pratik yadav

Follow 347 Followers About

Ssrf to Read Local Files and Abusing the AWS metadata

pratik yadav Apr 21, 2019 · 4 min read

Hello Guys ,

I am Pratik Yadav ,Currently working as Security Engineer in one Crypto Exchange platform:)

Well This not a English Grammer blog so Please ignore any grammatical mistakes. Previously I wrote a blog post about the **payment bypass bug** which i founded on a Program and i Received lot of positive feedback so it motivated me to share one more findings with the community .If you haven't read that blog you can read it by following this link .

So lets talk about the bug :-

Firstly I crawled and manually performed all the operations as a user on that application . And after that i checked every possible request on burp http history and Frankly saying i was looking out for a url redirection Vulnerability . So i Searched on Burp of possible parameters of url= And ended up with a url which basically was something like this

my image was stored on aws bucket . As we have got a clear idea that it is loading content. Why not to load content from other domain ? Try RFI ?

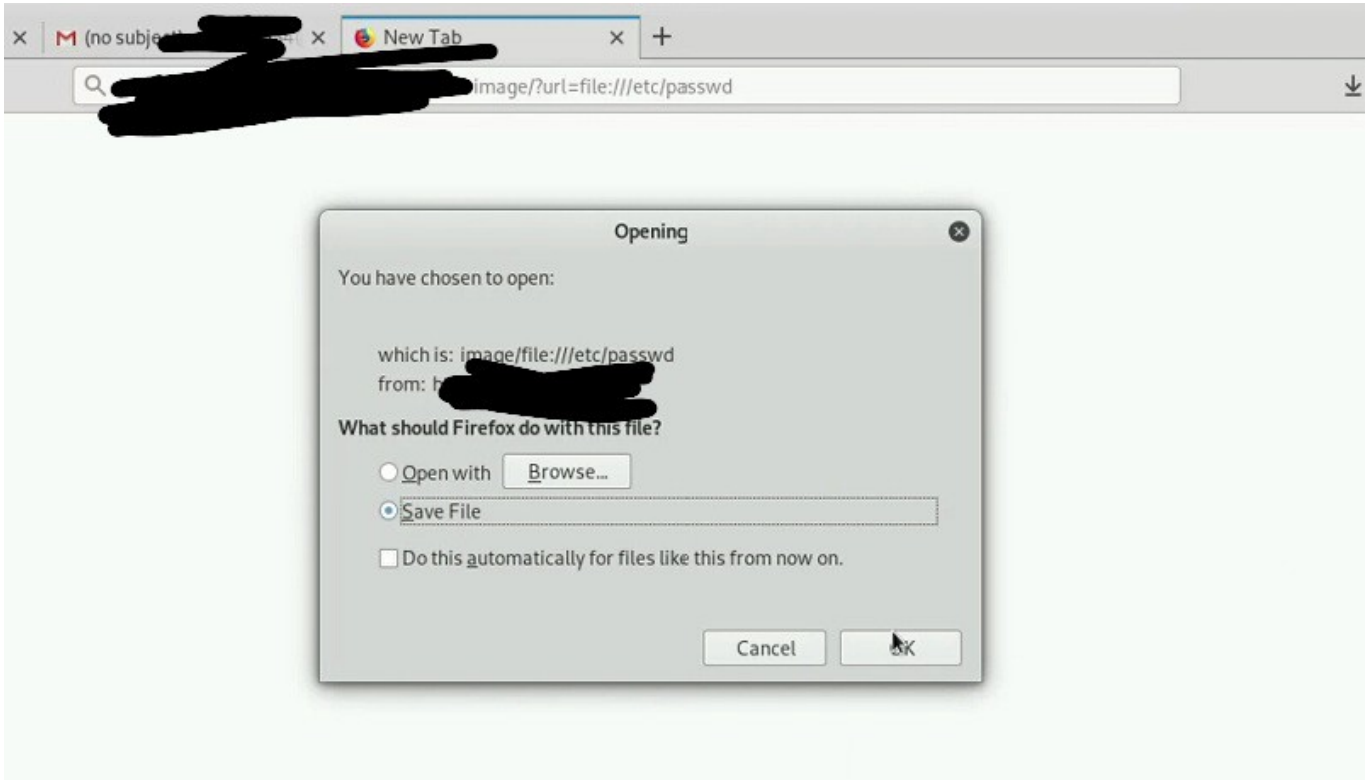
Steps that i followed:-

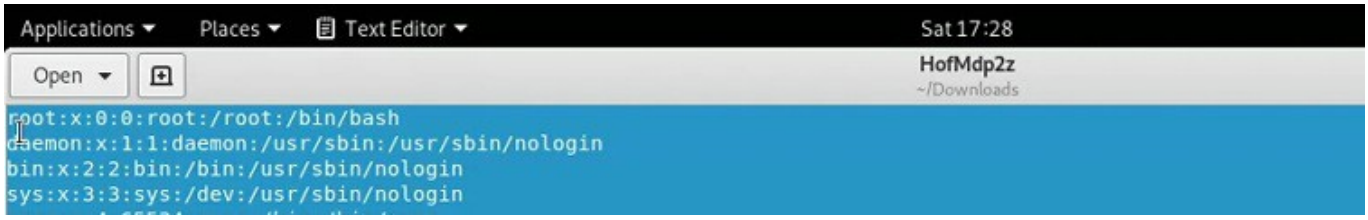
Trying Xss :)(Failed)

1. Firstly I tried to get a Xss .So i have read many blog about getting a xss . All I did is Simply added http://brutelogic.com.br/poc.svg in the url . So the Final crafted url was like https://example.com/viewimage/?url=http://brutelogic.com.br/poc.svg . **So I Visited the url but it was not loading the content but a simple text file gets downloaded And it was having nothing in it.**

Trying to Read Local Files (Success)

1. Next I tried URL schemas to read internal and make server perform actions (`file:///` , `dict://` , `ftp://` , `gopher://` ..) So the final crafted url was https://example.com/viewimage/?url=file:///etc/passwd
   And Again a Text File gets downloaded like

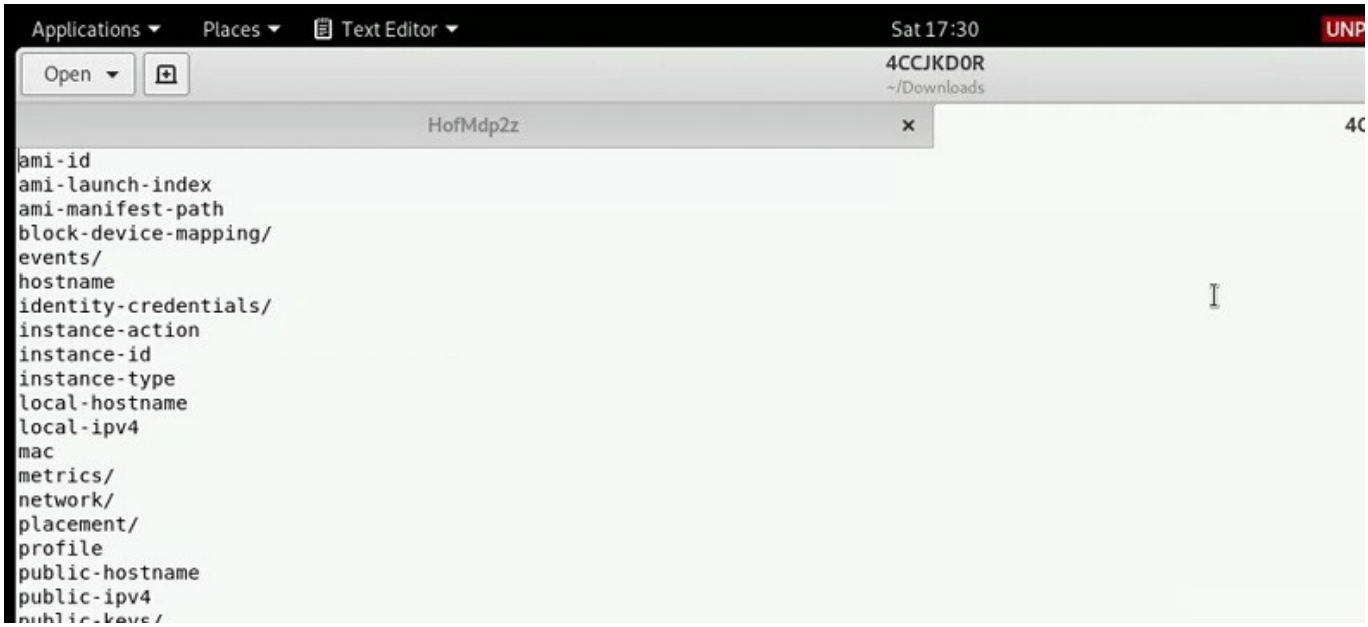




Escalating it More :)
Now as I have seen this site was loading images from aws . So i Thought why not to extract internal AWS metadata.

Reading Aws EC2 Metadata

I replaced the url parameter by To view all categories of instance metadata from within a running instance, use the following URI: `http://169.254.169.254/latest/meta-data` So the Final crafted url https://example.com/viewimage/?url=`http://169.254.169.254/latest/meta-data` Now as expected it again downloaded a Txt file and after viewing. it showed below information



4. Now Reading Secrets to Make it more critical :) to rain aws access key , secretaccess key simply i appended this url in the parameter in

And it rained the secret access key , token ,region ,etc. So now using aws client I can export this details and could have get access :)

Simply This bug allowed me to achieve RCE using a SSRF Vulnerability .

To Read More about escalating it After getting the secret key etc you can follow this blog https://www.notsosecure.com/exploiting-ssrf-in-aws-elastic-beanstalk/

"Secondly I would like to Thanks *ENCIPHERS Team for Training which i attended in Delhi And it was Beautifully conducted by Abhinav mishra , Narendra and abhishek Specially for Creating the Vulnerable application .*
And the SSRF challenge was same as that I founded in This private program"

Now what everyone looks for In Blog Post :)(Bounty)

1. Reported the vulnerability to the program via There BB Program
2. Within a Day I received the message that Bug was fixed and they rewarded Me with 4 Digit $ Bounty and also Bonus
3. But however they where not so happy with me because i escalated it more than enough just to demonstrate and they revoked the credentials :)

Thanks Hope you liked my blog post , Please do share this :)

Pratik Yadav

AWS Bug Bounty Infosecurity Infosec