Development of a web filter to control access to web resources

D.A. Bizin, S.A. Burlov

The flowchart of a web filter

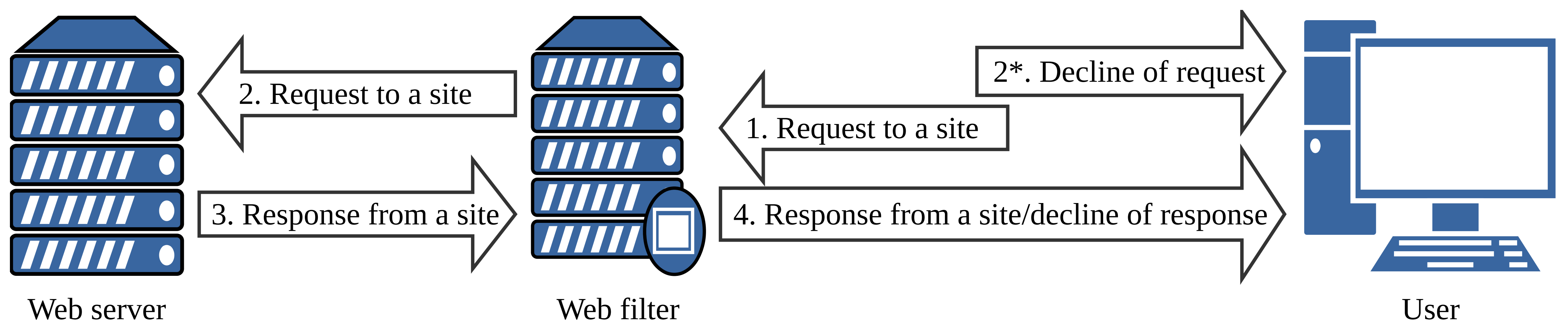


Fig. 1. The common flowchart of a web filter

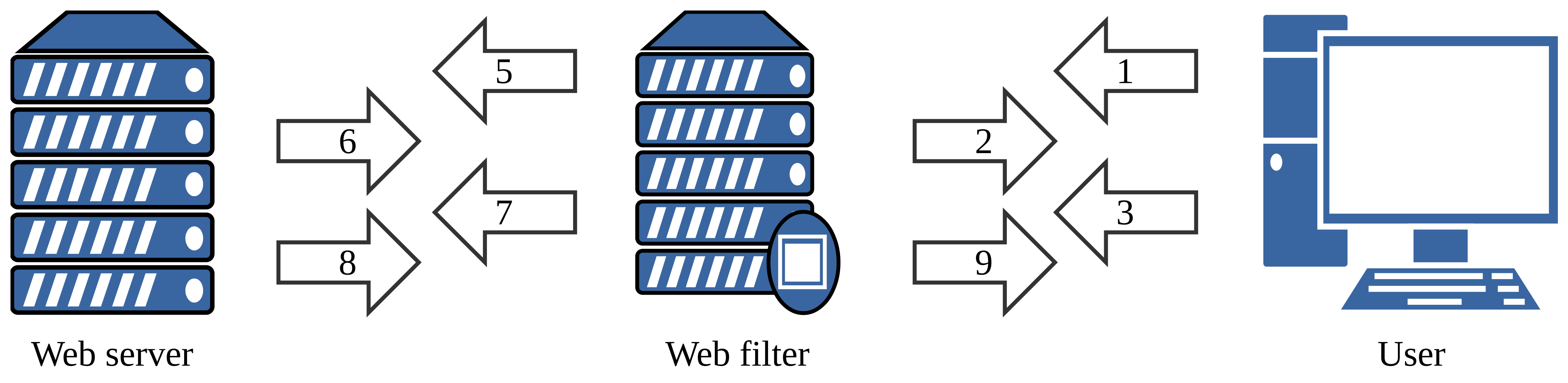


Fig. 2. The flowchart of a web filter which using approach of the Trusted MITM

1. The web filter intercepts a HTTPS request from a browser to a site.
2. The web filter generates a key pair (public and private keys) and the certificate for the domain name requested by the browser (the domain name are specified in Common Name and Subject Alternative Name fields of the certificate). The generated certificate are signed by the trusted root certificate which trusted by the browser. The web filter sends the generated certificate to the browser.
3. The browser checks the certificate generated by the web filter. In case of passing checks, the browser generated a symmetric key, encrypts it by the public key from the certificate of the web filter and sends it back.
4. The web filter decrypts the symmetric key using the private key and saves it.
5. The web filter does the HTTPS request from the step 1 to the web server.
6. The web server sends its certificate to the web filter.
7. The web filter checks the certificate and, in case of passing checks, generates a symmetric key, encrypts it by the public key from the certificate of the web server and sends it back to the web server.
8. The web server decrypts the symmetric key using the private key, encrypts a content using the symmetric key and sends it to the web filter.
9. The web filter decrypts the content using the symmetric key, encrypts it using the symmetric key saved on the step 4 and sends it to the browser.
10. The browser decrypts the content using the symmetric key and displays it.

Naive Bayes classifier

Naive Bayes classifier is used by the web filter for classify of a web content. The web filter makes a decision about restrict access to the content based on the classification data. The classes for filtering are specified manually.

$d=\{t_1,t_2,\ldots,t_n\}$ — a document (word vector), $C=\{c_j\}$ — a set of predefined classes

The purpose of the classifier is a finding class with the most hit probability of the document in this class: $arg(\underset{c_i}{max}(P(c_i|d)))$

The Bayes formula is used to calculate the probability: $P(c_i|d)=\dfrac{P(c_i)\cdot P(d|c_i)}{P(d)}$

$P(c_i)$ — the ratio of the number of documents in the class from the training set to the total number of documents

$P(d)$ — does not depend on a class therefore not affected of the search for the maximum

Because many documents have a large number of words, a "naive" assumption is made that any two words in the document are statistically independent of each other (two independent events). As a result of this assumption we have the following formula:

$$P(d|c_i)=\prod_{j=1}^{n} P(t_j|c_i)$$

Conclusion

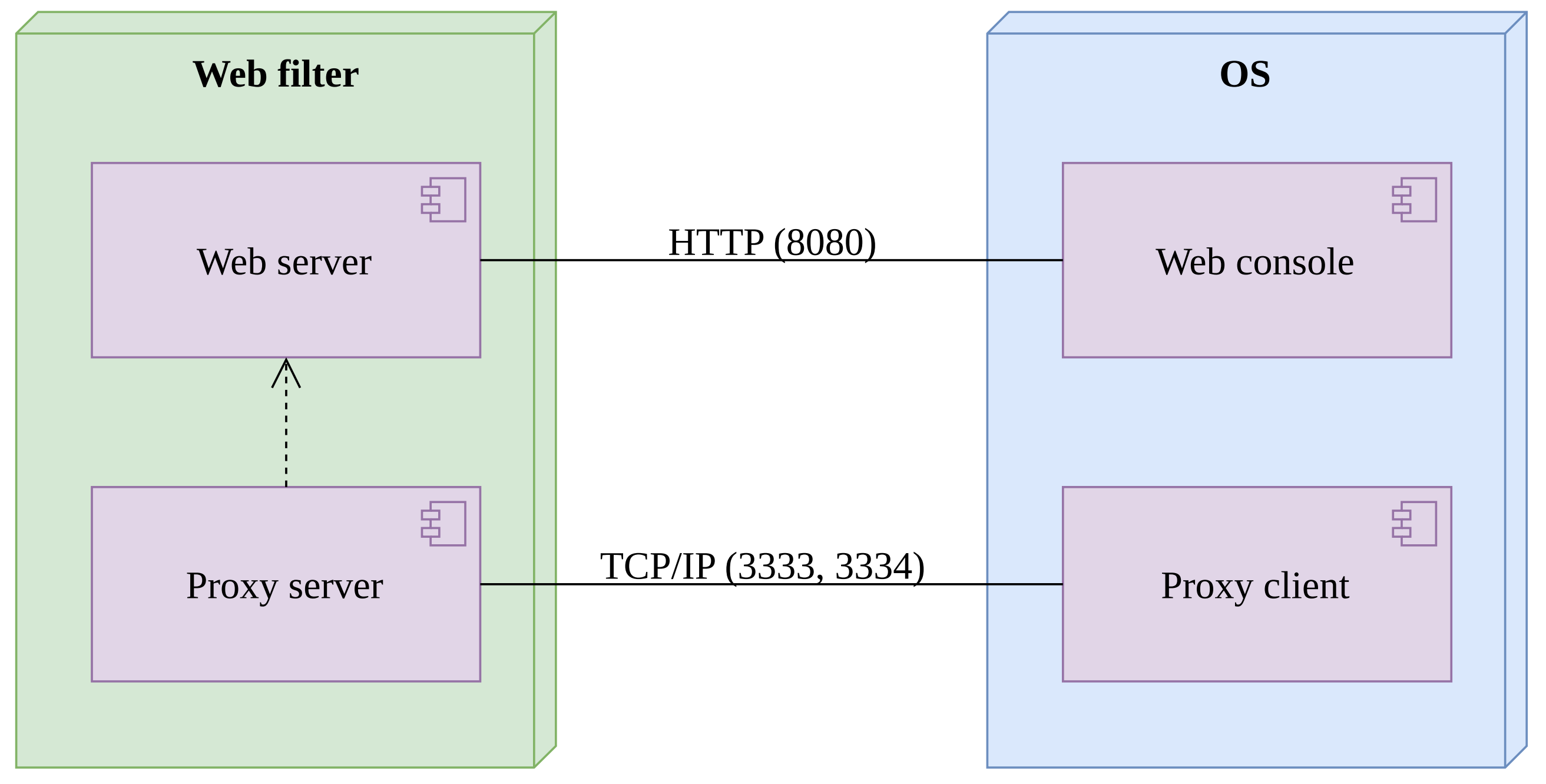
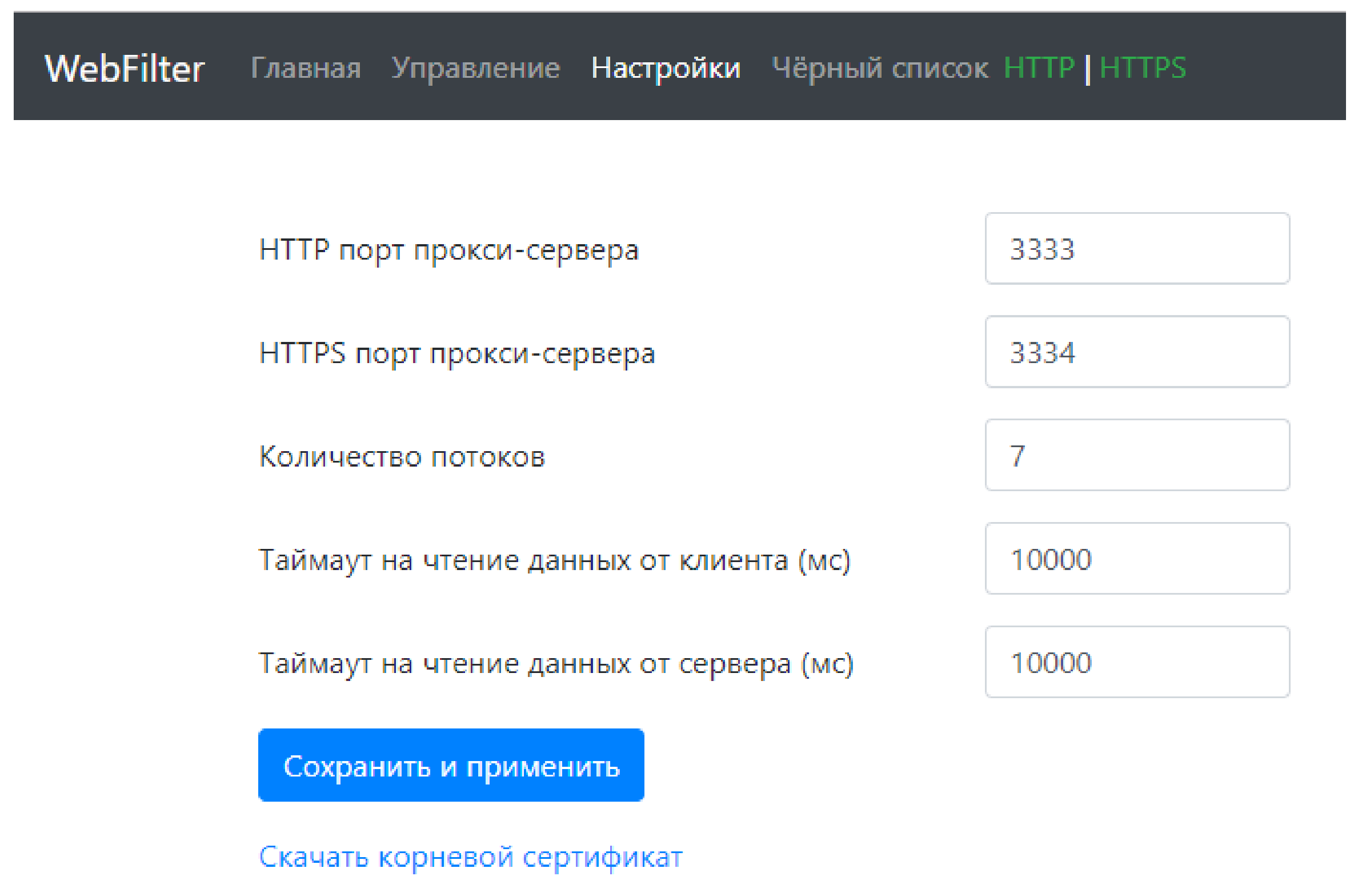


Fig. 3. The deployment diagram of the web filter

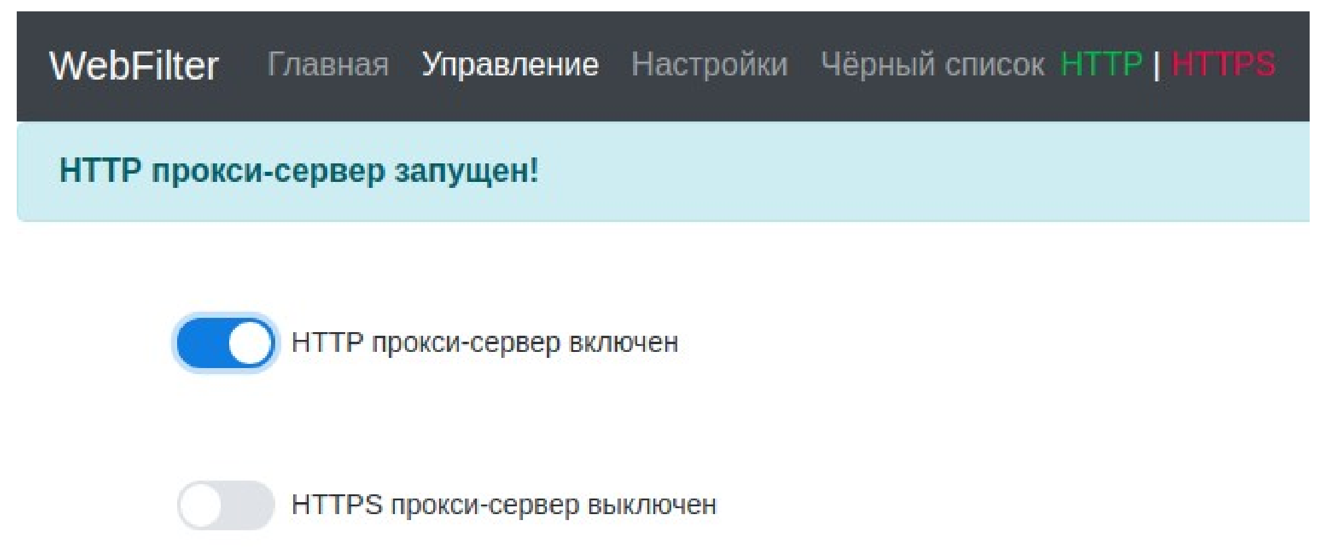


HTTP порт прокси-сервера	3333
HTTPS порт прокси-сервера	3334
Количество потоков	7
Таймаут на чтение данных от клиента (мс)	10000
Таймаут на чтение данных от сервера (мс)	10000

Fig. 4. The settings page of the web filter

In this paper was developed a software – the web filter to control access to web resources. This solution may be used, for example, in a educational institutions for the purposes of protection of minors from harmful information in the Internet.

The server's part of the web filter is written in Java. The web filter works in multithreading mode handling each request in standalone thread. The proxy server of the web filter is developed on TCP-socket layer. The web server is developed using Java Servlets. The H2 database which working in the embedded mode is used to store settings of the web filter. The web console for managing the web filter is written in JavaScript using jQuery and Bootstrap frameworks.



HTTP прокси-сервер запущен!

HTTP прокси-сервер включен

HTTPS прокси-сервер выключен

Fig. 5. The management page of the web filter

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY