stashcat

Use of GET Request Method With Sensitive Query Strings (CWE 598)

Jens van Laak

May 17, 2020

Stashcat is an instant messenger that is advertised to be safe and conformant to data protection regulations of the European Union. According to the vendor stashcat is used by German State Police in Niedersachsen and in Hessen and the German Armed Forces.

The webapplication of Stashcat uses session credentials for API access in HTTPS GET, POST and OPTIONS requests as parameters. The transmitted session credentials are sufficient for a third party to log in without a password authorization. End-to-end encryption is not implemented for files.

Access to logs of the stashcat webservers or to the logs of SSL-gateways gives unauthorized access to files accessible by the respective user.

Direct requests to the API server with tools like CURL circumvents all restrictive measures of the browser environment (e.g. blocking of file export).

1 Overview

Vendor Stashcat Gmbh (owned by heinekingmedia GmbH)

Vendor Website https://www.stashcat.com

Product stashcat

Version $\leq$ 3.9.1

Product Website https://app.stashcat.com

Impact Information Disclosure

Attack Vector direct HTTPS request to API server with SessionID

2 Description

Stashcat uses an API to communicate with the system's server. Most of the requests are handled via HTTPS GET, POST and OPTION requests.

Two parameters – client_key and device_id – build the sessionID and are sent in each HTTPS request as parameters and not as cookie. The sessionID stays valid over a longer period of time (days), if the user does not explicitly log off. The log off function is *not* available in an easy way.

While normal text messages are end-to-end encrypted files are *not*[1].

3 Description of the attack

An admin with access to either stashcat webserver logs or SSL-gateway logs can retrieve sessionIDs and connect to the system from a third party computer without any password authorisation.

4 Impact

The access to sessionID in combination with the possibility to access the system with the same sessionID from another system even with tools like CURL enables an attacker to get unauthorized access to the unencrypted files of the respective user and copy the files to his own system.

[1] After we contacted the vendor on 2. May 2020 they answered on 15. May 2020 that due to our statement the website text was recently changed to clarify that files are not end-to-end encrypted.