Development of a web filter to control access to web resources

D.A. Bizin, S.A. Burlov

Схема работы веб-фильтра

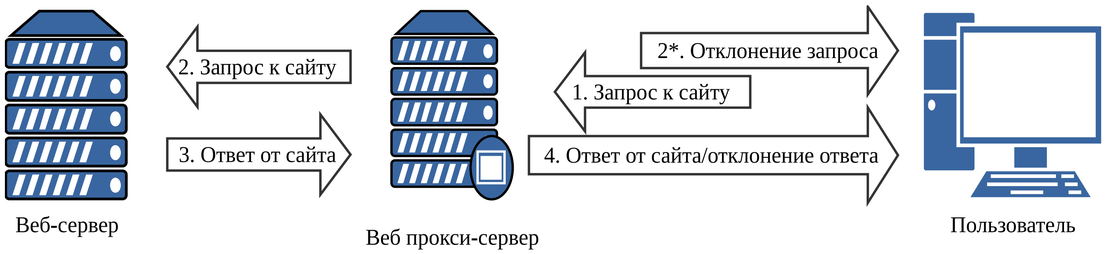

Рис. 1 — Общая схема работы веб-фильтра

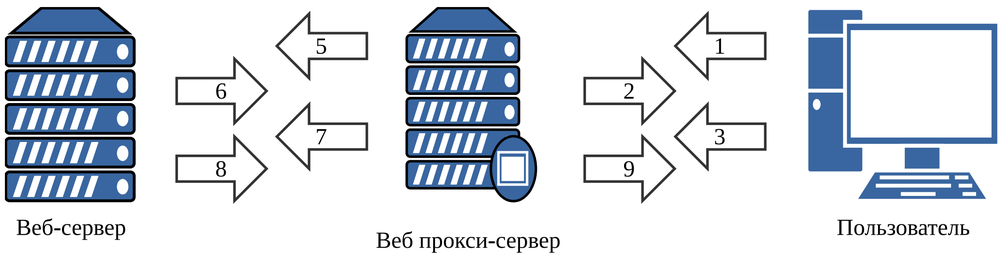

Рис. 2 — Схема работы веб-фильтра с HTTPS-трафиком (Trusted MITM)

1. Перехват веб-фильтром запроса браузера к веб-серверу по протоколу HTTPS.
2. Прокси-сервер генерирует ключевую пару (открытый и закрытый ключи) и сертификат для запрашиваемого браузером доменного имени (в поля Common Name и Subject Alternative Name сертификата проставляется доменное имя). Сгенерированный сертификат подписывается доверенным корневым сертификатом (которому доверяет браузер). Веб-фильтр отправляет сгенерированный сертификат клиенту.
3. Браузер проверяет сгенерированный сертификат веб-фильтра. В случае прохождения проверок, браузер генерирует симметричный ключ, шифрует его открытым ключом из сертификата прокси-сервера и отправляет обратно.
4. Веб-фильтр расшифровывает симметричный ключ, используя закрытую часть ключа, и сохраняет его.
5. Веб прокси-сервер делает запрос из шага 1 к веб-серверу по протоколу HTTPS.
6. Веб-сервер отправляет прокси-серверу свой сертификат.
7. Веб-фильтр проверяет сертификат и, в случае прохождения проверок, генерирует симметричный ключ, зашифровывает его открытым ключом из сертификата сервера и отправляет обратно веб-серверу.
8. Веб-сервер расшифровывает симметричный ключ, используя закрытую часть, и отправляет прокси-серверу контент, зашифрованный симметричным ключом.
9. Веб-фильтр расшифровывает контент при помощи симметричного ключа, шифрует его симметричным ключом, сохраненным на шаге 4, и отправляет клиенту.
10. Браузер расшифровывает контент при помощи симметричного ключа и отображает его.

Наивный байесовский классификатор

Наивный байесовский классификатор используется веб-фильтром для классификации контента. На основании данных классификации веб-фильтр принимает решение о запрете доступа к контенту. Категории для фильтрации можно задавать вручную.

$d=\{t_1, t_2, \ldots, t_n\}$ — документ (вектор слов), $C=\{c_j\}$ — множество заранее заданных категорий

Задача классификатора заключается в нахождении категории, в которую вероятнее всего попадает документ:

$$arg\left(\max_{c_i}\left(P\left(c_i \mid d\right)\right)\right)$$

Для вычисления вероятности используется формула Байеса: $P\left(c_i \mid d\right)=\frac{P\left(c_i\right)\cdot P\left(d \mid c_i\right)}{P\left(d\right)}$

$P\left(c_i\right)$ — отношение количества документов данной категории из обучающей выборки к общему числу документов

$P\left(d\right)$ — не зависит от категории, поэтому не влияет на поиск максимума

Из-за большого количества слов в документе делается «наивное» предположение о том, что любые два слова статистически не зависят друг от друга (два независимых события). В результате данного предположения имеем:

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

Результаты

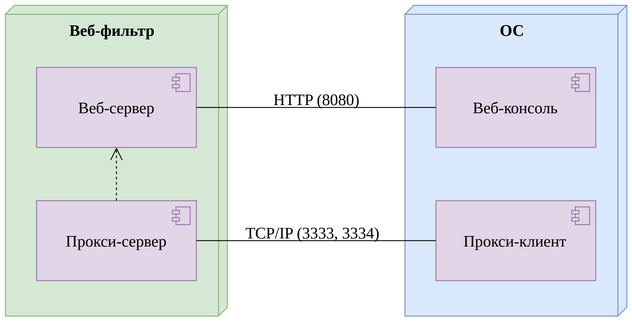

Рис. 3 — Диаграмма развертывания веб-фильтра

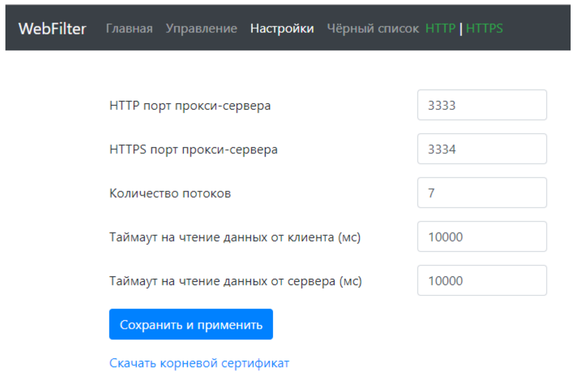

Рис. 4 — Страница настроек веб-фильтра

В результате проделанной работы был реализован программный продукт – веб-фильтр для контроля доступа к веб-ресурсам. Данное решение может быть использовано, например, в образовательных учреждениях с целью ограждения несовершеннолетних от нежелательной информации в интернете.

Серверная часть веб-фильтра написана на Java. Контент-фильтр работает в многопоточном режиме, обрабатывая каждый запрос в отдельном потоке. Прокси-сервер веб-фильтра реализован на уровне TCP-сокетов, веб-сервер работает на технологии Java Servlets. В качестве хранилища настроек используется база данных H2, работающая во встроенном режиме. Веб-консоль для управления работой веб-фильтра написана на JavaScript с использованием фреймворков jQuery и Bootstrap.

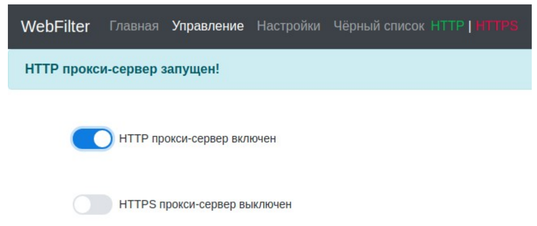

Рис. 5 — Страница управления веб-фильтром

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY