SEKURAK HACKING PARTY #2

- Wsparcie $$$ zapewniają nam: securitum.pl & nomino.pl

- Wsparcie medialne zapewniają nam:

SEKURAK HACKING PARTY #2

- Michał Sajdak
  - Wstęp & dwa słowa o Sekurak/Offline
- Michał Bentkowski
  - Hacking Google
- Mateusz “j00ru” Jurczyk
  - Fakty, mity i przemyślenia na temat offensive security
- Gynvael Coldwind
  - O książce, o programowaniu, o bezpieczeństwie

SEKURAK HACKING PARTY #2

- Historia
  - Pierwsze SHP – 100% warsztaty (cały dzień) w nieco spartańskich warunkach

SEKURAK HACKING PARTY #2

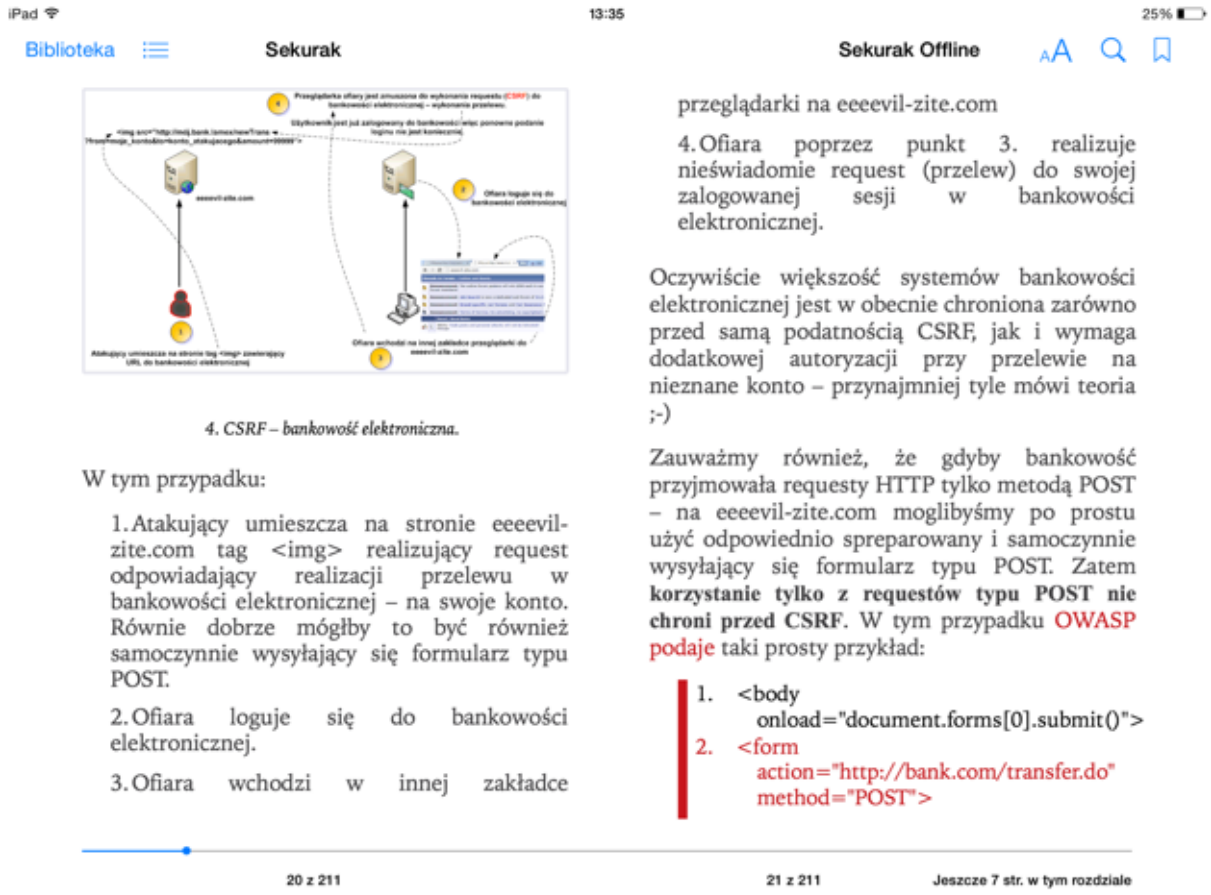
- Ciężko się jest zebrać do zorganizowania całodziennej imprezy
- Pomysł częstszych SHP, ale krótszych
- Czy się sprawdzi? Ocenicie sami ☺

sekurak.pl/sekurak-offline/

- Kto już czytał?
- Bezpłatny magazyn o ITsecurity
- #1 o bezpieczeństwie aplikacji www
- Profesjonalny skład sfinansowany przez http://securitum.pl/

sekurak.pl/sekurak-offline/

- Magazyn dostępny w:
  - PDF
  - Epub
  - Mobi
- Layout optymalizowany pod ekrany
- ~60 stron



4. *CSRF – bankowość elektroniczna.*

W tym przypadku:

1. Atakujący umieszcza na stronie eeeevil-zite.com tag <img> realizujący request odpowiadający realizacji przelewu w bankowości elektronicznej – na swoje konto. Równie dobrze mógłby to być również samoczynnie wysyłający się formularz typu POST.
2. Ofiara loguje się do bankowości elektronicznej.
3. Ofiara wchodzi w innej zakładce przeglądarki na eeeevil-zite.com
4. Ofiara poprzez punkt 3. realizuje nieświadomie request (przelew) do swojej zalogowanej sesji w bankowości elektronicznej.

Oczywiście większość systemów bankowości elektronicznej jest w obecnie chroniona zarówno przed samą podatnością CSRF, jak i wymaga dodatkowej autoryzacji przy przelewie na nieznane konto – przynajmniej tyle mówi teoria ;-)

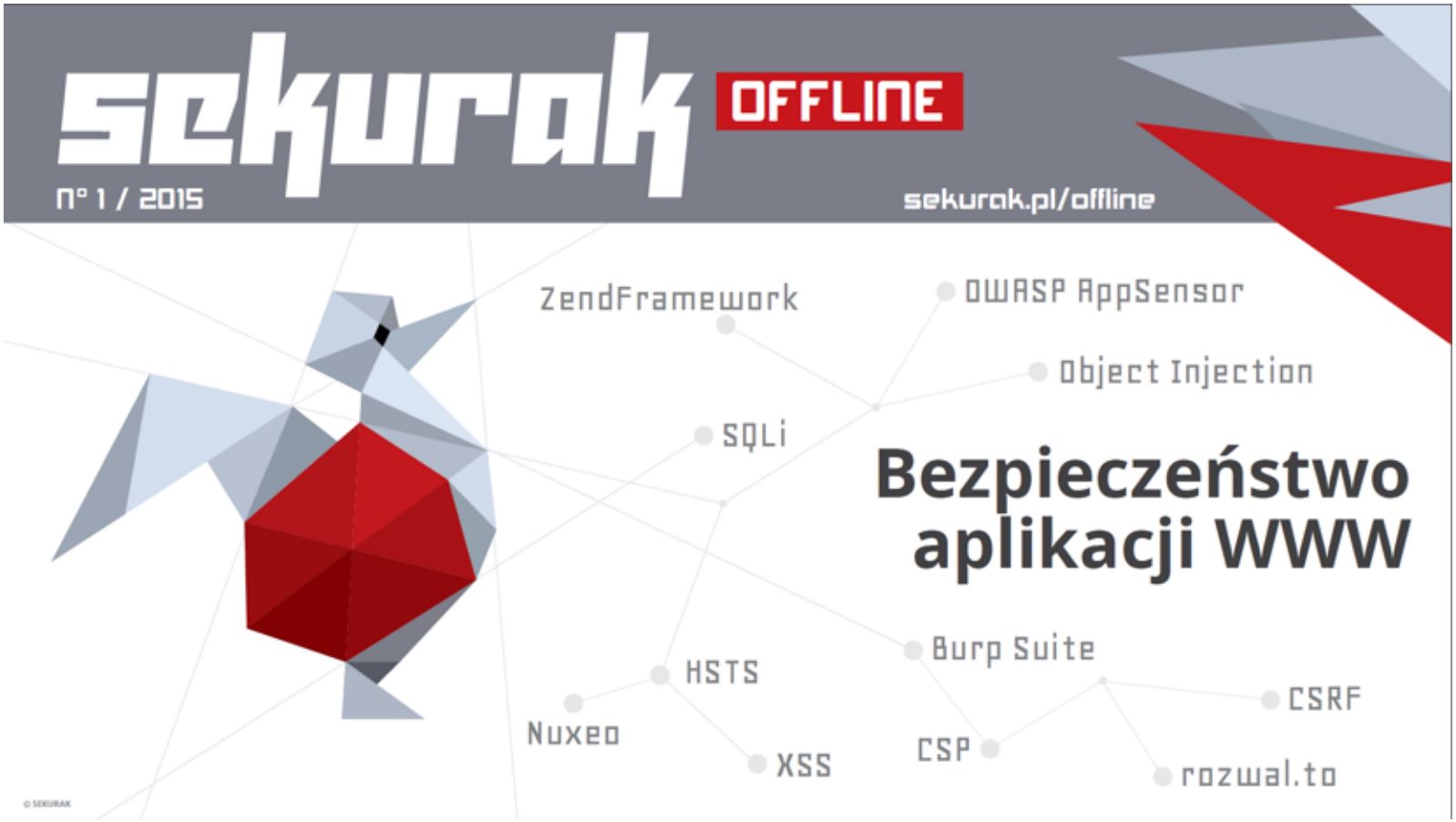
Zauważmy również, że gdyby bankowość przyjmowała requesty HTTP tylko metodą POST – na eeeevil-zite.com moglibyśmy po prostu użyć odpowiednio spreparowany i samoczynnie wysyłający się formularz typu POST. Zatem **korzystanie tylko z requestów typu POST nie chroni przed CSRF**. W tym przypadku OWASP podaje taki prosty przykład:

```
1. <body
     onload="document.forms[0].submit()">
2. <form
     action="http://bank.com/transfer.do"
     method="POST">
```

sekurak.pl/sekurak-offline/

- #1 numer wydany w te wakacje
- Treści dla początkujących / zaawansowanych
- Treści jak chronić aplikacje / jak znajdować luki
- Sumaryczna liczba pobrań: ~20 000

DEMO


sekurak
OFFLINE
Nº 1 / 2015
sekurak.pl/offline
ZendFramework
OWASP AppSensor
Object Injection
SQLi
Bezpieczeństwo aplikacji WWW
Burp Suite
HSTS
CSRF
Nuxeo
CSP
XSS
rozwal.to
© SEKURAK

sekurak.pl/sekurak-offline/

- Plany na przyszłość
- #2 – bezpieczeństwo aplikacji www
  - Kilka tekstów o podstawach
  - Błąd sec-high w Firefox (niedawno spatchowany)
    - Przyznane bounty $5000 :]
  - Błąd w dotnetnuke (full OS shell)
  - Bezpieczeństwo mechanizmów uploadu
  - Server Side Request Forgery
  - …

sekurak.pl/sekurak-offline/

- Plany na przyszłość
- #3 – monitoring bezpieczeństwa IT
  - Reagowanie na incydenty (w tym pełna procedura reagowania)
  - Wybrane case-y obsługi incydentów
  - Jak zorganizować monitoring od zera (technicznie / proceduralnie)
  - Analiza przepływów sieciowych
  - IDS-y
  - Narzędzia
  - …

sekurak.pl/sekurak-offline/

- Zapraszam do pobrania i dzielenia się magazynem ze znajomymi ☺
- Chcesz współpracować?
  - michal.sajdak@sekurak.pl
- Kolejne numery?
  - Można czytać na bieżąco sekurak.pl

- Informacja na maila (przed oficjalną premierą kolejnego numeru):
  - http://sekurak.pl/offline-zapisz/
  - Obecnie ok 1800 potwierdzonych osób