The rule for Unusual Sender Addresses focuses on identifying email addresses that appear random, suspicious, or inconsistent with typical email address formats. Heres a detailed explanation of how it works and how it detects random or suspicious email addresses:

How It Works

1. Understanding Email Address Formats

A typical email address follows a standard format:

**localpart@domain**

Localpart: The part before the `@` symbol (e.g., `support`, `user123`).

Domain: The part after the `@` symbol (e.g., `example.com`, `gmail.com`).

2. Detecting Unusual Patterns

Unusual or random email addresses often:

Contain random strings of characters or numbers (e.g., `a1b2c3@xyz.com`).

Use unusual or nonstandard domains (e.g., `xyz.abc`, `random123.com`).

Lack meaningful localparts (e.g., `asdfgh@example.com`).

To detect these patterns, the rule uses pattern matching and heuristics:

1. Regex Pattern Matching:

Use a regular expression (regex) to validate the email address format.

Example regex: `^[azAZ09.\_%+]+@[azAZ09.]+\.[azAZ]{2,}$`.

This ensures the email address follows a basic structure but does not guarantee legitimacy.

2. LocalPart Analysis:

Check if the localpart contains random strings (e.g., `a1b2c3`).

Flag localparts that lack meaningful words or patterns.

3. Domain Analysis:

Check if the domain is unusual or nonstandard (e.g., `xyz.abc`, `random123.com`).

Compare the domain against a list of known legitimate domains.

How It Finds Random Email Addresses

1. Regex Validation

Use a regex pattern to ensure the email address follows a basic format.

Example:

Valid: `support@example.com`.

Invalid: `support@random123` (missing toplevel domain).

2. LocalPart Heuristics

Analyze the localpart for randomness:

Check for excessive use of numbers or special characters (e.g., `a1b2c3`).

Flag localparts that do not contain meaningful words or patterns.

3. Domain Heuristics

Analyze the domain for randomness:

Check if the domain is unusually short or long (e.g., `xyz.abc`).

Compare the domain against a list of known legitimate domains.

Flag domains that are not in the list or appear random.

4. Combination of Rules

Combine regex validation, localpart analysis, and domain analysis to detect unusual email addresses.

Limitations

1. False Positives:

Some legitimate email addresses may use randomlooking localparts (e.g., `user123@company.com`).

Solution: Combine this rule with other sender analysis rules to reduce false positives.

2. Evolving Tactics:

Phishers may use more sophisticated patterns to evade detection.

Solution: Continuously update the rule with new patterns and heuristics.

A common **regex pattern** for validating email addresses is:

^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$

**Breakdown of the Regex:**

1. ^ → Start of the string.
2. [a-zA-Z0-9.\_%+-]+ → Matches the **local part** (before @), allowing letters, digits, dots, underscores, percent signs, plus, and hyphens.
3. @ → Ensures there is exactly **one "@" symbol** in the email.
4. [a-zA-Z0-9.-]+ → Matches the **domain name**, allowing letters, digits, dots, and hyphens.
5. \. → Matches the **dot (.)** before the domain extension.
6. [a-zA-Z]{2,} → Ensures the **TLD (Top-Level Domain)** has at least **two** letters (e.g., .com, .org, .io).
7. $ → End of the string.

**Examples of Valid Emails:**

✅ user@example.com  
✅ john.doe123@company.co.uk  
✅ name+alias@gmail.com

**Examples of Invalid Emails:**

❌ user@.com (Missing domain name)  
❌ john@doe@company.com (Multiple @ symbols)  
❌ name@company (Missing TLD)

If you're looking for a **more strict or lenient version**, let me know! 🚀