Useful API Calls a11y.text Useful API Calls We will cover some common API calls for scripting the Meterpreter and write a script using some of these API calls. For further API calls and examples, look at the Command Dispacher code and the REX documentation that was mentioned earlier. For this, it is easiest for us to use the irb shell which can be used to run API calls directly and see what is returned by these calls. We get into the irb by running the irb command from the Meterpreter shell.

```
meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds the meterpreter client
```

>> We will start with calls for gathering information on the target. Letâ€™s get the machine name of the target host. The API call for this is client.sys.config.sysinfo >> client.sys.config.sysinfo

=> {"OS"=>"Windows XP (Build 2600, Service Pack 3).", "Computer"=>"WINXPVM01"}

>> As we can see in irb, a series of values were returned. If we want to know the type of values returned, we can use the class object to learn what is returned: >> client.sys.config.sysinfo.class

=> Hash

>> We can see that we got a hash, so we can call elements of this hash through its key. Letâ€™s say we want the OS version only: >> client.sys.config.sysinfo['OS']

=> "Windows XP (Build 2600, Service Pack 3)."

>> Now letâ€™s get the credentials under which the payload is running. For this, we use the client.sys.config.getuid API call: >> client.sys.config.getuid

=> "WINXPVM01\labuser"

>> To get the process ID under which the session is running, we use the client.sys.process.getpid call which can be used for determining what process the session is running under: >> client.sys.process.getpid

=> 684 We can use API calls under client.sys.net to gather information about the network configuration and environment in the target host. To get a list of interfaces and their configuration we use the API call client.net.config.interfaces : >> client.net.config.interfaces

```
=> [#, #]
>> client.net.config.interfaces.class
```

=> Array As we can see it returns an array of objects that are of type Rex::Post::Meterpreter::Extensions::Stdapi::Net::Interface that represents each of the interfaces. We can iterate through this array of objects and get what is called a pretty output of each one of the interfaces like this: >> interfaces = client.net.config.interfaces

```
 => [#, #]
 >> interfaces.each do |i|
 ?> puts i.pretty
 >> end
 MS TCP Loopback interface
 Hardware MAC: 00:00:00:00:00:00
 IP Address  : 127.0.0.1
 Netmask     : 255.0.0.0


 AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
 Hardware MAC: 00:0c:29:dc:aa:e4
 IP Address  : 192.168.1.104
 Netmask     : 255.255.255.0
```

Next Useful Functions Prev Custom Scripting