PRJ-NET-036: Advanced DNS Firewall and Hybrid Query Resolution

Certification: AWS Certified Advanced Networking – Specialty
Domain: DNS and Hybrid Networking

---

1. Project Overview

This project addresses two critical aspects of enterprise networking: DNS security and hybrid DNS resolution. First, we will implement **Route 53 Resolver DNS Firewall**, a managed security service that allows you to filter and control outbound DNS traffic from your VPCs. This provides a crucial layer of defense against malware, ransomware, and data exfiltration by blocking queries to known malicious domains.

Second, we will configure **Route 53 Resolver Endpoints** to create a seamless, hybrid DNS architecture. This enables resources in your VPC to resolve DNS records in your on-premises data center, and conversely, allows on-premises servers to resolve records in AWS private hosted zones (e.g., for services like EC2 instances with private DNS names). This eliminates the need for complex, custom DNS forwarding solutions.

Key Objectives

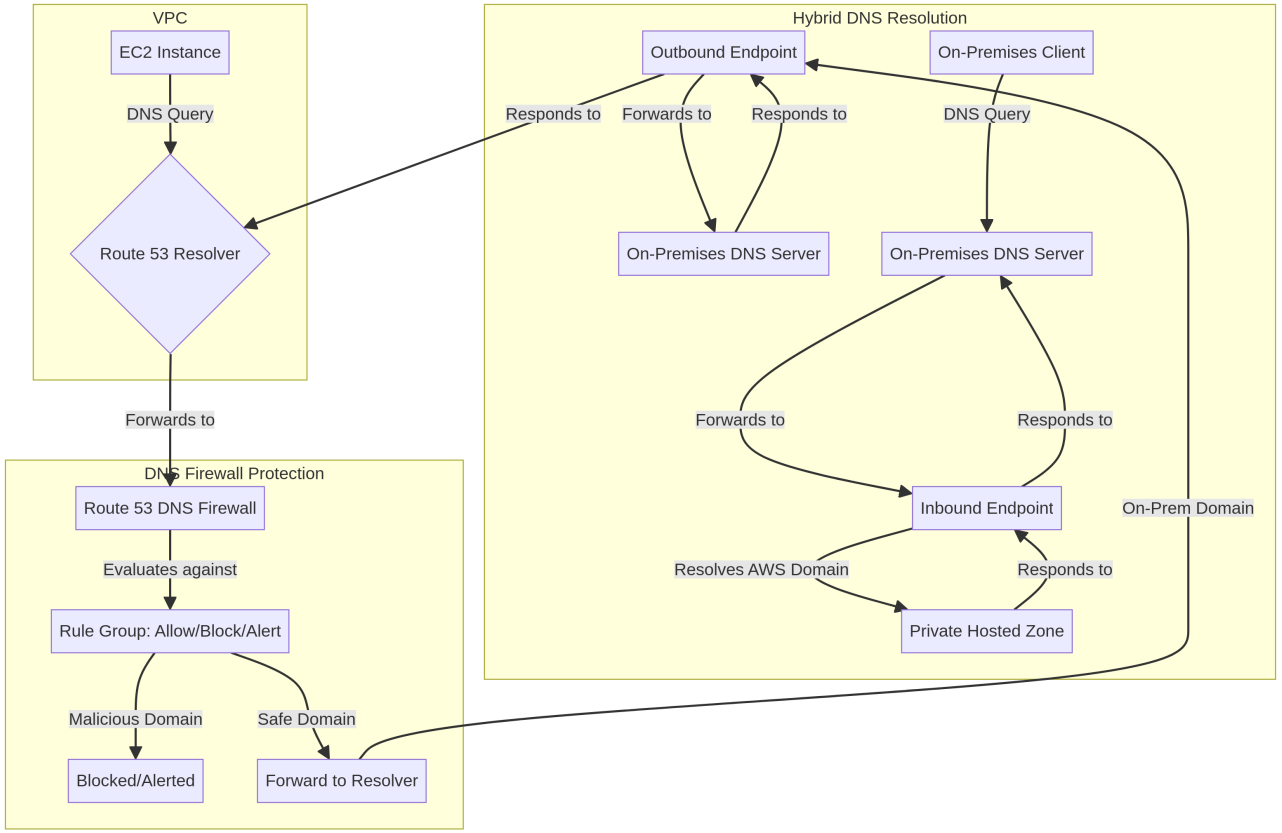
- Deploy Route 53 DNS Firewall to protect a VPC from malicious outbound DNS queries.
- Create and manage domain lists and rule groups for the firewall.
- Monitor firewall activity and alerts using CloudWatch and Route 53 query logs.
- Set up a **Route 53 Outbound Endpoint** to forward queries from a VPC to an on-premises DNS server.
- Set up a **Route 53 Inbound Endpoint** to allow on-premises servers to query private DNS records within AWS.

- Configure conditional forwarding rules to achieve a unified, hybrid DNS resolution.

---

2. Architecture

The architecture combines DNS Firewall for security with Resolver endpoints for hybrid connectivity.



DNS Firewall Flow:

1. **Query Initiated:** An EC2 instance inside a VPC makes a DNS query (e.g., `www.malicious-site.com`).
2. **Firewall Interception:** The query is automatically sent to the Route 53 Resolver, which is now protected by the **DNS Firewall**.
3. **Rule Evaluation:** The firewall evaluates the query against its associated **Rule Group**. The rule group contains rules, each linked to a domain list.
4. **Action Taken:**

- If the queried domain is on a **block list**, the firewall can either return a `NXDOMAIN` (domain does not exist), `NODATA`, or an `OVERRIDE` response (e.g., redirecting to a sinkhole IP). The query never leaves your VPC.
- If the domain is on an **allow list**, it is automatically forwarded.
- If the domain is not on any list, the query is forwarded to the standard Route 53 Resolver.

Hybrid Resolution Flow:

VPC to On-Premises (Outbound):

1. An EC2 instance queries an on-premises hostname (e.g., `server.corp.example.com`).
2. A **Route 53 Resolver Rule** is configured for the `corp.example.com` domain. This rule specifies that queries for this domain should be **forwarded**.
3. The query is sent to the **Outbound Endpoint**, which has elastic network interfaces (ENIs) inside your VPC.
4. The Outbound Endpoint forwards the query over your Direct Connect or VPN to the **on-premises DNS server**.
5. The on-premises server responds, and the answer is passed back to the EC2 instance.

On-Premises to VPC (Inbound):

1. An on-premises server queries an AWS private hostname (e.g., `ip-10-1-1-10.ec2.internal`).
2. The on-premises DNS server is configured with a **conditional forwarder** for the `ec2.internal` zone, pointing to the IP addresses of the **Route 53 Inbound Endpoint**.
3. The query arrives at the Inbound Endpoint within your VPC.
4. The Inbound Endpoint resolves the name against the AWS private DNS and returns the answer to the on-premises server.

---

3. Prerequisites

- An AWS account with administrative permissions.
- A VPC with at least one EC2 instance.
- (Conceptual) An on-premises network with its own DNS server, connected to AWS via Direct Connect or VPN.

---

4. Step-by-Step Guide: DNS Firewall

Step 4.1: Create Domain Lists

1. Go to the **VPC Console -> DNS Firewall -> Domain lists**.
2. **Create `Allowed-Domains` List:**
   - Click **Add domain list**.
   - **Name:** `Allowed-Domains`
   - Add domains you trust, like `amazon.com`, `example.com`.
3. **Create `Blocked-Domains` List:**
   - Create another list named `Blocked-Domains`.
   - Add a domain to test with, like `test-blocked-domain.com`.
   - AWS also provides **managed domain lists** for known malware and botnet C&C servers, which you can use.

Step 4.2: Create a Rule Group and Rules

1. Go to **DNS Firewall -> Rule groups -> Create rule group**.
2. **Name:** `My-VPC-Firewall-Rules`
3. **Add rule:**
   - **Name:** `Allow-Rule`
   - **Domain list:** Select `Allowed-Domains`.
   - **Action:** Allow

4. **Add another rule:**
    - **Name:** `Block-Rule`
    - **Domain list:** Select `Blocked-Domains`.
    - **Action:** Block (choose `NXDOMAIN` as the response).
5. **Rule priority:** Set the priority so the `Allow-Rule` is evaluated before the `Block-Rule`.
6. Create the rule group.

Step 4.3: Associate the Rule Group with a VPC

1. Select the rule group you just created.
2. Go to the **Associated VPCs** tab and click **Associate VPC**.
3. Select your VPC and associate it.

Step 4.4: Test the Firewall

1. SSH into an EC2 instance in the protected VPC.
2. **Test the blocked domain:**

    ```
    dig test-blocked-domain.com
    ```

    The response should have a status of `NXDOMAIN`, indicating the firewall blocked it.
3. **Test an allowed domain:**

    ```
    dig amazon.com
    ```

    This should resolve normally.
4. **Test a standard domain:**

```
dig google.com
```

This should also resolve normally, as it didn't match any rule and was passed to the default resolver.

---

5. Step-by-Step Guide: Hybrid Resolution

Step 5.1: Create Resolver Endpoints

1. Go to **Route 53 -> Resolver -> Endpoints**.
2. **Create Inbound Endpoint:**
   - **Name:** `On-Prem-to-AWS`
   - **VPC:** Select your VPC.
   - **Security Group:** A security group that allows DNS traffic (UDP/TCP 53) from your on-premises network.
   - Specify subnets in at least two AZs. Route 53 will create ENIs in these subnets. Note their IP addresses.
3. **Create Outbound Endpoint:**
   - **Name:** `AWS-to-On-Prem`
   - Follow a similar process, selecting a security group that allows outbound DNS traffic to your on-premises DNS server.

Step 5.2: Configure Resolver Rules (Outbound)

1. Go to **Route 53 -> Resolver -> Rules**.
2. **Create rule:**
   - **Name:** `Forward-to-On-Prem`
   - **Rule type:** Forward
   - **Domain name:** `corp.example.com` (the domain for your on-premises network).

- **VPCs to associate:** Your VPC.
- **Target IP addresses:** Enter the IP addresses of your on-premises DNS servers.
- **Outbound endpoint:** Select the `AWS-to-On-Prem` endpoint you created.

Step 5.3: Configure On-Premises DNS (Inbound)

1. On your on-premises DNS server (e.g., BIND, Windows DNS), configure a **conditional forwarder**.
2. For the domain `amazonaws.com` (or a more specific zone like `us-east-1.compute.internal`), set the forwarder IP addresses to the IPs of the **Inbound Endpoint** ENIs you created in Step 5.1.

Step 5.4: Test Hybrid Resolution

1. **Test Outbound:** From an EC2 instance in your VPC, try to resolve an on-premises hostname:

   ```
   dig server.corp.example.com
   ```

   The query should be forwarded via the outbound endpoint and resolved by your on-premises DNS.
2. **Test Inbound:** From a server on your on-premises network, try to resolve a private AWS hostname:

   ```
   nslookup ip-10-1-1-10.us-east-1.compute.internal
   ```

   The query should be forwarded to the inbound endpoint and resolved by Route 53.

---

6. Cleanup

1. **Disassociate the DNS Firewall rule group** from your VPC and then delete the rule group and domain lists.
2. **Delete the Route 53 Resolver rule.**
3. **Delete the Inbound and Outbound Resolver Endpoints.**
4. Remove the conditional forwarder configuration from your on-premises DNS server.

Business Context

The Problem

Cloud networks are complex and vulnerable to misconfigurations that expose resources to the internet. Organizations lack visibility into network traffic and struggle to implement proper network segmentation. Manual network management leads to security gaps and compliance violations.

The Solution

Secure network architecture with automated traffic inspection, network segmentation, and centralized management. Implements VPC design best practices, transit gateway for multi-VPC connectivity, and network firewall for deep packet inspection. Provides complete network visibility and control.

Business Value

- **Zero Trust Architecture:** Micro-segmentation prevents lateral movement of threats
- **Network Visibility:** Complete traffic logging and analysis for security and troubleshooting
- **Centralized Management:** Single pane of glass for multi-account network control
- **Compliance Ready:** Network logs and flow records for audit requirements

Risk Mitigation

Prevents unauthorized network access, blocks malicious traffic, contains breaches through segmentation, and ensures compliance with network security requirements.

GRC Mapping

Compliance Frameworks

- **NIST CSF:** PR.AC-5 (Network segregation), PR.PT-4 (Network protection), DE.CM-1 (Network monitoring)
- **ISO 27001:** A.13.1 (Network security management), A.13.2 (Information transfer)
- **CIS Controls:** Control 12 (Network Infrastructure Management), Control 13 (Network Monitoring)
- **Zero Trust Architecture:** NIST SP 800-207

Security Controls Implemented

- Network segmentation and micro-segmentation
- Stateful firewall and deep packet inspection
- VPC flow logs and traffic analysis
- Network access control lists (NACLs)
- Private connectivity (VPN/Direct Connect)

Audit Evidence

- VPC flow logs and network traffic records
- Firewall rules and policy configurations
- Network topology diagrams
- Security group and NACL configurations

Regulatory Alignment

- **PCI DSS:** Requirement 1 (Firewall configuration), Requirement 2 (Network segmentation)
- **HIPAA:** § 164.312(e) (Transmission security)
- **GDPR:** Article 32 (Network security measures)
- **SOC 2:** CC6.6 (Logical access - network security)