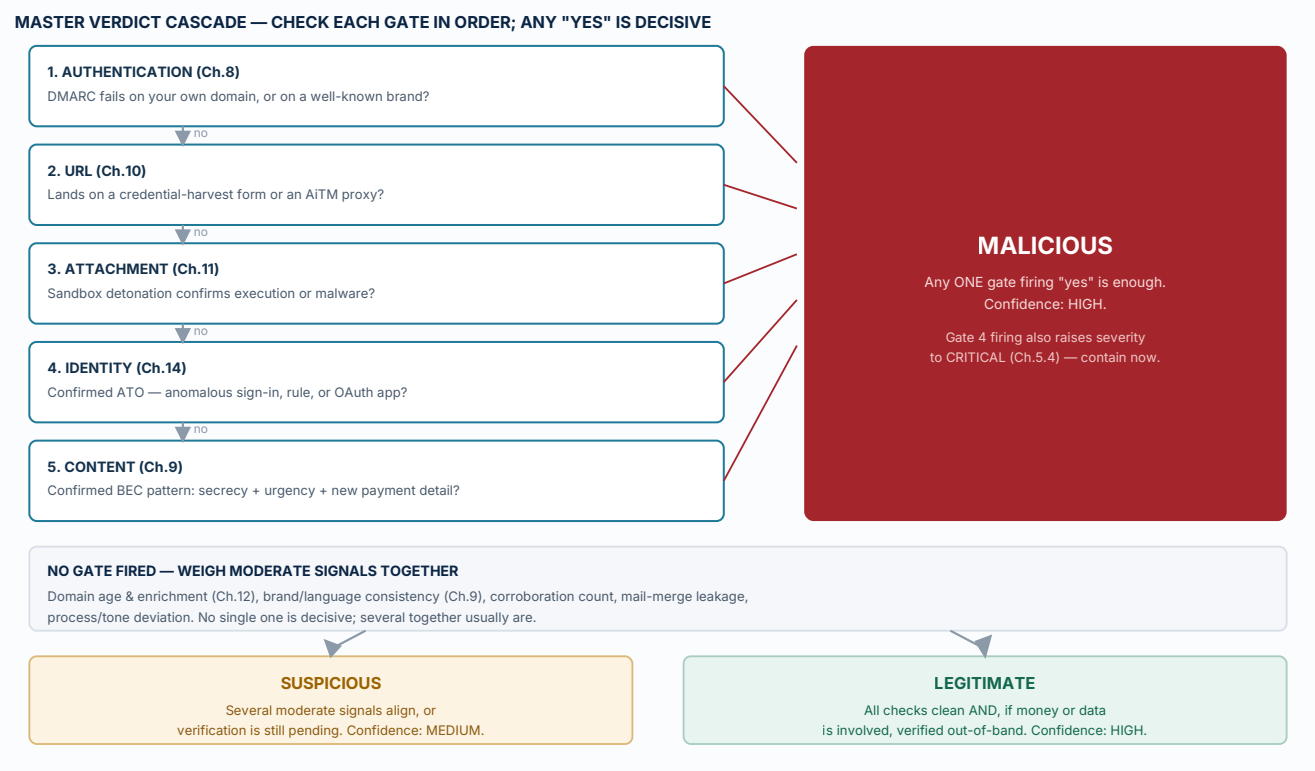

Figure 15.1 — The master verdict cascade. Five decisive gates, checked in any order in practice but shown here as a cascade for clarity. If none fire, the case is settled by weighing corroborating signals rather than by a single technical result.

15.2 Worked example — scoring the running case

Applying the cascade to the "Invoice 44821 / Contoso Billing" case threaded through Chapters 5–13:

PHASE	FINDING	GATE RESULT
Ch.8 Authentication	SPF/DKIM/DMARC all pass — for the attacker's own cousin domain, not yours.	**Not decisive alone**
Ch.10 URL	Redirect chain ends at a credential-harvest form styled as an AP portal login.	**GATE FIRES**
Ch.13 Blast radius	3 of 14 recipients clicked; `IsClickedThrough = true` for one.	Corroborating
Ch.14 Identity	One clicker shows a sign-in from a new ASN eleven minutes after the click.	**GATE FIRES**

VERDICT

Malicious. Confidence: High. Severity: Critical (identity gate fired). Two independent decisive gates — the credential-harvest landing page and the confirmed anomalous sign-in — make this straightforward to defend to any reviewer.

15.3 Aligning to your platform's own classification

Whatever internal wording you use, map it to the closing classification your case-management or XDR platform expects (for example, Microsoft Defender XDR's *true positive / benign positive / false positive* classification with a supporting determination). This keeps your metrics comparable over time and makes your case searchable by future analysts and by threat hunters building on your work.

15.4 Techniques by role, and tools

ROLE	TECHNIQUE	TOOLS
SOC L1	Proposes a preliminary verdict from triage and header/authentication findings.	Ticketing system verdict field
SOC L2/L3	Confirms with full evidence across all phases; applies the cascade explicitly.	Case notes, the cascade above
DFIR / IR	Signs off the verdict for incidents that escalate beyond routine phishing.	Incident record, formal report (Ch.17)
Threat Hunter	Supplies campaign context that can raise confidence even where single-message evidence is thin.	MISP / TI platform

15.5 KQL — a one-query case summary

Defender XDR | Roll up the key numbers for the report in one shot

```
let msgs = EmailEvents | where SenderFromDomain has "contoso-invoices.com" | where Timestamp >
ago(7d);
let clicks = UrlClickEvents | where NetworkMessageId in (msgs | project NetworkMessageId);
print
  Recipients      = toscalar(msgs | summarize dcount(RecipientEmailAddress)),
  DeliveredInbox  = toscalar(msgs | summarize countif(DeliveryLocation == "Inbox/folder")),
  Clickers        = toscalar(clicks | summarize dcount(AccountUpn)),
  ClickedThrough  = toscalar(clicks | summarize countif(IsClickedThrough == true))
```

15.6 How this shapes the next step

A **Malicious** or **Suspicious** verdict sends you into full containment (Chapter 16). A **Legitimate** verdict still deserves a short closing note — confirm with the sender through a second channel if money or data was involved, and close. Either way, the verdict and its supporting evidence become the backbone of the report you write in Chapter 17.

CHAPTER 16 · PHASE 12

Containment, Eradication and Recovery

Turn the verdict into action: stop the bleeding, remove every foothold the attacker planted, and bring the affected users back to normal, trusted operation.

PHASE 12 **Containment, Eradication and Recovery**

OBJECTIVE

Remove the malicious email from every mailbox that received it, block the infrastructure behind it, and — where compromise occurred — strip out every persistence mechanism the attacker planted, then restore normal access.

WHY THIS STEP IS IMPORTANT

Analysis that does not lead to action protects no one. This phase must be **thorough**, not just fast: reset a password but leave an attacker-owned OAuth app in place, and the attacker keeps their access regardless. Order matters too — contain before you fully tip your hand, especially in cases where the attacker is actively watching a compromised mailbox.

EXPECTED FINDINGS

Confirmation that mail was purged tenant-wide, that infrastructure is blocked, that sessions were revoked, and that every persistence mechanism identified in Chapter 14 has been removed and verified gone.

16.1 Contain, eradicate, recover — not the same thing

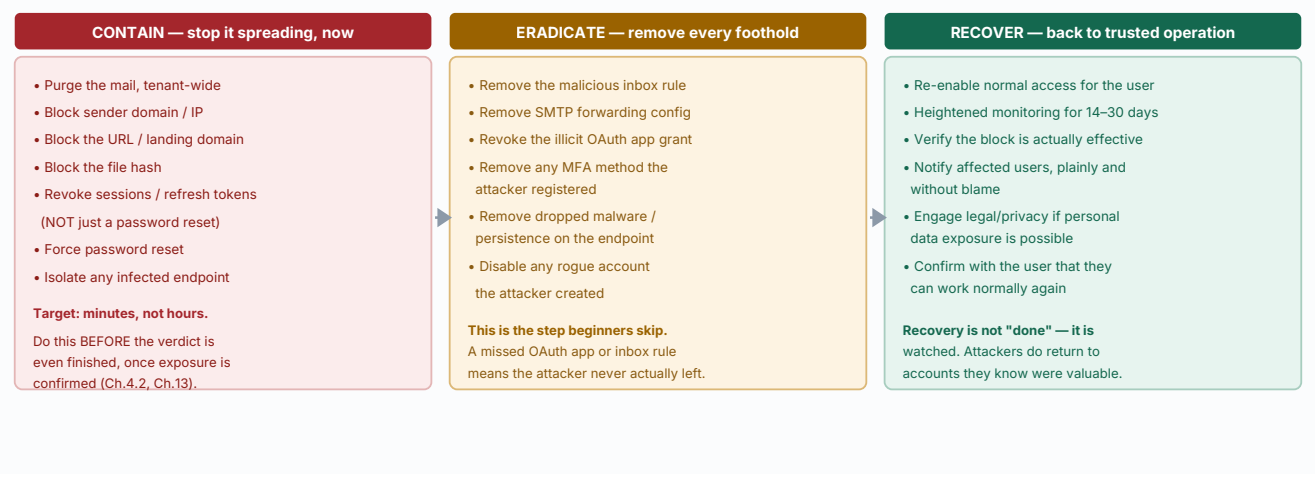

Figure 16.1 — Contain, eradicate, recover. They run roughly left to right in time, but containment and the start of eradication usually happen in the same sitting for a confirmed compromise — don't let "contain" become an excuse to delay "eradicate."

16.2 Command reference

`Exchange Online / Security & Compliance | Purge the mail tenant-wide`

```
# Search first, review the result set, THEN purge - never purge blind.
New-ComplianceSearch -Name "Phish-Invoice44821" -ExchangeLocation All `
  -ContentMatchQuery '(Subject:"Invoice 44821") AND (From:contoso-invoices.com)'
Start-ComplianceSearch -Identity "Phish-Invoice44821"
# Review New-ComplianceSearch results, THEN:
New-ComplianceSearchAction -SearchName "Phish-Invoice44821" -Purge -PurgeType SoftDelete
```

Microsoft 365 Defender | Block the infrastructure

```
# Tenant Allow/Block List - domain, URL and sender
New-TenantAllowBlockListItems -ListType Sender -Block -Entries "contoso-invoices.com" -NoExpiration
New-TenantAllowBlockListItems -ListType Url -Block -Entries "secure-contoso-billing.com" -
NoExpiration
# File hash block (Defender for Endpoint indicator)
# Portal: Settings > Endpoints > Indicators > File hash > Add > Action: Block and remediate
```

Entra ID | Revoke the session BEFORE, or as well as, resetting the password

```
# This is the step that actually defeats AiTM session/cookie theft.
Revoke-MgUserSignInSession -UserId "victim@example.com"
# Then force a password reset and require MFA re-registration
Update-MgUser -UserId "victim@example.com" -PasswordProfile @{ForceChangePasswordNextSignIn=$true}
```

Exchange Online | Eradicate persistence

```
# Remove the malicious inbox rule
Get-InboxRule -Mailbox victim@example.com | Where-Object {$_.Name -like "*RSS*" -or
$_.MoveToFolder}
Remove-InboxRule -Mailbox victim@example.com -Identity "<rule-identity>"
# Remove SMTP forwarding, if configured
Set-Mailbox victim@example.com -ForwardingSmtpAddress $null -DeliverToMailboxAndForward $false
# Review and revoke the illicit OAuth app grant (Entra ID > Enterprise Applications >  > Delete)
Remove-MgOauth2PermissionGrant -OAuth2PermissionGrantId "<grant-id>"
```

WARNING — VERIFY, DON'T ASSUME

After every containment action, verify it worked: query `UrlClickEvents` and `EmailEvents` to confirm no further deliveries from the blocked domain, and query `AADSignInEventsBeta` to confirm no further sign-ins on the revoked session. "I ran the command" is not the same as "it took effect."

Defender XDR | Verification query — confirm the block is holding

```
EmailEvents
| where Timestamp > ago(1d)
| where SenderFromDomain has "contoso-invoices.com"
| where DeliveryAction =≠ "Blocked"
| count
```

16.3 User communication

Tell affected users plainly what happened, what you did about it, and what — if anything — they need to do. Avoid blame; a user who reports a click honestly and quickly is exactly the behaviour you want to encourage for next time.

GOOD PRACTICE — A SHORT, CALM NOTIFICATION

"You received an email today impersonating our billing system. Our records show you clicked the link. We have reset your password and secured your account as a precaution — there is no action needed from you. If you notice anything unusual with your account in the next few days, please contact the SOC immediately. Thank you for reporting it so quickly."

16.4 Techniques by role, and the tools that do the job

ROLE	TECHNIQUE	TOOLS
SOC L1	Purge via Threat Explorer's built-in action; escalate blocking requests per policy.	Defender XDR Threat Explorer, Tenant Allow/ Block List
SOC L2	Bulk remediation via PowerShell across multiple mailboxes; endpoint indicator blocking.	Exchange Online PowerShell, Defender for Endpoint
IR / DFIR	Session revocation, OAuth app removal, and endpoint isolation for confirmed compromises.	Microsoft Graph PowerShell, Defender for Endpoint live response
Security Engineer	Converts one-off blocks into standing policy: DMARC enforcement, Conditional Access, transport rules.	Entra ID Conditional Access, EOP mail flow rules

16.5 Containment-completeness checklist

- [ ] Mail purged from every mailbox that received it, verified by re-query
- [ ] Sender domain/IP, landing URL/domain, and file hash all blocked
- [ ] Every session/refresh token revoked for confirmed-compromised accounts
- [ ] Passwords reset and MFA re-registered where an attacker may have added a method
- [ ] Every malicious inbox rule and SMTP forward removed
- [ ] Every illicit OAuth app consent revoked
- [ ] Infected endpoints isolated and remediated
- [ ] Affected users notified
- [ ] Legal/privacy engaged if personal data exposure is plausible

16.6 How this shapes the next step

Once every item on the checklist above is verified — not just actioned, but confirmed effective — the case is ready to close. Chapter 17 turns everything gathered from Chapters 5 through 16 into the written record that outlives the ticket.

CHAPTER 17 · PHASE 13

Reporting, Closure and Lessons Learned

The report is what survives after the ticket closes. Write it so a stranger, six months from now, understands exactly what happened and why it won't happen the same way twice.

PHASE 13 **Reporting, Closure and Lessons Learned**

OBJECTIVE

Document the investigation in a form useful to three different audiences at once — management, technical peers, and future analysts — formally close the case, and feed concrete improvements back into detection engineering and security awareness.

WHY THIS STEP IS IMPORTANT

An investigation with no written report might as well not have happened, from the organisation's point of view: nothing is learned, nothing improves, and the next analyst who hits a similar case starts from zero. Lessons-learned tracking is where a SOC's effectiveness compounds over months and years.

EXPECTED FINDINGS

A complete report, an updated detection-engineering backlog item (or an explicit decision not to act, with a reason), and — where relevant — an awareness-training input.

17.1 Report structure

SECTION	PURPOSE
Executive summary	Four to six sentences, no jargon: what happened, impact, current status.
Timeline	Timestamped sequence from delivery to closure.
Technical findings	One subsection per phase (header, authentication, content, URL, attachment, identity), each with its evidence.
IOC table	Every indicator, defanged, ready for sharing or blocklisting.
MITRE ATT&CK summary	The full technique chain observed, tactic by tactic.
Impact assessment	Recipients, clicks, confirmed compromise, data/financial exposure.
Response actions taken	Containment, eradication and recovery steps, with timestamps and verification.
Root cause / detection gap	Why did this reach the inbox, or why wasn't it caught sooner?
Recommendations	Specific, owned, dated actions — not vague aspirations.
Appendices	Full headers, raw KQL output, sandbox reports.

17.2 Sample incident report

CASE REFERENCE: SOC-14892 | "INVOICE 44821" CREDENTIAL-PHISHING INCIDENT

Executive summary. On 12 May, a credential-phishing email impersonating a supplier billing system was sent to 14 members of the Accounts Payable team. It was delivered to the inbox, and three recipients clicked the embedded link. One of those recipients' accounts showed a sign-in from an unfamiliar location shortly afterwards, confirming account compromise. The account was contained within 40 minutes of detection. No financial loss occurred. Root cause: the sending domain was newly registered and not yet reputation-flagged at delivery time.

Timeline (UTC, 12 May). 09:12 message delivered · 09:14–09:31 three clicks recorded · 09:26 anomalous sign-in on affected account · 10:40 user-reported via Report Phishing button · 10:44 triage complete, severity set Critical · 10:52 session revoked, password reset · 11:05 mail purged tenant-wide, domain/URL blocked · 11:20 inbox rule check completed (none found) · 12:10 report drafted, case closed.

MITRE ATT&CK chain observed. T1589 (recon, inferred) → T1583.001 (domain acquisition) → T1566.002 (spearphishing link) → T1204.001 (user execution) → T1078.004 (valid account sign-in).

Impact. 14 recipients, 3 clicks, 1 confirmed account compromise, 0 confirmed data exfiltration, 0 financial loss.

Root cause and detection gap. The sending domain was nine days old at delivery time and had no reputation history, so content-based detections did not fire with high confidence. Recommendation: tune the anti-phish policy's impersonation protection threshold and add a standing hunting query for newly-registered domains passing DMARC (Chapter 8.7) to the daily hunting rotation.

Recommendations (owner, due date). (1) Add domain-age enrichment to the phishing triage playbook — Security Engineering, two weeks. (2) Run the DMARC-passing new-domain hunt (Ch.8.7) on a daily schedule — Threat Hunting, one week. (3) Include this pretext in the next quarterly awareness campaign — Security Awareness, next cycle.

17.3 Lessons-learned tracking

Every closed case should answer three questions, tracked over time so patterns become visible across dozens of incidents rather than getting lost in individual tickets:

- **What let it in?** (detection gap → feeds Security Engineering)
- **What let it succeed, if it did?** (human factor → feeds Awareness)
- **What slowed the response down?** (process gap → feeds SOC process improvement)

17.4 Metrics worth tracking

METRIC	DEFINITION
MTTD	Mean time from delivery to detection (or user report).
MTTC	Mean time from detection to first containment action.
MTTR	Mean time from detection to full closure.
Click rate	Clicks ÷ recipients, per campaign — tracks awareness effectiveness over time.
Repeat-pretext rate	How often the same lure type recurs — tracks whether awareness content is working.

17.5 Techniques by role

ROLE	CONTRIBUTION TO THE REPORT
SOC L1	Drafts the initial ticket summary and timeline entries as they happen.
SOC L2/L3	Compiles the full technical findings and IOC table.
DFIR / IR	Signs off the formal report for incidents that escalated beyond routine handling.
Security Engineer	Owns the detection-gap recommendations and their delivery.
Threat Hunter	Files the campaign into the TI platform for future recognition.

17.6 Closure checklist

- [ ] Verdict recorded with supporting evidence and confidence level
- [ ] All containment-completeness items from Chapter 16.5 verified, not just actioned
- [ ] IOC table finalised and shared with relevant blocklists / TI platform
- [ ] Report written and filed, in the structure above
- [ ] At least one detection or process improvement identified, with an owner and a date — or an explicit, recorded decision not to act, with a reason
- [ ] Affected users notified; legal/privacy engaged if applicable

PRO TIP — CLOSE THE LOOP, NOT JUST THE TICKET

The ticket closes when response actions are verified. The *case* closes when the detection gap has an owner and a due date. Many SOCs are excellent at the first and poor at the second — and it is the second one that actually reduces next quarter's ticket volume.

17.7 End of the workflow

This completes the thirteen-phase investigation. Part Three now turns to the material that makes you faster and more confident at every one of these phases: a dedicated Microsoft 365 hunting methodology for threat hunters, worked real-world case studies covering different attack patterns, hands-on scenarios you can practise against, quick-reference cheat sheets, and interview preparation.

PART THREE

Field Reference: Hunting, Scenarios and the Analyst's Toolkit

The workflow is complete. This part makes you faster and more confident at every phase of it: a dedicated proactive-hunting methodology, worked case studies covering different attack shapes, hands-on scenarios you can practise against, consolidated reference tables, cheat sheets, and interview preparation.

CHAPTER 18

Microsoft 365 Hunting Methodology

Investigation reacts to an alert. Hunting goes looking before there is one. This chapter gives threat hunters a repeatable loop and ten standing hunts to run against Microsoft 365.

18.1 Hunting vs. investigating — what's different

	INVESTIGATION (CHAPTERS 5–17)	HUNTING (THIS CHAPTER)
Trigger	An alert, or a user report, already exists.	A hypothesis: "I think this pattern exists somewhere in our data."
Starting point	One message, one user, one campaign.	The whole tenant, over a wide time window.
Success looks like	A verdict and a closed ticket.	Either a new finding to investigate, or a confirmed absence — both are useful.
Output	Contained incident, written report.	New detection rule, or a documented negative result that narrows the next hunt.

18.2 The hunting loop

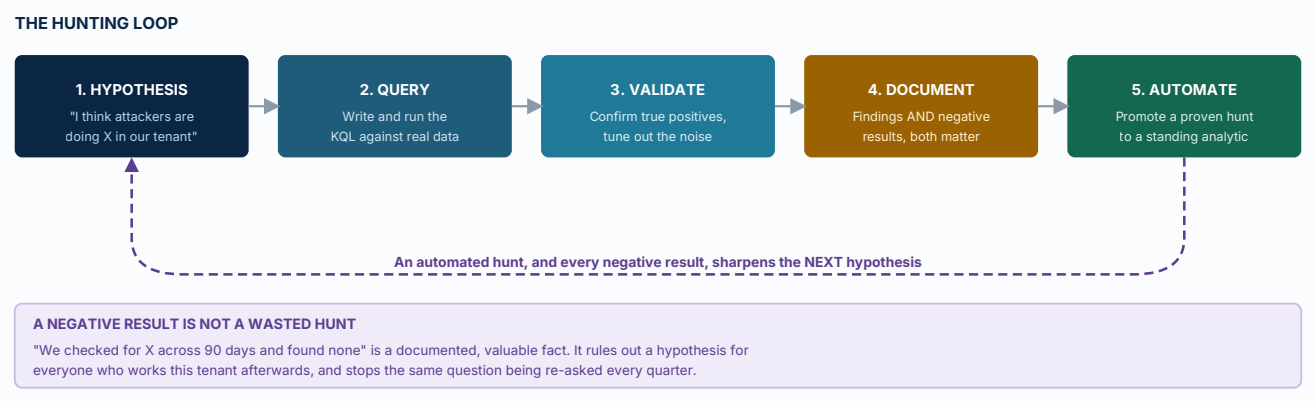

Figure 18.1 — The hunting loop. Unlike investigation, hunting has no natural end point — each cycle should leave the tenant a little better instrumented than the last.

18.3 Data sources at a glance

TABLE	WHAT IT GIVES A HUNTER
`EmailEvents` / `EmailUrlInfo` / `EmailAttachmentInfo`	Everything about the message itself — the starting point for any email-shaped hypothesis.
`UrlClickEvents`	Click behaviour, tenant-wide, independent of any one campaign.

TABLE	WHAT IT GIVES A HUNTER
`AADSignInEventsBeta` / `SigninLogs`	Identity truth — where hunts for account takeover live.
`CloudAppEvents` / `OfficeActivity`	Inbox rules, OAuth consents, admin actions — the persistence layer.
`DeviceProcessEvents` / `DeviceNetworkEvents`	Endpoint truth for anything with an attachment or a dropped payload.
`ThreatIntelligenceIndicator`	Your curated IOC feed, joinable against every table above.

18.4 Ten standing hunts for Microsoft 365 phishing

The first two are full write-ups from earlier chapters — run them on a schedule, not just during an incident. The rest are new, tenant-wide generalisations built for continuous hunting rather than one campaign.

#	HUNT	CADENCE	REFERENCE
1	Newly-seen domains passing DMARC (lookalike hunt)	Daily	Chapter 8.7
2	Executive display-name impersonation	Daily	Chapter 7.7
3	Tenant-wide click-then-anomalous-sign-in (AiTM)	Every 4 hours	Below, 18.4.1
4	Impossible-travel sign-ins following any phishing click	Every 4 hours	Below, 18.4.2
5	HTML-smuggling attachments	Daily	Below, 18.4.3
6	Tenant-wide risky OAuth consent sweep	Daily	Below, 18.4.4
7	Mass hidden inbox-rule sweep (all mailboxes)	Weekly	Generalises Chapter 14.6
8	Single-sender-to-many-internal-recipients spike	Weekly	Generalises Chapter 9.8
9	QR-code (quishing) pattern: zero URLs, image attachment, mobile sign-in follows	Weekly	Pattern-based, new
10	Repeat-offender infrastructure (same ASN/IP behind multiple distinct campaigns)	Monthly	Generalises Chapter 12.5

18.4.1 Tenant-wide click-then-sign-in correlation

Defender XDR | Every click, tenant-wide, followed by a sign-in from a different IP within 15 minutes

```
let riskyClicks = UrlClickEvents
    | where Timestamp > ago(1d)
    | where ActionType in ("ClickAllowed") or IsClickedThrough == true
    | project ClickTime = Timestamp, AccountUpn, ClickIP = IPAddress, Url;
riskyClicks
| join kind=inner (AADSignInEventsBeta | where Timestamp > ago(1d)) on AccountUpn
| where Timestamp between (ClickTime .. ClickTime + 15m)
| where IPAddress =≠ ClickIP
| summarize arg_min(Timestamp, *) by AccountUpn, ClickTime
| project AccountUpn, ClickTime, Url, SignInTime = Timestamp, IPAddress, Country, ErrorCode
| sort by ClickTime desc
```

18.4.2 Impossible travel following a click

Defender XDR | Geo-velocity check: same account, two countries, too little time between them

```
AADSignInEventsBeta
| where Timestamp > ago(1d)
| where ErrorCode == 0
| sort by AccountUpn, Timestamp asc
| serialize
| extend PrevCountry = prev(Country), PrevTime = prev(Timestamp), PrevUpn = prev(AccountUpn)
| where AccountUpn == PrevUpn and Country =≠ PrevCountry
| extend GapMinutes = datetime_diff('minute', Timestamp, PrevTime)
| where GapMinutes < 60 // two countries within an hour = physically implausible
| project AccountUpn, PrevCountry, Country, PrevTime, Timestamp, GapMinutes
```

18.4.3 HTML-smuggling attachment hunt

Defender XDR | HTML attachments carrying suspiciously large embedded payloads

```
EmailAttachmentInfo
| where Timestamp > ago(7d)
| where FileType in ("html", "htm")
| where FileSize > 51200 // 50 KB+ is large for a routine HTML attachment
| join kind=inner EmailEvents on NetworkMessageId
| project Timestamp, SenderFromAddress, RecipientEmailAddress, FileName, FileSize, DeliveryLocation
| sort by FileSize desc
```

18.4.4 Risky OAuth consent sweep

Defender XDR | New app consents carrying mail read/send scope, tenant-wide

```
CloudAppEvents
| where Timestamp > ago(7d)
| where ActionType has "Consent to application"
| extend Scopes = tostring(RawEventData.ModifiedProperties)
| where Scopes has_any ("Mail.Read", "Mail.Send", "offline_access", "Mail.ReadWrite")
| project Timestamp, AccountDisplayName, Application, Scopes, IPAddress
| sort by Timestamp desc
```

18.5 From hunt to automated detection

A hunt earns automation once it has produced true positives on at least two separate occasions with a manageable false-positive rate. Convert it into a Sentinel scheduled analytics rule (or a Defender XDR custom detection rule) with the same KQL, an appropriate run frequency, and a clear alert title that names the hypothesis, not just the query.

18.6 KQL performance tips for large tenants

- **Filter on time first.** Every query should open with a `Timestamp` filter before anything else.
- **Project early.** Drop unused columns immediately after the first filter to keep subsequent joins light.
- **Prefer `has` over `contains`** for whole-token matches — it uses the term index and is significantly faster on large tables.
- **Summarize before you join** where possible, so the join operates on an aggregated, smaller set.
- **Bucket with `bin()`** for trend queries rather than grouping on raw timestamps.

18.7 Techniques by role, and tools

ROLE	TECHNIQUE	TOOLS
Threat Hunter	Runs the hunting loop; owns the standing-hunt backlog and cadence.	Advanced Hunting, Sentinel notebooks/ workbooks
Security Engineer	Converts proven hunts into automated analytics rules.	Sentinel Analytics Rules, Defender custom detections
SOC L2/L3	Feeds candidate hypotheses from patterns noticed during routine investigation.	Case notes, hunt backlog

18.8 Coverage mapping

Track which MITRE ATT&CK techniques your standing hunts actually cover, using the ATT&CK Navigator (or an internal equivalent) to visualise gaps. A tenant with strong hunt coverage of Initial Access but nothing for Persistence or Collection is only watching half the kill chain described in Chapter 14.1.

18.9 How this feeds back into the workflow

Every confirmed finding from a hunt starts a new instance of the thirteen-phase workflow in Part Two, usually entering at Chapter 13 (blast-radius scoping) or Chapter 14 (post-compromise checks) rather than at Chapter 5, since the hunt itself already supplies the triage and much of the evidence.

CHAPTER 19

Real-World Investigation Case Studies

Three fictionalised but realistic cases, each shaped differently, walked through at speed to show the thirteen-phase workflow adapting to what the evidence actually looks like.

19.1 Case Study A — The Bulk Storage-Quota Campaign

Mass credential phishing, low sophistication MEDIUM

Background. An automated Defender for Office 365 alert fired for "email reported by user as malware or phish," naming a message with the subject "Your mailbox storage is full – action required."

Triage (Ch.5). Message trace showed **212 recipients** across the organisation, delivered to Inbox. No prior sibling in the tenant. Payload: one URL, no attachment. Severity set to Medium pending click data.

Header & authentication (Ch.7–8). Sender domain was ten days old; SPF and DKIM passed for that domain (attacker-owned), so DMARC passed too — a textbook lookalike-domain case, not a spoof.

Content and URL (Ch.9–10). Generic mail-merge greeting, no personalisation. The link led through one redirect to a credential-harvest page cloned from a generic webmail login.

Blast radius (Ch.13). 212 delivered, 34 opened, **9 clicked**, none showed a follow-on sign-in anomaly within the monitoring window.

EVIDENCE	VERDICT WEIGHT
Credential-harvest landing page confirmed	**Decisive (URL gate)**
No confirmed account compromise	**Lowers severity, does not lower verdict**

Verdict. Malicious, High confidence. **Response:** tenant-wide purge, domain/URL block, precautionary password reset for the 9 clickers, no session revocation needed (no compromise evidence). **Lesson learned.** Domain-age enrichment was not yet part of the automated anti-phish scoring; added to the detection backlog the same week (see Chapter 8.7's standing hunt).

19.2 Case Study B — The Gift-Card CEO Fraud

Payload-free business email compromise MEDIUM

Background. An HR coordinator received an email appearing to be from the company's CEO, sent from a personal Gmail address, asking her to buy six digital gift cards "for a client thank-you" and to send the codes by reply within the hour, while the CEO was "stuck in back-to-back calls."

Triage (Ch.5). No malicious URL, no attachment — nothing for automated filters to catch. Reported by the recipient herself, who found the request unusual. Severity: Medium, no technical payload.

Content analysis (Ch.9) — effectively the entire investigation. Every BEC checklist item fired: unusual request, urgency, an implicit request to avoid the normal purchasing process, and a channel that did not match how this executive normally communicated with this employee.

Authentication (Ch.8). Irrelevant to the verdict here — a personal Gmail account sending its own mail authenticates perfectly. This case is a clean illustration of why authentication passing proves authorisation, not trustworthiness.

EVIDENCE	VERDICT WEIGHT
Secrecy + urgency + unusual payment channel	**Decisive (content gate)**
Sender address does not match any known address for this executive	**Strong**

Verdict. Malicious, High confidence, reached from content evidence alone. **Response:** the employee had, correctly, not yet purchased anything; the SOC confirmed with the real CEO by phone within minutes, blocked the sender address, and briefed the finance and HR teams on the specific pretext. **Lesson learned.** Added to the next security-awareness cycle as a named example (with details generalised), and confirmed the organisation's out-of-band verification process was known and usable by non-finance staff too.

19.3 Case Study C — The AiTM Session Hijack of a Finance Executive

Adversary-in-the-Middle credential and session theft CRITICAL

Background. A Defender for Cloud Apps alert fired for "unfamiliar sign-in properties" on a Finance Director's account, forty minutes after Defender for Office 365 had logged a Safe Links click on the same account for a "document requires your signature" email.

Header & URL analysis (Ch.7, Ch.10). The email authenticated cleanly from a compromised partner mailbox — a real supplier had been compromised upstream. The link's redirect chain ended in a reverse-proxy page mirroring the organisation's actual single sign-on portal pixel-for-pixel.

Identity checks (Ch.14) — where this case was actually won or lost. Sign-in logs showed a successful authentication completing normal MFA, immediately followed by a sign-in from an unfamiliar ASN using the resulting session — no fresh MFA challenge on the second sign-in, the classic AiTM signature. Within the next twelve minutes, a new inbox rule appeared, moving any message containing "wire," "payment" or "urgent" to an obscure folder and marking it read.

EVIDENCE	VERDICT WEIGHT
Sign-in from new ASN with no fresh MFA challenge, minutes after the click	**Decisive (identity gate)**
Hidden inbox rule targeting financial keywords	**Decisive (identity gate)**

Verdict. Malicious, Critical severity, confirmed account takeover. **Response:** session revoked and password reset within eleven minutes of confirmation (Chapter 16.2); the inbox rule removed; a search for onward internal phishing came back clean; the compromised supplier was notified of their own breach. No wire transfer occurred — the rule had been active for under an hour when it was

found. **Lesson learned.** The organisation had no Conditional Access policy requiring a compliant, known device for finance-role sign-ins; this was added as a direct result, alongside the tenant-wide AiTM hunting query from Chapter 18.4.1 being promoted to a four-hourly automated rule.

CHAPTER 20

Attacker Tradecraft, Evasion Patterns and Analyst Pitfalls

The techniques keep evolving and the mistakes keep repeating. This chapter collects both in one place, drawn from patterns senior responders see over and over.

20.1 Evasion patterns beyond Chapter 3

Chapter 3.3 covered the core evasion families. These are newer or less obvious patterns worth recognising on sight.

PATTERN	HOW IT WORKS	WHY IT EVADES CONTROLS
Paste-and-run social engineering	A fake verification page tells the user their "browser needs an update" or they must "verify they are human," then walks them through opening a run dialog and pasting a command themselves.	No file is delivered at all — the user's own keyboard executes the payload, so there is nothing for a mail or web scanner to catch.
ISO/IMG nested in an archive	A disk-image file inside a ZIP, containing the real payload.	Windows does not apply Mark-of-the-Web to files extracted from a mounted disk image, silently defeating a common protection.
OneNote embedded objects	A `.one` file with an embedded script disguised behind a "Double-click to view" graphic.	OneNote's macro protections lag behind Office's, and the file type raises less suspicion.
Living off trusted domains	Hosting the lure or the payload on a genuine SaaS platform (form builders, e-signature tools, cloud storage, PDF generators).	The link's domain has an excellent reputation, because it is genuine — only the content placed on it is malicious.
Invisible-character obfuscation	Zero-width or homoglyph Unicode characters inserted into keywords or domains.	Defeats simple string-matching detections while rendering identically to the human eye.
Polyglot files	A single file that is simultaneously valid as two formats (e.g. a valid image and a valid archive).	Different tools in the analysis chain each see only the format they expect, and may disagree about what the file "is."

WARNING — RECOGNISE THE PATTERN, DON'T REPRODUCE THE MECHANICS

Knowing that "paste-and-run" pretexts exist is defensive knowledge every analyst needs. Analysing a specific sample your organisation received is legitimate incident response. Building or distributing working examples of these techniques is not something this guide will walk through, and isn't something a defensive investigation ever requires.

20.2 Fifteen mistakes beginner analysts make

#	MISTAKE	WHY IT'S A PROBLEM	DO THIS INSTEAD
1	Clicking the link directly "just to see"	Exposes your IP/identity to the attacker and can trigger cloaking that hides the kit from later, safer analysis.	Use an isolated VM or a URL-scanning service (Ch.10.1).
2	Treating a DMARC pass as "safe"	Proves authorisation, not trustworthiness — lookalike domains pass their own DMARC easily.	Always ask whether the passing domain is the real one (Ch.8.3).
3	Stopping at the reporting user	Misses the true blast radius; other recipients may already be compromised.	Always scope tenant-wide by `NetworkMessageId` or campaign cluster (Ch.13).
4	Asking the user to forward the email normally	Rewrites the header chain and can strip attachments — destroys the evidence.	Download from the entity page, or have the user attach it as a file (Ch. 6.1).
5	Ignoring payload-free BEC as "nothing to analyse"	The costliest fraud is often pure text with no technical IOC at all.	Apply the content/BEC checklist explicitly (Ch.9.4).
6	Resetting the password after a suspected AiTM click, and stopping there	A stolen session cookie remains valid after a password reset.	Always revoke sessions/refresh tokens as well (Ch.16.2).
7	Not checking for illicit OAuth app grants	These survive password resets and MFA re-registration entirely.	Check `CloudAppEvents` for consent grants on every confirmed compromise (Ch.14.6).
8	Treating a clean VirusTotal/ URLScan result as proof of safety	Fresh infrastructure has no reputation yet — absence of detections is not absence of risk.	Corroborate across several independent sources (Ch.12.6).
9	Closing the ticket with no root cause	Nothing improves; the same gap lets the next campaign through too.	Always answer "why did this reach the inbox?" (Ch.17.1).
10	Skipping chain-of-custody notes	Findings become hard to defend later, especially in HR, legal or insurance contexts.	Hash the sample and note who acquired it, when, and from where (Ch.6.2).
11	Judging by grammar quality alone	AI-assisted phishing is now frequently flawless in spelling and grammar.	Judge the ask, the lever and the channel, not the prose (Ch.9.2).
12	Pasting live indicators into a ticket or report	Creates a clickable phishing link inside your own case management system.	Defang everything, always: `hxxps://` , `[.]` (Ch.6.2).

#	MISTAKE	WHY IT'S A PROBLEM	DO THIS INSTEAD
13	Blocking an entire shared platform on one bad link	Causes collateral damage to every legitimate tenant on that platform.	Block the specific URL/path or file hash, not the whole domain, where the domain is shared infrastructure.
14	Closing once the mail is purged, without checking persistence	A hidden inbox rule or OAuth app means the attacker never actually left.	Run the full containment-completeness checklist for any confirmed click (Ch.16.5).
15	Not checking for onward internal phishing	A compromised account is frequently used to phish colleagues from a trusted internal address.	Always query `EmailDirection == "Intraorg"` from the affected sender (Ch.14.6).

20.3 Field wisdom from experienced responders

- **"The second click matters more than the first."** A user who clicks once made a mistake. A user who clicks the same style of lure repeatedly is a training gap and, potentially, a bigger future risk — track repeat-clicker rates, not just aggregate click rates.
- **"Your best IOC is often the thing the attacker didn't bother to fake."** Attackers polish the logo and the subject line. They rarely bother matching your organisation's actual invoice numbering scheme, PO format, or internal terminology — check those details.
- **"Write the ticket as if it will be read by someone unfamiliar with the case, a year from now."** Because eventually, it will be.
- **"A campaign that goes quiet for 48 hours often returns with a new domain."** Keep the enrichment watch running past the point the immediate ticket closes.
- **"Ask the user what they actually did, not just what happened to them."** "I clicked the link" and "I clicked the link and typed my password" require completely different responses.
- **"Escalate on one hard fact, not on a feeling."** A single decisive gate (Chapter 15.1) is a better reason to escalate than a vague sense that something looks off — find the fact first.
- **"If it's a quiet week, go hunting."** The tenants with the fewest surprises are usually the ones where someone hunts proactively between incidents (Chapter 18).

20.4 SOC and DFIR best-practice principles

- [ ] Assume the reporting user is not the only recipient, until the data proves otherwise
- [ ] Preserve evidence before you analyse it, every time, no exceptions
- [ ] Contain in parallel with analysis once exposure is confirmed — never wait for a perfect verdict
- [ ] Treat "no reputation data" as inconclusive, never as reassuring
- [ ] Lead every report with the two-sentence, non-technical version before the technical detail
- [ ] Track detection gaps as rigorously as you track the incidents that exposed them

CHAPTER 21

Hands-On Investigation Scenarios

Three practice cases, increasing in difficulty. Work each one from the given evidence before reading the solution — that's the only way this chapter actually builds skill.

21.1 Scenario 1 (Beginner) — "A vendor with a new number"

GIVEN EVIDENCE

Your ticketing system receives a user report. The email claims to be from `accounts@globex-supplies.com`, a real, existing vendor. The header block shows:

```
Return-Path: bounce@globex-suppIies-billing.net
From: "Globex Supplies Accounts" <accounts@globex-supplies.com>
Reply-To: globex.accounts.dept@outlook.com
Authentication-Results: spf=pass smtp.mailfrom=globex-suppIies-billing.net;
 dkim=pass header.d=globex-suppIies-billing.net; dmarc=fail action=quarantine
 header.from=globex-supplies.com; compauth=fail reason=001
Subject: Updated remittance details - please action before Friday
```

YOUR TASK

Before reading on: what does the `Return-Path` domain actually say, character by character, compared to the real vendor domain in `From:`? What does the DMARC result mean here, and what is your verdict?

SOLUTION WALKTHROUGH

Look closely at the Return-Path domain: `globex-suppIies-billing.net` — the fourth character group uses a capital `I` in place of a lowercase `l`, a classic homoglyph substitution that renders as "supplies" to the eye. This is a lookalike domain, not the vendor's real one. **DMARC fails** because neither SPF nor DKIM aligns with the genuine `globex-supplies.com` in the visible `From:` — this is domain spoofing, and the failure is exactly what should have stopped it (worth checking why it was still delivered — likely an allow-list override, per `compauth reason=001`). **Reply-To** diverts to a free Outlook address, unrelated to the vendor. **Verdict: Malicious, High confidence** — the authentication gate alone (Chapter 8, decisive) is sufficient; the Reply-To divergence and the "updated remittance details" pretext (Chapter 9) both corroborate.

Key takeaway: Read domains character by character, not at a glance. Homoglyphs are designed to survive exactly the kind of quick visual check most people do.

21.2 Scenario 2 (Intermediate) — "The PDF with no links"

GIVEN EVIDENCE

A message was sent to 60 recipients in Operations. `EmailUrlInfo` for the message returns **zero rows**. `EmailAttachmentInfo` shows one attachment: `MFA_Update_Required.pdf`, 240 KB. The email body reads only: *"Please see the attached notice regarding your multi-factor authentication enrolment."*

YOUR TASK

The email carries no URL at all — does that mean it's safe from a phishing-link perspective? What would you check inside the PDF before forming a verdict, and what telemetry would confirm impact if you're right?

SOLUTION WALKTHROUGH

Zero rows in `EmailUrlInfo` does **not** mean there is no link — it means there is no link Microsoft's parser found as text. A 240 KB PDF for a single sentence of text is unusually large: open it (in an isolated VM) and check for an embedded image. This is a **quishing** case — the PDF contains a QR code image with no accompanying clickable URL anywhere in the message (Chapter 10.1). Decoding the QR code reveals a credential-harvest URL. Because QR codes are typically scanned on a **personal mobile phone**, the relevant follow-up telemetry is not on the corporate endpoint: check `UrlClickEvents` for any hits at all (some organisations do capture the click if routed through Safe Links from a managed device), and check `AADSignInEventsBeta` for sign-ins from mobile clients in the relevant window, which is often the only trace this attack leaves in your telemetry (Chapter 13.5).

Key takeaway: "Zero URLs" is a data point, not a clean bill of health — always check whether a link exists as an image before concluding a message has no payload.

21.3 Scenario 3 (Advanced) — "Two sign-ins, one problem"

GIVEN EVIDENCE

TIME (UTC)	EVENT	DETAIL
14:02:11	URL click	`ActionType=ClickAllowed`, IP 41.203.x.x (user's normal home ISP)
14:03:47	Sign-in	IP 41.203.x.x, MFA satisfied via push notification, ErrorCode 0
14:11:22	Sign-in	IP 185.220.x.x (different country), `AuthenticationRequirement=singleFactorAuthentication`, ErrorCode 0
14:19:05	Audit event	`New-InboxRule`: moves mail containing "invoice", "payment" to folder "Archive 2", marks as read

YOUR TASK

Why is the second sign-in the critical line in this table, more than the click itself? What is your containment sequence, in order, starting right now?

SOLUTION WALKTHROUGH

The first sign-in at 14:03:47 looks entirely normal: same IP as the click, full MFA satisfied — this is the real user, on their own device. The second sign-in at 14:11:22, from a different country, requiring only **single-factor authentication**, is the signature: it means whatever authenticated at 14:11:22 already held a valid session artefact and was **not challenged for MFA again** — exactly what happens when an AiTM proxy captures the post-MFA session cookie from the first, genuine login and replays it (Chapter 10.2, Chapter 14.7). The inbox rule eight minutes later, hiding financial-keyword mail, confirms the attacker is already executing the BEC playbook (Chapter 14.1).

Containment sequence, in order (Chapter 16.2): (1) Revoke the session/refresh tokens immediately — this is the step that actually defeats a replayed cookie, more urgent than the password reset. (2) Force a password reset and MFA re-registration. (3) Remove the malicious inbox rule. (4) Check for OAuth app consents granted in the same window. (5) Search for any onward phishing sent from the account. (6) Notify the user and, if any payment was in flight, the finance team, immediately.

Key takeaway: In identity telemetry, the *authentication requirement* field matters as much as the IP address. A sign-in that skips MFA is a stronger AiTM signal than an unfamiliar location alone.

CHAPTER 22

MITRE ATT&CK Consolidated Reference

Every technique mapped throughout this guide, in one place, organised by tactic, with the chapter where each was discussed in depth.

22.1 How to use this reference

Use this chapter two ways: to look up a technique ID you've encountered and jump straight to the relevant investigation chapter, or to build a coverage map of which tactics your own detections and hunts actually address (see Chapter 18.8). The IDs below are current as of MITRE ATT&CK Enterprise, and each was independently verified against the framework while this guide was written.

22.2 Reconnaissance and Resource Development

TECHNIQUE	NAME	TACTIC	CHAPTER
T1589	Gather Victim Identity Information	Reconnaissance	1.2, 9
T1591	Gather Victim Org Information	Reconnaissance	1.2
T1598	Phishing for Information	Reconnaissance	9.6
T1583.001	Acquire Infrastructure: Domains	Resource Development	2.3, 12.3
T1583.003	Acquire Infrastructure: Virtual Private Server	Resource Development	12.3
T1584.004	Compromise Infrastructure: Server	Resource Development	12.3
T1585.001	Establish Accounts: Social Media Accounts	Resource Development	7.7
T1585.002	Establish Accounts: Email Accounts	Resource Development	7.7
T1586.002	Compromise Accounts: Email Accounts	Resource Development	7.7

22.3 Initial Access and Execution

TECHNIQUE	NAME	TACTIC	CHAPTER
T1566	Phishing	Initial Access	1, 3, 9.6
T1566.001	Spearphishing Attachment	Initial Access	3, 11.4

TECHNIQUE	NAME	TACTIC	CHAPTER
T1566.002	Spearphishing Link	Initial Access	3, 10.5
T1566.004	Spearphishing Voice	Initial Access	3 (TOAD)
T1189	Drive-by Compromise	Initial Access	10.5
T1204	User Execution	Execution	9.6
T1204.001	User Execution: Malicious Link	Execution	3, 10.5
T1204.002	User Execution: Malicious File	Execution	11.4
T1059.001	Command and Scripting Interpreter: PowerShell	Execution	11.4

22.4 Persistence and Defense Evasion

TECHNIQUE	NAME	TACTIC	CHAPTER
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys	Persistence	11.4
T1098.001	Account Manipulation: Additional Cloud Credentials	Persistence	14.4
T1098.003	Account Manipulation: Additional Cloud Roles	Persistence	14.4
T1114.003	Email Collection: Email Forwarding Rule	Collection	14.4
T1027	Obfuscated Files or Information	Defense Evasion	11.4
T1218	System Binary Proxy Execution	Defense Evasion	11.4
T1550.004	Use Alternate Authentication Material: Web Session Cookie	Defense Evasion	14.4
T1656	Impersonation	Defense Evasion	9.6, 10.5

22.5 Credential Access and Lateral Movement

TECHNIQUE	NAME	TACTIC	CHAPTER
T1056.003	Input Capture: Web Portal Capture	Collection, Credential Access	3.2

TECHNIQUE	NAME	TACTIC	CHAPTER
T1539	Steal Web Session Cookie	Credential Access	10.5
T1528	Steal Application Access Token	Credential Access	14.4
T1078.004	Valid Accounts: Cloud Accounts	Defense Evasion, Persistence, Privilege Escalation, Initial Access	14.4
T1534	Internal Spearphishing	Lateral Movement	14.4

22.6 Collection, Command and Control, and Impact

TECHNIQUE	NAME	TACTIC	CHAPTER
T1213	Data from Information Repositories	Collection	14.4
T1105	Ingress Tool Transfer	Command and Control	11.4
T1657	Financial Theft	Impact	1.2 (BEC objective)

PRO TIP — BUILD YOUR OWN COVERAGE HEAT-MAP

Colour each row above by whether you have (a) a preventive control, (b) a detection, and (c) a proven hunt covering it. Most organisations discover their coverage clusters heavily around Initial Access and thins out fast toward Lateral Movement and Collection — exactly where Chapter 14's checks live.

CHAPTER 23

Common Phishing Indicators and Red Flags Reference

Every red flag used throughout this guide, grouped by category, for a fast scan during live triage.

23.1 Header and authentication red flags

- [ ] Display name and the actual `From:` address don't match, or the domain is subtly misspelled
- [ ] `Return-Path` domain differs from the visible `From:` domain with no business reason
- [ ] `Reply-To` diverts to a free webmail address unrelated to the claimed sender
- [ ] DMARC fails, especially `p=reject` or `p=quarantine`, on your own domain or a well-known brand
- [ ] SPF/DKIM pass, but for a domain that is not the organisation the message claims to represent
- [ ] Received-chain hop count or geography inconsistent with the claimed sending infrastructure

23.2 Content and social-engineering red flags

- [ ] Artificial urgency ("within 24 hours," "immediately," "your account will be suspended")
- [ ] A request to keep the matter secret, or to bypass a normal approval/process step
- [ ] A channel switch request ("call/text me on this new number," "reply here instead of the usual thread")
- [ ] A new or changed payment detail, bank account, or beneficiary
- [ ] Generic greeting on a message that claims a personal relationship ("Dear Customer," "Dear User")
- [ ] Mail-merge tokens that failed to render (`%%FIRSTNAME%%` , `{{company}}`)
- [ ] Anchor text naming one domain while the underlying link points somewhere else entirely

23.3 URL red flags

- [ ] Multi-hop redirect chains, especially through an open redirect on an otherwise trusted domain
- [ ] Final landing page requests credentials for a well-known service on a domain that isn't that service's own
- [ ] The URL is personalised with the victim's email address and refuses to load without it
- [ ] Domain registered in the last 30 days, especially combined with a DV (domain-validated) certificate
- [ ] Page is pixel-identical to a real login flow — check whether it's an AiTM reverse proxy (Chapter 10.2)
- [ ] URL shortener or QR code hiding the true destination

23.4 Attachment red flags

- [ ] File's true type (by magic bytes) doesn't match its extension or icon
- [ ] Office document with an auto-executing macro, or "Enable Content" framed as required to "view" the document
- [ ] PDF containing embedded JavaScript or a launch action
- [ ] Password-protected archive with the password only ever supplied in the email body
- [ ] Unusually large HTML attachment for a short message (possible HTML smuggling)
- [ ] Disk image (ISO/IMG) nested inside a ZIP or RAR

23.5 Behavioural and identity red flags (post-click)

- [ ] Sign-in from a new country, ASN, or device shortly after a confirmed click
- [ ] A sign-in that completes with no fresh MFA challenge, immediately following one that did (AiTM signature)
- [ ] A new inbox rule that hides, moves, or deletes mail containing financial keywords
- [ ] A new OAuth application consent with broad mail scopes and an unverified publisher
- [ ] A new MFA method registered that the user did not initiate
- [ ] Mail sent internally, immediately after compromise, continuing the same pretext to new targets

23.6 How many red flags is "enough"?

Some red flags above are decisive alone — a failed DMARC check on your own domain, a confirmed AiTM landing page, a confirmed post-click sign-in anomaly. Most are corroborating: individually explainable, but persuasive together. Chapter 15's verdict cascade is the authoritative version of this logic; this chapter is the fast, scannable checklist to use while you're still gathering evidence.

CHAPTER 24

Investigation Checklists

Four checklists, meant to be used standalone during a live incident without flipping back through the whole book.

24.1 The L1 triage checklist (first five minutes)

- [ ] Confirm the report is genuine and pull the message via `NetworkMessageId`, not a forward
- [ ] Check delivery location: Inbox, Junk, or Quarantine
- [ ] Check `DeliveryAction`, `ThreatTypes`, and `ConfidenceLevel` already assigned by the platform
- [ ] Check sender domain age and SPF/DKIM/DMARC result at a glance
- [ ] Check whether other recipients exist, and whether anyone has clicked
- [ ] Assign an initial severity (Chapter 5.4) and escalate if any decisive signal is already visible

24.2 The full thirteen-phase master checklist

- [ ] **1. Intake & triage** — source identified, initial severity assigned (Ch.5)
- [ ] **2. Evidence acquired** — original message and headers preserved, hashed (Ch.6)
- [ ] **3. Headers analysed** — sender/return-path/reply-to consistency checked (Ch.7)
- [ ] **4. Authentication checked** — SPF/DKIM/DMARC results interpreted correctly (Ch.8)
- [ ] **5. Content analysed** — pretext, lever and ask identified (Ch.9)
- [ ] **6. URLs analysed** — full redirect chain traced, AiTM ruled in or out (Ch.10)
- [ ] **7. Attachments analysed** — static then dynamic, safely (Ch.11)
- [ ] **8. Threat intel enriched** — corroborated across independent sources (Ch.12)
- [ ] **9. Blast radius scoped** — full funnel built, tenant-wide (Ch.13)
- [ ] **10. Identity checked** — sign-ins, rules, OAuth apps, MFA registrations (Ch.14)
- [ ] **11. Verdict reached** — with confidence level and written rationale (Ch.15)
- [ ] **12. Contained, eradicated, recovered** — every item verified, not just actioned (Ch.16)
- [ ] **13. Reported and closed** — report filed, lessons captured with an owner and date (Ch.17)

24.3 Evidence-handling checklist

- [ ] Acquire the original message from the entity page or mail flow tool, never a manual forward
- [ ] Preserve full headers, exactly as received

- [ ] Hash every attachment before any analysis
- [ ] Record who acquired the evidence, when, and from which tool
- [ ] Store working copies separately from the original, untouched evidence

24.4 Containment-completeness checklist

- [ ] Mail purged tenant-wide, confirmed by re-query
- [ ] Sender domain/IP, landing URL, and file hash all blocked
- [ ] Sessions/refresh tokens revoked for any confirmed-compromised account
- [ ] Passwords reset and MFA re-registered where relevant
- [ ] Malicious inbox rules and SMTP forwarding removed
- [ ] Illicit OAuth app consents revoked
- [ ] Infected endpoints isolated and remediated
- [ ] Affected users notified; legal/privacy engaged if personal data exposure is plausible

24.5 Closure checklist

- [ ] Verdict recorded with supporting evidence and confidence level
- [ ] All containment items verified, not just actioned
- [ ] IOC table finalised and shared with relevant blocklists/TI platform
- [ ] Report written and filed
- [ ] At least one improvement identified, with an owner and a date, or an explicit documented reason not to act

CHAPTER 25

Quick-Reference Cheat Sheets

Five condensed pages meant to be kept open on a second monitor during a live investigation.

25.1 Header analysis cheat sheet

From:	The display name shown to the user. Never trust alone — check it against Return-Path and Authentication-Results.
Return-Path	Where bounces go; the true SMTP envelope sender. Compare its domain to the visible From: domain.
Reply-To	Where replies actually go. A mismatch with From: is one of the highest-value tells available.
Received chain	Read bottom-to-top for the true originating hop; top-to-bottom is presentation order only.
Authentication-Results	Contains spf=, dkim=, dmarc= and compauth=. Always check WHICH domain each passed for.
Message-ID	Should match the claimed sending infrastructure's typical format; useful for clustering.

25.2 SPF / DKIM / DMARC quick card

CHECK	ONE-LINE MEANING	IF IT FAILS
SPF	Is the sending IP authorised to send for the envelope-from domain?	Sending server wasn't on the domain's authorised list.
DKIM	Is the message body/header set cryptographically signed and untampered?	No valid signature, or signature doesn't match the claimed signing domain.
DMARC	Do SPF or DKIM *align* with the visible From: domain?	Neither aligns — the visible sender is very likely spoofed.

Remember: SPF/DKIM can legitimately pass for a domain that is *not* the one the message claims to represent (a lookalike domain sending its own, valid mail). Always ask "authenticated as whom?" — see Chapter 8.3.

25.3 Tool selection master table

TOOL	USE IT FOR	CHAPTER
VirusTotal	Hash, URL and domain reputation; relationship pivoting between IOCs.	11, 12

TOOL	USE IT FOR	CHAPTER
URLScan.io	Safe URL detonation, screenshot, DOM, and full redirect chain capture.	10
Any.Run / Hybrid Analysis	Interactive sandbox detonation of attachments and URLs; process-tree capture.	11
MXToolbox	SPF/DKIM/DMARC record lookup, blacklist checks.	8
WHOIS / RDAP	Domain registration date, registrar, registrant privacy status.	12
AbuseIPDB	IP reputation and abuse report history.	12
crt.sh (Certificate Transparency)	Finding sibling/typosquat domains sharing a certificate issuance pattern.	10, 12
CyberChef	Decode/encode/defang; base64, HTML entities, URL encoding.	9, 10
oletools / olevba	Office macro extraction and analysis without executing them.	11
pyzbar (or any QR decoder)	Decoding QR codes from attached images or PDFs safely.	10, 21
MISP	Internal/shared threat-intel platform; campaign clustering and sharing.	12, 18
Defender XDR – Threat Explorer	Message trace, recipient lists, one-click purge/ quarantine actions.	5, 16
Defender XDR – Advanced Hunting	KQL queries across email, identity, cloud-app and endpoint tables.	throughout
Entra ID sign-in logs / Identity Protection	Sign-in anomalies, risk detections, Conditional Access outcomes.	14
Defender for Cloud Apps	OAuth app governance, cloud activity, session policies.	14
Defender for Endpoint	Device isolation, indicator blocking, live response.	11, 16
Exchange Online PowerShell	Compliance search/purge, inbox rule and forwarding remediation.	16
Microsoft Graph PowerShell	Session/token revocation, forced password reset, OAuth grant removal.	16
ATT&CK Navigator	Visualising detection and hunt coverage across tactics.	18, 22

25.4 KQL quick-reference index

TABLE	MOST USEFUL FOR
`EmailEvents`	Delivery status, sender/recipient, cluster ID, confidence/threat scoring
`EmailUrlInfo`	Every URL carried by a message and where it sits in the body
`EmailAttachmentInfo`	Attachment name, type, hash; tenant-wide hash search
`UrlClickEvents`	Who clicked, from where, and whether Safe Links let it through
`AADSignInEventsBeta` / `SigninLogs`	Sign-in anomalies, MFA claims, impossible travel
`CloudAppEvents`	Inbox rules, OAuth consents, admin actions
`DeviceProcessEvents` / `DeviceNetworkEvents`	Attachment execution, process trees, C2 beacons
`ThreatIntelligenceIndicator`	Matching curated IOC feeds against tenant telemetry

25.5 Severity and decision cheat sheet

SEVERITY	TRIGGERING CONDITION
Critical	Confirmed account takeover, active malware execution, or imminent financial loss
High	Malicious verdict, confirmed clicks, no compromise confirmed yet
Medium	Malicious verdict, delivered but limited/no engagement
Low	Suspicious but unconfirmed, or benign with minor hygiene follow-up

Verdict logic in one line: **any one decisive gate (Ch.15.1) = Malicious.** No gate fired? Weigh corroborating signals (domain age, brand consistency, corroboration count) to reach Suspicious or Legitimate.

CHAPTER 26

Interview Questions and Answers

Twenty-eight questions across three difficulty tiers, drawn directly from the material in this guide — useful for interview preparation and for onboarding new analysts alike.

26.1 SOC L1 tier

Q. Q1. What is phishing, in one sentence, and how does it differ from ordinary spam?

A. Phishing is a deliberate attempt to manipulate a specific outcome — credentials, money, or execution — using a false identity, while spam is unsolicited but not necessarily deceptive about who sent it or what it wants (Chapter 1.1).

Q. Q2. What's the difference between the From: address and the Return-Path?

A. From: is the display sender shown to the user; Return-Path is the true envelope sender bounces go to. They can legitimately differ for mailing platforms, but an unexplained mismatch is a red flag (Chapter 7.3).

Q. Q3. What does SPF actually check?

A. Whether the IP address that sent the message is authorised, in DNS, to send mail for the envelope-from domain (Chapter 8.1).

Q. Q4. If SPF and DKIM both pass, is the email definitely safe?

A. No. They prove the message is authorised by whichever domain it claims to come from — which can be a lookalike domain the attacker legitimately owns. Always check which domain passed, not just that something passed (Chapter 8.3).

Q. Q5. What's the first thing you should do when a user reports a suspicious email?

A. Preserve it properly — pull it via the entity page or message trace, not a manual forward — then begin triage: confirm delivery location, check existing platform scoring, and see if others received it too (Chapter 5, Chapter 24.1).

Q. Q6. Why shouldn't you just forward a suspicious email normally to escalate it?

A. A normal forward can rewrite headers and strip or alter attachments, destroying exactly the evidence an investigation needs (Chapter 6.1).

Q. Q7. Name a content red flag in phishing that isn't a spelling mistake.

A. Artificial urgency combined with a request to bypass the normal process, or a request to keep the matter secret — AI-written phishing now often has flawless grammar (Chapter 9.2, 9.4).

Q. Q8. What does "defanging" a URL mean, and why do we do it?

A. Rewriting it so it can't be accidentally clicked — for example `hxxps://evil[.]com` — so that tickets, reports and chat messages never contain a live malicious link (Chapter 6.2).

Q. Q9. What are smishing, vishing and quishing?

A. Phishing delivered via SMS, voice call, and QR code respectively — the pretext and lever stay the same, only the channel changes (Chapter 3.1, Chapter 21.2).

Q. Q10. What's the first thing to check about an attachment, before opening anything?

A. Its true file type from magic bytes, not its extension or icon — and its hash against VirusTotal or an internal TI store, in case it's already known (Chapter 11.1).

26.2 SOC L2/L3 tier

Q. Q11. Explain SPF alignment vs. DKIM alignment in DMARC.

A. Alignment means the domain that passed SPF or DKIM matches (exactly, or per the organisational mode, by parent domain) the domain shown in the visible From: header. DMARC passes if either mechanism aligns; if neither does, the visible sender is very likely spoofed (Chapter 8.3).

Q. Q12. What is an AiTM attack, and why is it more dangerous than simple credential phishing?

A. A reverse proxy sits between the victim and the real login service, relaying traffic live so the victim completes genuine MFA — then the proxy steals the resulting session cookie. It's more dangerous because a password reset alone doesn't invalidate a stolen session; only revoking the session does (Chapter 10.2, Chapter 14.2).

Q. Q13. Walk through tracing a multi-hop redirect chain.

A. Defang the URL, submit it to a scanner such as URLScan.io rather than clicking it directly, follow each hop it records, and classify what the final landing page actually does — credential harvest, malware download, or an AiTM proxy (Chapter 10.1–10.3).

Q. Q14. Why do both static and dynamic attachment analysis, rather than just one?

A. Static analysis is safe and fast but can miss payloads that only reveal themselves on execution (e.g. downloaded second stages); dynamic detonation confirms real behaviour but carries more handling risk and time cost — static should always come first to inform how you detonate safely (Chapter 11.1–11.2).

Q. Q15. What would you check to determine if an account was compromised after a click?

A. Sign-in anomalies in the following minutes, new or modified inbox rules, new OAuth app consents, new MFA method registrations, and any onward mail sent internally from the account (Chapter 14.2).

Q. Q16. Why is a hidden inbox rule dangerous even after a password reset?

A. The rule itself, once created, keeps hiding or redirecting mail regardless of the account's password — it must be found and removed explicitly as a separate eradication step (Chapter 14.1, 16.1).

Q. Q17. What is HTML smuggling, and how would you detect it?

A. Embedding an encoded malicious payload inside an HTML attachment, assembled client-side by the browser so it never traverses the network as a recognisable file. Detect it by flagging unusually large HTML attachments relative to their apparent content (Chapter 18.4.3).

Q. Q18. How do you scope the blast radius of a campaign in Microsoft 365?

A. Pull every recipient by `NetworkMessageId` or campaign cluster, then layer in delivery location, click data, attachment execution, and any follow-on sign-in anomalies to build a funnel from "delivered" down to "confirmed compromised" (Chapter 13).

Q. Q19. What's the difference between a decisive and a corroborating verdict signal?

A. A decisive signal is sufficient on its own for a Malicious verdict (e.g. a confirmed AiTM proxy); a corroborating signal is individually weak but persuasive combined with others (e.g. a nine-day-old domain) (Chapter 15.1).

Q. Q20. Name two things that survive a password reset alone, and what actually removes them.

A. A stolen session cookie (removed only by revoking the session/refresh tokens) and an illicit OAuth app grant (removed only by explicitly revoking the consent) (Chapter 14.1, 16.2).

26.3 DFIR / Threat Hunter tier

Q. Q21. Design a hypothesis-driven hunt for AiTM activity across a large tenant.

A. Hypothesis: a session-replay sign-in follows a phishing click within minutes, from a different IP, without a fresh MFA challenge. Join `UrlClickEvents` to `AADSignInEventsBeta` on the account within a short time window, filter for a changed IP and a single-factor authentication requirement on the second sign-in, then validate against known-good travel patterns before converting to a standing detection (Chapter 18.4.1, Chapter 18.5).

Q. Q22. How would you attribute a campaign to a broader cluster with only a handful of IOCs?

A. Corroborate across independent sources — WHOIS/RDAP registration patterns, passive DNS co-hosting, certificate transparency for sibling domains, and internal/shared TI platform matches — treating agreement across sources as the real signal rather than any single one (Chapter 12.1, 12.5).

Q. Q23. How do you validate that a containment action actually took effect?

A. Re-query after the action: confirm no further deliveries from a blocked domain, no further sign-ins on a revoked session, and that a removed inbox rule no longer appears in the mailbox configuration — never treat "the command ran" as equivalent to "it worked" (Chapter 16.2).

Q. Q24. When do you convert a manual hunt into an automated detection rule?

A. Once it has produced true positives on at least two separate occasions with a manageable false-positive rate — earlier than that, you risk automating noise; later, you're leaving detection value on the table (Chapter 18.5).

Q. Q25. Walk through the MITRE ATT&CK chain for a typical AiTM-based BEC attack.

A. Reconnaissance (T1589/T1591) → infrastructure acquisition (T1583.001) → spearphishing link (T1566.002) → user execution (T1204.001) → web portal/session capture (T1056.003, T1539) → valid account sign-in (T1078.004) → persistence via forwarding rule (T1114.003) → financial theft as the ultimate impact (T1657) (Chapter 22).

Q. Q26. What's the risk of over-blocking on a single weak signal, and how do you avoid it?

A. Blocking shared infrastructure (a whole SaaS domain, an entire ESP) on one bad link causes collateral damage to unrelated legitimate senders. Scope blocks to the specific URL, path, or hash wherever the underlying domain is shared (Chapter 20.2, mistake #13).

Q. Q27. How would you build a metric that reflects phishing-response maturity, beyond raw counts?

A. Track MTTD/MTTC/MTTR alongside click-rate trend and repeat-pretext rate over time — raw incident counts say more about attacker volume than about your own response quality (Chapter 17.4).

Q. Q28. How would you write an incident report differently for an executive vs. a peer analyst?

A. The executive gets four to six plain-English sentences up front covering what happened, impact, and current status; the analyst gets the full phase-by-phase technical findings, IOC table, and ATT&CK chain underneath — both live in the same document, ordered so each reader stops exactly where their interest ends (Chapter 17.1).

CHAPTER 27

Glossary

Every acronym and term used in this handbook, in one alphabetical list.

AiTM (Adversary-in-the-Middle)
An attack where a reverse proxy sits between the victim and a real login service, relaying traffic live so it can capture the post-authentication session cookie. See Chapter 10.2.

ASN (Autonomous System Number)
An identifier for a block of internet routing space under one administrative control; useful for spotting when a sign-in or click comes from unfamiliar hosting.

ATT&CK
MITRE's globally-used knowledge base of adversary tactics and techniques, referenced throughout this guide as T-numbered technique IDs (for example, `T1566`). See Chapter 22.

BEC (Business Email Compromise)
Fraud conducted through email impersonation or account compromise, typically requesting a payment, gift cards, or a change to payment details, often with no technical payload at all. See Chapter 9.4.

Blast radius
The full set of recipients, clickers, and compromised accounts resulting from one campaign, tenant-wide. See Chapter 13.

C2 (Command and Control)
Infrastructure an attacker uses to communicate with and control malware after initial execution.

Compauth
Microsoft's composite authentication result, which factors SPF, DKIM, DMARC and sender reputation into one pass/fail/soft-fail verdict shown in the Authentication-Results header.

Defanging
Rewriting a URL, domain or IP so it cannot be accidentally clicked or resolved, e.g. `hxxps://evil[.]com`. See Chapter 6.2.

DFIR
Digital Forensics and Incident Response — the discipline (and often the team) responsible for confirmed, higher-impact incidents.

DKIM (DomainKeys Identified Mail)
An email authentication method that cryptographically signs message content so tampering can be detected. See Chapter 8.1.

DMARC (Domain-based Message Authentication, Reporting and Conformance)
A policy layer that requires SPF or DKIM to align with the visible From: domain, and tells receivers what to do on failure. See Chapter 8.1.

Dwell time
The time between an attacker gaining access and that access being detected or removed.

EOP (Exchange Online Protection)
Microsoft's baseline mail-filtering layer for Exchange Online, ahead of Defender for Office 365's deeper analysis.

HTML smuggling

Embedding an encoded malicious payload inside an HTML file, assembled client-side by the browser rather than transmitted as a recognisable file. See Chapter 18.4.3.

Homoglyph

A character that looks the same or very similar to another (e.g. a capital `I` and a lowercase `l`), used to build convincing lookalike domains. See Chapter 21.1.

IOC (Indicator of Compromise)

A concrete, observable artefact — a domain, IP, hash, or URL — associated with malicious activity.

KQL (Kusto Query Language)

The query language used across Microsoft Defender XDR's Advanced Hunting and Microsoft Sentinel.

LOLBins (Living-off-the-Land Binaries)

Legitimate, pre-installed system binaries abused by attackers to execute malicious actions while appearing to be normal system activity.

Lookalike domain

A domain designed to visually resemble a trusted one, through misspelling, homoglyphs, or added words, without being the same registered domain.

Magic bytes

The first few bytes of a file that identify its true type, independent of its file extension. See Chapter 11.1.

MFA (Multi-Factor Authentication)

Authentication requiring more than one proof of identity; can still be bypassed by session-token theft (see AiTM).

MISP

An open-source threat intelligence platform used for storing, correlating and sharing indicators and campaign clusters. See Chapter 12.1.

MTA (Mail Transfer Agent)

A server that relays email between systems, adding a `Received:` header at each hop. See Chapter 2.1.

MTTD / MTTC / MTTR

Mean Time to Detect / Contain / Resolve — core response-speed metrics. See Chapter 17.4.

MUA (Mail User Agent)

The email client the end user actually reads mail in (Outlook, a mobile app, a browser).

OAuth

An authorisation framework letting a user grant a third-party application access to their data without sharing their password — and a common post-compromise persistence mechanism when abused. See Chapter 14.1.

Passive DNS

A historical record of domain-to-IP resolutions over time, used to spot infrastructure reuse. See Chapter 12.1.

Quishing

Phishing delivered via a QR code, often to move the attack from a monitored corporate device to an unmonitored personal phone. See Chapter 21.2.

RDAP (Registration Data Access Protocol)

The modern, structured successor to WHOIS for looking up domain registration data.

Root cause

The underlying reason an attack succeeded or reached the inbox at all — the answer a good incident report must always include. See Chapter 17.1.

SCL (Spam Confidence Level)

A numeric score Exchange Online assigns a message, reflecting the likelihood it is spam.

Session cookie / refresh token

Artefacts issued after successful authentication that let a user (or an attacker who steals them) stay signed in without re-entering credentials or MFA. See Chapter 14.1.

Smishing / Vishing

Phishing delivered via SMS text message and voice call, respectively. See Chapter 3.1.

SOC (Security Operations Centre)

The team responsible for monitoring, triaging and responding to security alerts.

SPF (Sender Policy Framework)

An email authentication method publishing which IP addresses are authorised to send for a domain. See Chapter 8.1.

TOAD (Telephone-Oriented Attack Delivery)

An attack that starts by email but completes over a phone call, often with no malicious URL at all. See Chapter 3.1.

TTP (Tactics, Techniques and Procedures)

The behavioural patterns an attacker uses, at increasing levels of specificity — the basis of MITRE ATT&CK.

Typosquatting

Registering a domain that is a common misspelling or minor variation of a legitimate one.

UPN (User Principal Name)

A user's sign-in identity in Microsoft Entra ID, typically in email address format.

WHOIS

A protocol/service for looking up domain registration details such as registrar and creation date. See Chapter 12.1.

XDR (Extended Detection and Response)

A platform correlating signals across email, identity, endpoint and cloud apps into unified alerts and hunting data — Microsoft Defender XDR throughout this guide.

ZAP (Zero-hour Auto Purge)

A Microsoft Defender for Office 365 capability that retroactively removes mail already delivered once it is reclassified as malicious.

CHAPTER 28

References and Further Reading

Stable, authoritative sources to go deeper on any topic in this guide.

28.1 Standards and specifications

- **DMARC** — RFC 9989 (core protocol), RFC 9990 (aggregate reporting) and RFC 9991 (failure reporting), published by the IETF in May 2026 as Standards Track documents, together known as "DMARCbis." These obsolete the original RFC 7489 (2015, Informational) referenced in most pre-2026 literature; the record format (`v=DMARC1`) and core concepts are unchanged. See Chapter 8.1's note for what actually changed.
- **SPF** — RFC 7208, Sender Policy Framework for Authorizing Use of Domains in Email.
- **DKIM** — RFC 6376, DomainKeys Identified Mail (DKIM) Signatures.

28.2 Frameworks

- **MITRE ATT&CK** (attack.mitre.org) — the technique knowledge base referenced throughout Part Two and consolidated in Chapter 22.
- **MITRE D3FEND** (d3fend.mitre.org) — the defensive counterpart to ATT&CK, useful when mapping containment actions back to the techniques they counter.
- **NIST SP 800-61** — Computer Security Incident Handling Guide, the foundational reference for incident-response lifecycle and reporting structure underpinning Chapters 16–17.

28.3 Vendor and community documentation

- **Microsoft Learn** (learn.microsoft.com) — primary documentation for Microsoft Defender XDR, Advanced Hunting table schemas, Exchange Online PowerShell, and Microsoft Entra ID, all referenced across Part Two's KQL and PowerShell examples. Schemas and cmdlets evolve; always check current documentation against the queries in this guide before relying on them in production.
- **CISA** (cisa.gov) — U.S. Cybersecurity and Infrastructure Security Agency advisories on phishing campaigns and business email compromise.
- **SANS Institute** (sans.org) — incident-response and digital-forensics training resources referenced conceptually throughout Parts Two and Three.

> **A NOTE ON CURRENCY**
>
> Cloud platforms, KQL schemas, and even RFCs change. Treat the specific table names, cmdlets, and tool names in this guide as a strong, current starting point — not an assumption that they will never change again. Where it matters, verify against the vendor's current documentation before relying on a query or command in production.

Phishing Email Investigation Guide

From First Alert to Incident Closure

Written for SOC Analysts, Threat Hunters, DFIR Analysts, Incident Responders, Security Engineers, and students building a career in this field.

Author: Ganesha T
Cyber Security Consultant | Threat Hunter

Twenty-eight chapters · thirteen investigation phases · hands-on scenarios · hunting playbooks · quick-reference cheat sheets