Научная статья
Original article
УДК 004.424

КОНТРАБАНДА HTTP-ЗАПРОСОВ

SMUGGLING HTTP REQUESTS

Казарян Каторина Камоевна студент специалитета, Донской государственный технический университет, г. Ростов-на-Дону (344003 Россия г. Ростов-на-Дону, Гагарина 1), kazaryan_katorina@mail.ru

Kazaryan Katorina Kamoevna specialty student, Don State Technical University, Rostov-on-Don (344003 Russia, Rostov-on-Don, Gagarina 1), kazaryan_katorina@mail.ru

Аннотация: В данной статье указано, что такое контрабанда HTTP-запросов, уязвимости, связанные с контрабандой запросов, как работает контрабанда HTTP-запросов. Контрабанда HTTP-запросов использует несогласованность при синтаксическом анализе HTTP-запросов, не соответствующих RFC, через два HTTP-устройства (как правило, внутренний сервер и брандмауэр с поддержкой HTTP или внешний прокси).

Abstract: This article explains what HTTP request smuggling is, request smuggling vulnerabilities, and how HTTP request smuggling works. HTTP request smuggling exploits the inconsistency in parsing non-RFC HTTP requests through two HTTP devices (typically an internal server and an HTTP-enabled firewall or external proxy).

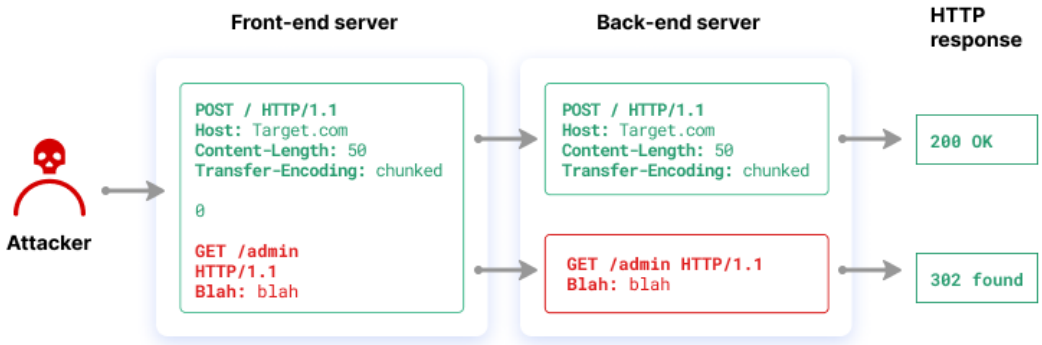
Ключевые слова: Контрабанда HTTP-запросов, уязвимости, контрабандой запросов

Keywords: HTTP request smuggling, vulnerabilities, request smuggling

Процесс контрабанды HTTP-запросов осуществляется путем создания нескольких настраиваемых HTTP-запросов, которые позволяют двум целевым объектам видеть две отдельные серии запросов.

Заголовок HTTP предлагает два различных способа указать, где заканчивается запрос: заголовок Transfer-Encoding и заголовок Content-Length. Уязвимость передачи HTTP-запросов возникает, когда злоумышленник отправляет оба заголовка в одном запросе. Это может привести к тому, что внешний или внутренний сервер неправильно интерпретирует запрос, передав злонамеренный HTTP-запрос.

Уязвимости, связанные с контрабандой запросов, позволяют киберпреступникам обходить меры безопасности , получать доступ к конфиденциальной информации и напрямую скомпрометировать различных пользователей приложений. Его также можно использовать для вторичных эксплойтов, включая обход брандмауэров, частичное отравление кеша и межсайтовый скриптинг (XSS).



Как работает контрабанда HTTP-запросов

Каковы последствия атак контрабанды HTTP-запросов?

Когда злоумышленнику удается выполнить атаку контрабанды запросов, он вводит вредоносный HTTP-запрос на веб-сервер, минуя внутренние меры безопасности. Это может позволить злоумышленнику:

- Получите доступ к защищенным ресурсам, таким как консоли администратора
- Получите доступ к конфиденциальным данным

- Взлом сеансов веб-пользователей
- Запускать атаки межсайтового скриптинга (XSS) без каких-либо действий со стороны пользователя.
- Взломать учетные данные

Примеры контрабанды HTTP-запросов

Большинство атак контрабанды HTTP-запросов используют слабость длины контента (CL), слабость кодирования передачи (TE) или и то, и другое. Три основных метода атаки известны как « **CL.TE** », что означает, что атака использует длину контента на передней стороне, а затем передает кодирование на задней стороне, « **TE.CL** » для противоположного и « **TE.TE** » для двойное использование кодирования передачи, как на передней, так и на задней стороне.

Уязвимости CL.TE (Content-Length.Transfer-Encoding)

Атака контрабанды HTTP-запросов CL.TE предполагает, что интерфейсный сервер отдает приоритет заголовку Content-Length, а внутренний сервер отдает приоритет заголовку Transfer-Encoding.

Атака проводится следующим образом. Первая часть запроса объявляет короткую длину блока, обычно 0. Внешний сервер считывает только первую часть запроса и передает вторую часть внутреннему серверу.

В следующем примере текст, начинающийся с «ВРЕДОНОСНЫЙ ЗАПРОС», передается внутреннему серверу, который рассматривает его как следующий запрос и обрабатывает его.

```
POST / HTTP / 1.1
Хост: website.com
Длина содержимого: 13
Кодирование передачи: фрагментированное

0
ВРЕДОНОСНЫЙ ЗАПРОС
```

TE.CL (Transfer-Encoding.Content-Length) Уязвимости

Атака с контрабандой HTTP-запросов TE.CL предполагает, что интерфейсный сервер отдает приоритет уязвимости кодирования передачи, в то время как внутренний сервер отдает приоритет уязвимости длины содержимого.

В этом типе атаки злоумышленник объявляет длину первого фрагмента до вредоносного запроса включительно. Второй фрагмент объявлен как имеющий нулевую длину, поэтому интерфейсный сервер предполагает, что запрос выполнен. Он передает запрос внутреннему серверу, который получает и обрабатывает его.

Вот как может выглядеть запрос:

```
POST / HTTP / 1.1
Хост: weak-website.com
Длина содержимого: 3
Кодирование передачи: фрагментированное

15

ВРЕДОНОСНЫЙ ЗАПРОС

0
```

Каждый из предложенных методов является отклонением от спецификации HTTP. Реальный код, который реализует спецификацию протокола, редко придерживается его с абсолютной точностью, и для разных реализаций характерно допускать различные отклонения от спецификации. Чтобы выявить уязвимость, необходимо найти какой-либо вариант заголовка Transfer-Encoding, который обработает только один из серверов (фронтенд или бэкенд), а другой проигнорирует.

В зависимости от того, будет ли это фронтенд или бэкенд сервер, оставшаяся часть атаки примет ту же форму, что была описана для уязвимостей CL.TE или TE.CL.

Уязвимости в HTTP возникают в тех случаях, когда фронтенд сервер пересылает несколько запросов на бэкенд сервер по одному и тому же

сетевому соединению, а протокол, используемый для внутренних подключений, несёт в себе риск того, что политика серверов по обработке длины запросов не согласована. Ниже перечислены несколько общих правил предотвращения возникновения уязвимостей в HTTP:

Используйте для каждого запроса своё подключение между бэкенд и фронтендом.

Используйте HTTP/2 для бэкенд подключений, так как этот протокол предотвращает неоднозначность в интерпретации длины сообщений.

Используйте одно и то же программное обеспечение веб-сервера для фронтенд и бэкенд серверов.

В некоторых случаях уязвимостей можно избежать, если заставить фронтенд сервер нормализовать неоднозначные запросы или заставить бэкенд сервер отклонять неоднозначные запросы и закрывать сетевое соединение. Однако эти подходы потенциально больше подвержены ошибкам, чем общие правила, указанные выше.

Литература

1. «Объём HTTP-трафика впервые превысил P2P» - net.compulenta.ru/322974/ — Компьюлента, 22 июня 2007. (Официальный документ от Ellacoya Networks - www.ellacoya.com/news/pdf/2007/NXTcommEllacoyaMediaAlert.pdf)
2. «HTTP-NG — объектно-ориентированный протокол передачи гипертекста» - www.osp.ru/cw/1998/20/29525/, «Открытые системы», 29 мая 1998 г. «(Proposed) HTTP-NG Working Group» - www.w3.org/Protocols/HTTP-NG/ — официальная страница W3C по разработке протокола HTTP-NG.
3. HTTP/1.1: Method Definitions - www.w3.org/Protocols/rfc2616/rfc2616-sec9.html (англ.). W3C.
4. См. первое предложение раздела «6.1.1 Status Code and Reason Phrase - tools.ietf.org/html/rfc2068#section-6.1.1» в RFC 2068. На стр. 40 - tools.ietf.org/html/rfc2068#page-40 есть также объявление в формате

расширенной БНФ-формы (Augmented BNF) «extension-code = 3DIGIT» для кодов расширений.

5. HTTP errors - www.flickr.com/photos/apelad/sets/72157594388426362/detail/

References

1. "The volume of HTTP traffic for the first time exceeded P2P" - net.compulenta.ru/322974/ - Compulenta, June 22, 2007. (Official document from Ellacoya Networks - www.ellacoya.com/news/pdf/2007/NXTcommEllacoyaMediaAlert.pdf)
2. "HTTP-NG - object-oriented hypertext transfer protocol" - www.osp.ru/cw/1998/20/29525/, Open Systems, May 29, 1998 "(Proposed) HTTP-NG Working Group" - www.w3.org/Protocols/HTTP-NG/ - W3C's official page on the development of the HTTP-NG protocol.
3. HTTP/1.1: Method Definitions - www.w3.org/Protocols/rfc2616/rfc2616-sec9.html W3C.
4. See the first sentence of section "6.1.1 Status Code and Reason Phrase - tools.ietf.org/html/rfc2068#section-6.1.1" in RFC 2068. On page 40 - tools.ietf.org/html/ rfc2068#page-40 there is also an Augmented BNF declaration "extension-code = 3DIGIT" for extension codes.
5. HTTP errors - www.flickr.com/photos/apelad/sets/72157594388426362/detail/

© Казарян К.К., 20221 Научно-образовательный журнал для студентов и преподавателей «StudNet» №1/2022.

Для цитирования: Казарян К.К. КОНТРАБАНДА HTTP-ЗАПРОСОВ// Научно-образовательный журнал для студентов и преподавателей «StudNet» №1/2022.