Obrona

Opera – analiza działania mechanizmu ochrony przed oszustwami

Marcin Kopeć

stopień trudności

Od kilku lat wszyscy jesteśmy świadkami nowej ery w historii globalnej sieci Internet. Otóż narzędzie, które początkowo miało służyć rozwojowi myśli technicznej, w momencie wkroczenia weń wielkiego biznesu spowodowało pociągniecie za nim osób chcących w łatwy i szybki sposób wzbogacić się cudzym kosztem – mowa tu o internetowych przestępcach.

Zapewne sam Sir Timothy Berners-Lee, tworząc podwaliny pod protokół HTTP, nie spodziewał się, że jego dziecko będzie w przyszłości groźną bronią w rękach zwykłych kryminalistów.

Celem poniższego artykułu będzie przyjrzenie się, w jaki sposób radzi sobie popularny klient protokołu HTTP – przeglądarka internetowa Opera – w ochronie nieświadomych użytkowników Sieci przed staniem się ofiarą internetowego oszustwa.

Zagrożenia

Jak wygląda oszustwo w ujęciu protokołu HTTP? Otóż w najprostszej implementacji polega ono na stworzeniu przez cyberprzestępcę spreparowanego serwisu internetowego danej instytucji. Najczęściej podrabia się witryny instytucji, w których klienci dokonują różnego rodzaju transakcji elektronicznych o charakterze finansowym – tzw. *e-commerce* (np. banki internetowe, serwisy aukcyjne).

Zadaniem spreparowanego serwisu jest najczęściej gromadzenie wrażliwych danych użytkowników, takich jak hasła dostępu czy kody autoryzacyjne do systemów bankowości internetowej (w celu późniejszego logowania się do nich i wyprowadzania środków pieniężnych z rachunków), czy też danych osobowych (aby móc je wykorzystać do innych form przestępstw, takich jak kradzieże tożsamości w celu np. dokonania wymuszenia). Cyberprzestępcom zdarza się zamieszczać na łamach fałszywych serwisów zwykłe prośby o dokonanie wpłat na konkretny rachunek bankowy. Przykładem może być tu fałszywy sklep internetowy, w którym należności za kupno towarów są regulowane na odmienny w stosunku do oryginalnego rachunek bankowy.

Z artykułu dowiesz się

- w jaki sposób działa mechanizm ochrony przed oszustwami zaimplementowany w przeglądarce internetowej Opera,
- czy informacja dostarczana użytkownikowi przez mechanizm ochronny charakteryzuje się dostatecznym poziomem rzetelności?

Co powinieneś wiedzieć

- znajomość protokołu HTTP – RFC 2616,
- postawy programowania w PHP, HTML.

Fałszywe serwisy internetowe, które przy wykorzystaniu metod socjotechnicznych starają się dokonać wyłudzenia, bywają najczęściej umieszczane na serwerach znajdujących się w krajach, w których obowiązujące regulacje prawne nie pozwalają na sprawną interwencję organów ścigania w przypadku otrzymania zgłoszenia z innego kraju o próbie bądź popełnieniu przestępstwa – dobrym przykładem są tu nasi wschodni sąsiedzi.

Powszechnie stosowaną metodą na uwiarygodnienie fałszywego serwisu internetowego jest zamieszczanie go pod adresem domenowym o łudząco podobnej do oryginalnej nazwie. Przykładowo gdy celem cyberprzestępcy jest oszukanie klientów banku *thisismybank.com*, może on spróbować zarejestrować domeny pod fałszywą witrynę z wykorzystaniem między innymi takich nazw jak: *thisis-mybank.com* (dodano znak rozdzielający nazwę), *thisissmybank.com* (dodano dodatkową literę), *thlsismybank.com*, *th1sismybank.com* (zamieniono jedną literę na inną, bądź na liczbę czy znak specjalny), czy tworząc poddomenę zawierającą w członie nazwę oryginalną jak np. *tihisismybank.com.xyzxyz.com*. Oszukańcze serwisy są zwykle udostępniane z wykorzystaniem *http://* w przeciwieństwie do oryginalnych, udostępnianych po *https://* wszystko po to, aby nie wzbudzić u użytkownika podejrzeń komunikatem o nieprawidłowym certyfikacie SSL.

Najczęściej, aby sprowokować klienta do odwiedzin fałszywej witryny, przestępcy stosują techniki spammingu – czyli masowego wysyłania wiadomości (np. przy pomocy protokołu poczty elektronicznej bądź komunikatorów internetowych), które zawierają informację mającą nakłonić adresata do odwiedzin. Wśród treści tych wiadomości znaleźć można wyjątkowo okazyjną ofertę specjalną – promocje, informację o zwycięstwie w konkursie czy też prośbę o potwierdzenie hasła. Czasami można odnaleźć link do oszukańczej strony zamieszczony w publicznym serwisie internetowym np. jako post na forum bądź komentarz w blogu.

Listing 1. *Zapytanie do serwera weryfikującego (tryb automatyczny)*

```
GET /?host=www.hakin9.org&ph=CrmIXGGAlC7mS455Q2szhQ==
&hdn=hz3g35Z4i/6JXNonVjl4Mg==HTTP/1.1
User-Agent: Opera/9.21 (Windows NT 5.1; U; pl)
Host: sitecheck.opera.com
Accept: text/html, application/xml;q=0.9,
application/xhtml+xml, image/png, image/jpeg, image/gif,
image/x-xbitmap, */*;q=0.1
Accept-Language: pl-PL,pl;q=0.9,en;q=0.8
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1
Connection: Keep-Alive
```

Listing 2. *Prosty skrypt udający działanie serwera weryfikującego*

```
<?php
/*
 * Użycie:
 * {na komputerze z Apache/PHP}
 * - zamieść plik jako http://twojastrona.com/index.php
 * - dodaj odpowiedni wpis do pliku hosts, np:
 *   Unix:
 *   echo 'XXX.XXX.XXX.XXX sitecheck.opera.com' >>
 *   /etc/hosts
 *   Windows:
 *   echo XXX.XXX.XXX.XXX sitecheck.opera.com >>
 *      %SystemRoot%/system32/drivers/etc/hosts
 *   gdzie XXX.XXX.XXX.XXX to adres IP twojastrona.com
 */

$mode = 1;

switch ($mode) {
    case 1:
       $trust = 'NV';
       break;
    case 2:
       $trust = 'V';
       break;
    case 3:
       $trust = 'W';
       break;
}

header ('Content-type: text/xml');
echo '<?xml version="1.0" encoding="utf-8"?>
      <trustwatch version="1.0">
         <package>
          <action type="searchresponse">
    ';
echo "         <trustlevel>$trust</trustlevel>
             <host>$host</host>
             <partner>0</partner>
            <serverexpiretime>0</serverexpiretime>
            <clientexpiretime>0</clientexpiretime>
    ";
if ($mode == 3)
   echo "
                  <blacklist>
                <ph>$ph</ph>
             </blacklist>
   ";
echo "
          </action>
        </package>
     </trustwatch>
    ";
?>
```

Reakcja producentów

Aby dać użytkownikom narzędzie pozwalające stwierdzić, czy dany serwis internetowy jest wiarygodny bądź stworzony w celu malwersacji, producenci oprogramowania ochronnego, a w ślad za nimi czołowi producenci przeglądarek internetowych, przygotowali swoje rozwiązania mające realizować powyższy cel.

Mechanizmy ochrony przed oszustwami zastosowano między innymi w najnowszych wersjach popularnych przeglądarek tj. Internet Explorer 7, Mozilla Firefox 2.x, czy Opera 9.x.

W artykule skoncentrowano się na badaniu zachowania Opery 9.21 (build 8776) w wersji polskiej pod systemy Windows.

Sposoby ochrony

W Operze mechanizm ochrony przed oszustwami działa w dwóch trybach.

Pierwszy – nazwijmy go umownie *automatycznym* – polega na każdorazowej weryfikacji adresu URL przed jego wywołaniem. Jest to tryb domyślnie wyłączony. Drugi, *manualny,* jest wywoływany ręcznie przez użytkownika poprzez kliknięcie symboli stanu wiarygodności strony – litery *i,* bądź znaku „*?* w prawym rogu paska adresowego, lub w przypadku autoryzowanych serwisów, udostępnianych *via https://,* po naciśnięciu symbolu kłódki i przejściu do zakładki *Ochrona przed oszustwami*.

Tryb automatyczny

Gdy użytkownik pragnie odwiedzić interesujący go serwis, w stronę serwera pełniącego funkcję weryfikatora witryn, dostępnego pod adresem *sitecheck.opera.com*, kierowane jest metodą `GET` zapytanie HTTP zawierające wartości w następujących parametrach:

- `host` – nazwa hosta,
- `ph` – hash z URL'a,
- `hdn` – hash z nazwy hosta.

Przykład załączony jest na Listingu 1.

Ciekawa cecha powyższego mechanizmu jest związana z faktem, że prośby o weryfikację serwisów *http://* kierowane są do serwera *sitecheck.opera.com* z wykorzystaniem protokołu HTTP, który sam w sobie nie posiada funkcji pozwalającej na uwiarygodnienie strony serwera podczas transmisji, natomiast tylko w przypadku serwisów *https://* sprawdzana jest wiarygodność strony serwera przy wykorzystaniu cechy protokołu HTTPS jaką niewątpliwie jest udział w infrastrukturze klucza publicznego. Tu zastosowano umieszczony na serwerze, wygenerowany przez autoryzowane CA certyfikat SSL.

Certyfikat weryfikowany jest w sposób prawidłowy, próby podstawienia własnoręcznie wygenerowanego certyfikatu serwera kończyły się pojawieniem stosownego komunikatu o jego nieprawidłowości.

Greylisting

Z informacji udostępnionych przez firmę Opera wynika, iż serwer weryfikujący witryny, po otrzymaniu zapytania od użytkownika, sprawdza je wykorzystując technologię *greylistingu*, korzystając z baz dwóch niezależnych dostawców. Organizacja *GeoTrust* za pomocą swojego produktu *TrustWatch*, udostępnia białą listę – czyli bazę danych witryn zaufanych, oraz czarną listę – bazę witryn, co do których zachodzą podejrzenia, że zostały stworzone w celu dokonania oszustwa. Bazy *TrustWatch* tworzone są przez pracujący w *GeoTrust* zespół ekspertów, na podstawie próśb o weryfikację stron kierowanych do nich przez użytkowników ich firmowego *Toolbar'a*. Potem dokonują oni klasyfikacji witryny, a następnie umieszczają ją na którejś z list: czarnej lub białej.

Drugim źródłem, którym w procesie analizy witryn posługuje się serwer weryfikujący, jest czarna lista tworzona przez społeczność ludzi uczestniczących w niekomercyjnym projekcie *PhishTank*, zapoczątkowanym przez twórców usługi OpenDNS.

Serwer weryfikujący pobiera czarną listę *PhishTank* co pewien okres czasu, w konsekwencji czego użytkownik Opery może nie otrzymać ostrzeżenia o najświeższych udostępnianych tam zgłoszeniach – co też można zaobserwować odwiedzając najnowsze linki udostępniane na stronach *PhishTank*. W przypadku sprawdzania stron z wykorzystaniem mechanizmu *TrustWatch* problem ten nie występuje, gdyż serwer weryfikujący kieruje zapytanie w czasie rzeczywistym, bezpośrednio do dostawcy.

W Sieci

- *http://www.opera.com,*
- *http://www.phishtank.com,*
- *http://www.trustwatch.com.*

Klasyfikacja

Serwer weryfikujący dostarcza odpowiedź na żądanie sprawdzenia strony w postaci dokumentu XML. Wśród dostarczanych parametrów najważniejszym jest *trustlevel*, który definiuje poziom zaufania do strony w następujący sposób:

- *NV – Not Verified* – strona nie została umieszczona ani na białej ani na czarnej liście,
- *V – Verified* – witryna znajduje się na białej i nie znajduje się na czarnej liście,
- *W – Warning* – witryna znajduję się na czarnej liście.

W parametrze *blacklist*, który pojawia się w przypadku wystąpienia stanu *W – Warning*, przekazywany jest hash z URL'a znajdującego się na czarnej liście.

Parametry *clientexpiretime* i *serverexpiretime*, jak domniemywam, określają czas życia informacji w procesie *cachingu*.

Sposób odpowiedzi z serwera na żądanie przeglądarki można zaobserwować wykorzystując załączo-

Stan	Protokół	Ikona
V	HTTP	
V	HTTPS	
NV	HTTP	
W	HTTP	Oszustwo
HTTPS z nieprawidłowym certyfikatem:		

Rysunek 1. *Tabela stanów mechanizmu ochrony przed oszustwami*

Listing 3. *Zapytanie do serwera weryfikującego (tryb manualny)*

```
GET /info/?host=www.hakin9.org&ph=CrmIXGGAlC7mS455Q2szhQ==
&hdn=hz3g35Z4i/6JXNonVjl4Mg==
&site=http%3A%2F%2Fwww.hakin9.org%2F HTTP/1.1
User-Agent: Opera/9.21 (Windows NT 5.1; U; pl)
Host: sitecheck.opera.com
Accept: text/html, application/xml;q=0.9,
application/xhtml+xml, image/png, image/jpeg, image/gif,
image/x-xbitmap, */*;q=0.1
Accept-Language: pl-PL,pl;q=0.9,en;q=0.8
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1
Connection: Keep-Alive, TE
TE: deflate, gzip, chunked, identity, trailers
```

ny skrypt (Listing 2), będący de facto prostym *emulatorem* serwera weryfikującego.

Raportowanie

Na podstawie otrzymanych informacji, mechanizm antyfraudowy przybiera odpowiedni stan, który w przypadku *W – Warning*, sygnalizowany jest przez pojawienie się komunikatu mającego poinformować użytkownika o niebezpieczeństwach związanych z wizytą strony – *opera: fraud-warning*, a także poprzez wyświetlenie odpowiedniego symbolu w prawym rogu paska adresowego. Tabelę stanów z symbolami załączono na Rysunku 1.

Tryb manualny

Jak już wspomniałem we wcześniejszej części artykułu, użytkownik ma możliwość skorzystania z weryfikacji manualnej. Po dostaniu się do zakładki *Ochrona przed oszustwami*, przeglądarka wysyła analogiczne zapytanie jak w przypadku trybu automatycznego, z tą różnicą, iż wcześniej kieruje metodą `GET` drugie zapytanie HTTP (Listing 3.) z dodatkowym parametrem *site*, zawierającym jako swoją wartość pełny URL w formie jawnej, który to użytkownik ma wyświetlony w pasku adresowym w chwili wysłania żądania o weryfikację strony.

W odpowiedzi serwer weryfikujący zwraca dokument w formacie HTML, który następnie jest wyświetlany w oknie zakładki. Zawiera on tekstową informację o poziomie wiarygodności strony, a także – w przypadku stanów *V* i *NV* – przycisk wywołujący metodę `POST` z dwoma parametrami: *url – hash* z URL'a oraz *opera* – numer wersji przeglądarki.

Po wywołaniu powyższej metody serwer weryfikujący odsyła użytkownika na witrynę *PhishTank*, aby ten mógł zgłosić stronę jako podejrzaną.

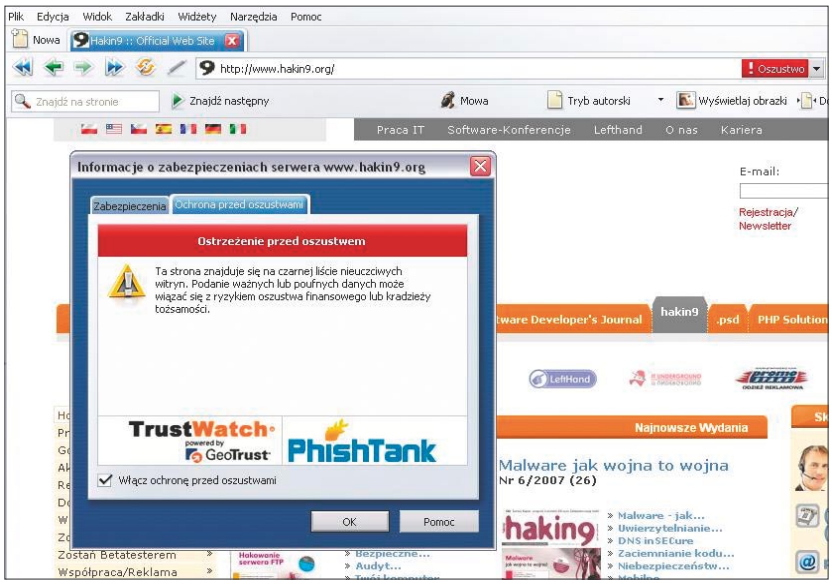
Ryzyka

Uważny czytelnik na pewno zauważył, że największym problemem bezpieczeństwa mechanizmu ochrony przed oszustwami Opery jest zastosowanie protokołu HTTP do weryfikacji adresów zaczynających się od *http://*. Powyższy stan rzeczy daje pewne możliwości nadużyć. Haker wiedząc, że komputer ofiary prędzej czy później będzie pytał swój serwer DNS o adres IP serwera *sitecheck.opera.com*, może w stosunku do niego zaimplementować atak *DNS Spoofing*, polegający na wysyłaniu spreparowanych odpowiedzi na żądania DNS, które rzekomo miałyby pochodzić z rzeczywistego serwera DNS ofiary, zawierających adres IP komputera hakera w miejsce oryginalnego IP serwera *sitecheck.opera.com* (tematykę DNS Spoofing omówiono w numer 2/2003 czasopisma hakin9). W momencie, gdy mechanizm antyfraudowy zacznie łączyć się z komputerem hakera aby prowadzić proces weryfikacji witryn, haker może wykorzystać powyższy stan rzeczy między innymi do uwiarygodnienia witryny stworzonej w celu dokonania oszustwa, wyświetlenia ostrzeżenia o zagrożeniu oszustwem na zaufanej witrynie, a także (co uważam za najmniej abstrakcyjny przykład, a zarazem najbardziej rzeczywiste zagrożenie) w przypadku włączonego u ofiary *trybu automatycznego*, może śledzić aktywność ofiary w Internecie, poprzez monitorowanie stron, które odwiedza.

Efekt jednego z przykładowych ataków załączono na Rysunku 2.

Podsumowanie

Mechanizm ochrony przed oszustwami Opery daje użytkownikowi możliwość uzyskania opinii o wiarygodno-



Rysunek 2. *Wykorzystując technikę spoofingu, wprowadzono w błąd użytkownika sprawdzającego wiarygodność serwisu www.hakin9.org*

ści danej witryny z dwóch niezależnych źródeł informacji, co jest niewątpliwie jego dużym plusem. W zamian użytkownik musi zaakceptować ryzyko związane z przesyłaniem żądań weryfikacyjnych niebezpiecznym kanałem – w tym miejscu należy podkreślić, że większość adresów w Internecie rozpoczyna się od *http://*.

Na zakończenie wspomnieć należy również o fakcie, że obecnie coraz częściej cyberprzestępcy odchodzą od tradycyjnych form phishingu, związanych z tworzeniem oszukańczych stron internetowych z podobną do oryginalnych nazwą, a raczej uciekają się do dużo bardziej wyrafinowanych technik oszustwa. Wykorzystywać można w tym celu złośliwe oprogramowanie, integrujące się z przeglądarką internetową i potrafiące podmieniać użytkownikowi zawartość przeglądanych stron w locie, a także mogące zafałszować informację o nieprawidłowym certyfikacie SSL. Idealnym przykładem jest tu koń trojański *Torpig/Sinowall/Anserin* (nazwa odmienna w zależności od stosowanej nomenklatury producentów oprogramowania antywirusowego), który to jest szeroko stosowany w rozproszonych systemach phishingowych.

W powyższym przypadku, jeśli użytkownik nie ma 100% gwarancji, że jego stacja robocza nie została skompromitowana, to nie powinien ufać jakimkolwiek mechanizmom sprawdzania wiarygodności witryn. ●

O autorze

Marcin Kopeć pracuje jako oficer bezpieczeństwa w jednym z czołowych polskich banków, należącym do największej finansowej grupy kapitałowej w Europie Środkowej. Na co dzień zajmuje się problematyką bezpieczeństwa transakcji elektronicznych, w tym przeciwdziałaniem oszustwom w sektorze bankowości internetowej. Ponadto specjalizuje się w dziedzinie systemów detekcji/przeciwdziałania włamaniom oraz interesuje się kwestiami związanymi z bezpieczeństwem aplikacji webowych.