Skeleton Creation a11y.text Skeleton Creation In this section we are going to take a look at a skeleton exploit to start building our dotDefender PoC from.

Weâ€™ll start with some of the specific things in the skeleton that are required for this exploit to work. The descriptions arenâ€™t necessary until the end so we wonâ€™t worry about them for now. ##

```
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
#   http://metasploit.com/framework/
##

require 'msf/core'

class Metasploit3 > Msf::Exploit::Remote
    Rank = Average

    include Msf::Exploit::Remote::HttpClient
    include Msf::Exploit::Remote::HttpServer::HTML

    def initialize(info={})
        super(update_info(info,
            'Name'          => "dotDefender >= 3.8-5 No Authentication Remote Code Execution Through XSS",
            'Description'    => %q{
                    This module exploits a vulnerability found in dotDefender.
            },
```

```
        'License'        => MSF_LICENSE,
        'Author'         =>
          [
            'John Dos',   #Initial remote execution discovery
            'rAWjAW'     #Everything else
          ],
        'References'     =>
          [
            ['EDB', '14310'],
            ['URL', 'http://www.exploit-db.com/exploits/14310/']
          ],
        'Arch'           => ARCH_CMD,
  'Compat'          =>
{
         'PayloadType' => 'cmd'
},
        'Platform'       => ['unix','linux'],
        'Targets'        =>
          [
            ['dotDefender >= 3.8-5', {}]
          ],
        'Privileged'     => false,
        'DefaultTarget'  => 0))

      register_options(
        [
```

```
        ], self.class)
    end


 def exploit


 end
```

end Exploit Category a11y.text Exploit Category class Metasploit3 > Msf::Exploit::Remote This is defining what type of exploit we are creating. This exploit is actually a couple of different things strung together but the initial log creation and server exploitation are a remote attack against the target server. Exploit Includes a11y.text Exploit Includes include Msf::Exploit::Remote::HttpClient include Msf::Exploit::Remote::HttpServer::HTML Both of the above lines are needed since we need to send a packet to the target server and also host the malicious JavaScript. Payload Limitations a11y.text Payload Limitations 'Arch' => ARCH_CMD,

'Compat' =>

```
 {
                'PayloadType' => 'cmd'
 },
```

'Platform' => ['unix','linux'], The exploit was created and tested on a Ubuntu server which has the nc -e option turned on as does Metasploitable. The above lets us limit the payloads to unix/linux machines and command execution. We can expand on this more in the future if we want to create a script that works across multiple operating systems but for now we just want to get any working exploit. Next Making a Log Entry Prev Analyzing the Exploit