PHISHING

RUNBOOK/PLAYBOOK

Phishing playbook guides SOC teams in detecting, analyzing, and responding to phishing threats.

- SOC phishing detection and response guide.
- Defines roles, triage, and investigation steps.
- Focuses on email, credential, and social engineering threats.
- Ensures quick containment and awareness.
- Promotes continuous improvement and prevention.

Kumar Bineet Ranjan

Contents

1. Introduction & Objective

This document is a comprehensive, end-to-end Phishing Incident Response Playbook designed for SOC Analysts and Incident Response teams.

It provides operational procedures, technical background, detection triggers, triage and analysis steps, containment playbooks, eradication and recovery guidance,

Forensics requirements, reporting templates, and automation scripts — all in one single master document.

Objectives:

- Reduce time to detect and contain phishing incidents.

- Provide repeatable, auditable steps for SOC teams.

- Preserve forensic evidence for legal or regulatory needs.

- Improve prevention and detection controls through lessons learned and automation.

Scope:

- Email (including attachments and links), SMS (smishing), Voice (vishing), QR (qrishing), social media phishing, waterhole attacks, and hybrid campaigns.

- Applies to corporate mailboxes, managed devices, cloud applications (Office 365 / Microsoft 365, G Suite), and corporate network resources.

2. Roles & Responsibilities

This section lists roles, responsibilities, and expected SLAs for phishing response.

Roles:

- L1 Analyst (Triage)

* Initial intake of alerts and user reports.
* Preserve artifacts (.eml, attachments, screenshots).
* Run quick authentication checks (SPF/DKIM/DMARC/ARC).
* Set ticket priority and escalate to L2 when criteria met.
* SLA: Initial triage within 15 minutes.

- L2 Analyst (Threat / Investigation)

* Full static and dynamic analysis, sandboxing, IOC extraction.
* Build SIEM hunts and assess scope.
* Recommend containment and remediation actions.
* SLA: Preliminary analysis within 2 hours for High incidents.

- L3 / Incident Response Lead

* Approve containment, coordinate cross-team remediation.
* Communication with Legal, PR, and Executive stakeholders.
* Final incident report and lessons learned.
* SLA: Containment decision within 2 hours for Critical incidents.

- Mail Gateway Admin

* Implement quarantine, blocklists, and authentication (SPF/DKIM/DMARC) configuration.
* Provide mailbox exports.

- **Network / Proxy Admin**

  * Block malicious domains/IPs at perimeter devices.

  * Provide PCAPs and proxy logs.

- **EDR / Endpoint Admin**

  * Quarantine endpoints, collect forensic artifacts, and assist with remediation.

- **Legal / Compliance / PR**

  * Assess legal obligations and notifications.

  * Prepare any external communications.

- **HR / Awareness**

  * Deliver user notifications and retraining as needed.

Escalation Matrix:

- L1 escalate to L2 if: (spf=fail and dkim=none) OR user_entered_credentials OR suspicious attachment executed.

- L2 escalate to L3 if: confirmed credential compromise, persistent malware, or executive targeting.

Documentation & Ticketing:

- Every step must be logged in the incident ticket with timestamps (IST), analyst name, artifacts collected, and actions taken.

3. Phishing Taxonomy — Full Details (16 Variants)

This section catalogs phishing types. For each type we describe: definition, attack flow, example message/header, indicators/IOCs, analyst action sequence (commands), sandbox expectations, SIEM hunts, containment, prevention, and user communication templates.

3.1 Email Phishing (Mass / Spray)

Definition:

Large-scale non-targeted email campaigns sent to vast lists. Typical goals: credential harvesting or commodity malware distribution.

Attack flow:

1. Register lookalike domain or use compromised MTA.

2. Craft generic message template.

3. Send via botnet or bulk mailer to harvested addresses.

4. Recipients click link or open attachment -> credential submission or payload execution.

Example header (inspect these fields):

Return-Path: <bounce@paypal-notice.com>

Received: from mail-198-51-100.example.net (198.51.100.77)

From: "PayPal" <no-reply@paypal.com>

Reply-To: support.helpdesk101@gmail.com

Authentication-Results: mx.company.com; spf=fail smtp.mailfrom=paypal-notice.com; dkim=none; dmarc=fail

Indicators / IOCs:

- Newly registered domain (<30 days).

- URLs containing brand name + verify/update tokens.

- Low detection on VT and URL reputation.

- Attachments with macros (.docm, .xlsm) or ZIP with EXE.

Analyst Action Sequence:

1. Export raw .eml (include full headers) and compute SHA256:

   sha256sum INC-2025-0001_email.eml

2. Extract headers:

   grep -i "Authentication-Results" INC-...eml

   grep -i "^Received:" INC-...eml | tail -n 5

3. DNS & Auth checks:

   dig +short TXT paypal-notice.com

   dig +short TXT _dmarc.paypal.com

4. WHOIS:

   whois paypal-notice.com | egrep -i 'Creation Date|Registered'

5. Submit URL/attachment to VirusTotal and URLScan

6. Quick SIEM hunt (Splunk):

   index=email sourcetype=o365 "paypal-notice.com" | stats count by RecipientEmailAddress, Subject

Sandbox expectations:

- HTML credential forms posting to /submit.php or similar.

- Macro-based attachments executing PowerShell -EncodedCommand and downloading payloads.

Containment:

- If credentials submitted OR malware executed: Critical -> isolate endpoints, reset credentials, escalate to IR Lead.

- If clicked only: quarantine similar mails and block domain; monitor for 72 hours.

Prevention:

- DMARC policy enforcement, gateway sandboxing, domain age blocking thresholds.

User template (clicked):

Subject: Security notice — suspicious link clicked

Hi <Name>,

You clicked a suspicious link. Do NOT enter any credentials. SOC has blocked the site and will monitor your account. Reply if you entered credentials.

3.2 Spear Phishing (Targeted)

Definition:

Targeted phishing tailored using OSINT (LinkedIn, public profiles) to impersonate colleagues or vendors.

Attack flow:

1. Reconnaissance (profiles, company sites).

2. Craft personalized message (project names, contacts).

3. Send to specific user(s) with malicious attachment or link.

Example:

From: accounts@trustedvendor.com

To: finance@company.com

Subject: Invoice INV-2025-018 — Project Helios

Attachment: INV-2025-018.xlsm

Indicators:

- Personalized content, accurate internal references.

- Macro-enabled attachment (.xlsm/.docm).

- Sender domain not matching historical vendor IPs or DKIM selector.

Analyst actions:

1. Preserve raw .eml & attachment, compute hashes.

2. Check vendor historical senders:

Splunk: index=email sender="accounts@trustedvendor.com" | stats count by src_ip

3. Extract macros using olevba:

olevba INV-2025-018.xlsm

4. Sandbox attachment with macros enabled in isolated VM. Capture process tree, network callbacks.

Sandbox artifacts:

- Macro => PowerShell => download payload => persistence (schtasks/service).

Containment:

- If payload observed: isolate host, collect memory & disk image, escalate to L3.

- If credential harvest: block domain, reset affected accounts.

Prevention:

- Strip macros at gateway, allow signed macros only, vendor verification process.

3.3 Whaling (Executive-targeted)

Definition:

Highly targeted attacks against executives/finance to request high-value transfers or confidential info.

Attack flow:

1. Gather exec-specific information.

2. Send urgent request disguised as CEO or internal exec.

3. Request out-of-process action (wire transfer).

Example:

From: "CEO" <ceo@company.com>

Subject: Confidential — Wire approval needed

Body: Kumar, wire 20,00,000 INR to vendor X now.

Indicators:

- Urgent language, out-of-procurement requests.

- May use internal display name with spoofed address.

- If DKIM/SPF pass, check for compromised account via sign-in logs.

Analyst actions:

1. Immediately call the purported sender using known internal number (not reply-to).

2. Preserve .eml and escalate to IR Lead.

3. Query sign-in logs for anomalous sign-ins:

SigninLogs | where UserPrincipalName == "ceo@company.com" and TimeGenerated > ago(24h)

Containment:

- Freeze payment approvals until verified.

- If mailbox compromised: reset password, revoke sessions, force MFA re-enroll.

Prevention:

- Enforce two-person approvals for wire transfers; out-of-band verification mandatory.

3.4 Business Email Compromise (BEC)

Definition:

Attacks aimed at financial fraud or data exfiltration by impersonating vendors or internal users.

Attack flow:

1. Impersonate vendor or compromise vendor mailbox.

2. Send payment detail updates or invoice modifications.

3. Target finance/payroll processes.

Indicators:

- Bank account change instructions.

- Minimal text, no attachments.

- Lookalike domains or compromised vendor accounts.

Analyst actions:

1. Preserve email chain and verify vendor via independent contact.

2. SIEM search for similar payment-change messages:

   index=email ("change bank" OR "update bank") | table _time, From, To, Subject

3. If funds transferred incorrectly, engage legal and bank.

Containment & Controls:

- Two-person verification for payment changes.

- Mandatory phone confirmation for vendor changes.

3.5 Smishing (SMS)

Definition:

SMS messages with malicious links or social-engineering prompts.

Attack flow:

1. SMS sent with short link or phone number.

2. User clicks link on mobile -> credential page or malicious app.

Analyst actions:

1. Request screenshot of SMS and originating number.

2. If device managed, check MDM logs and DNS queries.

3. Submit URL to URLScan and sandbox using mobile UA.

Containment:

- Block domain at DNS/proxy; push MDM policies.

- Reset credentials if submitted.

Prevention:

- MDM URL filtering and user education.

3.6 Vishing (Voice)

Definition:

Phone-based social engineering to extract OTPs, passwords, or convince users to install remote access.

Indicators:

- Caller pressure, request for OTP or passwords, caller-ID spoofing.

Analyst actions:

- Log caller ID and time; correlate with email/SMS incidents.

- If credentials/OTPs disclosed, reset and escalate.

Prevention:

- Policy: IT will never request passwords or OTPs over phone.

3.7 QRishing (QR Code Phishing)

Definition:

Malicious QR codes leading to phishing pages or APK downloads.

Attack flow:

1. QR code placed on poster/email/document.

2. User scans and visits malicious URL.

Analyst steps:

1. Decode QR to obtain URL (use online decoder or smartphone).

2. Submit to URLScan and sandbox.

3. Block domain and remove physical QR if in office.

Prevention:

- Disable auto-open on mobile; educate staff.

3.8 Pharming (DNS Poisoning)

Definition:

DNS or hosts-file manipulation redirects users to malicious servers even when typing correct domain.

Indicators:

- TLS certificate mismatch, different resolved IPs across users.

Analyst actions:

- nslookup domain (compare across resolvers)

- dig +trace domain

- openssl s_client -connect domain:443 -servername domain -showcerts

Containment:

- Contact DNS provider; rotate NS credentials; enable DNSSEC.

3.9 Clone Phishing

Definition:

Cloned legitimate emails with links replaced by malicious ones.

Detection:

- Compare Message-ID, original link targets, and timestamps.

Analyst actions:

- Find original cached email and compare.

- Sandbox replaced link and capture IOCs.

Containment:

- Quarantine clones and notify recipients.

3.10 Watering Hole / Drive-by

Definition:

Compromise of legitimate third-party sites frequented by target group to serve exploits or redirect to phishing.

Indicators:

- Multiple distinct users visiting same third-party just before compromise.

Analyst steps:

- Correlate proxy logs for common referrer.

- Sandbox the compromised site; capture JS and redirect chain.

Containment:

- Block site and notify site owner.

3.11 Malicious Attachments (Macro/Executable)

Definition:

Attachments with macros or embedded executables that download and run payloads.

Analyst commands:

- olevba sample.docm

- python -c "import base64; print(base64.b64decode('<b64>'))"

- strings and yara on binaries

EDR hunts:

- Search for powershell -enc usage and common download utilities (bitsadmin, certutil).

Containment:

- Isolate host, collect memory (winpmem), and image disk.

3.12 Credential Harvesting (Web-form Phishing)

Definition:

Fake login pages mimicking real services to collect credentials.

Evidence to capture:

- POST endpoint, field names, final collector IPs, cookies.

Commands:

- curl -sL "<url>" -o page.html

- grep -i "<form" page.html -n

- grep -Eo 'name="[^"]+"' page.html | sort -u

Containment:

- Block collector domain; reset submitted accounts; enforce MFA.

3.13 Hybrid (Multi-channel) & 3.14 Angler (Social Media)

Definition:

Multi-channel campaigns combining email, SMS, and calls; Angler phishing uses social channels impersonating customer service.

Analyst actions:

- Correlate across mail, SMS, telephony logs, and social monitoring.

- Request takedown for social impersonators; notify customers and block domains/numbers.

Prevention:

- Brand monitoring, register domain variants, and verified social profiles.

3.15 Registrar/DNS Compromise (Pharming variant)

Definition:

Registrar account compromise resulting in authoritative DNS changes and global redirect.

Detection:

- Nameserver changes and domain resolving to attacker-controlled IP.

Actions:

- Contact registrar to lock domain and restore records; rotate DNS credentials.

3.16 Cross-type Meta-signals

Meta-signals that indicate high risk:

- Reply-To != From combined with body containing credential keywords (password, login).

- Authentication-Results showing spf=fail and dmarc=fail for a known brand.

- Landing pages hosted on reputable cloud but POST to low-reputation collectors.

Suggested SIEM meta-rule (Splunk):

index=email ("Authentication-Results"="*spf=fail*" OR "dmarc=fail") OR (Reply-To!=From AND body="*password*")

| stats count by SenderFromAddress, RecipientEmailAddress, Subject

4. Email Authentication Deep Dive (SPF / DKIM / DMARC / ARC)

Purpose: give analysts a precise, protocol-level understanding of how authentication is applied and validated during mail transfer, what header fields show, why checks succeed/fail, how attackers commonly exploit the mechanics, and exactly how to validate & triage evidence.

---

1) Mail transfer & where authentication happens (SMTP walkthrough)

An SMTP session looks like:

- TCP connect from client (sending MTA) → server (receiving MTA). The receiving MTA records the **source IP** (the connecting IP).
- SMTP handshake: EHLO / HELO. The HELO identity is often logged in Received: but is not trusted.
- MAIL FROM:<bounce@domain.com> — this sets the **envelope sender** (Return-Path) that SPF will be evaluated against.
- RCPT TO:<recipient@dest.com> — recipients list.
- DATA — message headers + body transmitted. At the sender, if DKIM is enabled, the sending MTA usually signs the message **during** or immediately after DATA (it inserts the DKIM-Signature: header before sending).
- Receiving MTA receives full message, then performs **SPF**, **DKIM verification**, **DMARC** evaluation (which combines SPF/DKIM alignment with From:), and possibly stores Authentication-Results: in headers or logs the result.

Key point: SPF uses the *connecting IP* + *MAIL FROM*; DKIM signs message content (headers+body) and is checked after message is received and canonicalized; DMARC inspects results and alignment to the From: header.

---

2) SPF — deep mechanics & evaluation

What SPF actually contains (DNS)

An SPF record is a DNS TXT string like:

"v=spf1 ip4:198.51.100.0/24 a mx include:spf.vendor.net -all"

- v=spf1 — version marker.

- ip4:198.51.100.0/24 — this CIDR is allowed.
- a — the domain's A record addresses are allowed.
- mx — the domain's MX records IPs are allowed.
- include:spf.vendor.net — check the SPF of spf.vendor.net for allowed IPs (useful for third-party senders).
- -all — explicit fail if nothing matched.

SPF evaluation algorithm (simplified)

- Receiver sees connecting IP X and envelope MAIL FROM domain D.
- Resolve D's TXT records; parse SPF record; evaluate mechanisms left-to-right.
- For each mechanism:
    - ip4 / ip6: check if IP falls in CIDR.
    - a: resolve A record(s) of D and compare.
    - mx: resolve MX records of D, then resolve each MX host to IP(s).
    - include: recursively evaluate the included domain's SPF record (each include may cause additional DNS lookups).
- If a mechanism matches, return the mechanism's qualifier result (+ pass, - fail, ~ softfail, ? neutral).
- If no mechanism matched and all is present, return its qualifier (commonly -all fail).
- If an error occurs (DNS timeout, malformed record) the result could be permerror or temperror.

Important operational limits

- **10 DNS-lookup limit**: SPF expands includes/A/MX lookups, and RFC limits SPF evaluation to 10 DNS lookups to avoid amplification loops. If evaluation would exceed this, result is permerror.
- **Macro expansion**: SPF supports macros (e.g., %{i} for client IP, %{s} for sender) — rarely used by ordinary domains but can appear in advanced policies.
- **SPF checks envelope only**: SPF does *not* check From: header shown to users. This is why spoofing can use a benign MAIL FROM while forging the visible From:.

Example SPF evaluation (concrete)

- SMTP connecting IP: 203.0.113.77
- MAIL FROM: bounce@amazn-billing.com
- SPF record: v=spf1 ip4:198.51.100.0/24 include:spf.protection.outlook.com -all
- 203.0.113.77 not in 198.51.100.0/24; check include:spf.protection.outlook.com → if not included → -all triggers → **SPF=fail**.

SOC implications & checks

- **Check Authentication-Results**: spf=fail smtp.mailfrom=amazn-billing.com.

- **If mail forwarded**: SPF often fails because forwarder IP is not in origin SPF — expect forwarder-induced SPF failures. Use ARC/SRS to help discriminate.
- **Hard fail vs softfail**: `-all` (hard fail) is actionable (reject/quarantine); `~all` (softfail) is a weaker signal.

---

3) DKIM — deep mechanics, canonicalization & verification

DKIM key & DNS layout

- DKIM uses a key pair: private key held by sending MTA; public key published in DNS at `selector._domainkey.example.com`.
- Example DNS TXT for `s=mail2025` and `d=example.com`:

```
mail2025._domainkey.example.com.  TXT  "v=DKIM1; k=rsa; p=MIGfMA0GCSqGSI..."
```

- `p=` is the base64 RSA public key.

DKIM signature header explained (fields)

A typical header:

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=example.com; s=mail2025;
 h=from:to:subject:date:message-id; bh=<bodyhash>; b=<signature>; t=1600000000; x=1600003600
```

- `v` version, `a` algorithm, `c` canonicalization for header/body, `d` signing domain, `s` selector.
- `h` list of signed headers (order matters).
- `bh` base64 body hash; `b` the signature over signed headers and body hash.
- `t` signature timestamp; `x` signature expiration (optional).

Canonicalization (why it matters)

- **simple**: exact match — any whitespace/linebreak/rewrap changes lead to verification failure.
- **relaxed**: normalizes header names to lowercase, collapses whitespace, strips trailing spaces — tolerant to common mailbox transformations.
- Typical real-world uses: `c=relaxed/relaxed` (headers relaxed, body relaxed).

DKIM verification flow

1. Receiver reads `DKIM-Signature`, gets `d` and `s`.
2. Fetch `s._domainkey.d` public key from DNS.
3. Canonicalize the body and headers per `c=` and compute body hash — compare to `bh=`.
4. Verify signature `b=` using public key and header hash.

5. If verification succeeds → dkim=pass; else fail.

Edge cases & details

- **Multiple DKIM signatures:** a message may include multiple DKIM-Signature headers (multiple selectors). Each can be verified independently.
- **Signed headers list**: if h= does not include subject or date, attacker could have modified those fields and still produce a valid signature. Check h= list for crucial headers (From ideally included).
- **Signature expiration & timestamp**: t and x allow receivers to detect expired or replayed signatures.

Example (concrete)

- Sender signs with d=amazn-billing.com, s=mail2025. DNS contains mail2025._domainkey.amazn-billing.com with public key p=.... Receiver canonicalizes and verifies — passes → dkim=pass header.d=amazn-billing.com.

SOC implications

- **If DKIM=pass and d= aligns with From: domain**, this is strong evidence the message content wasn't tampered and the signing domain authorized the send. Still check domain age and sender intent.
- **If DKIM fails and mailing list/footer modified message**, consider c=relaxed/relaxed scenarios and check ARC.
- **If DKIM present but d= is a suspicious domain** — the signer domain may be attacker-controlled and can sign its own malicious messages.

---

4) DMARC — deep mechanics & reporting

DMARC DNS record and tags

Typical:

_dmarc.example.com. TXT "v=DMARC1; p=quarantine; rua=mailto:dmarc-agg@example.com; ruf=mailto:dmarc-afrf@example.com; adkim=r; aspf=r; pct=100"

- p= policy (none/quarantine/reject).
- rua= aggregate report URI (mailto:...).
- ruf= forensic (ARF) report URI.
- adkim (DKIM alignment mode: s strict or r relaxed).
- aspf (SPF alignment mode).
- pct (apply policy to percentage of messages).

DMARC evaluation steps

1. Receiver gets DKIM / SPF results.
2. For DKIM: does DKIM pass and is d= aligned to From: (exact match for strict, or organizational domain match for relaxed)?
3. For SPF: does SPF pass and is MAIL FROM aligned with From: per alignment mode?
4. If either DKIM or SPF passes *and* aligns → DMARC PASS. Else DMARC FAIL → apply p= policy.

RUA / RUF reports

- **RUA**: aggregate XML reports (daily summary) that include sending IPs, counts, SPF/DKIM results — good for diagnosing abuse.
- **RUF**: forensic/ARF reports — may contain full message headers/bodies (privacy concerns). Many receivers don't send RUF for privacy.

Example DMARC decision

- From: = amazon.com
- DKIM: d= = amazn-billing.com (not aligned)
- SPF: MAIL FROM = amazn-billing.com, SPF fails.
  → DMARC FAIL.

SOC implications

- **If DMARC p=reject** on sender domain and mail still reaches inbox, your gateway may be misconfigured (not enforcing DMARC) — investigate.
- **Use RUA** to see volume and sources of spoofing — add frequent offenders to blocklists.
- **pct** tag can be used to gradually roll out enforcement.

---

5) ARC — how it preserves authentication through forwarding

Why forwarding breaks auth

- When a message is forwarded, the forwarder's sending IP becomes sender to the final recipient — that IP probably isn't listed in original domain SPF → SPF fail. If forwarder rewrites parts of the message (adds footer), DKIM may break.

ARC mechanism (high level)

- Each intermediate party that handles the mail adds an **ARC set** (three headers): ARC-Authentication-Results, ARC-Message-Signature, ARC-Seal. Each set is numbered: i=1, i=2, etc.

- ARC-Authentication-Results captures authentication results at that hop (what that hop saw).
- ARC-Message-Signature signs the message at that hop.
- ARC-Seal signs the ARC set chain.
- Final receiver evaluates ARC sets and can trust an earlier hop's authentication when the chain of ARC seals is valid and the forwarder is trusted.

SOC implications

- When SPF/DKIM fail but ARC exists, check ARC-Authentication-Results to see if an earlier hop validated the message. Evaluate whether the forwarder is a known/trusted forwarder (mailing list provider) before trusting ARC. ARC is not a panacea — it requires the final receiver to accept ARC and to judge trustfulness.

---

6) Why authentication checks can be inconclusive (and how attackers abuse that)

Common real-world reasons for ambiguous auth:

- **Forwarding** → SPF fails; DKIM may break if content altered.
- **Third-party senders (vendors)** → SPF may not include third-party IPs; vendor DKIM may sign with vendor domain (d=vendor.com) and DMARC alignment fails unless vendor signs on behalf of your domain (delegated signing).
- **Short-lived domains** → low reputation and VT detection but technically valid SPF/DKIM.
- **Compromised legitimate accounts** → attacker sends via legitimate mailbox; SPF/DKIM/DMARC pass because the account is valid.
- **Signing by attacker-controlled domain** → DKIM may pass but d= is not your brand; DMARC may still fail if not aligned.

Attacker evasion techniques mapped to auth logic

- **Use of compromised vendor/infrastructure** → auth checks may pass; content/content-analysis must detect malice.
- **Host credential pages on reputable cloud providers** (e.g., forms.gle, s3.amazonaws.com) — reputation of provider can shield the URL. Detection should include form analysis (fields named password) and redirect chains.
- **Open redirect abuse**: attacker uses legitimate site redirect to final landing; initial URL looks safe but final domain is bad. Use redirect tracing.
- **Reply-To diversion**: From: may look legitimate but Reply-To: points to attacker-controlled address — social engineering for responses.

---

7) How to check & validate authentication as an analyst (practical steps & commands)

7.1 — Extract header evidence

- Save raw .eml. Look for:
  - Authentication-Results: (may show SPF/DKIM/DMARC and sometimes action=).
  - DKIM-Signature: header(s).
  - Received: chain (bottom-most external Received: is first external hop).
  - Return-Path: (envelope MAIL FROM).

Example extraction:

```
# show auth header
grep -i "Authentication-Results" INC-123_email.eml

# list Received lines (bottom-most external hop)
grep -i "^Received:" INC-123_email.eml | sed -n '1,200p' | tail -n 10
```

7.2 — DNS lookups

- SPF:

```
dig +short TXT amazn-billing.com  # look for v=spf1
```

- DKIM:
  - Extract s= selector from DKIM-Signature header, then:

```
dig +short TXT mail2025._domainkey.amazn-billing.com
```

- DMARC:

```
dig +short TXT _dmarc.amazn-billing.com
```

7.3 — DKIM verification manual (high-level)

- Compute the canonicalized body hash as specified and compare to bh=. Use an available DKIM verifier (many mailtools include this). Manual verification is complex; use library/tool (e.g., opendkim-testmsg, Python dkim module).

7.4 — WHOIS & domain age

```
whois amazn-billing.com | egrep 'Creation Date|Created|Registered On'
```

7.5 — Use RUA data if you're domain owner

- Collect aggregate RUA XML; parse to find sending IPs, counts, and alignment results. This helps identify widespread spoofing.

---

8) Recommended org-side settings & best practices (practical configuration guidance)

SPF

- Keep SPF compact and accurate; avoid `ptr`. Use `include:` for third-party senders and verify includes don't exceed DNS lookup limit.
- Use `-all` only after testing (`~all` during rollout).

DKIM

- Use `rsa-sha256` with minimum 2048-bit keys (at least 1024-bit but 2048 preferred). Rotate keys periodically.
- Ensure your `h=` includes critical headers: `from`, `subject`, `date`. Use `relaxed/relaxed` canonicalization for robustness.

DMARC

- Deploy DMARC in phases: `p=none` with `rua` to collect reports → analyze → `p=quarantine` → finally `p=reject`.
- Use `pct` to ramp policy application gradually. Ensure third-party vendor email flows are mapped and include vendor IPs or delegated DKIM properly.

ARC / Forwarders

- Deploy ARC on trusted forwarders/mail relays to preserve auth across legitimate forwarding. Use SRS (Sender Rewriting Scheme) on forwarders if possible.

---

9) Examples — annotated headers & sample outputs

Example 1 — Spoofed email header (interpreted)

Received: from smtp.somehost.net (203.0.113.77) by mx.company.com ...
Return-Path: <bounce@amazn-billing.com>
From: "Amazon Billing" <support@amazon.com>
Authentication-Results: mx.company.com; spf=fail smtp.mailfrom=amazn-billing.com; dkim=none; dmarc=fail action=quarantine

Interpretation: Envelope domain `amazn-billing.com` failed SPF; no DKIM; visible `From:` is `amazon.com` (forgery). DMARC fails — treat as spoof.

Example 2 — DKIM pass, SPF fail, DMARC pass (delegated signing)

DKIM-Signature: d=vendor.com; s=key1; ...
Authentication-Results: mx.company.com; spf=fail smtp.mailfrom=mailer.vendor.com; dkim=pass header.d=vendor.com; dmarc=pass
From: notifications@company.com

Interpretation: Vendor signs on behalf of company (delegated signing); DKIM passes and aligns to From: (or company uses d= alignment); DMARC passes even though SPF fails. Contact vendor to verify.

Example 3 — ARC chain present

```
ARC-Seal: i=1; cv=none; ...
ARC-Authentication-Results: i=1; mx.forwarder.com; spf=pass; dkim=pass; dmarc=pass
Authentication-Results: mx.company.com; spf=fail; dkim=none; dmarc=fail
```

Interpretation: The forwarder performed SPF/DKIM and DMARC checks successfully and sealed them with ARC; the final receiver shows SPF/DKIM fail because of forwarding, but ARC proves earlier authenticity — evaluate the trustworthiness of the forwarder before accepting.

---

10) Attacker checklist — how they exploit each mechanism

- **SPF bypass**: send mail via compromised legitimate MTA whose IP is allowed; or set MAIL FROM to domain they control and use display From: to impersonate brand.
- **DKIM bypass**: sign with attacker domain (DKIM passes but d= != From:) or compromise a legitimate signing key.
- **DMARC bypass**: use third-party signed mail (vendor signs on your behalf) or exploit relaxed alignment.
- **Forwarder abuse**: use thousands of forwards or exploit mailing lists to cause checks to break; use ARC if available to appear legitimate.

---

11) SOC detection & playbook changes informed by these internals

Because authentication checks are nuanced, SOC playbook should:

1. **Always gather full headers & ARC sets** before acting — because forwarding can cause SPF/DKIM to fail falsely.
2. **Use multi-signal detection**: combine auth results with domain age, WHOIS, URI analysis, sandbox behavior, and user action (clicks).
3. **Use RUA data proactively** (domain owners): mode detection for spoofing sources — add automatic blocking for IPs appearing repeatedly in RUA.
4. **Detect vendor-signed legitimate mail with content checks** — if DKIM passes but content requests credentials, treat as suspicious and verify with vendor/out-of-band.
5. **Create SIEM rules that include ARC**: if ARC-Authentication-Results shows pass and forwarder is trusted, deprioritize; else raise priority.
6. **Alert on SPF over-limit errors** (permerror due to >10 DNS lookups) — misconfigured SPF may hide real auth.

---

12) Recommended configuration snippets (for defenders)

SPF example (include vendors):

v=spf1 ip4:198.51.100.0/24 include:spf.protection.outlook.com include:_spf.sendgrid.net -all

DKIM key guidance:

- Use rsa-sha256, 2048-bit keys: publish p= in DNS; rotate yearly or on suspected key exposure.

DMARC staged rollout:

1. _dmarc.example.com TXT "v=DMARC1; p=none; rua=mailto:agg@ex.com; aspf=r; adkim=r; pct=100" (monitor)
2. After review, p=quarantine for 30 days.
3. Finally p=reject once vendor flows are validated.

---

13) Final: FAQ & common analyst pitfalls

Q: SPF=fail → Is it always phishing?
A: No — forwarding commonly causes SPF to fail. Check for ARC and forwarder identity. If no ARC and spoofed From:, treat as suspicious.

Q: DKIM=pass → Can I trust the message?
A: DKIM proves the signing domain authorized the message — but if d= is not the expected brand or the domain is newly registered, be cautious. Verify alignment and vendor relationships.

Q: DMARC=none → Is it safe?
A: No — p=none means the domain owner is only *monitoring*. Lots of domains still use p=none. Use RUA reports and other signals.

Q: What about Cloud-hosted phishing pages?
A: Cloud providers are often abused. Detect by form content (password fields), redirect chains, and hosting owner reputation — block the final landing domain and report to provider.

5. Detection & Triage — Step-by-step (Analyst Playbook) — expanded

Goal: Triage every reported phishing message quickly, preserve evidence intact, determine immediate risk (credential capture / malware / exec target), and escalate appropriately.

Phase: Intake (0–5 minutes)

Purpose: Capture who reported it, when, and gather the raw evidence before anything changes. Fast intake prevents accidental forwarding/interaction which can destroy useful metadata.

Actions (detailed):

1. **Record reporter, timestamp (IST), assign incident ID**
    - Format: INC-YYYYMMDD-#### (e.g., INC-20251019-0001).
    - Ticket fields to fill immediately: Reporter name/email, Report method (user/phishing mailbox/automated), Report timestamp (IST), Assigned analyst.
    - Why: consistent naming helps automation, correlation, and audit trails.
2. **Pull raw .eml from mail system and save to evidence store (do not forward)**
    - How: Export the original message (raw MIME). In O365: use eDiscovery or Message Trace to obtain raw MIME; in Exchange on-prem use Export-Mailbox/Search-Mailbox methods. If using a web UI, use "Download original message" or save as .eml.
    - Evidence naming convention: INC-20251019-0001_original.eml.
    - Why: .eml contains full headers and Received chain that are often lost when messages are forwarded or copied.
3. **Capture screenshots**
    - Minimum: List view (shows sender & subject), opened email (content), hover link preview (show the real URL preview).
    - Save images as PNG: INC-20251019-0001_screenshot_list.png, _open.png, _linkhover.png.
    - Why: Screenshots preserve how the user saw the email (useful for user education, legal evidence, or when raw message is later sanitized).

Pitfalls:

- Never forward the suspicious mail to other inboxes without redaction — forwarding rewrites headers. Use attachments or raw .eml files in the evidence store.

Phase: Preserve (0–10 minutes)

Purpose: Preserve artifacts in forensically sound manner and record integrity (hashes) for chain of custody.

Actions:

1. **Save attachments as original files; compute SHA256**
    - Extract attachments from .eml (tools: munpack, ripmime, python email libraries, or Exchange export).
    - Command example (Linux): ripmime -i INC-20251019-0001.eml -d artifacts/
    - Compute hash: sha256sum artifact.docx > artifact.docx.sha256
    - Save original filenames as sent, plus evidence name: INC-20251019-0001_attachment_invoice.docx.
2. **Save .eml and attachments to read-only evidence repository**
    - Evidence repo: network share or forensic vault with controlled write/admin policy. Set permissions to read-only for analysts.
    - Recommended: preserve original file ACLs and timestamps; if copying to a repository that changes metadata, document the difference in the ticket.
3. **Log artifact paths and SHA256 in ticket**
    - Ticket fields: path, filename, SHA256, extracted file type, who saved it, timestamp.
    - Example entry:
    - Evidence:
    - - /evidence/INC-20251019-0001/INC-20251019-0001_original.eml SHA256: abc...
    - - /evidence/INC-20251019-0001/invoice.docx SHA256: def...

Why/Outcome:

- Chain of custody and ability to re-run static/dynamic analysis later with identical artifacts.

Phase: Quick Authentication & Header checks (0–15 minutes)

Purpose: Rapidly determine if message passed SPF/DKIM/DMARC and identify suspicious relay hops or external IPs.

Checklist + concrete commands (run against the saved .eml):

1. **Extract Authentication-Results**
    - Command: grep -i "Authentication-Results" INC-20251019-0001.eml
    - Read results: look for spf=pass/neutral/fail, dkim=pass/fail, dmarc=pass/quarantine/reject.
    - Interpretation:
        - spf=pass with envelope-from matching your org domain is good; fail or neutral is suspicious.
        - dkim=fail for a signed message — could be tampered or wrong selector.
2. **Extract Received chain and identify external IPs**
    - Command: grep -i "^Received:" INC-20251019-0001.eml | tail -n 10
    - Parse Received lines from top to bottom to find first external relay IPs (first hop from outside your org). Note IPs and ASN/Geo.
    - Tools to enrich: whois, curl ipinfo.io/<ip> (if allowed), Passive DNS.
3. **Run DNS commands (replace <domain> and <selector> appropriately)**

    - dig +short TXT example.com — find SPF record (look for v=spf1 ... -all).
    - dig +short TXT selector._domainkey.example.com — verify DKIM public key for the DKIM selector captured in headers.
    - dig +short TXT _dmarc.example.com — check DMARC policy and reporting addresses.
    - Interpretation: missing/weak SPF/DKIM/DMARC increases likelihood of spoofing.
4. **WHOIS domain age**
    - whois suspectdomain.com | egrep -i 'Creation Date|Registered'
    - Interpretation: newly created domains (days-old) are higher risk; compare with first seen timestamps from VT/URLScan.

Quick output summary to include in ticket: Authentication summary, Received chain IPs with geo/ASN, SPF/DKIM/DMARC status, WHOIS age.

Pitfalls:

- Some mail vendors rewrite or add Authentication-Results — note which server produced the header (header origin), and inspect earliest untrusted Received line.

Phase: Classification & Escalation

Purpose: Decide immediate severity so appropriate resources and SLAs are used.

Classification guide (expanded):

- **Critical**
    - Indicators: active credential submission form on landing page, user confirmed they entered credentials, malware that executed with persistence or exfiltration, email targeted at executives (CEO/CFO) asking for wire transfers.
    - Response: escalate to L2/L3 immediately and convene IR Lead. Consider account lockout, password resets, and potential legal/communications.
- **High**
    - Indicators: user clicked link or downloaded an unknown attachment but no credentials submitted or malware execution confirmed.
    - Response: L2 analysis within 1–2 hours (sandboxing, IOCs extraction, hunt for lateral activity).
- **Medium/Low**
    - Indicators: mass spam, generic phishing, known spam domain, no clicks.
    - Response: L1 may close and provide education. Optionally quarantine similar messages.

Document in ticket:

- **Initial findings** (IST timestamps)

- Path to .eml, thumbnails/screenshots
- Brief auth summary (spf/dkim/dmarc)
- WHOIS and VT/URLScan results (IDs)
- Classification rationale and escalation decisions

Example ticket template (copyable)

Incident ID: INC-20251019-0001
Reported by: user@example.com
Reported (IST): 2025-10-19 08:45 IST
Analyst: Kumar
Initial classification: High (User clicked link)
Evidence:
- /evidence/INC-20251019-0001/INC-20251019-0001_original.eml SHA256: ...
- /evidence/.../invoice.docx SHA256: ...
Auth Summary: SPF=fail (envelope-from: xyz.com), DKIM=none, DMARC=none
WHOIS: suspectdomain.com created 2025-10-17 (2 days old)
VT URLScan: VT ID: 12345 (0/70 detections), URLScan ID: 67890
Next steps: Sandbox attachment, block domain at proxy, hunt for clicks.

---

6. Investigation & Analysis — Static & Dynamic (deep detail)

Goal: Fully understand the malware or credential capture mechanism, extract robust IOCs, and produce defensible findings.

Static Analysis — step-by-step

What to parse and why:

- **Headers:** Authentication-Results, Received, Return-Path, Message-ID, From, Reply-To.
  - Message-ID anomalies (e.g., forged or missing domain) can indicate spoofing.
  - Return-Path vs From mismatch indicates spoofing of display name.
- **DKIM selector verification**
  - Extract selector from DKIM-Signature: s=<selector>; d=<domain> then dig selector._domainkey.domain TXT.
  - Confirm the published public key matches signature algorithm; dkimverify tools can verify.
- **WHOIS and certificate checks for landing domains**
  - WHOIS: registrar, creation date, registrant country, contact email.
  - TLS cert: openssl s_client -connect suspectdomain.com:443 -servername suspectdomain.com then openssl x509 -noout -text to check issuer and validity dates.
  - Self-signed certs or certs issued to different CNs are suspicious for credential handlers.
- **VirusTotal / URLScan submission**

    - Submit attachments (hashes) and URLs to VirusTotal; record detection ratio, first seen, and associated domains/IPs.
    - For URLs use URLScan to capture screenshots and redirect chain — URLScan also records host headers and TLS cert info.
    - Record IDs: VT:<id>, URLScan:<id> in ticket.
- **File-type and macros**
    - Open docx as zip (unzip) and inspect word/vbaProject.bin for macros.
    - Use oletools (olevba) to extract macro code and search for Shell, PowerShell, CreateObject("WScript.Shell"), -EncodedCommand.
    - Hash values: compute MD5/SHA1/SHA256 for matching in EDR.

Dynamic Analysis (Sandbox) — step-by-step

Key guideline: Never expose sandbox to production network. Use isolated lab, NAT controlled, and capture full network traffic.

1. **Sandbox selection & configuration**
    - Tools: Cuckoo, Any.Run, VMRay. Configure the VM to simulate typical enterprise host (Office + browser + plugins) but with no access to corporate resources.
    - Enable full logging: process tree, API calls, registry, file system changes, scheduled tasks.
2. **For attachments**
    - If Office file: enable macros in a controlled VM only if static analysis shows macros — otherwise avoid executing until confident.
    - Use different macro flags: enable macro execution, record process tree, capture spawned cmd.exe/powershell.exe commands.
    - Capture artifacts: files dropped (paths & hashes), registry Run keys, scheduled tasks, LNK creation.
3. **For URLs**
    - First perform non-interactive fetch: curl -IL <url> to capture redirect chain and Location: headers.
    - Save landing HTML for static review: curl -L <url> -o landing.html (in sandbox).
    - If a form exists, submit **dummy credentials** — use synthetic emails and passwords to observe server response form fields and any exfiltration behaviors (e.g., redirect to thanks.html or to collector.php).
    - Do NOT use real credentials.
4. **Network capture**
    - Capture PCAP via tcpdump -w run.pcap or built-in sandbox PCAP export.
    - Analyze DNS requests: what domains were queried, frequency, TTLs, and any patterns of DGA-like behavior.
    - HTTP(S): even though encrypted, observe SNI, endpoint IPs, and certificate info.
5. **Record & document**
    - Produce a sandbox report (PDF/HTML) with:

        - Process tree
        - Files written & their hashes
        - Registry changes (keys/values)
        - Network calls (domains, IPs)
        - Screenshots of observed UI
    - Save PCAP and raw logs as evidence.

Pitfalls & mitigations:

- Malware that detects sandbox artifacts may behave benignly — consider using multiple sandbox configurations and user interaction simulation (e.g., clicking inside Word).
- Some sandboxes need to simulate a real user (Office ActiveX, realistic file paths) to trigger payload.

IOC Extraction

What to extract and how to validate:

- **Domains & URLs:** full URL strings, redirect chains (capture with URLScan).
- **IPs:** all external IPs contacted; resolve to ASN and reputation lookups (AbuseIPDB).
- **File hashes:** MD5/SHA1/SHA256 for any dropped files or attachments.
- **C2 domains:** from network calls; correlate with VT/Passive DNS.
- **Form field names:** e.g., `username`, `password`, or custom names — helpful for detection signatures.

Validate IOCs:

- Cross-check with VT/AbuseIPDB/Passive DNS to reduce false positives before pushing to prevention devices.
- Tag each IOC with confidence level and source.

Examples of suspicious behaviors to look for

- Office macro invoking: `powershell -nop -w hidden -enc <base64>` — base64 encoded payloads are high risk.
- Binary creating scheduled task `schtasks /create /tn "updateSys"` — persistence indicator.
- Credential form posting to third-party collectors: POST endpoint to IP address or unusual domain, especially without HTTPS or with self-signed cert.

Evidence packaging

Package contents:

- raw `.eml`

- extracted attachments (original)
- sandbox report (PDF/HTML)
- PCAP from sandbox
- process logs & registry exports
- IOC list (CSV) with fields: IOC, type, first seen, source, confidence
- Hash list

Label package with incident ID and ensure read-only storage.

---

7. Hunting & Correlation (SIEM Queries and Playbooks) — detailed usage

Goal: Determine scope and impacted users, identify follow-up evidence (clicks, sign-ins, payloads) and contain.

Splunk examples — explanations & tuning

1. **Find recipients of suspicious domain**
2. index=email sourcetype=o365 "suspectdomain.com"
3. | stats count by RecipientEmailAddress, SenderFromAddress, Subject
    - Purpose: find all mail recipients who received messages containing suspectdomain.com.
    - Tuning: add | dedup RecipientEmailAddress for unique list. Add a time range earliest=-7d as needed.
4. **Proxy clicks**
5. index=proxy sourcetype=proxy "suspectdomain.com"
6. | stats count by src_ip, user, dest_url, _time
    - Purpose: find which users/hosts actually clicked through to the suspect domain.
    - Notes: proxy logs may obfuscate SNI or use TLS interception; ensure you have correct fields.
7. **PowerShell encoded commands (EDR)**
8. index=edr sourcetype=processes ProcessName="powershell.exe" CommandLine="*-enc*"
9. | stats count by Host, User, CommandLine
    - Purpose: detect obfuscated PowerShell usage indicative of post-exploit actions.
    - Tuning: filter out known admin maintenance tasks by white-listing hosts or users.

Sentinel KQL examples — explanation

- **Sign-in anomalies**
- SigninLogs
- | where TimeGenerated > ago(7d)
- | where IPAddress !in ("<known-corp-ips>")
- | project TimeGenerated, UserPrincipalName, IPAddress, Location, DeviceDetail

    - Purpose: find unexpected sign-ins that coincide with users who clicked phishing links.
    - Note: correlate `UserPrincipalName` with proxy logs or email recipient list.

Scheduled hunts

- **Daily:** new domains observed in incoming mail with domain age < 7 days.
    - Implementation: query mail logs for `Received` URLs/domains, check domain age via WHOIS or passive DNS API.
- **Weekly:** parse DMARC aggregate (RUA) reports for sources with high spoofing volume.

IOC enrichment — tools & order

- Check IOCs via:
    - **AbuseIPDB** (IP reputation)
    - **VirusTotal** (file & URL detection)
    - **Passive DNS** (history of domain/IP relationships)
    - Internal allow/block lists (to avoid blocking critical services)
- Tag IOCs with `malicious`, `suspicious`, or `benign` plus source info and confidence.

Playbook (operational steps)

1. **Run recipient search for domain/subject** (email index): build containment list.
2. **Identify users who clicked** (proxy logs): determine which accounts/hosts are at immediate risk.
3. **Correlate with sign-in logs** for suspicious authentication attempts post-click.
4. **Query EDR** for any matching file hash or command lines on those hosts.
5. **Build containment list** (hosts & accounts) for immediate actions (see Containment Playbooks).

Practical advice when running hunts

- Always snapshot findings into ticket (include queries and time windows).
- Save query results with the incident ID for auditability.
- Use automation where possible (playbooks in SIEM) to speed up repetitive tasks.

---

8. Containment Playbooks (Actionable Steps by Scenario) — detailed steps & commands

Each action must be recorded with approver and timestamp. Below are step-by-step containment actions per scenario.

Playbook A — Clicked Link Only (No Credentials)

Assumptions: User clicked link but did not submit credentials or download files.

Actions:

1. **Block domain/IP at proxy & firewall**
   - Procedure: create deny rule for suspectdomain.com at proxy; add IP(s) to firewall blocklist (with expiration if necessary).
   - Document rule ID and approver.
2. **Quarantine similar messages at gateway**
   - Create mail gateway rule to quarantine messages with matching subject patterns or From domain.
   - Example rule: quarantine if FromDomain in (suspectdomain.com) OR message contains suspicious_subject_pattern.
3. **Notify user & provide instructions**
   - Template: "Do NOT enter credentials. Clear browser cache, close browser, run corporate AV scan, do not re-open link. If you submitted credentials, contact security immediately."
   - Provide tailored remediation steps (browser clearing steps, company-specific instructions).
4. **Monitor user account for unusual sign-ins for 72 hours**
   - Set up watchlist or alert for account sign-ins from new locations or risky IPs.
   - Escalate if there are anomalous sign-ins.

Approver: L2 Analyst (record name and timestamp in ticket).

Playbook B — Credentials Submitted

Assumptions: Evidence user submitted credentials (self-reported or observed via sandbox/landing page with POST).

Actions:

1. **Immediately reset user password and revoke refresh tokens**
   - Example Azure AD commands (PowerShell/AzureAD/Graph):
     - Revoke tokens: Revoke-AzureADUserAllRefreshToken -ObjectId <userObjectId>
     - Reset password via admin portal or Set-AzureADUserPassword.
   - Record who performed reset & timestamp.
2. **Force MFA re-enrollment & revoke app passwords**
   - Disable app passwords and re-issue new MFA enrollment requirement.
3. **Hunt sign-in logs and block offending IPs**
   - Query sign-in logs for suspicious attempts post-credential submission. Block IPs at firewall and in conditional access if needed.

4. **User communication**
    - Notify user and their manager; provide incident summary, what was changed (password reset), and next steps for verification.

Approver: IR Lead (document approval).

Playbook C — Malware Executed

Assumptions: malware executed (EDR telemetry, sandbox, host analysis).

Actions:

1. **Isolate host via EDR (network quarantine)**
    - Use EDR console to cut network access. Document time and who issued isolation.
2. **Acquire volatile memory & disk image**
    - Memory: winpmem or EDR built-in acquisition.
    - Disk: FTK Imager, dd, or EDR imaging. Capture hashes of images.
3. **Kill malicious processes & collect artifacts**
    - Use EDR to kill processes and gather artifacts (DLLs, dropped files, registry exports).
    - Document process tree and command lines.
4. **Decide rebuild vs cleanup**
    - Based on persistence indicators (rootkit, signed drivers, hidden services) and risk tolerance: typically rebuild image when persistence or rootkit suspected.

Approver: IR Lead.

Playbook D — Mass Campaign / Brand Spoof

Assumptions: large distribution, brand impersonation, high volume.

Actions:

1. **Block sending domain & associated IPs**
    - Contact providers if necessary (report abuse) and block at gateway.
2. **Create gateway rule to quarantine messages**
    - Rule examples: if From header contains @brand.com but DKIM/SPF fails.
3. **Publish internal advisory & targeted user education**
    - Send communication to affected user groups with sample screenshots and clear remediation steps.

Approver: Mail Admin / IR Lead.

Logging & approvals

- Every action: who, when (IST), reason, revert plan (if applicable), ticket reference.
- Example log entry:
- 2025-10-19 09:15 IST — Blocked suspectdomain.com at proxy (RuleID: PROXY-234) — Analyst: Kumar — Approver: Raj (L2)

---

9. Eradication & Recovery — detailed steps & validation

Goal: Remove all malicious artifacts, ensure accounts and hosts are secure, and validate no continued adversary presence.

Host remediation (detailed)

1. **Remove persistence**
    - Check & remove: Scheduled Tasks, HKCU\Software\Microsoft\Windows\CurrentVersion\Run*, Services, Startup folder, WMI persistence.
    - Use forensic tools (Autoruns) to enumerate.
2. **AV/EDR update & full scan**
    - Ensure signatures/engines are up-to-date and run full enterprise scan on remediated host.
3. **Re-image host when necessary**
    - Rebuild if rootkits discovered, unknown bootkits, or irrecoverable persistence.
    - Document preservation of forensics (image taken before rebuild).
4. **Patch OS and applications**
    - Apply latest security patches (OS + relevant apps) before returning to production.

Account remediation

1. **Reset passwords & revoke tokens**
    - Force immediate password change and revoke refresh tokens and OAuth grants.
2. **Disable legacy auth**
    - Block legacy auth protocols to prevent credential replay.
3. **Force MFA re-enrollment and verify devices**
    - Validate device registrations and remove unknown devices.

Validation / Monitoring

1. **Monitor sign-in logs for 30 days**
    - Watch for unusual IPs, new devices, or risk signals.
2. **Re-run SIEM hunts for IOCs**

    - Re-run the hunting queries from section 7, expand time windows to include pre- and post-incident activity.
3. **Re-enable accounts only after validation**
    - Re-activate only when no suspicious signals persist and user confirms expected behavior.

Communication

- **Notify user & manager when access restored**
    - Provide a summary of remediation steps, what was removed, and verification steps user should take (e.g., confirm MFA devices).
- **Executive communication**
    - If execs affected, prepare tailored summary with impact, containment timeline, and remediation status.

Extra practical notes, templates & utilities

Short checklist you can print and carry:

1. Intake: get .eml, screenshots, assign INC-....
2. Preserve: extract attachments, compute SHA256, store read-only.
3. Quick auth: parse Authentication-Results, Received chain, run dig and whois.
4. Classify: Critical / High / Medium / Low — escalate accordingly.
5. Static analysis: DKIM, WHOIS, cert check, VT/URLScan.
6. Dynamic: Sandbox safe execution, PCAP, process logs.
7. IOC extraction & enrichment: VT, AbuseIPDB, Passive DNS.
8. Hunt: SIEM queries across mail/proxy/signins/EDR.
9. Contain: block & quarantine, reset creds if needed.
10. Eradicate & recover: image/memory, rebuild if needed, monitor 30 days.

Quick commands summary (Linux environment)

- Extract headers: `sed -n '1,200p' INC.eml | egrep -i '(Authentication-Results|Received|From|Return-Path|Message-ID|DKIM-Signature)'`
- Get SPF: `dig +short TXT example.com`
- DKIM public key: `dig +short TXT selector._domainkey.example.com`
- WHOIS: `whois suspectdomain.com | egrep -i 'Creation Date|Registered'`
- Hash: `sha256sum filename > filename.sha256`
- Redirect chain: `curl -IL http://suspecturl/`

Recommended fields to always include in the ticket

- Incident ID, reporter, timestamps (IST), evidence paths & hashes, authentication summary, WHOIS & cert notes, sandbox IDs, VT/URLScan IDs, classification,

containment actions (with approver names/times), eradication steps, monitoring plan.

10. Forensics & Evidence Collection

Artifacts to collect:

- Raw .eml (complete headers).
- Attachments in original format.
- Screenshots (email, hover preview).
- Sandbox report (with screenshots and PCAP).
- Endpoint artifacts: memory dump, disk image, registry exports, event logs.
- Network artifacts: proxy logs, DNS logs, PCAPs.

Hashing & chain-of-custody:

- Compute SHA256 for every file collected:
  - ❖ sha256sum file > file.sha256
- Log collector name, timestamp (IST), storage path, and hash in ticket.
- Store artifacts in read-only evidence repository with ACLs and audit logs.

Legal preservation:

- If incident may trigger regulatory notification, preserve additional logs and document steps for legal review.

11. Preventive Controls & Detection Rules

Mail gateway rules:

- Quarantine if Reply-To != From AND message contains link/attachment.
- Quarantine if Authentication-Results show spf=fail AND dkim=none for messages claiming to be from high-risk brands.
- Quarantine messages where sending domain age < 7 days and body contains credential language.

Proxy/WAF rules:

- Block external POSTs that contain 'password' to non-approved domains.
- Intercept and warn users when visiting known credential collection domains.

EDR rules:

- Alert on Office macros spawning PowerShell or cmd.exe.
- Alert on Scheduled Task / New Service creation matching known persistence patterns.

SIEM detection:

- Scheduled rules to check for new domains sending inbound mail.
- Aggregate DMARC RUA reports to identify spoofing trends.

User training:

- Phishing simulations focused on spear/phishing and BEC scenarios.
- Targeted training for finance, HR, and execs.

Technical hardening:

- Enforce DMARC p=quarantine/reject once flows are validated.
- Enforce MFA on all accounts and conditional access policies for risky sign-ins.

Recommended automations:

- .eml triage script (Python): parses .eml, extracts From, Return-Path, Received chain, Authentication-Results, DKIM selector, runs dig for SPF/DKIM/DMARC and produces triage summary.
- IOC push automation: on confirmation, push domains/IPs/hashes to gateway/firewall via API.
- DMARC RUA ingestion: parse aggregate XML and ingest into SIEM for easy analysis.

Example .eml triage pseudo-code (Python):

```
import email, subprocess

msg = email.message_from_file(open('INC.eml'))

from_hdr = msg['From']

auth = msg['Authentication-Results']

# extract DKIM selector and run dig...

# produce JSON summary
```

13. Post-Incident Activities & Metrics

Metrics to track:

- Time-to-triage (goal <= 15 minutes).
- Time-to-containment for critical incidents (goal <= 2 hours).
- Number of incidents with credential exposure.
- Number of incidents resulting in endpoint compromise.

Continuous improvement:

- Update detection rules and gateway policies from IOCs.
- Review incidents in weekly IR meeting; update playbook.
- Run targeted phishing simulations, measure user click rates and training effectiveness.

Lessons learned:

- For each incident, document root cause, what worked, gaps, and actions to prevent recurrence?

14. Appendices — Commands, Examples & Checklists

Appendix A — Quick Commands

- dig +short TXT <domain>
- dig +short TXT <selector>._domainkey.<domain>
- dig +short TXT _dmarc.<domain>
- whois <domain>
- sha256sum <file>
- grep -i "Authentication-Results" INC.eml
- curl -IL <url>

Appendix B — SIEM Query Library (examples)

(Refer to Section 7 for Splunk and Sentinel examples)

Appendix C — Incident Report Template (fields)

- Incident ID, Summary, Detection source, Timeline (IST), Affected users, Analysis, Containment, Eradication, IOCs, Recommendations.

Appendix D — One-Page Analyst Checklist

- Preserve .eml & attachments
- Capture screenshots
- Run auth checks (SPF/DKIM/DMARC)
- WHOIS domain age
- Submit to VT/URLScan
- Sandbox if risky
- Hunt in SIEM for recipients & clicks
- Contain per scenario

- Collect forensic artifacts
- Final report & lessons learned

Appendix E — Example annotated header (for training)

- Return-Path: <bounce@amazn-billing.com> <-- envelope from
- From: "Amazon Billing" <support@amazon.com> <-- visible From (may be forged)
- DKIM-Signature: v=1; a=rsa-sha256; d=amazn-billing.com; s=mail2025; ...
- Authentication-Results: mx.company.com; spf=fail smtp.mailfrom=amazn-billing.com; dkim=none; dmarc=fail

15. Phishing Incident Checkpoints — Quick Tick-List (IST timestamps required)

Use these checkpoints as rapid yes/no/NA items during every phishing incident. Mark each with ✅ / ❌ / **N/A** and include **IST timestamp** and your initials for every checked item.

I. Incident Intake (0–5 min)

- [ ] Incident ID created (INC-YYYYMMDD-####)
- [ ] Reporter recorded (name/email/method)
- [ ] Raw .eml exported and saved to evidence repo (do not forward)
- [ ] Screenshots captured: list view / opened email / hover-link preview

II. Preservation (0–10 min)

- [ ] All attachments extracted and saved in original format
- [ ] SHA256 computed for .eml + each attachment and recorded
- [ ] Evidence copied to read-only repository (ACLs verified)
- [ ] Artifact paths + hashes logged in ticket

III. Quick Authentication & Header Triage (0–15 min)

- [ ] Authentication-Results header extracted and recorded (SPF/DKIM/DMARC)
- [ ] Received chain extracted; external relay IPs identified
- [ ] SPF record checked (dig TXT)
- [ ] DKIM selector looked up and public key checked
- [ ] DMARC (_dmarc.) record checked
- [ ] WHOIS / domain age checked for suspect domains

IV. Initial Classification & Escalation

- ☐ User clicked link?
- ☐ User reported entering credentials?
- ☐ Attachment executed on endpoint? (EDR/sandbox evidence)
- ☐ Classification set (Critical / High / Medium / Low)
- ☐ Escalation performed (L1→L2 / L2→L3) if required

V. Static Analysis Checkpoints

- ☐ DKIM signature(s) present / verified
- ☐ Return-Path vs From mismatch noted
- ☐ Attachments scanned with VT / YARA / antivirus
- ☐ Macros detected? (olevba / unzip)
- ☐ TLS cert checked for landing domain(s)

VI. Dynamic Analysis / Sandbox

- ☐ Sandbox job created
- ☐ Sandbox executed in isolated lab (no prod access) **Time (IST): _____**
- ☐ PCAP captured and saved
- ☐ Process tree / dropped files / registry changes documented
- ☐ Form POST behavior observed (dummy creds used)

VII. IOC Extraction & Enrichment

- [ ] Domains / URLs listed in ticket
- [ ] IP addresses resolved and enriched (ASN/reputation)
- [ ] File hashes recorded (MD5/SHA1/SHA256)
- [ ] IOCs checked against VirusTotal / AbuseIPDB / PassiveDNS
- [ ] IOC confidence assigned (High/Med/Low)

VIII. Containment Actions (record approver & time)

- [ ] Blocked domain(s) at proxy/WAF
- [ ] Blocked IP(s) at firewall
- [ ] Quarantined similar messages at gateway (rule created)
- [ ] Endpoint isolated via EDR (network quarantine)
- [ ] Password reset & tokens revoked (if creds submitted)
- [ ] MFA forced re-enroll (if applicable)

IX. Hunting & Correlation

- [ ] Recipient search executed in mail index (save results)
- [ ] Proxy logs checked for clicks (save results)
- [ ] Sign-in logs correlated for compromised accounts
- [ ] EDR searched for matching hashes / cmdlines

X. Forensics & Evidence Packaging

- [ ] Memory dump acquired (if host compromised)
- [ ] Disk image captured and hashed
- [ ] All evidence bundled to package (list included)
- [ ] Chain-of-custody logged (collector name + IST)

XI. Remediation & Recovery

- [ ] Persistence removed (tasks/services/Run keys)
- [ ] AV/EDR full scan completed on affected hosts
- [ ] Hosts re-imaged (if applicable)
- [ ] Accounts re-enabled only after validation

XII. Communication & Reporting

- [ ] User notified with remediation steps (template used)
- [ ] Manager / Exec notified (if needed)
- [ ] Legal/PR engaged (if required)
- [ ] Final incident report drafted and uploaded to ticket

XIII. Post-Incident & Metrics

- [ ] IOCs pushed to prevention stacks (proxy/firewall/gateway/EDR)
- [ ] SIEM detection rules updated / new saved searches created
- [ ] Lessons learned review scheduled (date/time IST)

- [ ] Phishing simulation / awareness action planned