**Jigyas Sharma**

**Dr. Alexandru Bardas**

**EECS 765**

**29 September 2023**

**Programming Assignment 2: Report**

**Introduction**

In this assignment, we look at a buffer overflow vulnerability in an application called “WinAmp”. The aim of the assignment is to prove that even though the application does not require any network privileges, it is possible to exploit the application in a way to achieve a reverse shell over the network. Furthermore, during this development we learn about using gadgets to tackle the implementation of Guard Stacks which were put to avoid buffer overflows.

**Running the Exploit**

***Exploit Generation File: winampBOE(Final Exploit).pl***

***Exploit File: mcvcore.maki***

***LHOST: 127.0.0.1***

***LPORT: 5555***

**Step 1:** Run the exploit generation file and pipe the output to “mcvcore.maki”. Use the command “*perl winampBOE(Final Exploit).pl > mcvcore.maki*”.

**Step 2:** Place the “mcvcore.maki” file in the WinAmp skins folder under Bento.

**Step 3:** Open a windows admin shell and listen on port 5555. Use the command “nc -nvlp 5555”. The shell would return something along the lines of “Listening on any [5555]”.

**Step 4:** Start the WinAmp application and load the bento skin.

**Step 5:** Upon loading the skin, the listen port receives a shell over TCP.

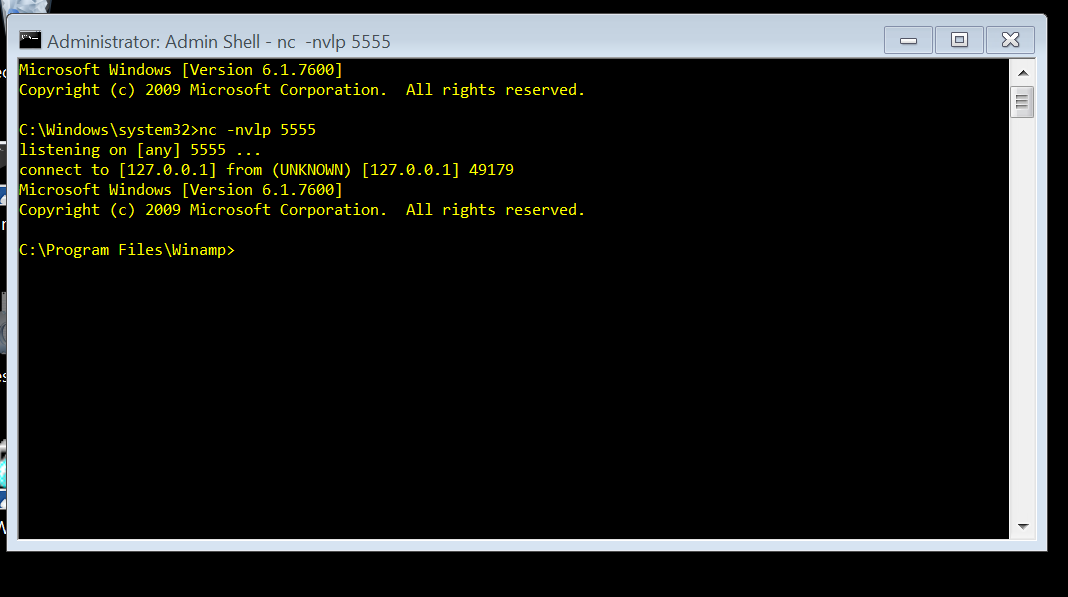


Figure 1: Bento skin returns shell

**Developing the Exploit**

**Structure of the Input**

**Determining the Parameters used in the Malicious Input**

***SEH CHAIN Next Record: 16760***

***Address to Exception Handler: 16756***

***Architecture: Little Endian***

***Address of POP POP RET: 7c373b7f***

***Shell Code: “shellcode.txt”***

**Generating Malicious Input**

**References**

*None Applicable*

**Collaborations**

**Ina Fendel :** Long Input for pattern\_create and pattern\_offset