

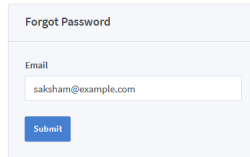
# CVE-2020-26163 BigBlueButton | Host Header Injection

Time to read: 3 min | 25 May, 2020 | Saksham Anand

Back in April, one of the systems I was testing was a video conferencing application, known as [BigBlueButton](#), an open source challenger to Zoom.

The BigBlueButton installation comes with a user friendly interface, known as [Greenlight](#), which ties in nicely with the BigBlueButton server. While most of the corporate installations would be using LDAP authentication, at times, installation will be based on standard username and password login mechanism, which is handled by Greenlight.

Part of the standard authentication process in Greenlight is the 'Forgot Password?' functionality. Which when clicked, prompts the user for their email and presents a submit button, as shown in the screenshot below:



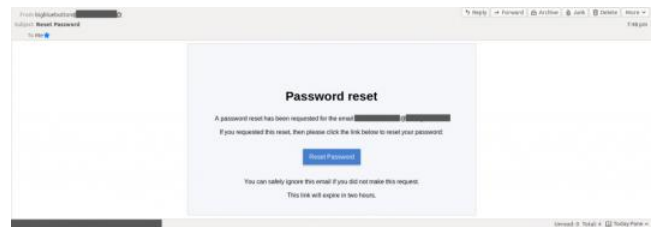
For the purpose of this demonstration 'saksham@example.com' is being used as the victim's email, however, in reality, an attacker will use the corporate email address of their victim (i.e. an executive in some company). When the form is populated and submitted, the requests can be captured through Burp Suite or through the network tab of the browser debugger. The captured request headers will show a range of different entries, including host header and origin header, as shown below:

```
Host: bigbluebutton.example.com
Origin: https://bigbluebutton.example.com
```

In a real world scenario, the domain in the header will be replaced with something that looks familiar with the existing domain name that a company has. For example if a company owns 'example.com', then an attacker might register 'example.co', to increase the likelihood of a successful phishing attack. For the purpose of this demonstration, the host and the origin headers were replaced with the following entries:

```
Host: www.sakshamanand.com
Origin: https://www.sakshamanand.com
```

The request was then released from the client and accepted by the server under the 302 response code. Shortly after which, an email appeared in the example inbox, as shown below:



When the blue 'Reset Password' button is clicked, the following link opens in the browser:

```
https://www.sakshamanand.com/gi/password_reset/TOKEN_REDACTED/edit
```

From here, two attacks are possible:

- The attacker can capture the victim's password reset token, from the link sent to the attack domain, and use it to create a random password. From which they can do damage that is contained within the BigBlueButton installation, however, this will lock the victim out of BigBlueButton and the attack may not last long.
- Or, the attacker can create a fake BigBlueButton password reset portal, something that looks exactly like the real one (combined with the almost similar domain name, example.co), then capture the new password that the victim picks. This would allow the attacker to continue using the vulnerable BigBlueButton account, without locking the victim out, and the attacker can try use the new password in other applications within the company (under the assumption that the victim might have used the same password elsewhere).

These attacks are generally possible in package based software, where the software is aimed at multiple consumers for self installation. The application was vulnerable to this issue as it was directly reading the headers without validating them. Generally, in order to mitigate against Host Header Injection attacks, a whitelist needs to be implemented and the code must check against that whitelist to sanitise incoming/outgoing HTTP requests.

This issue was reported to BigBlueButton in May 2020 and was promptly fixed by the project maintainers under [this pull request](#). A CVE for this issue was also released by MITRE on 30th September under <https://cve.mitre.org/cve-bin/cvename.cgi?name=CVE-2020-26163>

