



Security

# REST API Design, Development & Management

<http://www.acloudfan.com>

[raj@acloudfan.com](mailto:raj@acloudfan.com)

Summary of a course that covers the A to Z of RESTful API. More information.

*Updated: Dec 27<sup>th</sup>, 2016*

# Discounted access to the courses:

raj@acloudfan.com  
<http://www.acloudfan.com>



<https://www.udemy.com/ibm-bluemix/?couponCode=BLUE100>

Coupon Code = **BLUE100**



<https://www.udemy.com/rest-api/?couponCode=REST100>

Coupon Code = **REST100**

## PS:

- For latest coupons & courses please visit: <http://www.acloudfan.com>
- Enter to **WIN Free access** – please visit: <http://www.acloudfan.com/win-free-access>



Authentication



Authorization



Functional attacks



# Basic Authentication, API Key/Secret, JWTs



<http://www.acloudfan.com>

[raj@acloudfan.com](mailto:raj@acloudfan.com)

Summary of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)

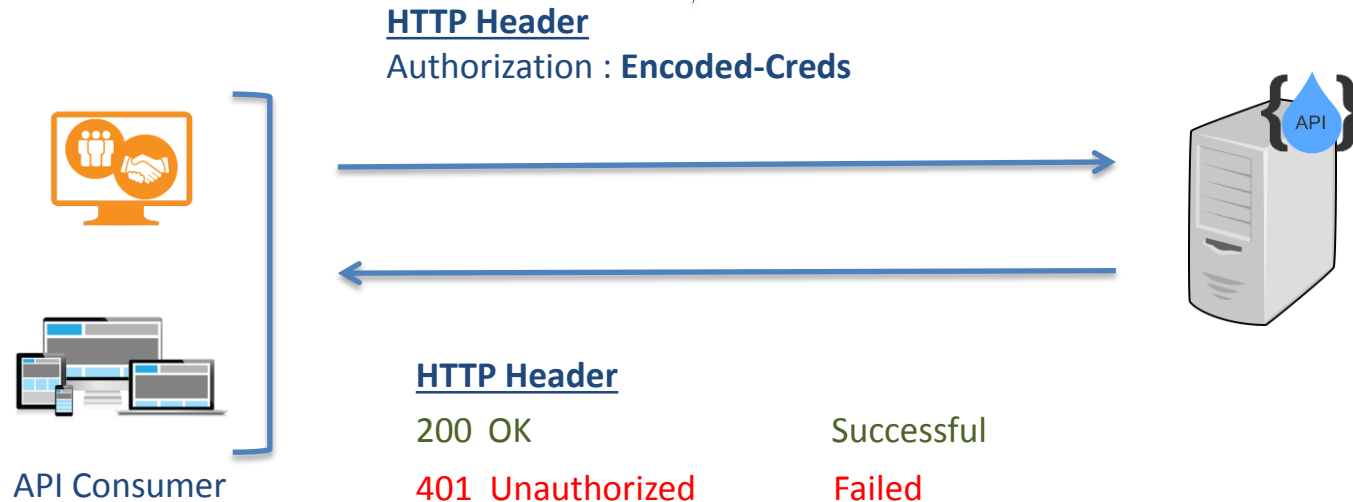
# Basic Authentication

raj@acloudfan.com

<http://www.acloudfan.com>



- Base64 encoded
- User : Password

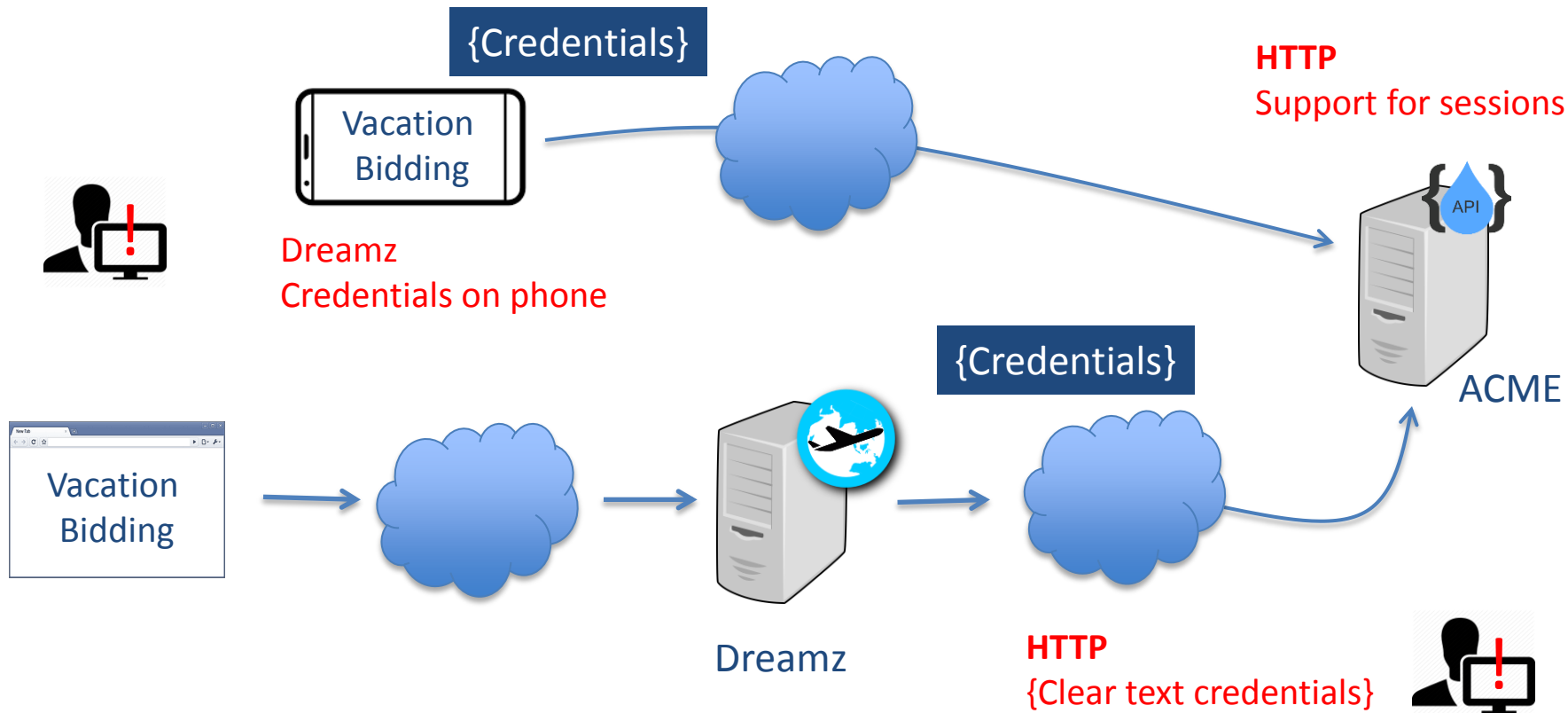


NOT a good idea with HTTP – OK with TLS i.e., HTTPS

# Basic Authentication Weaknesses

raj@acloudfan.com

<http://www.acloudfan.com>





- Sending the API Key & Secret/Signature
  - HTTP Header
  - Query parameters
  - Request body

- API key & secret management
- Security scheme
- API key/secret provisioning
- Rate limiting & analytics



API Management  
Platform

apigee



API Management  
IBM





- **Key** = Long strings of random characters
- API Key/Secret is issued by the API provider
  - Consumer identity
  - Security
  - Rate limiting
- Design considerations related to key/secret management



# Tokens?

raj@acloudfan.com

<http://www.acloudfan.com>



- Encoded string
  - Hashing or private key for encryption
- Eliminates the need for sessions on API
  - HTTP Header
  - Query parameters
  - Request body
- Issuer can control the validity
  - Expiry
  - Revocation



facebook

twitter

LinkedIn



# JWT

raj@acloudfan.com

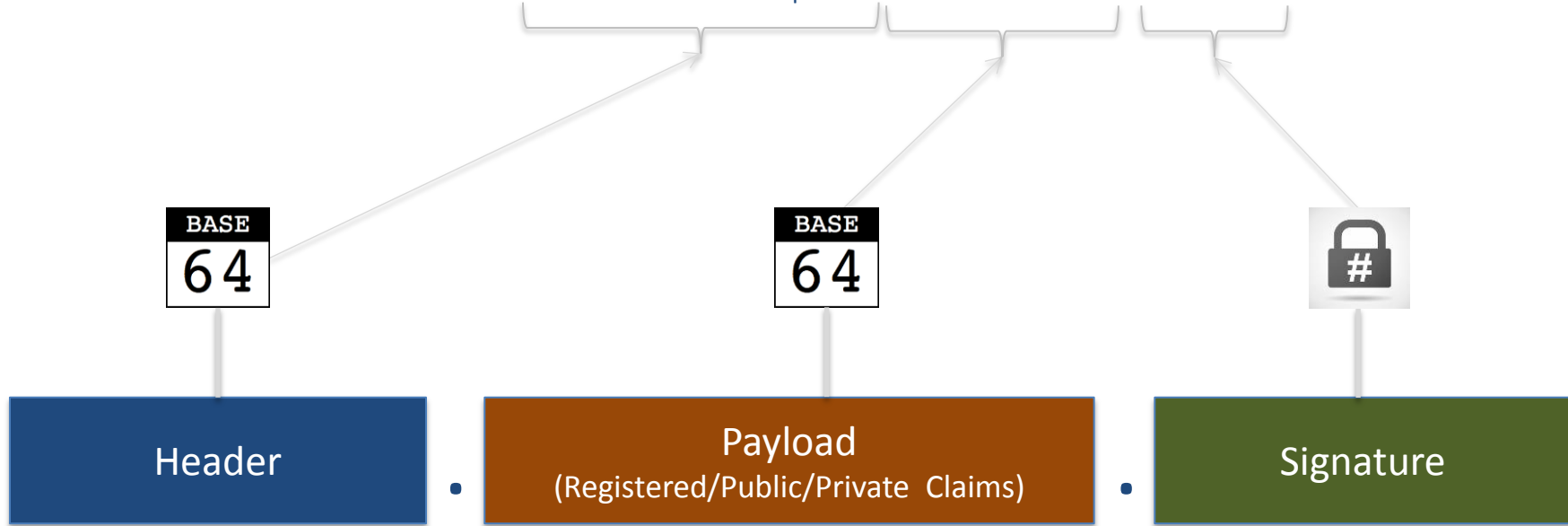
<http://www.acloudfan.com>



## JSON Web Tokens

<https://tools.ietf.org/html/rfc7519>

GciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. F789JHSJH8888SFFLE. HFKKSFHSEF9



Self contained

JSON based

Standard RFC 7519



# Authorization : OAuth2.0

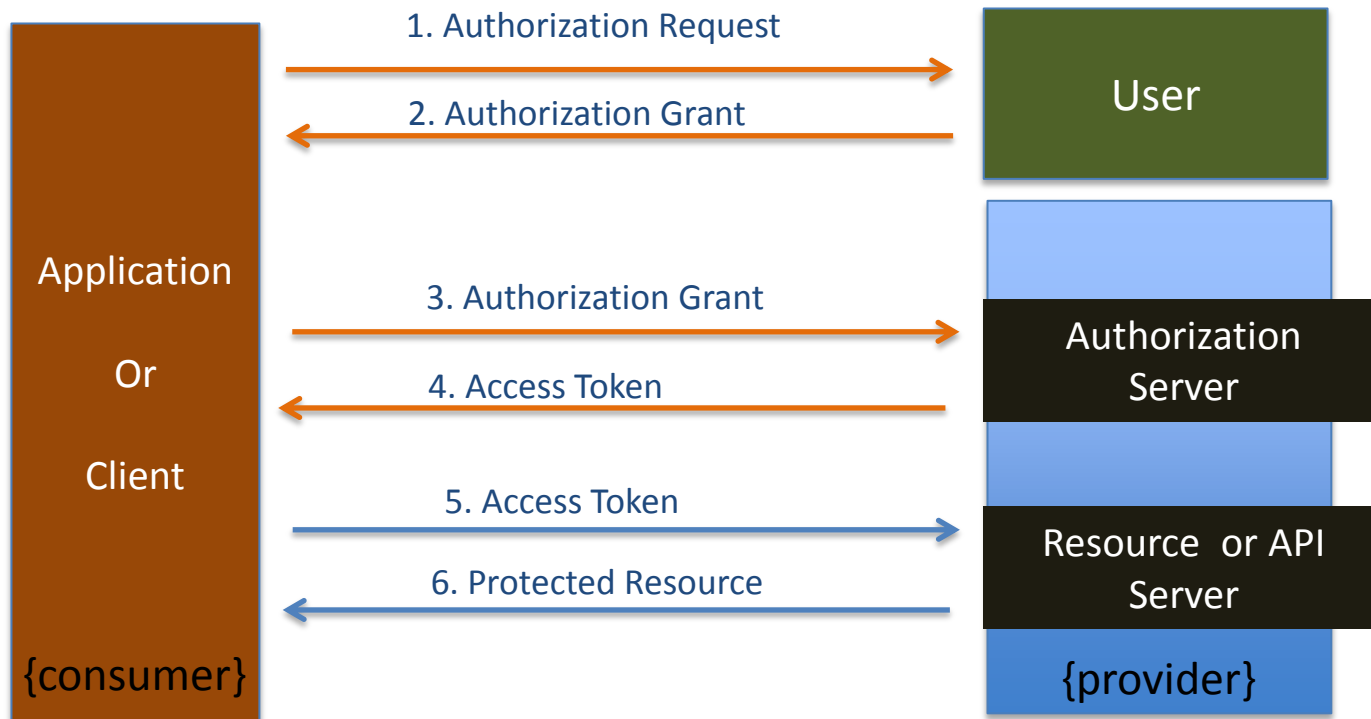


<http://www.acloudfan.com>

[raj@acloudfan.com](mailto:raj@acloudfan.com)

Summary of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)



# OAuth2.0 Grants

raj@acloudfan.com

<http://www.acloudfan.com>



Authorization Scope Grant

Implicit Grant

Resource Owner Credentials Grant

Client Credentials Grant

Refresh Token Grant

**facebook.**



For single page apps & mobile apps

User provides credentials to the app

No authorization needed from User

Refresh token => New access tokens



<https://developer.spotify.com/web-api/authorization-guide/#authorization-code-flow>

FLOW	CAN FETCH A USER'S DATA BY REQUESTING ACCESS?	USES SECRET KEY? (KEY EXCHANGE MUST HAPPEN SERVER-SIDE!)	ACCESS TOKEN CAN BE REFRESHED?
Authorization Scope Grant ✓	Yes	Yes	Yes
Client Credentials Grant ✓	No	Yes	No
Implicit Grant ✓	Yes	No	No
Refresh Token Grant ✓			

~~Resource Owner Credentials Grant~~



- Scopes of the user data (API)
- Type of OAuth grants to be supported
  - Authorization || Implicit Grant Private data
  - Client Credentials Public data
  - For trusted clients would you use “Resource Credentials grant”?
- Implementing the OAuth
  - Build

API Management  
Platform



# Functional Attacks



<http://www.acloudfan.com>

[raj@acloudfan.com](mailto:raj@acloudfan.com)

Parts of a course that covers the A to Z of RESTful API. More information.

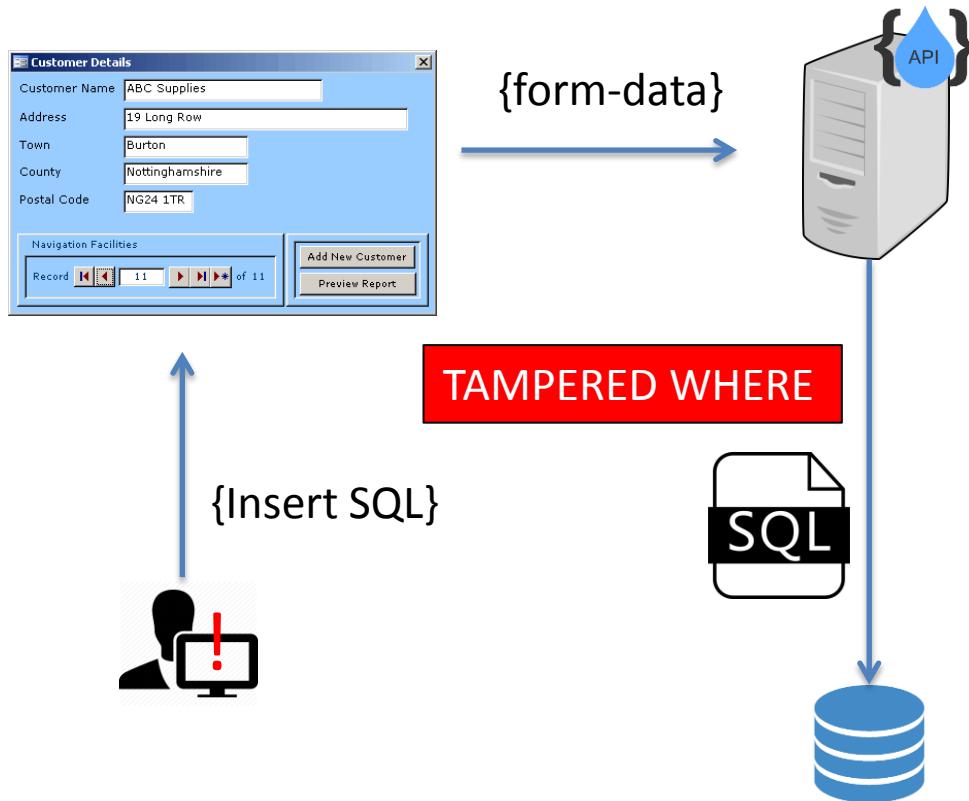
[UDEMY Link With Coupon Discount](#)



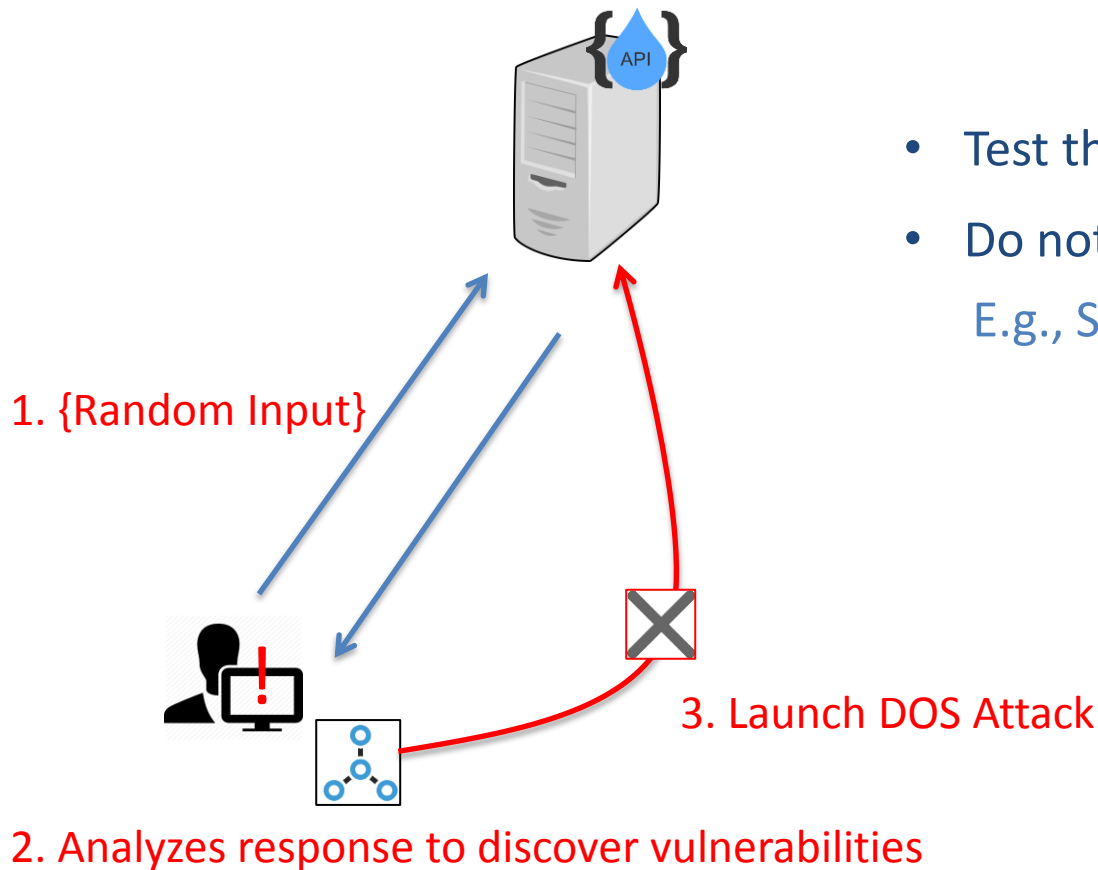
# SQL Injection

[https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection)  
raj@acloudfan.com

<http://www.acloudfan.com>



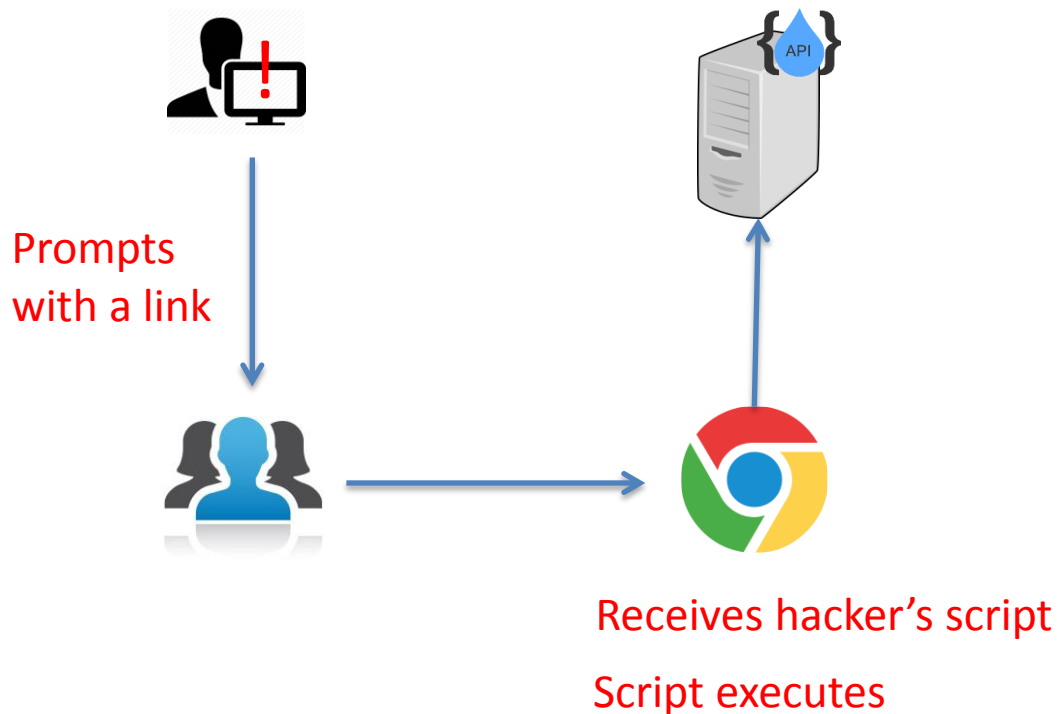
- Read sensitive data
- Modify or delete data
- Take admin action on DB
  - XPath, JSON Path, XSLT



- Test the API with *fuzzing* in mind
- Do not send back the internal errors  
E.g., SQL exceptions



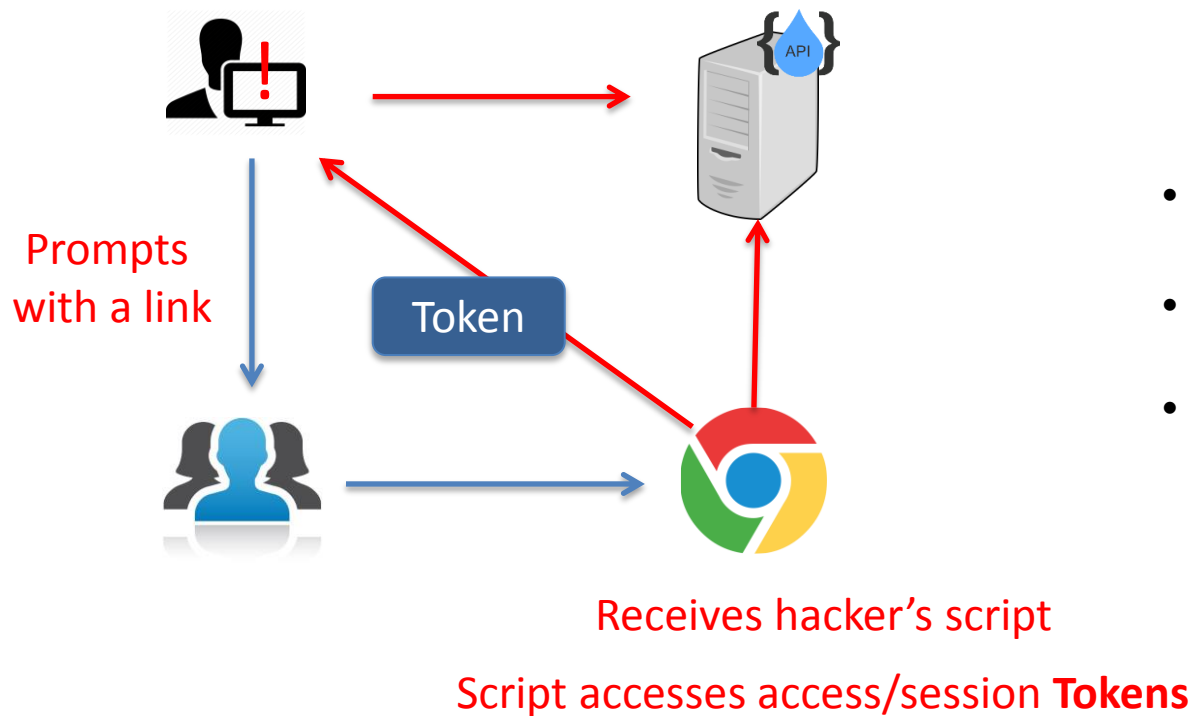
Attacker forces an end user to execute script



- Use POST instead of GET
- Break transaction into steps
- Add custom headers

# Token Hijacking

raj@acloudfan.com  
<http://www.acloudfan.com>



- Ensure expiry of token
- Un-predictable token patterns
- Additional security headers



- Follow the BEST practices for coding and REST API
- Create a process for code review (or adopt peer programming)
- Test & Monitor continuously; Invest in tools
- Select an appropriate security model for API
- Consider an API gateway or API management solution
- Set aside budget for API testing 😊



**SoapUI**