**Checking Security Properties of Cloud Service REST APIs**

Most modern cloud and web services are programmatically accessed through REST APIs. This paper discusses how an attacker might compromise a service by exploiting vulnerabilities in its REST API. We introduce four security rules that capture desirable properties of REST APIs and services. We then show how a stateful REST API fuzzer can be extended with active property checkers that automatically test and detect violations of these rules. We discuss how to implement such checkers in a modular and efficient way. Using these checkers, we found new bugs in several deployed production Azure and Office365 cloud services, and we discussed their security implications. All these bugs have been fixed.

**EXISTING SYSTEM:**

Scanning of Swagger-based Representational State Transfer (REST) APIs - In addition to scanning Simple Object Access Protocol (SOAP) web services, Qualys WAS leverages the Swagger specification for testing REST APIs. Users need to only ensure the Swagger version 2.0 file (JSON format) is visible to the scanning service, and the APIs will automatically be tested for common application security flaws. - Enhanced API Scanning with Postman Support - Postman is a widely-used tool for functional testing of REST APIs. A Postman Collection is a file that can be exported from the tool that clubs together related requests (API endpoints) and shares them with other users. These collections are exported in JSON format. With the release of Postman Collection support in Qualys WAS, customers have the option to configure their API scans using the Postman Collection for their API.

**DISADVANTAGES OF EXISTING SYSTEM:**

* SOAP APIs are largely based and use only HTTP and XML.
* On other hand Soap API requires more resources and bandwidth as it needs to convert the data in XML which increases its payload and results in the large sized file.
* On other hand SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern.

**PROPOSED SYSTEM:**

REST APIs are implemented on top of the ubiquitous HTTP/S protocol, and offer a uniform way to create (PUT/POST), monitor (GET), manage (PUT/POST/PATCH) and delete (DELETE) cloud resources. Cloud service developers can document their REST APIs and generate sample client code by describing their APIs using an interface-description language such as Swagger (recently renamed OpenAPI) [25]. A Swagger specification describes how to access a cloud service through its REST API, including what requests the service can handle, what responses may be received, and the response format.

**ADVANTAGES OF PROPOSED SYSTEM:**

* REST APIs are usually simple to build and adapt.
* With the initial URI, the client does not require routing information.
* Tools for automatically testing cloud services via their REST APIs and checking whether these services are reliable and secure are still in their infancy. Some tools available for testing REST APIs capture live API traffic, and then parse, fuzz, and replay the traffic with the hope of finding bugs.
* Clients can have a generic ‘listener’ interface for notifications.
* The approach is implemented as a semi-automatic code generation tool in Django, a Python web framework.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 500 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Coding Language : Python
* Tool : PyCharm, Visual Studio Code
* Database : MYSQL

**REFERENCE:**

Irum Rauf Abo Akademi University,Turku, Finland, Elena Troubitsyna KTH – Royal Institute of Technology, Stockholm, Sweden, “**Checking Security Properties of Cloud Service REST APIs**”, Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) IEEE Xplore: 23 July 2018, DOI: 10.1109/DSN.2018.00060