**WHAT IS A FUZZER?**

A *Fuzzer* is a tool used by security professionals to provide invalid and unexpected data to the inputs of a program. A typical Fuzzer tests an application for [buffer overflow](http://en.wikipedia.org/wiki/Buffer_overflow), invalid format strings, directory traversal attacks, command execution vulnerabilities, SQL Injection, XSS, and more.

Because the Metasploit Framework provides a very complete set of libraries to security professionals for many network protocols and data manipulations, it is a good candidate for quick development of a simple fuzzer.

Metasploit’s Rex Library

The Rex::Text module provides lots of handy methods for dealing with text like:

* [Buffer](http://en.wikipedia.org/wiki/Data_buffer) conversion
* Encoding (html, url, etc)
* Checksumming
* Random string generation

The last point is extremely helpful in writing a simple fuzzer. This will help you writing fuzzer tools such as a simple URL Fuzzer or full Network Fuzzer.

For more information about Rex, please refer to the [Rex API documentation](http://www.rubydoc.info/github/rapid7/metasploit-framework/Rex/Text).

Here are some of the functions that you can find in Rex::Text :

root@kali:~# grep "def self.rand" /usr/share/metasploit-framework/lib/rex/text.rb

def self.rand\_char(bad, chars = AllChars)

def self.rand\_base(len, bad, \*foo)

def self.rand\_text(len, bad='', chars = AllChars)

def self.rand\_text\_alpha(len, bad='')

def self.rand\_text\_alpha\_lower(len, bad='')

def self.rand\_text\_alpha\_upper(len, bad='')

def self.rand\_text\_alphanumeric(len, bad='')

def self.rand\_text\_numeric(len, bad='')

def self.rand\_text\_english(len, bad='')

def self.rand\_text\_highascii(len, bad='')

def self.randomize\_space(str)

def self.rand\_hostname

def self.rand\_state()