missing-user-entrypoint

Open

sims-nosql-api/Dockerfile:29

Analyzed | ⋮ | Ignore... | Fix

Assistant thinks this is a true positive

Container runs .NET application as root, allowing a compromised application to give attackers full system access to the container.

Agree or Disagree

Finding description | Rule description

The `ENTRYPOINT` instruction runs the .NET application without first specifying a non-root `USER` in the Dockerfile. This means the container will execute the `dotnet sims-nosql-api.dll` process as **root** by default.

Exploit scenario:

1. An attacker discovers a vulnerability in the .NET application running in the container (e.g., a remote code execution bug in a web endpoint exposed on port 8080).
2. The attacker sends a malicious request to trigger the RCE and gains shell access to the container as the root user.
3. With root access, the attacker can now:
   - Read all files in the container without restrictions
   - Modify or delete application files and configuration
   - Access secrets or environment variables stored in the container
   - Break out of the container to attack the host system or other containers

Since there is no `USER` directive before the `ENTRYPOINT`, the dotnet process inherits root privileges, giving the attacker maximum control if they compromise the application.

Your code | Example code

sims-nosql-api/Dockerfile:29 build process

```
10  WORKDIR /src
11
12  # 🔹 KORRIGIERTER COPY-BEFEHL
13  COPY ["sims-nosql-api.csproj", "./"]
14  RUN dotnet restore "./sims-nosql-api.csproj"
15
16  # 🔹 Quellcode kopieren und Build durchführen
17  COPY . .
18  RUN dotnet build "./sims-nosql-api.csproj" -c $BUILD_CONFIGURATION -o /app/build
19
20  # 🔹 Veröffentlichen (Publish)
21  FROM build AS publish
22  ARG BUILD_CONFIGURATION=Release
23  RUN dotnet publish "./sims-nosql-api.csproj" -c $BUILD_CONFIGURATION -o /app/publish /p:UseAppHost=false
24
25  # 🔹 Finale Laufzeitstufe
26  FROM base AS final
27  WORKDIR /app
28  COPY --from=publish /app/publish .
29  ENTRYPOINT ["dotnet", "sims-nosql-api.dll"]
30
```

Assistant suggested fix | Rule fix

```
ENTRYPOINT ["dotnet", "sims-nosql-api.dll"]
```

```
# 🔹 Finale Laufzeitstufe
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .

# Create a non-root user named 'appuser' with no home and
no password
RUN adduser --disabled-password --no-create-home appuser

# Switch to the non-root user
USER appuser

ENTRYPOINT ["dotnet", "sims-nosql-api.dll"]
```

1. Create a non-root user in the final stage of your Dockerfile with a command like `RUN adduser --disabled-password --no-create-home appuser`.
2. Switch to that user with `USER appuser` just before the `ENTRYPOINT` instruction.
3. The lines should look like:
   - `RUN adduser --disabled-password --no-create-home appuser`
   - `USER appuser`
   - `ENTRYPOINT ["dotnet", "sims-nosql-api.dll"]` Alternatively, if your image is based on Alpine Linux, use `adduser -D -H appuser` to add the user. Running your application as a non-root user reduces the risk of a compromised app impacting the container or host system.

Customize fix

Activity

New note

Assistant thinks this file is low risk because it relates to build process

17 minutes ago by Semgrep Assistant

Assistant thinks this is a true positive

17 minutes ago by Semgrep Assistant

Assistant analysis started

RULE DETAILS

- High severity
- Medium confidence
- Monitor
- CWE-269
- missing-user-entrypoint

FINDING DETAILS

- 3 days ago
- is241307/simsfh_ws25_SAST
  managed-scan
- main
- 5cef085
  by puywei