SteamRipper32

VULNERABILITY REPORT

SUNDAY, MAY 02, 2021

MODIFICATIONS HISTORY

Version	Date	Author	Description
1.0	05/02/2021	Ritesh Puvvada	Initial Version

TABLE OF CONTENTS

GENERAL INFORMATION

SCOPE

VIT-P has mandated us to perform security tests on the following scope:

- A buffer overflow occurs when the size of information written to a memory location exceeds what it was allocated. This can cause data corruption, program crashes, or even the execution of malicious code.

ORGANISATION

The testing activities were performed between 05/31/2021 and 05/01/2021.

EXECUTIVE SUMMARY

VULNERABILITIES SUMMARY

Following vulnerabilities have been discovered:

Risk	ID	Vulnerability	Affected Scope
High	IDX-001	SteamRipper32	All over the system

TECHNICAL DETAILS

STEAMRIPPER32

CVSS SEVERITY	High		CVSSV3 SCORE	8.1
CVSSV3 CRITERIAS	Attack Vector :	**Local**	Scope :	**Changed**
	Attack Complexity :	**High**	Confidentiality :	**High**
	Required Privileges :	**None**	Integrity :	**High**
	User Interaction :	**None**	Availability :	**High**
AFFECTED SCOPE	All over the system			
DESCRIPTION	Buffers are memory storage regions that temporarily hold data while it is being transferred from one location to another. A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the data to the buffer overwrites adjacent memory locations. For example, a buffer for log-in credentials may be designed to expect username and password inputs of 8 bytes, so if a transaction involves an input of 10 bytes (that is, 2 bytes more than expected), the program may write the excess data past the buffer boundary. Buffer overflows can affect all types of software. They typically result from malformed inputs or failure to allocate enough space for the buffer. If the transaction overwrites executable code, it can cause the program to behave unpredictably and generate incorrect results, memory access errors, or crashes.			
OBSERVATION	A **buffer overflow**, or **buffer overrun**, occurs when more data is put into a fixed-length **buffer** than the **buffer** can handle. The extra information, which has to go somewhere, can **overflow** into adjacent memory space, corrupting or overwriting the data held in that space.			
TEST DETAILS				

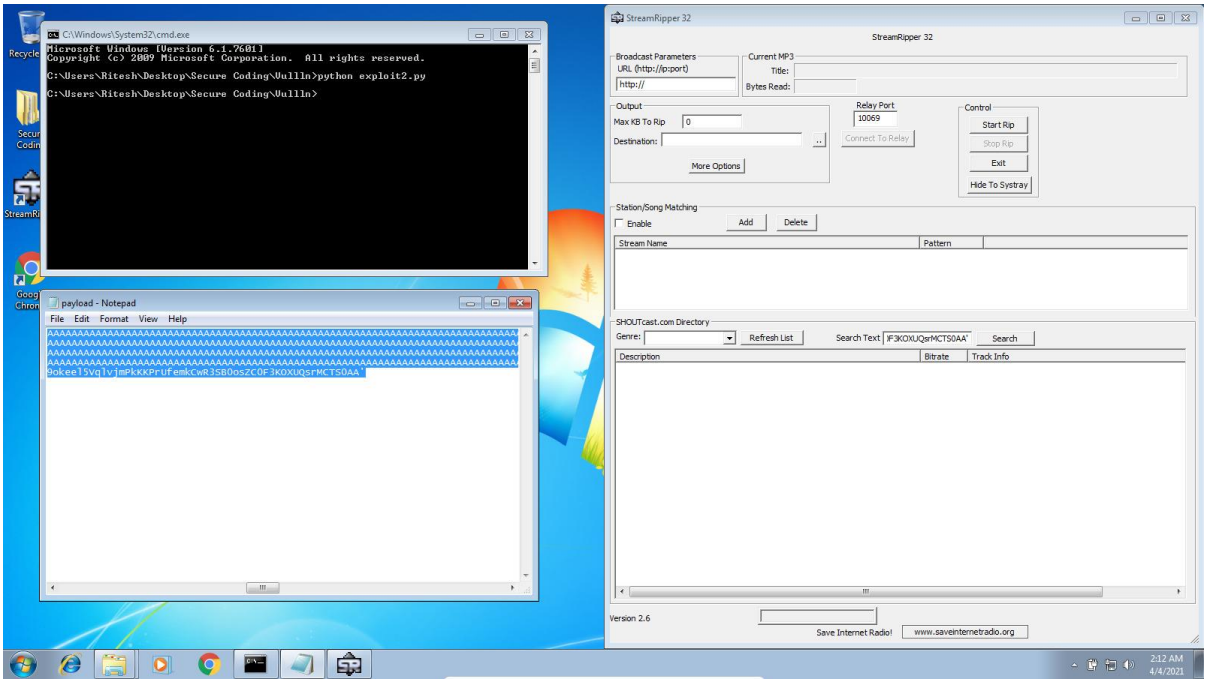

Image 1 – Capture1.PNG

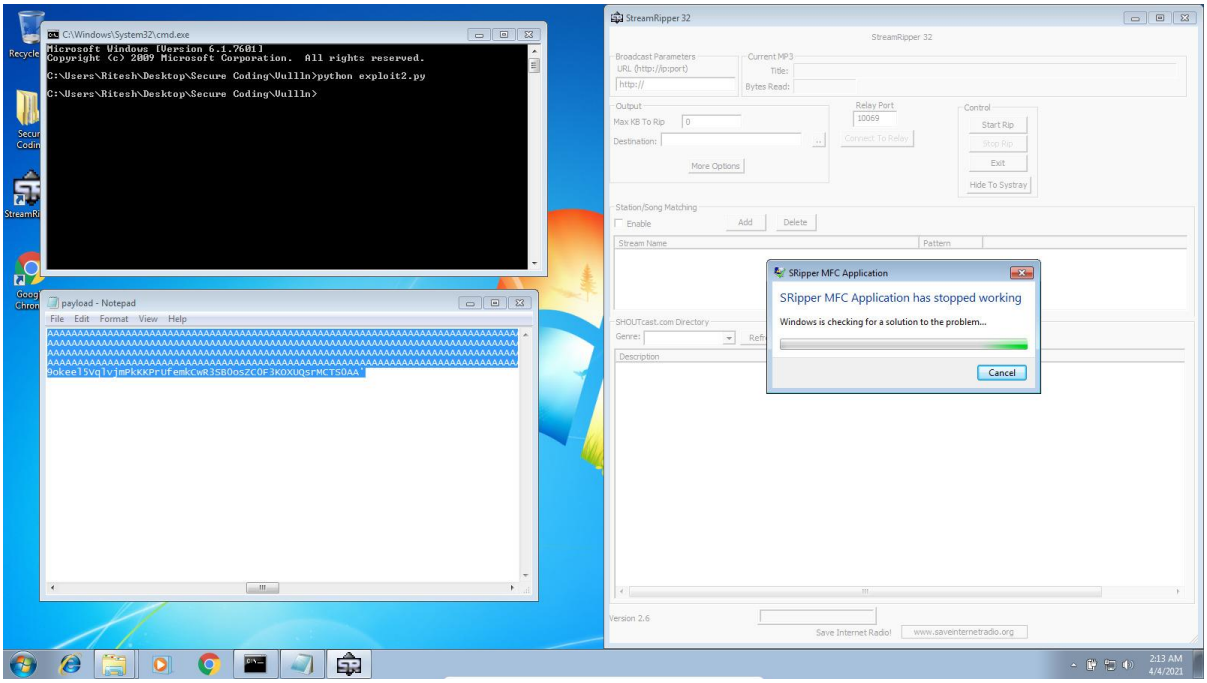

Image 2 – Capture2.PNG

REMEDIATION	• **Address space randomization (ASLR)**—randomly moves around the address space locations of data regions. Typically, buffer overflow attacks need to know the locality of executable code, and randomizing address spaces makes this virtually impossible. • **Data execution prevention**—flags certain areas of memory as non-executable or executable, which stops an attack from running code in a non-executable region. • **Structured exception handler overwrite protection (SEHOP)**—helps stop malicious code from attacking Structured Exception Handling (SEH), a built-in system for managing hardware and software exceptions. It thus prevents an attacker from being able to make use of the SEH overwrite exploitation technique. At a functional level, an SEH overwrite is achieved using a stack-based buffer overflow to overwrite an exception registration record, stored on a thread's stack. Security measures in code and operating system protection are not enough. When an organization discovers a buffer overflow vulnerability, it must react quickly to patch the affected software and make sure that users of the software can access the patch.
REFERENCES	https://owasp.org/www-community/vulnerabilities/Buffer_Overflow