**🧠 1. Artemis Nedir?**

**Artemis**, AWS ortamında çalışan, IAM politika ve konfigürasyonlarını inceleyerek **potansiyel güvenlik açıklarını ve kötü yapılandırmaları** analiz eden açık kaynaklı bir araçtır.

📌 Temel amacı:

* IAM kullanıcılarının yetkilerini analiz etmek
* Güvenlik açığı olabilecek alanları (örneğin: \* yetkisi, public erişim, privilege escalation) tespit etmek
* Sürekli ve otomatik kontrol sağlamak

**🛠️ 2. Özellikleri**

| **Özellik** | **Açıklama** |
| --- | --- |
| 🔐 IAM Risk Analizi | Yetkilerin gereğinden fazla olup olmadığını kontrol eder |
| 🧠 Privilege Escalation Tespiti | Yetki yükseltme yollarını raporlar |
| 📋 Policy Denetimi | Geniş yetkili policy'leri bulur |
| 📥 S3, EC2, Lambda Analizi | Kaynak seviyesinde zafiyetleri listeler |
| 🧪 Sürekli Güvenlik Kontrolü | Otomatikleştirilebilir taramalar sunar |

**⚙️ 3. Kurulum (Kali / Debian / Linux)**

**✅ 1. Gerekli bileşenleri yükle:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

✅ 2. Artemis’i indir:

metin, ekran görüntüsü, yazı tipi, grafik içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**🔐 4. AWS Bağlantı Ayarları**

Artemis, AWS CLI ile yapılandırılmış **profile** üzerinden çalışır.  
Aşağıdaki gibi .aws/credentials dosyası oluşturulmalı:

[artemis-profile]

aws\_access\_key\_id = YOUR\_KEY

aws\_secret\_access\_key = YOUR\_SECRET

region = us-east-1

Test etmek için:

aws sts get-caller-identity --profile artemis-profile

**🚀 5. Kullanım Komutları**

**✅ Temel tarama:**

python3 artemis.py --profile artemis-profile

**✅ Belirli hizmet için (örnek: IAM):**

python3 artemis.py --profile artemis-profile --service iam

**✅ Tüm riskleri CSV çıktısı olarak almak:**

python3 artemis.py --profile artemis-profile --export risks.csv

**✅ Otomatikleştirilmiş tüm taramaları çalıştırmak:**

python3 artemis.py --profile artemis-profile --full

**📋 6. Örnek Terminal Çıktısı**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**🧠 7. Risk Analizi Özellikleri**

| **Tespit Türü** | **Açıklama** |
| --- | --- |
| 🌐 Public erişim | Herkese açık S3 bucket'lar veya kaynaklar |
| 🔺 Geniş yetkiler | iam:\*, s3:\* gibi wildcard policy'ler |
| 🪜 Privilege escalation | IAM policy yoluyla yetki yükseltme ihtimali |
| 🚫 Kullanılmayan kullanıcılar | Risk oluşturabilecek unutulmuş IAM hesapları |
| 🔑 Credential sızıntısı | Root key veya inline policy açığı tespiti |

**📊 8. Kullanım Senaryoları**

* 🧪 Red Team: Hangi IAM kullanıcısı ile privilege escalation yapılabilir?
* 🛡️ Blue Team: IAM policy’leri güvenli mi?
* 📋 Uyumluluk Testi: AWS hesabı PCI-DSS / SOC2 için güvenli mi?
* 🔍 Denetim: Hangi kaynaklar tüm internete açık?

**🎯 9. Öğrenim Kazanımları**

* IAM güvenliğini değerlendirme becerisi kazanma
* AWS politikalarının analizini öğrenme
* Red/Blue Team bakış açısıyla risk tespiti yapma
* Güvenli bulut yapılandırması hakkında farkındalık edinme
* Python tabanlı güvenlik analiz araçlarını kullanma