SECFENSE

Secfense User Access Security Broker

www.secfense.com

User identity protection for any web application:

- without software developers,
- without changes in the application code,
- without vendor-lock.

Secfense has two main goals which are:

1. increasing a level of security of the authentication process
2. adding customizable authorization steps wherever it's necessary

What makes Secfense different is that the deployment of a security layer is instantaneous and there is no interference with the protected application code.

Benefits

- Security policies can now be consistent and globally applied for all the applications, employees, contractors and customers. That, instead of dozens of individual security policies making it impossible to manage and control.

- Complete protection against phishing effects, man-in-the-middle attacks and stealing of logged-in users sessions. That, instead of leaving the safety in the hands of users and their passwords.

- Instantaneous deployment of a security layer within just minutes. That, instead of extra costs of application development, instead of problems with integration and instead of adding more and more technologies to support

- Granting access to sensitive resources only to privileged users'. That, instead of low-level employees with high access rights accessing sensitive resources that they don't need for their work.

- Giving privileged users' ability to authorize specific actions or transactions. That instead of low-level employees committing unwanted mistakes that could cost millions.

Architecture

Secfense in big simplification is a form of a security layer, which is spread between any internal applications and their users.

Secfense is also an enterprise service bus (ESB) for security modules such as two-factor authentication (2FA). Each 2FA method is completely independent from the protected applications. This means that it can be exchanged instantly without affecting the application workflow.

Operating principle

The engine of Secfense allows to create security layers for applications without really knowing them. It means that Secfense can be added both in the HTTP(s) protocol layer and in the layers closer to the user (Document Object Model).

Learning Stage

In the learning process stage, Secfense launches a probe into the target application to scan it for requests (and responses) related to user authentication. It is at this stage that both the network traffic patterns and the user interface patterns are collected.

In most cases, the learning process is automatic and takes just several seconds. In rare cases, where patterns are not recognized automatically, the administrator performs manual tuning.

After applying the previously learned pattern, the application becomes instantly protected by the selected 2FA method. During the next login, application users will be prompted to activate the 2FA component.

This is possible by intercepting requests at the user interface level and blocking unauthorized traffic in the HTTP(s) layer.

The user remains in the domain of the protected application (no redirection to an external service), and the 2FA registration/use process becomes an integral part of the protected application.

This mechanism works for both traditional applications (where the HTML code is rendered completely on the server side), as well as the so-called SPA (Single-Page Apps).

IT IS IMPORTANT TO STRESS OUT THAT SECFENSE DOES NOT STORE OR ANALYZE USER PASSWORDS, ONLY USER NAMES IN THE CONTEXT OF A GIVEN APPLICATION.

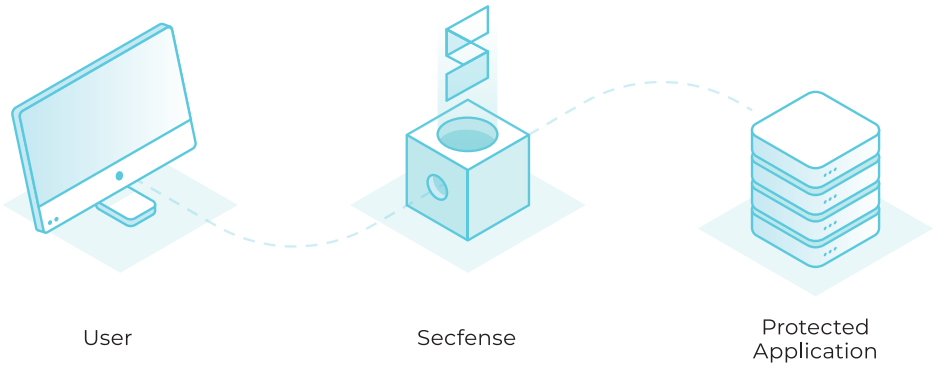
Integration

Secfense is delivered to the customer in the form of a physical or virtual appliance.

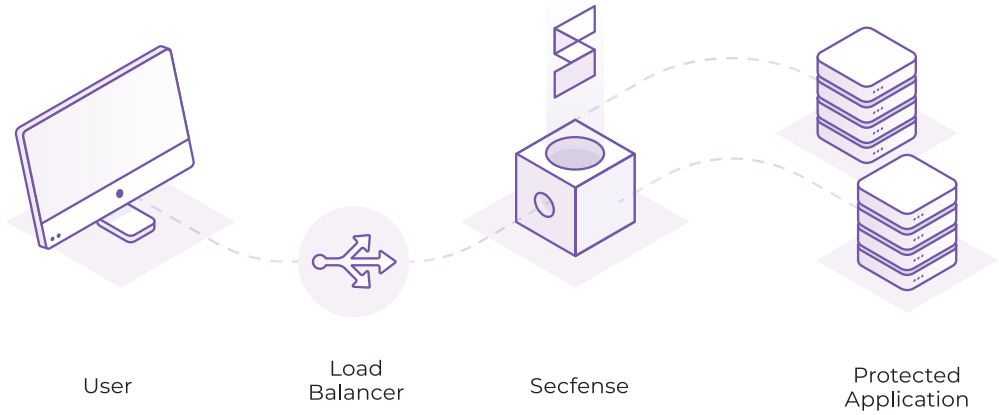
It is placed on the route between users and applications in such a way that it can analyze and modify HTTP(s) traffic - usually near load-balancer or application firewall. Depending on the client's preferences, Secfense can work as TLS termination proxy or work on already decrypted traffic. Due to the confidentiality of communications Secfense never communicates with external hosts, nor does it "call home" for reporting purposes.

Secfense installation options:

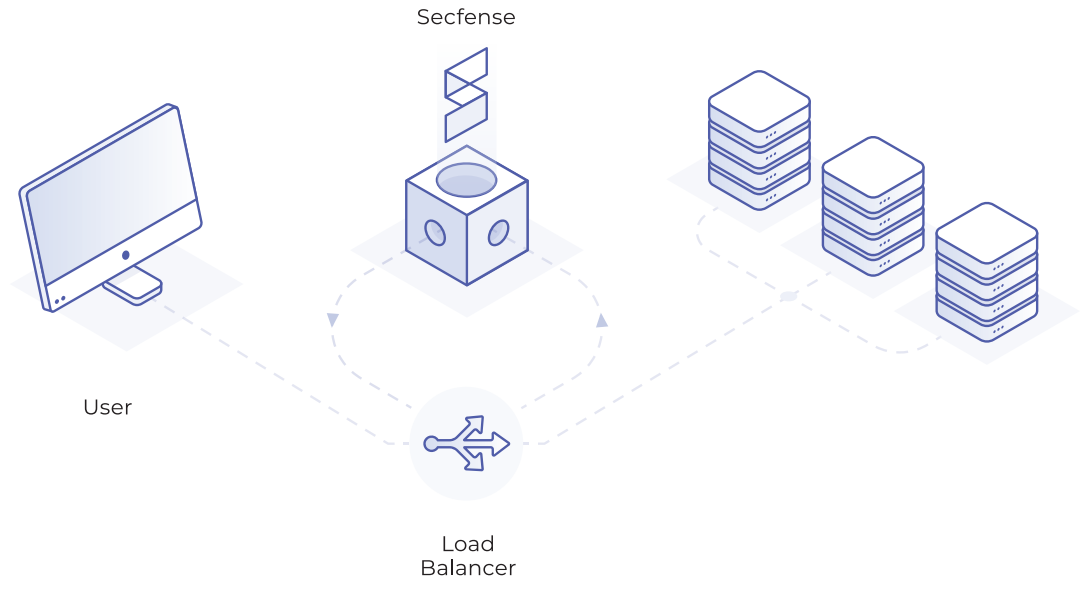
a) Inline (Secfense as SSL / TLS proxy)



b) Inline (Secfense working on decrypted traffic)



c) On a stick (flow on the load balancer is “wrapped” by Secfense)



Microauthorizations of the key transactions

One of the crucial functionalities of Secfense are microauthorizations. This functionality makes it possible to stop the user when he reaches for some specific resources or wants to perform some specific actions in the protected application.

In such a case, Secfense takes over communication and triggers one of two scenarios:
1. In the OWNER scenario, Secfense asks the user to re-authenticate,
2. In the SUPERVISOR scenario, Secfense asks the user with supervision privilages.

Since Secfense, as described above, works as an intermediate security layer, so microauthorizations can be added inside of the application anywhere, and it only takes minutes to deploy.

Microauthorizations (in the OWNER scenario) introduce an increased level of granulation under the Principle of Least Privilege. This means additional protection against attack on a stolen active session or other attacks against an already logged-in user (including real-time phishing or malware).

Microauthorizations (in the OWNER scenario) leave authorization of particularly sensitive resources requests in the hands of selected and trusted users. Regardless of the scenario, the additional effect of microauthorizations is the protection of sensitive resources against risks such as:
- automatic export (with or without the consent of the user)
- uncontrolled leakage of confidential data through the application interface

MICROAUTHORIZATIONS ONLY MAKE SENSE WHEN THEY ARE EFFORTLESS FOR THE USER

That is why the best tool to include in microauthorizations are U2F / FIDO2 cryptographic keys or local authenticators compliant with the WebAuthn standard. The 2FA methods based on one-time codes (SMS, TOTP) will not work because of too much of user involvement in the process.

In the case of the OWNER scenario, the access to the protected resource requires the user to simply touch the cryptographic key that was used during the authentication.

In the case of SUPERVISOR scenario, access to the protected resource requires the same action as above but needs to be performed by the privileged user with a privileged cryptographic key.

ALL EVENTS RELATED TO MICROAUTHORIZATIONS ARE LOGGED IN THE SECFENSE EVENT LOG (OR STREAMED TO AN EXTERNAL LOGIN SYSTEM) AND CAN BE ANALYZED TO DETECT ANOMALIES.