Maximizing the Potential of AWS WAF

Fully Automating threat detection with Custom Managed Rules and Proprietary Threat Intelligence

Shota Sugawara, Hirofumi Kawauchi

NTT-ME CORPORATION

About Me

Shota Sugawara

Cloud Security Operations Engineer

Work Experience

5+ years' background in cyber security

- Tech-lead of Cloud Security Operation Services
- Vulnerability assessments and security testing

Public Speaking

- AWS official Seminar
  Web Security Measures Against Advanced Cyberattacks

External Activities

- CTF Competitions

About Me

Hirofumi Kawauchi, PhD.
SOC Manager

Work Experience

10+ years' background in cyber security
- SOC Manager for Cloud Security Operations
- Led incident response, vulnerability management
- Launched Managed Security Service
- SOC analyst, threat intel, device management

Public Speaking

- BSides Las Vegas 2024
- Interop Tokyo 2023
- ICT-ISAC, university classes, several events, etc.

Certificates

- CISSP, GCFA, GPEN, AWS-SAP/SCS
- NTT Group Certified Security Principal

Agenda

Enhancing and Optimizing the Use of AWS WAF Managed Rules

- Our SOC Service with AWS WAF
- Challenges in WAF operation
- SOC-defined rules
- Blocking Decision Process

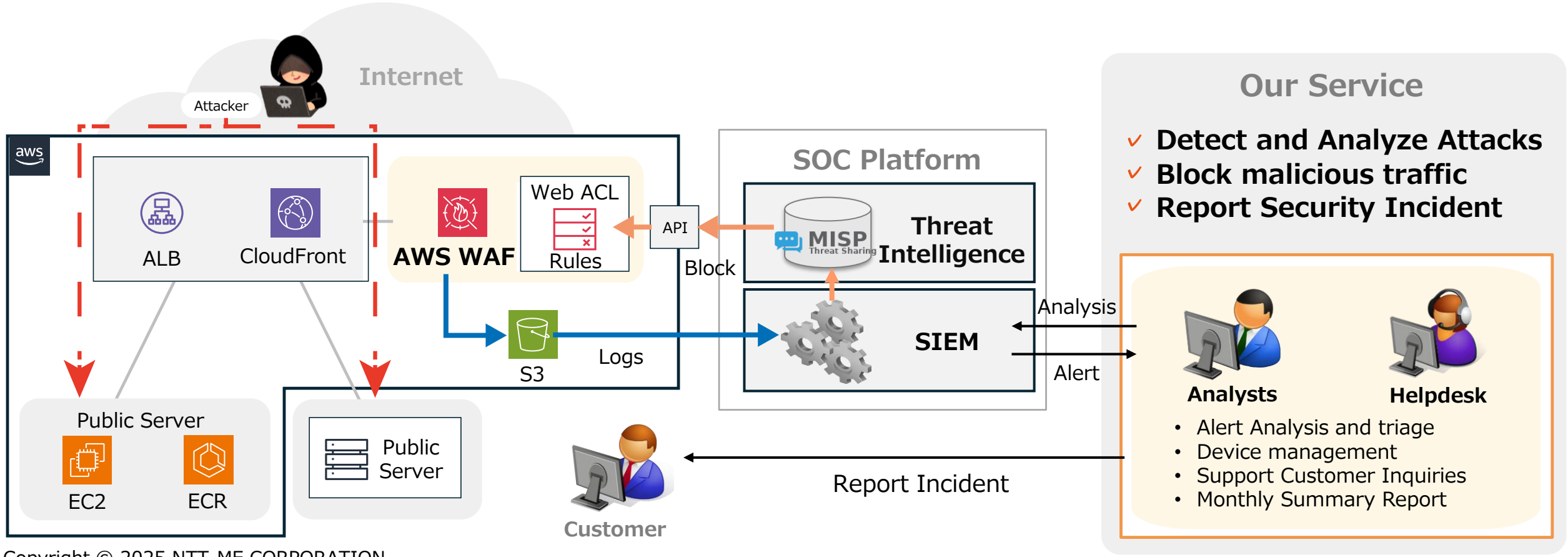
Automated Attack Detection and Blocking with Threat Intelligence

- Background of automation
- Automated Detection and Blocking System
- Effectiveness of Our New System

Enhancing and Optimizing the Use of AWS WAF Managed Rules

Our SOC Service with AWS WAF

- Offer 24/7 managed security services to protect customer's assets such as servers, endpoints, cloud environments, etc.
- Protect public servers using **AWS WAF, our Proprietary Threat intelligence, SIEM, and custom detection and response functions**



Challenges in WAF operation

- **We aim to fully leverage AWS managed rules for efficient WAF operation**
  - **Reduce false positives and improve detection accuracy**

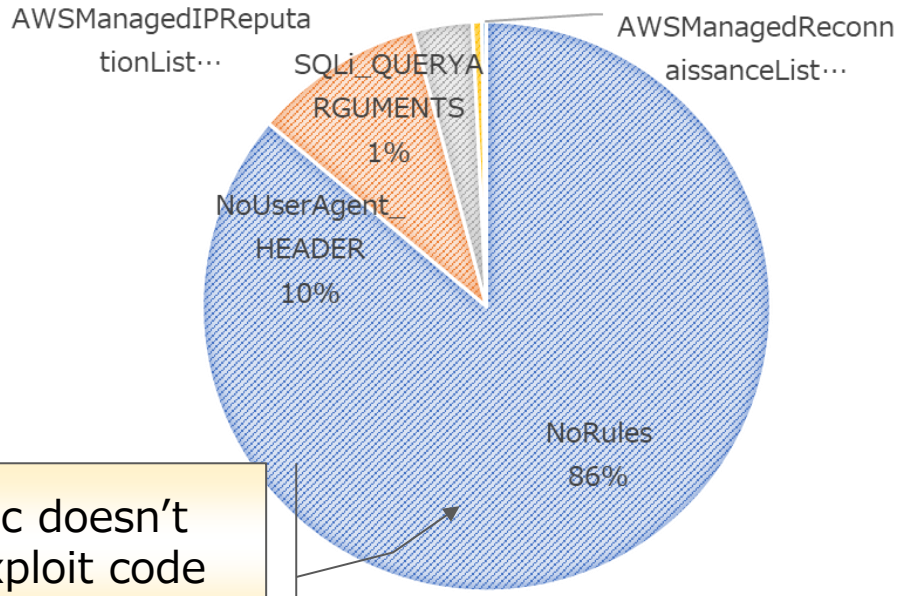
- **To meet these goals, we require the implementation of fine-grained rules**
  - **Inappropriate responses may reduce the effectiveness of defenses and result in remaining risks**

Response to False Positives	Common Pitfalls
Exclude the matched rules when false positives occur • Changing mode from "Block" to “Count"	**Should be determined by IP, country, and path** • Uniform change may result in the waste of effective rules
Add the IPs to the allowlist when false positives occur • Adding all IPs when reported by end users	**Allowlist should be limited to the minimum necessary** • Exercise caution when adding to the allowlist, even if the source country is Japan

AWS Managed Rules : HostingProviderIPList

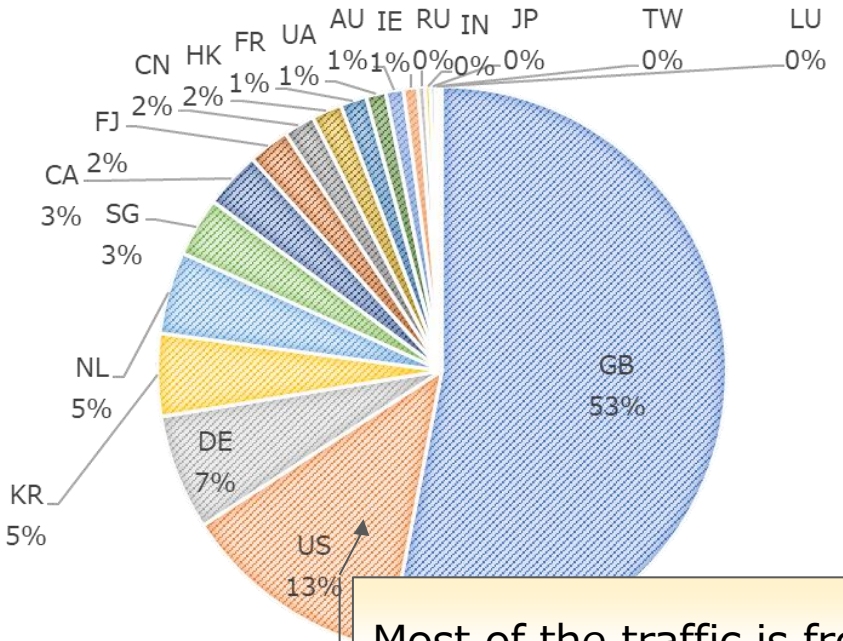
- **HostingProviderIPList**
  - Blocking access from **cloud provider IPs** (non-end-user traffic)
  - **Risk of unintentionally blocking ISP housing or DaaS service IPs**

Attacks detected simultaneously with HostingProviderIPList



Most traffic doesn't contain exploit code

Sources detected by HostingProviderIPList

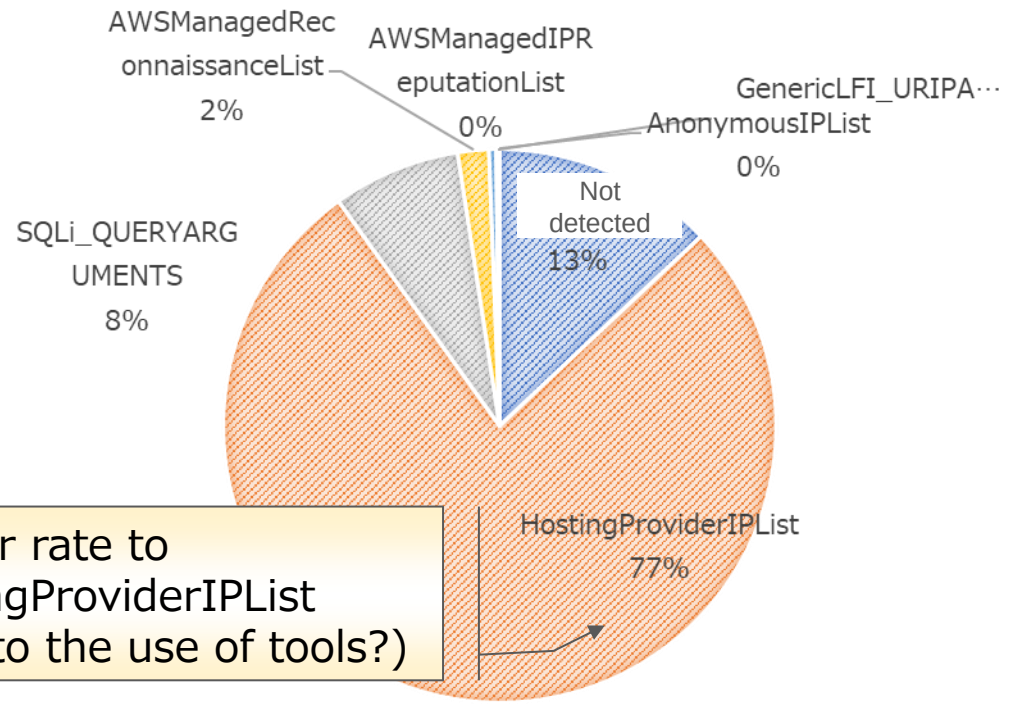


Most of the traffic is from overseas (Could it really be legitimate?)

AWS Managed Rules : NoUserAgent_HEADER

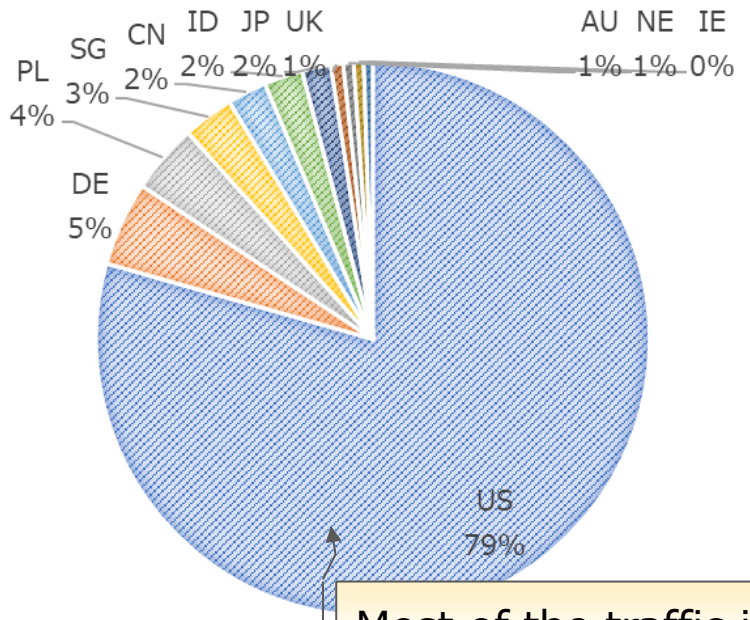
- **NoUserAgent_HEADER**
  - Blocking requests **without a User-Agent header** (non-browser traffic)
  - **Risk of blocking legitimate end-user access without User-Agent**
    (Tools or API integration, etc.)

Attacks detected simultaneously with NoUserAgent_HEADER



- Similar rate to HostingProviderIPList (Due to the use of tools?)

Sources detected by NoUserAgent_HEADER

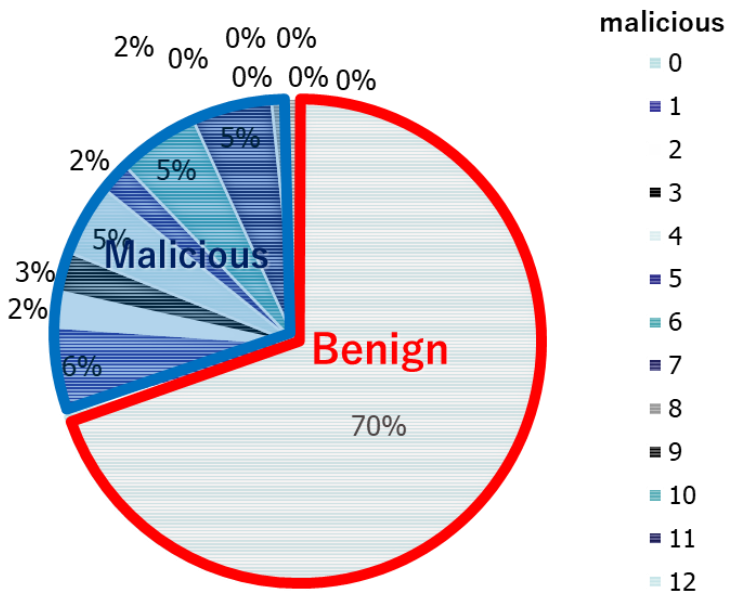


Most of the traffic is from overseas (Could it really be legitimate?)

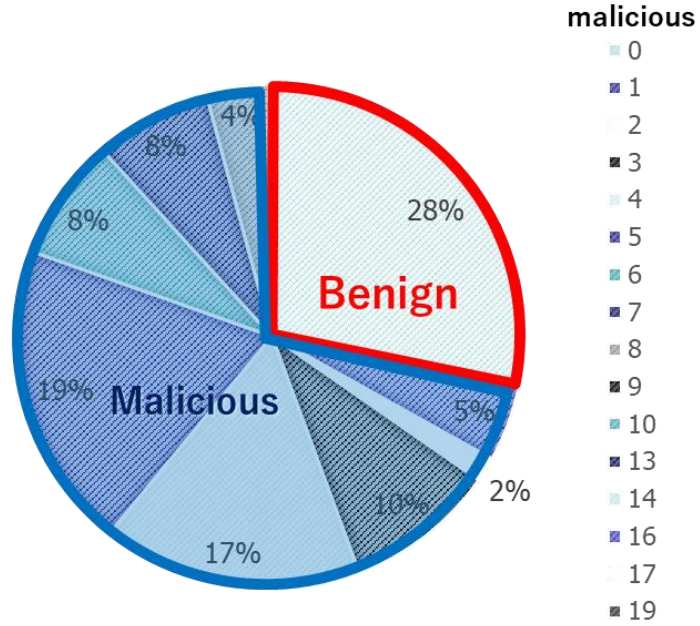
Characteristics of traffic

- **Improve detection accuracy by considering more traffic characteristics**
  - Target rules: HostingProviderIPList, NoUserAgent_HEADER, etc.
  - **Exclude from blocking based on the characteristics of destination path and source IP**
  - **Determine whether to block traffic based on source country**

Maliciousness
Source Country: Japan, US

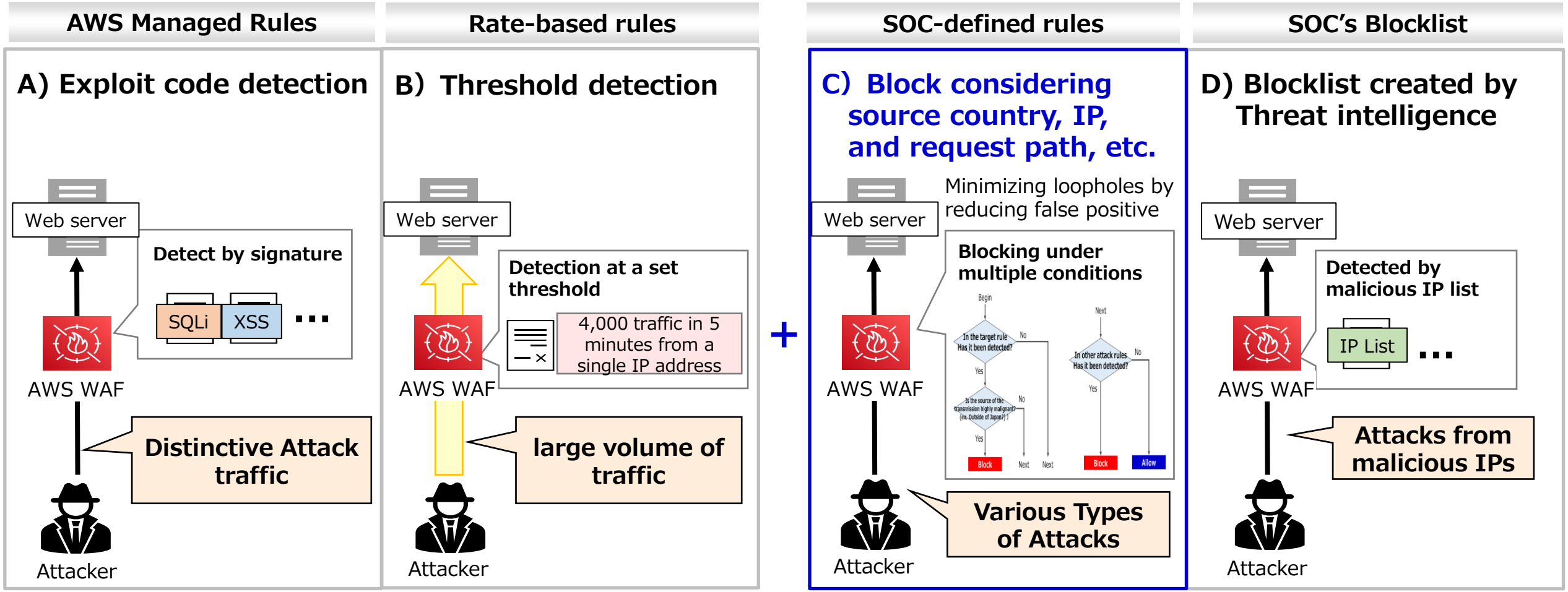


Maliciousness
Source Country: Except Japan, US



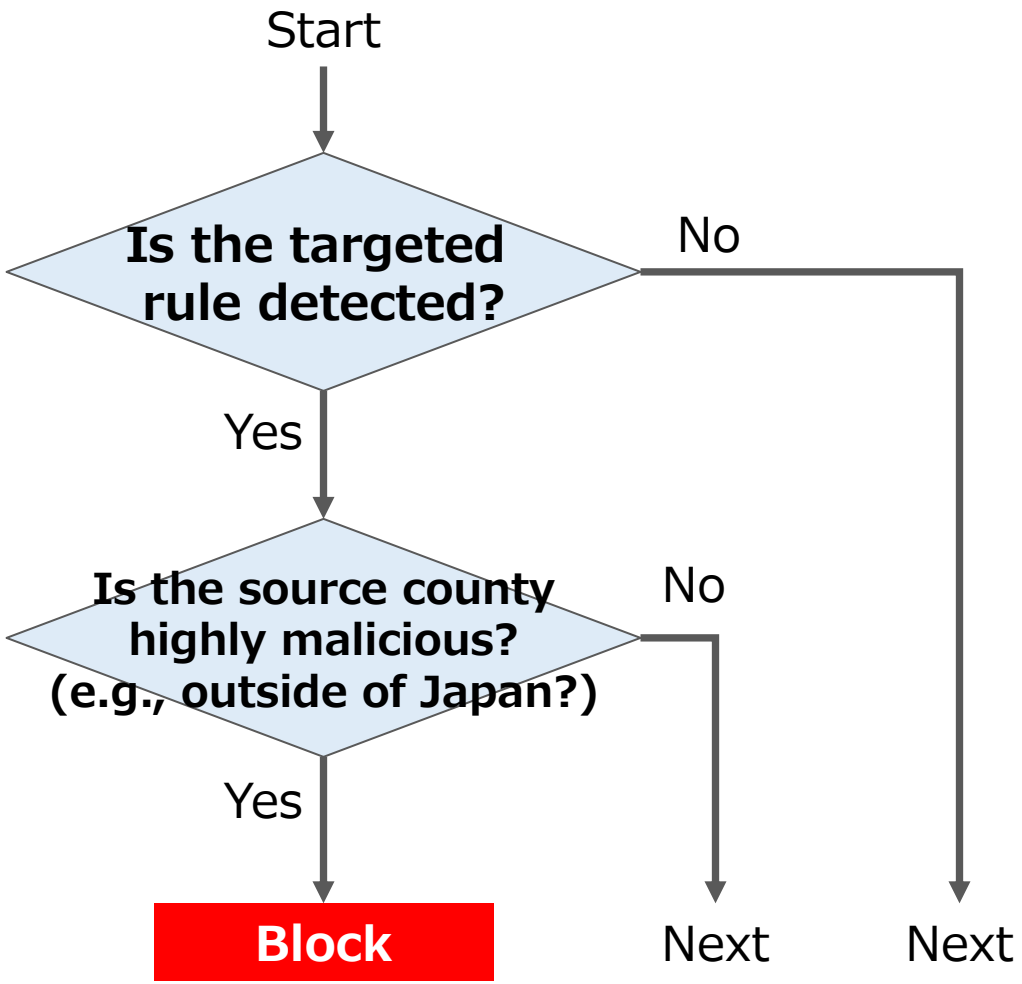
Our Approach: SOC-defined rules

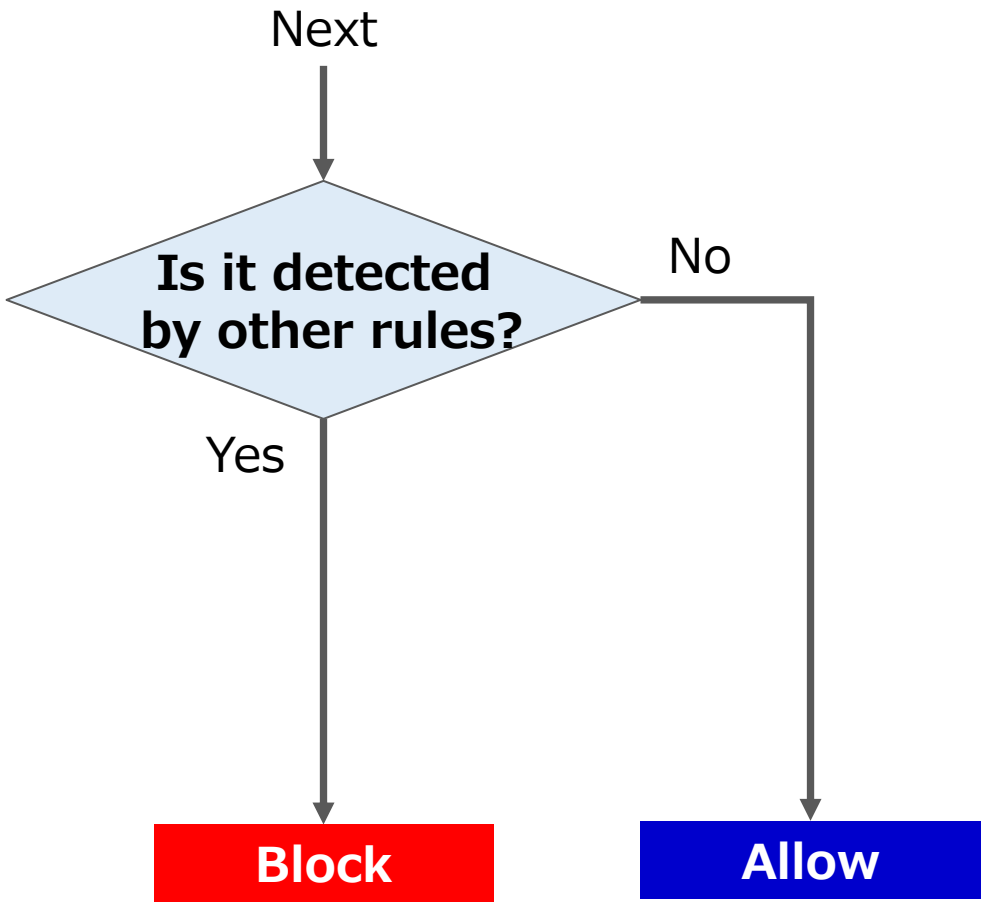
- **Combine AWS pre-built functions with our custom functions**
  - **AWS pre-built functions:** AWS Managed rules (A), Rate-based rules (B)
  - **Our custom functions:** SOC-defined rules (C), Threat intelligence–based blocklist (D)



Blocking Decision Workflow

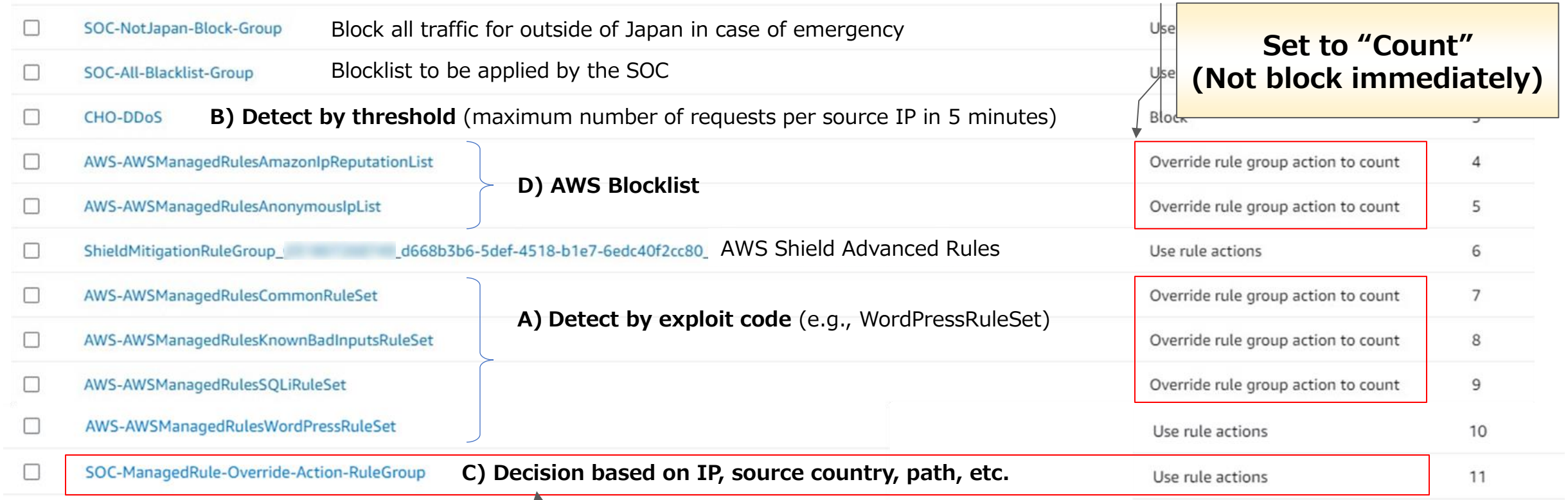
- **Evaluate whether to block based on the detected rule, source country, etc.**





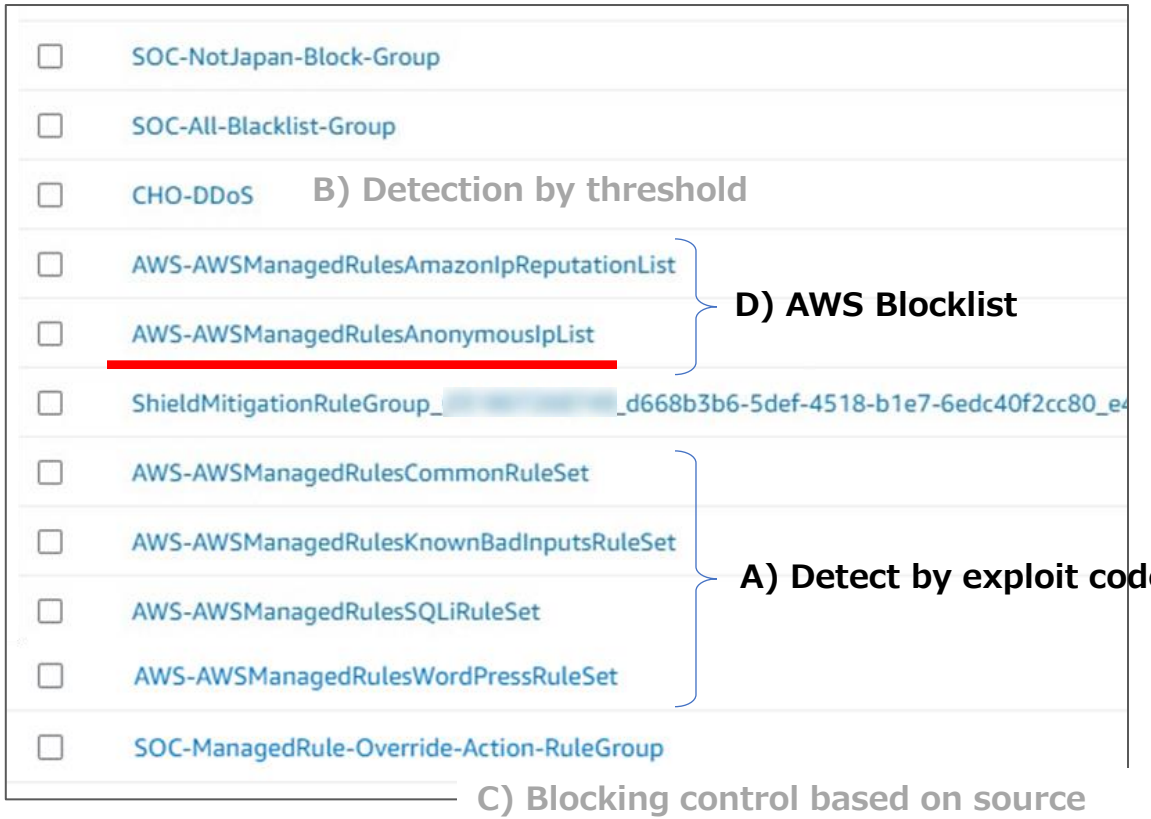
How to apply to WebACL

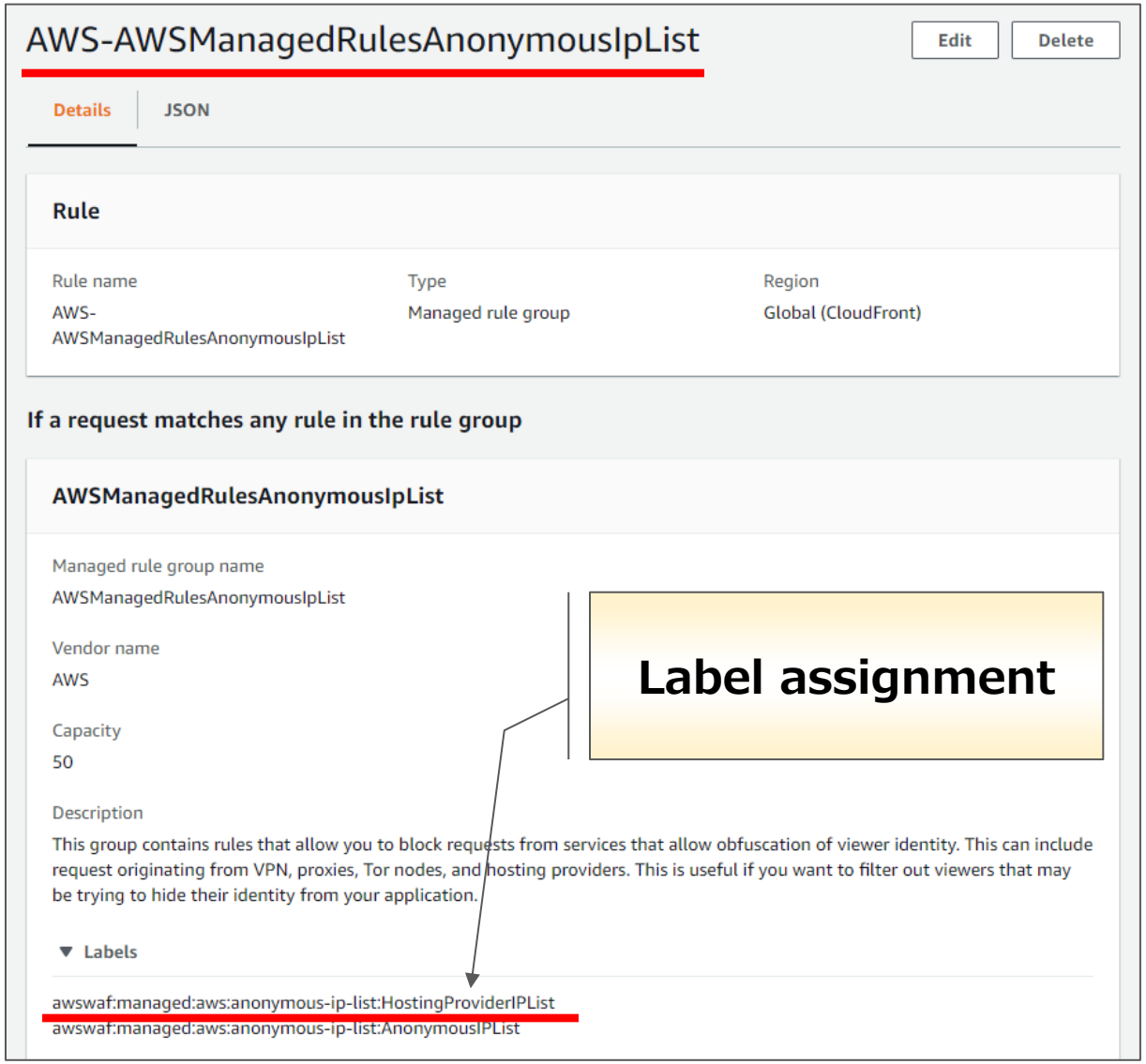
- **Set all detected rules in count mode. Then, determine whether to block**
- **Determine if block based on the source country using GeoIP and the alert details**



Blocking decision process

1. **Assign labels of the matched rules.**
   - e.g., Labeling HostingProviderIPList

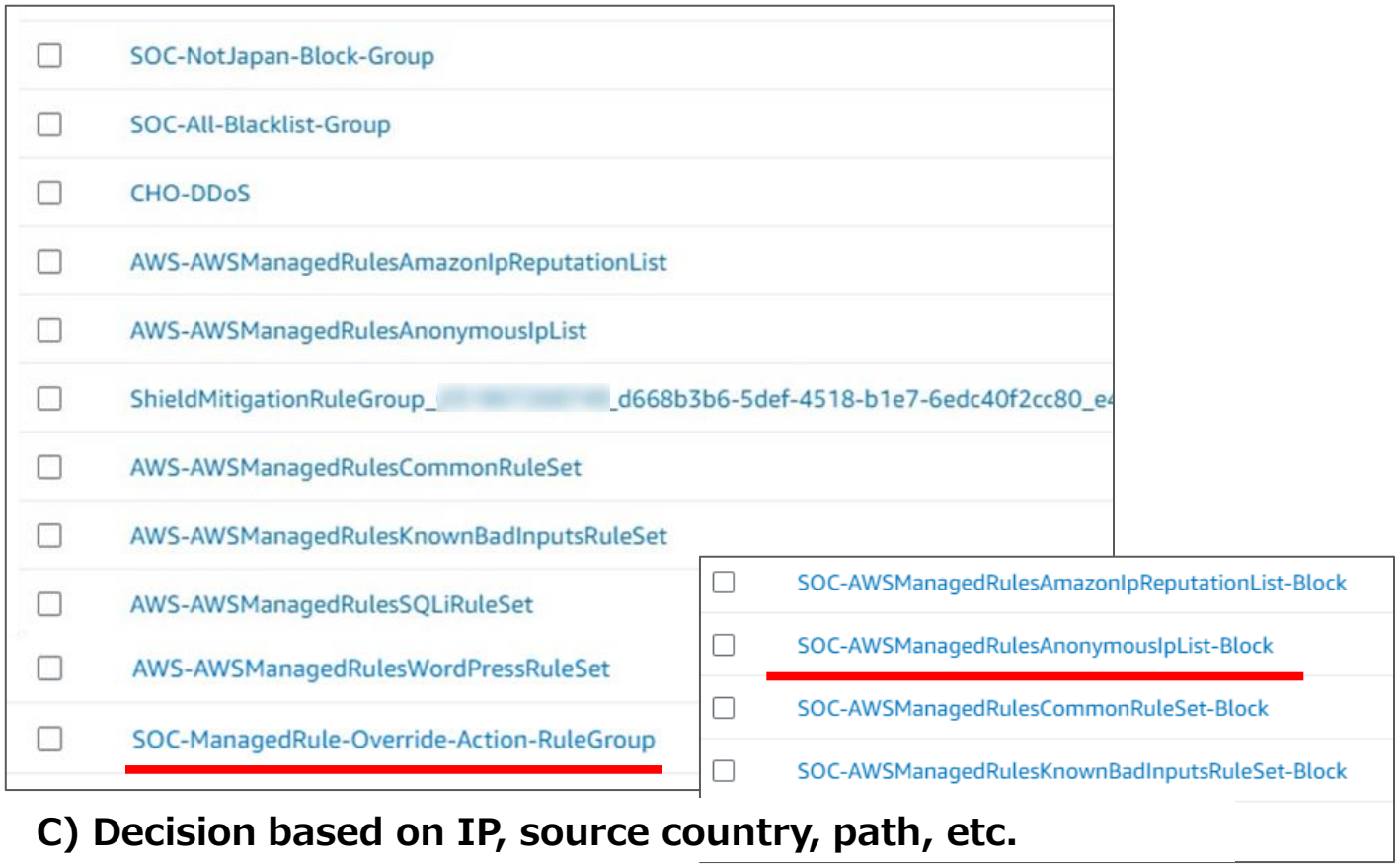




Blocking decision process

2. Take actions based on matched rules and source countries

- e.g., Labels such as HostingProviderIPList
- e.g., Country excluding JP or US



C) Decision based on IP, source country, path, etc.

```
{
  "Name": "SOC-AWSManagedRulesAnonymousIpList-Block",
  "Priority": 1,
  "Statement": {
    "AndStatement": {
      "Statements": [
        {
          "LabelMatchStatement": {
            "Scope": "NAMESPACE",
            "Key": "awswaf:managed:aws:anonymous-ip-list:HostingProviderIPList"
          }
        },
        {
          "NotStatement": {
            "Statement": {
              "GeoMatchStatement": {
                "CountryCodes": [
                  "JP",
                  "US"
                ]
              }
            }
          }
        }
      ]
    }
  },
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "SOC-AWSManagedRulesAnonymousIpList-Block"
  }
}
```

Does the label match?

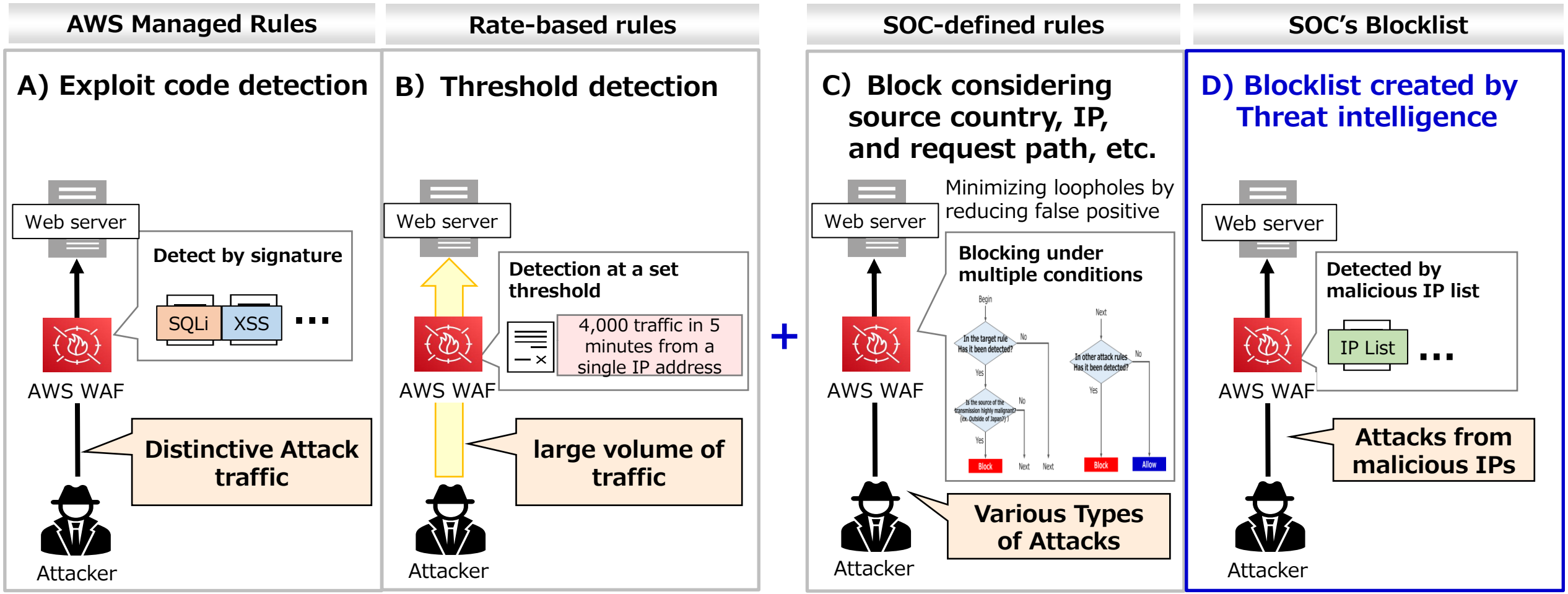
Are the country codes different from JP or US?

If yes, then it can be blocked

Automated Attack Detection and Blocking with Threat Intelligence

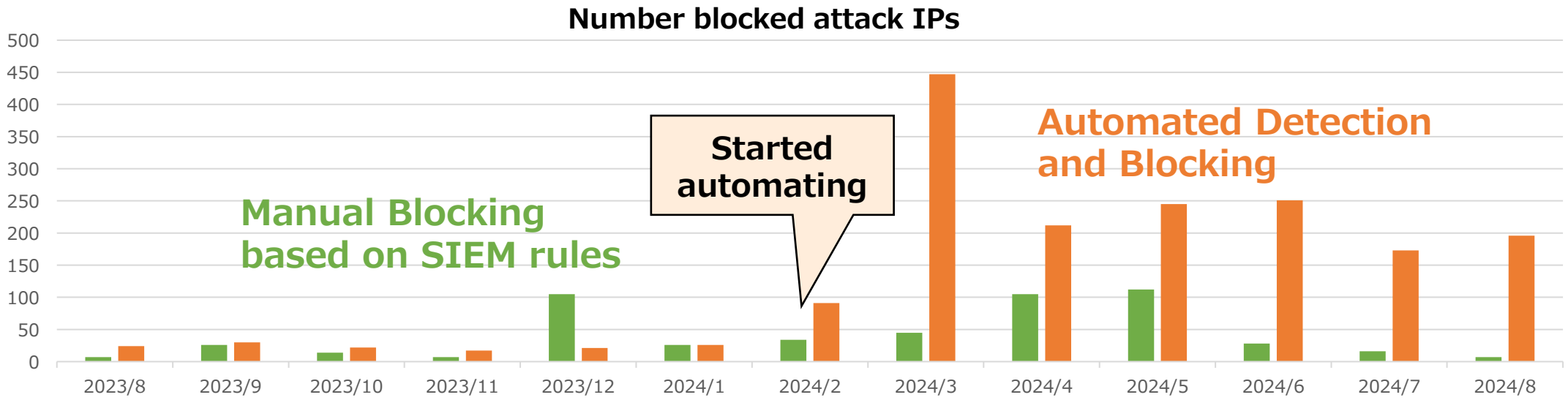
Our Approach: Threat intelligence–based blocklist

- **Combine AWS pre-built functions with our custom functions**
  - **AWS pre-built functions:** AWS Managed rules (A), Rate-based rules (B)
  - **Our custom functions:** SOC-defined rules (C), **Threat intelligence–based blocklist (D)**



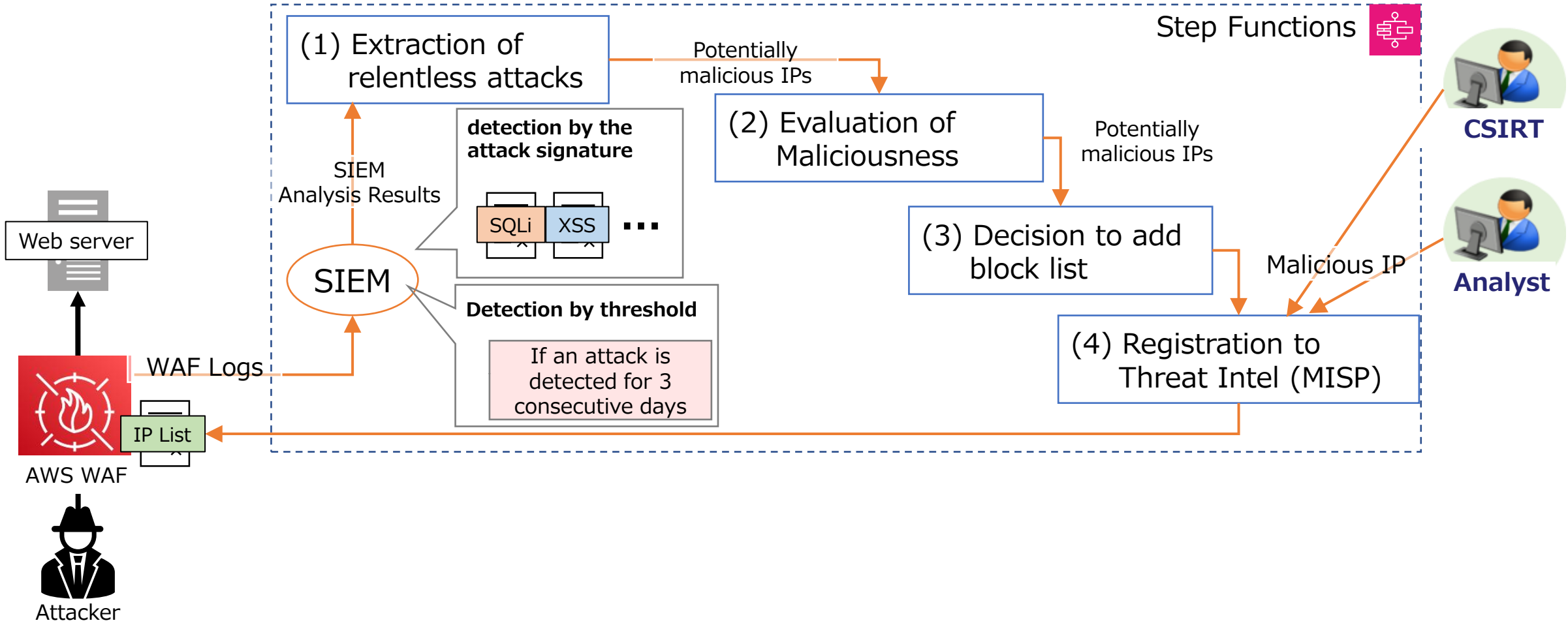
Background of automation

- **Previously, persistent attacks were detected by SIEM rules and manually blocked by analysts**
  - However, SIEM rule coverage may be incomplete, raising concerns about undetected threats
- **Newly launched daily inspection from a different perspective (approx. 1 hour/day)**
  - Short-term threshold exceeded ⇒ Extract long-term attacks below the threshold
  - Single-site attacks ⇒ Extract cross-site attacks across multiple sites
- **Finally, achieved fully automated detection and blocking system**



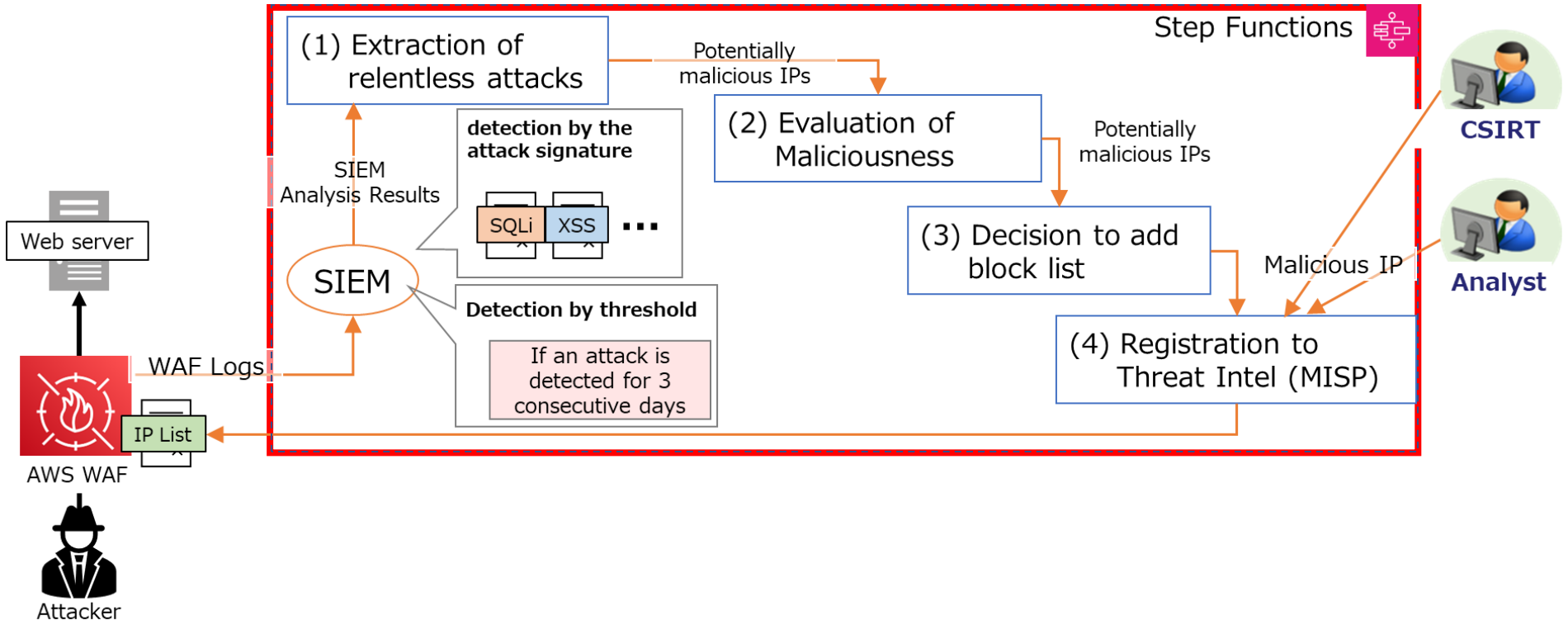
Automation of attack detection and blocking

- **Effectively mitigate persistent attacks with automated functions**
  - Built a unified system by integrating AWS services with external sources and threat intelligence



Automated detection and blocking system

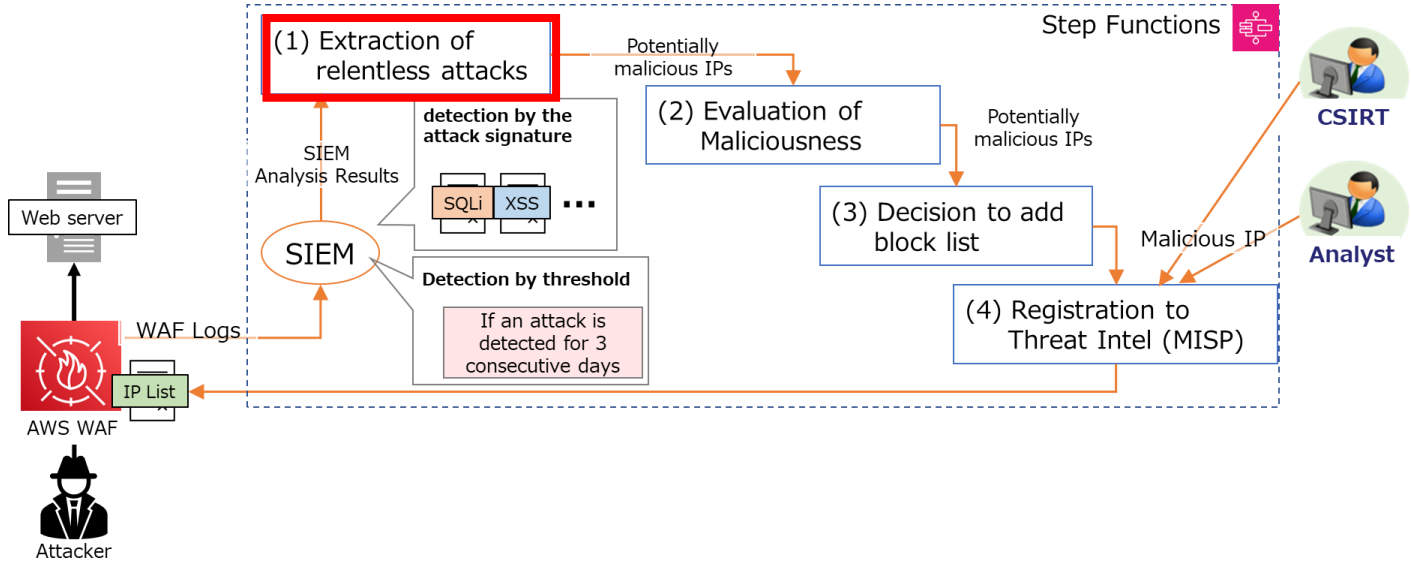
- **Automate the registration of blocked source IPs:**
  (1) Extraction > (2) Evaluation > (3) Decision > (4) Registration
- **Achieve full automation by orchestrating components using AWS Step Functions**
  - Each component (1) - (4) is AWS Lambda functions and maintained by multiple members
  - Utilizing AWS Step Functions for control of (1) - (4)

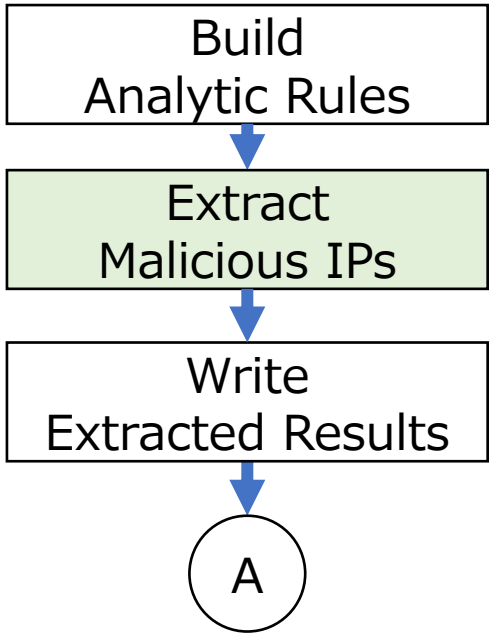


Automated detection and blocking system

Respond to various attack patterns and extract malicious IPs

- Attack Pattern (1): Slow attack
  - Evidence : Little accesses every day
  - Extraction : If found block logs for 3 consecutive days
- Attack pattern (2): Scanning activity
  - Evidence : Significant amount of access in a short period of time using vulnerability scanners
  - Extraction : If found 100 block logs per X hours

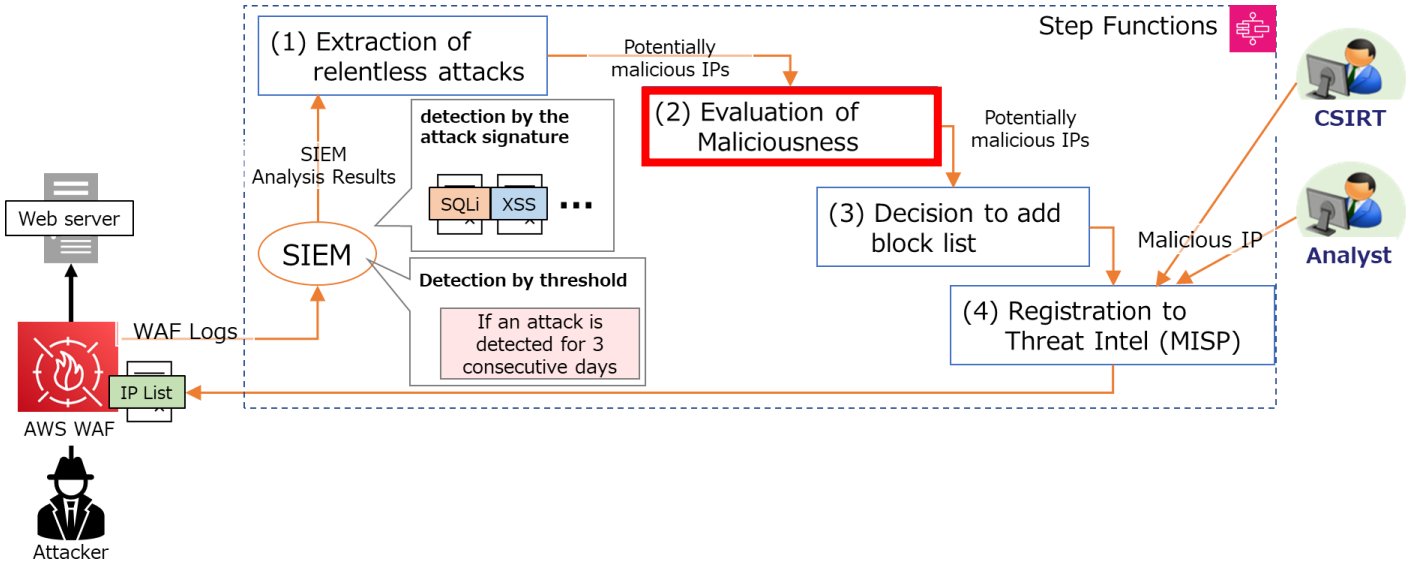


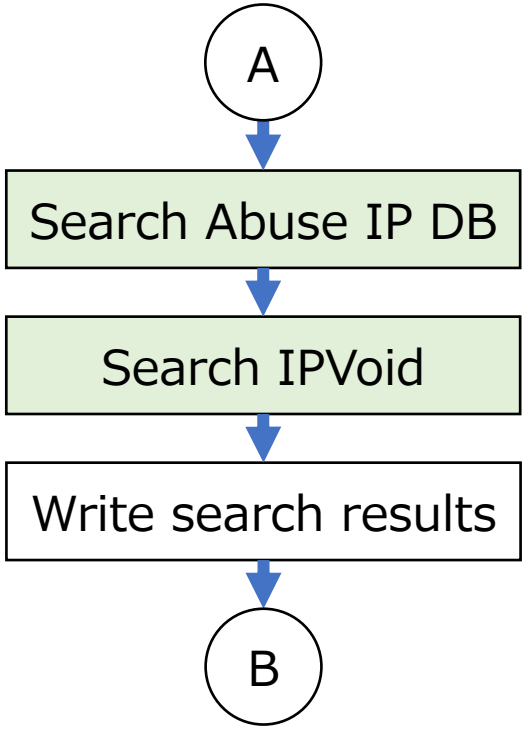


Automated detection and blocking system

Investigate whether the IPs are truly malicious

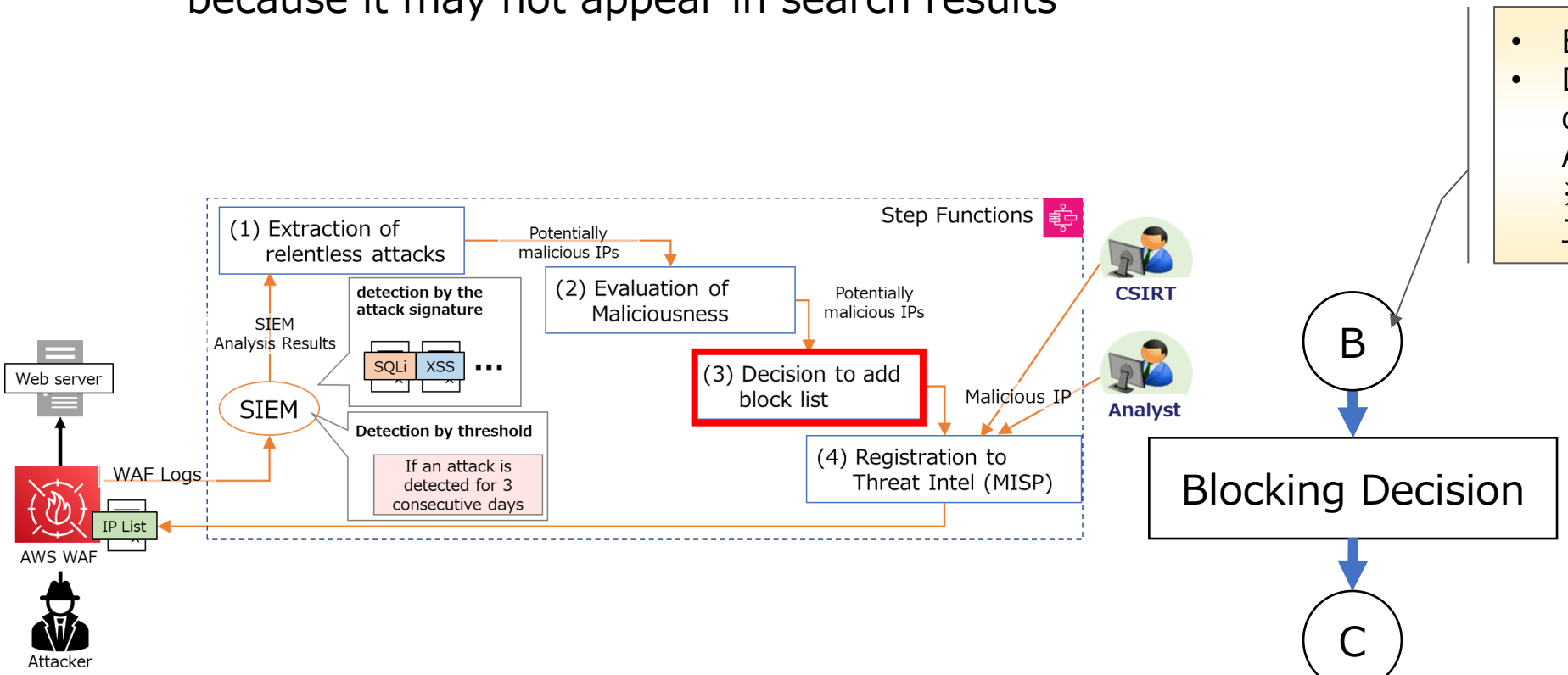
- IPVoid and Abuse IP DB: Utilize two reputation services considering Pros/Cons
  - IPVoid : Able to acquire reputations from various services, although the search may take time
  - Abuse IP DB: Swift Search and acquire community reputation results
- Evaluation of hundreds of IPs may occur timeouts ⇒ Step Functions

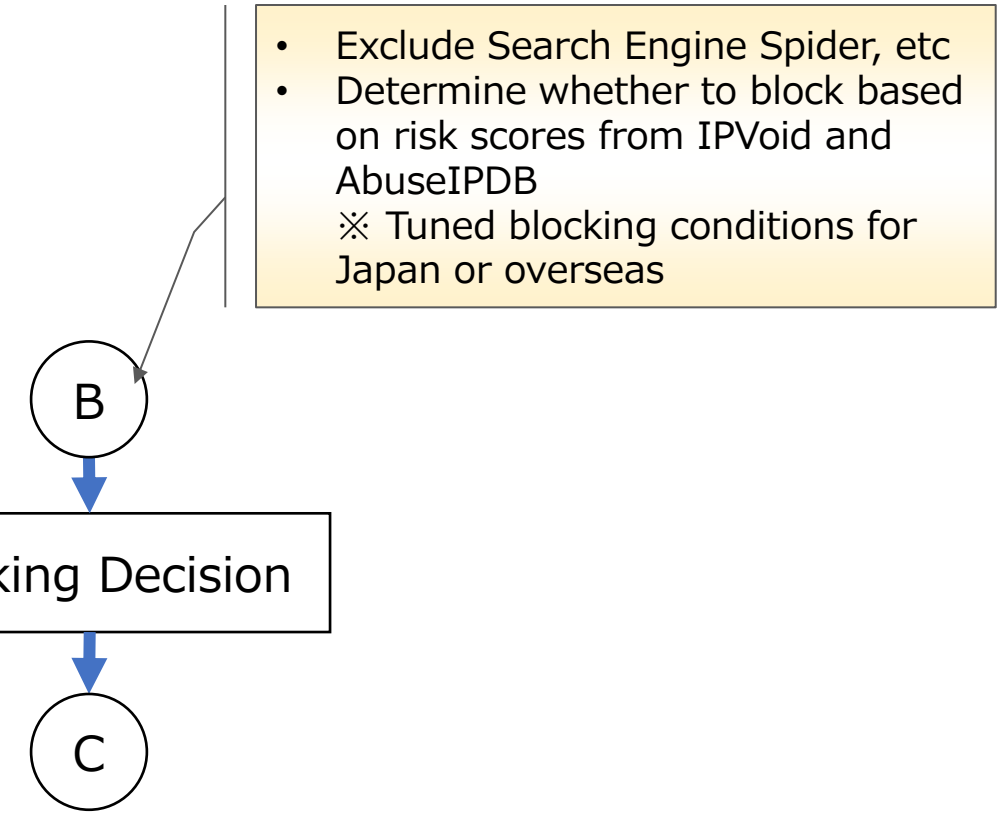




Automated detection and blocking system

- **Avoid "inadvertently" blocking IPs required for business purposes**
- **Screening to prevent false positive blocking**
  - e.g., Do not block if the IP is used for Search Engine Spider (Search site crawler) because it may not appear in search results

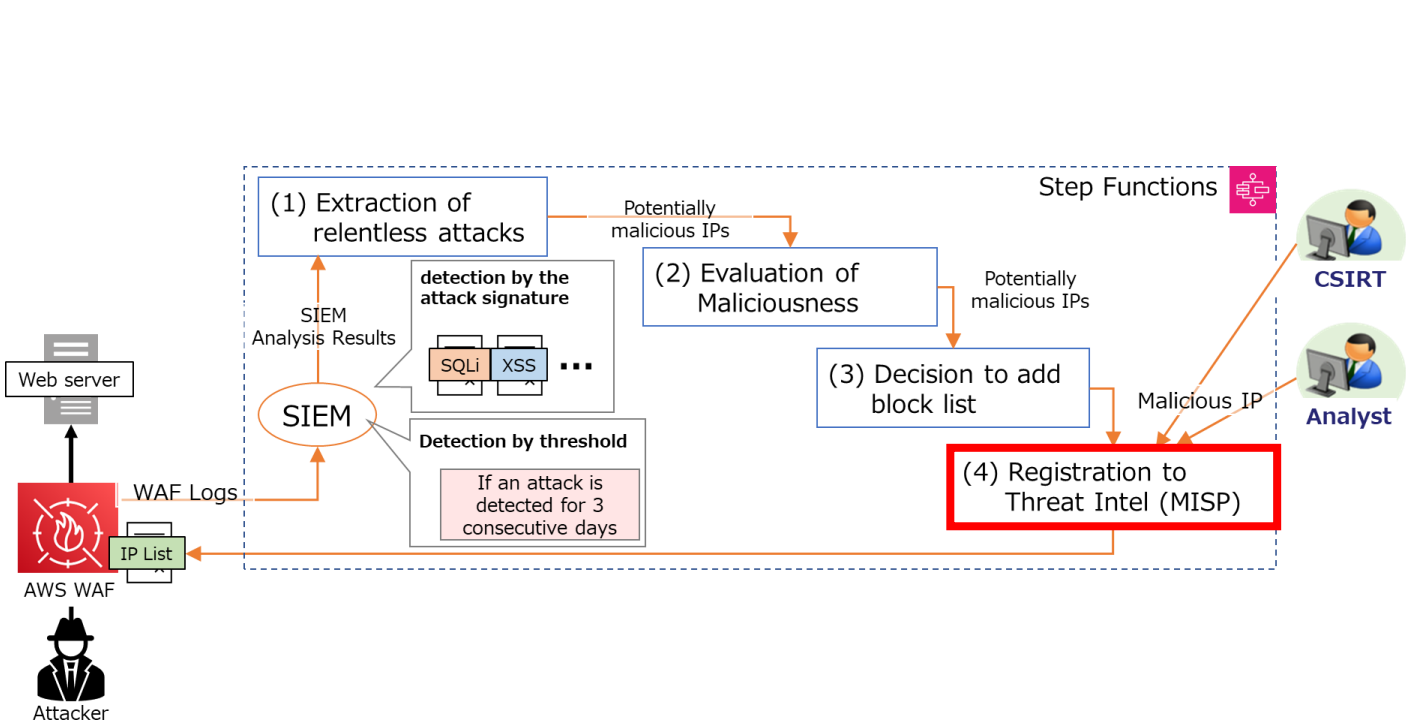


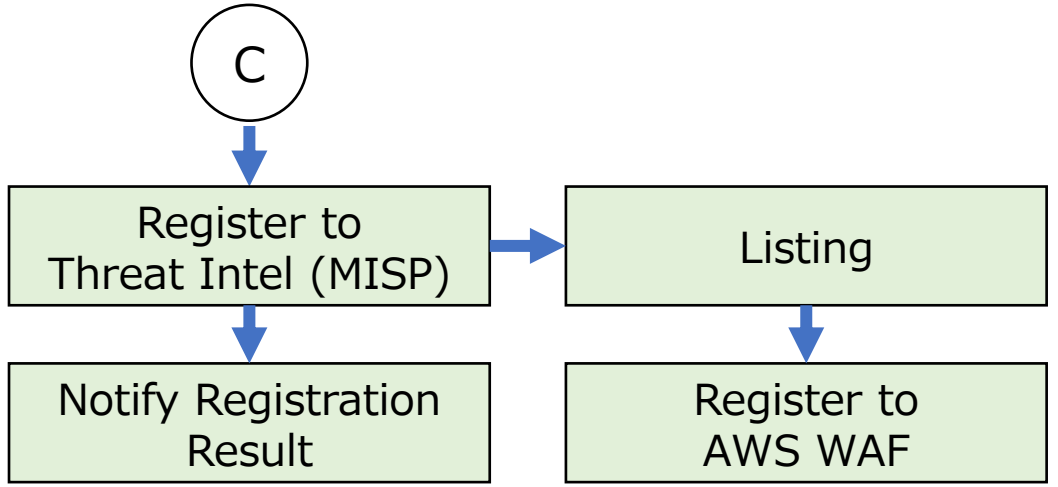


Automated detection and blocking system

Register the IPs to be blocked on the Threat Intelligence platform (MISP)

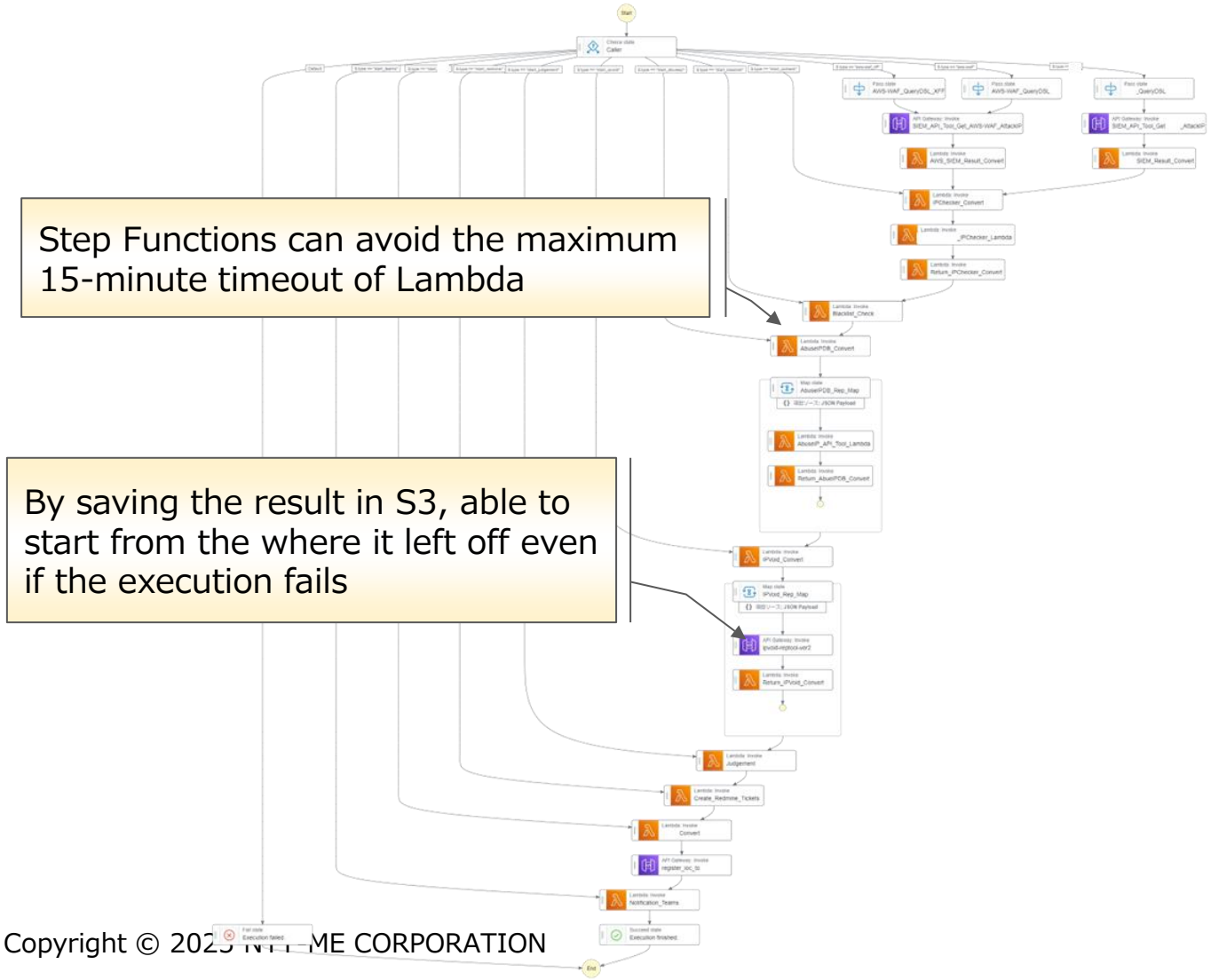
- IPs on MISP are automatically added to AWS WAF's "IPSets" via API
- MISP aggregates malicious IPs identified by SOC analysts, CSIRT, external sources etc. and add them to 'IPSets' for blocking





Step Functions

- **AWS service (Step Functions) automates a series of processes**
  - **A task from analysis to blocking takes just 5 minutes (It used to take 1 hour/day)**



Step Functions can avoid the maximum 15-minute timeout of Lambda

By saving the result in S3, able to start from the where it left off even if the execution fails

SIEM Analysis and Extraction

Evaluation: Avoidance of accidental blocking (confirmation of ALLOW list)

Evaluation: 1st Reputation (Abuse IP DB)

Evaluation: 2nd Reputation (IP Void)

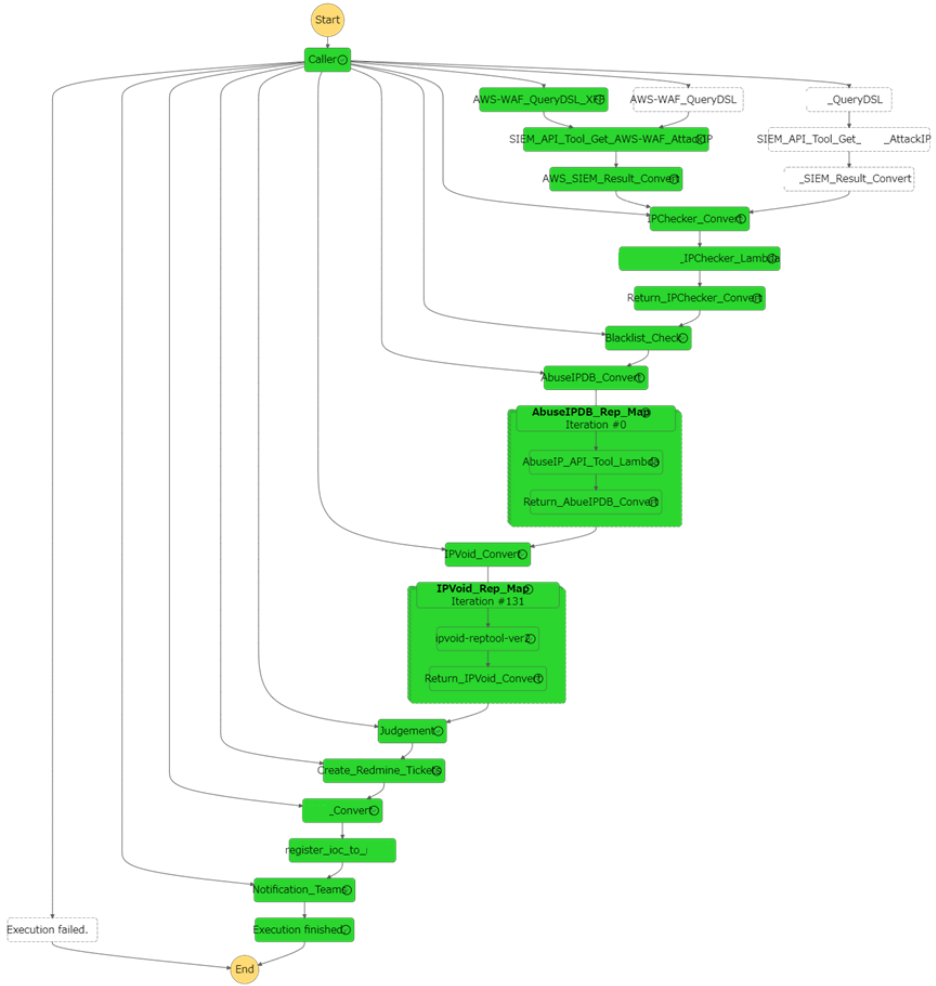
Decision: Blocking Decision

Registration: Registration to the Threat Intel platform (MISP)

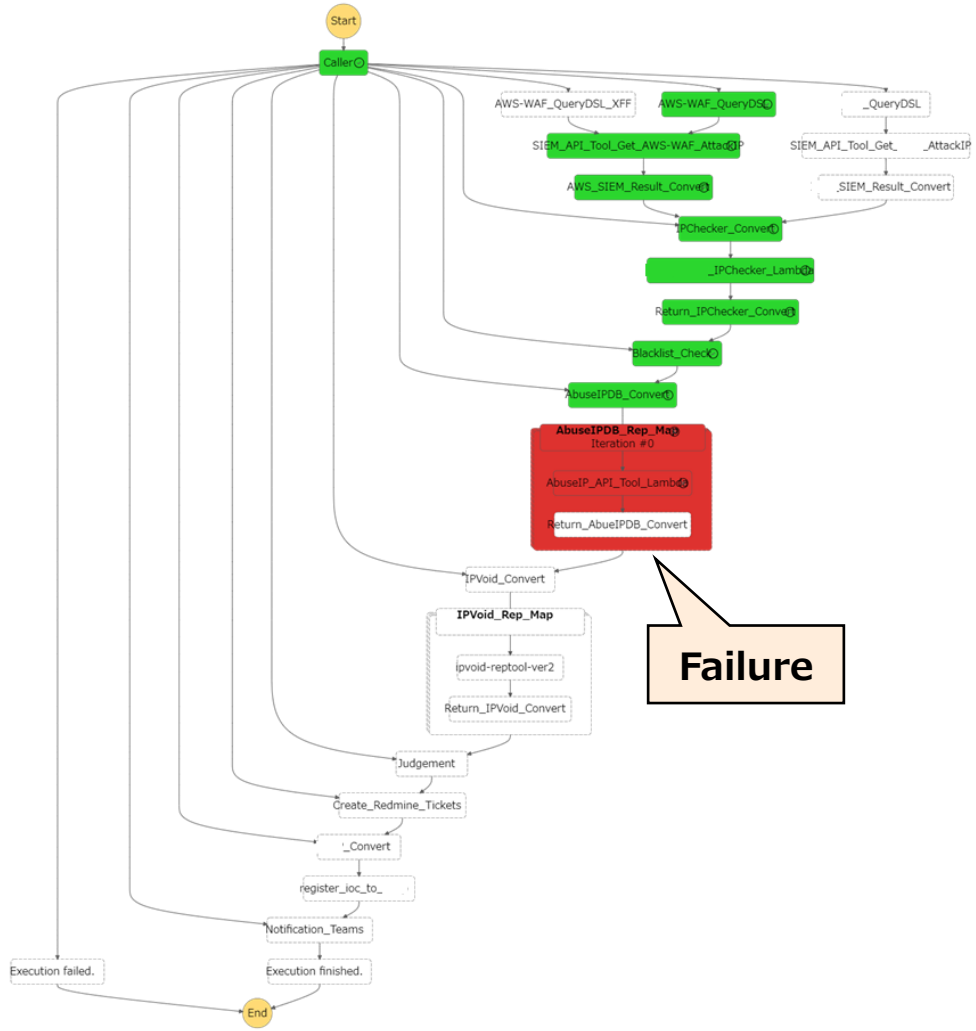
Step Functions

- **Execution success: Green, Execution failure: Red, making it easy to understand**

Execution result : Success

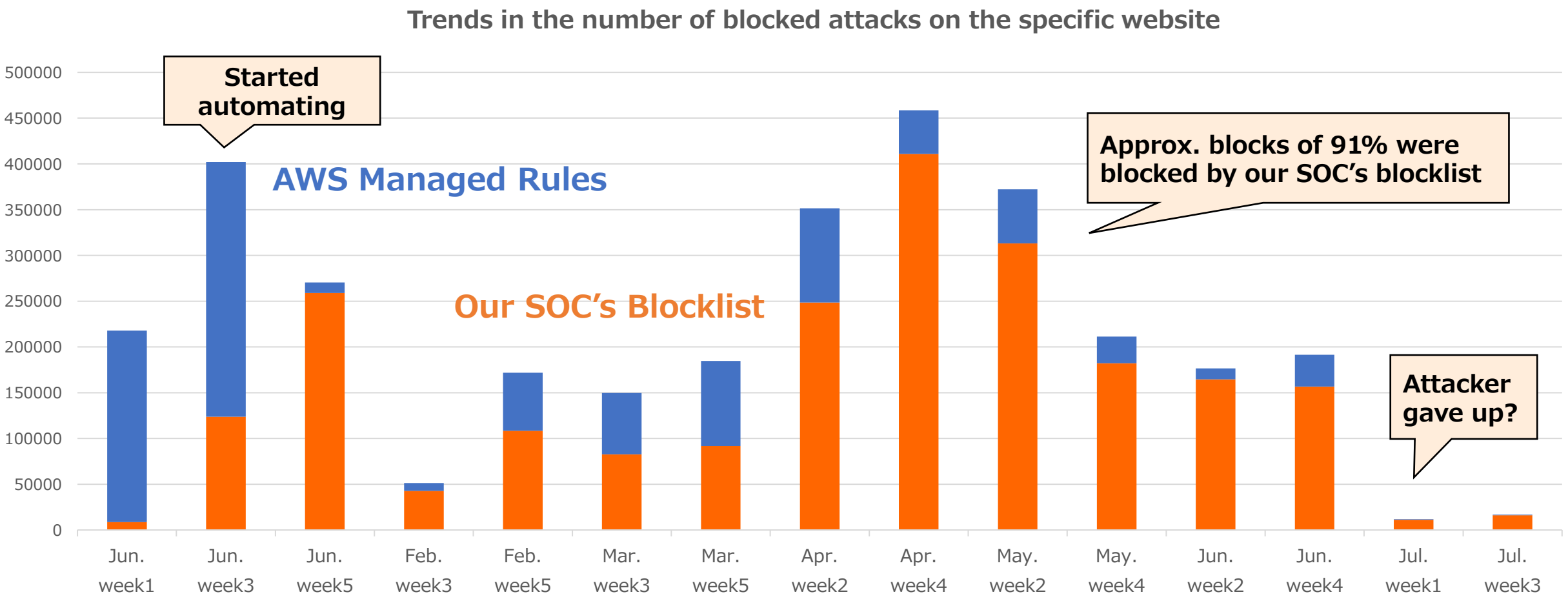


Execution result : Failure



Effectiveness of New System

- **Our SOC's blocklist effectively block and mitigate attacks on a specific website**
  - Responding to attackers persistently target a website by changing their IPs



Conclusion

- **Introduced Fully Automated threat detection and blocking system using Custom Managed Rules and Proprietary Threat Intelligence for AWS WAF**

- **We can customize AWS WAF to build “My WAF “, tailored to our company’s needs**
  - My WAF = AWS WAF + SOC expertise + AWS services (Lambda, Step Functions)

- **With limited resources (people, time), it is essential to maximize the potential of AWS services for ideal security operations**

- **Implementation is not the goal — continuous operation is key**