Security Report

EvilTwitter

Group E

abea, beba, gujo, luka, sena

IT University of Copenhagen
Denmark

26 - 4 - 2021

Contents

1 Security Assessment

1.1 Risk Identification

1.1.1 Assets identification

The following assets was used in the EvilTwitter project

- **Droplet on DigitalOcean**: Contains both the webserver and webclient. Also holds the connectionstring for the postgres cluster as a docker secret.
- **Public repository hosted on github.com**: Contains the repository, and stores relevant keys for connecting to the droplet in github secrets.
- **Postgres cluster on DigitalOcean**: Postgres server holding all data provided by users.

1.1.2 Risk Sources and Scenarios

The assets identified in section 1.1.1 have the various vulnerabilities, which will be identified in this section followed by an assessment of the severity via a risk matrix in section 1.2.1. Finally solutions to these risks will be proposed in section 1.2.2. The identified risks are:

1. **Github Actions**: The adversary clones the repository and changes the Github Actions Workflow to get information such as the connection string from docker secrets
2. **Github repository owner's profile gets hacked**: The owner of the repository gets hacked and the attached secrets and deploy keys can be exploited. This will give the adversary direct access to the server.
3. **DDoS attack**: The adversary could do a DDoS attack on the MiniTwit service, making it unavailable for users.
4. **The cloud provider getting hacked**: an adversary could gain access to all the infrastructure and destroy the data, leak it, or extort us.
5. **Nuget Package getting hacked**: A Nuget package which the system depends on could be tampered with and get infected with malware. This would be a supply chain attack and could be used to for instance exploit the connection string for the PostgreSQL cluster.
6. **IP Spoofing/Eavesdropping to get password**: The adversary could gain access to a user's account by IP Spoofing/Eavesdropping to get the hashed passwords of a user, and if they know our hashing algorithm, and the user has a simple password, that they can look up from a rainbow table.

1.2 Risk Analysis

1.2.1 Risk Matrix

		Probability		
		High	Medium	Low
Impact	High	1, 3		2, 4
	Medium		6	5
	Low			

Figure 1: Assessment of risks to the system with the probability of the given risk happening on the x-axis, and the impact of said risk on the system on the y-axis. The number references the risks shown on the list in section 1.1.2

1.2.2 Discuss Scenarios

Solutions to the risk scenarios identified in section 1.1.2 will be provided

- **Github Actions** : Limit commit access to only trusted collaborators
- **Github repository owner's profile gets hacked**: Enable two-factor authentication and setting a strong password.
- **DDoS attack**: sign up for some dos protection service such as Cloudfare, limit the number of request each user can send, add CAPTCHA to stop bots from creating accounts.
- **The cloud provider getting hacked**: store backups at different providers as well as offline
- **Nuget Package getting hacked**: peg the version of every nuget package to a specific version and only update that said package once it its integrity can be confirmed
- **IP Spoofing/Eavesdropping to get password**: We can use an https protocol and we can make strict restrictions on the passwords like minimum 8 characters with upper and lower chase characters and at least one numerical digit. Furthermore, we can use salt and pepper in our password hashing algorithm.