Brutus - Introduction & Overview Jan 28th 2000

What does Brutus do?

In simple terms, Brutus is an online or remote password cracker. More specifically it is a remote interactive authentication agent. Brutus is used to recover valid access tokens (usually a username and password) for a given target system. Examples of a supported target system might be an FTP server, a password protected web page, a router console a POP3 server etc. It is used primarily in two contexts :

- To obtain the valid access tokens for a particular user on a particular target.
- To obtain any valid access tokens on a particular target where only target penetration is required.

What is a target?

Well that depends on you. As far as Brutus is concerned a target is a remote system and possibly a remote user on a remote system, there is more. To engage any given target we require an attack method, generally we only perform one type of remote attack - that is we attempt to positivley authenticate with the target by using a number of access token combinations. A target may provide no available attack methods, it may provide one or it may provide several.

What is an attack method?

In the context of Brutus, it is a service provided by the target that allows a remote client to authenticate against the target using client supplied credentials. For instance a UNIX server sat on a network somewhere may be offering Telnet and FTP services to remote users. Both telnet and FTP require the remote user to authenticate themselves before access is granted. For both these services the required credentials are usually a username and a password, therefore we have two available attack methods : FTP or Telnet. Some target systems will provide no opportunity for attack (at least not a remote authentication attack), perhaps they offer no remote services, perhaps they only offer anonymnous remote services (that require no authentication) or perhaps they offer authenticated remote services but use mechanisms to prevent authentication attacks such as account lockout or one time passwords of some sort.

Which attack method is best?

Again, that depends on some factors which may include :

- Is the target service available to any remote system? (*Yes is good)*
- Does the target service require a single token (e.g. just a password) or multiple tokens (e.g. Username & password & domain?) *(Single tends to be easier)*
- Does the target service feature account lockouts or large delays before returning the result of the authentication attempt? *(Yes is bad)*
- Does the target service allow us to maintain a persistant connection? *(Yes is good)*
- Is the service supported by Brutus, if not can it be defined? *(Yes is essential)*
- Will a positive authentication against the service actually be useful for the overall objective? *(Yes helps)*

Basically, the fastest most reliable attack method is always the one to choose if you have a choice. Generally trouble free methods include HTTP (Basic Auth) which is pretty fast, does not include lockouts or authentication delays - however the results may not be much use as often HTTP (Basic Auth) account information is separate from system account databases. The fastest remote service I have found to date is NetBus! Not only is it incredibly quick to authenticate against but a successful password aquisition will yield extreme target penetration.

I still don't get it, what does it do?

Find some service where you need to enter your username and password to gain access, type in a username and password and see what happens, then do it again, and again, and again, and again until you gain access and are positivley authenticated or until you get bored. Pretty straightforward really.

What the hell are you on about?

Ah, it's a game of two halves....

Whatever, can i get pr0n with it?

Probably...

I don't need this Brutus junk

You know where the recycle bin is.

Are you some sort of assh0le?
Yes, why? You want to make something of it sunshine?

Brutus functionality - a brief tour around the application

The Main Brutus Window
This is the screen that is displayed when you first start Brutus. You will see the screen is divided into sections detailed below :

- Menu Bar - Here you can load/save sessions and services, you can also access the word list tools screen.
- Targetting Information - Here you can select the address of the target system and the target system type (a.k.a. attack method.)
- Connection Options - Here you can select the maximum number of simultaneous connections to make to the target system, be careful here - more is not necessarily better. You can also specify the TCP port number of the target system, the connection timeout (in seconds) and any SOCKS proxy that you may wish to use.
- Type Specific Options - This section will change depending on what type of service is currently selected (see Targetting Information section.) Usually the 'Define Sequence' button allows you to tweak the authentication sequence, there are also other options.
- Authentication Options - Here you can choose to use a username and password or just a password. You may choose a single, specified username or a user list file. With respect to the password component, you can choose a word list file, a combo (username & password) list file or brute force passwords (in which case you can specify the keyspace that you wish to try.)
- Positive Authentication Results - A list of any positive authentication results that have been obtained.
- Log Window & Status Indicators - The log window displays any messages that Brutus thinks you should know about. Underneath the log window you will see a progress bar indicator which unsuprisingly indicates the progress of the attack. To the immediate right of the progress bar you will see a number of greyed out error indicators, these will flash red when an error of the stated type occurs. For example if the 'reject' indicator is red it means that the target is rejecting connections, either because it is not listening on the specified target port or because Brutus is hitting the target with more simultaneous connections than the target can handle. In general, other than the 'Quick Kill' indicator, it is better that these indicators do not flash.

Wordlist Generation/Tools Window
This screen is available by clicking on 'Wordlist Generation' under the 'Tools' menu on the main screen. The following actions are available to you from the 'action' drop down list :

- Convert List (LF > CRLF) - This is very handy for converting UNIX style text files to DOS style textfiles. In many word lists that you may find the line break is indicated using a single LF character as opposed to the DOS standard CRLF character pair.
- Only Word Length - This will copy the input file to the output file removing any words that do not fit the specifed word length parameters.
- Remove Duplicates - This will remove all duplicate entries from the word list (a common side effect of running permutations on a word list) - THIS IS DISABLED IN BRUTUS AET2!!
- Permutations - This will copy the input file to the output file running a set of permutations on each word as it is copied, typically turning one word into around 50+ variants (remember that when you're sat scratching your head whilst trying to permutate a 50Mb word file.)
- Create New List - This will take any seed words you provide and create a new word list from scratch after applying the selected permutations.
- Create New List for User - As above however, this will create a combo-list where both the username and the password are specified on each line. The username and any seed words you specify will be used to create the list.
- Create New List for Users - As above however, this is for multiple users. Instead of specifying a single username, the input file will be a standard user list file.

As of Brutus AET2 these routines have not been optimised however they aren't too slow either.

Proxy Definition Window
This screen is available from the 'Define' button in the connection options section from the main window.
It speaks for itself, basically 3 SOCKS versions are supported, optional proxy authentication is supported. HTTP proxies are pretty straightforward and will be added at some point soon.

HTML Form Authentication Definition Window

This screen is available from the 'Define Sequence' button in the HTTP (Form) options section from the main window. Here you can define the form structure to Brutus of any given HTML form. This will include the various form fields, any cookies to be submitted in requests, the HTTP referer field to send (if any) and of course the authentication response strings that Brutus uses to determine the outcome of an authentication attempt. As with other authentication types there are two response strings available; none, either or both of which may indicate positive or negative authentication results. At the top of this screen is an edit box labelled 'Target Form' and a button marked 'Learn form settings', you may enter the URL of the target HTML form in here and Brutus will attempt to fetch and interpret the form.

HTML Form Viewer Window
This screen is available from the 'Learn form settings' button on the HTML Form Authentication Definition Window. If Brutus can successfully read forms of the fetched HTML page (no frames, please direct Brutus at the individual framsets!) then each form will be interpreted and the relevant fields for each form will be displayed. Any cookies received during the request will also be logged here. Simply mark the relevant user and password fields of the form (i.e. the fields that correspond to the username and password editboxes on the HTML form) and hit 'Accept.' You can edit these values once returned to the previous window.

Authentication Sequence Definition Window
This screen is available from the 'Define Sequence' button in the Type options (except where the type is an HTML Form in which case the HTML Form Authentication Definition Window is displayed, or where the type is HTTP Basic in which case this button will be inaccessible.) This window is core to designing new authentication types for use with Brutus. Brutus handles each authentication attempt as a series of stages, as each stage is completed the authentication attempt is progressed until either a positive or negative aithentication result is returned at which point Brutus can either disconnect and retry or loop back to some stage within the authentication sequence. It is possible to view the authentication sequence by hitting the 'View' button which may make things clearer.

Brute Force Generation Window
Again, this window is self explanatory. It is used for defining the keyspace range that Brutus will use to generate passwords whilst in Brute force mode. Either choose one of the predefined ranges or define your own keyspace using the 'Custom range' option. Note the order of the characters can be important, by default they are arranged in order of letter frequency within written English. Guess what the min. length and max. length parameters mean? Don't get carried away with brute forcing, for an example try selecting 'Full Keyspace' with 14 characters maximum length and then engage the target with Brutus. You will see that Brutus has calculated a total of 6,158,335,059,490,089,995 attempts! Check the estimated completion time in the bottom right of the main screen - yes by the time it's finished insects will rule the Earth (apparently it will be the Bees although my money is on the Ants.)

Using Brutus - very briefly

Brutus CAN work very well, very fast if used correctly. Brutus CAN also sit there doing very little, it CAN sit there APPEARING to do lots whilst actually doing nothing. It's all in the authentication sequence, you have to get it right and it is not very forgiving. I intend to change this by providing enhanced protocol learning functions and incorporating a 'trace mode' which will permit viewing and debugging of network exchanges between Brutus and the target. What I have found very useful is to use a network sniffer (personally I use NetMon) to monitor the TCP traffic generated by Brutus and the target, this provides an invaluable insight into what the application protocol is actually doing, in many ways I think it is actually better than just reading the RFC for a given service. Another useful tool is telnet or netcat, use it to manually authenticate with the target and see for your own eyes what that swine target is trying to tell your pal and mine, Brutus.

Brutus does very weak target verification before starting, in fact all it does is connect to the target on the specified port, thats all. It is a good idea to manually check your target before you spend three days trying to bruteforce and anonymous FTP server. Also, I've said it before and I'll say it again - use positive responses in your authentication response strings if you can, you are far less likely to get false positives. However the trade-off is that Brutus is less likely to detect an error in the respose from the server (i.e. lockout).

Even if your authentication sequence is perfect Brutus AET2 is a test release with very little testing (thats why it's a test release.) If it won't seem to do what you want and you don't know why, drop me a copy of the BAD file and I can perhaps check it out (no promises though.)

The GUI in Brutus AET2 was never meant to support what it is currently supporting, I had only ever intended to use this as a test container. Consequently you may notice where I have SQUEEZED stuff to fit into a small space, you also may notice GUI state inconsistancies. Watch out for them and let me know if you find any

It almost goes without saying that Brutus is ONLY for use in situations where the target system administrator/custodian has AUTHORISED the action. Many target systems will log authentication failures and consequently will log any attempted engagement with Brutus.

Enjoy

www.hoobie.net/brutus
brutus@hoobie.net