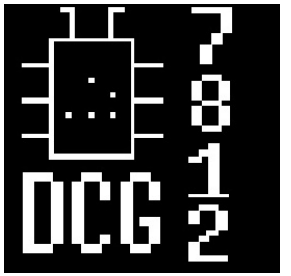


Триста два швейцарских армейских ножа для багхантера

10.08.2013

DCG #7812

Chaos Constructions, г. Санкт-Петербург

Андрей Лабунец – @isciurus

О себе

Учусь в ТюмГУ[1]

isciurus.blogspot.com – почитать

@isciurus – зафолловить, твитнуть

isciurus@gmail.com – связаться

1. место, где ломают Хром без повреждений памяти

Авторизация в веб через OAuth. Чё такое?

— Фреймворк для построения протоколов авторизации blah-blah-blah

— Вместо пароля к аккаунту теперь токен

— Данные передаются через различные кросс-доменные каналы: редиректы, postMessage, Флэш и т.д.

Авторизация в веб через OAuth. Чё интересного?

— Веб – огромный вектор для атак сам по себе (например, WebUI в Хроме)

— Протоколы авторизации – связывают домены, расширяют простор для атак

Авторизация в веб через OAuth. Как её дизайнят?

— Стандарты не описывают важные детали реализации (кросс-доменная передача и другое)

— Стандарты и код сражаются с багами и фичами браузеров, хрупким веб-стеком и жадными баг-хантерами

— Благодаря стандартам, ломать всех теперь можно примерно одинаково

Авторизация в веб через OAuth. Как её кодят?

— Быстро поменять протокол не получится: теперь на нём завязаны ваши клиенты

— Делать только глядя в RFC не получится: например, CSRF-защита не обязательна по стандарту

Авторизация в веб через OAuth. Как исследуют её безопасность?

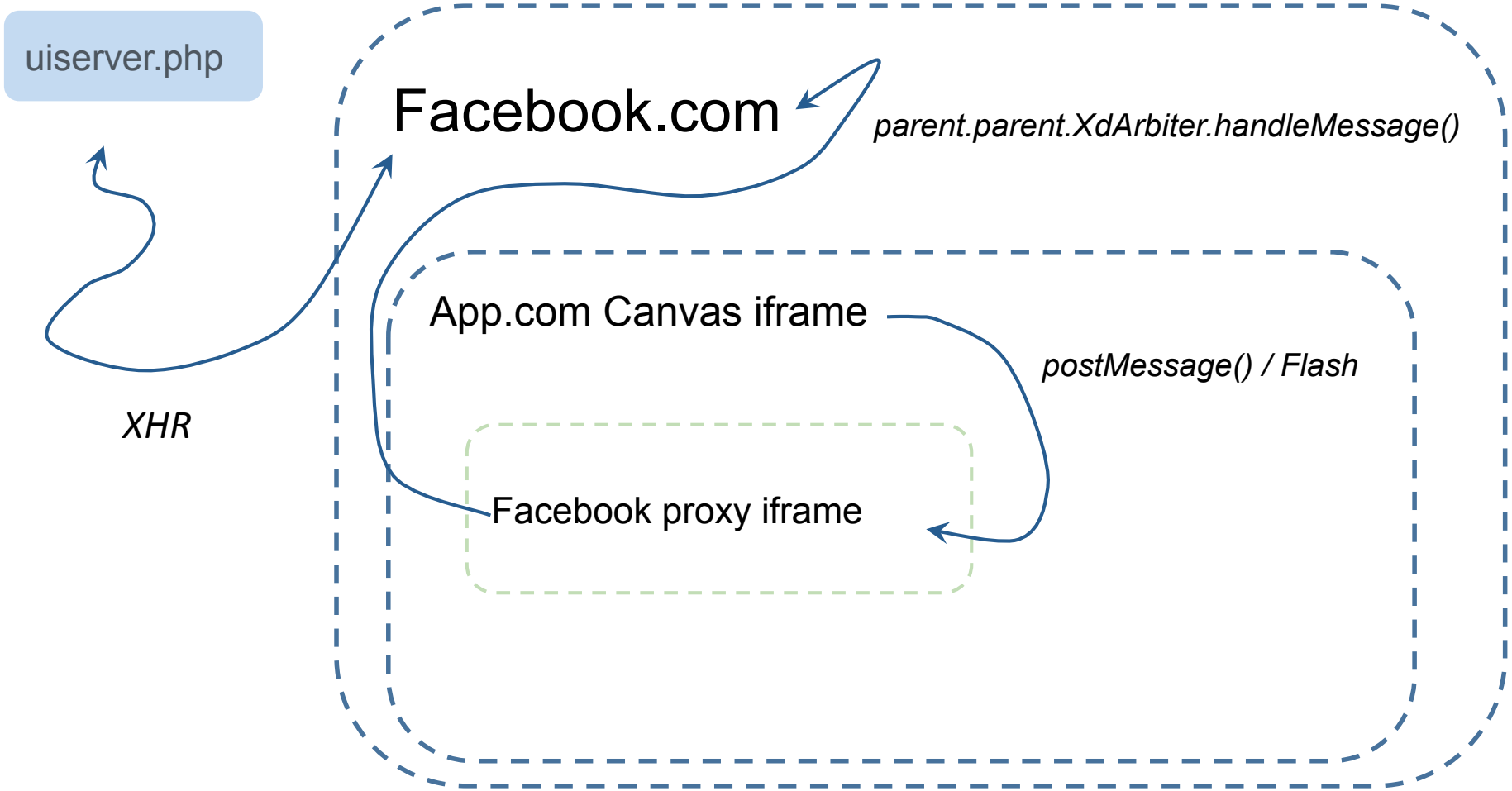
— Перечисление найденных уязвимостей реализации

— Формальная верификация (о, боже)

— Фишинг, пермишшны и все такое

Пара интересных багов.
DOM XSS в Фейсбуке через OAuth

Пара интересных багов.
DOM XSS в Фейсбуке через OAuth



Пара интересных багов.
DOM XSS в Фейсбуке через OAuth

— XHR-скрипт *uiserver.php* возвращает 302-ой код в ответ на OAuth-запрос

— Если домен тот же самый, то редирект прозрачен для XHR-хэндлера

— Подделав *redirect_uri*, можно заставить XHR-хэндлер Фейсбука выполнить любые данные со своего домена как javascript

— А свой код можно загрузить на Фейсбук, упаковав его в картинку

— Почитать подробнее:
http://isciurus.blogspot.ru/2013/04/a-story-of-9500-bug-in-facebook-oauth-20.html

Пара интересных багов.
DOM XSS в Фейсбуке через OAuth

Значит, проблема OAuth – нестатичные *redirect_uri*?

Пара интересных багов.
RCE в Хроме через OAuth

Пара интересных багов.
RCE в Хроме через OAuth

А точнее, через фиксацию сессии в Chrome Sync

Задачи:

1. Обойти CSRF-защиту
2. Обойти изоляцию сайтов (стать доверенным signin-рендерером) и зафиксировать сессию
3. Доставить свой код через расширения

Пара интересных багов.
RCE в Хроме через OAuth

1. Обход ЦСРФ
   - Межсайтовый скриптинг на поддомене Гугла
   - ... благодаря кривому транспорту токенов в OAuth-прокси
   - OAuth вынесен на accounts.google.com
   - ... поэтому XSS на самом важном поддомене

   То, что надо

Пара интересных багов. RCE в Хроме через OAuth

2. Стать доверенным рендерером

— Глядя на ссылку и её домен accounts.google.com, Хром решает, что это логин в Chrome Sync

— … и назначает текущий процесс доверенным

— На самом деле, ссылка ведет на конечную точку OAuth у Гугла и редиректит на чужой домен, откуда триггерится DOM XSS и фиксируется сессия

Пара интересных багов. RCE в Хроме через OAuth

3. Доставить код через расширения
   - Некоторая магия с синхронизацией NPAPI-компонентов

Одна интересная мысль

Одна интересная мысль

— *redirect_uri* на этот раз статичные,

— и пользователь вовсе не обязан авторизировать наше OAuth-приложение на Гугле

— Но редирект все равно возможен

— С accounts.google.com в любое место

Одна интересная мысль

Как пользоваться автоматическими редиректами (без авторизации):

— OAuth Гугла: immediate=true

— OAuth Фейсбука: display=none

— OpenID: checkid_immediate

— Да это же open redirect!

— This is a vulnerability © owasp.org

Одна интересная мысль

Что объединяет протоколы авторизации:

- OAuth/OpenID есть у любого уважающего себя сайта
- Конечная точка вешается на главный поддомен
- Всегда есть безусловный редирект на домен клиента
- Создать свой клиент очень просто

Одна интересная мысль

Когда протокол авторизации может выйти боком:

— Браузер обрабатывает данные с домена, на котором есть конечная точка OAuth/OpenID

— Атакующий частично или полностью контролирует url

— Система доверяет данным с домена провайдера

Одна интересная мысль

Как решать проблему?

— Избавиться от редиректов при авторизации вообще

— Помнить, что страницы с собственного домена могут отредиректить куда угодно

— Убрать безусловные редиректы