PORTFOLIO:
Analiza zachowań użytkowników serwisów internetowych

Autorzy: Marek Zachara

Centrum Inteligentnych Systemów Informatycznych Akademia Górniczo-Hutnicza im. Stanisława Staszica al. Mickiewicza 30, 30-059 Kraków
budynek C-2 pokój 426 tel.: 12 617 44 53 www.isi.agh.edu.pl **isi@agh.edu.pl**

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

1. Streszczenie

W ramach projektu opracowane i przebadane zostały metody reprezentacji zachowań użytkowników serwisów internetowych za pomocą grafów – a także ich wykorzystanie do identyfikacji zachowań nietypowych, potencjalnie wskazujących na próby ataku.
Jak wykazały wstępne testy, zastosowanie skierowanych grafów ważonych oraz 'rozproszonej wiedzy' – czyli komunikacji pomiędzy kilkoma serwisami WWW pozwala na wykrywanie części potencjalnych ataków – i to bez konieczności specyficznej konfiguracji czy narzutu związanego z bieżącym utrzymaniem systemu.
Obecnie najistotniejszą kwestią wydaje się weryfikacja możliwości opracowanych metod i przygotowanego prototypu w rzeczywistym środowisku produkcyjnym o większej skali.

2. Opis problemu

Coraz większa część aktywności ludzi przenosi się do internetu. Obejmuje to zarówno zakupy, życie towarzyskie (serwisy społecznościowe) czy sprawy urzędowe (np. polska platforma ePUAP).
Liczba aktywnych serwisów internetowych na świecie to obecnie ok. 180 milionów (wg Netcraft Web Server Survey).
Niestety, wraz ze wzrostem ilości treści i możliwości usług internetowych rośnie też ich potencjalna atrakcyjność dla (potencjalnych) przestępców, natomiast świadomość zagrożeń i wiedza na temat bezpieczeństwa wśród twórców i administratorów tych serwisów często nie nadąża za potrzebami. Dla potwierdzenia tego faktu poniżej przedstawione zostało krótkie streszczenie kilku adekwatnych raportów

2.1 Skala zagrożeń

Trudno jest dokładnie zdiagnozować ogólny stopień bezpieczeństwa sieci. Pewny pogląd dają jednak raporty przygotowywane przez firmy zajmujące się profesjonalnie tym tematem. I tak, iVIZ w swoim raporcie z 2013 r. podaje że 99% z ok. 300 testowanych serwisów ich klientów posiadało przynajmniej jedną podatność na atak, przy średnio 35 takich podatnościach w serwisie. 82% z tych podatności to były błędy krytyczne.
WhiteHat podaje nieco mniejszą wartość - ok. 86% testowanych serwisów posiadało przynajmniej jedną poważną podatność przy średnio 56 odkrytych podatnościach w każdym serwisie. Symantec, w raporcie 'Internet Security Threat Report' z 2013 podaje jeszcze mniejszą (choć w dalszym ciągu znaczną) liczbę - 53% serwisów zawierających podatności na ataki, przy czym w tym przypadku można domniemywać że było to mniej szczegółowe badanie (automatyczne, średnio ok. 1400 serwisów dziennie).
Biorąc to pod uwagę można bezpiecznie ocenić że zdecydowana większość serwisów internetowych (poza nielicznymi wyjątkami) zawiera istotne błędy pozwalające atakującym na wykorzystanie ich podatności.

2.2 Podsumowanie ryzyka

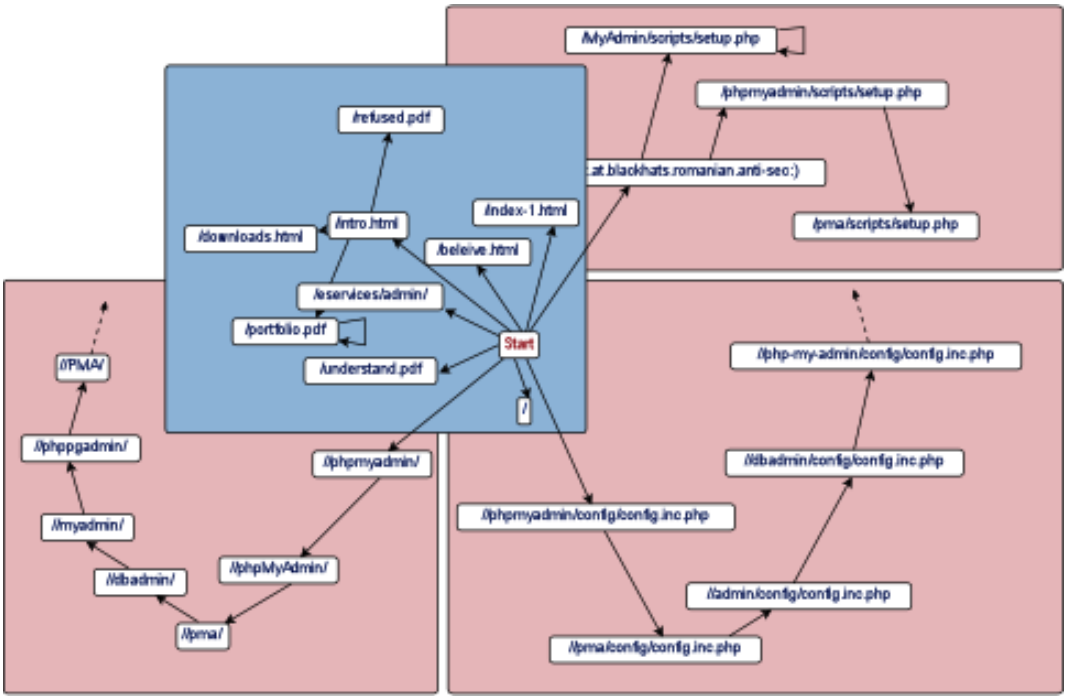
Jak widać, bezpieczeństwo serwisów internetowych pozostawia wiele do życzenia. Biorąc pod uwagę potencjalne skutki ataków (np. przejęcie konta użytkownika, wykonanie nieautoryzowanych transakcji, kradzież tożsamości itp.), widać że jest to istotny problem społeczno-techniczny. Opracowanie i wprowadzenie metod redukujących to ryzyko jest więc istotnym zagadnieniem rozwojowym. Jednym ze sposobów na poprawę bezpieczeństwa serwisów jest opracowana i opisana niżej metoda.

3. Opracowana metoda

Proponowana metoda ochrony serwisów internetowych polega na monitorowaniu zapytań (ang. *request*) przychodzących od danego użytkownika, a następnie tworzeniu modelu zachowań tych użytkowników.

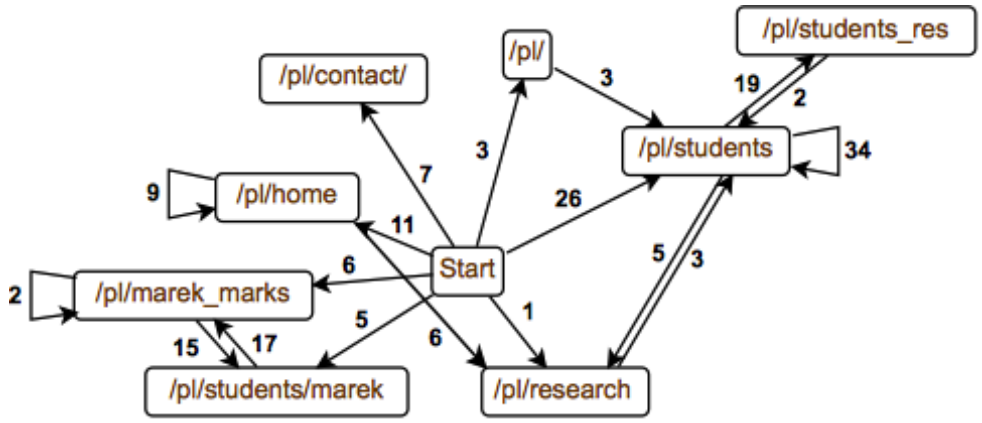
3.1. Identyfikacja nietypowych zachowań lokalnych

Poniższy rysunek nr 1 ilustruje przykładowe (autentyczne) sekwencje zapytań użytkowników przychodzące do monitorowanego serwera. Na niebieskim tle przedstawione zostały sekwencje zapytań od 'uczciwych' użytkowników, pozostałe ilustrują zaś próby ataku.



Rysunek 1: Przykładowe sekwencje zapytań.

Rysunek ten ilustruje dodatkowo jeszcze jeden aspekt – a mianowicie przyjętą w opracowanej metodzie reprezentację zachowania użytkownika jako grafu skierowanego, w którym węzły odpowiadają kolejnym zapytaniom, natomiast krawędzie oznaczają ich sekwencje.
Jeżeli dodatkowo każde przejście pomiędzy stronami będzie zwiększać o jeden wagę danej krawędzi to otrzymamy graf ważony o przykładowej strukturze zaprezentowanej na rysunku 2, gdzie waga danej krawędzi odpowiada względnej częstotliwości danego przejścia (w stosunku do sumy wag krawędzi wychodzących z danego węzła).



Rysunek 2: Przykład modelu zachowań reprezentowanego przez graf ważony.

Dokonując okresowej erozji wag grafu (redukcji ich wartości) otrzymuje się dynamiczny model dostosowujący się do aktualnej struktury serwisu i typowych zachowań użytkowników. Dzięki temu możliwa jest identyfikacja zapytań 'podejrzanych', które nie są zgodne z takimi typowymi zachowaniami. Zapytania takie będą bowiem relatywnie rzadkie (poniżej 1%) dla normalnego serwisu obsługującego setki użytkowników.

3.2. Wspólna weryfikacja zagrożeń

Poza lokalnym modelowaniem i identyfikacją nietypowych zapytań, możliwe jest również wykorzystanie rozproszonej wiedzy wielu serwerów w celu poprawy skuteczności i jakości detekcji (przede wszystkim redukcji tzw. '*false positives*'). Opracowana metoda opiera się na fakcie dużej powtarzalności wzorców ataku, co wynika z tego iż znaczna ich część jest prowadzona albo przez tzw. '*script kiddies*' – czyli młodzież ściągającą gotowe narzędzia z internetu, bądź też przez malware. Tezę tą wspiera m.in. wspomniany wcześniej raport Symantec, w którym napisano że tylko w jednym miesiącu (05.2012) malware o nazwie LizaMoon był odpowiedzialny za ponad milion zakończonych sukcesem ataków typu *SQL Injection*.

Bazując na tych założeniach opracowany został sposób wymiany informacji pomiędzy serwerami pozwalający na identyfikację powtarzających się wzorców zapytań przychodzących do różnych serwerów – dzięki czemu możliwe jest istotne zwiększenie pewności poprawnej identyfikacji. Każdy z serwerów prowadzi ocenę na bazie własnego modelu zachowań, uwzględniając dodatkowo raporty opublikowane przez inne serwery – i publikując swoje wyniki (listę zapytań ocenionych jako podejrzane). Ponieważ publikacja przez serwer treści zapytań (a nawet URL-i) nie jest możliwa ze względu na ryzyko ujawnienia poufnych informacji, wykorzystany został mechanizm generowania tzw. skrótów kryptograficznych (ang. *hash*). Porównując takie skróty opublikowane przez inne serwery ze skrótami wygenerowanymi lokalnie, serwer uzyskuje ew. potwierdzenie że identyczne zapytanie zostało wysłane do innego serwera – i przez niego również ocenione jako podejrzane. Pozwala to praktycznie w 100% wyeliminować potencjalne fałszywe alarmy (choć kosztem nieznacznego spadku detekcji prawdziwych ataków).

3.3. Główne założenia (uwarunkowania)

Opracowany system bazuje na kilu założeniach, które należy tutaj wymienić. Dotyczą one charakteru ataków które można wykrywać opracowaną metodą. Są to w szczególności:

Zapytania nietypowe, wychodzące poza standardowe ścieżki (ang. *page flow*) użycia aplikacji.

Ataki powtarzalne – prowadzone przez automatyczne skrypty i/lub złośliwe oprogramowanie (malware)

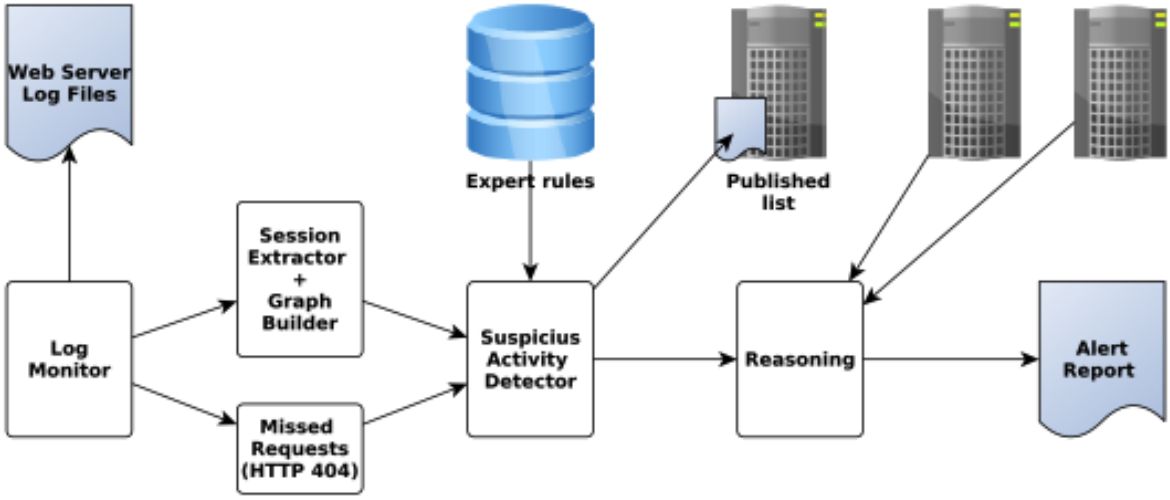
Ataki rozproszone, stosowane wobec grupy serwerów, bez konkretnego celu – mające na celu znalezienie takich które posiadają podatności które mogą być wykorzystane

W stosunku do takich ataków, skuteczność opracowanych metod jest relatywnie duża. Należy jednak podkreślić że opisany system (w przypadku użycia metod kolektywnej oceny zdarzeń) nie będzie sobie radził z specyficznymi atakami prowadzonymi w celu uzyskania dostępu do konkretnego jednego serwera. W takiej sytuacji należy zastosować inne narzędzia.

Warto jednak wspomnieć że praktycznie nie ma możliwości zagwarantowania 100% bezpieczeństwa żadnego systemu informatycznego – i odpowiednio zdeterminowana grupa (wyposażona w odpowiednie narzędzia) jest w stanie uzyskać dostęp praktycznie do każdego jednego systemu (patrz przykład StuxNet).

4. Prototyp i wstępne wyniki

Na potrzeby weryfikacji opracowanych metod skonstruowany został prototyp systemu, wykonany w języku Java. Prototyp ten działa na zasadzie monitorowania logów serwera WWW (Apache), budowania odpowiednich modeli, analizy i raportowania. Schemat poglądowy modułów został przedstawiony na rysunku 3.



Rysunek 3: Schemat poglądowy prototypu systemu

Kolejne wpisy w dzienniku zdarzeń (log) web serwera są parsowane, a następnie tworzony jest graf modelu zachowań (opisany wcześniej). Przejścia pomiędzy stronami, które nie mają wsparcia w przygotowanym grafie (oraz wszystkie zapytania o nieistniejące strony – błąd 404) są przekazywane do modułu identyfikującego, który to moduł po uwzględnieniu grupy zdefiniowanych reguł (np. dotyczących ignorowania zapytań przez przeglądarkę o ikonę strony – '*favicon.ico*'). Przekazuje potencjalnie podejrzane requesty do ostatniego modułu (Reasoning) oraz aktualizuje publikowaną listę *hash*-y.
Ostatni moduł ocenia stopień pewności co do zidentyfikowania podejrzanego zapytania, uwzględniając ew. zgłoszenie identycznych problemów przez inne współpracujące serwery i raportuje, w czasie rzeczywistym, zidentyfikowane zagrożenia.

4.1. Wstępne wyniki

Prototyp systemu został przebadany w testowym środowisku składającym się z trzech publicznych serwisów internetowych. Były to relatywnie małe serwisy, otrzymujące poniżej 10 tys. zapytań miesięcznie. Prototyp systemu poprawnie zidentyfikował ponad 30%

potencjalnie niebezpiecznych zapytań (co zostało obliczone w stosunku do 'ręcznie' przeanalizowanych/oznaczonych zapytań). Należy podkreślić że wynik ten został osiągnięty bez konieczności nauki/konfigurowania a'priori systemu, oraz bez jednego fałszywego alarmu (*false positive*). Istotnym ograniczeniem wpływającym na wyniki było małe środowisko testowe – jedynie trzy serwery i mała liczba zapytań dziennych. Z pewnością zwiększenie liczby współpracujących serwisów podniosłoby istotnie poziom detekcji.

5. Podsumowanie

Opracowane metody i przygotowany prototyp systemu oferuje interesujące i unikalne cechy. W szczególności pozawala na identyfikację potencjalnie groźnych zapytań kierowanych do serwera – w czasie rzeczywistym i to bez konieczności specyficznej konfiguracji. Pozwala to na uzyskanie dodatkowej warstwy ochrony aplikacji internetowych, przy minimalnych kosztach. Ponieważ nie ma możliwości zagwarantowania 100% bezpieczeństwa systemów informatycznych – szczególnie tych podłączonych do internetu - zadaniem administratorów jest więc generalnie podwyższanie poziomu ochrony tak, aby zatrzymać jak największą grupę potencjalnych 'włamywaczy'.
Opisane metody dobrze wbudowują się w ten model postępowania. Oferują relatywnie duży wzrost ochrony systemu przy minimalnych kosztach i niewielkiej uciążliwości (są 'przeźroczyste' dla użytkowników a przy tym nie generują praktycznie fałszywych alarmów dla administratorów).
Na ten moment wydaje się że najistotniejszym zadaniem byłoby przetestowanie prototypu w rzeczywistym dużym środowisku produkcyjnym, co pozwoliłoby na ocenę jego wydajności oraz zapewne – wzrost skuteczności.