MITRE CALDERA Windows 10 x64 Komple Kurulum Kılavuzu

İçindekiler

Sistem Gereksinimleri

Minimum Gereksinimler:

- **İşletim Sistemi**: Windows 10 x64 (Build 1809 veya üzeri)
- **RAM**: 8 GB (16 GB önerilir)
- **Disk Alanı**: 20 GB boş alan
- **İşlemci**: 4 çekirdek (8 çekirdek önerilir)
- **Python**: 3.8 - 3.11 arası (3.12+ henüz tam desteklenmiyor)
- **İnternet Bağlantısı**: Gerekli

Gerekli Yazılımlar:

- Python 3.11.x
- Anaconda/Miniconda
- Git
- Visual Studio Build Tools 2022
- Node.js (v18.x veya üzeri)
- PostgreSQL 15 (opsiyonel)

Ön Hazırlık

1. Windows PowerShell Yönetici Yetkisi

PowerShell'i yönetici olarak açın:

```powershell
# ExecutionPolicy ayarlama
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser -Force
```

2. Windows Defender İstisnası (Opsiyonel)

CALDERA'nın agent'ları antivirus tarafından engellenebilir:

```powershell
# CALDERA dizini için istisna ekleme
Add-MpPreference -ExclusionPath "C:\caldera"
```

Python ve Conda Kurulumu

1. Miniconda İndirme ve Kurulum

```powershell
# Miniconda installer'ı indir
Invoke-WebRequest -Uri "https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe" -OutFile "$e

# Sessiz kurulum
Start-Process -FilePath "$env:TEMP\miniconda.exe" -ArgumentList "/S", "/D=C:\Miniconda3" -Wait

# PATH'e ekleme
$env:Path += ";C:\Miniconda3;C:\Miniconda3\Scripts;C:\Miniconda3\Library\bin"
[Environment]::SetEnvironmentVariable("Path", $env:Path, [EnvironmentVariableTarget]::User)
```

2. Conda Yapılandırması

```powershell
# Conda'yı başlat
& "C:\Miniconda3\Scripts\conda.exe" init powershell

# PowerShell'i yeniden başlat ve devam et
```

3. Python 3.11 Ortamı Oluşturma

```powershell
# CALDERA için özel conda ortamı
conda create -n caldera python=3.11 -y
conda activate caldera

# Temel paketleri yükle
conda install -c conda-forge pip setuptools wheel -y
```

Visual Studio Build Tools Kurulumu

1. Build Tools İndirme

```powershell
# VS Build Tools 2022 indir
Invoke-WebRequest -Uri "https://aka.ms/vs/17/release/vs_buildtools.exe" -OutFile "$env:TEMP\vs_buildtools.exe"
```

2. Gerekli Bileşenleri Kurma

```powershell
# C++ build tools ve Windows SDK kurulumu
Start-Process -FilePath "$env:TEMP\vs_buildtools.exe" -ArgumentList `
    "--quiet", "--wait", "--norestart", "--nocache", `
    "--add", "Microsoft.VisualStudio.Workload.VCTools", `
    "--add", "Microsoft.VisualStudio.Component.Windows10SDK.19041", `
    "--add", "Microsoft.VisualStudio.Component.VC.Tools.x86.x64", `
    "--add", "Microsoft.VisualStudio.Component.VC.CMake.Project" -Wait
```

3. Ortam Değişkenlerini Ayarlama

```powershell
# VS ortam değişkenleri
$vsPath = "${env:ProgramFiles(x86)}\Microsoft Visual Studio\2022\BuildTools"
$env:Path += ";$vsPath\VC\Auxiliary\Build"
[Environment]::SetEnvironmentVariable("Path", $env:Path, [EnvironmentVariableTarget]::User)
```

Git Kurulumu

```powershell
```

```powershell
# Git for Windows indir ve kur
Invoke-WebRequest -Uri "https://github.com/git-for-windows/git/releases/download/v2.43.0.windows.1/Git-2.43.0-64-

# Sessiz kurulum
Start-Process -FilePath "$env:TEMP\git-installer.exe" -ArgumentList "/VERYSILENT", "/NORESTART" -Wait

# PATH'e ekleme
$env:Path += ";C:\Program Files\Git\bin"
[Environment]::SetEnvironmentVariable("Path", $env:Path, [EnvironmentVariableTarget]::User)
```

Node.js ve npm Kurulumu

```powershell
# Node.js LTS indir
Invoke-WebRequest -Uri "https://nodejs.org/dist/v20.11.0/node-v20.11.0-x64.msi" -OutFile "$env:TEMP\node-installer.

# MSI kurulumu
Start-Process msiexec.exe -ArgumentList "/i", "$env:TEMP\node-installer.msi", "/quiet", "/norestart" -Wait

# npm güncelleme
npm install -g npm@latest
```

PostgreSQL Kurulumu (Opsiyonel)

CALDERA varsayılan olarak SQLite kullanır, ancak production için PostgreSQL önerilir:

```powershell
# PostgreSQL 15 indir
Invoke-WebRequest -Uri "https://get.enterprisedb.com/postgresql/postgresql-15.5-1-windows-x64.exe" -OutFile "$env

# Kurulum (şifre: caldera123)
Start-Process -FilePath "$env:TEMP\postgresql-installer.exe" -ArgumentList `
    "--mode", "unattended", `
    "--unattendedmodeui", "none", `
    "--prefix", "C:\PostgreSQL\15", `
    "--serverport", "5432", `
    "--superpassword", "caldera123", `
    "--servicename", "postgresql-15" -Wait
```

CALDERA Kurulumu

1. CALDERA Repository'sini Klonlama

```powershell
# Çalışma dizini oluştur
New-Item -ItemType Directory -Force -Path C:\caldera
Set-Location C:\caldera

# CALDERA'yı klonla
git clone https://github.com/mitre/caldera.git .
git checkout master  # veya belirli bir release: git checkout 4.2.0
```

2. Python Bağımlılıklarını Kurma

```powershell
# Conda ortamını aktif et
conda activate caldera

# pip'i güncelle
python -m pip install --upgrade pip

# Gerekli build araçlarını kur
pip install --upgrade setuptools wheel cython

# CALDERA gereksinimlerini kur
pip install -r requirements.txt

# Ek güvenlik ve performans paketleri
pip install cryptography pycryptodome psutil
```

3. Frontend Bağımlılıklarını Kurma

```powershell
# Plugin dizinine git
Set-Location C:\caldera\plugins\magma

# npm bağımlılıklarını kur
npm install

# Frontend'i derle
npm run build
```

4. Varsayılan Yapılandırma Dosyası

```powershell
# default.yml dosyasını oluştur
@"
# CALDERA Yapılandırma Dosyası
host: 0.0.0.0
port: 8888
memory: True
log_level: INFO

users:
  red:
    red: admin
  blue:
    blue: admin
  admin:
    admin: admin

plugins:
  - access
  - atomic
  - builder
  - compass
  - debrief
  - fieldmanual
  - gameboard
  - human
  - manx
  - mappreview
  - response
  - sandcat
  - stockpile
  - training

# Veritabanı ayarları (PostgreSQL kullanmak için)
# database:
#   type: postgres
#   host: localhost
#   port: 5432
#   database: caldera
#   username: caldera_user
#   password: caldera123
"@ | Set-Content -Path C:\caldera\conf\default.yml
```

CALDERA Yapılandırması

1. İleri Düzey Yapılandırma

```powershell
# local.yml dosyası oluştur (production için)
@"
# Production Yapılandırması
host: 0.0.0.0
port: 8888
memory: False
log_level: WARNING

# SSL/TLS Yapılandırması
ssl:
  enabled: True
  keyfile: conf/ssl/server.key
  certfile: conf/ssl/server.crt

# Güvenlik ayarları
api_key_blue: $(New-Guid)
api_key_red: $(New-Guid)

# Rate limiting
rate_limit:
  enabled: True
  max_requests: 100
  window_seconds: 60

# LDAP entegrasyonu (opsiyonel)
ldap:
  enabled: False
  server: ldap://dc.domain.local
  base_dn: DC=domain,DC=local
"@ | Set-Content -Path C:\caldera\conf\local.yml
```

2. SSL Sertifikası Oluşturma

```powershell
```

```
# OpenSSL ile self-signed sertifika
New-Item -ItemType Directory -Force -Path C:\caldera\conf\ssl

# OpenSSL yoksa Chocolatey ile kur
choco install openssl -y

# Sertifika oluştur
openssl req -x509 -newkey rsa:4096 -keyout C:\caldera\conf\ssl\server.key -out C:\caldera\conf\ssl\server.crt -days 365
```

Plugin Kurulumları

1. Ek Plugin'ler

powershell

```
Set-Location C:\caldera\plugins

# Arsenal plugin (ek yetenekler)
git clone https://github.com/mitre-attack/arsenal.git

# Emu plugin (adversary emulation)
git clone https://github.com/mitre/emu.git

# Her plugin için bağımlılıkları kur
Get-ChildItem -Directory | ForEach-Object {
    if (Test-Path "$($_.FullName)\requirements.txt") {
        Write-Host "Installing requirements for $($_.Name)"
        pip install -r "$($_.FullName)\requirements.txt"
    }
}
```

2. Özel Ability'ler ve Adversary'ler

powershell

```powershell
# Özel ability dizini
New-Item -ItemType Directory -Force -Path C:\caldera\data\abilities\custom

# Örnek ability
@"
- id: 8c6b1e3a-2c4d-4e6f-8a0b-1c3e5f7a9d2b
  name: Windows Defender Status Check
  description: Check Windows Defender status
  tactic: discovery
  technique:
    attack_id: T1518.001
    name: Security Software Discovery
  platforms:
    windows:
      psh:
        command: |
          Get-MpComputerStatus | Select-Object AntivirusEnabled,RealTimeProtectionEnabled,BehaviorMonitorEnabled | C
"@ | Set-Content -Path C:\caldera\data\abilities\custom\defender-check.yml
```

Servis Olarak Çalıştırma

1. Windows Service Wrapper (NSSM)

powershell

```powershell
# NSSM indir
Invoke-WebRequest -Uri "https://nssm.cc/release/nssm-2.24.zip" -OutFile "$env:TEMP\nssm.zip"
Expand-Archive -Path "$env:TEMP\nssm.zip" -DestinationPath "$env:TEMP\nssm"
Copy-Item "$env:TEMP\nssm\nssm-2.24\win64\nssm.exe" -Destination "C:\Windows\System32"

# CALDERA servisini oluştur
nssm install CALDERA "C:\Miniconda3\envs\caldera\python.exe" "C:\caldera\server.py"
nssm set CALDERA AppDirectory "C:\caldera"
nssm set CALDERA DisplayName "MITRE CALDERA"
nssm set CALDERA Description "Automated Adversary Emulation Platform"
nssm set CALDERA Start SERVICE_AUTO_START

# Servisi başlat
Start-Service CALDERA
```

2. Başlangıç Script'i

powershell

```powershell
# start_caldera.ps1
@"
# CALDERA Başlatma Script'i
Set-Location C:\caldera
& C:\Miniconda3\envs\caldera\python.exe server.py --log DEBUG
"@ | Set-Content -Path C:\caldera\start_caldera.ps1
```

Sorun Giderme

1. Port Kontrolleri

powershell

```powershell
# 8888 portunu kontrol et
netstat -an | findstr :8888

# Windows Firewall kuralı ekle
New-NetFirewallRule -DisplayName "CALDERA" -Direction Inbound -LocalPort 8888 -Protocol TCP -Action Allow
```

2. Log Kontrolü

powershell

```powershell
# CALDERA loglarını görüntüle
Get-Content C:\caldera\logs\caldera.log -Tail 50 -Wait
```

3. Bağımlılık Kontrolleri

powershell

```powershell
# Python paketlerini kontrol et
pip list | Select-String -Pattern "aiohttp|cryptography|pyyaml"

# Eksik paketleri tekrar kur
pip install --force-reinstall -r requirements.txt
```

4. Veritabanı Bağlantı Testi (PostgreSQL)

powershell

```powershell
# psql ile test
& "C:\PostgreSQL\15\bin\psql.exe" -U postgres -c "CREATE DATABASE caldera;"
& "C:\PostgreSQL\15\bin\psql.exe" -U postgres -c "CREATE USER caldera_user WITH PASSWORD 'caldera123';"
& "C:\PostgreSQL\15\bin\psql.exe" -U postgres -c "GRANT ALL PRIVILEGES ON DATABASE caldera TO caldera_user;"
```

Erişim ve İlk Kullanım

1. **Web Arayüzü**: https://localhost:8888
2. **Varsayılan Kullanıcılar**:
   - Red Team: `red` / `admin`
   - Blue Team: `blue` / `admin`
   - Admin: `admin` / `admin`
3. **API Erişimi**:

```powershell
# API ile operation başlatma örneği
$headers = @{
    "KEY" = "your-api-key"
    "Content-Type" = "application/json"
}

$body = @{
    "name" = "Test Operation"
    "adversary" = @{
        "adversary_id" = "1234"
    }
    "planner" = "atomic"
} | ConvertTo-Json

Invoke-RestMethod -Uri "https://localhost:8888/api/v2/operations" -Method POST -Headers $headers -Body $body -S
```

Güvenlik Önerileri

1. **Varsayılan Şifreleri Değiştirin**
2. **SSL/TLS Kullanın**
3. **API Key'leri Güçlü Tutun**
4. **Network Segmentasyonu Uygulayın**
5. **Düzenli Yedekleme Yapın**
6. **Audit Loglarını Aktif Edin**

Faydalı Komutlar

```powershell
```

```
# CALDERA durumunu kontrol et
Get-Service CALDERA

# Agent'ları listele
curl -H "KEY: your-api-key" https://localhost:8888/api/v2/agents

# Operation'ları listele
curl -H "KEY: your-api-key" https://localhost:8888/api/v2/operations

# Yeni adversary yükle
curl -X POST -H "KEY: your-api-key" -F "file=@adversary.yml" https://localhost:8888/api/v2/adversaries
```

Kaynaklar

- CALDERA Resmi Dokümantasyon
- CALDERA GitHub
- MITRE ATT&CK
- CALDERA Plugin Geliştirme

Not: Bu kurulum kılavuzu test ortamları için hazırlanmıştır. Production ortamlarında ek güvenlik önlemleri alınmalıdır.