Exploit module Format a11y.text Exploit module Format Formatting our Exploit Module a11y.text Formatting our Exploit Module The format of an Exploit Module in Metasploit is similar to that of an Auxiliary Module but there are more fields. There is always a Payload Information Block . An Exploit without a Payload is simply an Auxiliary Module. A listing of available Targets is outlined. Instead of defining run() , exploit() and check() are used. Exploit Module Skeleton a11y.text Exploit Module Skeleton class Metasploit3 > Msf::Exploit::Remote

```
include Msf::Exploit::Remote::TCP

def initialize
    super(
        'Name'          => 'Simplified Exploit Module',
        'Description'   => 'This module sends a payload',
        'Author'        => 'My Name Here',
        'Payload'       => {'Space' => 1024, 'BadChars' => â€œ\x00â€•},
        'Targets'       => [ ['Automatic', {} ] ],
        'Platform'      => 'win',
    )
    register_options( [
        Opt::RPORT(12345)
    ], self.class)
end

# Connect to port, send the payload, handle it, disconnect
def exploit
    connect()
```

```
        sock.put(payload.encoded)
        handler()
        disconnect()
    end
```

end Defining anÂ Exploit Check a11y.text Defining anÂ Exploit Check Although it is rarely implemented, a method called check() should be defined in your exploit modules whenever possible. The check() method verifies all options except for payloads. The purpose of doing the check is to determine if the target is vulnerable or not. Returns a defined Check value. The return values for check() are: CheckCode::Safe â€“ not exploitable CheckCode::Detected â€“ service detected CheckCode::Appears â€“ vulnerable version CheckCode::Vulnerable â€“ confirmed CheckCode::Unsupported â€“ check is not supported for this module. proftp banner module | Metasploit unleashed Banner Grabbing : Sample check() Method a11y.text Banner Grabbing : Sample check() Method def check

```
    # connect to get the FTP banner
    connect

    # grab banner
    banner = banner = sock.get_once

    # disconnect since have cached it as self.banner
    disconnect

    case banner
        when /Serv-U FTP Server v4\.1/
            print_status('Found version 4.1.0.3, exploitable')
            return Exploit::CheckCode::Vulnerable
```

```
        when /Serv-U FTP Server/
            print_status('Found an unknown version, try it!');
            return Exploit::CheckCode::Detected

        else
            print_status('We could not recognize the server banner')
            return Exploit::CheckCode::Safe
    end

    return Exploit::CheckCode::Safe
end
```
Next Exploit Mixins Prev Exploit Development Goals