

# WEBFOCUS Security Issue 01

Wersja 0.1 19-08-2009

Wersja 0.2 04-11-2009

## Blip API CSRF

API serwisu Blip.pl posiada poważny brak w zabezpieczeniach, pozwalający na przejęcie kontroli nad każdym kontem poprzez skierowanie ofiary na specjalnie przygotowaną stronę WWW.

### Błąd krytyczny

#### Opis:

Kierując aktualnie zalogowanego w serwisie użytkownika na specjalnie przygotowaną stronę, możliwe jest wykonanie każdej operacji API w imieniu danego użytkownika bez konieczności uwierzytelniania. Jest to klasyczny błąd CSRF (Cross-Site Request Forgery<sup>1</sup>).

#### Konfiguracje podatne na atak:

- Każda przeglądarka internetowa z włączonym JavaScriptem – możliwość odczytywania wiadomości prywatnych
- Każda przeglądarka internetowa z pluginem Adobe Flash Player – możliwość odczytywania wiadomości prywatnych, wysyłania wiadomości prywatnych, zmiany awatara, zmiana tła i inne.

#### Historia:

Luka istnieje prawdopodobnie od początku działania API Blipa.

---

<sup>1</sup> Więcej o CSRF: <http://seclab.stanford.edu/websec/csrf/>

## Szczegółowy opis

API serwisu Blip<sup>2</sup> udostępnia programistom szereg operacji wykorzystujących zasoby serwisu. Część z nich wymaga uwierzytelnienia, gdyż odwołuje się do zasobów prywatnych związanych z konkretnym kontem użytkownika. Operacje API udostępnione są jako wywołania REST, np.: <http://api.blip.pl/statuses?limit=1> pobiera ostatni status aktualnie zalogowanego użytkownika. Uwierzytelnianie w API działa na dwa sposoby:

- wykorzystując schemat autoryzacji HTTP Basic, w którym login i hasło przesyłane są przez nagłówek HTTP,
- poprzez mechanizm ciasteczek (cookies): aktualnie zalogowany na Blipie użytkownik nie musi wykonywać uwierzytelniania HTTP Basic jeśli zapytanie będzie zawierać prawidłowe ciastko (zmienna `_blip_session`). Ta funkcja jest nieudokumentowana w API, jednakże ciasteczka zalogowanej osoby ważne są na całą domenę `.blip.pl`, a zatem domenę `api.blip.pl` również.

Dostęp do zasobów API możliwy jest również z poziomu aplikacji Flash znajdujących się na dowolnych stronach, niezależnie od lokalizacji (tzw. Cross-domain Access). API świadomie daje taką możliwość, pozwalając na większą różnorodność tworzonych mash-upów. Definicja dostępu dla aplikacji Flash znajduje się w pliku <http://api.blip.pl/crossdomain.xml>

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" />
  <site-control permitted-cross-domain-policies="all"/>
  <allow-http-request-headers-from domain="*" headers="*" />
</cross-domain-policy>
```

Flash Player sprawdza plik `crossdomain.xml` za każdym razem, gdy aplikacja Flash próbuje pobrać dane z serwera znajdującego się w innej domenie. Jeśli plik `crossdomain.xml` pozwala na dostęp do serwera z domeny, na której znajduje się aplikacja Flash, wtedy dane mogą być pobrane.

W tej chwili API pozwala na dostęp aplikacjom Flash z każdej domeny.

Obecnie wszystkie najpopularniejsze przeglądarki współdzielą sesje w obrębie wszystkich otwartych zakładek lub okien. Oznacza to, że nie musimy logować się wielokrotnie chcąc uruchomić ten sam serwis w kilku oknach. Z drugiej strony, po zalogowaniu do serwisu Blip, możemy w drugim oknie otworzyć nową sesję z API bez konieczności uwierzytelnienia.

Problem pojawia się w momencie, kiedy zalogowana na Blipie osoba zostanie skierowana na stronę, która wywoła (np. przy pomocy JS, jako *cross-site access*) jakąś z metod API. Ponieważ użytkownik posiada już uwierzytelnioną sesję z domeną `blip.pl`, przeglądarka wykorzysta tę sesję do uwierzytelnienia dostępu do API. W konsekwencji wywołanie metody się powiedzie, jednak taka strona internetowa może mieć złe zamiary.

---

<sup>2</sup> <http://blip.pl/api-0.02.html>

## Przykładowe scenariusze wykorzystania luki

### A) Odczytanie wiadomości prywatnych

Poniższy kod:

```
<script src="http://api.blip.pl/private_messages?callback=store">
</script>
```

odczytuje 50 ostatnich prywatnych statusów aktualnie zalogowanej na blipie osoby. Funkcja `store` może służyć do zapisania tych statusów na obcym serwerze, w celu późniejszego przeczytania. W dodatku można takie wywołanie powtarzać zmieniając parametr `offset`, w celu przeczytania wszystkich starszych wiadomości. Czytanie wiadomości przez API wymaga tylko wywołań HTTP GET, zatem można tego dokonać wykorzystując wyłącznie JavaScript.

### B) Ustawianie dowolnego statusu

Przy pomocy dodatkowego obiektu Flash można obejść ograniczenia JavaScriptu i wywołać metody API inne niż HTTP GET (np. POST, DELETE, PUT). Co prawda sam Flash w przeglądarce pozwala wywołać jedynie metody GET i POST, jednak API Blipa pozwala zasymulować PUT oraz DELETE przy pomocy metody POST dodając dodatkowy parametr `_method=metoda`.

### C) Blip Worm

Wysyłając obserwowanym osobom wiadomość z linkiem kierującym na spreparowaną stronę, można wywołać lawinę nowych wiadomości. Każda kolejna osoba wchodząca na taką stronę wysyłałaby nieświadomie wiadomość do swoich obserwowanych, lub osób obserwujących wybrany tag. W konsekwencji można łatwo i bardzo szybko rozprzestrzenić takiego robaka wśród dużej grupy osób korzystającej z serwisu<sup>3</sup>. Ten scenariusz jest bardzo groźny jeśli zostanie połączony z możliwością odczytywania wiadomości prywatnych.

## Sugerowane kroki w celu rozwiązania problemu

1. Wprowadzenie uwierzytelnienia z użyciem HTTP Auth (zamiast cookies) dla wszystkich krytycznych z punktu widzenia prywatności operacji API, tj: wszystkie metody na zasobie `/private_messages` oraz metody POST, PUT, DELETE na wszystkich pozostałych zasobach.
2. Sprawdzenie pola Referer w nagłówkach HTTP.
3. Ograniczenie dostępu dla aplikacji napisanych we Flashu. w pliku `crossdomain.xml`
4. Użycie dodatkowego mechanizmu challenge-response przy uwierzytelnianiu zapytań API.

---

<sup>3</sup> Podobny problem pojawił się w serwisie mikroblogowym Śledzik. Mimo, że serwis ten nie posiadał luki pozwalającej wysłać wiadomość automatycznie (wiadomość musiała być zaakceptowana przed wysłaniem), to jednak dzięki efektowi kuli śnieżnej wywołał sporą lawinę wiadomości.