Learn More

About Cisco
Portcullis Labs
Research and Development

- Home
- Advisories
- Blog
- Presentations
- Tools
- Whitepapers
- Downloads

enum4linux

Published 16/09/2008 | By MRL

A Linux alternative to enum.exe for enumerating data from Windows and Samba hosts.

Key features

- RID cycling (When RestrictAnonymous is set to 1 on Windows 2000)
- User listing (When RestrictAnonymous is set to 0 on Windows 2000)
- Listing of group membership information
- Share enumeration
- Detecting if host is in a workgroup or a domain
- Identifying the remote operating system
- Password policy retrieval (using polenum)

Overview

Enum4linux is a tool for enumerating information from Windows and Samba systems. It attempts to offer similar functionality to enum.exe formerly available from www.bindview.com.

It is written in Perl and is basically a wrapper around the Samba tools smbclient, rpclient, net and nmblookup.

The tool usage can be found below followed by examples, previous versions of the tool can be found at the bottom of the page.

Dependencies

You will need to have the Samba package installed as this script is basically just a wrapper around rpcclient, net, nmblookup and smbclient.

Usage

```
$ enum4linux.pl -h
enum4linux v0.8.2 (https://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2006 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar functiona
to enum (http://www.bindview.com/Services/RAZOR/Utilities/Windows/enum_readme.cfm
Some additional features such as RID cycling have also been added for convenience

This is an ALPHA release only.  Some of the options supported by the original "en
aren't implemented in this release.

Usage: /usr/local/bin/enum4linux.pl [options] ip

Options are (like "enum"):
-U              get userlist
-M              get machine list*
-N              get namelist dump (different from -U|-M)*
-S              get sharelist
-P              get password policy information*
-G              get group and member list
-L              get LSA policy information*
-D              dictionary crack, needs -u and -f*
-d              be detailed, applies to -U and -S
-u username     specify username to use (default "")
-p password     specify password to use (default "")
-f filename     specify dictfile to use (wants -D)*

* = Not implemented in this release.

Additional options:
-a              Do all simple enumeration (-U -S -G -r -o -n)
-h              Display this help message and exit
-r              enumerate users via RID cycling
-R range        RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
-s filename     brute force guessing for share names
-k username     User(s) that exists on remote system (default: administrator,guest
Used to get sid with "lookupsid known_username"
Use commas to try several users: "-k admin,user1,user2"
-o              Get OS information
-i              Get printer information
-w workgroup    Specify workgroup manually (usually found automatically)
-n              Do an nmblookup (similar to nbtstat)
-v              Verbose.  Shows full commands being run (net, rpcclient, etc.)

RID cycling should extract a list of users from Windows (or Samba) hosts which ha
RestrictAnonymous set to 1 (Windows NT and 2000), or "Network access: Allow
anonymous SID/Name translation" enabled (XP, 2003).

If no usernames are known, good names to try against Windows systems are:
- administrator
- guest
- none
- helpassistant
- aspnet

The following might work against samba systems:
- root
- nobody
- sys
```

```
NB: Samba servers often seem to have RIDs in the range 3000-3050.
```

Examples

Below are examples which demonstrate most of the features of enum4linux. Output has been edited for brevity in most cases.

Verbose mode

Before we delve into the features of enum4linux, it’s worth pointing out that verbose mode shows you the underlying commands being run by enum4linux (rpcclient, smblient, etc.). This is useful if you want to use the underlying commands manually, but can’t figure out the syntax to use. Note the lines beginning with [V] in the output below:

```
$ enum4linux.pl -v 192.168.2.55
[V] Dependent program "nmblookup" found in /usr/bin/nmblookup
[V] Dependent program "net" found in /usr/bin/net
[V] Dependent program "rpcclient" found in /usr/bin/rpcclient
[V] Dependent program "smbclient" found in /usr/bin/smbclient
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Enumerating Workgroup/Domain on 192.168.2.55 ------
[V] Attempting to get domain name with command: nmblookup -A '192.168.2.55'
[+] Got domain/workgroup name: WORKGROUP

----- Getting domain SID for 192.168.2.55 -----
[V] Attempting to get domain SID with command: rpcclient -U''%'' 192.168.2.55 -c
Domain Name: WORKGROUP
Domain Sid: S-0-0
[+] Host is part of a workgroup (not a domain)

----- Session Check on 192.168.2.55 -----
[V] Attempting to make null session using command: smbclient //'192.168.2.55'/ipc
[+] Server 192.168.2.55 allows sessions using username '', password ''
```

The “Do Everything” option

As you read through the following section you’ll probably think that there are a lot of options you need to remember. If you just want enum4linux to try to enumerate all the information it can from a remote host, just use the -a option:

```
$ enum4linux.pl -a 192.168.2.55
```

NB: This won’t do dictionary-based share name guessing, but does pretty much everything else.

Obtain list of usernames (RestrictAnonymous = 0)

This feature is similar to enum.exe -U IP. It returns a complete list of usernames if the server allows it. On Windows 2000 the RestrictAnonymous registry setting must be set to 0 for this feature to work. The user list is show twice in two different formats because type different underlying commands are used to retrieve the data.

```
$ enum4linux.pl -U 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Users on 192.168.2.55 -----
```

```
index: 0x1 RID: 0x1f4 acb: 0x210 Account: Administrator Name: Desc: Built-in acco
index: 0x2 RID: 0x3ee acb: 0x10 Account: basic Name: basic Desc:
index: 0x3 RID: 0x3ed acb: 0x10 Account: blah Name: Desc:
index: 0x4 RID: 0x1f5 acb: 0x215 Account: Guest Name: Desc: Built-in account for
index: 0x5 RID: 0x3e9 acb: 0x214 Account: IUSR_PORTCULLIS Name: Internet Guest Ac
index: 0x6 RID: 0x3ea acb: 0x214 Account: IWAM_PORTCULLIS Name: Launch IIS Proces
index: 0x7 RID: 0x3ec acb: 0x10 Account: mark Name: Desc:
index: 0x8 RID: 0x3e8 acb: 0x214 Account: TsInternetUser Name: TsInternetUser Des

user:[Administrator] rid:[0x1f4]
user:[basic] rid:[0x3ee]
user:[blah] rid:[0x3ed]
user:[Guest] rid:[0x1f5]
user:[IUSR_PORTCULLIS] rid:[0x3e9]
user:[IWAM_PORTCULLIS] rid:[0x3ea]
user:[mark] rid:[0x3ec]
user:[TsInternetUser] rid:[0x3e8]
```

Obtain a list of usernames (using authentication)

users regardless of RestrictAnonymous settings. In the example below we use the administrator account, but any account will do:

```
$ enum4linux.pl -u administrator -p password -U 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Users on 192.168.2.55 -----
index: 0x1 RID: 0x1f4 acb: 0x210 Account: Administrator Name: Desc: Built-in acco
index: 0x2 RID: 0x3ee acb: 0x10 Account: basic Name: basic Desc:
index: 0x3 RID: 0x3ed acb: 0x10 Account: blah Name: Desc:
index: 0x4 RID: 0x1f5 acb: 0x215 Account: Guest Name: Desc: Built-in account for
index: 0x5 RID: 0x3e9 acb: 0x214 Account: IUSR_PORTCULLIS Name: Internet Guest Ac
index: 0x6 RID: 0x3ea acb: 0x214 Account: IWAM_PORTCULLIS Name: Launch IIS Proces
index: 0x7 RID: 0x3ec acb: 0x10 Account: mark Name: Desc:
index: 0x8 RID: 0x3e8 acb: 0x214 Account: TsInternetUser Name: TsInternetUser Des

user:[Administrator] rid:[0x1f4]
user:[basic] rid:[0x3ee]
user:[blah] rid:[0x3ed]
user:[Guest] rid:[0x1f5]
user:[IUSR_PORTCULLIS] rid:[0x3e9]
user:[IWAM_PORTCULLIS] rid:[0x3ea]
user:[mark] rid:[0x3ec]
user:[TsInternetUser] rid:[0x3e8]
```

Obtaining a List of Usernames via RID Cycling (RestrictAnonymous = 1)

To obtain the usernames corresponding to a default range of RIDs (500-550,1000-1050) use the -r option:

```
$ enum4linux.pl -r 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Target information -----
Target ........... 192.168.2.55
RID Range ........ 500-550,1000-1050
```

```
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

----- Users on 192.168.2.55 via RID cycling (RIDS: 500-550,1000-1050) -----
[I] Assuming that user "administrator" exists
[+] Got SID: S-1-5-21-1801674531-1482476501-725345543 using username '', password
S-1-5-21-1801674531-1482476501-725345543-500 W2KSQL\Administrator (Local User)
S-1-5-21-1801674531-1482476501-725345543-501 W2KSQL\Guest (Local User)
S-1-5-21-1801674531-1482476501-725345543-513 W2KSQL\None (Domain Group)
S-1-5-21-1801674531-1482476501-725345543-1000 W2KSQL\TsInternetUser (Local User)
S-1-5-21-1801674531-1482476501-725345543-1001 W2KSQL\IUSR_PORTCULLIS (Local User)
S-1-5-21-1801674531-1482476501-725345543-1002 W2KSQL\IWAM_PORTCULLIS (Local User)
S-1-5-21-1801674531-1482476501-725345543-1004 W2KSQL\mark (Local User)
S-1-5-21-1801674531-1482476501-725345543-1005 W2KSQL\blah (Local User)
S-1-5-21-1801674531-1482476501-725345543-1006 W2KSQL\basic (Local User)
```

You can specify a custom range of RIDs using the -R option. This implies -r, so your don't have specify the -r option:

```
$ enum4linux.pl -R 500-520 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Target information -----
Target ........... 192.168.2.55
RID Range ........ 500-520
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

----- Users on 192.168.2.55 via RID cycling (RIDS: 500-520) -----
[I] Assuming that user "administrator" exists
[+] Got SID: S-1-5-21-1801674531-1482476501-725345543 using username '', password
S-1-5-21-1801674531-1482476501-725345543-500 W2KSQL\Administrator (Local User)
S-1-5-21-1801674531-1482476501-725345543-501 W2KSQL\Guest (Local User)
S-1-5-21-1801674531-1482476501-725345543-513 W2KSQL\None (Domain Group)
```

Before RID cycling can start, enum4linux needs to get the SID from the remote host. It does this by requesting the SID of a known username / group (pretty much the same thing every other RID-cycling tool does). You can see in the above output a list of known usernames. These are tried in turn, until enum4linux finds the SID of the remote host.

If you've very unlucky, this list won't be good enough and you won't be able to get the SID. In this case, use the -k option to specify a different known username:

```
$ enum4linux.pl -k anotheruser -R 500-520 192.168.2.55
```

You can specify a list using commas:

```
$ enum4linux.pl -k user1,user2,user3 -R 500-520 192.168.2.55
```

Group membership

If the remote host allow it, you can get a list of groups and their members using the -G option (like in enum.exe):

```
$ enum4linux.pl -G 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Groups on 192.168.2.55 -----
[+] Getting builtin groups:
```

```
group:[Administrators] rid:[0x220]
group:[Backup Operators] rid:[0x227]
group:[Guests] rid:[0x222]
group:[Power Users] rid:[0x223]
group:[Replicator] rid:[0x228]
group:[Users] rid:[0x221]

[+] Getting builtin group memberships:
Group 'Guests' (RID: 546) has members:
W2KSQL\Guest
W2KSQL\TsInternetUser
W2KSQL\IUSR_PORTCULLIS
W2KSQL\IWAM_PORTCULLIS
Group 'Users' (RID: 545) has members:
NT AUTHORITY\INTERACTIVE
NT AUTHORITY\Authenticated Users
W2KSQL\mark
W2KSQL\blah
W2KSQL\basic
Group 'Replicator' (RID: 552) has members:
Group 'Power Users' (RID: 547) has members:
Group 'Administrators' (RID: 544) has members:
W2KSQL\Administrator
W2KSQL\mark
W2KSQL\blah
Group 'Backup Operators' (RID: 551) has members:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:
group:[None] rid:[0x201]

[+] Getting domain group memberships:
Group 'None' (RID: 513) has members:
W2KSQL\Administrator
W2KSQL\Guest
W2KSQL\TsInternetUser
W2KSQL\IUSR_PORTCULLIS
W2KSQL\IWAM_PORTCULLIS
W2KSQL\mark
W2KSQL\blah
W2KSQL\basic
```

As with the -U option for user enumeration, you can also specify -u user -p pass to provide login credentials if required. Any user account will do, you don’t have to be an admin.

Enum4linux uses rpcclient’s lsaquery command to ask for a host’s Domain SID. If we get a proper SID we can infer that it is part of a domain. If we get the answer S-0-0 we can infer the host is part of a workgroup. This is done by default, so no command line options are required:

```
$ enum4linux.pl 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Getting domain SID for 192.168.2.55 -----
```

```
Domain Name: WORKGROUP
Domain Sid: S-0-0
[+] Host is part of a workgroup (not a domain)
```

Getting nbtstat Information

human-readable information about the remote host.

```
$ enum4linux.pl -n 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Nbtstat Information for 192.168.2.55 -----
Looking up status of 192.168.2.55
W2KSQL <00> - B <tt>Workstation Service
W2KSQL <20> - B </tt><tt>File Server Service
WORKGROUP <00> - </tt><tt>B </tt><tt>Domain/Workgroup Name
INet~Services <1c> - </tt><tt>B </tt><tt>IIS
WORKGROUP <1e> - </tt><tt>B </tt><tt>Browser Service Elections
W2KSQL <03> - B </tt><tt>Messenger Service
IS~W2KSQL <00> - B </tt><tt>IIS
ADMINISTRATOR <03> - B </tt><tt>Messenger Service</tt>

MAC Address = 00-0C-29-A4-12-6C
```

Listing Windows shares

If the server allows it, you can obtain a complete list of shares with the -S option. This uses smbclient under the bonnet which also seems to grab the browse list.

Enum4linux will also attempt to connect to each share with the supplied credentials (null session usually, but you could use -u user -p pass to use something else). It will report whether it could connect to the share and whether it was possible to get a directory listing.

```
$ enum4linux.pl -S 192.168.2.55
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Enumerating Workgroup/Domain on 192.168.2.55 ------
[+] Got domain/workgroup name: WORKGROUP

----- Share Enumeration on 192.168.2.55 -----
Domain=[WORKGROUP] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]

Sharename Type Comment
--------- ---- -------
IPC$ IPC Remote IPC
ADMIN$ Disk Remote Admin
C$ Disk Default share
session request to 192.168.2.55 failed (Called name not present)
session request to 192 failed (Called name not present)

Server Comment
--------- -------
W2KSQL
WEBVULNB
WINORACLE
```

```

Workgroup Master
--------- -------
PTT SBS
WORKGROUP WEBVULNB

----- Attempting to map to shares on 192.168.2.55 -----
//192.168.2.55/IPC$ Mapping: OK Listing: DENIED
//192.168.2.55/ADMIN$ Mapping: DENIED, Listing: N/A
//192.168.2.55/C$ Mapping: DENIED, Listing: N/A
```

Some hosts don't let your retrieve a share list. In these situations, it is still possible to perform a dictionary attack to guess share names. First we demonstrate the -S option failing:

```
$ enum4linux.pl -S 192.168.2.76
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Share Enumeration on 192.168.2.76 -----
[E] Can't list shares: NT_STATUS_ACCESS_DENIED

----- Attempting to map to shares on 192.168.2.76 -----
```

The output below show the use of the -s option with a dictionary file guess the names of some shares:

```
$ enum4linux.pl -s share-list.txt 192.168.2.76
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Session Check on 192.168.2.76 -----
[+] Server 192.168.2.76 allows sessions using username '', password ''

----- Brute Force Share Enumeration on 192.168.2.76 -----
c$ EXISTS
e$ EXISTS
admin$ EXISTS
ipc$ EXISTS, Allows access using username: '', password: ''
```

Getting OS information

The -o option gets OS information using smbclient. Certain versions of Windows (e.g. 2003) even return service pack information.

```
$ enum4linux.pl -o 192.168.2.76
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- OS information on 192.168.2.76 -----
[+] Got OS info for 192.168.2.76 from smbclient: Domain=[PTT] OS=[Windows 5.1] Se
[E] Can't get OS info with srvinfo: NT_STATUS_ACCESS_DENIED
```

Printer information

You can get some information about printers known to the remote device with the -i option. I don't know why you'd want to do this. I only implemented it because I could. 🙂

```
$ enum4linux.pl -i 192.168.2.69
```

```
Starting enum4linux v0.8.2 ( https://labs.portcullis.co.uk/application/enum4linux

----- Getting printer info for 192.168.2.69 -----
flags:[0x800000]
name:[\\192.168.2.69\SharedFax]
description:[\\192.168.2.69\SharedFax,Microsoft Shared Fax Driver,]
comment:[]
```

enum4linux-0.8.9.tar.gz

April 26, 2013

31.2 KiB
MD5 hash: d1873cdce2db870a7b9e92cbedbfb603
DETAILS

enum4linux-0.8.8.tar.gz

April 26, 2013

17.4 KiB
MD5 hash: 6a417bf56c69cc5062724963ac7e65d5
DETAILS

enum4linux-0.8.7.tar.gz

April 26, 2013

17.3 KiB
MD5 hash: 11ec51ead78e01e18ad5611f5ba5a8fc
DETAILS

enum4linux-0.8.6.tar.gz

April 26, 2013

17.2 KiB
MD5 hash: 7d76d40fe668a83b9029db2120aa13ad
DETAILS

enum4linux-0.8.5.tar.gz

April 26, 2013

16.7 KiB
MD5 hash: 8e13328c502e8a0834bf91c203471b96
DETAILS

enum4linux-0.8.4.tar.gz

April 26, 2013

16.5 KiB
MD5 hash: 40e99278d5025b69a947aaa182761833
DETAILS