Metasploit Mixins and Plugins

A Quick Diversion into Ruby

- Every *Class* only has one parent
- A class may include many *Modules*
- Modules can add new *methods*
- Modules can overload old methods
- Metasploit modules inherit *Msf::Module* and include mixins to add features.

Ruby : Mixins and Plugins | Metasploit Unleashed

Metasploit Mixins

Mixins are quite simply, the reason why Ruby rocks.

- Mixins *include* one class into another
- This is both different and similar to inheritance
- Mixins can override a class' methods

Mixins can add new features and allows modules to have different 'flavors'.

- Protocol-specific (HTTP, SMB)
- Behaviour-specific (brute force)
- *connect()* is implemented by the TCP mixin
- *connect()* is then overloaded by FTP, SMB, and others

Mixins can change behavior.

- The Scanner mixin overloads *run()*
- Scanner changes *run()* for *run_host()* and *run_range()*
- It calls these in parallel based on the THREADS setting

- The *BruteForce* mixin is similar

Metasploit Plugins

Plugins work directly with the API.

- They manipulate the framework as a whole
- Plugins hook into the event subsystem
- They automate specific tasks that would be tedious to do manually

Plugins only work in the msfconsole.

- Plugins can add new console commands
- They extend the overall Framework functionality

```
class MyParent
     def woof
          puts “woof!”
     end
end

class MyClass > MyParent
end

object = MyClass.new
object.woof() => “woof!”

================================================================

module MyMixin
     def woof
          puts “hijacked the woof method!”
     end
end

class MyBetterClass > MyClass
     include MyMixin
end
```