**Intelligence**

Difficulty: Medium

OS: Windows

**Nmap**

Starting off with our aggressive nmap scan, we see ports 53, 80, 88, 135, 139, 389, 445, 464, 593, 636, 3268, and 3269 are open. A lot of these have to do with ldap so we assume that is the place to search, but first a little enumeration.

|  |
| --- |

**Enumeration**

Looking around smb and rpc, we find nothing of use.

Performing a fuzz scan brings back only one web directory, “documents”. Attempting to go to this directory on the web server results in a 403 error.

Going over to ldap, we begin by gathering some basic information about the service. First, we pull out python3 to execute some commands to connect to ldap. Attempts to use ldapsearch show connection errors, so we use custom python commands.

| *import ldap3*  *server = ldap3.Server('x.X.x.X', get\_info = ldap3.ALL, port =636, use\_ssl = True)*  *connection = ldap3.Connection(server)*  *connection.bind()* |
| --- |

With our python commands, we have successfully made a connection to the ldap service. Now we can query for some basic information.

| *server.info* |
| --- |

We could have also acquired this information through nmap’s “**ldap-rootdse.nse**” script

Going around ldap seems to do nothing. Stepping back, we go look at the website again and see there are some fairly useless documents there. However, these documents are named in a specific way. They begin with the year, month, and day, then end with “-upload.pdf”. If we can construct our own wordlist and fuzz for pdfs, we may find something interesting.

First, we create a bash script to generate all the days in the year 2020. When we execute this, we redirect it to a file called “dates.txt”. After we have this file, we concatenate “-upload.pdf” to every line to complete the wordlist.

| *Sed -e ‘s/$-upload.pdf/’ -i dates.txt* |
| --- |

Performing another fuzz scan, but this time in “/documents”, we acquire a number of pdfs.

Before we go and painstakingly attempt to look at each of these files, we are also going to generate a list for 2021 and 2019. Doing so and running a fuzz scan also shows a few more documents

|  |
| --- |