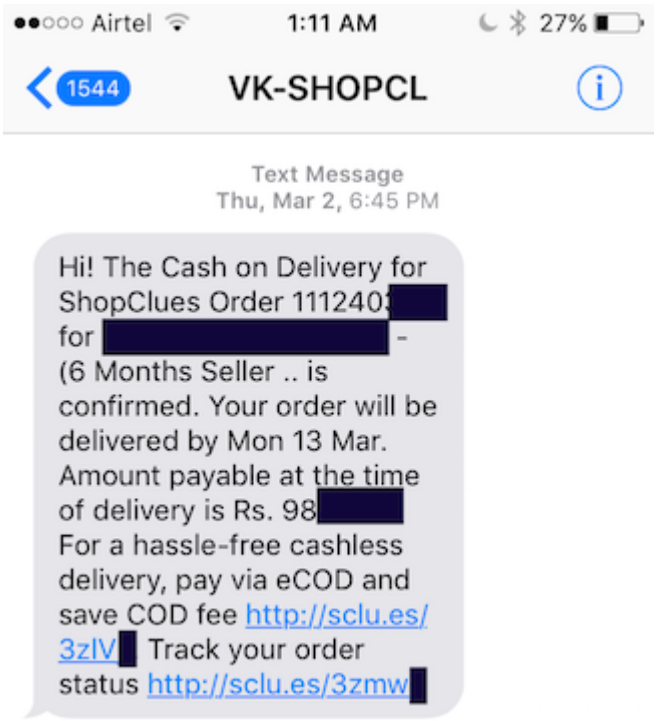
Shopclues user data leak via short links

Fallible Follow
May 25, 2017 · 3 min read

Shopclues is an Indian unicorn ecommerce marketplace. Every user who places an order on Shopclues receives a text message with a link to track the order.



The short links e.g. — http://sclu.es/3zlVa http://sclu.es/3zlV**a**, http://sclu.es/3zlV**b**, http://sclu.es/3zlV**c** were actually serially iterable, , .. etc are all valid links which reveals **name, phone numbers, email, home address, order details** of Shopclues users. A simple script to iterate over all of them can be written as follows:

```
$ for i in {a..z}; \
do for j in {a..z}; \
do for k in {a..z}; \
do curl "http://sclu.es/3z$i$j$k" -vvv; \
done; done ; done
```

shopclues.sh hosted with ❤ by **GitHub** **view raw**

you ask? Lets do a little math to understand this, if the code is 6 alphabets long (all caps), we have these many possibilities:

$$26^6 = 308,915,776$$

You may think that this is a huge number, no one can guess any random link. And that would be correct except that depending on how many links you have generated till now, the guessability of links goes on increasing. Lets say you are an E-commerce website which has generated 1 million short links. Now if you divide:

We reported this to Shopclues in the first week of March 2017. We received no acknowledgment email from them. A separate email sent to their cofounders did not we found it fixed on 3 May 2017. The URL now asks for email id & phone number to confirm that the request is made by the authorised user.

$$308,915,776/1,000,000 = 308.915$$

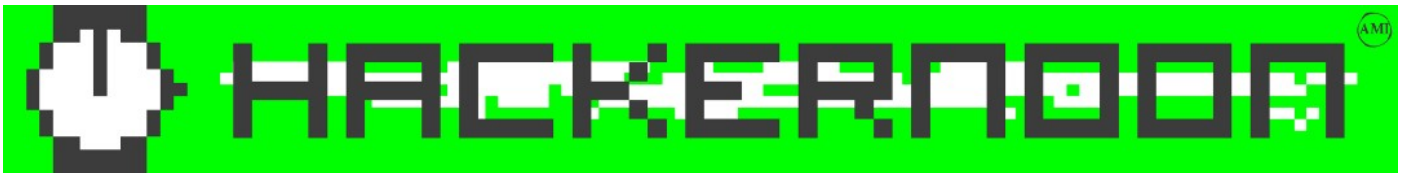
On an average, with every 308 wrong guesses one will be able to guess a correct link. So, if you decide to put user personal sensitive data behind short links which can be accessed by anyone without authentication, think again.

The vulnerability in case of Shopclues was even more severe as they didn't even randomize the short links (though it does look gibberish if you just look at a single link and not a bunch of links together), they have merely incremented it serially.

Do something to increase the sample space of non-working short links — make the sample space huge that it makes the guessing (almost) impossible.

Try a combination of these:

- Generate short links **randomly**, not serially.
- Use longer **id** in short links rather than 5 or 6 characters — determine the length depending on how many active links you will have at a time.
- If you really need to put sensitive data behind short links, you will have to use additional layer of authentication and authorization.



Sign up for Get Better Tech Emails via HackerNoon.com

By HackerNoon.com

how hackers start their afternoons. the real shit is on hackernoon.com. Take a look

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

Security India Ecommerce Flipkart Cybersecurity

About Help Legal

Get the Medium app

Download on the App Store GET IT ON Google Play