

#BugBounty — Compromising User Account- ”How I was able to compromise user account via HTTP Parameter Pollution(HPP)”



Avinash Jain (@logicbomb_1)

Follow

Jul 7, 2018 · 3 min read

Hi Guys,

As the title suggests , this particular blog is about “How I was able to compromise user account exploiting HTTP Parameter Pollution (HPP) vulnerability. HPP—What, Why, How? and below is the short description—

HTTP Parameter Pollution, as implied by the name, pollutes the HTTP parameters of a web application in order to perform or achieve a specific malicious task/attack different from the intended behaviour of the web

application. Supplying multiple HTTP parameters with the same name may cause an application to interpret values in unanticipated ways.

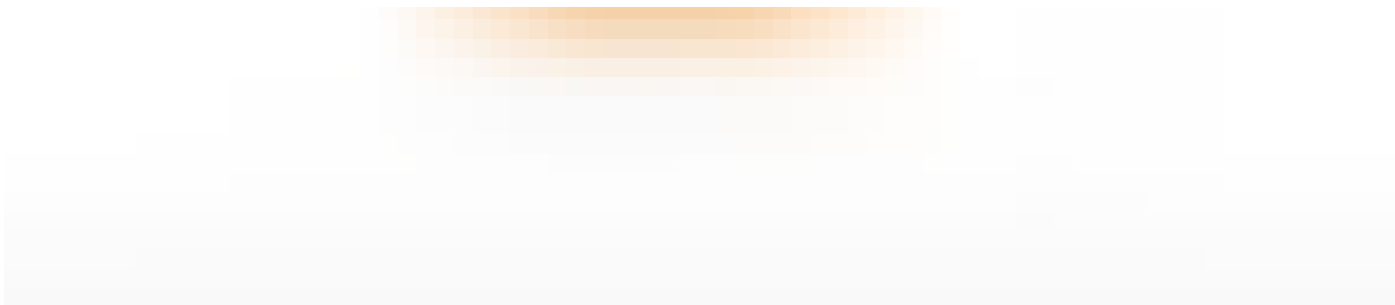
Recently, I have been reading a lot about this, some of the hackerone reports and how different application handles it, made it more clear for me. Targeting it, I found the same in one of the Indian Online Shopping Site and I then escalated it to “User Account Compromise”.Let’s see how I was able to do so—

Like every other shopping apps have, this website was also having the functionality to share the shortlisted shopping items on social media sites or we may call it social sharing button. Below is the redacted HTTP request for the same —

<https://www.redacted.com/zephyr-mini-alpha-board/p/&pid=ETYDBYSKDDZQGDJD&vi=XXXX>

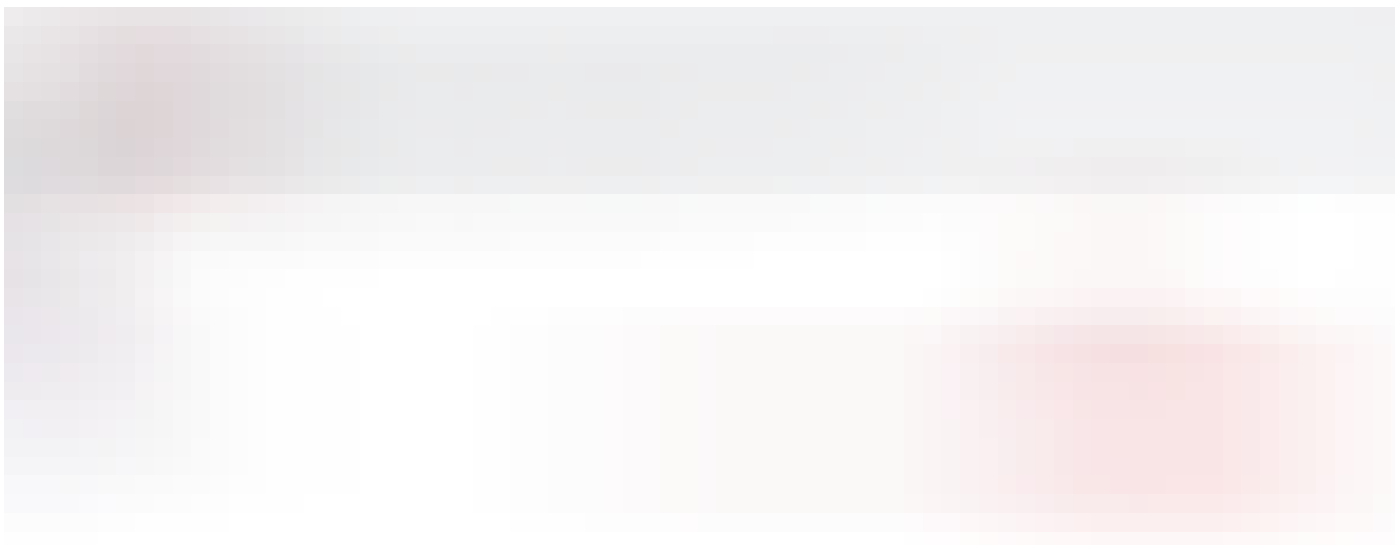
Here was a simple HPP vulnerability where adding the parameter ‘u’ (<https://www.redacted.com/zephyr-mini-alpha-board/p/?u=http://www.evill.com&pid=ETYDBYSKDDZQGDJD&vi=XXXX>) and sharing it on facebook will change the content to <https://www.facebook.com/sharer.php?>

u=https://www.redacted.com/zephyr-mini-alpha board/p/?u-
http://www.evil.com&pid=ETYDBYSKDDZQGDJD&vi=XXXX and instead of sharing the shortlisted item, “**evil.com**” was getting shared on victim’s Facebook page. *Now as I knew that HPP occurs due to the way the different web servers and development frameworks handle multiple parameters hence it’s basically a backend application technology issue. So if it is occurring on social sharing button , it can be present in other part of the application as well.* In the search to exploit it more, I went to test login authentication page where there was a reset password functionality.



Reset Password Page

Before that, I analysed the backend application framework being used using “Wappalyzer” which shows that “JSP and Apache” were in use. Now, providing victim’s mail id and submitting it, triggers the below HTTP request-



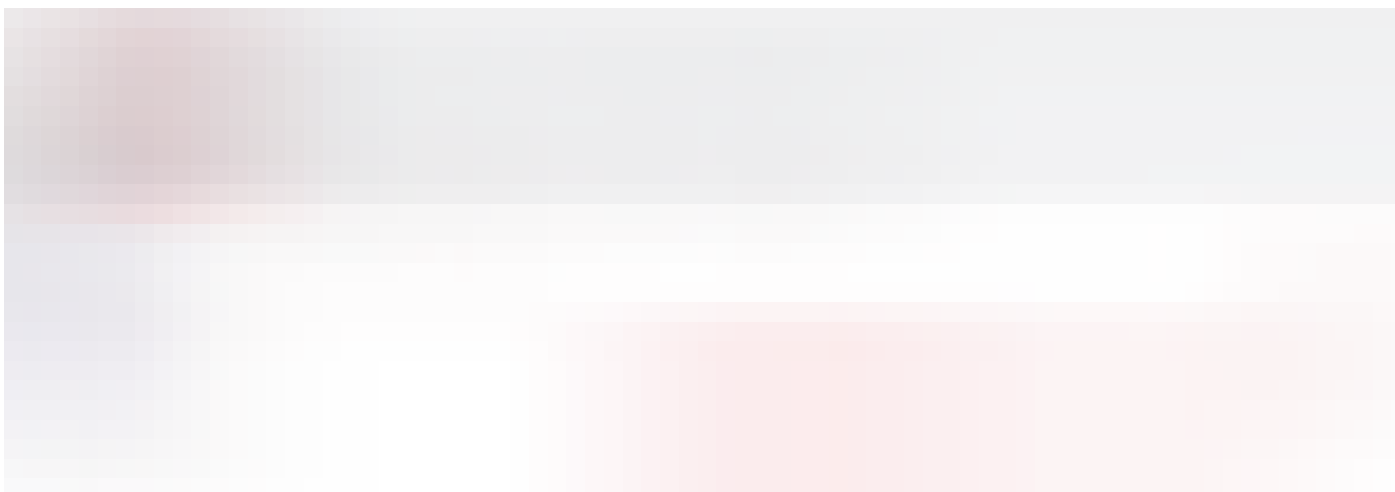
Reset Password HTTP Request

According to the functionality, it generated the reset password link taking the value as “harrysonito@gmail.com” and send it to the same mail id.

Now as I knew that HPP is existing in the framework and “JSP and Apache” were being used as backend application and in order to know more about this framework ,how they handle parameter pollution, this helped me-

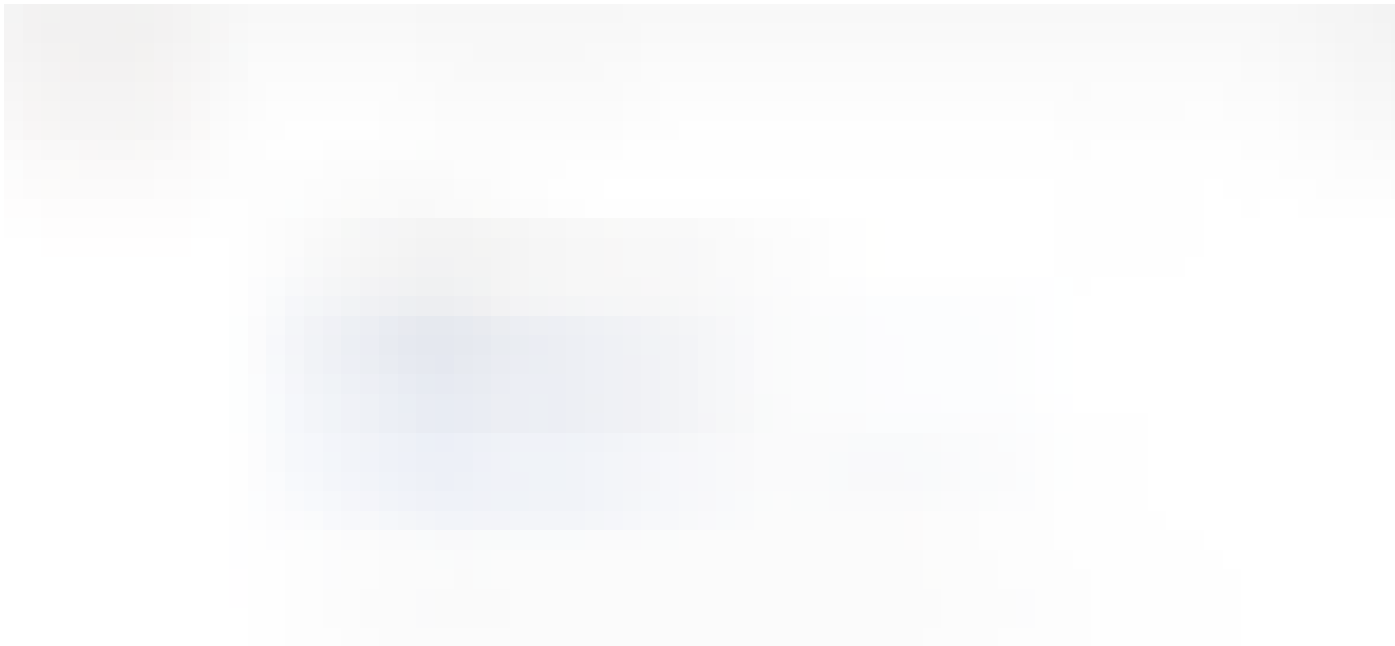
<https://www.acunetix.com/blog/whitepaper-http-parameter-pollution/>

and I went to check whether supplying multiple “email” parameters causes the backend application framework to work differently. I added attacker’s email id and the tampered request looks like-



Tampered HPP Reset Password Request

and as I was expecting , the backend application (JSP in this case)took the value of first “email” parameter to generate the password reset link and used the value supplied in the second “email” value to trigger the send mail to “petercheckk852234@gmail.com . Because of which “attacker” receive the link to reset password of “victims” account. :) and I then opened the link, set the password for victim account and able to login into the same to finally takeover the account and this is how I was able to compromise any user account via HTTP Parameter Pollution.



. . .

Thanks for reading!

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

Security

Hacking

Ethical Hacking

Penetration Testing

Bug Bounty

765 claps



4



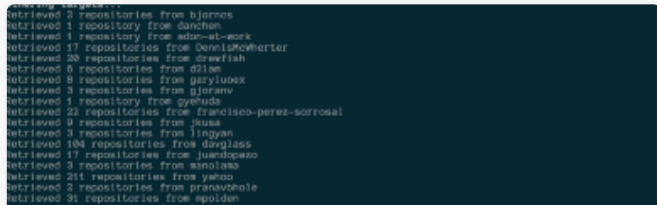
...



Avinash Jain (@logicbomb_1)

Follow

Lead Infrastructure Security Engineer @groferseng | DevSecops | Part time BugBounty Hunter | Acknowledged by Google, NASA, Yahoo, United Nations, BBC etc.



More from Avinash Jain (@logicbomb_1)

Credentials leaked in public? Here's what Grofers implemented to...

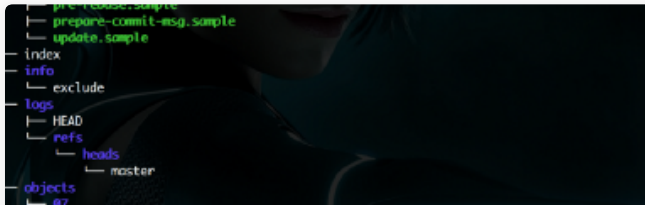


Avinash Jain (@log...

Nov 3, 2018 · 4 min re



1.3K



Related reads

Hidden directories and files as a source of sensitive information...



bl4de

Dec 17, 2018 · 15 min r



1.1K



Related reads

The BatchOverflow Bug and How to Catch All Bugs



Kiran Garimella


May 11, 2018 · 12 min r



211



Responses

 Write a response...

Show all responses