**Giddy**Difficulty: Medium

Type: Windows

**Nmap**

Starting with our aggressive nmap scan, we find ports 80, 443, and 3389 are open.

|  |
| --- |

**Enumeration**

Looking at the web server, we find a picture of a happy looking dog. There is nothing else on the site, so we start a web fuzzer looking for directories.

| *ffuf -w /opt/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://10.10.10.104/FUZZ* |
| --- |

Going to the “remote” web directory, we get a prompt for windows powershell through a web interface. The site tells us it needs “https” in the header, so we do that and are able to properly look at the page. Exploring the page, we find the web shell requires a username, password, and computer name. There is another option to use computer credentials instead of gateway credentials.

|  |
| --- |

Since we do not have credentials, we move on.

Our fuzzer also found a “mvc” web directory. This one leads to a page containing inventory along with a login

|  |
| --- |

Testing the login, I register and am able to create an account without having to confirm an email address.

Searching here, we come across a search bar.

|  |
| --- |

Whenever there is a place for user input, we always have to test it out for vulnerabilities. Doing this, we find the server is running this search bar through sql. To prevent an error, we put in sql comment marks.

|  |
| --- |

To speed up the process of finding a sql injection, we are goint to capture the request with burp suite, then copy the request to a file. After this we are going to modify the request and send it to sqlmap.

| (modified request with a star “\*” so sqlmap knows what to replace) |
| --- |

While that runs, we notice the URL also contains search information concerning the inventory items. Attempting SQL injection here leads to a success.