PHISHING:

Spam: Unwanted bulk messages mainly for **advertising** and annoyance.
Phishing: Fraudulent messages designed to **trick you into giving sensitive information.**

Attack vector = the path or method an attacker uses to enter a system, such as email, USB, websites, Wi-Fi, attachments, or social engineering tricks.

Many products help combat spam and phishing, but realistically these emails still can get through. you will need to gather information about the email to update your security products to prevent malicious emails from making their way back into a user's inbox.

The Email Address:
what makes up an email address?

1. **User Mailbox (or Username)**
2. **@**
3. **Domain**

The person responsible for the contribution to the way we communicate was Ray Tomlinson.

billy@johndoe.com.

1. The **user mailbox** is billy
2. **@** (thanks Ray)
3. The **domain** is johndoe.com

The domain is like your street and the mailbox is the user’s address, and email-sending protocols (SMTP, IMAP/POP3) transport the message between them

SMTP delivers emails between mail servers; IMAP/POP3 deliver emails from the server to the user.

Why does Billy’s receiver side also use an SMTP server?

Because **SMTP is used for receiving *from other mail servers*, not from users.**

- **SMTP = server-to-server delivery protocol**

- **IMAP/POP3 = server-to-user retrieval protocols**

The syntax for email messages is known as the Internet Message Format (IMF).

📧 Email Example (Simple)

What you SEE in your inbox:

From: newsletters@ant.anki-tech.com
(This looks normal — like a company newsletter.)

Hidden header:

Reply-To: reply@ant.anki-tech.com
(This is where **your reply will ACTUALLY go.)**

🔍 What this means in simple words:

- The email **looks** like it came from:
  ✔ newsletters@ant.anki-tech.com

- But if you click **Reply**, your message will be sent to:
  ❗ reply@ant.anki-tech.com
  (a different address)

🧠 Simple real-world analogy:

Someone sends you a letter saying:
"Hi, I am Ali. My address is **House 10**…
But if you want to reply, send your letter to **House 99**."

Why House 99?
Maybe it's their *real* address… or maybe it's a **scam address**.

🌐 Why attackers use this?

They make the **From** address look legit
→ but replies go to **their own email**.

Example phishing scenario:

- **From:** support@paypal.com (looks official)
- **Reply-To:** scammer@gmail.com (real attacker)

Envelope-To

- This header shows that this email was delivered to the mailbox of a subscriber whose email address is user@example.com.

Received" lines:

Every time an email passes through a mail server, that server adds a **"Received" line** in the email header.

- It says:
  "I got this email from this server at this time."
- So an email has **many Received lines**, one for each server it passed through.

How to read them

- **Read from bottom → top.**
    - Bottom line = **first server that handled the email** (closest to sender).
    - Top line = **last server** (closest to you).

Think of it as a **trail of breadcrumbs** showing the email's path.

- **X-Originating-IP** header → this is usually the **sender's real IP**.
- **If X-Originating-IP is missing:**
  → Check the **Received** lines from bottom up.
  → Look for the first external IP address that isn't your own server's IP.

Example:

```
X-Originating-IP: 10.140.188.3
```

Here, **10.140.188.3** = the computer that sent the email.

Return-Path

- The email address for return mail. This is the same as "Reply-To:".

Once you find the email sender's IP address, where can you retrieve more information about the IP?**http://www.arin.net**

HTML is what makes it possible to add these elements to an email.: photos,links

These headers describe **the actual content of the email**, not just attachments.

BEC (Business Email Compromise) means.

A BEC is when an adversary gains control of an internal employee's account and then uses the compromised email account to convince other internal employees to perform unauthorized or fraudulent actions.

1. Spam

Unwanted junk emails sent to many people.
👉 Annoying, sometimes harmless, sometimes contains ads.

2. MalSpam

Malicious spam.
👉 Spam that contains malware, links to malware, or harmful attachments.

3. Phishing

Fake emails pretending to be from a trusted company.
👉 Goal: steal sensitive info like passwords or credit card details.

4. Spear Phishing

Highly targeted phishing.
👉 Attacker researches a specific person or company and crafts a realistic message for them.

5. Whaling

Phishing targeting top-level executives (CEO, CFO).
👉 Goal: get money transfers, sensitive business info.

6. Smishing

Phishing using SMS/text messages.
👉 Example: “Your bank account is locked, click here.”

7. Vishing

Phishing using voice calls.
👉 Fake tech support, fake bank calls, etc.

The attacker usually wants one of two things:

1️⃣ Steal your login details (harvest credentials)

Example: fake Microsoft login page.

2️⃣ Infect your computer with malware

Example: malicious PDF, ZIP, Word document.

So phishing emails share common signs.

1. Fake sender (email spoofing)

The email pretends to come from Amazon, Microsoft, or a bank.
Example:
`support@amaz0n.com`

---

2. Urgent subject / scary message

They try to make you panic:

- “Invoice overdue”
- “Your account is suspended”
- “Password expires today”

---

3. The email looks like a real company

HTML formatting makes it look official (logo, buttons, colors).
But it might still be fake.

---

4. Poor grammar or messy formatting

Even though they try to look official, attackers often make mistakes.

---

5. Generic greeting

“Dear Customer”,
“Dear User”,
instead of your real name.

---

6. Suspicious links (hyperlinks)

Links that look normal but lead to malicious sites.
Sometimes they use **URL shorteners** like:
`bit.ly/xxxx`

---

7. Malicious attachments

Fake invoices, fake job offers, fake delivery receipts that contain malware.

What is DEFANGING? (super simple)

Defanging = making a dangerous link **safe to share** by breaking it so no one can click it by mistake.

Example:

Real malicious link:

```
http://badsite.com
```

Defanged version:

```
hxxp[://]badsite[.]com
```

Now it cannot be clicked accidentally.

SOC analysts **always defang** links before sharing them

CyberChef is a tool that:

- defangs/refsangs URLs
- encodes/decodes
- analyzes data
- extracts strings
- Base64 decoding
- and more

Original URL analogy

Think of a URL as a **Coca-Cola bottle** you're handing around:

- It's real, full, and someone could **drink it (click it)** and get sick (malware).

```
http://coke.com/drink
```

Defanged URL analogy

Defanging is like **covering the bottle and labeling it "Do Not Drink"**:

- It's still recognizable as Coke
- But nobody can accidentally drink it (click the link)

`hxxp[://]coke[.]com/drink`

You change:

- **`http://` → `hxxp[://]`**
- **Every `.` in domain → `[.]`**
- **Do *not* change the query string after `?`**

Envelope-To — Final Recipient,This shows who received it. here is real and verified by the mail system.

Reply-To is **where replies are sent** (can be faked), Envelope-To is **who actually got the email** (harder to fake).

X-Originating-IP — Real Sender

Phishing Emails in Action:

Real website: `paypal.com`

Typosquatting: `paypa1.com` (letter “l” replaced with “1”)

CC = visible recipients,
BCC = hidden recipients

CC (Carbon Copy)

Everyone in the email can **see who is CC’d.**

BCC (Blind Carbon Copy)

No one can **see** who is BCC’d — **the recipients are hidden.**

BCC matters when an attacker sends the SAME email to many victims

Example:

Attacker sends 1 email to **200 people**.

He writes this in the *visible* field:

```
To: updates@apple.com
```

But secretly adds:

```
BCC: ali@yahoo.com
BCC: sara@gmail.com
BCC: john@live.com
… (many more)
```

All 200 victims **only see the fake "To:"**
None of them see who else received it.

So *only the sender* knows all the recipients.

Phishing Analysis Tools:

pertinent information an analyst (you) is to collect from the email header:

1. Sender **email address**
2. Sender **IP** address
3. **Reverse lookup** of the sender IP address
4. Email **subject line**
5. Recipient email address (this information might be in the **CC/BCC field**)
6. **Reply-to email address (if any)**
7. **Date/time**

Below is a checklist of the **artifacts an analyst (you)** needs to collect from the **email body:**

1. **Any URL links** (if an **URL shortener service** was used, then we'll need to **obtain the real URL link**)
2. The **name of the attachment**
3. The **hash value of the attachment** (hash type MD5 or SHA256, preferably the latter

Email header analysis:

Info that **we need to collect can be obtained visually** from an **email client or web client** (such as Gmail, Yahoo!, etc.). But some information, **such as the sender's IP address and reply-to information, can only be obtained via the email header.**

bat is a tool from Google that can assist us with analyzing email headers called Messageheader from the Google Admin Toolbox.

MTA (Message Transfer Agent)

Think of **MTA as a *mail truck*** 🚚

It moves emails between mail servers.

Examples of MTAs:

- Postfix
- Exim
- Sendmail
- Microsoft Exchange Transport

It never interacts with the user directly.

🧑‍💻 **MUA (Mail User Agent)**

Think of MUA as your *mailbox and mail app* 📱📥

It's the application you use to **read, send, and manage emails**.

Examples of MUAs:

- Gmail (web)
- Outlook (desktop or mobile)
- Apple Mail

MTA = server-to-server

MUA = server-to-user

urlscan.io is a tool that **safely checks** what happens **when you open a suspicious URL** — but *without you **actually opening it on your own computer***.

Think of it like a robot browser that visits the website for you.

When you **extract URLs** from an **email,** don’t only c**heck the full link — also check the *root domain* (main domain)** because attackers often hide malicious activity behind subdomains.

Example:
URL: `https://login.secure-update.apple-id.verify.com/reset`
Root domain: `verify.com`

Even if the subdomain looks legit (apple-id, login, secure), the **real root domain** may be malicious.

✔ **Summary:** Always analyze the **main domain** behind the URL, not just the full link.

Malware Sandbox:

There are online tools and services where **malicious files can be uploaded and analyzed** to better understand **what the malware was programmed to do**. These **services are known as malware sandboxes.**

PhishTool:

A tool that will help with **automated** phishing analysis is PhishTool.

- It **automates many steps** of **analyzing phishing emails.**
- Combines:
    - Threat intelligence
    - OSINT

- Email metadata (headers, IPs, SMTP info)
- Auto-analysis workflows

✅ Makes analyzing emails **faster, safer, and more organized**

Header : It is all the line in source in 1 paragraph where 1 line gap header ends.

Phishing Prevention:

SPF (Sender Policy Framework)

SPF (Sender Policy Framework) authenticates the **sending server.**

It answers the question: *“Is this email coming from a* ***server allowed by the domain owner****?”*

SPF does not check the **content or integrity of the message; i**t’s only about the sender.

How SPF Works (Workflow)

Here’s the flow:

1. Email is sent from **user@domain.com.**
2. **Receiving server** gets the email.
3. Receiving server looks up the **SPF record of** domain.com **in DNS.**
4. It checks the **sending IP against the SPF record.**
5. Based on the result, it takes an action.

Example:

v=spf1 ip4:127.0.0.1 include:_spf.google.com -all

v=spf1 → This is the **version** and start of SPF record.

ip4:127.0.0.1 → This **IP is authorized** to **send email** for this domain.

include:_spf.google.com → Any IPs allowed by **_spf.google.com** are also **allowed.**

-all → All other servers not listed should be **rejected.**

Notes:

~all = soft fail (flag)

-all = hard fail (reject)

?all = neutral   (accept)

Key difference:

- SoftFail = “Not on the list, but let’s be lenient.”/ (Mark as suspicious but allow)
- Fail = “Definitely not allowed, reject it.”

An SPF record is a DNS TXT record containing a list of the IP addresses that are allowed to send email on behalf of your domain

SPF vs DKIM

Feature	SPF (Sender Policy Framework)	DKIM (DomainKeys Identified Mail)
Purpose	Checks **which servers are allowed to send emails for a domain**	Checks **if the email itself was really sent by the domain and wasn’t altered**

DKIM:(DomainKeys Identified Mail) :

DKIM stands for **DomainKeys Identified Mail.**

It's a way to verify that an **email really comes from the domain it says it does.**

Think of it like putting a **digital signature on your email.**

1. Sending email:
    - The sending mail server uses a private key to sign the email.
    - This signature is unique and proves the email is from that domain.
2. Receiving email:
    - The receiving mail server looks up the public key from the domain's DNS.
    - It uses the public key to check if the signature is valid.
3. Verification:
    - If the signature matches, the email is authentic → it goes to the inbox.
    - If it doesn't match, the email may be flagged as spam or rejected.

✅ Key point: DKIM is more reliable than SPF because it can survive email forwarding.

DKIM Record Components

A DKIM record lives in DNS and looks like this:

```
v=DKIM1; k=rsa; p=<public_key>
```

- `v=DKIM1` → Version of DKIM (usually DKIM1).
- `k=rsa` → Type of key (RSA is standard).
- `p=<public_key>` → The **public key** that the receiving server uses to verify the email.

Think of it as: **Private key signs the email** → **Public key verifies it**.

A DKIM record exists in the DNS, but it is more complex than SPF. SPF fails when emails are forwarded because the email comes from a new server not listed in SPF. DKIM still works after forwarding because the signature is inside the email, so forwarding doesn't change it.
an email header that was marked as spam with a DKIM result of `permerror`, indicating a permanent failure in DKIM verificationThis could be the result of an invalid signature, a missing or incorrect DNS record, a forwarding server making a modification, or a misconfiguration in DKIM setup.

Domain-Based Message Authentication, Reporting, and Conformance (DMARC)

DMARC : open source standard
DMARC checks whether the domain in the email actually matches the domain verified by **SPF or DKIM.**

If the domains don't match (alignment fails), DMARC tells the **receiving server** what to do with the **email (accept, quarantine, or reject).**

Example DMARC record:

```
v=DMARC1; p=quarantine; rua=mailto:postmaster@website.com
```

- v=DMARC1 → DMARC version
- p=quarantine → What to do if DMARC fails (send to spam)

- rua=mailto:postmaster@website.com → Where to send DMARC reports (optional)

Other policies:

- p=none → Just monitor
- p=quarantine → Send suspicious emails to spam
- p=reject → Block emails that fail DMARC

DKIM = Server signs the email to prove the domain is real.

S/MIME = Person signs/encrypts the email to prove identity and protect content.
Users verify each other AND encrypt emails

S/MIME Working (SUPER EASY VERSION)

1. Both users need certificates

- Bob has a certificate → contains his public key
- Mary has a certificate → contains her public key

2. Bob signs the email with his PRIVATE key

This proves:

- Bob wrote it
- Email wasn’t changed
  Mary uses Bob’s public key to verify the signature.

3. For encryption, Bob uses MARY'S public key

So only Mary can read it.
Mary decrypts with her private key.

4. Mary replies the same way

Now both have each other's keys → secure two-way communication

S/MIME is a protocol used to **digitally sign** and **encrypt** emails.

Digital Signatures (using private key)

Encryption (using receiver's public key)

Technical Defenses

These are tools and systems that work **automatically** to detect and stop phishing or malicious emails **before they reach the user**:

Email Filtering:Secure Email Gateways (SEGs):
Link Rewriting:
Converts URLs in emails to safe versions
Sandboxing:

User-Facing Tools & Training

Even with technical defenses, some phishing emails get through. Users need **visual cues and training**:

Trust & Warning Indicators

Phishing Simulation Exercises:
User Awareness Training:

Technical defenses try to **block threats automatically**.

User-facing measures prepare humans to **recognize and respond** when a threat gets through.

X originating ip can be fake :

List of Attack Indicators: Signs or evidence suggesting malicious activity, such as unusual processes, unexpected network connections, or suspicious file changes.