

REST API Security Testing within the IEC 62443-4-1 Standard

- Identifying *security requirements* specific for REST APIs
- Developing a *methodology* for automated security testing that:
 - has **high coverage** of IEC 62443-4-1
 - targets **accuracy** of security testing results
- Creating a link between IEC 62443-4-1 and automated tests

Introduction

- **API security** = information security + network security + application security
- **REST APIs** are based on 6 principles:
 - uniform interface, stateless, cacheable, client-server, layered, code on demand
- **IEC 62443-4-1** defines secure development lifecycle requirements
 - Practice 5 of IEC 62443-4-1 defines 4 *security testing types*: security requirements, threat mitigation, vulnerability, and penetration testing

The problem

- Connection between standard requirements and REST API security tests
- Security testing automation with high IEC 62443-4-1 coverage

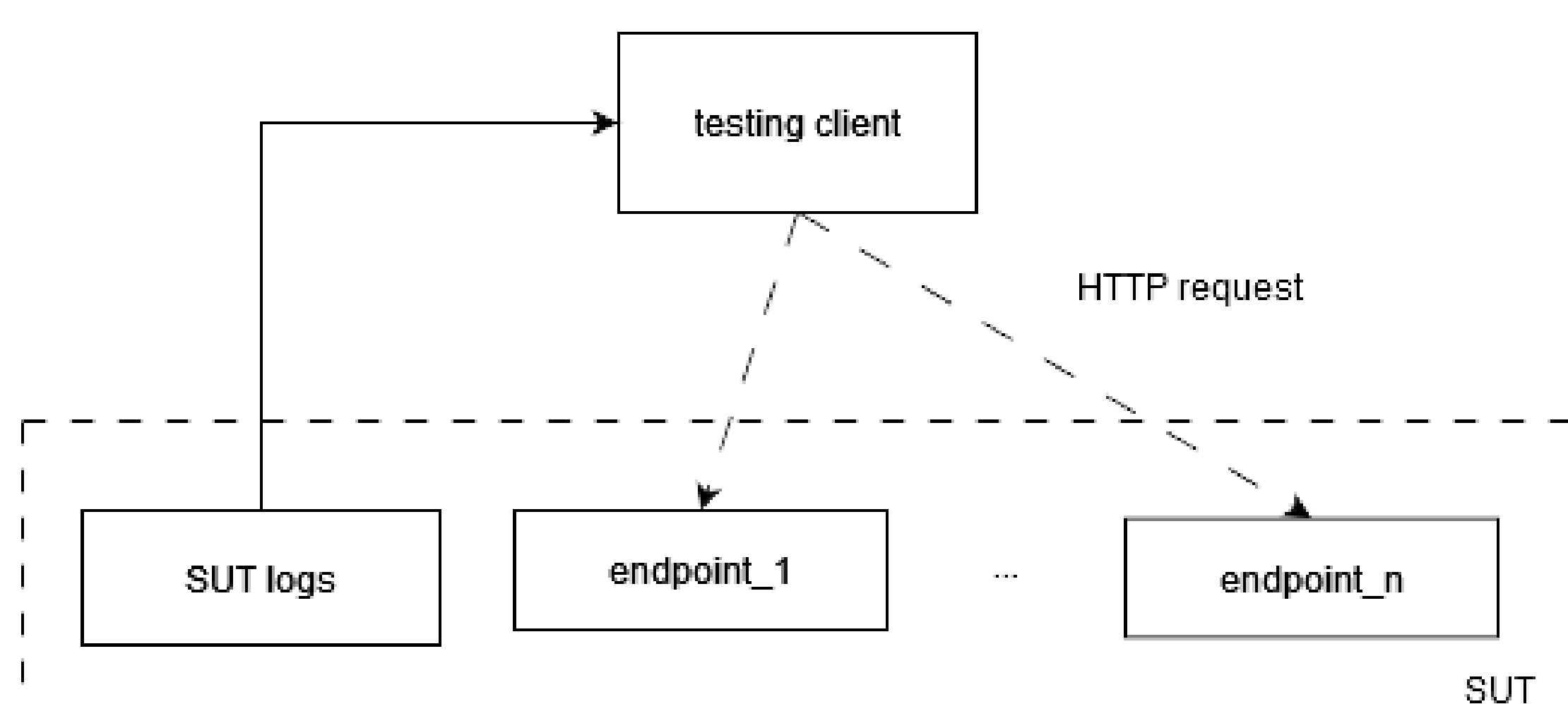


Figure 1: A testing setup. Testing client can send *HTTP requests* to the system under test (SUT) and can retrieve *logs* from it via available connection.

Security testing methodology

- Preparing (manual):
 - **defining security requirements**
 - **defining threat model**
- Testing (automated):
 - executing tools as separate testcases: Schemathesis, ZAP, RESTler
 - validating tool results using retrieved logs
- Analysing IEC 62443-4-1 coverage based on conducted tests

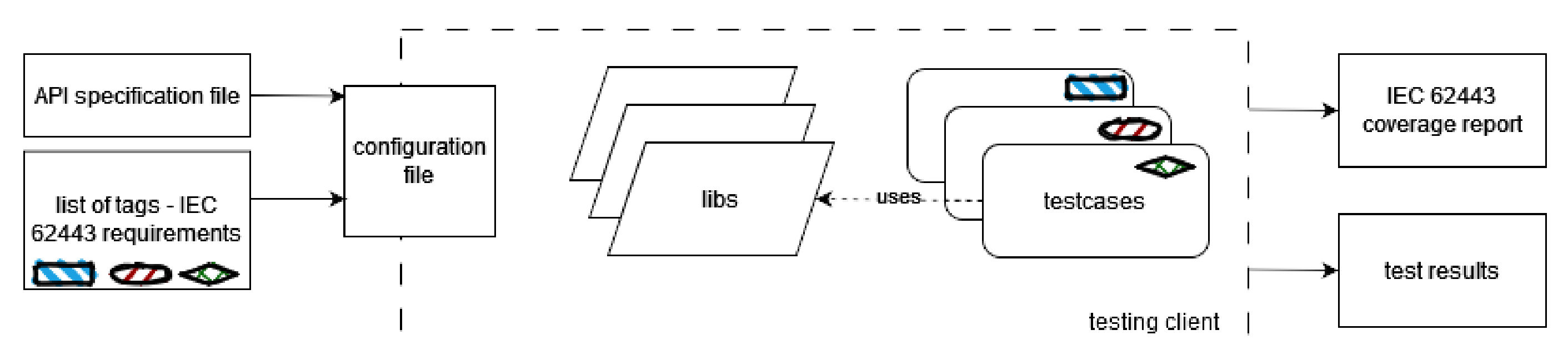


Figure 2: Testing client design. Testing client is a **configurable and extensible** testcases collection that runs chosen tools for security testing.

Implementation

- Each tool is executed within a *pytest* test
- Log retrieval is done via *SSH* or *Serial* connection
- Each test is *tagged* with corresponding IEC 62443-4-1 requirement