

Wstęp

Software Asset Management jest pojęciem ściśle związanym z dynamicznie rozwijającym się rynkiem oprogramowania. Pojawiające się różne rodzaje licencji określające sposób, w jaki można korzystać z poszczególnych rozwiązań, stworzyły przestrzeń dla firm takich, jak IBM, na stworzenie aplikacji ułatwiającej zarządzanie licencjami w systemach informatycznych przedsiębiorstw. W dzisiejszych realiach poważne przedsiębiorstwa muszą działać transparentnie, nie mogą sobie bowiem pozwolić na oskarżenia o nadużywanie lub nieprzestrzeganie wymogów licencyjnych. Wiąże się to nie tylko z kosztami ponoszonymi wskutek naliczanych kar, ale też ze stratami wizerunkowymi.

Każdego dnia pojawiają się nowe podatności i zagrożenia, na jakie wystawione są systemy informatyczne. Twórcom oprogramowania zależy, aby ich produkt był jak najbardziej na nie odporny. Wymusza to publikowanie nowych wersji oprogramowania, w którym poprawiono błędy i załatano luki umożliwiające niepożądane działanie systemu.

Niniejsza praca jest próbą połączenia powyższych pojęć w jednym narzędziu. Wykorzystując informacje gromadzone w aplikacji IBM Big Fix Inventory, można informować użytkownika o tym, że w jego systemie informatycznym zainstalowane jest oprogramowanie, w którym wykryta została podatność. Właściciel systemu na podstawie informacji o rodzaju podatności i o stopniu zagrożenia z nią związanego, może zdecydować jakie akcje należy podjąć, by zapewnić stabilność i bezpieczeństwo systemu.

Algorytm opracowany w ramach pracy umożliwia wykrycie w systemie informatycznym wersji oprogramowania znajdujących się na liście obciążonych podatnościami, publikowanej przez amerykańską agencję NIST. Jego działanie polega na dopasowaniu informacji o oprogramowaniu zainstalowanym i używanym w systemie, zebranych przez IBM Big Fix Inventory, z informacjami zebranymi przez NIST.

Cel pracy

Niniejsza praca stanowi badanie możliwości wykorzystania różnych algorytmów porównania tekstu oraz ich połączeń w celu wykrywania oprogramowania, które może powodować podatność systemu informatycznego na zagrożenia. W ramach pracy zostaną wykonane następujące czynności:

- Analiza poszczególnych algorytmów pod kątem danych wejściowych.
- Na podstawie powyższej analizy zostanie opracowany algorytm pozwalający na połączenie dostępnych informacji o podatnościach występujących w oprogramowaniu informatycznym oraz informacji o oprogramowaniu zainstalowanym w badanym systemie.
- Implementacja algorytmu w języku Python
- Przeprowadzenie testów skuteczności algorytmu
- Analiza otrzymanych wyników

Podczas opracowywania algorytmu szczególna uwaga zwrócona będzie w kierunku skuteczności wykrycia zagrożonej wersji oprogramowania zainstalowanej w systemie informatycznym monitorowanym przez narzędzie IBM Big Fix Inventory. Efektem pracy będą przedstawione wnioski dotyczące efektywności poszczególnych algorytmów porównania tekstu oraz ich połączeń opracowanych w ramach pracy. Wnioski zostaną wyciągnięte na podstawie wyników testów poszczególnych rozwiązań.