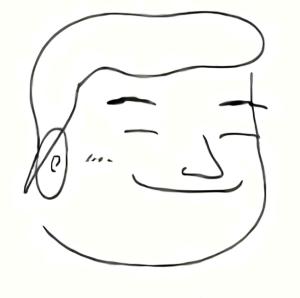
iam-policy-finder

CEL(Common Expression Language)で書いた条件にマッチしたIAM Policyを見つけるくん

2024.07.19 クラシコム・カヤック技術勉強会

@fujiwara

自己紹介



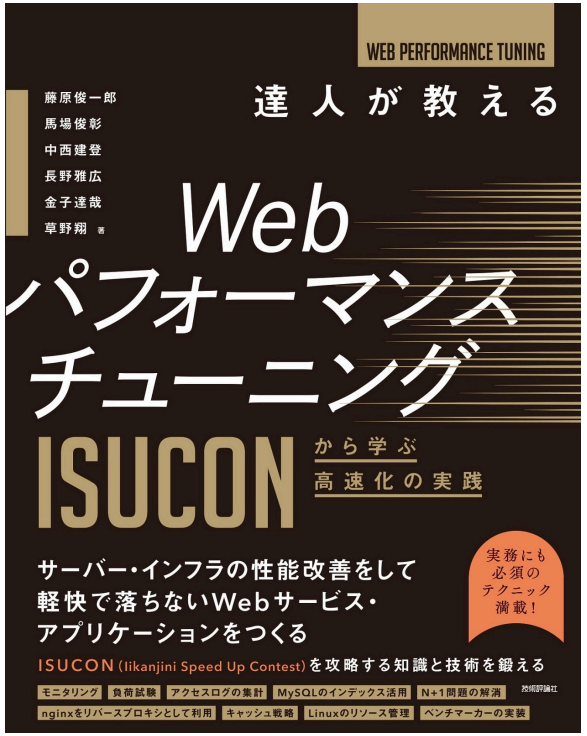
面白法人カヤック

@fujiwara (X/Twitter, GitHub, Bluesky)

面白法人カヤック SREチーム
ISUCON 🥇 優勝4回 / 運営(出題)4回

github.com/kayac/ecspresso
github.com/fujiwara/lambroll

最近の趣味はランニングとOSS



[Action Required] Add Lambda ListTags permissions for users calling GetFunction API

ある日AWSからメールが 🤔

> Lambda GetFunction API の認証に変更に際し、お客様のアクションが必要になる可能性があるため、ご連絡しております。

> これまで、ListTags API を明示的に使用する場合にのみ、ListTags への権限が必要でした。しかし、GetFunction API 権限を持つプリンシパルにおいても、GetFunction 呼び出しにより出力されたタグ情報にはアクセスできました。2024 年 7 月 27 日以降、Lambda は GetFunction API を呼び出すプリンシパルに ListTags API に対する明示的な許可権限が設定されたポリシーがある場合にのみタグデータを返すようになります。GetFunction API を呼び出すロールに対して、拒否ポリシーが設定されているか、ListTags API へのアクセスを明示的に許可するポリシーがない場合、Lambda は GetFunction API 呼び出しへの応答でタグデータを返さなくなります。

要約すると

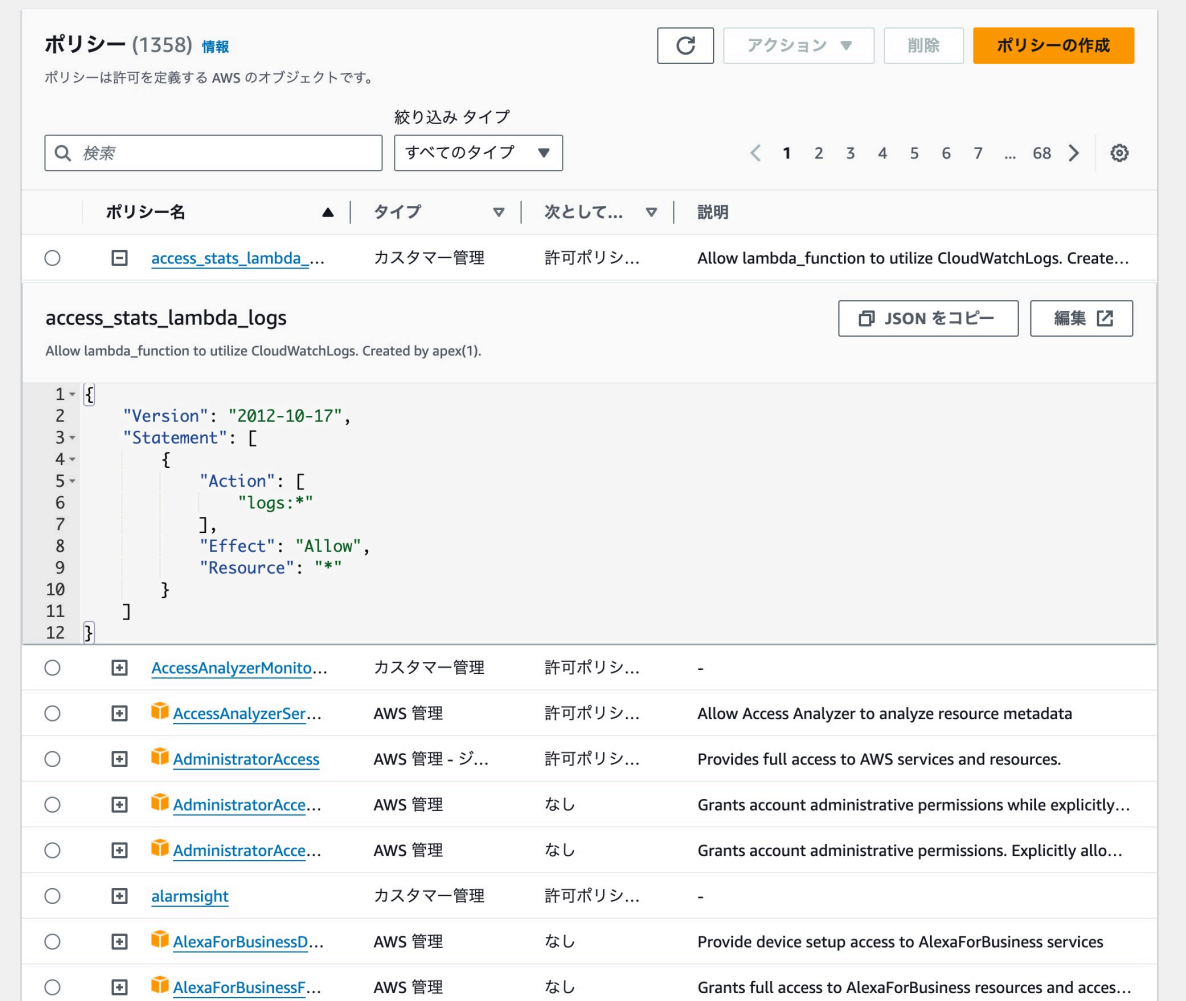
(これまで) Lambda GetFunction APIで `lambda:ListTags` がなくてもタグが取得できた
(これから) `lambda:ListTags` 権限がないとタグが取得できなくなる

このアカウントには
`lambda:GetFunction` があるけど `lambda:ListTags` がないポリシーがあるよ
期日までに `lambda:ListTags` を許可しないとタグが取れなくなるよ！

というお知らせ
しかし **「どのポリシーが該当しているかは教えてくれない」**

どうやってIAMポリシーを洗い出すか

マネコン目視は……(1358) ←



AWSの人に聞いたら教えてくれた方法

`aws iam get-account-authorization-details`

```
            },
            {
                "Document": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Action": [
                                "iam:CreateRole",
                                "iam:CreatePolicy",
                                "iam:AttachRolePolicy",
                                "iam:PassRole",
                                "lambda:GetFunction",
                                "lambda:CreateFunction",
                                "lambda:DeleteFunction",
                                "lambda:InvokeFunction",
                                "lambda:GetFunctionConfiguration",
                                "lambda:UpdateFunctionConfiguration",
                                "lambda:UpdateFunctionCode",
                                "lambda:CreateAlias",
                                "lambda:UpdateAlias",
                                "lambda:GetAlias",
                                "lambda:ListVersionsByFunction",
                                "logs:FilterLogEvents",
                                "cloudwatch:GetMetricStatistics"
                            ],
_
```

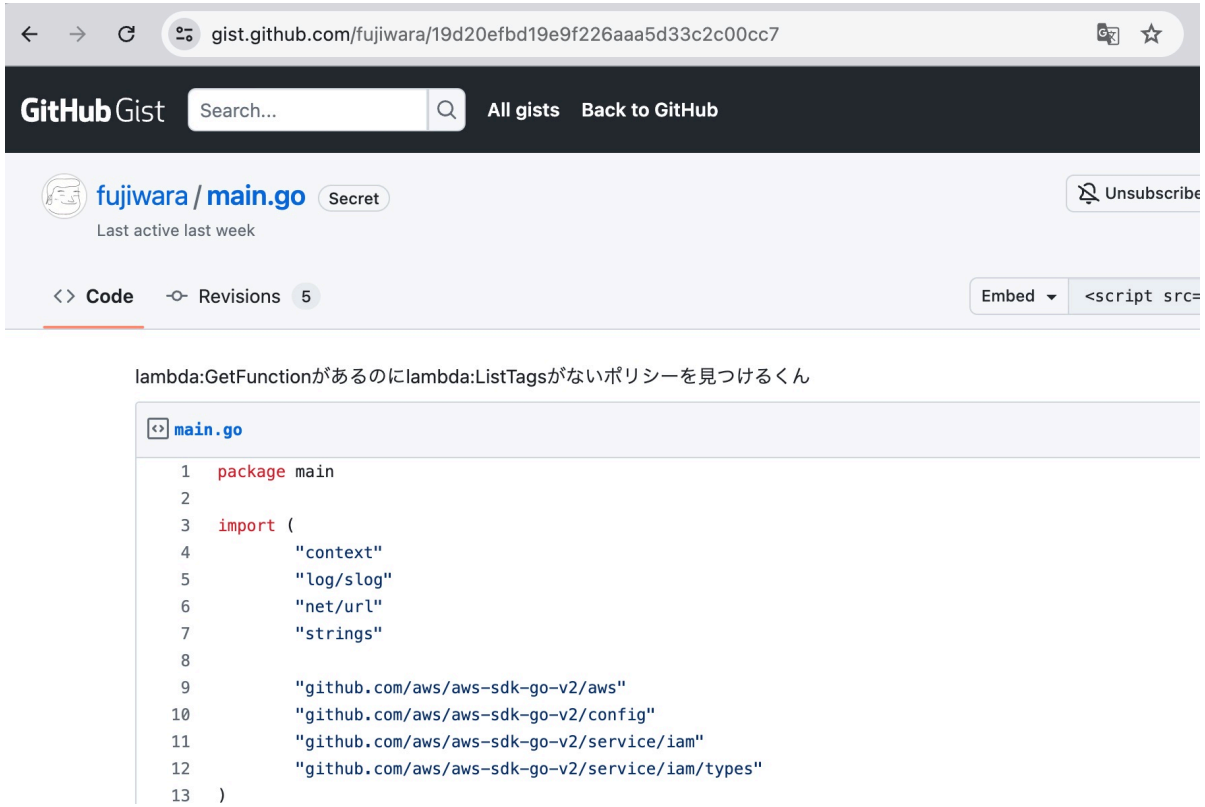
多少マシだが……結局目視?

`jq` 芸は意外と難しい

Policy JSONは配列でも文字列でもよい要素(`Action` や `Resource`)があるため

しかたないなあ…

https://gist.github.com/fujiwara/19d20efbd19e9f226aaa5d33c2c00cc7



ぱっと AWS SDK Go でコードを書いた

1. `GetAccountAuthorizationDetails` APIでアカウント内の許可一覧を取得
   User, Group, Role, Policy(AWSマネージドを含む)
2. ポリシーの文字列に対して雑に評価

```
func detect(s string) bool {
  d, _ := url.QueryUnescape(s) // APIの結果はURL escapeされている
  l := strings.ToLower(d)      // Actionは大文字小文字を区別しないので小文字に揃える
  return strings.Contains(l, `"lambda:getfunction"`) && !strings.Contains(l, `"lambda:listtags"`)
}
```

3. できた！！

結果

見つかったもの

`AWSSupportServiceRolePolicy` ← AWSのマネージドポリシー

どうしろと…

AWSマネージドポリシーを何かにattachしてると問題が起きるので
通知自体は仕方ないんだけど…

ECSでもなんかメール来てる

> [要対応] IAM ポリシーを更新して明示的な ecs:TagResource の「許可」を追加してください

ECSでも`ecs:CreateCluster`時に`ecs:TagResource`が必要になると言われている
特定の条件でポリシーを探したいことはこれからも多そう

💡 評価式部分を汎用化したら使い回せるのでは?

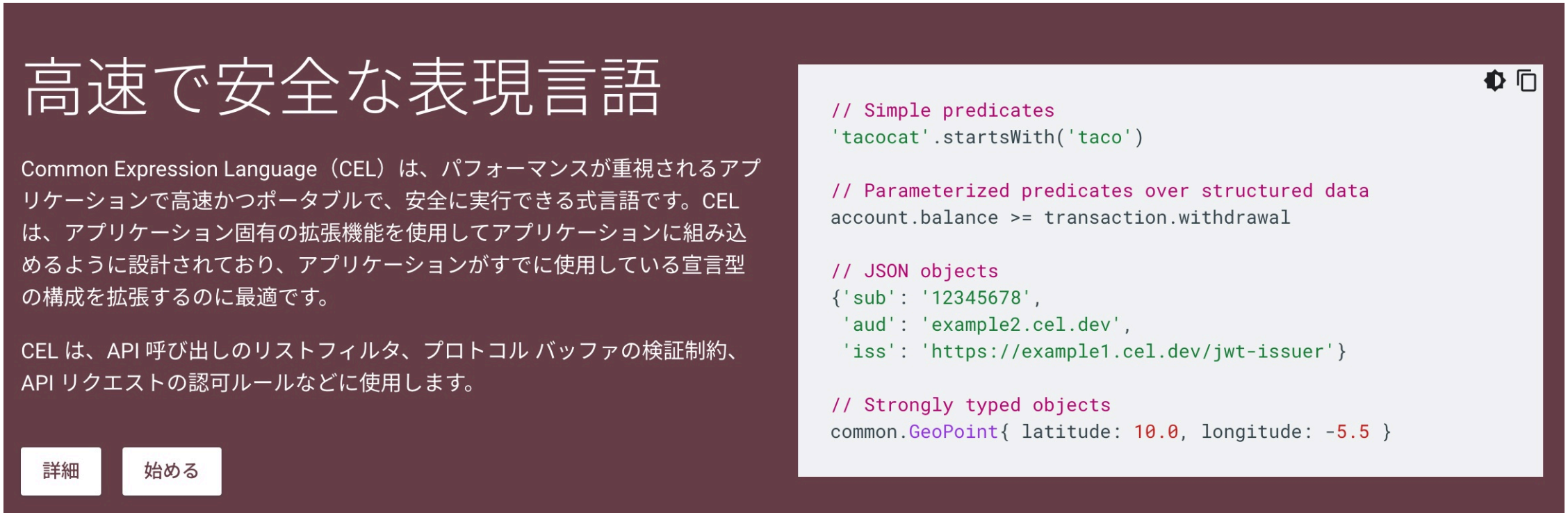
この部分をハードコードではなく外部から与えれば...

```
return strings.Contains(l, `"lambda:getfunction"`) && !strings.Contains(l, `"lambda:listtags"`)
```

外部から与えた式で値を評価する

Common Expression Language(CEL) https://cel.dev

Googleが開発した式言語



CELのいいところ

- 式評価を自作するのは大変
- 汎用言語(Lua, JavaScript, mruby...)を組み込むと、なんでもできすぎる
  - 🔨を持つと全てが釘に見える現象
  - チューリング完全な言語には停止問題がある(無限ループするかも?)
- CELは非チューリング完全で式評価に特化、高速、安全に実行可能
- Go, Java, C++に組み込み可能
  - GoogleCloud, Kubernetes, istioなどで既に実用されている
- 型あり (bool, int, uint, double, string, timestamp, duration...)
- 演算子 `!` , `&&` , `||` , `==` , `!=` , `>` , `<` ...
- 文字列関数 `matches` (正規表現) `startsWith` ...

CEL 式の例

```
// 数値の範囲チェック
age >= 18 && age < 65

// 文字列の前方一致や正規表現マッチ
name.startsWith('Foo')
name.matches('^Foo.*')

// リスト内の要素に条件を満たすものがあるか
roles.exists(role, role == 'admin')

// 日付、時間の比較
request.date < timestamp('2023-12-31T23:59:59Z')
timeout >= duration('10s')
```

これらを組み合わせて評価した結果を任意の値(boolだけではなく)で返せる

IAMポリシーを探すコードにCELを組み込み

できました **github.com/fujiwara/iam-policy-finder**

fujiwara/**iam-policy-finder**

iam-policy-finder is finder of AWS IAM Policies.

3 Contributors　0 Issues　1 Star　0 Forks

1. GetAccountAuthorizationDetails でポリシーを取得
2. ポリシーをCELで式評価をして真になるものを出力

名前で検索する例

```
$ iam-policy-finder 'Name == "AmazonEC2FullAccess"'
```

```
time=2024-07-18T17:37:13.110+09:00 level=INFO msg="starting scan" expr="Name == \"AmazonEC2FullAccess\"" filter=[]
time=2024-07-18T17:37:26.272+09:00 level=INFO msg=found policy=AmazonEC2FullAccess versions="[v5 v4 v3 v2 v1]" attached=2
time=2024-07-18T17:37:33.377+09:00 level=INFO msg=finished found=5 scanned=975
```

- `Name` : policy名

JSONを文字列比較する例

「Documentに`"lambda:GetFunction"`があり`"lambda:ListTags"`がない」

```
Document.matches('"lambda:[Gg]et[Ff]unction"')
&&
!Document.matches('"lambda:[Ll]ist[Tt]ags"')
```

- `Document`: ポリシーのJSON文字列
- `matches`: 文字列に対して正規表現マッチ

IAM policy actionは **大文字小文字を区別しない(!!)** ので同一視するために正規表現

正規化したポリシーを評価する例

「`s3:*` を許可している」

```
Statement.exists(s, (s.Action.exists(a, a == "s3:*") && s.Effect == "Allow"))
```

- `Statement` : 正規化したpolicy statement
  - `Action` , `Resource` は文字列でもリストでも許されるのですべてリストに
  - `"Resource":"*"` → `"Resource":["*"]`
- `exists(要素, 式)` : リストの要素を式で評価してどれかが真になれば真

Actionを小文字に正規化して比較

`s3:GetObject`, `s3:getobject`, `S3:getObject`...←全部同じ意味
CELには大文字小文字を変換したり無視して比較する方法が(デフォルトでは)ない

`--lc` オプション 正規化時にAction要素を全部小文字にする

「Actionに `lambda:GetFunction` があるが `lambda:ListTags` がない」

```
Statement.exists(s,
  (
    s.Action.exists(a, a == "lambda:getfunction")
    &&
    !s.Action.exists(a, a == "lambda:listtags")
  )
)
```

`Document` (JSON文字列)は小文字になりません
(ユーザー定義関数が作れるので今後追加予定)

ECSの件を早速調べてみた

「`ecs:CreateCluster`があるが`ecs:TagResource`がないもの」

```
// ecs.cel
Statement.exists(s, (
    s.Action.exists(a, a == "ecs:createcluster")
    &&
    !s.Action.exists(a, a == "ecs:tagresouce")
))
```

```
$ iam-policy-finder ecs.cel --dump --lc
```

- 式はファイルに書いてもよい
- `--dump` : 見つかったポリシーJSONを出力
- `--lc` : Actionを小文字に正規化

```
$ iam-policy-finder --lc ecs.cel --dump
time=2024-07-18T18:02:23.881+09:00 level=INFO msg=found
policy=AmazonEC2ContainerServiceforEC2Role versions="[v7 v6 v5 v4 v3 v2 v1]" attached=1
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:DeregisterContainerInstance",
        "ecs:DiscoverPollEndpoint",
        "ecs:Poll",
        "ecs:RegisterContainerInstance",
        "ecs:Submit*"
      ],
      "Resource": "*"
    }
  ]
}
```

AmazonEC2ContainerServiceforEC2Role ← AWSマネージドポリシー!!!!