

Øvelse 30 – Grundlæggende taint-analyse med Snyk

Information

Formålet med denne øvelse er at introducere grundlæggende taint-analyse med værktøjet [Snyk](#).

I opgaven skal du udføre analysen på en ASP.NET Core-applikation. Repositoriet, du skal klonе, er her: <https://github.com/brunobritodev/ASP.NET-Core-SQL-Injection-Example>.

Som navnet på applikationen indikerer, indeholder applikationen en sårbarhed, der gør den modtagelig over for injection-angrebet *SQL Injection*.

Sårbarheden, som vi skal arbejde med, er implementeret i Controller-klassen `HomeController` i metoden `SearchStudentUnsecure`.

Sårbarheden er meget åbenlys, hvilket gør den velegnet til at lære om taint-analyser.

Taint-analysen udføres ved, at en algoritme følger en inputværdi fra kilden (source) til dens endelige mål (sink).

I applikationen er source browserens input til applikationen, som ses i metoden:

```
public async Task<IActionResult> SearchStudentUnsecure(string name)
```

Source må altså være argumentet `name`. Sink i applikationen er databasen, hvilket også kan ses i koden for metoden `SearchStudentUnsecure`:

```
var conn = _context.Database.GetDbConnection();
var query = "SELECT FirstName, LastName FROM Student WHERE FirstName LIKE '%" +
    name + "%'";
IEnumerable<Student> students;

try
{
    await conn.OpenAsync();
    students = await conn.QueryAsync<Student>(query);
}
```

Kodelinjen `await conn.QueryAsync<Student>(query)` laver et kald til databasen med en SQL-string, som indeholder ikke-valideret data. Det vil sige, at *tainted* data nu får lov til at bevæge sig direkte fra source til sink.

I opgaven skal I efterprøve dette ved at anvende værktøjet Snyk CLI som taint-analyseværktøj.

Værktøjet er det samme som anvendt i [Opgave 29](#).

Du skal dog først oprette en konto og aktivere `Code scan`-funktionen. Følg guiden [her](#).

Derefter skal du blot følge trinene i instruktionerne.

▮ **Vigtigt:** Skanning for sårbarheder i tredjepartsbiblioteker (f.eks. NuGet-pakker) er **ikke** det samme som statisk kodeanalyse.

Tredjepartsskanning handler om at finde kendte sikkerhedsproblemer i eksterne komponenter, mens statisk kodeanalyse (som taint-analyse) analyserer selve koden i jeres projekt.

Mange værktøjer som Snyk kan udføre begge dele, så det er vigtigt at kende forskellen, da studerende ofte forveksler 3. parts skanninger med statisk kode analyse.

▮ Instruktioner

1. ▮ Klon projektet fra [dette repository](#).
2. ▮ I den klonede projekts mappe, skift til mappen `SqlInjection`.
3. ▮ Udfør taint-analysen ved at følge instruktionerne fra afsnittet `Testing a repository from its root folder` i denne [dokumentation](#).
4. ▮ Notér, hvilket linjenummer SQL-injection sårbarheden bliver fundet på.
5. ▮ Verificér i koden, at linjenummeret for sårbarheden er der, hvor den forurenede data sendes til sink.

▮ Refleksionsspørgsmål

1. ▮ Hvordan hjælper taint-analyse med at identificere sikkerhedsrisici i software?
2. ▮ Hvilke begrænsninger har automatiserede statiske analyseværktøjer som Snyk i forhold til kompleks kode og kontekst(domæne viden)?
3. ▮ Hvordan kan viden om kildedata (source) og destinationssted (sink) i koden forbedre sikkerhedsarbejdet?
4. ▮ Hvordan bør fundne sårbarheder håndteres i udviklings- og driftsprocessen for at øge sikkerheden?

▮ Kobling til relevante standarder og principper

Element i øvelsen	Relateret standard / princip	Kommentar
Taint-analyse og statisk kodeanalyse	ISO/IEC 27002:2022, afsnit 8.29 (Security testing)	Anbefaler brug af sikkerhedstest og sårbarheds analyse som en del af sikkerhedstest.
	OWASP ASVS 5.0.0, Kapitlet "Assessment and Certification"	Afsnittene "Verification Mechanisms" og "The Role of Automated Security Testing Tools" omtaler anvendelsen hensigt og uhensigtsmæssig brug af automatiseret værktøjer
	Secure by Design	Fremhæver værdien af analyseværktøjer, og viser svagheder ift. manglende invarians ved objekters instantiering

▮ *Taint-analyse er et effektivt værktøj til at identificere potentielle sikkerhedsbrud ved at spore hvordan data bevæger sig gennem applikationen fra usikre kilder til kritiske operationer.*

Last update: 2025-10-30 12:07:19