Spis treści

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 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.

2 Reprezentacja danych

2.1 Common Vulnerabilities and Exposures

Common Vulnerabilities and Exposures jest systemem zapewniającym publiczne, darmowe informacje na temat ekspozycji i podatności systemów informatycznych. Celem stworzenia CVE było utworzenie standardu, który ułatwiłby dzielenie się informacjami pomiędzy różnymi narzędziami i organizacjami zajmującymi się tematyką cyberbezpieczeństwa. Dzięki CVE możliwa stała się komunikacja pomiędzy tymi narzędziami, co pozwoliło na bardziej efektywną walkę z atakami na systemy informatyczne.

W CVE podatność zdefiniowana jest jako słabość w logice obliczeniowej, znaleziona w oprogramowaniu lub komponentach sprzętowych, które, jeśli zostałyby odkryte, niekorzystnie wpłynęły by na poufność, integralność lub dostępność systemu. Naprawa podatności zwykle wymaga zmian w kodzie programu, czasem również zmian w specyfikacji. Ekspozycją (TODO przemyśleć tłumaczenie "exposure") jest błędna konfiguracja systemu lub błąd w oprogramowaniu, który daje dostęp do informacji i zdolności systemu, użytych później przez hakera jako wytrych[2].

Lista Common Vulnerabilities and Exposures składa się CVE Entry, czyli wpisów CVE, które identyfikują poszczególne podatności.
Każdy wpis na liście CVE składa się z:

- CVE ID, który jest unikatowym identyfikatorem wpisu. Zapisany jest obecnie w formacie CVE-YYY-NNNN, gdzie YYYY jest liczbą oznaczającą rok publikacji wpisu, a NNNN jest numerem identyfikacyjnym w danym roku. pole NNNN składa się co najmniej z czterech cyfr, jednak od 2014 roku może ich być więcej niż cztery.

- Krótki opis podatności lub ekspozycji systemu, który może zwierać takie informacje, jak nazwa producenta podatnego oprogramowania.

- Odnośniki do innych, powiązanych wpisów CVE oraz wszelkie inne odnośniki, takie jak raporty i porady podane przez producenta.

Wpisy tworzone są przez zespół Mitre zajmujący się CVE oraz producentów oprogramowania, tak zwanych CVE Numbering Authority (CNA), których aplikacje zawierają podatności. CNA to organizacja zajmująca się dystrybuowaniem CVE ID. Ma ona ograniczone możliwości przypisywania CVE ID, dokładnie określone i udokumentowane, zwykle obejmujące programy lub sprzęt pochodzące maksymalnie od kilku producentów. Ograniczenia te stworzone są po to, by kompetencje zbyt wielu osób nie nachodziły na siebie, co mogłoby powodować chaos[3].

CVE ID jest polem wpisu CVE. Jest to unikatowy numer identyfikacyjny pozwalający jednoznacznie odróżnić od siebie wpisy na liście CVE. Składa się on z trzech podidentyfikatorów: identyfikatora listy, identyfikatora roku oraz identyfikatora wpisu w danym roku. Identyfikator listy może przyjmować dwie wartości:

- CVE - najczęściej używany, oznacza, że wpis widnieje na liście CVE
- CAN - pochodzący od angielskiego słowa "candidate", czyli kandydat, oznacza, że w momencie publikacji listy eksperci z CVE jeszcze nie zatwierdzili tego wpisu, jako zweryfikowanej podatności. Od pewnego momentu niemożliwym stało się, by zarząd CVE zajmował się każdym wpisem z osobna, gdyż było ich zbyt dużo, w związku z czym zaprzestano stosować identyfikatora CAN.

Kolejny podidentyfikator składa się z czterech cyfr i oznacza rok, w którym wpis został opublikowany. Możliwa jest sytuacja, że podatność została wykryta wcześniej, ale przy przypisywaniu CVE ID liczy się rok, w którym to ID zostało przypisane. Ostatni człon CVE ID zbudowany jest z co najmniej czterech cyfr oznaczających identyfikator unikatowy w skali roku. Od 2014 roku, w związku z rosnącą ilością wpisów na liście CVE, przyjmuje się, że ten człon może posiadać więcej niż cztery cyfry.

Opis wpisu CVE stworzony jest zwykle przez zespół CVE, organizację CNA lub osoby indywidualne zgłaszające odkrytą podatność. Opisy powinny zapewniać informacje na temat produktu w którym została wykryta podatność, identyfikatora wersji tego produktu oraz jego dostawcy. Powinien też zostać wyszczególniony typ podatności, jej wpływ na systemy informatyczne, a także informacje jak głębokiego dostępu do sytemu potrzebuje haker, by móc skorzystać z danej podatności.

Możliwe jest też wskazanie części kodu lub konkretnego komponentu odpowiadającego za podatność. Niestety, autor wpisu CVE nie zawsze jest w posiadaniu wszystkich informacji. Wynika to z faktu, że nie wszystkie z powyższych informacji są dostępne publicznie, co znacząco utrudnia wykonanie pełnego, oddającego całość problemu opisu. Z uwagi na znaczenie pola opisu z punktu widzenia niniejszej pracy magisterskiej, zostanie ono dodatkowo przeanalizowane w osobnej sekcji.

Odniesienia CVE jest polem, w którym wpisane są wszystkie istotne odnośniki do danej podatności. Mogą to być odnośniki do informacji o podobnych podatnościach, odnośniki do informacji o podatnościach powiązanych w jakiś sposób z danym wpisem CVE, jak również informacje o kontakcie do producenta danego oprogramowania lub sprzętu. Odnośniki te powinny dawać możliwość znalezienia rozwiązania poprawiającego daną podatność.

Data stworzenia wpisu, jak nazwa wskazuje, jest datą oznaczającą dzień, w którym powstał wpis. Nie oznacza ona jednak, że to w tym dniu został przypisany CVE ID. W przypadku wpisów tworzonych bezpośrednio przez zespół Mitre CVE, data ta pokrywa się z datą przypisania CVE ID. Jednak w przypadku wpisów tworzonych przez CNA dopuszczalna jest sytuacja, w której CVE ID jest zajęte przez daną organizację z góry, przez co nie zostaje ono użyte przez dłuższy czas. Dopiero po wykryciu konkretnej podatności w oprogramowaniu lub sprzęcie danej organizacji, wpisana zostaje data stworzenia wpisu i zaczyna on oficjalnie widnieć na liście CVE.

W wyjątkowych sytuacjach wpisy na liście CVE mogą zawierać dodatkową informację o stanie, w jakim aktualnie się znajdują. Są to stany, w których wpis nie jest jeszcze oficjalnie opublikowany na liście. Mogą to być:

- "RESERVED" - stan, w którym zajęty został identyfikator CVE ID, jednak szczegóły dotyczące danej podatności nie są jeszcze znane. Wpis może zostać opublikowany w każdej chwili, po uzupełnieniu brakujących informacji.
- "DISPUTED" - stan, w którym dana podatność jest jeszcze dyskutowana przez różne zainteresowane podmioty. Może to być na przykład sytuacja, w której zastrzeżenia do wpisu ma zespół CVE. W przypadku korzystania z wpisu oznaczonego jako DISPUTED należy, w poszukiwaniu najświeższych informacji, dodatkowo sprawdzić pole odniesień.

- "REJECT" - wpis odrzucony przez zespół CVE. Powodem odrzucenia wpisu może być na przykład powtórzenie istniejącego wcześniej wpisu.

2.2 Dane publikowane przez National Institute of Standards and Technology

National Institute of Standards and Technology (NIST) jest amerykańską agencją federalną, która zajmuje się szeroko pojętą metrologią. Wśród obszarów zainteresowań NIST znalazły się również podatności w systemach informatycznych[?]. National Vulnerability Database (NVD) jest wynikiem prac prowadzonych przez Instytut, które miały na celu usystematyzowanie i jak najlepsze opisanie podatności oraz ich wpływu na systemy informatyczne. NVD jest rządowym repozytorium, które za pomocą protokołu Secutiry Content Automation Protocol (SCAP) pozwala na ustandaryzowane automatyzowanie zarządzania podatnościami. NVD zawiera listy kontrolne bezpieczeństwa, wady oprogramowania związane z bezpieczeństwem, błędy konfiguracyjne, nazwy produktów oraz ocenę ich wpływu na system. Dane w NVD aktualizowane są na bieżąco, zaraz po przeanalizowaniu ich przez zespół ekspercki[4].

W protokole SCAP wykorzystuje się system Common Vulnerabilities and Exposures (CVE), który jest zarządzany i rozwijany przez firmę Mitre Corporation. Bazy danych NVD zawierają indeksy CVE rozszerzone dodatkowo o ocenę podatności według metryk Common Vulnerability Scoring System (CVSS). Pozwala to na podjęcie automatycznej decyzji o akcjach związanych z pojawieniem się podatności w systemie informatycznym. Baza danych NVD jest w pełni zsynchronizowana z listami CVE, co oznacza, że wszelkie poprawki i zmiany w listach CVE widoczne są natychmiast w bazie NVD. CVE znajdujące się w bazie NVD mogą znajdować się w jednym z siedmiu stanów:
Received - CVE został niedawno opublikowany do słownika CVE i został odebrany przez NVD
Awaiting Analysis - CVE został przeznaczony do analizy eksperckiej ze strony NVD, która zwykle zajmuje do 24 godzin
Undergoing Analysis - CVE jest obecnie analizowane przez ekspertów NVD
Analyzed - CVE zostało całkowicie przeanalizowane oraz wszystkie dodatkowe dane zostały do niego przypisane. Każda z analiz może być opisana

jednym z trzech podtypów:

Initial - używany by pokazać, że została przeprowadzona dopiero pierwsza analiza danego CVE

Modified - używana by pokazać, że została przeprowadzona analiza związana z modyfikacją danych w CVE

Reanalysis - używany by pokazać, że ponowna analiza została przeprowadzona z powodu innego niż modyfikacja CVE

Modified - CVE było poprawione przez źródło, które je opublikowało, co oznacza, że wyniki analizy NVD mogą być już nieaktualne

Deferred - jeśli CVE znajduje się w tym planie oznacza to, że nie jest planowana jego analiza

Rejected - CVE w tym stanie znajduje się w bazie danych NVD, nie wyświetla się jednak w wynikach wyszukiwania

Dodatkową kategorią indeksu CVE jest kategoria "Reserved". CVE występujące w słowniku, które są oznaczone takim polem, nie zostają włączone do bazy danych NVD[4].

W ramach rozwijania możliwości list CVE, baza danych NVD wprowadza nowe, zaawansowane metody wyszukiwania podatności. W National Vulnerability Database możliwe jest filtrowanie wpisów na podstawie systemu operacyjnego, na którym występuje dana podatność, czy kategoria podatności według opisu Common Weakness Enumeration. Co więcej, możliwe jest wyszukanie wszystkich podatności związanych z danym producentem lub produktem. Wyniki wyszukiwania można również ograniczyć w czasie. NVD wprowadza również możliwość wyszukania wszystkich wpisów CVE, których autorem jest dana jednostka CPE.

Dane znajdujące się w bazie danych NVD przygotowane są do pobrania z oficjalnej strony National Vulnerability Database w formacie XML w wersji 1.2.1, XML w wersji 2.0 lub JSON w wersji testowej 0.1 beta. Do dalszej analizy oraz zastosowania w testach algorytmu został wybrany format XML w wersji 2.0, ponieważ jest to najnowsza, stabilna wersja generowana jako pobieralny plik. W pliku tym znajdują się wszystkie informacje dostępne w ramach bazy danych NVD.

2.3 Analiza pola Opis CVE

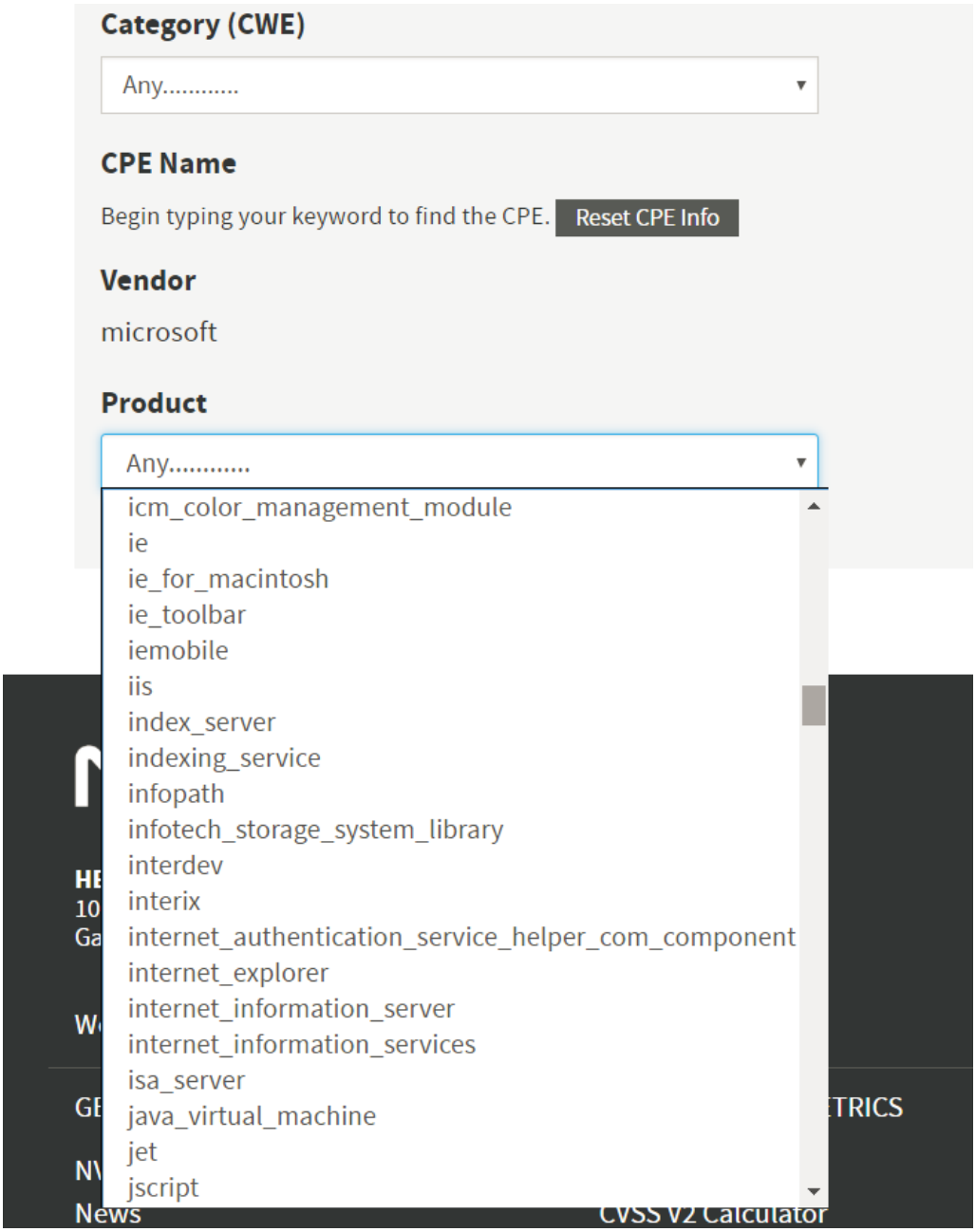
Jak wspomniano w sekcji 2.1, pole opisu we wpisach CVE jest jednym z kluczowych pól indeksów CVE, z uwagi na informacje, jakie powinno przechowywać. Szczególnie istotne są te dotyczące produktu, jego wersji oraz dostawcy danego produktu. Niestety brak odpowiedniego, zaplanowanego formatowania utrudnia ekstrakcję tych danych z opisu. Dodatkowo nie zawsze wpisy CVE są w pełni opisane. W wielu przypadkach podatności zgłaszane są przez podmioty, które nie mają dostępu do wszystkich pożądanych informacji. W dodatku liczba osób redagujących opisy sprawia, że pod uwagę należy brać styl pisania, różnice w pisowni lub nazewnictwie produktów, wersji i producentów.

Do dalszej, szczegółowej analizy wybrano kilka wpisów CVE dotyczących produktu Internet Explorer firmy Microsoft. Na ich przykładzie można pokazać najbardziej znaczące cechy pola opisu wpisów CVE oraz omówić ich konsekwencje i problemy przy ich automatycznym przetwarzaniu.

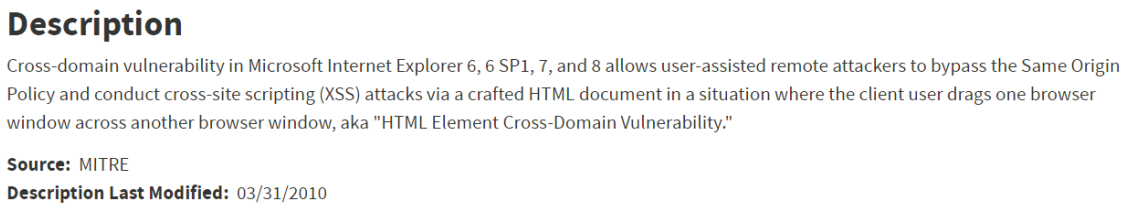
Wpisy wyszukano za pomocą bazy danych NVD, z uwagi na możliwość filtrowania wyników wyszukiwania na podstawie producenta i produktu. Już na tym etapie można było zauważyć, że nazewnictwo w różnych wpisach może przybierać odmienne formy. Na Rysunku 1 widać, że zależnie od autora wpisu wybrany produkt może być inaczej nazwany. W tym przypadku nazwy "ie" oraz 'internet explorer" oznaczają to samo oprogramowanie.

Na rysunkach 2-5 można zauważyć iż opisy poszczególnych podatności mają podobną, jednak nie identyczną strukturę. W dwóch przypadkach, na rysunkach 2 i 3, wpis zaczyna się od określenia zakresu wpływu podatności na system, następnie podana jest nazwa producenta oraz nazwa produktu. Kolejnym elementem opisu jest wskazanie wersji produktu, w których wykryto daną podatność. W dalszej części opisu następuje wyjaśnienie istoty danej podatności. W opisach znajdujących się na rysunkach 4 oraz 5 pominięto część dotyczącą zakresu wpływu.

Analizując poszczególne opisy można łatwo wywnioskować, że najbardziej różniącym się między poszczególnymi opisami elementem, jest określenie wersji produktu. W niektórych przypadkach opis ogranicza się do podania wersji produktu jako pojedynczej cyfry. W innych jest to doprecyzowane poprzez dopisanie po kropce dodatkowego identyfikatora wersji. Można spotkać również



Rysunek 1: Wybór filtrów w wyszukiwarce bazy danych NVD

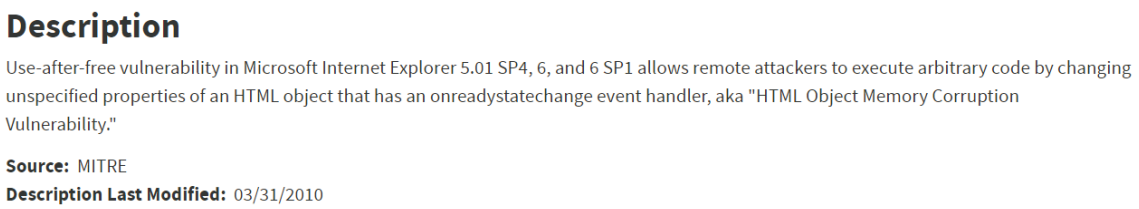


Description

Cross-domain vulnerability in Microsoft Internet Explorer 6, 6 SP1, 7, and 8 allows user-assisted remote attackers to bypass the Same Origin Policy and conduct cross-site scripting (XSS) attacks via a crafted HTML document in a situation where the client user drags one browser window across another browser window, aka "HTML Element Cross-Domain Vulnerability."

Source: MITRE
Description Last Modified: 03/31/2010

Rysunek 2: Pole opisu wpisu CVE-2010-0494



Description

Use-after-free vulnerability in Microsoft Internet Explorer 5.01 SP4, 6, and 6 SP1 allows remote attackers to execute arbitrary code by changing unspecified properties of an HTML object that has an onreadystatechange event handler, aka "HTML Object Memory Corruption Vulnerability."

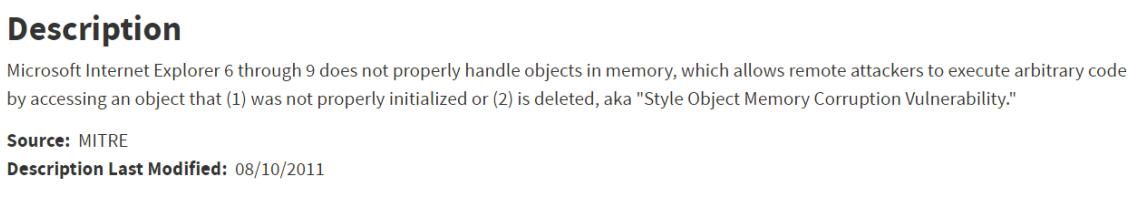
Source: MITRE
Description Last Modified: 03/31/2010

Rysunek 3: Pole opisu wpisu CVE-2010-0491

identyfikator wersji oznaczający pakiet, tak zwany Service Pack. W dodatku niektóre opisy zawierają mieszankę tych oznaczeń determinując poszczególne wersje z różną precyzją. Na rysunku 4 jest ukazany również przypadek, w którym podany jest zakres wersji produktu, w których można spotkać daną podatność. Wszystkie wspomniane różnice mogą powodować wysoki stopień trudności automatycznego przetwarzania opisu. Szczególnie biorąc pod uwagę fakt, że w omówionych przypadkach opisy dotyczyły tego samego produktu.

Na zdjęciach 6-9 przedstawiono opisy dotyczące różnych produktów firmy Microsoft. Przeanalizowane zostaną pod kątem różnic w opisach wpisów CVE dotyczących aplikacji wytworzonych i rozwijanych przez tego samego producenta.

Jak można zauważyć struktura opisu została zachowana jak w przypadku Microsoft Internet Explorer. To znaczy, że na początku opisu znajdują się informacje identyfikujące producenta i produkt, następnie podane są wersje produktu, w których występuje dana podatność. Zachowana została również część nazewnictwa poszczególnych wersji. Niektóre produkty mają jednak nowe



Description

Microsoft Internet Explorer 6 through 9 does not properly handle objects in memory, which allows remote attackers to execute arbitrary code by accessing an object that (1) was not properly initialized or (2) is deleted, aka "Style Object Memory Corruption Vulnerability."

Source: MITRE
Description Last Modified: 08/10/2011

Rysunek 4: Pole opisu wpisu CVE-2011-1964

Description

Microsoft Internet Explorer 8 and 9 does not properly handle objects in memory, which allows remote attackers to execute arbitrary code by accessing a deleted object, aka "OnRowsInserted Event Remote Code Execution Vulnerability."

Source: MITRE
Description Last Modified: 06/12/2012

Rysunek 5: Pole opisu wpisu CVE-2012-1881

Current Description

Double free vulnerability in Microsoft Outlook 2007 SP3 and 2010 SP1 and SP2 allows remote attackers to execute arbitrary code by including many nested S/MIME certificates in an e-mail message, aka "Message Certificate Vulnerability."

Rysunek 6: Pole opisu wpisu CVE-2013-3870. Produkt: Microsoft Outlook

Description

Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 Gold and SP1; Office 2011 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 do not properly handle memory during the opening of files, which allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel File Format Memory Corruption Vulnerability."

Source: MITRE
Description Last Modified: 05/08/2012

Rysunek 7: Pole opisu wpisu CVE-2012-0141. Produkt: Microsoft Excel

Description

Cross-site scripting (XSS) vulnerability in Microsoft SharePoint Server 2010 SP1 and SP2 and 2013 allows remote attackers to inject arbitrary web script or HTML via a crafted POST request, aka "POST XSS Vulnerability."

Source: MITRE
Description Last Modified: 09/11/2013

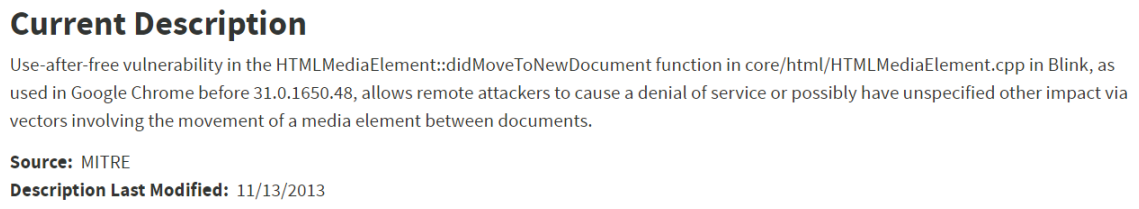
Rysunek 8: Pole opisu wpisu CVE-2013-3180. Produkt: Microsoft SharePoint Server

Description

Microsoft Communicator 2007 R2, Lync 2010, Lync 2010 Attendee, and Lync Server 2013 do not properly handle objects in memory, which allows remote attackers to execute arbitrary code via an invitation that triggers access to a deleted object, aka "Lync RCE Vulnerability."

Source: MITRE
Description Last Modified: 05/14/2013

Rysunek 9: Pole opisu wpisu CVE-2013-1302. Produkt: Microsoft Lync

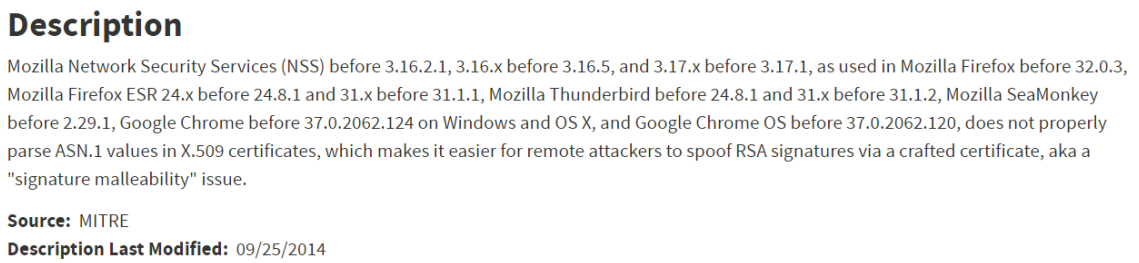


Current Description

Use-after-free vulnerability in the HTMLMediaElement::didMoveToNewDocument function in core/html/HTMLMediaElement.cpp in Blink, as used in Google Chrome before 31.0.1650.48, allows remote attackers to cause a denial of service or possibly have unspecified other impact via vectors involving the movement of a media element between documents.

Source: MITRE
Description Last Modified: 11/13/2013

Rysunek 10: Pole opisu wpisu CVE-2013-6622. Produkt: Google Blink



Description

Mozilla Network Security Services (NSS) before 3.16.2.1, 3.16.x before 3.16.5, and 3.17.x before 3.17.1, as used in Mozilla Firefox before 32.0.3, Mozilla Firefox ESR 24.x before 24.8.1 and 31.x before 31.1.1, Mozilla Thunderbird before 24.8.1 and 31.x before 31.1.2, Mozilla SeaMonkey before 2.29.1, Google Chrome before 37.0.2062.124 on Windows and OS X, and Google Chrome OS before 37.0.2062.120, does not properly parse ASN.1 values in X.509 certificates, which makes it easier for remote attackers to spoof RSA signatures via a crafted certificate, aka a "signature malleability" issue.

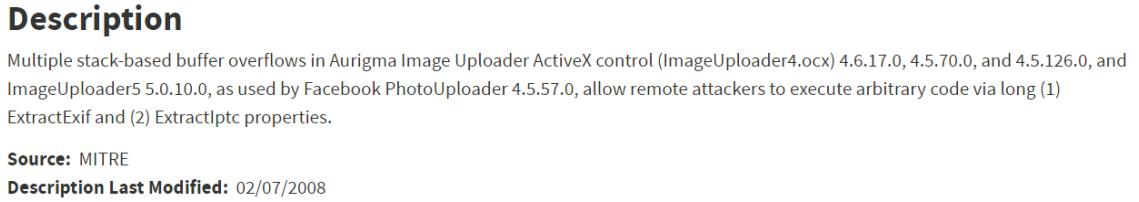
Source: MITRE
Description Last Modified: 09/25/2014

Rysunek 11: Pole opisu wpisu CVE-2014-1568. Produkt: Mozilla Network Security Services

identyfikatory wersji, którymi jest rok publikacji oraz występujące w opisie produktu Microsoft Excel słowo "Gold". Dodatkowym elementem opisu jest również wyszczególnienie w nim innych produktów, w których występuje opisywana podatność. Analiza rysunków 6-9 pozwala stwierdzić, że w ramach produktów wytwarzanych przez jedną firmę istnieją różnice w nazewnictwie poszczególnych wersji, nie jest ono jednak znacząco bardziej skomplikowane, niż nazewnictwo wersji w ramach jednego produktu.

Na rysunkach 10-13 znajdują się opisy podatności występujących w produktach stworzonych przez różnych producentów. Pozwoli to przeanalizować różnice pomiędzy strukturami opisów zapewnianych przez zróżnicowane środowiska.

Analizując rysunki 10-13 można stwierdzić, że nazewnictwo wersji poszczególnych produktów mogą przybierać zupełnie różne formy. Ale mają

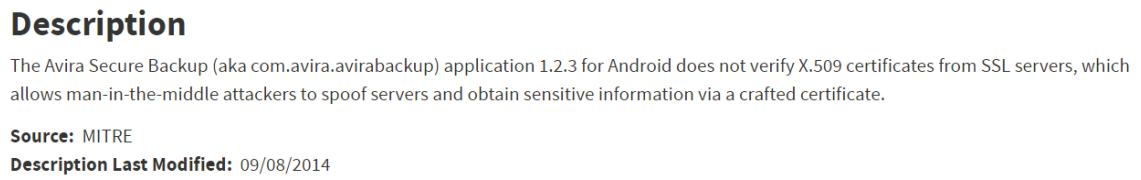


Description

Multiple stack-based buffer overflows in Aurigma Image Uploader ActiveX control (ImageUploader4.ocx) 4.6.17.0, 4.5.70.0, and 4.5.126.0, and ImageUploader5 5.0.10.0, as used by Facebook PhotoUploader 4.5.57.0, allow remote attackers to execute arbitrary code via long (1) ExtractExif and (2) ExtractIptc properties.

Source: MITRE
Description Last Modified: 02/07/2008

Rysunek 12: Pole opisu wpisu CVE-2008-0660. Produkt: Aurigma Image Uploader ActiveX



Rysunek 13: Pole opisu wpisu CVE-2014-5576. Produkt: Avira Secure Backup

również cechy wspólne. Takim łącznikiem jest na przykład stopniowanie oznaczeń pomiędzy poszczególnymi aktualizacjami. Gdy zmiany w danej wersji są niewielkie, zmienia się tylko ostatni człon w oznaczeniu wersji. Jednak gdy zmiany mają większy wpływ na działanie produktu, wówczas zmieniane są wyższe stopnie w nazwie wersji. Wśród różnic natomiast należy wymienić rzędy liczb definiujących kolejne wersje. Rząd ten, w wybranej próbce opisów, waha się od liczb jednocyfrowych, do liczb czterocyfrowych. Odmienna jest też liczba stopni poszczególnych wersji. Co więcej na rysunku 11 spotkać można oznaczenie "3.16.x" obejmujące wszystkie niższe stopniem aktualizacje wersji "3.16". Informacje te potwierdzają złożoność zagadnienia automatycznego dopasowania wersji produktów zainstalowanych w analizowanym przez IBM Big Fix Inventory systemie.

Bibliografia

[1] https://cve.mitre.org/about/faqs.html

[2] https://cve.mitre.org/about/terminology.html

[3] "CVE Overview for Prospective CNAs", Mitre Corporation, Version 1.0, 29 Wrzesień 2018

[4] https://nvd.nist.gov