PENETRATION TEST REPORT

ICUBE
Sun 2 Apr 2019

Muhammad Yuga Nugraha – System Administrator

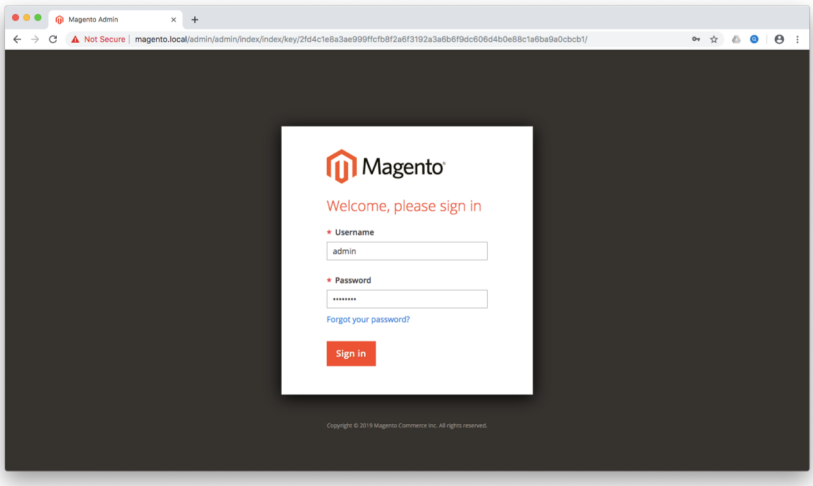
Structure Query Languange injection (SQLi)

Vulnerability	: Unauthenticated SQL injection
Severity	: Critical
Summary	: An unauthenticated user can execute arbitrary code through an SQL injection vulnerability, which causes sensitive data leakage.
Affected(s)	: Magento version 2.2.0 <= 2.3.0

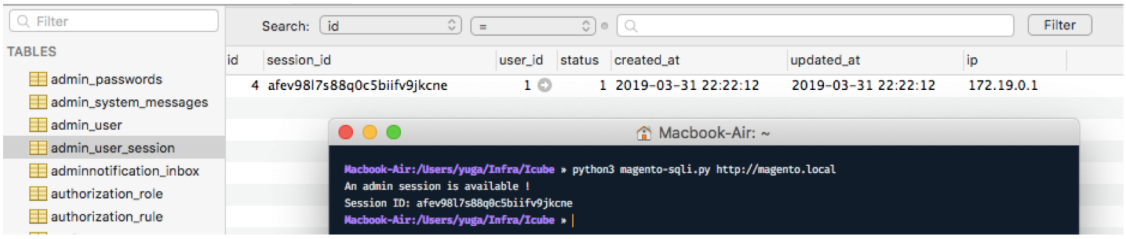
Description : SQL injection is a type of injection attacks to reading information in database from many vector like form, parameter, etc to allow SQL command executed without direct access into database application. Attacker can steal sensitive information using this vulnerability or manipulate data on database. Possible to more access with upload Shell into server and privileges escalation.

Proof of Concept

1. Login in to backend as admin or user, so that the last session saved in database.



2. Download exploit from https://github.com/ambionics/magento-exploits/blob/master/magento-sqli.py and run it, the exploit will show last session id on backend, this is **possible** to get information or manipulate table.

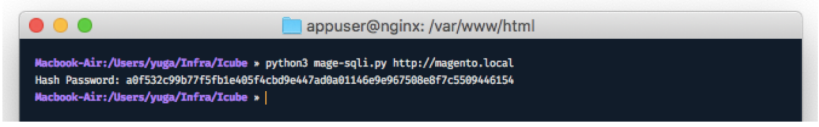


This exploit using some payload to get information last session id from database like this.

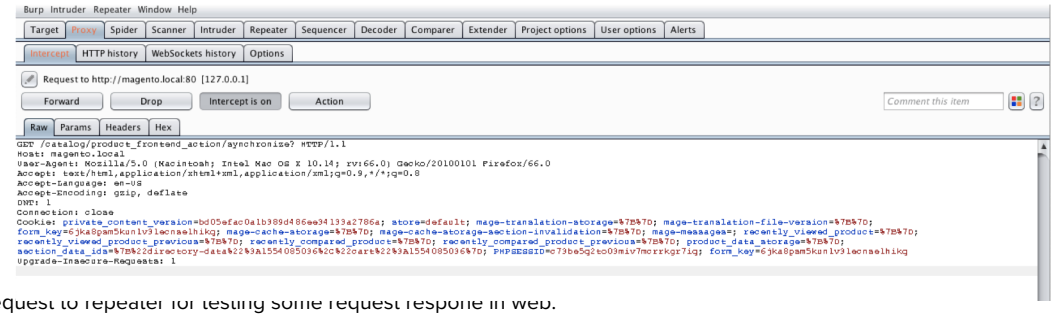
> *SELECT COUNT(*) FROM admin_user_session WHERE TIMESTAMPDIFF(SECOND, updated_at, NOW()) BETWEEN 0 AND 900 ORDER BY created_at DESC, updated_at DESC LIMIT 1*

From that exploit we can modify be like this:

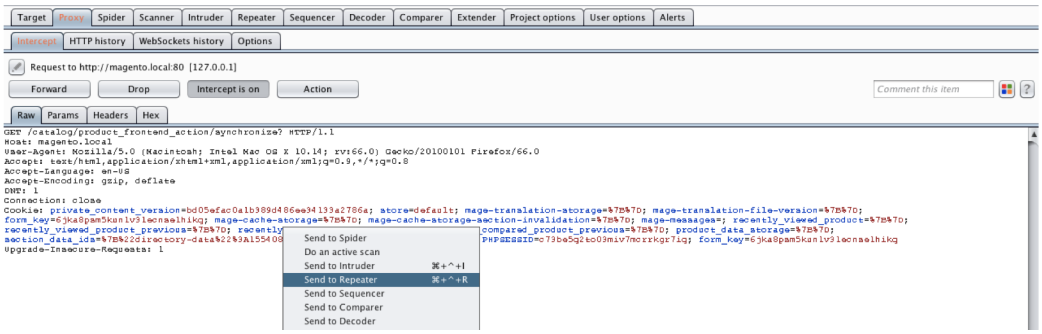
> *SELECT password_hash FROM admin_passwords*



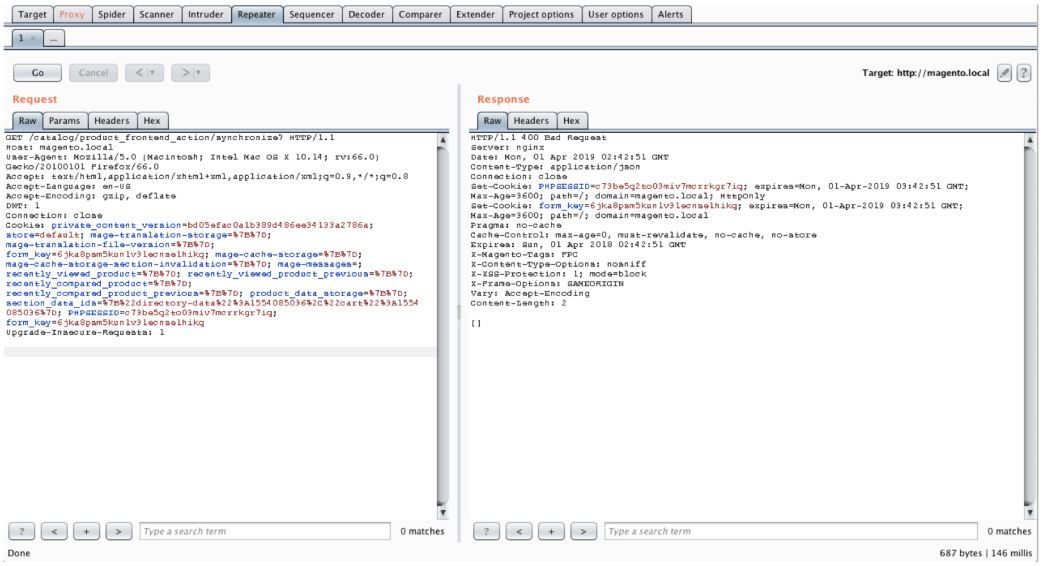
We will test endpoint response using SQL payload on Burp Suite



Send request to repeater for testing some request response in web.



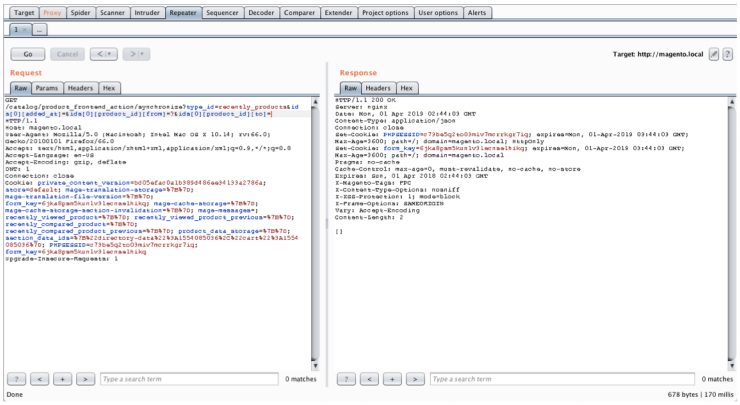
The response say **400 bad request** because wrong value in endpoint.



Let’s add some parameter

/catalog/product_frontend_action/synchronize?type_id=recently_products&ids[0][added_at]=&ids[0][product_id][from]=?&ids[0][product_id][to]=

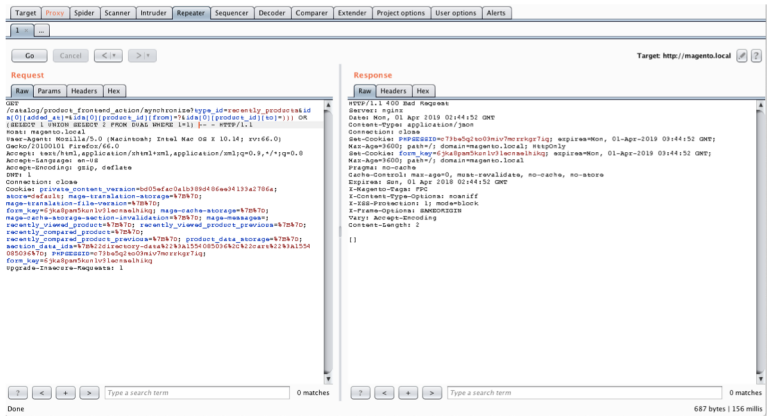
And the result response is **200 Ok**, it means this parameter can be use to input some injection or value.



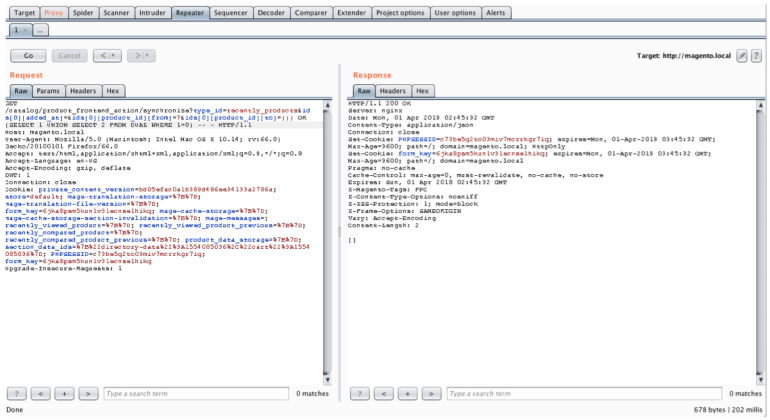
Add SQL payload into endpoint:

/catalog/product_frontend_action/synchronize?type_id=recently_products&ids[0][added_at]=&ids[0][product_id][from]=?&ids[0][product_id][to]=))) OR (SELECT 1 UNION SELECT 2 FROM DUAL WHERE 1=1) -- -

This payload using SQL logic **OR 1=1 -- -** to bypass conditional on Magento code because every endpoint give 400 bad request if parameter wrong and this logic can be use to give response 200 from server.



Change value 1 to 0 to get 200 response.



Impact : - Attacker can steal sensitive information.
- Attacker can manipulate database using SQL query (payload).
- Attacker can use other techniques to get more access like Remote Code Execution.

Remediation : Patch bug from https://magento.com/tech-resources/download#download2288

References : https://www.ambionics.io/blog/magento-sqli
https://meetanshi.com/blog/install-magento-2-patch-prodsecbug-2198/