Operatora płatności jakim jest Dotpay Spółka z o.o. zna prawie każdy, kto chociaż raz robił zakupy przez Internet. Dziś postanowiłem opisać sytuację, w której jak się okazuje za transakcję można zapłacić jeden grosz.

Dotpay jako operator płatności w swoim regulaminie zobowiązuje się do tego, by wszelakie płatności prowadzone poprzez dostarczaną platformę płatności były spójne i bezpieczne.

Jednym z sposobów na weryfikację i kontrolę transakcji jest suma kontrolna.

„**Suma kontrolna** (ang. checksum) – liczba uzyskana w wyniku sumowania lub wykonania innych operacji matematycznych na przesyłanych danych, przesłana razem z danymi i służąca do sprawdzania poprawności przetwarzanych danych. Komputer wysyłający dane oblicza ich **sumę kontrolną** i dołącza ją do pakietu danych."

Ostatnim czasem w związku z tworzeniem nowej strony na potrzeby BSides Warsaw zrodził się pomysł stworzenia nowego systemu biletowego. Wybór operatora płatności był mocno ograniczony. Mogłem wybrać Dotpay lub PayU. Jak się okazało po krótkim reserczu jedynie Dotpay spełniał postawione przeze mnie wymagania. Dodatkowym atutem było to, że już raz implementowałem rozwiązania płatności dostarczane przez Dotpay.

Jako, że miałem już gotową implementację wystarczyło ją tylko podpiąć i przetestować. Oczywiście implementacja musiała spełniać jeden dodatkowy warunek. Mieć załatane wszystkie bugi, ponieważ będzie wystawiona do użytku uczestników, jakby nie patrzeć, konferencji o tematyce bezpieczeństwa informatycznego. Niech pierwszy rzuci kamieniem ten z nas, który nigdy nie próbował „obejść" aplikacji, z których korzystał. A może tylko ja mam tendencje do sprawdzania funkcjonowania takich rzeczy. Nieważne czy jest to aplikacja do sprzedaży biletów PKP czy portal do zamawiania jedzenia.

Postanowiłem zacząć od dokładnego przewertowania dokumentacji platformy transakcyjnej dostarczonej przez Dotpay. Oczywiście kilka razy już ją przeglądałem. Tym razem jednak poszukiwałem możliwości na zmianę kwoty transakcji w taki sposób, aby płatność wynosiła 1 grosz oraz aby udało się zakończyć płatność powodzeniem, tak by system biletowy uznał ze zapłaciłem taką kwotę jaką powinienem.

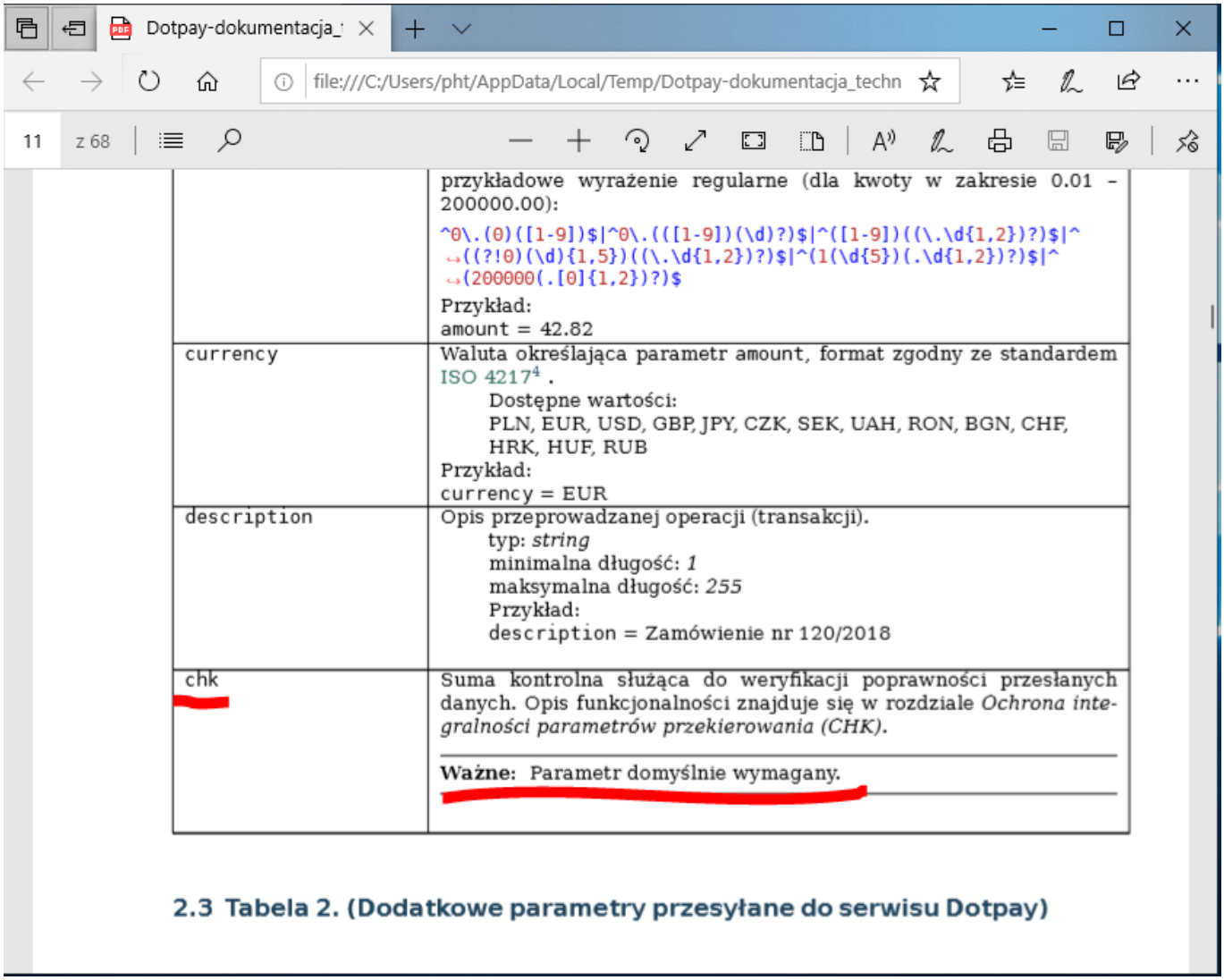
W związku z tym, iż w oparciu o moje dotychczasowe doświadczenia – większość testów robiłem jednak na wersji testowej – modyfikacja jednego parametru jest niemożliwa w związku z wymuszaniem przez Dotpay przesłania razem z danymi również sumy kontrolnej. Oczywiście PRAWIE wszystkie dane do wyliczenia tej sumy mamy w komunikacji,

którą możemy podsłuchać i przeanalizować.

Na potrzeby tego artykułu stworzyłem prosty formularz, który przesyła określone dane do serwisu Dotpay. Z założenia środowisko testowe powinno być identycznym z środowiskiem produkcyjnym, po to by opracowując poprawki, nowe rozwiązania czy też pisząc implementację danej funkcji móc testować ją dowoli w warunkach odzwierciedlających te prawdziwe „bojowe" sytuacje. Biorąc pod uwagę moją wrodzoną przekorę i lekko anarchistyczne podejście do standaryzacji założyłem, właściwe jak zawsze, że:

- Środowisko testowe nie jest takie samo jak środowisko produkcyjne, bo ktoś zapomniał kliknąć aktualizuj lub z innego trywialnego powodu
- Potencjalny „haker", gdyby szukał bugów, działał by na środowisku testowym dlatego, że ewentualne kombinacje nie zwróciłyby niczyjej uwagi, bo to przecież jest środowisko testowe
- Środowisko produkcyjne ma jakiś bug – to akurat w myśli mojego motta „No system is safe"

Jak już wcześniej wspomniałem do poprawnego przesłania danych i ich zaakceptowania przez platformę Dotpay wymagane jest obliczenie i przesłanie sumy kontrolnej. W tym przypadku zgodnie z dokumentacją jest to parametr nazwany „chk".



	przykładowe wyrażenie regularne (dla kwoty w zakresie 0.01 - 200000.00): `^0\.(0)([1-9])$\|^0\.(([1-9])(\d)?)$|^([1-9])((\.\d{1,2})?)$\|^((?!0)(\d){1,5})((\.\d{1,2})?)$|^(1(\d{5})(.\d{1,2})?)$\|^(200000(.[0]{1,2})?)$` Przykład: amount = 42.82
currency	Waluta określająca parametr amount, format zgodny ze standardem ISO 4217[4] . Dostępne wartości: PLN, EUR, USD, GBP, JPY, CZK, SEK, UAH, RON, BGN, CHF, HRK, HUF, RUB Przykład: currency = EUR
description	Opis przeprowadzanej operacji (transakcji). typ: *string* minimalna długość: *1* maksymalna długość: *255* Przykład: description = Zamówienie nr 120/2018
chk	Suma kontrolna służąca do weryfikacji poprawności przesłanych danych. Opis funkcjonalności znajduje się w rozdziale *Ochrona integralności parametrów przekierowania (CHK)*. **Ważne:** Parametr domyślnie wymagany.

2.3 Tabela 2. (Dodatkowe parametry przesyłane do serwisu Dotpay)

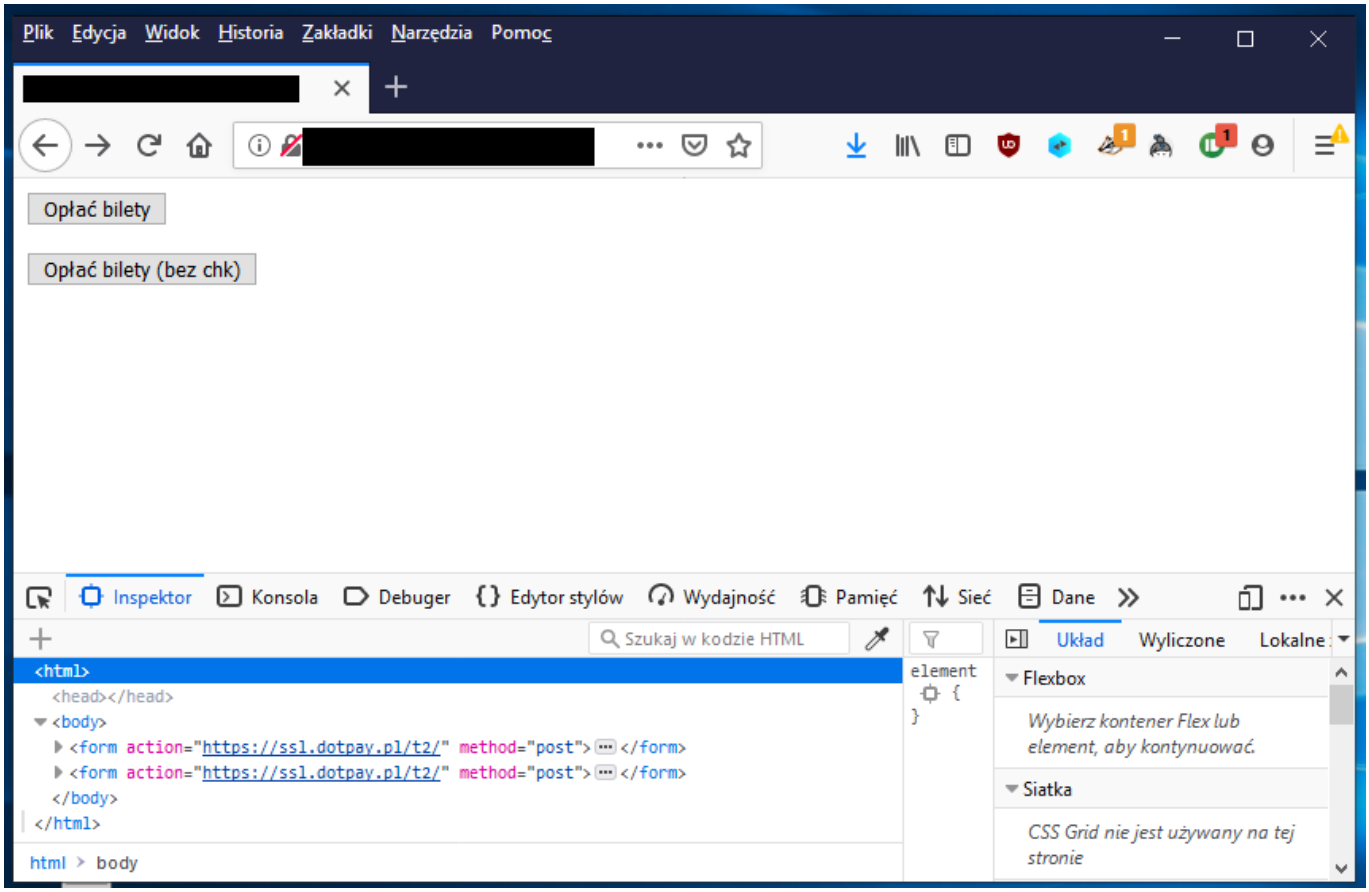
Oczywiście pierwszą rzeczą za jaka się zabrałem było spreparowanie skryptu, który będzie przesyłał dane. Zacząłem od wersji, która była w pełni zgodna z założeniami dokumentacji.

```php
function payment($amount, $description, $control)
{
    $PIN = [redacted];
    $LANG = "PL";
    $ID = [redacted];
    $amount = str_replace(",",".",$amount);
    $AMOUNT = "$amount";
    $CURRENCY = "PLN";
    $DESCRIPTION = "$description";
    $CONTROL = $control;
    $CHANNEL_GROUPS = "K,T,P,I,G";
    $URL = [redacted];
    $TYPE = 0;
    $IGNORE_LAST_PAYMENT = "1";

    $chk =
        $PIN.
        $LANG.
        $ID.
        $AMOUNT.
        $CURRENCY.
        $DESCRIPTION.
        $CONTROL.
        $CHANNEL_GROUPS.
        $URL.
        $TYPE.
        $IGNORE_LAST_PAYMENT;

    $chk = hash('sha256', $chk);

    echo "<form action=\"https://ssl.dotpay.pl/t2/\" method=\"post\">
    <input type=\"hidden\" name=\"id\" value=\"$ID\"/>
    <input type=\"hidden\" name=\"amount\" value=\"$AMOUNT\"/>
    <input type=\"hidden\" name=\"currency\" value=\"$CURRENCY\"/>
    <input type=\"hidden\" name=\"description\" value=\"$DESCRIPTION\"/>
    <input type=\"hidden\" name=\"channel_groups\" value=\"$CHANNEL_GROUPS\"/>
    <input type=\"hidden\" name=\"control\" value=\"$CONTROL\"/>
    <input type=\"hidden\" name=\"ignore_last_payment_channel\" value=\"1\"/>
    <input type=\"hidden\" name=\"type\" value=\"$TYPE\"/>
    <input type=\"hidden\" name=\"lang\" value=\"$LANG\"/>
    <input type=\"hidden\" name=\"chk\" value=\"$chk\"/>
    <input type=\"hidden\" name=\"url\" value=\"$URL\"/>
    <button type=\"submit\" class=\"btn btn-primary btn-block btn-lg ".$status."\" value=\"Opłać\">
        Opłać bilety
    </button>
</form>";
```

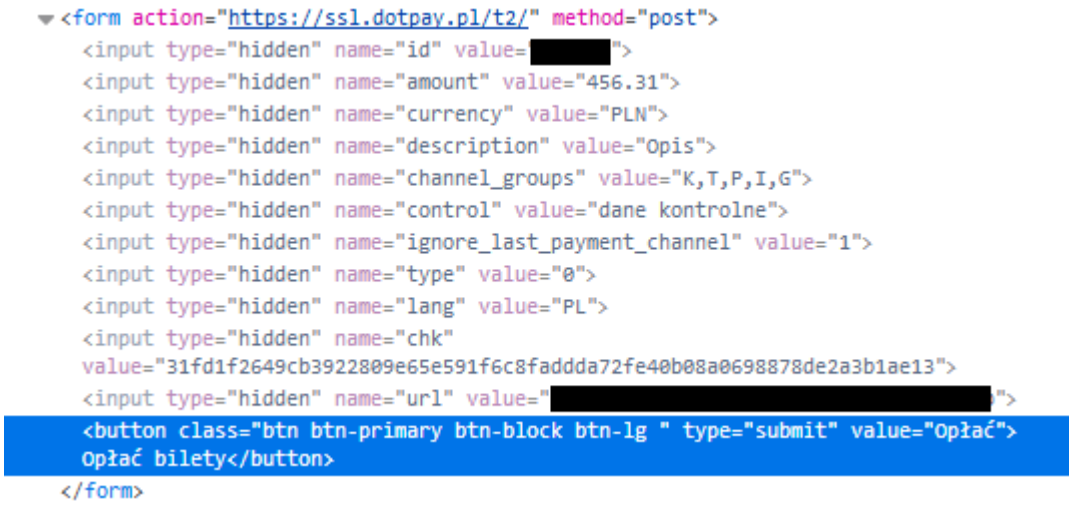
Postanowiłem przetestować też wersję, która będzie mniej zgodna z dokumentacją.
Więc przygotowałem funkcje bez parametru „chk”.

```php
function payment_w_chk($amount, $description, $control)
{
    $PIN = [redacted]
    $LANG = "PL";
    $ID = [redacted]
    $amount = str_replace(",",".",$amount);
    $AMOUNT = "$amount";
    $CURRENCY = "PLN";
    $DESCRIPTION = "$description";
    $CONTROL = $control;
    $CHANNEL_GROUPS = "K,T,P,I,G";
    $URL = [redacted]
    $TYPE = 0;
    $IGNORE_LAST_PAYMENT = "1";

    $chk =
        $PIN.
        $LANG.
        $ID.
        $AMOUNT.
        $CURRENCY.
        $DESCRIPTION.
        $CONTROL.
        $CHANNEL_GROUPS.
        $URL.
        $TYPE.
        $IGNORE_LAST_PAYMENT;

    $chk = hash('sha256', $chk);

    echo "<form action=\"https://ssl.dotpay.pl/test_payment/\" method=\"post\">
    <input type=\"hidden\" name=\"id\" value=\"$ID\"/>
    <input type=\"hidden\" name=\"amount\" value=\"$AMOUNT\"/>
    <input type=\"hidden\" name=\"currency\" value=\"$CURRENCY\"/>
    <input type=\"hidden\" name=\"description\" value=\"$DESCRIPTION\"/>
    <input type=\"hidden\" name=\"channel_groups\" value=\"$CHANNEL_GROUPS\"/>
    <input type=\"hidden\" name=\"control\" value=\"$CONTROL\"/>
    <input type=\"hidden\" name=\"ignore_last_payment_channel\" value=\"1\"/>
    <input type=\"hidden\" name=\"type\" value=\"$TYPE\"/>
    <input type=\"hidden\" name=\"lang\" value=\"$LANG\"/>
    <input type=\"hidden\" name=\"url\" value=\"$URL\"/>
    <button type=\"submit\" class=\"btn btn-primary btn-block btn-lg ".$status."\" value=\"Opłać\">
        Opłać bilety (bez chk)
    </button>
</form>";


}
```

Przygotowałem też prosty index.php, który miał służyć jako wyzwalacz zarówno dla funkcji zgodnej jak i nie zgodnej z dokumentacją. W założeniu funkcja zgodna miała mnie zaprowadzić na stronę wyboru metody płatności, a ta bez chk – spowodować błąd.



Oczywiście, zgodnie z moimi przewidywaniami, wersja zgodna z dokumentacją zadziałała – to było oczywiste. Mniej oczywiste było to co wydarzyło się potem. Jako zwolennik metodologii badawczych mówiących, iż zawsze trzeba sprawdzić co najmniej dwie możliwości, postanowiłem spróbować „zapłacić” przy pomocą funkcji pozbawionej parametru „chk” co w założeniu i odniesieniu do dokumentacji powinno być niemożliwe. Jak się okazało – było. Jednakże nie uprzedzajmy faktów.

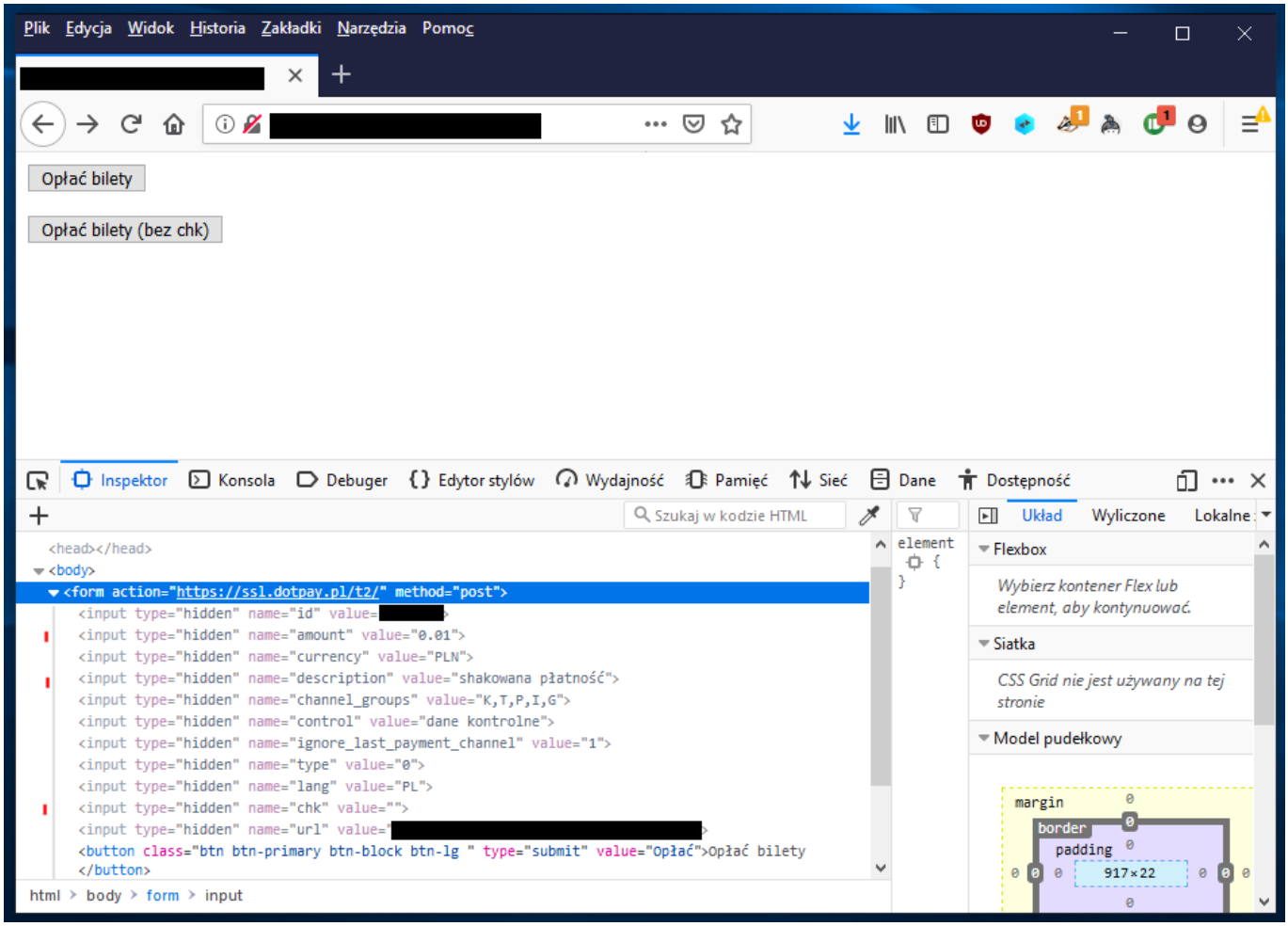


Na powyższym obrazku widać, jak wygląda formularz służący do przesyłania danych do platformy. Zawiera on wszystkie niezbędne dla poprawnego działania parametry. Między innymi generowaną za pomocą funkcji skrótu sha-256 sumę kontrolną. Ta funkcja działa bez zarzutu.

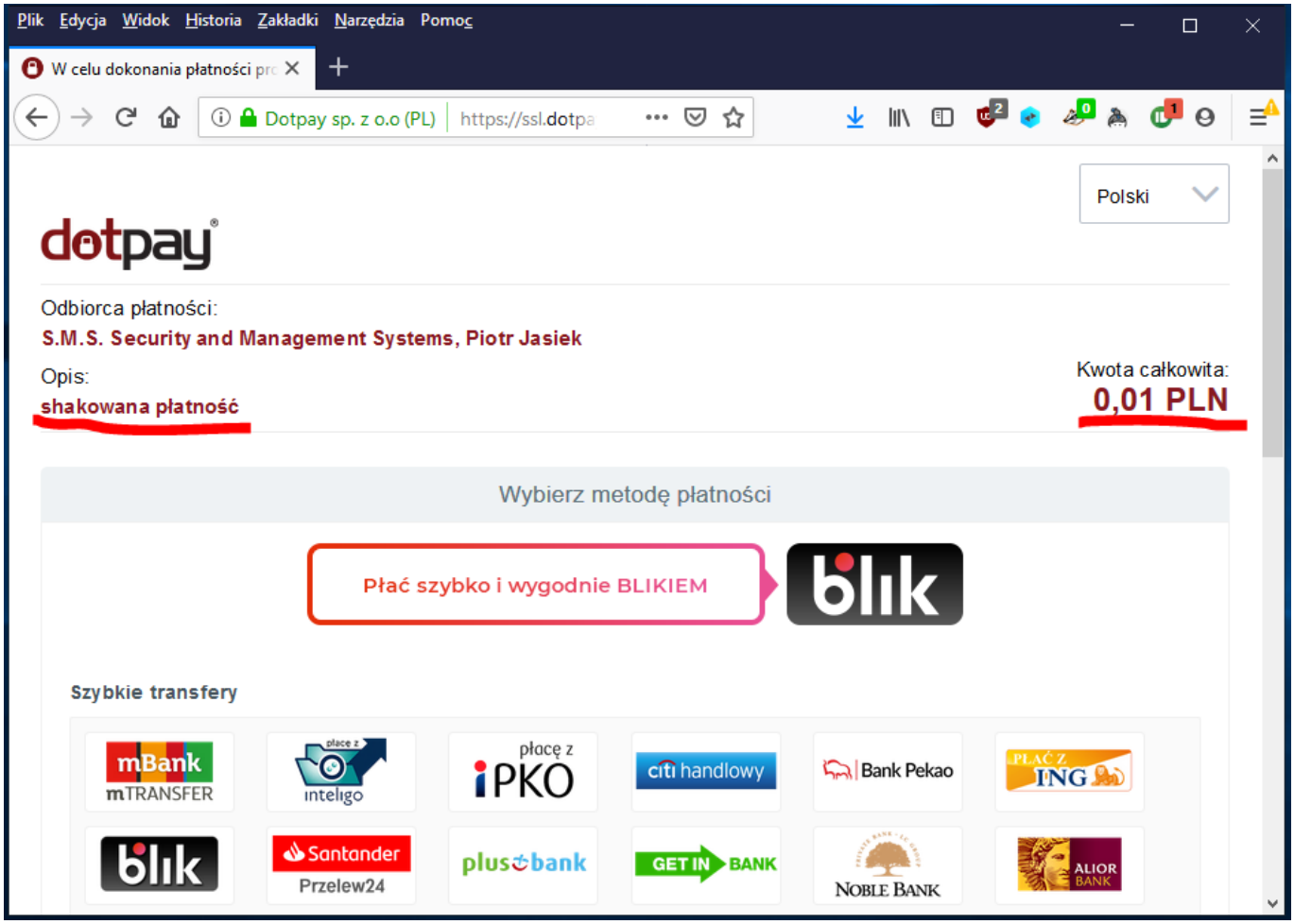
Jak pisałem wcześniej przygotowałem też funkcję bez tego parametru. Jej formularz prezentuje się poniżej.

Jak widać różnią się one jedynie obecnością parametru chk. Będąc pewnym, że ta funkcja zaprowadzi mnie do komunikatu o błędzie kliknąłem „Opłać”. Mojemu zdziwieniu nie było końca, kiedy okazało się, że nie zostałem przekierowany do ekranu błędu a do ekranu płatności, tak jakby nie wystąpił żaden błąd.

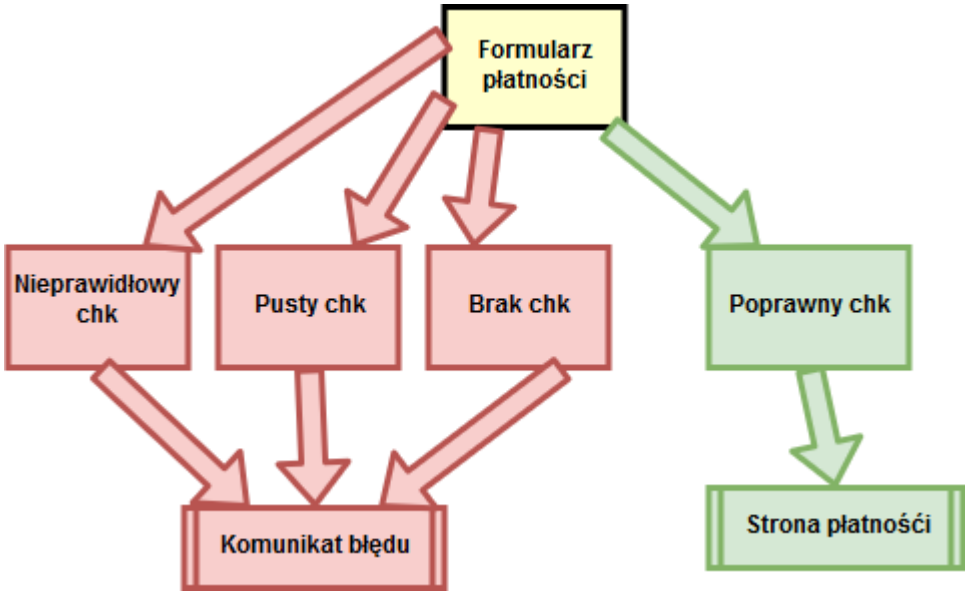
Idąc tym tropem postanowiłem zmodyfikować dane pierwszej funkcji.



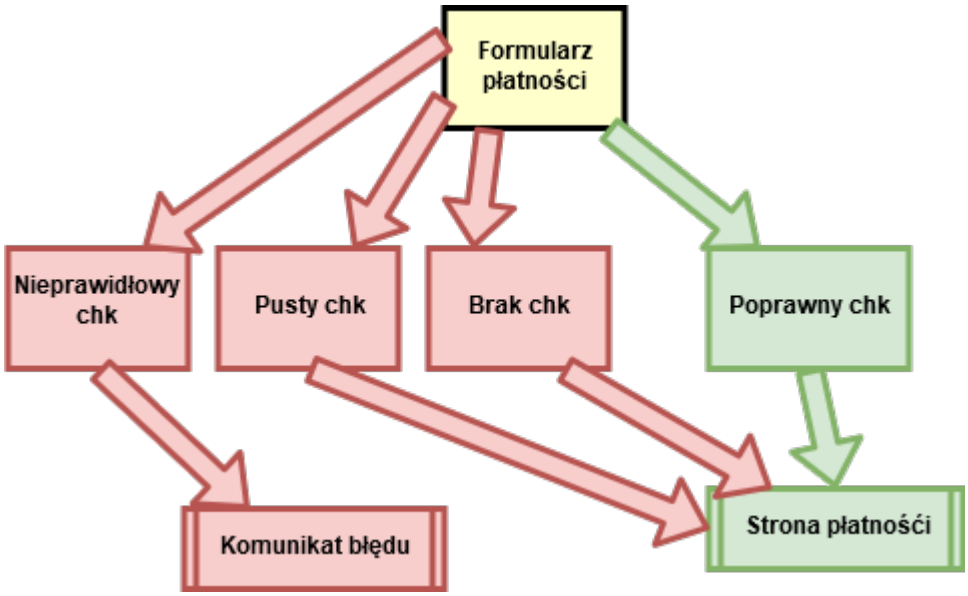
O dziwo, całkowicie wbrew dokumentacji, moim oczom ponownie ukazał się ekran płatności ze spreparowanymi przeze mnie danymi.



Jak widać, Dotpay na platformie produkcyjnej nie tylko nie weryfikuje zawartości parametru „chk", ale i nie weryfikuje tego czy został on wysłany. Kolejną ciekawostką jest to, że w przypadku podania parametru sumy kontrolnej, ale uzupełniania go nieprawidłowymi danymi otrzymałem komunikat o błędzie co oznacza, że procedura weryfikacji jest bardzo dziwna. Dla wyjaśnienia przygotowałem schemat przebiegu tego procesu. Na początek jednak jak powinno to wyglądać:



Oraz jak wygląda to w rzeczywistości:



Oczywiście, płatność poprzez platformę Dotpay nie odbywa się tylko poprzez jeden etap i możliwym jest w bardzo prosty sposób zabezpieczenie się przed tego typu błędami czy też próbami zmiany kwoty. Z drugiej strony jednak, jeśli dokumentacja zapewnia mnie o tym, że istnieje parametr zapewniający integralność i nienaruszalność danych, które są przesyłane do serwisu transakcyjnego to nie muszę już ich weryfikować.

Otóż nic bardziej mylnego. W dokumentacji Dotpay w rozdziale pod tytułem „ODBIERANIE

INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC)” można znaleźć następującą informację:

> **Ostrzeżenie:** Brak weryfikacji wartości parametrów `amount`, `currency` oraz `signature` po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe.

Oznaczać może to, że Dotpay doskonale zdaje sobie sprawę z sytuacji, w której możliwa jest modyfikacja kwoty, a parametr sumy kontrolnej nie spełnia zamierzonego efektu.

Jak działa proces płatności Dotpay? W skrócie –

1. Formularz na stronie www sklepu zbiera dane
2. Dane zostają przekazane do panelu wyboru płatności
3. Płatnik zostaje przekierowany do banku w celu zapłacenia
4. Po zapłaceniu następuje przeslanie potwierdzenia płatności do sklepu

Na tym ostatnim etapie sprzedawca może weryfikować, czy zgadza się suma kontrolna oraz kwota. Co prawda Dotpay jasno przestrzega na temat weryfikacji wyżej wymienionych, lecz co w sytuacji gdyby niefrasobliwy programista zapomniał o tym lub nie doczytał… Podstawowym pytaniem, które rodzi się w tym momencie, jest to, czemu tak zaawansowana platforma transakcyjna nie jest jednak „idioto-odporna” i czemu środowisko testowe tak bardzo różni się od produkcyjnego. Można by przyjąć, iż dokumentacja jasno mówi o konieczności weryfikacji parametrów nie mniej jednak sam na początku nabrałem się na to, jak przedstawia sytuacje środowisko testowe.

Co na to Dotpay?

Kilka dni temu, kiedy zebrałem wszystko co zauważyłem w całość skontaktowałem się z Dotpayem. Wysłałem mail na adres kontaktowy bok@dotpay.pl i poprosiłem o kontakt z osobą techniczną odpowiedzialną za bezpieczeństwo. Opisałem fakt znalezienia przeze mnie problemu ze spójnością. Już po kilku godzinach skontaktował się ze mną jeden z pracowników Dotpay. Zostałem zapewniony, iż każde zgłoszenie jest traktowane przez nich poważnie i z dużą dokładnością. Konkretną odpowiedź otrzymałem w ciągu tygodnia.

> Jako Dotpay staramy się wychodzić na przeciw oczekiwaniom naszych klientów dostarczając bezpieczne, a jednocześnie proste w integracji rozwiązania

płatnicze. Dokładając wszelkich starań po naszej stronie staramy się wspierać klientów na każdym etapie integracji, jak również
zapewniamy pełne wsparcie techniczne w trakcie współpracy. Mimo to, zdarzało się że nasi Klienci nie stosowali najwyższych poziomów zabezpieczeń, które zawsze rekomendujemy.

Sytuacja, która została opisana jest nam oczywiście znana i od dłuższego już czasu pracujemy z naszymi kontrahentami nad minimalizacją ryzyk w tym obszarze.

W związku z tym po naszej stronie prowadzone są cyklicznie działania o charakterze edukacyjnym – poprzez komunikację mailową, informacje w Panelu klienta, czy bezpośredni kontakt opiekuna biznesowego. Okresowo przeprowadzamy kampanie informacyjne zachęcające do weryfikacji modelu wdrożenia i aktualizacji poziomu zabezpieczeń po stronie sklepu, która obejmie swoim zasięgiem Akceptantów współpracujących z Dotpay. Prowadzimy także regularne aktualizacje dokumentacji, które większy nacisk kładą
na wyszczególnienie obszaru związanego z bezpieczeństwem.

Wchodząc w szczegóły analizy opisanej sytuacji, jest jednak odrobinę inaczej, niż wynika to z artykułu. Przede wszystkim w połowie zeszłego roku zmienione zostały domyślne konfiguracje nowo tworzonych kont w Dotpay. Od tego czasu każde konto testowe i produkcyjne zachowuje się
tak, jak to obserwuje Pan obecnie na środowisku testowym. Na ile udało mi się znaleźć Pańskie konta na środowisku testowym, korzystał Pan z takich założonych po wspomnianym terminie, a dostęp produkcyjny uzyskał Pan wcześniej, stąd inne zachowanie, zgodne z tym opisanym w artykule.

Jako iż typowy proces wdrożenia, o który wspomina artykuł, rozpoczyna się od założenia środowiska testowego, nie powinno być więc wrażenia bezpieczeństwa (wynikającego z testów), które jest złudne (bo system produkcyjny zachowuje się inaczej). Oczywiście w każdej chwili, na Pańską prośbę, jesteśmy w stanie Pańskie konto produkcyjne przekonfigurować.

Aby podnieść bezpieczeństwo serwisu, jeśli nasz klient ma konto starsze niż rok, a jest pewien, iż komplet jego płatności zawiera podpis cyfrowy (wspomniany w artykule parametr ‘chk’), powinien poprosić nas o włączenie wymagalności podpisu cyfrowego, przesyłając maila na adres administracja@dotpay.pl.
Niezależnie od tego warto spojrzeć na sposób obsługi parametrów w notyfikacji

> zwrotnej (URLC), ze szczególnym uwzględnieniem kwoty oraz podpisu cyfrowego. W przypadku braku wiedzy technicznej proponujemy kontakt z wsparciem technicznym Dotpay (tech@dotpay.pl), gdzie zweryfikujemy ustawienia konta oraz indywidualnie przekażemy odpowiednie zalecenia.

Jak wynika z odpowiedzi, Dotpay wie o problemie i konsekwentnie stara się go zniwelować. Bardzo cieszy nas takie podejście. Mamy nadzieję, iż nie tylko Dotpay takie ma. Oczywiście większość odpowiedzialności i wysiłku spoczywa na klientach Dotpay, którzy muszę odpowiednio zaimplementować API płatności dostarczone przez Dotpay. Mimo to, mamy nadzieję, iż operatorowi płatności uda się jak najszybciej doprowadzić do momentu w którym wszystkie konta klienckie będą poprawnie skonfigurowane i zabezpieczone.

Co mam robić?

Tak jak w odpowiedzi Dotpay. Jeśli jesteś ich klientem i nie masz pewności czy ten błąd występuje również u Ciebie możesz skontaktować się ze mną wysyłając mail na adres piotr.jasiek@blog.s-m-s.pl. Pomogę Ci zweryfikować czy Twoja implementacja Dotpay na pewno przesyła odpowiednie parametry oraz czy wymuszenie podpisu cyfrowego jest włączone.