ATTACK DEFENSE

by PentesterAcademy

Name	Fingerprinting Webapp (CLI)
URL	https://attackdefense.com/challengedetails?cid=1814
Type	Beginner Skills : Linux For Pentesters

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

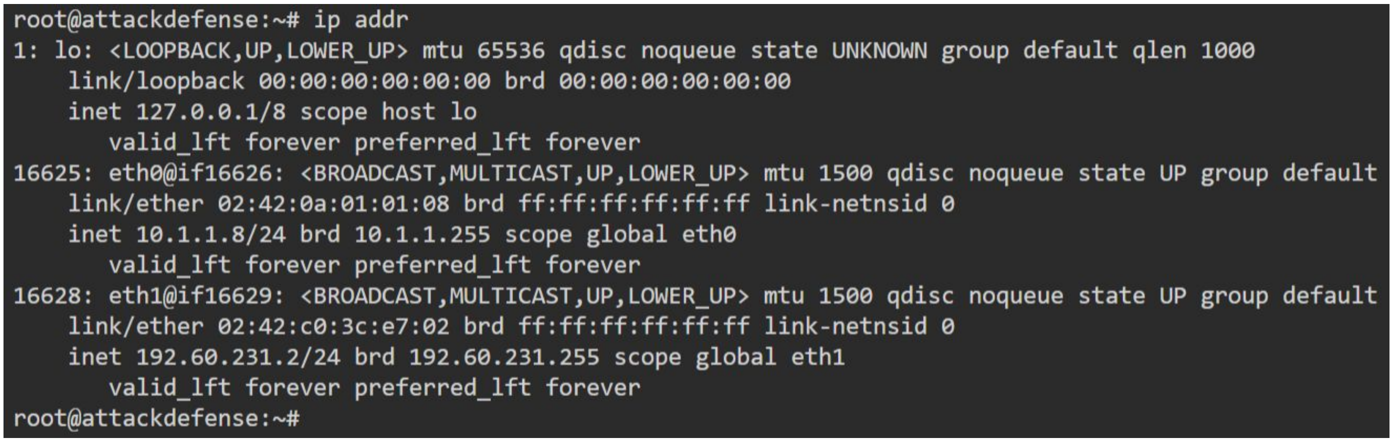
Objective: Fingerprint the WebApp running on target machine using the following utilities/tools:

- **curl**
- **wget**
- **nmap**
- **lynx**
- **browsh**

Solution:

Check the IP address of the machine.

Command: ip addr

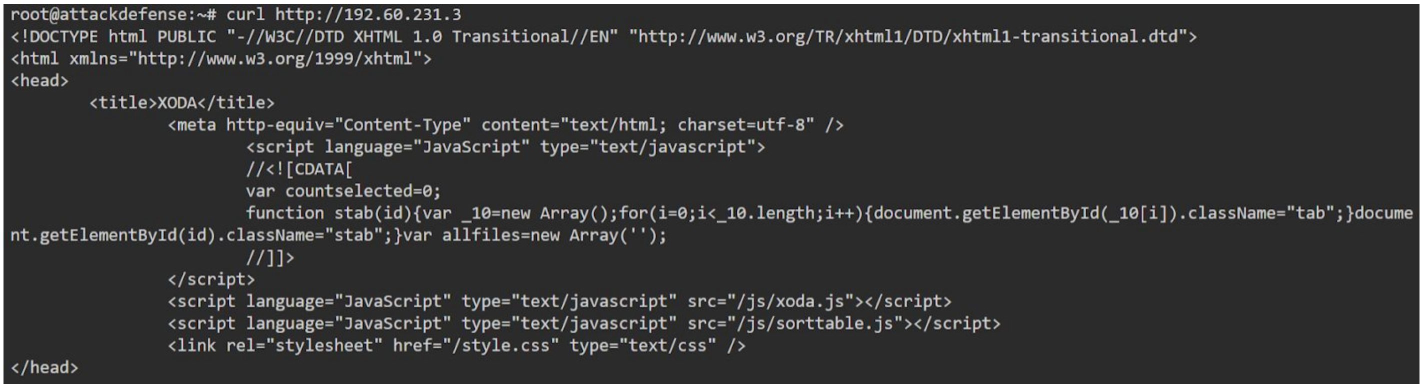


```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
16625: eth0@if16626: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:08 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.8/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
16628: eth1@if16629: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:3c:e7:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.60.231.2/24 brd 192.60.231.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

The IP of user’s machine is 192.60.231.2, so as per the guidelines the IP of remote Linux machine should be 192.60.231.3

Method 1: Using curl

Command: curl 192.60.231.3

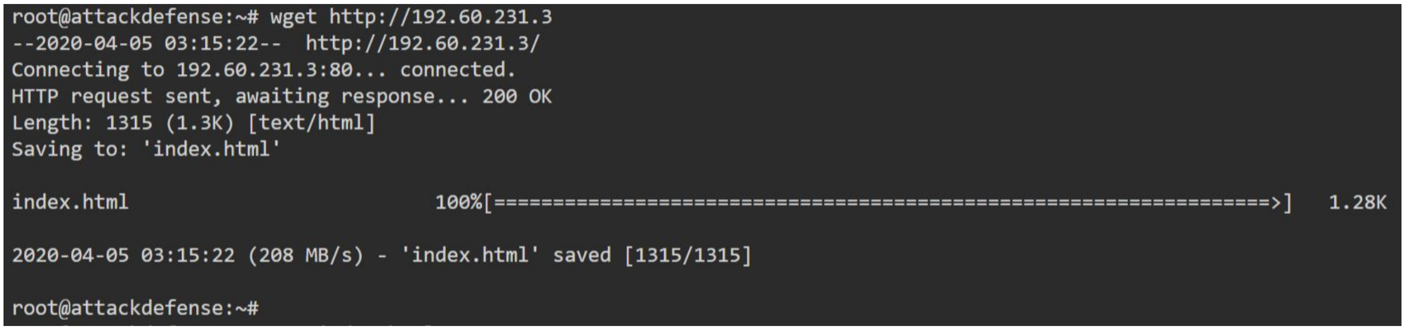


```
root@attackdefense:~# curl http://192.60.231.3
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>XODA</title>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                        <script language="JavaScript" type="text/javascript">
                        //<![CDATA[
                        var countselected=0;
                        function stab(id){var _10=new Array();for(i=0;i<_10.length;i++){document.getElementById(_10[i]).className="tab";}docume
nt.getElementById(id).className="stab";}var allfiles=new Array('');
                        //]]>
                </script>
                <script language="JavaScript" type="text/javascript" src="/js/xoda.js"></script>
                <script language="JavaScript" type="text/javascript" src="/js/sorttable.js"></script>
                <link rel="stylesheet" href="/style.css" type="text/css" />
</head>
```

Method 2: Using wget

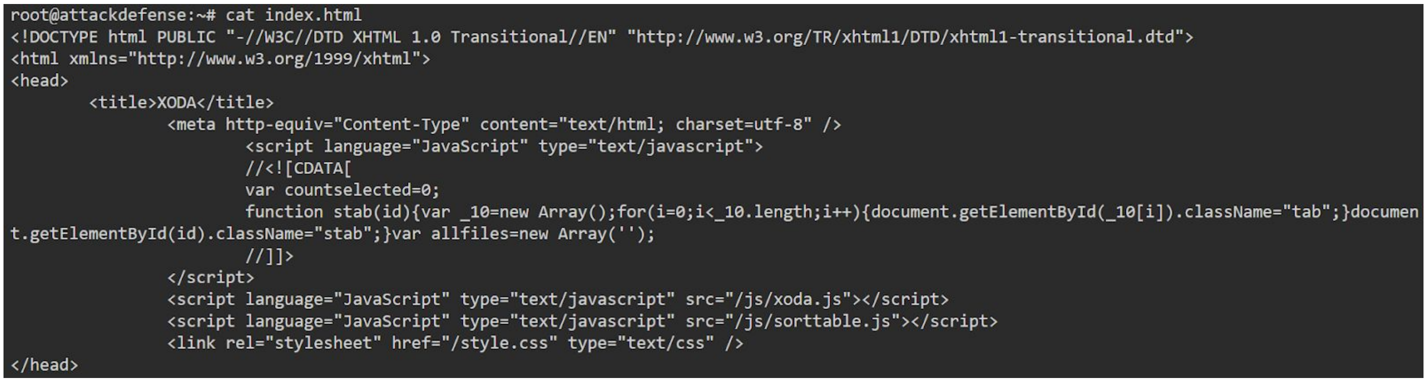
wget can be used to download the HTML page and then read it to know about the application.

Command: wget 192.60.231.3



```
root@attackdefense:~# wget http://192.60.231.3
--2020-04-05 03:15:22--  http://192.60.231.3/
Connecting to 192.60.231.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1315 (1.3K) [text/html]
Saving to: 'index.html'

index.html                    100%[===================================================================>]   1.28K

2020-04-05 03:15:22 (208 MB/s) - 'index.html' saved [1315/1315]

root@attackdefense:~#
```

Command: cat index.html

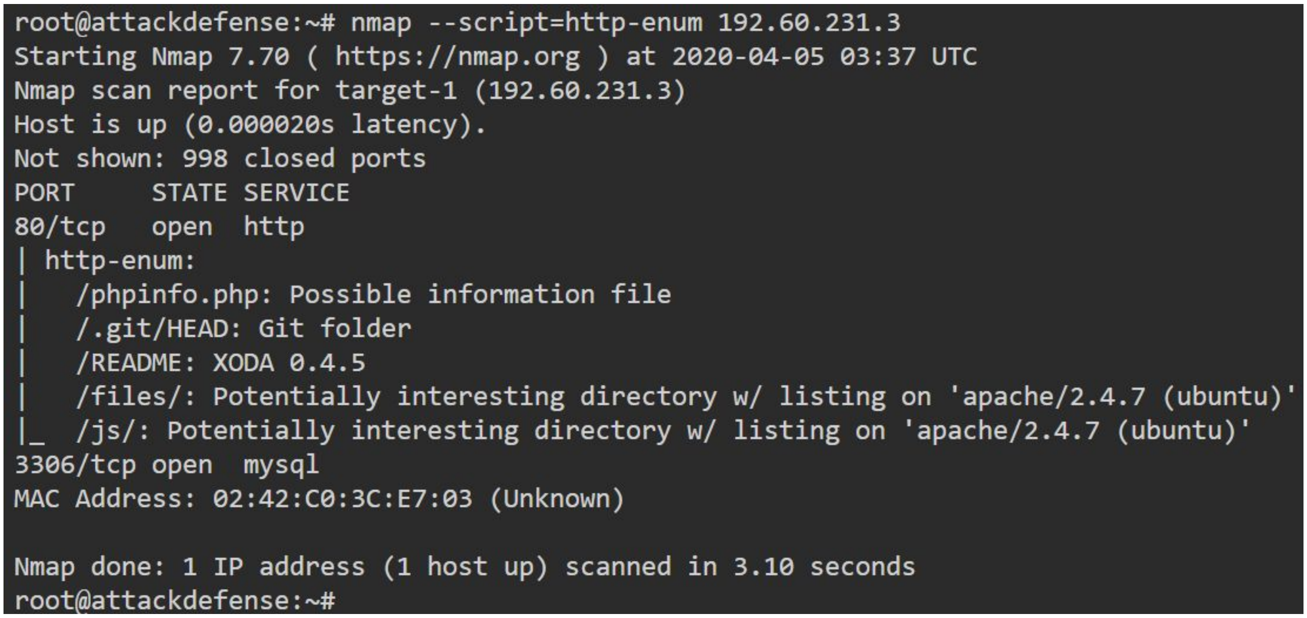


```
root@attackdefense:~# cat index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>XODA</title>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                        <script language="JavaScript" type="text/javascript">
                        //<![CDATA[
                        var countselected=0;
                        function stab(id){var _10=new Array();for(i=0;i<_10.length;i++){document.getElementById(_10[i]).className="tab";}documen
t.getElementById(id).className="stab";}var allfiles=new Array('');
                        //]]>
                </script>
                <script language="JavaScript" type="text/javascript" src="/js/xoda.js"></script>
                <script language="JavaScript" type="text/javascript" src="/js/sorttable.js"></script>
                <link rel="stylesheet" href="/style.css" type="text/css" />
</head>
```

Method 3: Using nmap

Nmap script http-enum can be used to to know about the application.

Command: nmap --script=http-enum 192.60.231.3



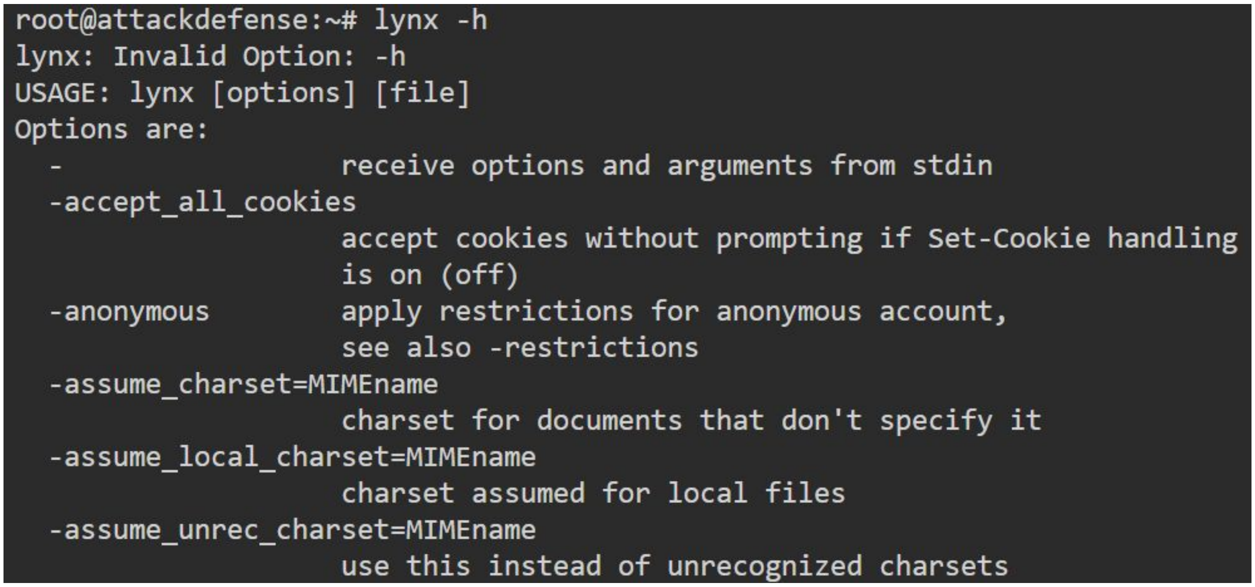
```
root@attackdefense:~# nmap --script=http-enum 192.60.231.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-05 03:37 UTC
Nmap scan report for target-1 (192.60.231.3)
Host is up (0.000020s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
80/tcp   open  http
| http-enum:
|   /phpinfo.php: Possible information file
|   /.git/HEAD: Git folder
|   /README: XODA 0.4.5
|   /files/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
|_  /js/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
3306/tcp open  mysql
MAC Address: 02:42:C0:3C:E7:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 3.10 seconds
root@attackdefense:~#
```

Method 4: Using lynx

Lynx tool can be used to view local and remote HTML pages.

Check the help options for the tool

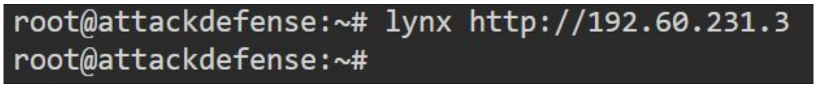
Command: lynx -h



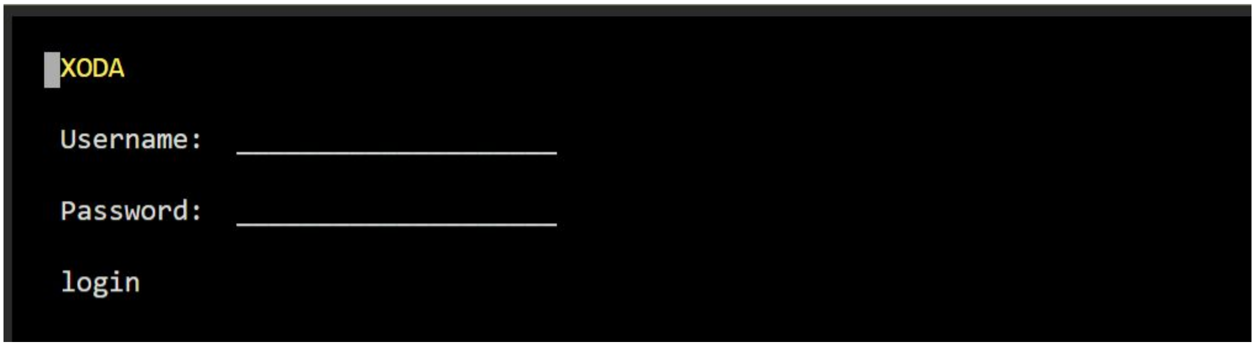
```
root@attackdefense:~# lynx -h
lynx: Invalid Option: -h
USAGE: lynx [options] [file]
Options are:
  -                    receive options and arguments from stdin
  -accept_all_cookies
                       accept cookies without prompting if Set-Cookie handling
                       is on (off)
  -anonymous           apply restrictions for anonymous account,
                       see also -restrictions
  -assume_charset=MIMEname
                       charset for documents that don't specify it
  -assume_local_charset=MIMEname
                       charset assumed for local files
  -assume_unrec_charset=MIMEname
                       use this instead of unrecognized charsets
```

Use lynx to open the remote web page.

Command: lynx http://192.60.231.3

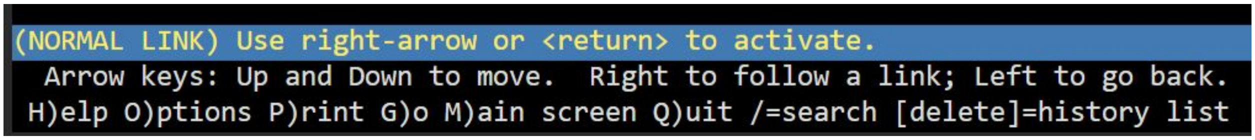


```
root@attackdefense:~# lynx http://192.60.231.3
root@attackdefense:~#
```



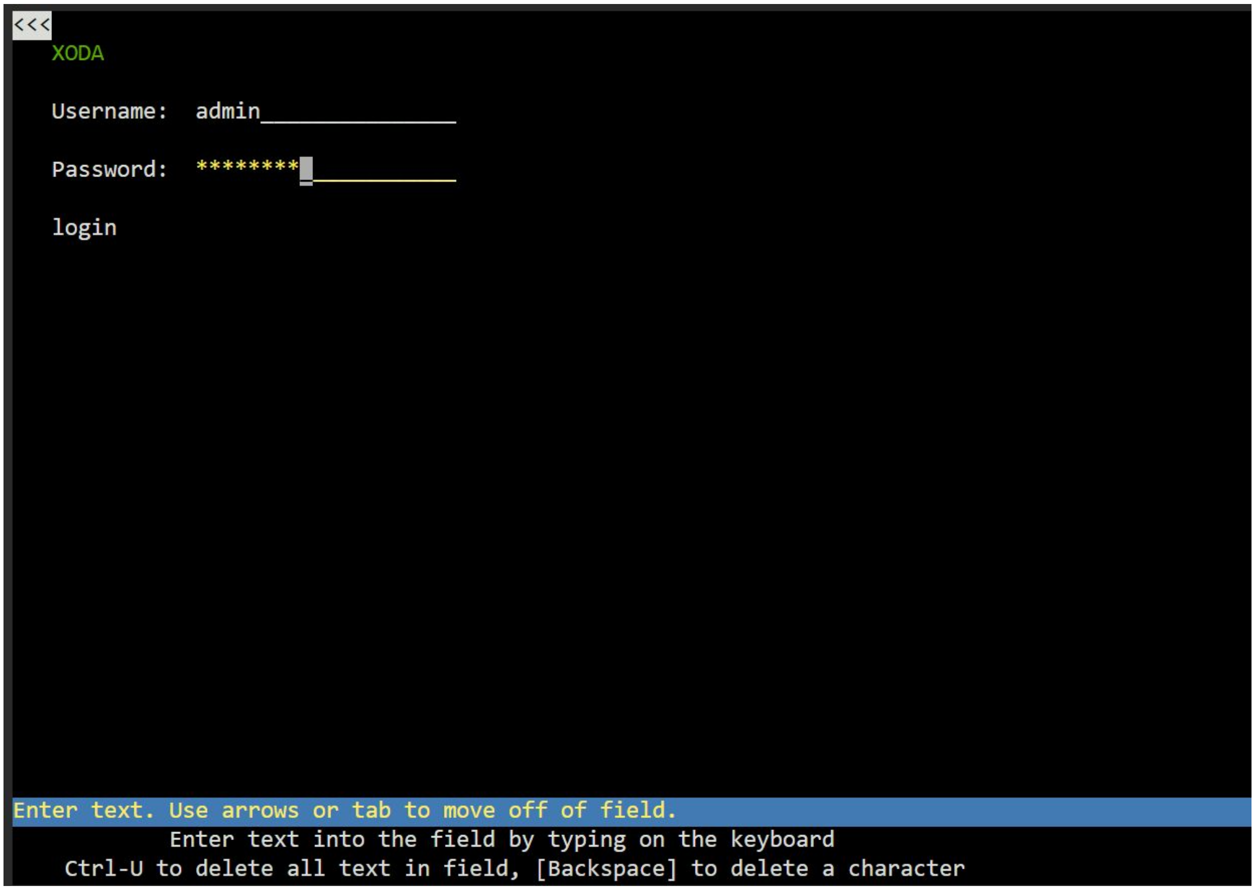
```
XODA

 Username: ______________________

 Password: ______________________

 login
```

The options to interact with lynx appears in the bottom part.



```
(NORMAL LINK) Use right-arrow or <return> to activate.
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

It also supports typing into text fields and submitting values to forms.



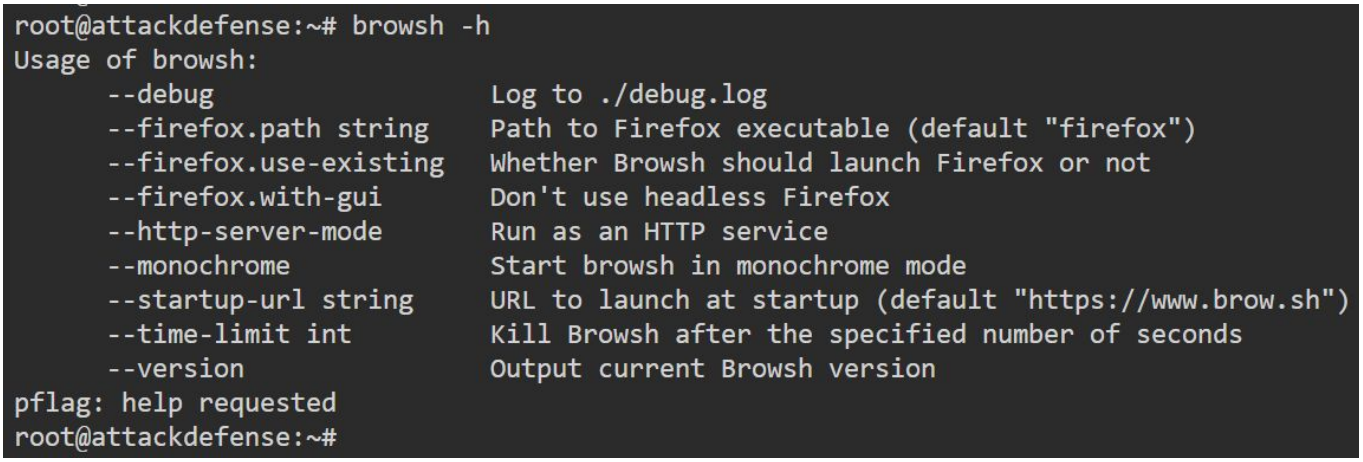
```
<<<
   XODA

   Username:  admin_______________

   Password:  ********____________

   login

Enter text. Use arrows or tab to move off of field.
                Enter text into the field by typing on the keyboard
      Ctrl-U to delete all text in field, [Backspace] to delete a character
```

Method 5: Using browsh

Browsh uses firefox to represent the web page on CLI.

Check the help options for the tool

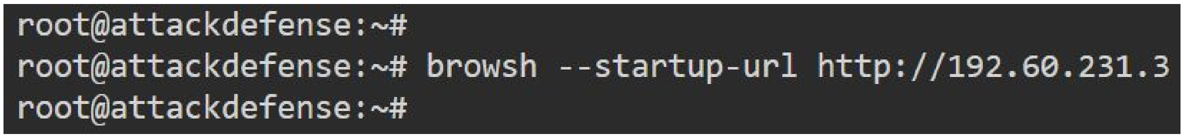
Command: browsh -h



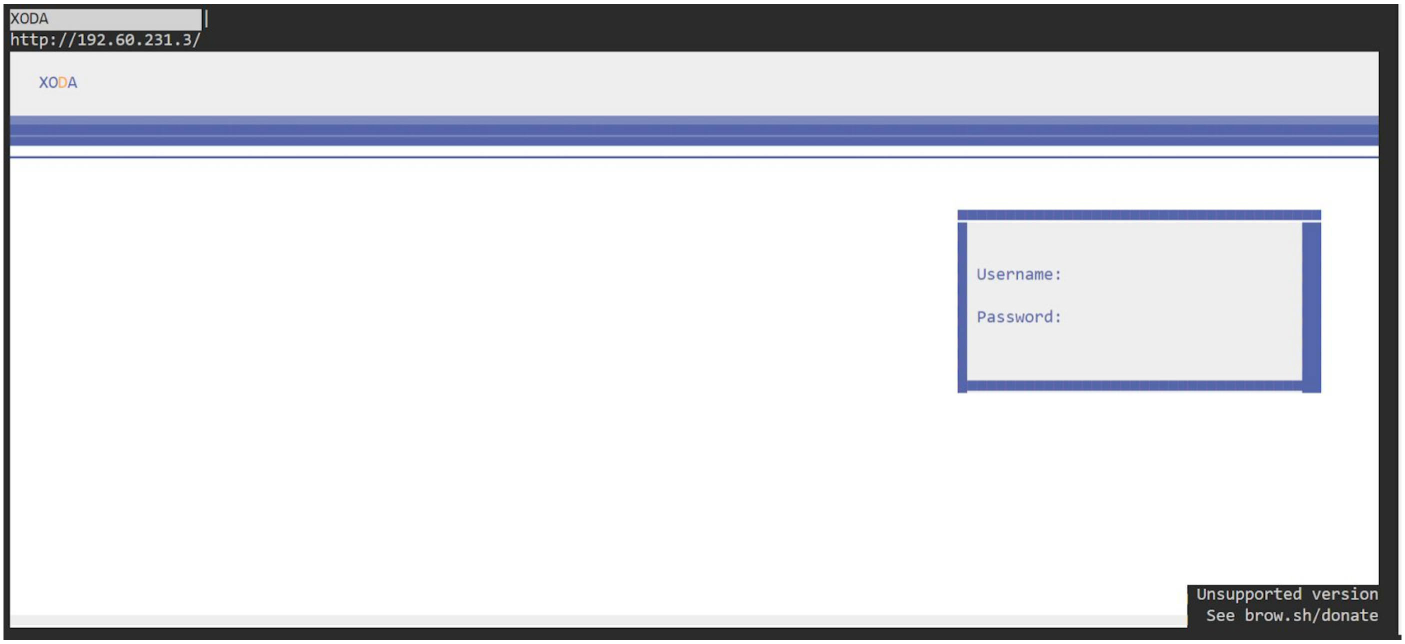
```
root@attackdefense:~# browsh -h
Usage of browsh:
      --debug                   Log to ./debug.log
      --firefox.path string     Path to Firefox executable (default "firefox")
      --firefox.use-existing    Whether Browsh should launch Firefox or not
      --firefox.with-gui        Don't use headless Firefox
      --http-server-mode        Run as an HTTP service
      --monochrome              Start browsh in monochrome mode
      --startup-url string      URL to launch at startup (default "https://www.brow.sh")
      --time-limit int          Kill Browsh after the specified number of seconds
      --version                 Output current Browsh version
pflag: help requested
root@attackdefense:~#
```

Use browsh to open the remote web page.

Command: browsh --startup-url http://192.60.231.3



```
root@attackdefense:~#
root@attackdefense:~# browsh --startup-url http://192.60.231.3
root@attackdefense:~#
```



XODA
http://192.60.231.3/

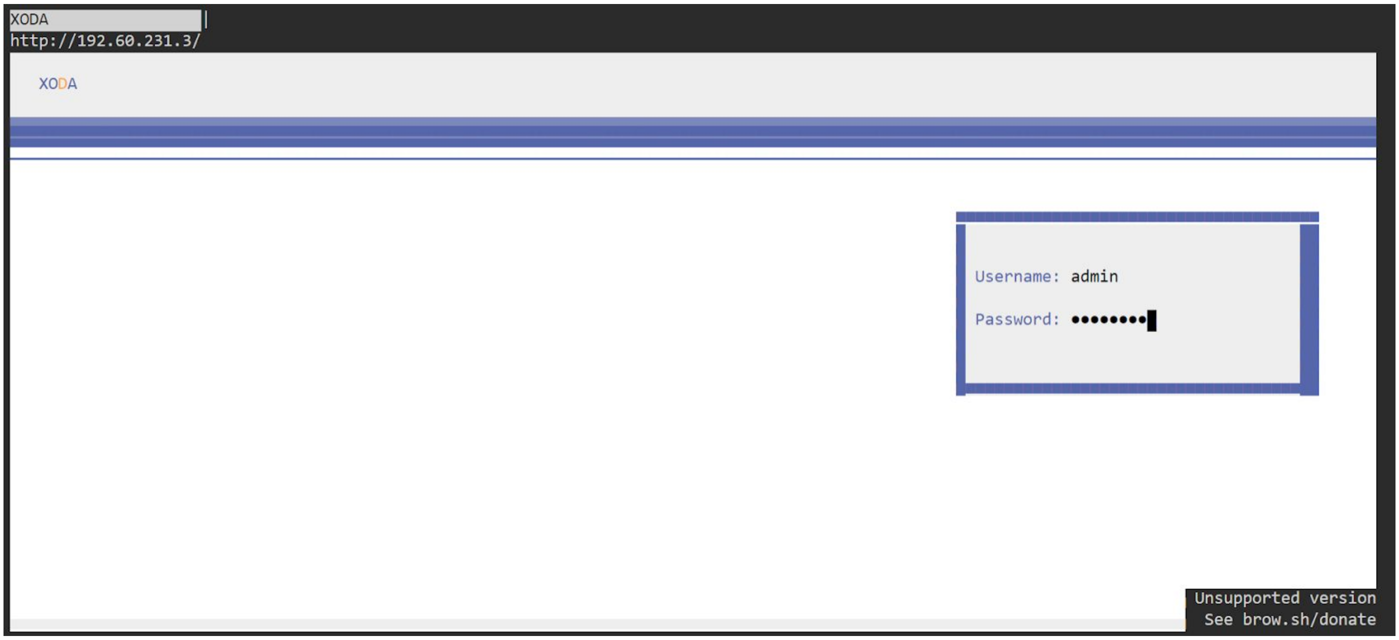
XODA

Username:

Password:

Unsupported version
See brow.sh/donate

It also supports submitting values to forms and interacting with clickable links/buttons.



XODA
http://192.60.231.3/

XODA

Username: admin

Password: ••••••••

Unsupported version
See brow.sh/donate

References:

- Lynx (https://linux.die.net/man/1/lynx)
- Browsh (https://www.brow.sh/)