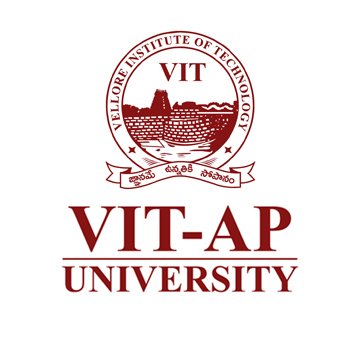
|  |
| --- |
| Buffer Overflow Vuln |
| Vulnerability Report |
| Thursday, June 10, 2021 |



modifications history

| **Version** | **Date** | **Author** | **Description** |
| --- | --- | --- | --- |
| 1.0 | 06/10/2021 | R B CH S TARUN - 19BCN7122 | Initial Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. General Information 4

1.1 Scope 4

1.2 Organisation 4

2. Executive Summary 5

3. Technical Details 6

3.1 title 6

4. Vulnerabilities summary 8

# General Information

## Scope

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

* Exploiting the memory vulnerabilities.

## Organisation

The testing activities were performed between 06/09/2021 and 06/11/2021.

# Executive Summary

# Vulnerabilities summary

Following vulnerabilities have been discovered:

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **ID** | **Vulnerability** | **Affected Scope** |
| Medium | VULN-001 | Memory Vuln |  |

# Technical Details

## Memory Vuln

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CVSS Severity | Medium | | CVSSv3 Score | | 6.1 |
| **CVSSv3 criterias** | Attack Vector : | **Physical** | Scope : | **Changed** | |
| Attack Complexity : | **High** | Confidentiality : | **Low** | |
| Required Privileges : | **Low** | Integrity : | **High** | |
| User Interaction : | **Required** | Availability : | **Low** | |
| **Affected scope** |  | | | | |
| **Description** | **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.  image  Image 1 – 2 it produced a output file.PNG  Here, this python script which containing the shell code, which over rides the memory locations and crashes the program and trigger the command prompt.  after generating the payload we injected the payload into stream ripper to test and find the vulnerability, as our test progresses we found that it has buffer overflow, which makes it crash the whole program with the payload we generated. | | | | |
| **Observation** | image  Image 1 – crashed.PNG | | | | |
| **Test details**  image  Image 1 – crashed.PNG | | | | | |
| **Remediation** | Developers can protect against buffer overflow vulnerabilities via security measures in their code, or by using languages that offer built-in protection.  In addition, modern operating systems have runtime protection. Three common protections are:   * **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. | | | | |
| **References** |  | | | | |