Bezpieczeństwo sklepów internetowych
Sesje i ciasteczka

Poznańskie Centrum Superkomputerowo-Sieciowe
Zespół Bezpieczeństwa

Ciasteczka (cookies) to niewielkie porcje danych, wysyłane przez serwer WWW i zapisywane po stronie użytkownika. Ciasteczka są stosowane najczęściej w przypadku liczników, sond, sklepów internetowych czy stron wymagających logowania. Raport ten zawiera informacje i opisy mechanizmów obsługujących *cookies* wykorzystywane do identyfikacji użytkowników zalogowanych do sklepu internetowego. W dokumencie znajdują się przykłady błędnej konfiguracji platformy serwerowej, przykłady błędnych mechanizmów obsługi ciasteczek.

Wersja 1.0 03/01/2008

Zespół Bezpieczeństwa PCSS zajmuje się analizami, badaniami oraz konsultacjami w zakresie bezpieczeństwa teleinformatycznego. Bierze udział w wielu europejskich projektach badawczych oraz świadczy usługi w zakresie testów penetracyjnych dla klientów komercyjnych. Więcej informacji na stronach zespołu (http://security.psnc.pl).

SPIS TREŚCI

Wprowadzenie

Zakupy internetowe stały się w ostatnich miesiącach masowe. Najlepszym dowodem tego faktu okazała się przedświąteczna blokada Poczty Polskiej, spowodowana olbrzymią ilością przesyłek zakupionych przez klientów w sklepach internetowych.

Klienci, zachęceni niższymi cenami (niekiedy różnice sięgają kilkunastu procent wartości), powszechnie wybierają tę formę zakupu towaru. Kolejną z dogodności związanych z zakupami przez Internet jest możliwość szybkiego porównania oferty nawet kilkuset różnych sklepów.

Oprócz wspomnianych zalet, dokonywanie zakupów w Internecie ma też swoje wady. Robiąc zakupy w sklepach internetowych jesteśmy zmuszeni do podania swoich danych osobowych w celu otrzymania zamówionego towaru. Kupując sprzęt znacznej wartości nie jesteśmy „panem w czarnym swetrze ubranym w jeansy" kupującym wymarzony sprzęt RTV. W Internecie stajemy się Janem Nowakiem, mieszkającym przy ulicy Słonecznej 12 w Toruniu.

Coraz częściej kupowane są w Internecie towary o znacznej wartości. Eksperci przewidują, że do roku 2006-2010 obroty handlu elektronicznego wzrosną co najmniej do kwoty niemal 130 mld dolarów[1]. Według wstępnych szacunków, przychody polskiego e-handlu w roku 2007 ocenić należy na 8 mld PLN[2].Robiąc internetowe zakupy zdradzamy swój stan majątkowy, upodobania, profil psychologiczny. Nasze dane, zachowane w przepastnych bazach sklepów internetowych, są niezwykle cenne dla wszelkiej maści agencji reklamowych, konkurencyjnych sklepów czy też... urzędów skarbowych. Mogą one również posłużyć kryminalistom w celu wyszukania potencjalnych zamożniejszych ofiar, dokonywania wymuszeń itp.

Mając na uwadze powyższe zagrożenia, Zespół Bezpieczeństwa PCSS przeprowadził test 50 wybranych sklepów internetowych w celu sprawdzenia, na ile bezpieczne jest dokonywanie zakupów w tej formie. Sklepy wybrano w losowy sposób spośród istniejących polskich rozwiązań tego typu (z posiadanych przez nas danych wynika że ich liczbę można szacować na w przybliżeniu 3.300[3].) Nie ograniczaliśmy się do konkretnego typu sklepu – w teście wzięły udział sklepy ze sprzętem komputerowym, odzieżą, kosmetykami czy też sprzętem RTV-AGD.. Nie była brana również pod uwagę platforma serwerowa, na której został zainstalowany dany sklep internetowy. W wyborze sklepów nie uwzględnialiśmy skryptów obsługujących sklepy internetowe – gotowe środowiska e-commerce, skrypty pisane na konkretne zamówienie, wynajęte środowiska.

[1] http://www.komputerwfirmie.pl/itbiznes/1,54785,3250071.html
[2] http://inwestycje.pl/it_ebiznes/rosnie_polski_e_handel;19262;0.html
[3] ibidem

Przeprowadzone badania nie są kompletnymi testami bezpieczeństwa wybranych sklepów. Posłużyły one wyłącznie do sprawdzenia poziomu bezpieczeństwa implementacji mechanizmów tworzenia sesji użytkownika przez skrypty serwera.

HTTP Cookies i sesje

Najnowsze wiadomości, wyniki sportowe czy notowania giełdowe *online*, portale społecznościowe, poszukiwanie pracy – wreszcie będące przedmiotem niniejszego raportu zakupy w Internecie: wszystkie te usługi (i wiele innych, tu nie wymienionych) dostępne są dla każdego użytkownika dzięki protokołowi HTTP. Hyper Text Transfer Protocol oraz jego szyfrowany odpowiednik, HTTPS (Secure HTTP) są protokołami bezstanowymi, to znaczy „nie pamiętają" one historii pakietów wymienianych wcześniej w ramach konkretnego połączenia. Oba protokoły nie utrzymują również same w sobie informacji o bieżącym statusie konkretnej transmisji.

Taka sytuacja znacząco upraszcza sam protokół, jak i jego implementację na serwerach czy w przeglądarkach klienta, niesie jednak za sobą poważne konsekwencje w odniesieniu do funkcjonalności aplikacji webowych. Wiele z nich wymaga zapamiętania bieżącego stanu połączenia. Wyobraźmy sobie, jaką udręką dla użytkownika byłaby konieczność logowania się na każdej podstronie odwiedzanego serwisu. A ile zamieszania wprowadziłby brak możliwości sprawdzenia, czy konkretny użytkownik zagłosował już w sondzie internetowej, jakie produkty ma w swoim koszyku sklepowym? Jak miałoby wyglądać generowanie statystyk odwiedzin na stronach?

Te i wiele innych problemów rozwiązuje się, wprowadzając dodatkowe mechanizmy przechowywania danych o połączeniach – tzw. **sesjach**. Po stronie serwera informacje te przechowywane są najczęściej w plikach, po stronie klienta – w niewielkich strukturach o formacie tekstowym, zwanych ciasteczkami (cookies). W razie potrzeby serwer webowy – bądź przeglądarka klienta – sięga po cookie skojarzone z potrzebną sesją, odczytuje niezbędne informacje i postępuje w odpowiedni do ich zawartości sposób.

Alternatywami dla ciasteczek jest przekazywanie danych sesji przy pomocy ukrytych pól formularzy lub w ciągu adresu URL. Pierwszy z tych sposobów nadmiernie komplikuje budowę serwisów internetowych, natomiast drugi jest nie do zaakceptowania ze względów bezpieczeństwa (przede wszystkim mając na uwadze możliwość łatwego przechwycenia danych w wyniku podsłuchiwania ruchu sieciowego przez napastnika bądź nieświadomego ujawnienia danych sesyjnych przez użytkownika). Z tego powodu olbrzymia większość serwisów stosuje mechanizm ciasteczek, któremu poświęcony jest niniejszy raport.

Ciasteczka

Struktura ciasteczka

Cookie jest niewielką porcją danych w formacie tekstowym. Składa się ona z następujących informacji:

- nazwa oraz skojarzona z nią wartość. Wartość może składać się z dowolnych znaków z wyjątkiem niektórych znaków specjalnych (które jednak można zakodować heksadecymalnie). Para nazwa-wartość jest jedyną ściśle wymaganą wielkością do utworzenia ciasteczka.
- expires (data wygaśnięcia) – wartość tekstowa w określonym przez odnośne dokumenty RFC formacie, określająca moment utraty ważności cookie. Kiedy wskazany czas upływa, ciasteczko zostaje usunięte z dysku przez przeglądarkę. Jeżeli wartość expires nie jest podana, ważność ciasteczka wygasa z chwilą zamknięcia sesji.
- domain (domena) – atrybut określa, do których serwerów przeglądarka będzie mogła wysłać cookie (widoczność ciasteczka). Jeżeli wartość ta nie jest bezpośrednio podana podczas definicji ciasteczka, przyjmuje wartość domeny, z której pochodzi polecenie utworzenia cookie.
- path (ścieżka) – umożliwia określenie ścieżki na serwerze, spod której (wliczając ewentualne podkatalogi) ciasteczko będzie widoczne. Jeśli w definicji ciasteczka pominie się wartość path, domyślnie przypisana jej zostanie ścieżka do strony, która wygenerowała żądanie definicji cookie.
- secure (bezpieczny) – atrybut ten nie ma określonej wartości, może być po prostu podany lub nie. Jeśli jest wyspecyfikowany, cookie będzie wysłane jedynie wtedy, gdy sesja będzie realizowana przy pomocy bezpiecznej wersji protokołu – HTTPS.

Załóżmy, że pewien serwis internetowy, np. http://www.example.com, implementuje licznik odwiedzin, który powinien być zwiększany jedynie przy pierwszych odwiedzinach konkretnego użytkownika na stronie (nie częściej niż np. raz na godzinę). Cel ten można osiągnąć poprzez sprawdzenie, czy przeglądarka użytkownika zachowała ciasteczko skojarzone z licznikiem (ewentualnie na podstawie danych sesyjnych przechowywanych po stronie serwera). Jeżeli tak nie jest, stan licznika będzie zwiększony o 1, a do przeglądarki użytkownika wysyłane jest nowe cookie, wygasające za godzinę. Serwer wysyła informacje o ciasteczku przy pomocy nagłówka Set-Cookie (w jednym pakiecie HTTP może być ich wiele). Z kolei przeglądarka odsyła dane sesji przechowywane w cookie, wykorzystując nagłówek Cookie (w jednym nagłówku może znajdować się wiele par nazwa-wartość).

Zarządzanie ciasteczkami po stronie użytkownika odbywa się w sposób zautomatyzowany przy pomocy wbudowanej funkcjonalności przeglądarek

internetowych. Najczęściej użytkownik może wyłączyć ciasteczka w ogóle (w wyniku czego przestanie działać przeważająca liczba usług, z których zamierza skorzystać), akceptować ciasteczka z określonych witryn, podejmować decyzję za każdym razem (co byłoby dziś niezwykle nużące i pracochłonne) lub wreszcie akceptować wszystkie ciasteczka (w tym potencjalnie niebezpieczne).

Przykłady

Przykład 1. Struktura nagłówka HTTP Set-Cookie wysyłanego przez serwer

```
Set-Cookie: licznik=yes; expires=Thursday, 10-Jan-2008 11:00:00
GMT; domain=www.example.com; path=/licznik_dir
```

Przykład 2. Struktura nagłówka Cookie wysyłanego przez przeglądarkę

```
Cookie: licznik=yes; session_id=87as8r03ju4d0q0-fq=ffvgfqf
```

Przykład 3. Definicja ciasteczka w kodzie PHP

```
<?
setcookie(“licznik”, “yes”, time()+3600, “/licznik_dir”,
“www.example.com”);
?>
```

Przykład 4. Definicja ciasteczka w kodzie ASP

```
<%
HttpCookie cookie = new HttpCookie(“licznik”);
cookie.Name       = “licznik”;
cookie.Value      = “yes”;
cookie.Expires    = #01/10/2008 11:00:00#;
cookie.Domain     = “www.example.com”;
cookie.Path       = “/licznik_dir”;
Response.Cookies.Add(cookie);
%>
```

Badania

Pierwszymi testami wykonywanymi w badanych sklepach internetowych było sprawdzenie, jak zachowują się skrypty serwera w przypadku otrzymania od użytkownika danych cookies ze zmieniona zawartością. Odpowiedni dobór danych wysłanych przez użytkownika do serwera pozwala na identyfikację sposobu przechowywania danych sesji na serwerze. Dodatkowo, w przypadku błędnej konfiguracji języka PHP, istnieje możliwość uzyskania informacji o pełnej ścieżce uruchamianych na serwerze skryptów.

Test 1. Znak „!”

Pierwszy test miał na celu weryfikację zachowania serwera w przypadku otrzymania od użytkownika ciasteczka z danymi zawierającymi znak „!”. Test ten służy do identyfikacji systemów, w których dane należące do określonej sesji przechowywane są w plikach. W przypadku, kiedy mechanizm obsługi sesji wykryje w danych przesyłanych od użytkownika niepoprawny symbol, wyświetlany jest komunikat o błędzie. Komunikat ten zostanie wyświetlony wówczas, gdy włączone jest raportowanie o błędach pojawiających się w działaniu skryptu PHP.

Przykłady komunikatów z błędami:

Warning: session_start() [function.session-start]: The session id contains invalid characters, valid characters are only a-z, A-Z and 0-9 in **/includes/functions/sessions.php** on line **67**

Warning: session_start() [function.session-start]: Cannot send session cache limiter - headers already sent (output started at /includes/functions/sessions.php:67) in **/includes/functions/sessions.php** on line **67**

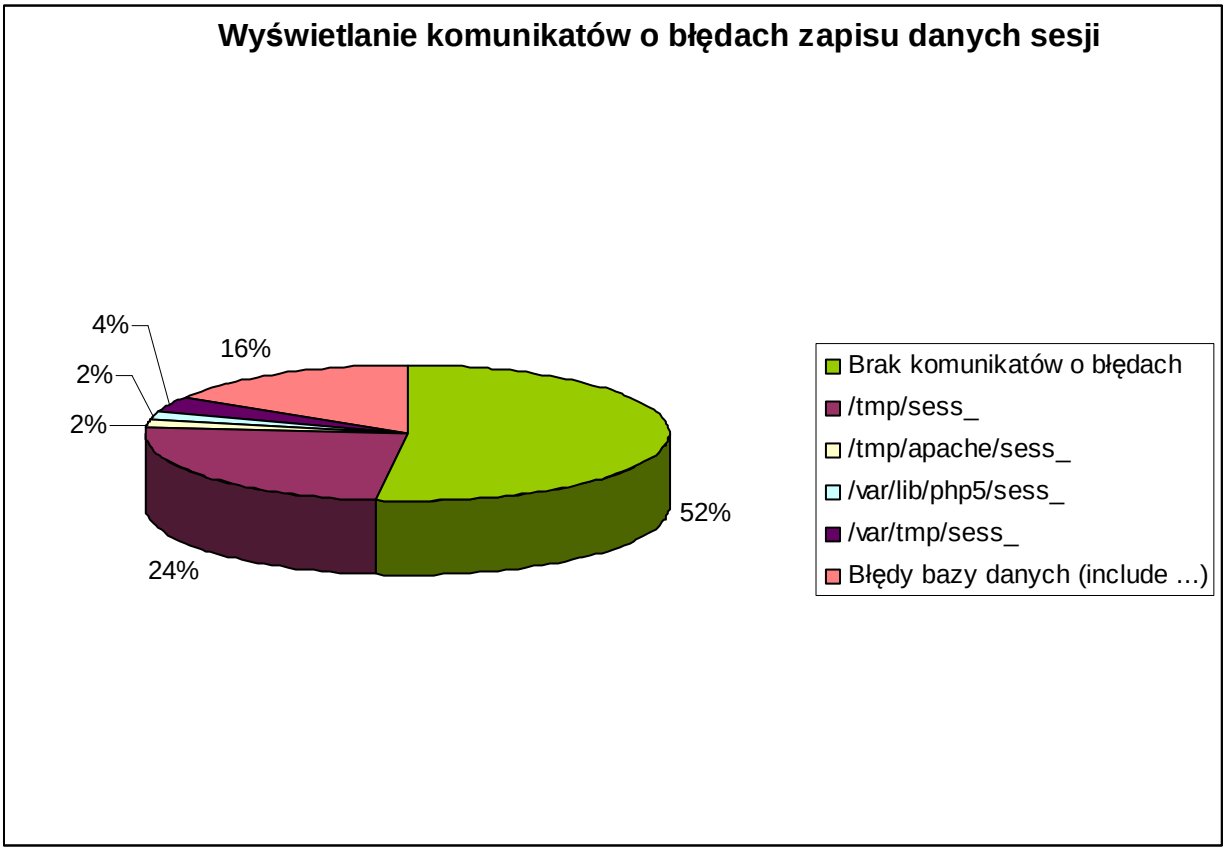
Komentarz do testu:

Mechanizm wyświetlania błędów w przeglądarce użytkownika ma służyć do identyfikacji błędów w aplikacjach webowych będących w fazie testowej. Prawidłowo skonfigurowane środowisko produkcyjne nie powinno wyświetlać użytkownikowi komunikatów o błędach w funkcjonowaniu skryptów.

Test 2. Ciasteczka z danymi o długości powyżej 256 znaków

Test ten sprawdzał zachowanie serwera w przypadku otrzymania od użytkownika ciasteczka z danymi o długości powyżej 256 znaków. Test służy do identyfikacji systemów, w których dane należące do konkretnej sesji przechowywane są w plikach. W wypadku, kiedy mechanizm obsługi sesji będzie próbował zapisać informacje w pliku z danymi sesyjnymi, wyświetlany będzie komunikat o błędzie polegającym na braku możliwości stworzenia pliku o podanej nazwie. Komunikat ten zostanie wyświetlony wówczas, gdy włączone jest raportowanie o błędach pojawiających się w działaniu skryptu PHP.

Przykład komunikatu o błędzie:
Warning: session_start() [function.session-start]:
open(/tmp/sess_xxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYY
YYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYY
YYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxx
xxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxxYYYYYYYYYYxxxxxxxxxx
YYYYYYYYYYxxxxxxxxxxYYYYYYYYYY, O_RDWR) failed: File name too long (63)
in **/includes/functions/sessions.php** on line **67**

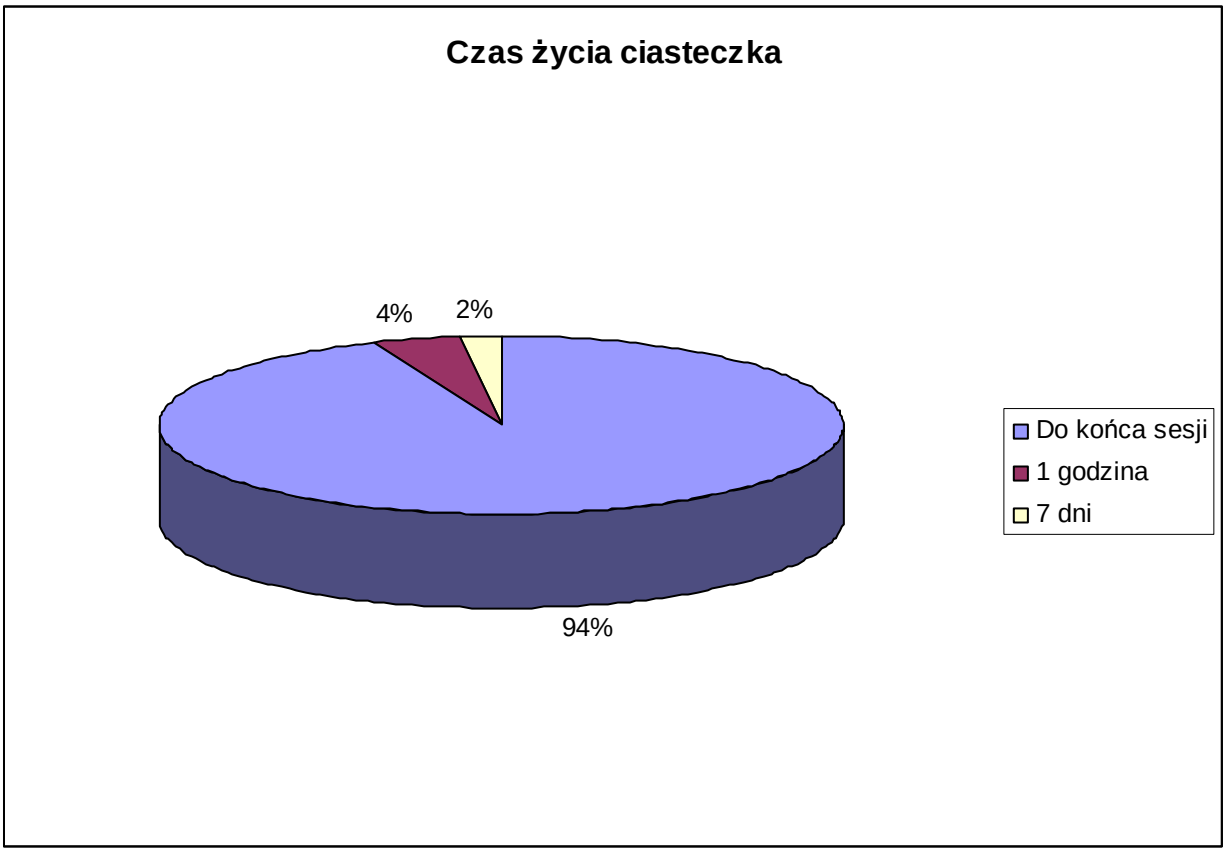


Komentarz do testu:

Wyświetlany komunikat o błędzie wskazuje katalog, w którym mechanizm obsługi sesji próbuje zapisać dane sesyjne. Zapisywanie danych sesyjnych w katalogu /tmp na serwerach firm hostingowych nie jest rozwiązaniem bezpiecznym, gdyż każdy z pozostałych użytkowników mających własne serwisy na tych serwerach ma możliwość uzyskania danych sesyjnych aktualnie zalogowanych użytkowników.

Test 3. Czas życia ciasteczka

Najważniejszą właściwością ciasteczka jest jego czas życia. Według standardu OWASP długość życia ciasteczek służących do identyfikacji zalogowanego użytkownika powinna wynosić najwyżej 5 minut.

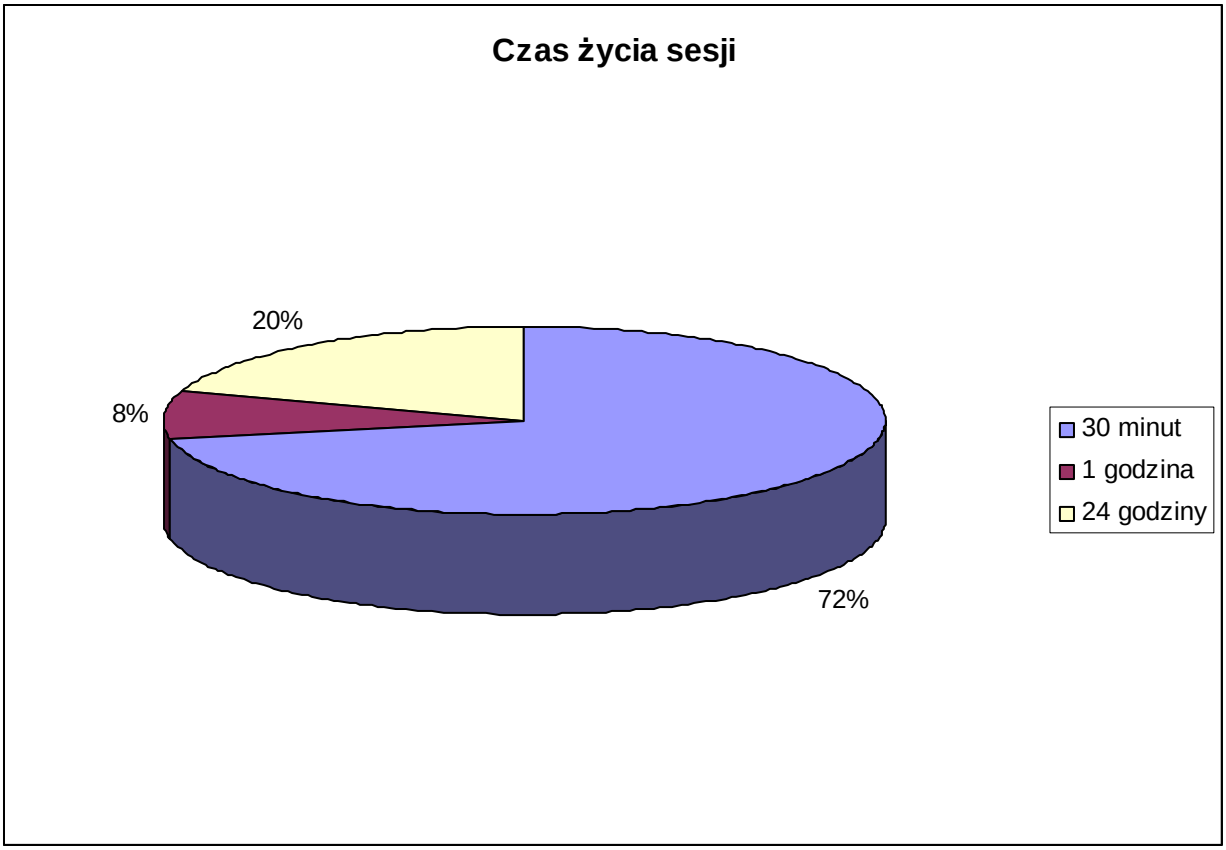


Komentarz do testu:

Większość administratorów systemów e-commerce ustawia długość życia ciasteczka na czas trwania sesji – ciasteczka te są usuwane z dysku podczas kończenia pracy przeglądarki. Taki sposób postępowania powoduje zagrożenie związane z faktem, że jeśli użytkownik nie zakończy pracy z portalem poprzez naciśnięcie przycisku Wyloguj, system nie potrafi rozpoznać sytuacji, w której przeglądarka internetowa została zamknięta. W przypadku sesji opartych o pliki czas życia sesji wynosi 24 godziny!

Test 4. Czas życia sesji

Podczas tego testu sprawdzaliśmy ile wynosi czas życia sesji na serwerze których uruchamia skrypty sklepu.



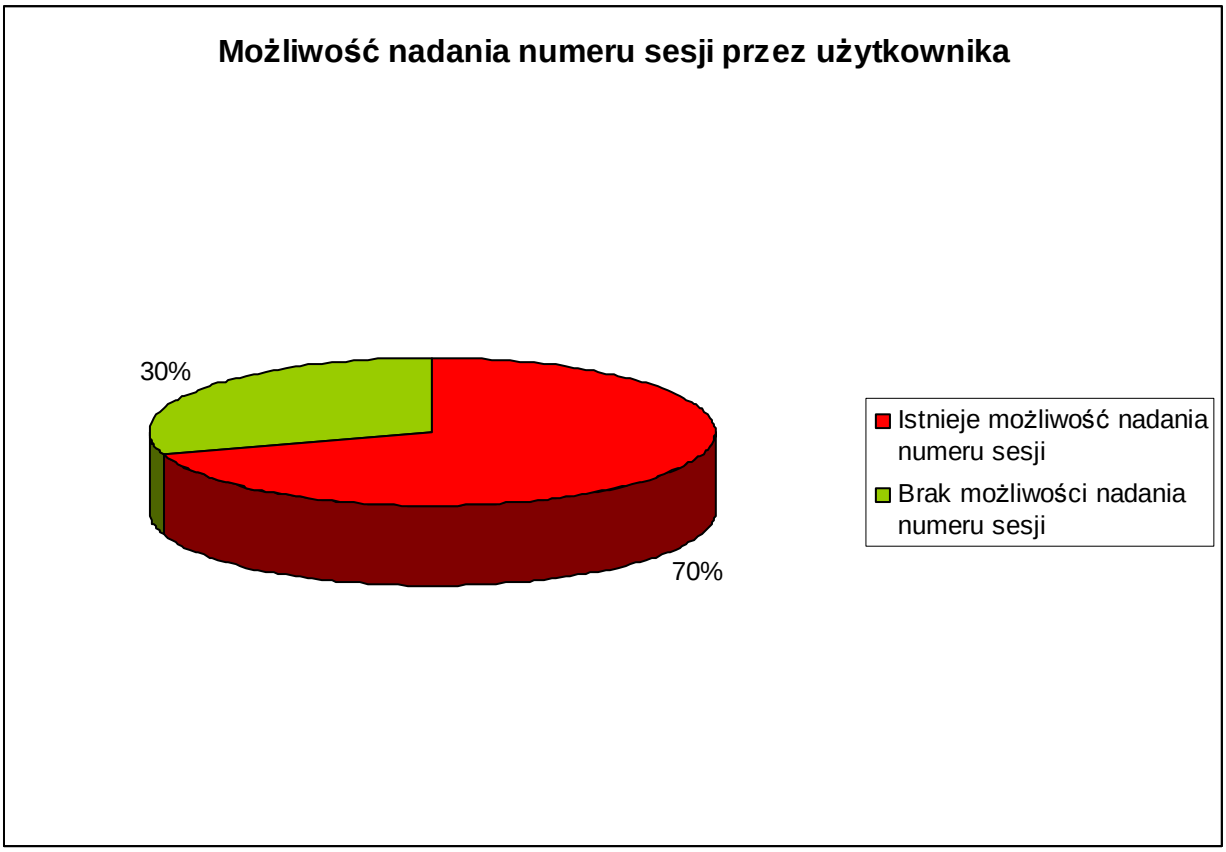
Komentarz do testu:

Należy mieć na uwadze, że czas życia ciasteczka nie równa się czasowi życia sesji na serwerze. W przypadku kiedy użytkownik kończy zakupy w sklepie zamykając przeglądarkę internetową, przeglądarka nie wysyła informacji do serwera, aby ten zakończył istniejącą sesje. Test ten pokazuje jak długo trwa nasza sesja po ostatnim naszym żądaniu do serwera.

Test 5. Możliwość nadania numeru sesji przez użytkownika

Podczas tego testu użytkownik wysyłał do serwera ciasteczko z własnym nowym numerem sesji. Następnie dodawaliśmy dowolny produkt do koszyka. Po kolejnym uruchomieniu przeglądarki i ściągnięciu strony z ustalonym

numerem sesji sprawdzaliśmy, czy znajduje się w nim zamówiony przed chwilą towar.



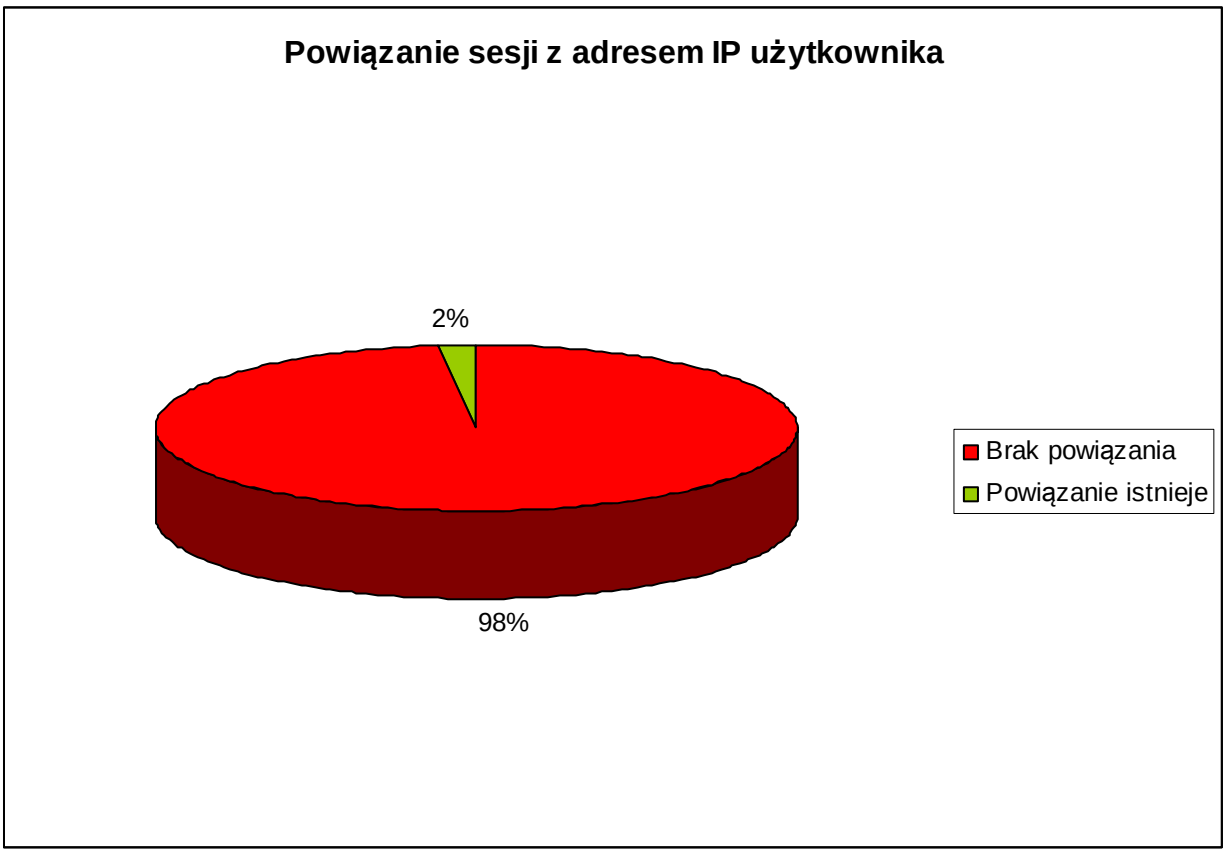
Komentarz do testu:

Sytuacja, w której użytkownik ma możliwość stworzenia sesji o wybranym numerze jest bardzo niebezpieczna. Napastnik wysyłający odpowiednio spreparowany link do ofiary będzie znał numer sesji, jaka może zostać nawiązana przez zaatakowaną osobę. Przykładowo skompromitowany link
http://website.kom/<script>document.cookie="sessionid=abcd";</script>[4]

Test 6. Powiązanie klucza sesji z adresem IP komputera użytkownika

Podczas tego testu sprawdzaliśmy, czy istnieje możliwość użycia ciasteczka zalogowanego użytkownika na komputerze z innym adresem IP. Test ten jest przykładem ataku, podczas którego napastnik zdobywa ciasteczko ofiary.

[4] http://www.owasp.org/index.php/Session_fixation

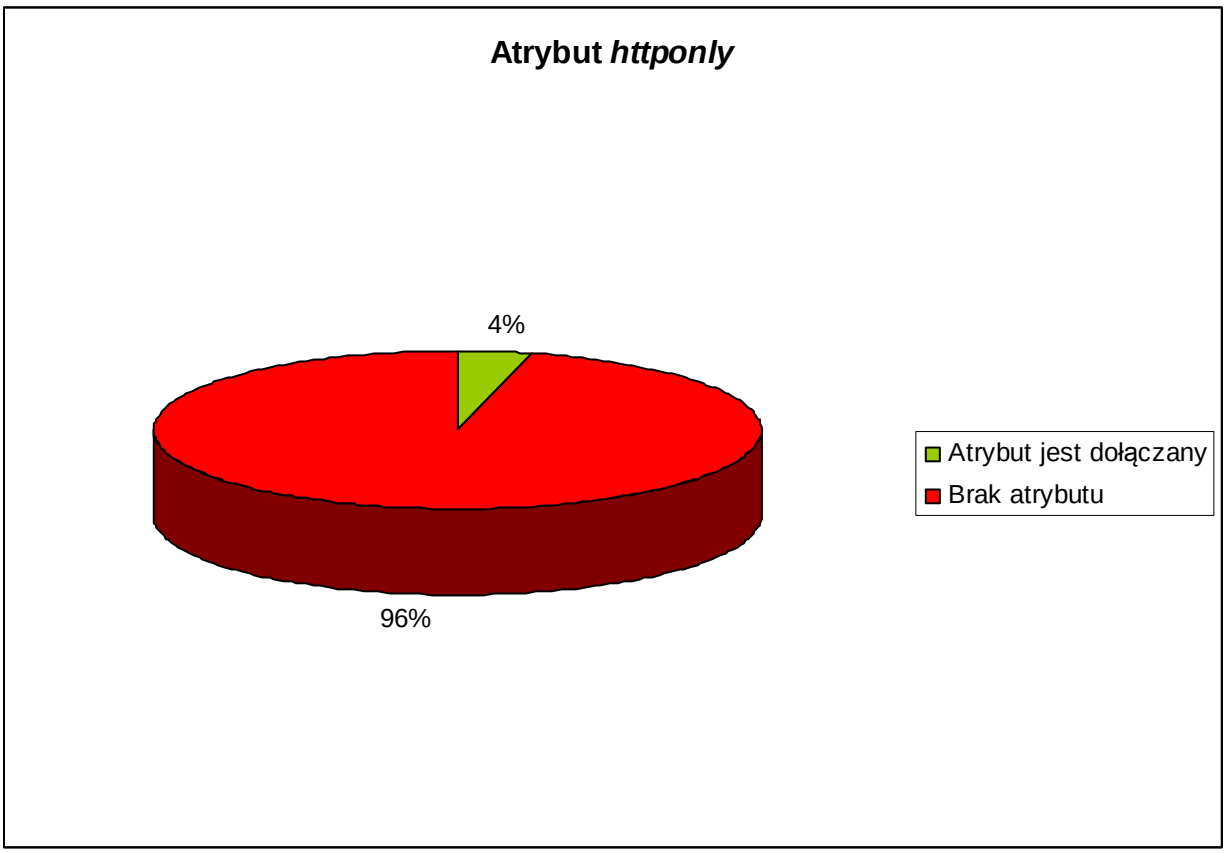


Komentarz do testu:

Brak powiązania numerów sesji z adresami IP sprawia, że dokonywanie zakupów poprzez najpopularniejsze nieszyfrowane sieci bezprzewodowe jest bardzo niebezpieczne. Napastnik w łatwy sposób może uzyskać dostęp do numeru sesji ofiary.

Test 7. Ciasteczka *httponly*.

Ostatni test polegał na weryfikacji występowania atrybutu *httponly* w przesyłanych ciasteczkach. Atrybut ten sprawia, że ciasteczko będzie wykorzystane tylko podczas wysyłania zadań do serwera przez przeglądarkę.



Komentarz do testu:

Ciasteczko typu *httponly* nie będzie dostępne z poziomu języka JavaScript (document.cookie). Jest to skuteczne zabezpieczenie przed kradzieżą numeru sesji przez napastnika wykorzystującego podatności typu XSS (Cross Site Scripting), które występują w przeważającej większości dostępnych w Internecie serwisów[5], w tym także sklepów internetowych.

[5] http://www.webappsec.org/projects/statistics