D. Rule撰寫方法

1. 避免因為每個人定義變數方式會不同，因此有將變數定義方式寫在變數清單裡面
2. 必須按以下順序指定變數宣告、定義和用法：
   meta:
   events:
   match:
   outcome:
   condition:

Meta

Meta 部分由多行組成，每行定義一個鍵值對。鍵部分必須是不含引號的字串，值部分必須是引號的字串：

`<key> = "<value>"`

```
meta:
    author = "Google SecOps"

    description = "同一客戶 & 同一來源 & 10分鐘內 (count>1000 & 不重複目的IP>5 & 不重複目的port >300)"
    //規則描述
    mitre_attack_tactic =""
    mitre_attack_technique=""
    mitre_attack_technique_id=""
    //Mitre_attack戰術分布
    level = "第一級"
    //1.第一級 2.第二級 3.第三級 4.第四級
    severity = "LOW"
    //1.LOW 2.MEDIUM 3.HIGH 4.CRITICAL
    alert_match = "$namespace,$principal_ip over 10m "
    //符合的時間範圍區段
    alert_condition = "#e >1000 and $dc_target_ip >5 and $dc_target_port >300"
    //觸發閥值
    context = "資訊蒐集"
    //1.惡意內容 2.惡意程式 3.資訊蒐集 4.入侵嘗試 5.入侵攻擊 6.服務阻斷 7.資訊內容安全 8.詐欺攻擊 9.系統弱點 10.其他
```

Events

- 變數聲明
- 事件變數過濾器
- 事件變數連接

變數聲明

對於變數聲明，使用以下語法：

- `<EVENT_FIELD> = <VAR>`
- `<VAR> = <EVENT_FIELD>`

```
events:
//事件判斷條件區
    not $e.target.port = 443
    //使用reference list 做ip zone mapping
    //內對內
    /*(strings.concat($e.principal.namespace, ",", $e.principal.ip) in regex %ip_zone
    or re.regex($e.principal.ip,`^10\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|
    or re.regex($e.principal.ip,`^172\.(?:16|17|18|19|2[0-9]|3[0-1])\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9
    or re.regex($e.principal.ip,`^192\.168\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`))
    and
    (strings.concat($e.target.namespace, ",", $e.target.ip) in regex %ip_zone
```



```
    or re.regex($e.target.ip,`^10\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0
    or re.regex($e.target.ip,`^172\.(?:16|17|18|19|2[0-9]|3[0-1])\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[
    or re.regex($e.target.ip,`^192\.168\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`)) */

    //內對外
    /*(strings.concat($e.principal.namespace, ",", $e.principal.ip) in regex %ip_zone
    or re.regex($e.principal.ip,`^10\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|
    or re.regex($e.principal.ip,`^172\.(?:16|17|18|19|2[0-9]|3[0-1])\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9
    or re.regex($e.principal.ip,`^192\.168\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`))
    and
    (not strings.concat($e.target.namespace, ",", $e.target.ip) in regex %ip_zone
    and not re.regex($e.target.ip,`^10\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5
    and not re.regex($e.target.ip,`^172\.(?:16|17|18|19|2[0-9]|3[0-1])\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0
    and not re.regex($e.target.ip,`^192\.168\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`))

    //外對內
    (not strings.concat($e.principal.namespace, ",", $e.principal.ip) in regex %ip_zone
    and not re.regex($e.principal.ip,`^10\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[
    and not re.regex($e.principal.ip,`^172\.(?:16|17|18|19|2[0-9]|3[0-1])\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4
    and not re.regex($e.principal.ip,`^192\.168\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`
    and
    ( strings.concat($e.target.namespace, ",", $e.target.ip) in regex %ip_zone
    or re.regex($e.target.ip,`^10\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0
    or re.regex($e.target.ip,`^172\.(?:16|17|18|19|2[0-9]|3[0-1])\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[
    or re.regex($e.target.ip,`^192\.168\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`))
    $namespace = $e.principal.namespace
    //使用data table 做ip zone mapping
    //%test2.namespace = $namespace and net.ip_in_range_cidr($e.target.ip, %test2.principal_ip)
    $hostname = $e.principal.hostname
    $principal_ip = $e.principal.ip
    $target_ip = $e.target.ip
```

Match

Match在檢查符合條件之前列出群組事件的符合變數。每次符合時都會傳回這些欄位。

- 指定該部分中每個匹配變數代表什麼 `events`。
- 指定用於關聯 `over` 關鍵字之後的事件的時間持續時間。時間持續時間之外的事件將被忽略。
- 使用以下語法指定時間長度：`<number><m/h/d>`

  其中 `m/h/d` 分別表示分鐘、小時和天。
- 您可以指定的最短時間為 1 分鐘。
- 您可以指定的最長時間為 48 小時。

```
match:
    $namespace,$principal_ip over 10m

```

Outcome

每個結果變數可以有不同的資料類型，這取決於用於計算它的表達式。支援以下結果資料類型：

- integer
- floats
- string
- lists of integers
- lists of floats
- lists of strings

```
outcome:
    //outcome section, you can define up to 20 outcome variables
     ==============================這個先放==============================
    //事件數
    $event_count = count_distinct($e.metadata.id)
```



```
    //風險分數
    $risk_score = max(35)
    //觸發時間區間
    $event_time = strings.concat(timestamp.get_timestamp(min($gcp.metadata.event_timestamp.seconds)), "~", timestamp.get_timestamp(ma
    //事件描述
    $description = "單一來源IP對目的IP，觸發10分鐘內不重複目的IP大於5且不重複目的port大於300個，共達1000次以上"
    //==============================以下先不放==============================
    //來源ip數
    $dc_principal_ip = count_distinct($e.principal.ip)
    //目的ip數
    $dc_target_ip = count_distinct($e.target.ip)
    //來源port數
    $dc_principal_port = count_distinct($e.principal.port)
    //目的port數
    $dc_target_port = count_distinct($e.target.port)
    //來源user數
    $dc_principal_user = count_distinct($e.principal.user.userid )
    //目的user數
    $dc_target_user = count_distinct($e.target.user.userid )
    //總sent bytes數
    $total_sent_bytes = sum($e.network.sent_bytes)

    //總received bytes數
    $total_received_bytes = sum($e.network.received_bytes )
    //http response數
    $successful_access = sum(if($e.network.http.response_code = 200, 1,0))
    $forbidden_access = sum(if($e.network.http.response_code = 403, 1,0))
    $inbox_not_found = sum(if($e.network.http.response_code = 404, 1,0))
    $inbox_other = sum(if($e.network.http.response_code != 404
        and $e.network.http.response_code != 403
        and $e.network.http.response_code != 200, 1,0))
    //事件總類
    $product_event_type = array_distinct($e.metadata.product_event_type )

```

Condition

指定該部分中定義的事件和占位符的匹配條件 `events`

計數字元

該 `#` 字元是該部分中的特殊字元 `condition`。如果它在任何事件或占位符變數名稱之前使用，它表示滿足所有部分條件的不同事件或值的數量 `events`。

例如，`#c > 1` 表示變數 `c` 必須出現超過 1 次。

價值特徵

該 `$` 字元是該部分中的特殊字元 `condition`。如果它在任何結果變數名稱之前使用，它代表該結果的值。

如果它在任何事件或占位符變數名稱（例如 `$event`）之前使用，則它代表 `#event > 0`。

事件和占位符條件

在此列出事件和占位符變數的條件謂詞，並用關鍵字 `and` 或連接 `or`。關鍵字可以在任何條件之間使用，但僅當規則只有單一事件變數時才能使用 `and` 關鍵字。

```
condition:
    #e >1000 and $dc_target_ip >5 and $dc_target_port >300
    //事件數大於1000且目標連接IP大於5個且目標連接端口大於300個
```

